

Acleirton de Sá e Sousa

**Desenvolvimento de uma aplicação de Escalas  
Médicas na Clínica de Reabilitação Santa Ana  
em Picos - PI**

Picos - PI  
27 de novembro de 2018

Acleirton de Sá e Sousa

## **Desenvolvimento de uma aplicação de Escalas Médicas na Clínica de Reabilitação Santa Ana em Picos - PI**

Monografia submetida ao Curso de Bacharelado em Sistemas de Informação da Universidade Federal do Piauí, Campus Senador Helvídio Nunes de Barros, como requisito parcial para obtenção de grau de Bacharel em Sistemas de Informação. Orientadora: Ma. Patricia Vieira da Silva Barros.

Universidade Federal do Piauí  
Campus Senador Helvídio Nunes de Barros  
Bacharelado em Sistemas de Informação

Picos - PI  
27 de novembro de 2018

**FICHA CATALOGRÁFICA**  
**Serviço de Processamento Técnico da Universidade Federal do Piauí**  
**Biblioteca José Albano de Macêdo**

**S725d**      Sousa, Acleirton de Sá e  
Desenvolvimento de uma aplicação de escalas médicas  
na clínica de reabilitação Santa Ana em Picos - PI / Acleirton de  
Sá e Sousa.– 2018.  
CD-ROM : il.; 4 ¾ pol. (44 f.)  
Trabalho de Conclusão de Curso (Curso Bacharelado em  
Sistemas de Informação) – Universidade Federal do Piauí,  
Picos, 2018.  
Orientador(A): Profª. Ma. Patrícia Vieira da Silva Barros.

1. Escala Médica. 2. Aplicativos. 3. Dispositivos  
móveis. 4. Sistemas web. I. Título.

**CDD 005.2**

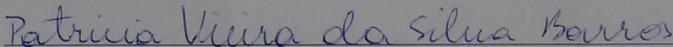
DESENVOLVIMENTO DE UMA APLICAÇÃO DE ESCALAS MÉDICAS NA CLÍNICA  
DE REABILITAÇÃO SANTA ANA EM PICOS – PI

ACLEIRTON DE SÁ E SOUSA

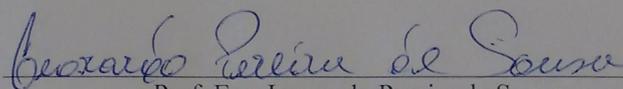
Monografia **aprovada** como exigência parcial para obtenção do grau de Bacharel em  
Sistemas de Informação.

Data de Aprovação

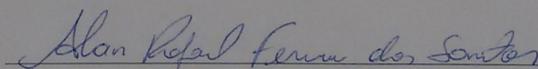
Picos – PI, 27 de novembro de 2018



Profª. Ma. Patricia Vieira da Silva Barros  
Orientadora



Prof. Esp. Leonardo Pereira de Sousa  
Membro



Prof. Me. Alan Rafael Ferreira dos Santos  
Membro

# Agradecimentos

Agradeço primeiramente à Deus nosso criador por me abençoar e me guiar todos os dias. Agradeço aos professores por repassar conteúdos essenciais para minha formação acadêmica, à Universidade Federal do Piauí pela oportunidade.

Agradeço a meu pai, minha mãe, meus irmãos, agradeço às minhas duas filhas Ana Clara e Iane Júlia que dos amores é o mais puro e mais lindo que existe, e a todos meus familiares e amigos(as) que ao longo dessa jornada estiveram comigo me dando forças para este sonho se tornar realidade.

*A tarefa não é tanto ver aquilo que ninguém viu, mas pensar o que ninguém ainda pensou sobre aquilo que todo mundo vê.*

*Arthur Schopenhauer*

# Resumo

Clínicas e Hospitais possuem como objetivo principal, prestar serviços de saúde para a população. Assim, precisam manter informações e se comunicar com as pessoas que necessitam de seu uso, com os mais variados tipos de problemas. Com isso, informações sobre os dias e horários que são oferecidos os serviços estão disponíveis apenas em murais, dificultando que as informações cheguem até as pessoas. Portanto, surge a necessidade da utilização de ferramentas que possam disponibilizar informações gerais para os usuários que utilizam os serviços de saúde. O presente trabalho possui o objetivo de desenvolver duas aplicações que funcionam de forma integrada, possuindo como principais funções a administração de informações sobre profissionais da saúde e suas escalas dentro da Clínica de Reabilitação Santa Ana, através de um sistema *web*, e a disponibilização dessas informações de forma atualizada, para a população com o aplicativo *mobile*. Essas aplicações, que compartilham o mesmo Banco de Dados, tendo informações atualizadas pelos responsáveis da clínica na aplicação *web* e apresentadas para toda a população interessada que possui *smarthphone* com conexão com a *Internet*, que tenha instalado o aplicativo. Essas duas aplicações têm como intuito contribuir com a comunicação entre a clínica e usuários que terão facilidade de obter as informações sobre serviços disponíveis e escalas dos profissionais. Através da aplicação de questionário, pode-se concluir que as aplicações *mobile* e sistema *web* atingiram os objetivos esperados para funcionamento.

**Palavras-chaves:** Escala Médica. Aplicativos. Dispositivos móveis. Sistemas *web*.

# Abstract

Clinics and Hospitals have the main objective of providing health services to the population. Thus, they need to keep information and communicate with people who need it, with the most varied types of problems. With this, information about the days and times that the services are offered, are only available on murals, making it difficult for the information to reach people. Therefore, there is a need to use tools that can provide general information to users who use health services. The present work aims to develop two applications that work in an integrated way, having as main functions the administration of information about health professionals and their scales within the Santa Ana Rehabilitation Clinic, through a web system, and the provision of this information updated for the population with the mobile application. These applications, which share the same database, have information updated by the heads of the clinic in the web application and presented to all interested population that owns smarthphone with Internet connection, that has installed the application. These two applications are intended to contribute to the communication between the clinic and users who will be able to obtain information about available services and scales of professionals. Through the application of a questionnaire, it can be concluded that the mobile applications and web system reached the expected objectives for operation.

**Key-words:** Medical Scale. Applications. Mobile devices. Web systems.

# Lista de ilustrações

Figura 1 – Diagrama de caso de uso. . . . .	26
Figura 2 – Diagrama de classes. . . . .	27
Figura 3 – Tela de login do administrador. . . . .	29
Figura 4 – Área do administrador - logado. . . . .	30
Figura 5 – Administrador adiciona médicos. . . . .	30
Figura 6 – Profissionais cadastrados no sistema. . . . .	31
Figura 7 – Administrador adiciona escalas. . . . .	31
Figura 8 – Tela inicial do Aplicativo e Tela Especialidades. . . . .	32
Figura 9 – Profissionais por especialidade e Escala. . . . .	33
Figura 10 – Interface completa do aplicativo. . . . .	33

# Lista de tabelas

Tabela 1 – Comparativo entre Trabalhos Relacionados. . . . .	24
--	----

# Lista de abreviaturas e siglas

API	Application Programming Interface
App	Aplicativo
C++	Linguagem de programação
E.M	Escala Médica
HTTP	Hypertext Transfer Protocol
I/O	Entrada e Saída
iOS	Iphone Operating System
JSON	JavaScript Object Notation
NoSQL	Non Relational
PWA	Progressive Web App
REST	Representational State Transfer
UML	Unified Modelling Language
URL	Uniform Resource Locator
XML	Extensible Markup Language

# Sumário

<b>1</b>	<b>Introdução</b>	<b>13</b>
1.1	Contexto	14
1.2	Objetivos	14
1.2.1	Geral	14
1.2.2	Específicos	14
1.3	Justificativa	15
1.4	Metodologia	15
1.5	Contribuições	15
<b>2</b>	<b>Referencial Teórico</b>	<b>17</b>
2.1	Engenharia de Software	17
2.2	Qualidade de <i>Software</i>	18
2.3	Avaliação de <i>Software</i>	18
2.4	Android	18
2.5	Ferramentas para Desenvolvimento de Aplicações <i>Web</i>	19
2.5.1	REST	19
2.5.2	JavaScript	20
2.5.3	TypeScript	20
2.5.4	Node.js	20
2.5.5	Framework Express.js	21
2.5.6	Angular	21
2.6	Ferramentas para Desenvolvimento de Aplicações <i>Mobile</i>	22
2.6.1	Ionic	22
2.6.2	Cordova	22
2.7	MongoDB	23
2.8	Ferramentas e métodos escolhidos	23
2.9	Trabalhos Relacionados	24
<b>3</b>	<b>Proposta de Desenvolvimento das Aplicações de Escalas Médicas</b>	<b>26</b>
3.1	Etapa Desenvolvimento	26
3.2	Modelagem	26
3.2.1	Caso de Uso	26
3.2.2	Diagrama de Classes	27
<b>4</b>	<b>Resultados e Discussões</b>	<b>29</b>
4.1	Sistema <i>Web</i>	29

---

4.1.1	Área administrativa do Aplicativo . . . . .	29
4.2	Sistema <i>Mobile</i> . . . . .	32
4.2.1	Interface do aplicativo <i>mobile</i> . . . . .	32
<b>5</b>	<b>Testes . . . . .</b>	<b>34</b>
<b>6</b>	<b>Conclusões e Trabalhos Futuros . . . . .</b>	<b>37</b>
	<b>Referências . . . . .</b>	<b>38</b>
	<b>Apêndices</b>	<b>40</b>
	<b>APÊNDICE A Questionário . . . . .</b>	<b>41</b>
	<b>APÊNDICE B Código de atualização de profissionais . . . . .</b>	<b>42</b>
	<b>APÊNDICE C Código Escala dos profissionais . . . . .</b>	<b>44</b>

# 1 Introdução

A evolução das tecnologias fez com que atualmente aparelhos e computadores, que antes custavam valores muito altos e possuíam volume físico enorme, se resumissem em pequenas máquinas, com grandes tecnologias embutidas em chips muito pequenos e com valores acessíveis para grande parte da sociedade.

A computação tem avançado muito desde o século XX, e esse avanço continua expressivo no século XXI. Nos dias atuais, a evolução das tecnologias, uso de computadores pessoais e aparelhos celulares *smarthphones*, e a importância dos sistemas *web*<sup>1</sup> e aplicativos, tornou-se cada vez mais comum o uso dessas aplicações para diversas funções do dia a dia.

Nos últimos anos, muitas das funções de um computador vem sendo gradativamente transferidas para dispositivos que cabem na palma da mão (MOORE, 2007). Com tecnologias inseridas nesses aparelhos celulares *smarthphones*, dispositivos móveis inteligentes, que estão cada vez mais presentes na sociedade, o crescimento do uso de aplicativos desenvolvidos para esses *smarthphones* e de sistemas *web* responsivos<sup>2</sup>, tem sido consideravelmente grande nos últimos anos, fazendo assim com que esses estejam cada vez mais presentes na vida profissional e pessoal, tornando mais simples e eficiente a resolução de questões habituais e da realização de tarefas cotidianas da sociedade.

Com a crescente utilização de dispositivos móveis, tornou-se comum as empresas possuírem seus sites e aplicativos próprios, disponibilizando-os em servidores na *Internet* e em lojas virtuais, para que as pessoas possam usar e conhecer seus produtos e serviços, aproximando assim a empresa de seu público alvo e ajudando essa a alcançar seus objetivos. Tornando-se comum o uso de aplicações nessas empresas para ter um maior contato com as pessoas, disponibilizando informações de forma mais ampla e práticas.

Que é o caso das aplicações desenvolvidas no presente trabalho, que teve como objetivo desenvolver um sistema *web* administrativo e um aplicativo *mobile*, para a Clínica de Reabilitação Santa Ana, estabelecida na cidade de Picos-PI, para uma melhor forma de disponibilidade de informação e comunicação para com a clínica.

Visando a contribuição e a melhoria das informações da Clínica, buscou-se o desenvolvimento dessas aplicações que trabalharão em conjunto no provimento de informações sobre o atendimento na clínica, disponíveis a usuários que tenham instalado o aplicativo no seu *smartphone*.

---

<sup>1</sup> Palavra inglesa pela qual a rede mundial de computadores *internet* se tornou conhecida.

<sup>2</sup> Adapta-se em diferentes tamanhos de telas.

## 1.1 Contexto

A Clínica de Reabilitação Santa Ana tem a necessidade de informar para seus usuários informações sobre horários de atendimento dos médicos e a especialidade dos mesmos que atendem em cada dia e horário.

A problemática encontra-se na dificuldade que existe pelos usuários em, por exemplo, saber o dia e horário de atendimento de um profissional na clínica, muitas vezes sendo necessário realizar ligação telefônica, ou ir até a clínica.

Atualmente, essas informações são repassadas de maneiras diferentes, por exemplo: através de papéis disponibilizados em mural da clínica, por meio de anúncios em meios de comunicação, ou até mesmo em *sites*.

Com a grande utilização de aparelhos celulares conectados à *Internet*, e por meio da grande parte da população necessitar dos serviços de saúde, as empresas que os provêm estão cada vez mais aderindo a meios tecnológicos para informar e manter contato com seus usuários.

Com isso, os sistemas *web* e aplicativos *mobile* são meios tecnológicos que permitem uma grande abrangência da população. Assim, esse problema pode ser resolvido usando o aplicativo *mobile* que disponibiliza as informações referentes ao dia e horário de atendimento de cada profissional na clínica.

## 1.2 Objetivos

### 1.2.1 Geral

Desenvolver uma aplicação de Escalas Médicas para a disponibilização de informações sobre a escala de profissionais da clínica de Reabilitação Santa Ana.

### 1.2.2 Específicos

- Coletar informações de escala da clínica;
- Verificar tecnologias existentes;
- Desenvolver um sistema *web* administrativo;
- Desenvolver um aplicativo *mobile* para *Android*;
- Criar um Banco de Dados para armazenar informações inseridas no sistema administrativo e disponibilizar no aplicativo *mobile*;
- Disponibilizar para os usuários a escala médica da clínica através do aplicativo E.M Escala Médica.

### 1.3 Justificativa

A dificuldade na obtenção de informações sobre a distribuição de plantões médicos, na Clínica de Reabilitação Santa Ana, principalmente para pacientes que habitam em localidades distantes da clínica, foi motivação para a criação da plataforma E.M Escala Médica. Essa aplicação tornará acessíveis informações essenciais para quem precisa saber dias e horários de atendimentos médicos com especialidades e datas específicas.

O presente trabalho se justifica da necessidade de haver informações disponíveis de forma mais acessível aos usuários, quando, existe dificuldades por parte dos usuários a obtenção de informações sobre os dias e horários de atendimento de profissionais na Clínica de Reabilitação Santa Ana.

### 1.4 Metodologia

A base da plataforma E.M Escala Médica será desenvolvida utilizando os princípios da arquitetura REST, pois essa arquitetura disponibiliza princípios que permite a criação de aplicações com o poder da interoperabilidade, ou seja, permite a comunicação, sem falhas, com outras aplicações, não importando a tecnologia utilizada nesses sistemas. Um quesito necessário para a E.M Escala médica, já que essa precisa funcionar com integrações entre um sistema *web* e um aplicativo *mobile* que compartilham a mesma interface de provimento de dados.

A escolha das tecnologias utilizadas para criação da plataforma é baseada nos recursos disponibilizados por cada uma. Foram selecionadas tecnologias novas, criadas e recomendadas por grandes empresas (como a *Google*, por exemplo), e que possuem recursos nativos para trabalhar com o protocolo HTTP, que é essencial para o desenvolvimento utilizando a arquitetura REST.

Seguindo modelo de desenvolvimento ágil para organizar o trabalho de implementação da aplicação, o método ágil utilizado será o *Scrum*, pois esse método propõe ciclos de desenvolvimento bem definidos com entrega de funcionalidades a cada ciclo (chamado de *Sprint*). Assim, é possível criar e testar cada parte da plataforma, tanto separadamente (ao final de cada *Sprint*), como após a integração entre API, aplicação *web* e aplicativo *mobile*.

### 1.5 Contribuições

O presente trabalho em seu todo disponibiliza para a sociedade acadêmica a apresentação de um modelo arquitetural de desenvolvimento (REST) que busca a criação de aplicações de forma modularizada com objetivo de tornar mais simples o desenvolvimento e principalmente a manutenção dessas aplicações, já que possuem módulos bem definidos.

Podendo tanto o modelo arquitetural, como as tecnologias apresentadas, serem observadas e utilizadas em trabalhos acadêmicos futuros com base neste trabalho.

Além da contribuição para a sociedade acadêmica, a plataforma E.M Escala Médica contribui com a facilitação na obtenção de informações sobre especialidades médicas e escalas médicas na Clínica de Reabilitação Santa Ana para usuários, principalmente para as pessoas que possuem dificuldade em ir até a clínica pessoalmente, assim, apresentando contribuições para a sociedade em geral.

## 2 Referencial Teórico

O presente capítulo é voltado para a discussão dos conceitos e assuntos referentes a este trabalho. Dividido em seções que mostram conceitos das tecnologias utilizadas para o desenvolvimento da aplicação E.M Escala Médica.

### 2.1 Engenharia de Software

Área da computação voltada à especificação, desenvolvimento, manutenção e criação de *software*, com a aplicação de tecnologias e práticas de gerência de projetos e outras disciplinas, visando organização, produtividade e qualidade. Ela trata de aspectos relacionados ao estabelecimento de processos, métodos, técnicas, ferramentas e ambientes de suporte ao desenvolvimento de *software*.

Qualquer projeto é formado por processos e estes são constituídos de atividades que, fazendo uso de ferramentas, técnicas e artefatos, são executados de forma organizada e/ou sincronizada.

Na produção de software, os primeiros e mais importantes passos estão na definição de requisitos, nas fases finais na implementação. A fase de especificação inclui o desenho do profissional com suas competências sociais e técnicas, ou seja, as capacidades que devem possuir para realizar atividades de desenvolvimento de software (CUNHA, 2015).

“Engenharia de Software é a aplicação de abordagens sistemáticas, disciplinadas e quantificáveis no desenvolvimento e manutenção de software. Desta forma, se preocupa em como realizar as diversas atividades envolvidas no processo de desenvolvimento de software de forma que se tenha um produto elaborado com maior qualidade e menor custo. Neste contexto, é uma área de conhecimento bastante abrangente envolvendo desde atividades mais técnicas como programação até áreas mais gerenciais como controle de qualidade nos processos utilizados.” (ARAÚJO; SPÍNOLA, 2007).

As competências técnicas do Engenheiro de *Software* são estabelecidas pelas Diretrizes Curriculares de Computação. Algumas delas são: Analisar e selecionar tecnologias adequadas para a construção de *software*; Avaliar a qualidade de sistemas de *software*; Integrar sistemas de *software*; Gerenciar projetos de *software* conciliando objetivos conflitantes, com limitações de custos, tempo e com análise de riscos; Aplicar adequadamente normas técnicas; Qualificar e quantificar seu trabalho baseado em experiências e experimentos; Exercer múltiplas atividades relacionadas a *software*, como: desenvolvimento, evolução, consultoria, negociação, ensino e pesquisa; Conceber, aplicar e validar princípios, padrões e boas práticas no desenvolvimento de *software*; Analisar e criar modelos relacionados ao desenvolvimento de *software*; Identificar novas oportunidades de negócios e desenvolver soluções inovadoras (CUNHA, 2015).

## 2.2 Qualidade de *Software*

Se perguntarmos a um usuário, provavelmente, ele dirá que um produto de *software* é de boa qualidade se ele satisfizer suas necessidades, sendo fácil de usar, eficiente e confiável. Essa é uma perspectiva externa de observação pelo uso do produto. Por outro lado, para um desenvolvedor, um produto de boa qualidade tem de ser fácil de manter, sendo o produto de *software* observado por uma perspectiva interna. Já para um cliente, o produto de *software* deve agregar valor a seu negócio (qualidade em uso).

[Sommerville \(2003\)](#) afirma que a qualidade de um *software* é medida pelo grau em que atende seus requisitos, seu comportamento quando em funcionamento, a estrutura e a organização do programa fonte e também a documentação associada.

A qualidade de *software* é uma área de conhecimento da engenharia de *software* que objetiva garantir a qualidade do *software* através da definição e normatização de processos de desenvolvimento.

## 2.3 Avaliação de *Software*

É uma etapa fundamental e importante da finalização do *software*, para que sejam assegurados os objetivos no início da produção do *software*.

[Webber et al. \(2009\)](#) citam que a avaliação de *software* se apóia em técnicas utilizadas em avaliação de *software* para uso geral. Porém, para esta finalidade específica, a avaliação deve contemplar aspectos como as teorias que embasam o desenvolvimento do *software* e a adequação contextual. Assim, as métricas devem ser significativas, ou seja, os resultados obtidos devem agregar informação útil à avaliação da qualidade.

De acordo com [Koscianski e Soares \(2007\)](#), além da usabilidade, aspectos como funcionalidade, manutenibilidade, confiabilidade, eficiência e portabilidade devem ser avaliados. Esses elementos podem ser divididos em atributos menores, para atender as especificidades de cada programa.

## 2.4 Android

O *Android* é o sistema operacional móvel da *Google* é o principal sistema operacional presente na maioria dos *smarthphones*. O *Android* fornece uma estrutura de aplicativo avançada que permite criar aplicativos para dispositivos móveis em um ambiente de linguagem *Java*.

Aplicativos para *Android*<sup>1</sup> são criados como uma combinação de componentes distintos que podem ser invocados individualmente. Por exemplo: uma *atividade* individual fornece

---

<sup>1</sup> Baseado na documentação oficial disponível em [AndroidDevelopers...](#) (2018)

uma única tela para uma interface do usuário e um serviço realiza o trabalho no segundo plano de forma independente.

Em um componente, é possível iniciar outro componente usando uma *intent*. É possível iniciar um componente em um aplicativo diferente, como uma atividade em um aplicativo de mapas para exibir um endereço. Esse modelo fornece vários pontos de entrada para um único aplicativo e permite que qualquer aplicativo se comporte como o “padrão” de um usuário para uma ação que outros aplicativos podem invocar.

O *Android* fornece uma estrutura de aplicativo adaptativa que permite fornecer recursos exclusivos para diferentes configurações de dispositivos. Por exemplo: é possível criar diferentes arquivos XML de *layout* para diversos tamanhos de tela e o sistema determina qual *layout* deverá aplicar com base no tamanho da tela do dispositivo atual.

## 2.5 Ferramentas para Desenvolvimento de Aplicações *Web*

Sistemas projetados para utilização através de um navegador com uso da *internet*. Pode ser projetado para diversas finalidades.

O número de utilizadores de aplicações *Web* atualmente é incalculável. As previsões para o aumento desse número de usuários, são cada vez mais otimistas. Com esse enorme número de pessoas utilizando serviços de aplicações *Web*, fica complicado de satisfazer ao gosto de todos os usuários (PRESSMAN, 2002).

Para o desenvolvimento de sistemas *web*, existem no mercado atual, muitas opções de linguagens de programação e *frameworks*. Cada uma dessas tecnologias pode ser escolhida dependendo do tipo de aplicação que se deseja desenvolver. O mercado nos últimos anos vem pedindo cada vez mais aplicações que sejam independentes, ou seja, que possam ser integradas com outras independentes das tecnologias utilizadas em cada uma delas.

Essas aplicações funcionam como uma *interface*, que podem ser acessadas e integradas com qualquer outra aplicação, que possua os endereços de seus métodos e ações.

Um dos padrões mais utilizado para a criação dessas aplicações, conhecidas como *Application Programming Interface* (API), é o RESTful.

### 2.5.1 REST

Segundo Richardson e Ruby (2007), REST é um conjunto simples de princípios que todos os dias os desenvolvedores de *softwares* para *web*, podem utilizar para criar aplicativos em rede.

O RESTful é o padrão que aplica os princípios REST. Ele trabalha com métodos *Hypertext Transfer Protocol* - HTTP (em português Protocolo de Transferência de Hipertexto) e com o formato de dados JSON.

É através dos métodos HTTP (GET, POST, PUT, DELETE, entre outros), que as APIs desenvolvidas com esse padrão, permitem serem acessadas por outras aplicações, ou seja, é através desses métodos que elas recebem e enviam dados.

## 2.5.2 JavaScript

O *JavaScript* foi criada pela *Netscape* em parceria com a *Sun Microsystems* para fornecer um meio de adicionar interatividade as páginas *web*. Sua primeira versão, o *JavaScript* 1.0, foi lançada em 1995 e implementada em março de 1996 no navegador *Netscape Navigator* 2.0 quando o mercado era dominado pela *Netscape*.

O *JavaScript*<sup>2</sup> é uma excelente linguagem. Atualmente a principal linguagem para programação client-side em navegadores web. Bastante utilizada do lado do servidor através de ambientes como o *node.js*.

## 2.5.3 TypeScript

O *TypeScript* é uma linguagem *open source*, criada e desenvolvida pela *Microsoft*, que é compilada para *JavaScript*. Apresenta-se como uma solução para alguns dos problemas mais inerentes aos projetos *JavaScript* com alguma dimensão. A possibilidade de especificação de tipos em *TypeScript* contribui para reduzir o número de bugs no código final e para aumentar a produtividade do programador. Para além dos tipos base introduzidos, a linguagem permite a criação de novos tipos personalizados (ABREU, 2017).

O *TypeScript* compila para *JavaScript*, para que seja possível ser entendido pelos navegadores. É a linguagem padrão utilizada no Angular.

## 2.5.4 Node.js

*Node.js* foi criado em 2009 por Ryan Dahl com a ajuda inicial de 14 colaboradores. Essa tecnologia foi criada para resolver o problema de tarefas do tipo bloqueantes, que paralisam um processo enquanto utilizam um I/O (entrada e saída) no servidor. Trata-se de uma plataforma altamente escalável e de baixo nível, pois a programação é realizada diretamente com diversos protocolos de rede e *internet* ou utilizando bibliotecas que acessam recursos do sistema operacional, principalmente recursos de sistemas baseados em Unix (PEREIRA, 2013).

O *Node.js* é uma plataforma construída para a linguagem *JavaScript*, sendo essa sua linguagem de programação padrão.

Teixeira (2013) explica que o Javascript contribuiu muito para o sucesso do Node.js e cita alguns motivos para o uso do mesmo:

- Facilidade no uso: O Node.js é fácil de instalar e configurar;

---

<sup>2</sup> <https://www.javascript.com>

- *Javascript*: A linguagem *Javascript* é largamente utilizada em todo o mundo, o que possibilita uma melhor adaptação ao Node;
- O Node é simples: Os desenvolvedores do Node buscaram tornar o mais simples possível o desenvolvimento utilizando a plataforma.

### 2.5.5 Framework Express.js

O *Express*<sup>3</sup> fornece uma camada fina de recursos fundamentais para aplicativos da web, sem obscurecer os recursos do Node.js.

Este é o *framework web* mais popular para *Node.js*. Usando um conjunto robusto de recursos, os desenvolvedores podem criar, *multi-page single*, e aplicações *web*.

O *Express.js* fornece vários recursos essenciais para criação de aplicações RESTful. Sendo os principais deles, as bibliotecas para trabalhar com recursos HTTP, permitindo trabalhar diretamente com todos os seus métodos, para busca de dados (GET), inserção de dados (POST), atualização (PUT) e exclusão (DELETE) e vários outros, e os *middlewares*, responsáveis por tratamentos de dados, validações e roteamento de *URLs* (*Uniform Resource Locator*, em português Localizador Uniforme de Recursos) que possibilitam o acesso aos métodos da API, juntamente com os métodos *HTTP*.

### 2.5.6 Angular

O *Angular*<sup>4</sup> é uma plataforma que facilita a criação de aplicativos para *web*, combina modelos declarativos, injeção de dependência, ferramentas de ponta a ponta e práticas recomendadas integradas para resolver desafios de desenvolvimento. Angular permite que os desenvolvedores criem aplicativos para hospedagem na *web*, em dispositivos móveis ou na área de trabalho.

O *Angular* é um *Framework JavaScript*, para criação de aplicações *Front-End* que permite criar estruturas de páginas completas de sistemas, além de possibilitar de forma razoavelmente simples, a conexão com *APIs RESTful*. Sendo responsável em muitos casos por enviar, receber e apresentar todas as informações do sistema.

A construção do *Angular* enfatiza pontos importantes do desenvolvimento. São eles, segundo Freeman (2014):

- Expansão: facilidade no entendimento de complexos aplicativos desenvolvidos, uma vez que se entenda o básico do *Angular*. Isso permite que a aplicação cresça e seja incrementada conforme a necessidade do usuário;
- Manutenção: a modularização da aplicação facilita o *debug* do sistema, simplificando a manutenção em longa data;

<sup>3</sup> <https://expressjs.com/pt-br>

<sup>4</sup> <https://angular.io/>

- Testes: capacidade de encontrar falhas e corrigir estas falhas de maneira simples e rápida antes da implantação ao usuário.

## 2.6 Ferramentas para Desenvolvimento de Aplicações *Mobile*

Aplicativo ou App ou programa de *software* são programas que podem ser instalados em celulares *smarthphones* para inúmeras finalidades, como comunicação, serviços, informações, entretenimento, etc. Cada aplicativo desempenha objetivos específicos para o qual foi desenvolvido.

A utilização de aplicações móveis tem registado um crescimento exponencial, o qual pode ser explicado pela oferta de uma diversidade de ferramentas capazes de responder às mais variadas necessidades dos utilizadores por meio de um simples clique (TAROUÇO, 2013).

### 2.6.1 Ionic

O *Ionic*<sup>5</sup> foi criado em 2013 pela Drifty Co. O foco do projeto foi a construção de uma estrutura que se concentrasse no desempenho com os padrões modernos da *web*.

O *Ionic* usa o *TypeScript* para seu código e facilita a criação de aplicativos móveis e web progressivos de alto desempenho (ou PWAs) com aparência e beleza em qualquer plataforma ou dispositivo.

O *Ionic* é um *Framework javascript* que utiliza o *Angular* com a linguagem *typescript*, permite a criação de aplicativos híbridos<sup>6</sup>. Trabalha em conjunto com o *framework Cordova*.

### 2.6.2 Cordova

Cordova é uma plataforma de desenvolvimento móvel com APIs (Application Programming Interface) que permitem que o desenvolvedor acesse funções nativas do dispositivo, como câmera ou acelerômetro. Quando se cria uma aplicação com o Ionic, por padrão não são adicionados plugins do Cordova para que o aplicativo não fique sobrecarregado com plugins que não serão utilizados. O desenvolvedor pode adicionar plugins em sua aplicação de acordo com a necessidade através de alguns comandos simples e uma conexão com a internet (GONÇALVES, 2017).

<sup>5</sup> <https://ionicframework.com/what-is-ionic>

<sup>6</sup> Pode ser executado em sistemas operacionais diferentes

## 2.7 MongoDB

Usado para as Aplicações *Web* e *Mobile* o MongoDB<sup>7</sup> é um banco de dados de documentos de código aberto projetado para facilidade de desenvolvimento e dimensionamento. O MongoDB possui escalabilidade e flexibilidade com consulta e indexação, e facilidade de desenvolvimento e dimensionamento.

É um *software* de banco de dados orientado a documentos e multiplataforma, escrito na linguagem C++. Classificado como um programa de banco de dados NoSQL, o MongoDB usa documentos semelhantes a JSON com esquemas.

Banco de dados NoSQL, utilizado para armazenar os dados do sistema, de forma não estruturada e utilizando dados no formato JSON.

## 2.8 Ferramentas e métodos escolhidos

A aplicação E.M Escala Médica é formada pela integração de um sistema *web* administrativo e de um aplicativo *mobile*, que utilizam o mesmo banco de dados e consequentemente a mesma API.

A decisão de fazer o desenvolvimento de uma API única para prover todos os serviços da aplicação, foi baseada na facilidade e ganho de tempo, em ter que criar apenas uma base para servir todos os recursos necessitados pelas aplicações clientes (*Front-End* administrativo e aplicativo *mobile*), além de permitir que em atualizações futuras sejam utilizadas quaisquer linguagens para criação de ferramentas de integração com os serviços da API.

Para o desenvolvimento da aplicação foram utilizadas várias tecnologias, que foram integradas para proporcionar maior poder tecnológico para a criação das funcionalidades necessárias. A seguir serão descritas as tecnologias e sua importância para este trabalho:

- Banco de Dados MongoDB

Banco de dados NoSQL, utilizado para armazenar os dados do sistema, de forma não estruturada e utilizando dados no formato JSON. Usado para armazenar informações inseridas no sistema administrativo e disponibilizar no aplicativo *mobile*.

**Para a Aplicação Web** (Sistema administrativo para inserção e atualização de informações.)

- Uso da ferramenta *Angular*

O *framework JavaScript Angular* na sua versão 4, foi utilizado para criar o *Front-And* administrativo da aplicação. Ele proporcionou além de recursos para criação das páginas e a poderosa linguagem *TypeScript*, bibliotecas *HTTP Client*, para conexão

---

<sup>7</sup> <https://www.mongodb.com>

com a API. O *Angular* tem chamado a atenção por sua leveza e pela facilidade com a qual trabalha os conceitos e os paradigmas do modelo moderno.

- Ferramenta *Node.Js*

A plataforma *Node.js* por ser uma tecnologia criada especialmente para ser utilizada no desenvolvimento de aplicações em rede, foi a principal ferramenta utilizada, juntamente com sua linguagem de programação *JavaScript*, que permitiram a criação de um elaborado serviço de conexão, inserção, atualização e buscas em banco de dados.

- *Framework Express.js*

O *framework express* foi utilizado principalmente para prover recursos de roteamento para os métodos da API e para permitir a utilização dos métodos HTTP.

**Para a Aplicação *Mobile*** (Aplicativo para disponibilizar informações sobre a escala de médicos da clínica para os usuários.)

- Uso da ferramenta *Ionic*

A ferramenta *Ionic* foi utilizada na sua versão 4 para criação do aplicativo *mobile* para o sistema operacional *Android*. O *Ionic* usa em sua estrutura, o *Angular* e é responsável pela conexão entre aplicativo e API.

## 2.9 Trabalhos Relacionados

Esta seção apresenta os trabalhos relacionados ao presente trabalho. Para melhor visualização, a Tabela 1 elenca três fatores comparativos: Facilidade de uso, Multiplataforma e Sistema operacional. A seguir são comentados.

Tabela 1: Comparativo entre Trabalhos Relacionados.

Trabalho	Facilidade de uso	Multiplataforma	Sistema operacional
(SILVA; SILVA; RUPPERT, 2012)	Não	Não	<i>Android</i>
(GUTERRES <sup>1</sup> ; LIMA; SILVA, 2017)	Sim	Não	<i>Android</i>
(CUNHA, 2012)	Sim	Sim ( <i>Web</i> )	Desenvolvido para <i>Web</i>
(ANDRADE, 2013)	(Testes de usabilidade não realizado)	Não	<i>Android</i>
(SANTOS et al., 2017)	Sim	Não	<i>Android</i>
E.M Escala Médica	Sim	Sim (híbrido)	<i>Android</i> e <i>Web</i>

**Facilidade de uso:** Dos trabalhos citados na tabela 1, os de GUTERRES<sup>1</sup>, LIMA e SILVA (2017), CUNHA (2012) e SANTOS et al. (2017), foram definidos como aplicações que possuem grande facilidade de uso, segundo descrição, após serem realizados testes com usuários.

Já o trabalho de SILVA, SILVA e RUPPERT (2012) foi desenvolvido apenas um protótipo, e os testes apresentaram pouca satisfação dos usuários com a interface do aplicativo

e o de [ANDRADE \(2013\)](#) não realizou teste até o momento de escrita do trabalho relacionado ao aplicativo.

**Multiplataforma:** Os 5 trabalhos que são correlacionados com o projeto desenvolvido neste trabalho, de acordo com descrição, foram implementados para plataformas específicas (*Android* ou *Web*).

O estudo específico de [CUNHA \(2012\)](#) realizou uma análise considerando que o software desenvolvido precisaria ser executado em apenas uma plataforma, sendo essa a mais adequada para ser executada em computadores pessoais, pois o sistema seria utilizado internamente dentro de uma clínica, apenas em notebooks e computadores de mesa.

**Sistema operacional:** Os trabalhos de [SILVA, SILVA e RUPPERT \(2012\)](#) e [GUTERRES<sup>1</sup>, LIMA e SILVA \(2017\)](#), [ANDRADE \(2013\)](#) e [SANTOS et al. \(2017\)](#), foram desenvolvidos para plataforma *Android*, não sendo essas aplicações disponibilizadas para mais nenhuma plataforma.

Já o trabalho de [CUNHA \(2012\)](#), foi desenvolvido para ser executado em qualquer sistema operacional que permita a instalação de um navegador *web*.

Os trabalhos relacionados procuram proporcionar serviços semelhantes (na mesma área), o de [SILVA, SILVA e RUPPERT \(2012\)](#) disponibiliza meios de apresentação de imagens tridimensionais, resultadas de consultas médicas.

O de [GUTERRES<sup>1</sup>, LIMA e SILVA \(2017\)](#) desenvolveu um protótipo para auxiliar no uso de medicamentos, já o de [CUNHA \(2012\)](#) foi criado para gestão de clínicas médicas.

O trabalho de [SANTOS et al. \(2017\)](#) foi proposto para disponibilizar informações para avaliação de fragilidade de idosos, enquanto o de [ANDRADE \(2013\)](#), foi feito para auxiliar na descoberta de patologias como problemas no sono, com a disponibilização de informações para os pacientes.

A aplicação E.M Escala Médica possui páginas totalmente responsivas, foi desenvolvida para sistemas *Web* (o sistema administrativo) e para sistema *Android* (aplicativo *mobile*) e tem a função de apresentar as escalas médicas para usuários dos serviços de saúde disponíveis na clínica.

## 3 Proposta de Desenvolvimento das Aplicações de Escalas Médicas

### 3.1 Etapa Desenvolvimento

Para o desenvolvimento das aplicações do sistema *web* administrativo e aplicativo *mobile* para escalas médicas da Clínica Santa Ana será usado a *Unified Modelling Language - UML* para modelagem do sistema, Banco de Dados MongoDB para armazenar os dados do sistema, criar uma *Application Programming Interface (API)* através do padrão RESTful, usar o *Framework Express.js* para recursos de roteamento para os métodos da API e utilização dos métodos HTTP, o *Node.js* para serviço de conexão, inserção, atualização e buscas em banco de dados, o *Angular* para criar o *Front-And* administrativo da aplicação e conexão com a API, e usando o *Ionic* para criação do aplicativo *mobile* e fazer a conexão entre o aplicativo e API.

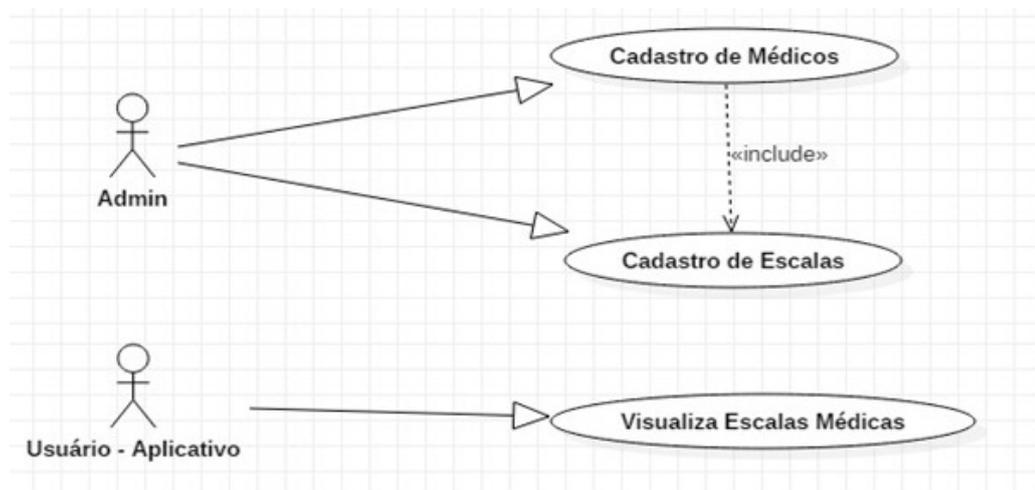
### 3.2 Modelagem

A linguagem padrão para modelagem *Unified Modelling Language - UML* (em português, Linguagem de Modelagem Unificada) útil no desenvolvimento de aplicações permite representar a estrutura, principais funcionalidades e comunicação entre objetos do sistema.

Realizada a elicitação de requisitos para o desenvolvimento das aplicações temos os seguintes diagramas:

#### 3.2.1 Caso de Uso

Figura 1: Diagrama de caso de uso.



A Figura 1 mostra os requisitos funcionais do sistema, funcionalidades do sistema.

### Documentação Caso de Uso

Caso de Uso 01 – Cadastro de médicos.

Descrição: Este caso de uso serve para cadastrar, atualizar, listar e excluir médicos.

Ator: Administrador.

Pré-condição: O administrador precisa está logado para realizar as ações.

Pós-condição: as informações precisam serem enviadas para o aplicativo mobile do usuário.

Caso de Uso 02 – Cadastro de Escalas.

Descrição: Este caso de uso serve para cadastrar, atualizar, listar e excluir escalas de médicos.

Ator: Administrador.

Pré-condição: O administrador precisa está logado para realizar as ações.

Pós-condição: As informações precisam serem enviadas para o aplicativo mobile do usuário.

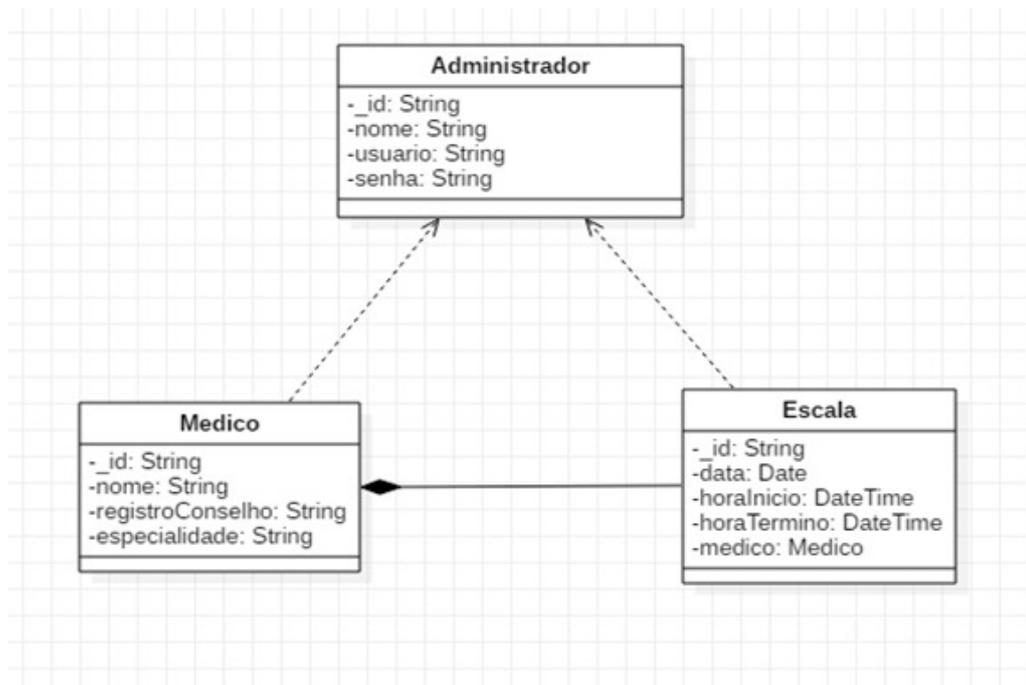
Caso de Uso 03 – Visualiza Escalas Médicas.

Descrição: Este caso de uso serve para apresentar as informações para o usuário do aplicativo.

Ator: Usuário do aplicativo.

### 3.2.2 Diagrama de Classes

Figura 2: Diagrama de classes.



Na Figura 2 é possível observar a estrutura conceitual do sistema, onde são mostradas as classes, atributos, operações e as relações entre os objetos do mesmo.

Como o banco de dados é compartilhado entre as aplicações que consomem a API (aplicativo *mobile* e sistema *web* administrativo), o diagrama de classes apresentado já dispõe de toda a modelagem da plataforma.

A aplicação possui três (3) classes principais, são elas: *Administrador* (responsável por todos os cadastros e atualizações do sistema), *Medico* (possui suas informações profissionais disponibilizadas para os usuários) e *Escala* (representa a escala dos médicos).

As classes *Medico* e *Escala* possuem uma composição entre si, já que a *Escala* depende do seu Espelho (classe *Medico*) para existir.

## 4 Resultados e Discussões

Este capítulo apresenta o desenvolvimento de uma aplicação administrativa para atualização de informações sobre profissionais, suas especialidades e suas escalas de trabalho na Clínica de Reabilitação Santa Ana, que possui sua sede na cidade de Picos – PI e atende em média 300 pessoas por dia. Levando em consideração a apresentação do desenvolvimento e as tecnologias e recursos utilizados.

A aplicação E.M Escala Médica desenvolvida para a clínica se destaca pelo uso de tecnologias novas e escaláveis, principalmente na criação da aplicação *web*. Além de trabalhar de forma integrada, as aplicações permitem aos usuários atualizar e visualizar informações *online*, diferentemente da maioria das plataformas estudadas.

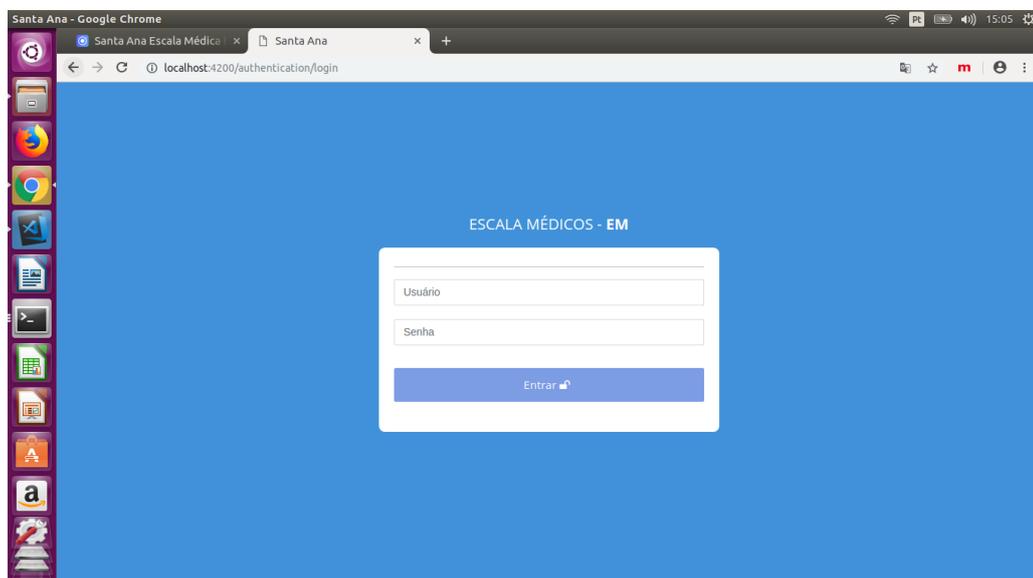
### 4.1 Sistema *Web*

O sistema administrativo foi criado com a funcionalidade de autenticação, cadastro e atualização de informações sobre os médicos e suas escalas de trabalho. As informações cadastradas neste sistema são armazenadas no banco de dados da API e compartilhadas com o aplicativo.

#### 4.1.1 Área administrativa do Aplicativo

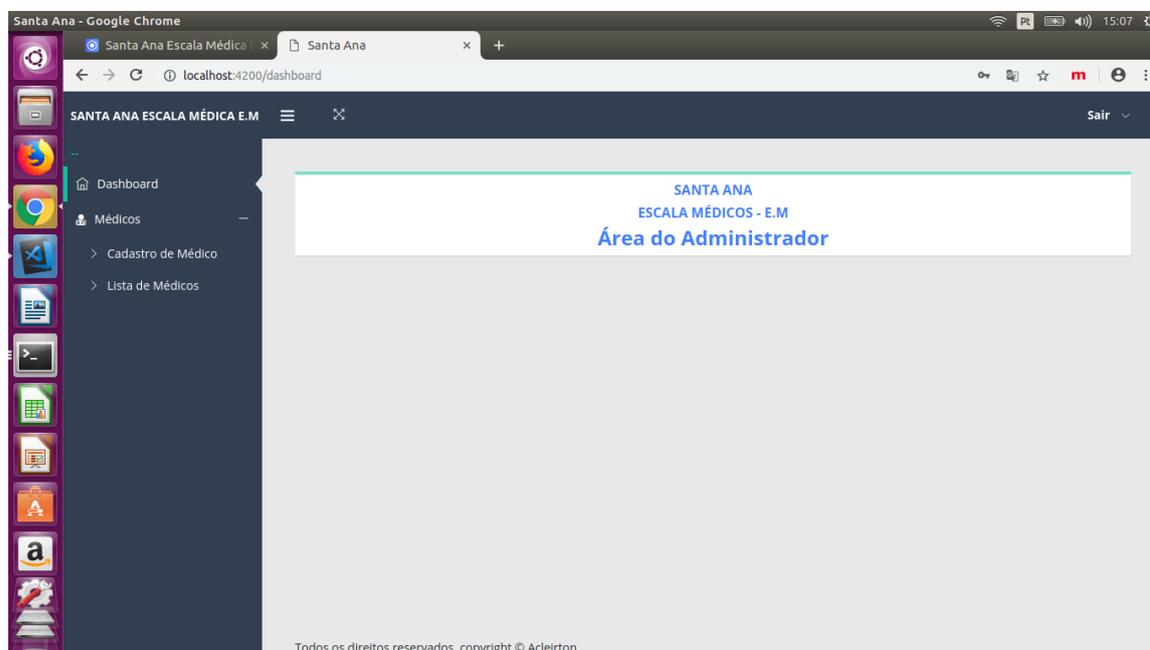
Aqui está a área onde o administrador do sistema faz todas as operações necessárias de atualização do sistema. A seguir imagens da área administrativa do aplicativo:

Figura 3: Tela de login do administrador.



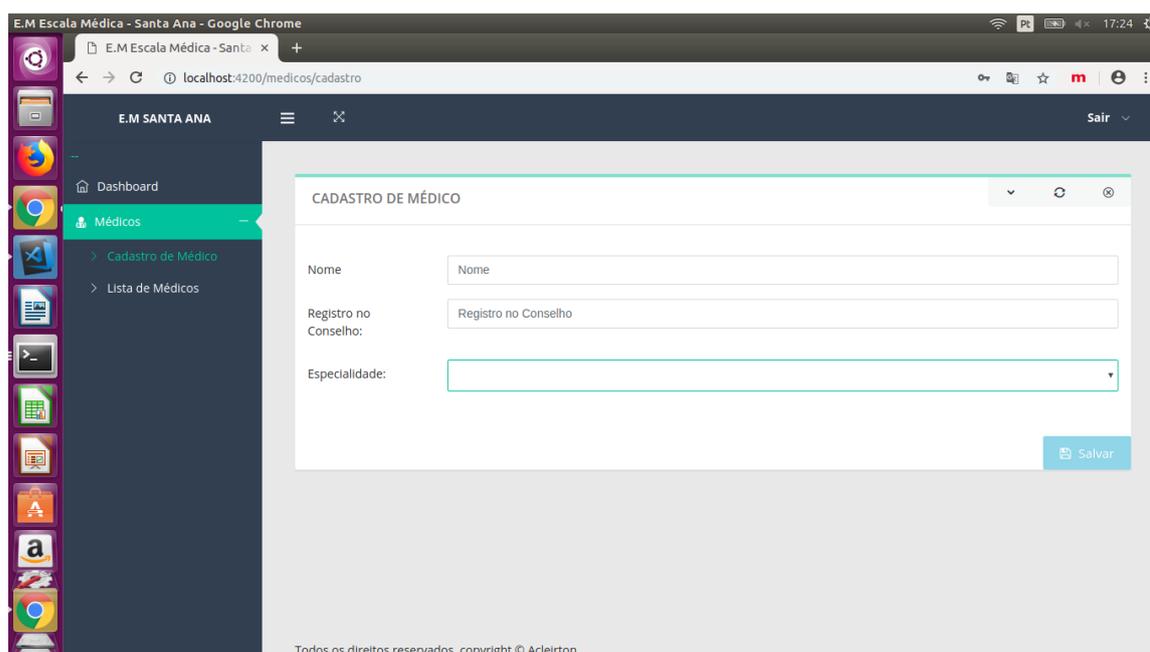
A Figura 3 mostra a área do administrador, onde o mesmo faz sua autenticação no sistema através de usuário e senha para poder ter acesso ao sistema.

Figura 4: Área do administrador - logado.



A Figura 4 mostra a área o administrador do sistema onde são disponibilizadas as opções (Menu) *Cadastro de Médico* para cadastros dos profissionais e *Lista de Médicos*, onde possível visualizar informações e fazer alterações.

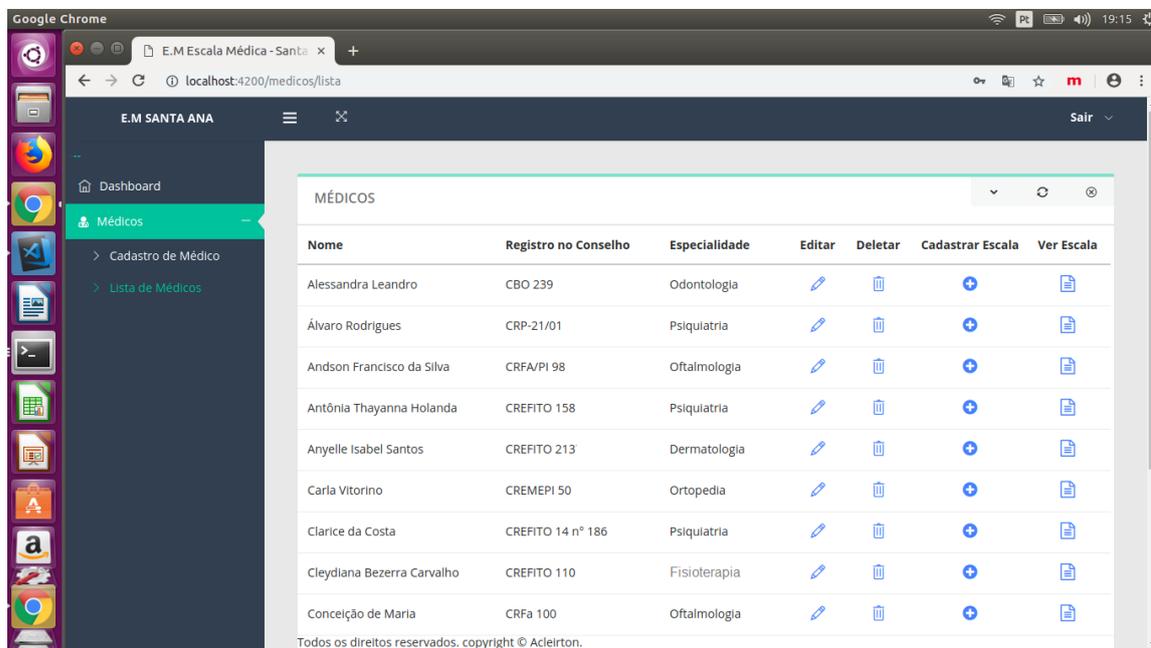
Figura 5: Administrador adiciona médicos.



A Figura 5 mostra a área onde o administrador do sistema adiciona os profissionais no sistema (Os dados cadastrados são: Nome do profissional, seu registro no conselho compe-

tente e sua especialidade). Quando o administrador insere dados válidos no formulário e clica em *Salvar*, os dados são enviados para a API, validados e salvos no Banco de Dados.

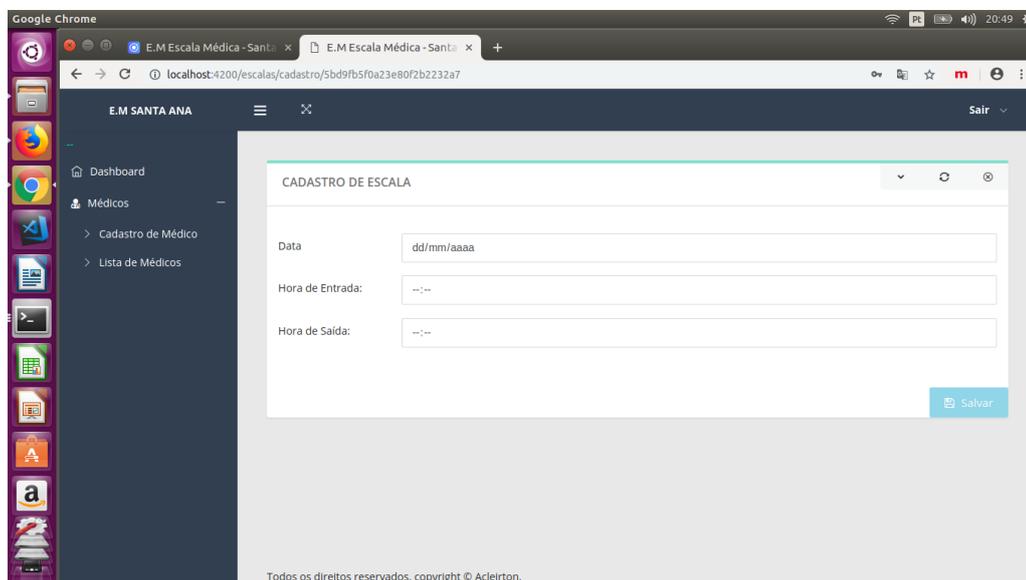
Figura 6: Profissionais cadastrados no sistema.



A Figura 6 mostra a área onde o administrador do sistema ver os dados de todos os profissionais cadastrados e tem as opções de *Editar* para fazer alterações no cadastro do profissional, *Deletar* para excluir um profissional cadastrado, *Cadastrar Escala* para inserir a escala dos profissionais e *Ver Escala* para visualização das informações sobre as escalas dos profissionais.

As informações listadas são buscadas no banco de dados, e enviados para a página por meio de requisições feitas pela aplicação *Front-end* à aplicação *Back-end*.

Figura 7: Administrador adiciona escalas.



A Figura 7 mostra a área onde o administrador do sistema adiciona a escala de cada profissional inserindo data, hora de entrada e hora de saída e clica em *Salvar* para salvar no banco de dados.

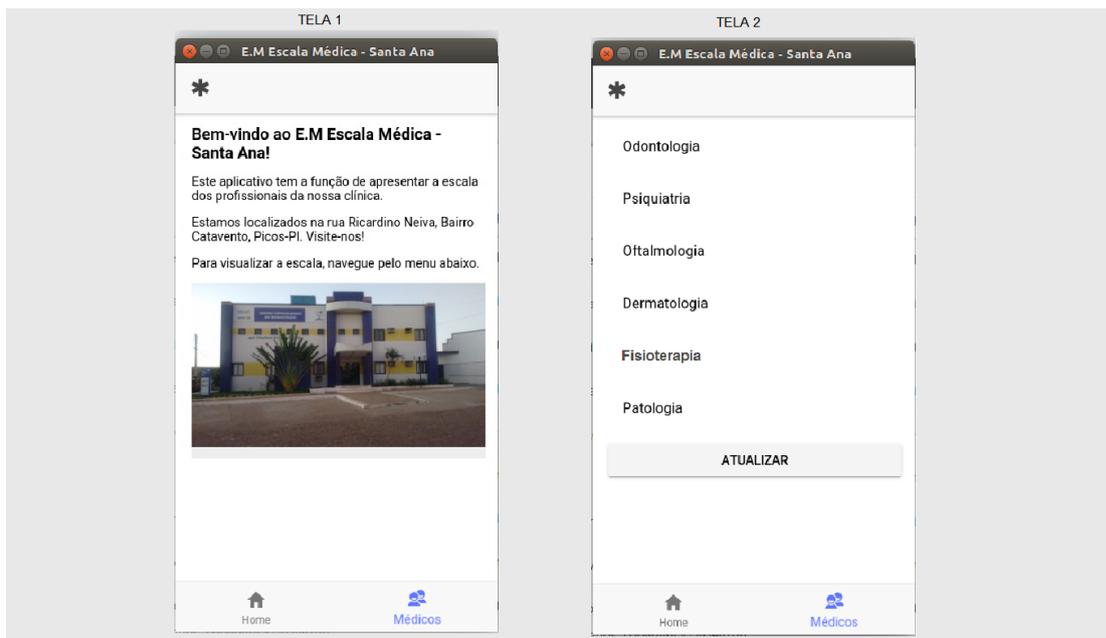
## 4.2 Sistema *Mobile*

O aplicativo E.M Escala Médica tem a função de realizar solicitações a API e apresentar as informações enviadas por esta. Sendo essas informações, os dados dos profissionais da clínica, suas especialidades e suas escalas.

### 4.2.1 Interface do aplicativo *mobile*

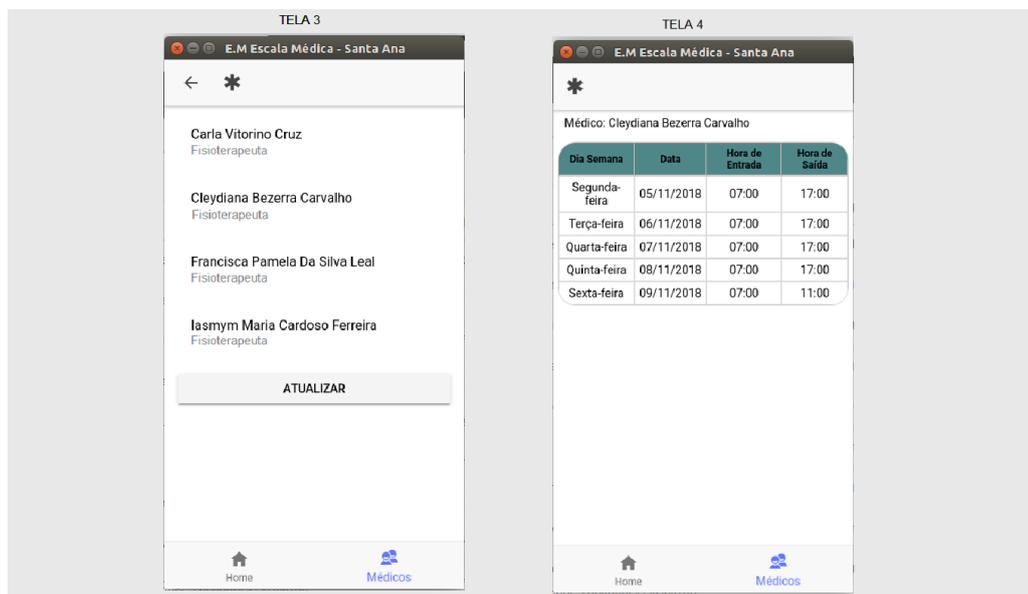
O Aplicativo *mobile* desenvolvido possui as seguintes telas: *Inicial*, *Especialidades*, *Profissionais* e *Escala*. Apresentadas a seguir:

Figura 8: Tela inicial do Aplicativo e Tela Especialidades.



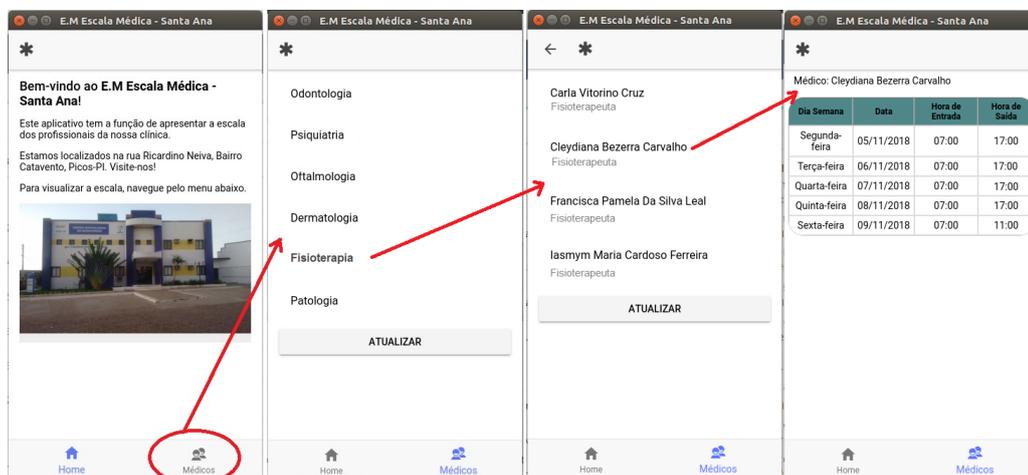
A Figura 8 mostra duas telas do aplicativo *mobile*, onde a Tela 1 os usuários tem o primeiro contato com o Aplicativo e pode navegar através dos *menus*. Clicando no *menu Médicos* vai para a Tela 2 com as especialidades médicas disponíveis na clínica. A Tela 2 apresenta as especialidades médicas, nesta tela usuário pode clicar em uma especialidade e ver os médicos que atendem a especialidade selecionada.

Figura 9: Profissionais por especialidade e Escala.



A Figura 9 apresenta as outras duas telas do aplicativo *mobile*. A Tela 3 mostra os profissionais da especialidade selecionada pelo usuário na tela anterior, nessa tela, clicando no nome de um profissional, será direcionado para a tela com a escala. A tela 4 apresenta a escala do profissional selecionado, com os detalhes sobre datas e horários de atendimento.

Figura 10: Interface completa do aplicativo.



A Figura 10 apresenta toda interface do aplicativo. Onde mostra o breve caminho percorrido pelo usuário para chegar a escala do profissional desejado. Abre o aplicativo, em seguida clica no menu MÉDICOS, vai aparecer as especialidades médicas disponíveis, o usuário escolhe clicando na especialidade, desejada em seguida é apresentado os médicos dessa especialidade, então o usuário escolhe o profissional que deseja saber a escala e clica no seu nome, e por fim terá a escala desse profissional.

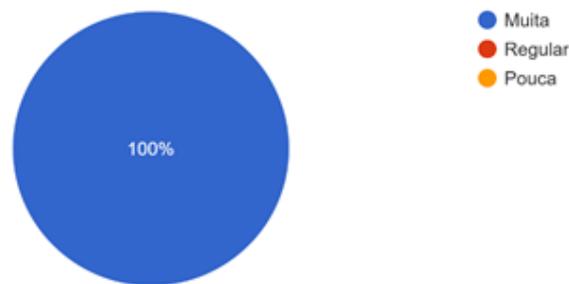
## 5 Testes

A metodologia utilizada para o teste foi a aplicação de questionário, que foi realizada de forma individual, após cada usuário utilizar o aplicativo *mobile*.

Para a elaboração do questionário de teste (Apêndice A), foi utilizado o princípio de pesquisa avaliativa, ou seja, as perguntas pedem uma avaliação do usuário sobre as funcionalidades do aplicativo.

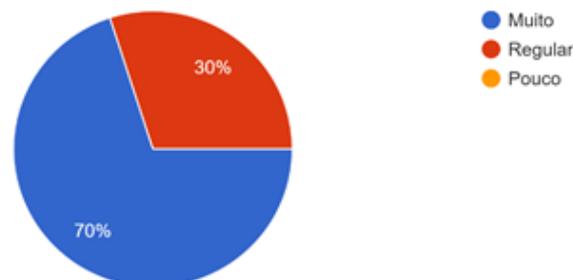
Em testes realizados com 10 pessoas, entre funcionários e pacientes da Clínica de Reabilitação Santa Ana, essas responderam perguntas específicas sobre a experiência de uso do aplicativo. Abaixo serão apresentadas as informações obtidas, e gráficos.

### 1 - Qual o nível de facilidade encontrada na utilização do aplicativo?



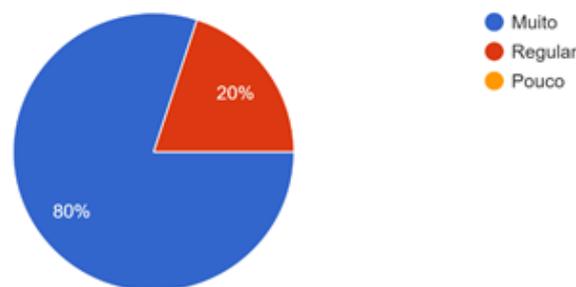
A Questão 1 sobre a facilidade encontrada na utilização no aplicativo teve cem por cento (100%) das respostas dos usuários como muita facilidade encontrada para usar o aplicativo.

### 2 - Para se chegar na informação desejada (escala do médico), a quantidade de cliques necessários é satisfatório?



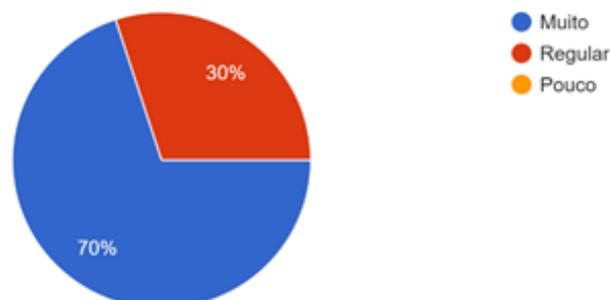
A Questão 2 sobre a quantidade de cliques para se chegar a informação desejada mostra que setenta por cento (70%) dos usuários responderam que a quantidade é muito satisfatória. Já trinta por cento (30%) responderam que acharam regular, indicando que poderia ser melhorada a interface, para que a informação fosse disponibilizada com mais facilidade.

**3 - Na sua opinião, o aplicativo Escala Médica será importante para a população que precisar saber informações sobre a escala médica da Clínica de Reabilitação Santa Ana?**



A Questão 3 sobre a opinião dos usuários, sobre a importância do aplicativo para os pacientes da clínica teve oitenta por cento (80%) das respostas afirmando que será muito importante, destacando as dificuldades enfrentadas atualmente em clínicas com relação a disponibilização de informações. Vinte por cento (20%) acham que terá uma importância regular, observando que existe outros meios para se conseguir as informações disponibilizadas no aplicativo.

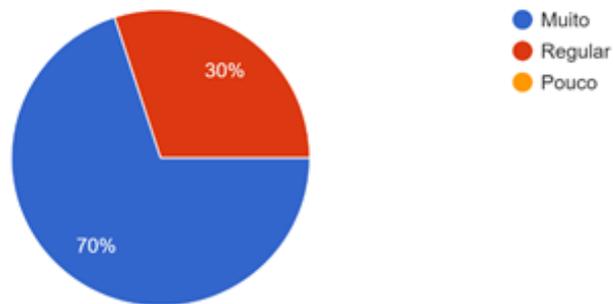
**4 - Qual o nível de facilidade encontrada para navegar até a tela com as informações de escala dos médicos?**



A Questão 4 sobre o nível de facilidade encontrada para navegar até a tela com as informações de escala dos médicos, setenta por cento dos usuários (70%) responderam

que acharam muito fácil a navegação pelo aplicativo *mobile*, já trinta por cento (30%) indicaram melhorias, como a retirada dos botões de atualizar informações, fazendo com que as informações sejam atualizadas automaticamente.

**5 - O tempo em que o aplicativo leva para apresentar as informações buscadas em cada tela, em sua opinião, é satisfatório?**



A Questão 5 sobre o tempo de resposta do aplicativo, levando em consideração a busca e apresentação das informações, considerando fatores como velocidade da banda de *Internet* utilizada no momento do teste com cada usuário, setenta por cento (70%) responderam que as informações são disponibilizadas de forma muito rápida, já trinta por cento (30%), responderam que a velocidade poderia melhorar. Questão de performance pode ser melhorada.

Mediante a fase de avaliação dos testes, e com base em todos os procedimentos metodológicos cabíveis e abordados no presente trabalho, a aplicação denominada E.M Escala Médicos, ficou validada e apta para uso, atendendo aos requisitos propostos na temática central do presente trabalho.

## 6 Conclusões e Trabalhos Futuros

O presente trabalho, que teve como objetivo o desenvolvimento de uma aplicação, para atualização e disponibilização de informações sobre os profissionais e suas escalas de trabalho na Clínica de Reabilitação Santa Ana, obteve êxito (como é apresentado no capítulo de desenvolvimento deste trabalho), como foi definido inicialmente.

Atualmente, na Clínica de Reabilitação Santa Ana, o processo de informação dos pacientes ocorre por meio de telefonemas ou através de contato pessoal com os pacientes. Assim, após debater com os gestores da clínica, foram definidas as necessidades prioritárias a serem resolvidas neste trabalho. A principal necessidade vista inicialmente foi a disponibilização das escalas dos médicos. Para atender a essa necessidade, o projeto apresentado no decorrer deste trabalho, foi desenvolvido o software e testado com usuários do aplicativo de Escala Médica – E.M.

Com isso, houve algumas dificuldades mais evidentes no decorrer do desenvolvimento, como também é previsto para o futuro processo de implantação da ferramenta. Algumas delas foram quanto à adaptação dos gestores da clínica em sempre estarem atualizando as informações no sistema administrativo e a dificuldade, quanto ao processo de desenvolvimento, dos mesmos em informar quais as reais necessidades da clínica e de seus pacientes.

O sistema e aplicativos, resultados deste trabalho, foram desenvolvidos seguindo regras de funcionamento pré-estabelecidos pelos gestores da Clínica de Reabilitação Santa Ana.

Além disso, as tecnologias utilizadas permitiram a utilização de padrões de desenvolvimento utilizados em grandes projetos conhecidos mundialmente, um exemplo de padrão utilizado, foi o padrão RESTful, que possibilita a implementação dos princípios REST.

Para a comunidade acadêmica, o presente trabalho disponibiliza detalhes sobre tecnologias emergentes, pesquisadas e utilizadas no mesmo. No capítulo de referencial teórico, foram apresentadas ferramentas que permitem a criação de aplicações de pequena, média e grande escala, que podem ser utilizadas em vários meios acadêmicos e da sociedade em geral.

Por fim, baseado no capítulo de desenvolvimento deste trabalho e nos testes realizados, pode-se concluir que a aplicação E.M Escala Médica conseguiu atingir os objetivos esperados. As aplicações foram desenvolvidas passo a passo com ciclos de apresentação ao cliente, resultando no final de seu desenvolvimento em um produto definido e aceito pelo mesmo, através do questionário aplicados a funcionários da clínica.

Para trabalhos futuros fica a sugestão para melhorar a interface do aplicativo, retirada dos botões de atualizar informações e melhorar a velocidade para o aplicativo apresentar as informações em cada tela. Outra sugestão seria o desenvolvimento do aplicativo *mobile* também para o sistema operacional *iOS* da *Apple*.

# Referências

- ABREU, L. Typescript – o javascript moderno para criação de aplicações. Lisboa: FCA – Editora de Informática, Ltda, 2017. Citado na página 20.
- ANDRADE, C. R. d. *Proposta e desenvolvimento de aplicativo móvel de representação de dados de EEG e PDC*. Tese (Doutorado) — Universidade de São Paulo, 2013. Citado 2 vezes nas páginas 24 e 25.
- ANDROIDDEVELOPERS Documentation for app developers. 2018. Disponível em: <<https://developer.android.com/>>. Acesso em: 15.10.2018. Citado na página 18.
- ARAÚJO, M.; SPÍNOLA, E. Editorial qualidade de software. Engenharia de software magazine, ano 1, ed. 1, 2007. Citado na página 17.
- CUNHA, F. A. V. T. d. Sistema para gerenciamento de consultas médicas. Blumenau, 2012. Citado 2 vezes nas páginas 24 e 25.
- CUNHA, P. R. F. Engenharia de software. Computação Brasil, Revista da Sociedade Brasileira de Computação, SBC, 2015. Disponível em: <[http://www.sbc.org.br/images/flippingbook/computacaobrasil/computa\\_28/cb2015-03.pdf](http://www.sbc.org.br/images/flippingbook/computacaobrasil/computa_28/cb2015-03.pdf)>. Acesso em: 25.11.2018. Citado na página 17.
- FREEMAN, A. Pro angularjs. Nova Iorque: Apress Media LLC., 2014. Citado na página 21.
- GONÇALVES, A. J. R. Desenvolvimento de aplicativos híbridos com o ionic framework. 2017. Citado na página 22.
- GUTERRES<sup>1</sup>, L. d. C.; LIMA, A. R. S.; SILVA, W. A. Utilização de aplicativo para dispositivo móvel na administração de medicamentos. 2017. Citado 2 vezes nas páginas 24 e 25.
- KOSCIANSKI, A.; SOARES, M. dos S. Qualidade de software: aprenda as metodologias e técnicas mais modernas para o desenvolvimento de software. São Paulo: Novatec Editora, 2007. Citado na página 18.
- MOORE, A. *Mobile as 7th of the mass media: an evolving history*. 2007. Citado na página 13.
- PEREIRA, C. R. Node.js: Aplicações web realtime com node.js. São Paulo: Casa do Código, 2013. Citado na página 20.
- PRESSMAN, R. S. Software engineering. McGRaw-Hill, 2002. Citado na página 19.
- RICHARDSON, L.; RUBY, S. Restful web services. gravenstein highway norte. Sebastopol. O'Reilly Media, 2007. Citado na página 19.
- SANTOS, T. Silva dos et al. Desenvolvimento de aplicativo para dispositivos móveis voltado para identificação do fenótipo de fragilidade em idosos. *Revista Brasileira de Geriatria e Gerontologia*, Universidade do Estado do Rio de Janeiro, v. 20, n. 1, 2017. Citado 2 vezes nas páginas 24 e 25.

SILVA, G. H. da; SILVA, J. V. da; RUPPERT, G. C. Desenvolvimento de aplicativos para visualização de imagens médicas em dispositivos móveis. In: *XII Workshop de Informática Médica (WIM), Curitiba. Anais do XXXII Congresso da Sociedade Brasileira de Computação (CSBC)*. [S.l.: s.n.], 2012. Citado 2 vezes nas páginas 24 e 25.

SOMMERVILLE, I. Testes de software. Tradução: Maurício de Andrade. 6. ed São Paulo: Addison Wesley, 2003. Citado na página 18.

TAROUCO, L. M. R. Um panorama da fluência digital na sociedade da informação. In: BEHAR, Patrícia A. (Org.). *Competências em Educação a Distância*. Porto Alegre: Penso, 2013. Citado na página 22.

TEIXEIRA, P. Hands-on node.js. p. 1, 2013. Citado na página 20.

WEBBER, C. et al. Ferramenta especialista para avaliação de software educacional. 2009. Disponível em: <<http://www.brie.org/pub/index.php/sbie/article/viewFile/1115-1018>>. Acesso em: 25.11.2018. Citado na página 18.

# Apêndices

## APÊNDICE A – Questionário

1 - Qual o nível de facilidade encontrada na utilização do aplicativo?

- a) Muita
- b) Regular
- c) Pouca

2 - Para se chegar na informação desejada (escala do médico), a quantidade de cliques necessários é satisfatório?

- a) Muito
- b) Regular
- c) Pouco

3 - Na sua opinião, o aplicativo Escala Médica será importante para a população que precisar saber informações sobre a escala médica da Clínica de Reabilitação Santa Ana?

- a) Muito
- b) Regular
- c) Pouco

4 - Qual o nível de facilidade encontrada para navegar até a tela com as informações de escala dos médicos?

- a) Muito
- b) Regular
- c) Pouco

5 - O tempo em que o aplicativo leva para apresentar as informações buscadas em cada tela, em sua opinião, é satisfatório?

- a) Muito
- b) Regular
- c) Pouco

# APÊNDICE B – Código de atualização de profissionais

```
1 import Mensagem from '../messages/mensagens';
2 import MEDICALJSONError from '../validators/medical_json_error';
3
4 //Classe médico
5 class Medico{
6   //construtor da classe, passando como parâmetro os modelos de Médico e Escala
7   constructor(Medico, Escala) {
8     this.medico = Medico;
9     this.escala = Escala;
10  }
11
12   //método que busca no banco de dados (mongoDB) e retorna todos os médicos
13   async listaMedicos(req, res) {
14     try {
15       const medicos = await this.medico.find({}); // busca no banco usando o método find, o m
16       res.send(medicos);
17     } catch (e) {
18       console.log(e.message);
19       res.status.send(Mensagem.ERRO_NA_REQUISICAO);
20     }
21   }
22
23   //método que busca pelo _id um médico no banco e retorna esse médico
24   async buscaMedico(req, res) {
25     try {
26       const medico = await this.medico.findOne({_id: req.params.id}); // busca no banco usan
27       res.send(medico);
28     } catch (e) {
29       console.log(e.message);
30       res.status.send(Mensagem.ERRO_NA_REQUISICAO);
31     }
32   }
33
34   //método que salva um médico
35   async salvaMedico(req, res) {
36     try {
37       const verificaCRM = await this.medico.findOne({crm: req.body.crm}, 'crm'); //verifica
38       if(verificaCRM == null){
39         const medico = new this.medico(req.body); //cria uma instancia do model de médico,
40         await medico.save(); //salva os dados usando a instancia criada e chamando o métod
41         res.status(201).send(Mensagem.SUCESSO_NA_REQUISICAO); //retorna a mensagem de suce
42       }else{
43         throw new MEDICALJSONError(Mensagem.CRM_JA_CADASTRADO); //retorna a mensagem de err
44       }
45     }
46   }
47 }
```

```
JS medicos.js x
45     } catch (e) { //aqui dentro retorna as mensagens quando dar erro do processo de salvar o
46       console.log(e.message);
47       if(e instanceof MEDICALJSONError){
48         res.status(400).send(e.msgJSON);
49       }else{
50         res.status.send(Mensagem.ERRO_NA_REQUISICAO);
51       }
52     }
53   }
54 }
55 //método que salva o médico
56 async atualizaMedico(req, res) {
57   try {
58     const verificaCRM = await this.medico.findOne({crm: req.body.crm}, '_id crm');
59     if(verificaCRM == null || verificaCRM._id == req.params.id){
60       const medico = await this.medico.update({_id: req.params.id}, req.body); //chama c
61       res.status(200).send(Mensagem.SUCESSO_NA_REQUISICAO);
62     }else{
63       throw new MEDICALJSONError(Mensagem.CRM_JA_CADASTRADO);
64     }
65   } catch (e) {
66     console.log(e.message);
67     if(e instanceof MEDICALJSONError){
68       res.status(400).send(e.msgJSON);
69     }else{
70       res.status.send(Mensagem.ERRO_NA_REQUISICAO);
71     }
72   }
73 }
74 //método para remover um médico e suas escalas
75 async removeMedico(req, res) {
76   try {
77     const medico = await this.medico.remove({_id: req.params.id}); //remove um médico
78     const escalas = await this.escala.find({medico_id: req.params.id}, '_id');
79     for(let i = 0; i < escalas.length; i++){
80       await this.escala.remove({_id: escalas[i]._id}); //remove todas as suas escalas
81     }
82     res.status(200).send(Mensagem.SUCESSO_NA_REQUISICAO);
83   } catch (e) {
84     console.log(e.message);
85     res.status.send(Mensagem.ERRO_NA_REQUISICAO);
86   }
87 }
88 }
89
90 export default Medico;
91
```

# APÊNDICE C – Código Escala dos profissionais

```

1 import Mensagem from '../messages/mensagens';
2 import MEDICALJSONError from '../validators/medical_json_error';
3 import moment from 'moment';
4
5 class Escala {
6   constructor(Escala) {
7     this.escala = Escala;
8   }
9
10  encontraDia(data) {
11    let dia;
12    if(data.getDay() == 0){
13      dia = "Domingo";
14    }else if(data.getDay() == 1){
15      dia = "Segunda-feira";
16    }else if(data.getDay() == 2){
17      dia = "Terça-feira";
18    }else if(data.getDay() == 3){
19      dia = "Quarta-feira";
20    }else if(data.getDay() == 4){
21      dia = "Quinta-feira";
22    }else if(data.getDay() == 5){
23      dia = "Sexta-feira";
24    }else if(data.getDay() == 6){
25      dia = "Sábado";
26    }
27    return dia;
28  }
29
30  async buscaEscalaMedico(req, res) {
31    try {
32      let dia, e = [];
33      const escala = await this.escala.find({medico_id: req.params.medico_id, data: {$gte: new Date()});
34      if(escala.length > 0){
35        const tam = escala.length;
36        for(let i = 0; i < escala.length; i++){
37          const d = new Date(escala[i].data);
38          dia = this.encontraDia(d);
39          e.push({_id: escala[i]._id, dia: dia, data: this.dataFormatada(escala[i].data), hora: escala[i].hora});
40          if(i == tam - 1){
41            res.send(e);
42          }
43        }
44      }else {
45        res.send(escala);
46      }
47    } catch (e) {
48      console.log(e.message);
49      res.status(400).send(Mensagem.ERRO_NA_REQUISICAO);
50    }
51  }
52
53  async buscaEscala(req, res) {
54    try {
55      const escala = await this.escala.findOne({_id: req.params.id});
56      const e = { _id: escala._id, data: this.dataFormatada(escala.data), hora: escala.hora };
57      res.send(e);
58    } catch (e) {
59      console.log(e.message);
60      res.status(400).send(Mensagem.ERRO_NA_REQUISICAO);
61    }
62  }
63 }

```

```
63
64 async salvaEscala(req, res) {
65   try {
66     let d = new Date(req.body.data);
67     d.setDate(d.getDate() + 1);
68     req.body.data = d;
69     const escala = await this.escala.findOne({medico_id: req.body.medico_id, data: {$eq: req.body.d
70     if(escala == null){
71       const escalaSalva = new this.escala(req.body);
72       await escalaSalva.save();
73       res.status(201).send(Mensagem.SUCESSO_NA_REQUISICAO);
74     }else{
75       throw new MEDICALJSONError(Mensagem.HORARIO_RESERVADO);
76     }
77   } catch (e) {
78     console.log(e.message);
79     if(e instanceof MEDICALJSONError){
80       res.status(400).send(e.msgJSON);
81     }else{
82       res.status.send(Mensagem.ERRO_NA_REQUISICAO);
83     }
84   }
85 }
86 async atualizaEscala(req, res) {
87   try {
88     let d = new Date(req.body.data);
89     d.setDate(d.getDate() + 1);
90     req.body.data = d;
91     const escala = await this.escala.findOne({medico_id: req.body.medico_id, data: {$eq: req.body.d
92     if(escala == null || escala._id == req.params.id){
93       await this.escala.update({_id: req.params.id}, req.body);
94       res.status(201).send(Mensagem.SUCESSO_NA_REQUISICAO);
95     }else{
96       throw new MEDICALJSONError(Mensagem.HORARIO_RESERVADO);
97     }
98   } catch (e) {
99     console.log(e.message);
100    if(e instanceof MEDICALJSONError){
101      res.status(400).send(e.msgJSON);
102    }else{
103      res.status.send(Mensagem.ERRO_NA_REQUISICAO);
104    }
105  }
106 }
```

```
106 }
107
108 async removeEscala(req, res) {
109   try {
110     await this.escala.remove({_id: req.params.id});
111     res.send(Mensagem.SUCESSO_NA_REQUISICAO);
112   } catch (e) {
113     console.log(e.message);
114     res.status(400).send(Mensagem.ERRO_NA_REQUISICAO);
115   }
116 }
117
118 //métodos para formatar data
119 dataFormatada(data) {
120   //data = new Date(moment(data).utc().add(1, 'day'));
121   let dataCompleta = this.acrescentaZeroDataHora(data.getDate()) + "/" + this.acrescentaZeroDataHor
122   return dataCompleta;
123 }
124
125 acrescetaZeroDataHora(data_hora) {
126   if (parseInt(data_hora) < 10) {
127     data_hora = "0" + data_hora;
128     return data_hora;
129   }
130   return data_hora;
131 }
132
133 data(data) {
134   let d = data.split("/");
135   data = d[2] + '-' + d[1] + '-' + d[0];
136   return data;
137 }
138 }
139
140 export default Escala;
```



**TERMO DE AUTORIZAÇÃO PARA PUBLICAÇÃO DIGITAL NA BIBLIOTECA  
“JOSÉ ALBANO DE MACEDO”**

**Identificação do Tipo de Documento**

- ( ) Tese  
( ) Dissertação  
( X ) Monografia  
( ) Artigo

Eu, ACLEIRTON DE SÁ E SOUSA, autorizo com base na Lei Federal nº 9.610 de 19 de Fevereiro de 1998 e na Lei nº 10.973 de 02 de dezembro de 2004, a biblioteca da Universidade Federal do Piauí a divulgar, gratuitamente, sem ressarcimento de direitos autorais, o texto integral da publicação Desenvolvimento de uma aplicação de Escalas Médicas na Clínica de Reabilitação Santa Ana em Picos – PI, de minha autoria, em formato PDF, para fins de leitura e/ou impressão, pela internet a título de divulgação da produção científica gerada pela Universidade.

Picos-PI 19 de dezembro de 2018.

Acleirton de Sá e Sousa  
Assinatura

Acleirton de Sá e Sousa  
Assinatura