

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE INFORMÁTICA
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE
SISTEMAS

RAMON WOLSKI KIERAS

SISTEMA PARA AGENDAMENTO DE SERVIÇOS

TRABALHO DE CONCLUSÃO DE CURSO

PONTA GROSSA

2019

RAMON WOLSKI KIERAS

SISTEMA PARA AGENDAMENTO DE SERVIÇOS

Trabalho de Conclusão de Curso apresentado como requisito parcial à obtenção do título de Tecnólogo em Análise e Desenvolvimento de sistemas, do Departamento Acadêmico de Informática / Coordenação do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, da Universidade Tecnológica Federal do Paraná.

Orientador: Prof. Dr. Richard Duarte Ribeiro

PONTA GROSSA

2019

AGRADECIMENTOS

Primeiramente a Deus pela oportunidade e pelas bençãos que recebi durante toda essa caminhada.

Agradeço a minha família por sempre me apoiar e colaborar para que mais uma trajetória da minha vida fosse concluída.

Agradeço também ao meu orientador, professor Dr Richard Duarte Ribeiro por todo o apoio e dedicação prestada para a conclusão desse projeto.

Por fim, expresso a minha gratidão a todos os professores da instituição, dos quais se empenharam para nos preparar para os desafios do mercado de trabalho.



Ministério da Educação
Universidade Tecnológica Federal do Paraná
Câmpus Ponta Grossa
Diretoria de Graduação e Educação Profissional
Departamento Acadêmico de Informática
Tecnologia em Análise e Desenvolvimento de Sistemas



TERMO DE APROVAÇÃO

SISTEMA PARA AGENDAMENTO DE SERVIÇOS

por

RAMON WOLSKI KIERAS

Este Trabalho de Conclusão de Curso (TCC) foi apresentado em 04 de junho de 2019 como requisito parcial para a obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas. O candidato foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

Prof. Dr. Richard Duarte Ribeiro
Orientador

Prof. MSc. Vinícius Camargo Andrade
Membro titular

Prof. MSc. Rogério Ranthum
Membro titular

Prof. MSc. Geraldo Ranthum
Responsável pelo Trabalho de Conclusão de
Curso

Prof. Dr. André Pinz Borges
Coordenador do curso

Tudo tem o seu tempo determinado, e há tempo para todo o propósito debaixo do céu.

Há tempo de nascer, e tempo de morrer; tempo de plantar, e tempo de arrancar o que se plantou;

Tempo de matar, e tempo de curar; tempo de derrubar, e tempo de edificar;

Tempo de chorar, e tempo de rir; tempo de prantear, e tempo de dançar;

Tempo de espalhar pedras, e tempo de ajuntar pedras; tempo de abraçar, e tempo de afastar-se de abraçar;

Tempo de buscar, e tempo de perder; tempo de guardar, e tempo de lançar fora;

Tempo de rasgar, e tempo de coser; tempo de estar calado, e tempo de falar;

Tempo de amar, e tempo de odiar; tempo de guerra, e tempo de paz.

(Eclesiastes 3:1-8)

RESUMO

KIERAS, Ramon W. **Sistema para agendamento de serviços**. 2019. 50 f. Trabalho de Conclusão de Curso (Tecnologia em Análise e Desenvolvimento de Sistemas) - Universidade Tecnológica Federal do Paraná. Ponta Grossa, 2019.

Profissionais que dependem de agendamentos prévios para realização de suas atividades como médicos, cabeleireiros, manicures e massagistas estão buscando cada vez mais divulgar seus serviços, visando facilitar a procura por seus serviços e o atendimento entre seus clientes. Este trabalho apresenta o desenvolvimento de uma solução *web* e *mobile*, responsável pela gestão de agendamentos de serviços prestados por profissionais, principalmente daqueles que necessitam de um agendamento prévio para a realização de seus serviços. O mesmo também possibilita a busca por parte de clientes que procuram por determinados serviços em uma cidade específica, facilitando a comparação entre valores, detalhes dos serviços e disponibilidade de horários. Para isso, foi utilizado o *framework* Laravel responsável em fornecer os dados que serão requisitados pelo cliente *web*, desenvolvido em Angular e pelo aplicativo móvel, construído por meio do *framework* Ionic utilizando da arquitetura REST.

Palavras-chave: Solução *Web*. Móvel. Serviço. Agendamento. Aplicativo.

ABSTRACT

KIERAS, Ramon W. **System for scheduling services**. 2019. 50 p. Work of Conclusion Course (Graduation in Tecnologia em Análise e Desenvolvimento de Sistemas) - Federal Technology University - Paraná. Ponta Grossa, 2019.

Professionals who depend on prior schedules to perform their activities as doctors, hairdressers, manicures and masseurs are increasingly seeking to publicize their services, seeking to facilitate the search for their services and care among their clients. This work presents the development of a web and mobile solution, responsible for the management of the schedule of services provided by professionals, mainly by those who need a prior appointment to perform their services. It also makes it possible for clients to search for certain services in a specific city, facilitating the comparison between values, details of services and availability of schedules. For its development Laravel framework was used, which was responsible for providing the data that will be requested by the web client, developed in Angular and by the mobile application, built through the Ionic framework using the REST architecture.

Keywords: Schedules. Mobile. Services. Professionals. Application.

LISTA DE ILUSTRAÇÕES

Figura 1 - Arquitetura do projeto.....	9
Figura 2 - Aplicativo Boa Consulta.....	10
Figura 3 - Sistema <i>web</i> Reservio.....	11
Figura 4 - Sistema <i>web</i> AgendaAi.....	12
Figura 5 - Desenvolvimento Incremental.....	14
Figura 6 - Diagrama de caso de uso do projeto.....	16
Figura 7 - Diretório de <i>controllers</i> do cliente <i>web</i>	21
Figura 8 - Diretório de arquivos HTML do cliente <i>web</i>	22
Figura 9 - Diretório dos arquivos do Ionic.....	23
Figura 10 - Diretório dos <i>Controllers</i> e <i>Models</i> do servidor.....	25
Figura 11 - Diretório do arquivo de rotas do servidor.....	26
Figura 12 - Cadastro de profissional.....	28
Figura 13 - Cadastro de serviços oferecidos pelo profissional.....	28
Figura 14 - Perfil do profissional.....	29
Figura 15 - Cadastro de Profissional.....	29
Figura 16 - Cadastro de cliente.....	31
Figura 17 - Perfil do cliente.....	32
Figura 18 - Pesquisa por profissional e serviço.....	32
Figura 19 - Agendamento de serviço.....	34
Figura 20 - Agendamento de serviço.....	34
Figura 21 - Agendamentos pendentes.....	35
Figura 22 - Detalhes do agendamento <i>web</i>	35
Figura 23 - Atendimentos Pendentes.....	36

LISTA DE QUADROS

Quadro 1 – Comparação entre as funcionalidades.....	12
Quadro 2 – Descrição resumida do diagrama de caso de uso do sistema.....	17
Quadro 3 – Planejamento das funcionalidades.....	18
Quadro 4 – Funcionalidades do 1o. Incremento.....	19
Quadro 5 – Funcionalidades do 2o. Incremento.....	24
Quadro 6 – Funcionalidades do 3o. Incremento.....	27
Quadro 7 – Funcionalidades do 4o. Incremento.....	30
Quadro 8 – Funcionalidades do 5o. Incremento.....	33

LISTA DE SIGLAS

API	<i>Application Programming Interface</i>
CSS	<i>Cascading Style Sheets</i>
HTML	<i>HyperText Markup Language</i>
HTTP	<i>Hypertext Transfer Protocol</i>
IOS	<i>Iphone Operating System</i>
JDK	<i>Software Development Kit</i>
JSON	<i>Javascript Object Notation</i>
MVC	<i>Model-View-Controller</i>
PHP	<i>Personal Home Page</i>
SMS	<i>Short Message Service</i>
URL	<i>Uniform Resource Locator</i>

LISTA DE ACRÔNIMOS

BRASSCOM	Associação Brasileira das Empresas de Tecnologia da Informação e Comunicação
----------	--

SUMÁRIO

1 INTRODUÇÃO.....	1
1.1 OBJETIVOS.....	2
2 REFERENCIAL TEÓRICO.....	3
2.1 A TECNOLOGIA DA INFORMAÇÃO NO COTIDIANO DAS PESSOAS.....	3
2.2 O DESENVOLVIMENTO DE SISTEMAS <i>WEB</i>	3
2.3 <i>FRAMEWORKS</i> PARA DESENVOLVIMENTO <i>WEB</i>	5
2.3.1 Laravel.....	5
2.3.2 Angular.....	6
2.4 <i>FRAMEWORK</i> PARA DESENVOLVIMENTO DE APLICATIVOS HÍBRIDOS.....	6
2.4.1 Ionic.....	7
2.5 SISTEMAS DISTRIBUÍDOS.....	7
2.5.1 JSON.....	8
2.5.2 Protocolo HTTP.....	8
2.5.3 <i>Web Service</i>	9
2.5.4 Arquitetura REST.....	9
2.5.5 Análise de sistemas similares.....	10
2.5.7 Metodologia de desenvolvimento.....	13
3 ANÁLISE DE REQUISITOS.....	15
3.1 DEFINIÇÃO DOS REQUISITOS.....	15
3.2 DIAGRAMA DE CASO DE USO.....	16
4 DESENVOLVIMENTO.....	18
4.1 PLANEJAMENTO DAS FUNCIONALIDADES.....	18
4.2 INCREMENTO 01 – INTERFACES <i>WEB</i> E <i>MOBILE</i>	19
4.2.1 Funcionalidades do Incremento.....	19
4.2.2 Desenvolvimento.....	19
4.2.3 Interfaces <i>Web</i>	20
4.2.4 Interfaces <i>Mobile</i>	22
4.2.5 Resultados.....	23
4.3 INCREMENTO 02 – <i>WEB SERVICE</i>	23
4.3.1 Funcionalidades do Incremento.....	24
4.3.2 Desenvolvimento.....	24
4.3.3 Resultados.....	26
4.4 INCREMENTO 03 – FUNCIONALIDADES DOS PROFISSIONAIS.....	26
4.4.1 Funcionalidades do Incremento.....	26
4.4.2 Desenvolvimento.....	27
4.4.3 Resultados.....	27
4.5 INCREMENTO 04 – FUNCIONALIDADES DOS CLIENTES.....	30
4.5.1 Funcionalidades do Incremento.....	30
4.5.2 Desenvolvimento.....	30

4.5.3 Resultados.....	31
4.6 INCREMENTO 05 – FUNCIONALIDADES DE AGENDAMENTO.....	32
4.6.1 Funcionalidades do Incremento.....	33
4.6.2 Desenvolvimento.....	33
4.6.3 Resultados.....	33
5 CONCLUSÃO.....	37
5.1 TRABALHOS FUTUROS.....	37
REFERÊNCIAS.....	38

1 INTRODUÇÃO

Segundo Hayde e Webster (2014), as novidades tecnológicas móveis reforçam a individualidade das pessoas em suas rotinas. As pessoas se tornaram muito dependentes de seus *smartphones*, *tablets* e demais dispositivos, e usam seus aparelhos em suas tarefas diárias como se fossem uma extensão de seu corpo. Para (FRANCO, 2014), isso pode ser positivo em diversos pontos. Pessoas que fazem uso de muitas informações digitais em seu cotidiano, encontram uma maior facilidade em realizar inúmeras funções do dia a dia, de forma a maximizar suas tarefas por meio desses dispositivos.

A BRASSCOM (2016), apontou que no ano de 2016 o Brasil foi o quarto maior mercado mundial de dispositivos móveis e que “no Brasil e nos EUA, o tempo gasto em dispositivos móveis superou o consumo de televisão” (BRASSCOM, 2016). Borges (2013) diz que a modernidade vem alcançando altos patamares, sendo pertinente aderir ao uso de dispositivos móveis que possibilitam aumento de liberdade, independência, planejamento, continuidade e autonomia.

Por meio do uso de *smartphones*, Alcantra (2011) diz que a mobilidade permite ter em mãos serviços, informações, comunicação e entretenimento, incluindo consultas bancárias, previsão de tempo, notícias, redes sociais e operações em tempo real. O autor destaca que por meio dessa mobilidade, o cliente pode optar em não sair de casa para realizar uma compra ou até mesmo o pagamento de uma dívida. Com o crescimento tecnológico aliado aos *smartphones*, as pessoas podem realizar estas e dezenas de outras tarefas para facilitar e agilizar sua rotina.

Estas tecnologias, segundo Borges (2013), contribuem de forma importante para a flexibilização do tempo das pessoas, tornando-as mais livres e com mais autonomia. Elas estão relacionadas diretamente com a ideia de uma maior mobilidade associada a uma maior sobreposição dos tempos e espaços profissional e pessoal.

Desse modo, torna-se viável a criação de um aplicativo para agendamento de serviços de profissionais como médicos, dentistas, psicólogos, cabeleireiros, manicures, massagistas ou outro que necessite de um agendamento prévio para atendimento.

Verificou-se também em uma consulta informal por meio do site google.com que ao se pesquisar por um profissional, poucos dados sobre o seu serviço estavam disponíveis. Existe, em geral, a necessidade de realizar-se ligações telefônicas adicionais para diversos profissionais para que se possa comparar valores, detalhes dos serviços, disponibilidade de horários, características de pagamentos, dentre outras.

Todas essas ações geram um gasto de tempo desnecessário. Essa tarefa poderia ser simplificada em um sistema que possuísse em seu banco de dados diversos profissionais de várias cidades, acompanhado do valor do serviço e horários disponíveis. Para o usuário que buscasse um atendimento seria necessário apenas filtrar o profissional por cidade, profissão e serviço, e em seguida, selecionar a melhor opção e confirmar o agendamento para a data e horário pretendidos.

Posto isto, outrossim, o sistema facilitaria a procura por profissionais, agendamentos de consultas e possivelmente os pagamentos. Essa seria uma forma de substituir o sistema tradicional de agendamento de profissionais e que provavelmente traria um impacto positivo em sua utilização.

1.1 OBJETIVOS

Para a elaboração deste projeto teve-se como objetivo geral o desenvolvimento de uma aplicação *web* e um aplicativo *mobile* para que profissionais autônomos possam disponibilizar, com mais facilidade, informações relacionadas a seus serviços, e para que seus clientes consigam realizar agendamentos com maior facilidade.

Associado ao objetivo geral foram determinados alguns objetivos específicos, citados abaixo:

- Desenvolvimento de uma aplicação de gestão *web*, em que profissionais poderão administrar os serviços prestados a seus clientes.
- Desenvolvimento de um aplicativo *mobile* utilizando abordagem híbrida com o objetivo de facilitar a consulta e o agendamento de serviços prestados por profissionais.
- Construção de uma solução em que a aplicação *web* se comunica com o aplicativo *mobile* transferindo dados para o seu funcionamento.

2 REFERENCIAL TEÓRICO

Neste capítulo são discutidos os seguintes assuntos: a tecnologia da informação no cotidiano das pessoas; desenvolvimento de sistemas *web*; *frameworks* para desenvolvimento *web*; Laravel; Angular; *frameworks* para desenvolvimento de aplicativos híbridos; Ionic.

2.1 A TECNOLOGIA DA INFORMAÇÃO NO COTIDIANO DAS PESSOAS

Segundo Kohn (2007), a sociedade vivenciou (e parece que ainda vive) o que ele chama de *Era Digital*. Na qual os computadores ocupam um espaço essencial em nosso modelo de sociabilidade, englobando vários setores da sociedade, comércio, política, serviços, entretenimento, informação e relacionamentos. Seus resultados transformaram nosso cotidiano, facilitando nossas atividades e buscando melhorar cada vez mais nossas vidas.

Para Kohn (2007), por meio das tecnologias digitais temos uma nova possibilidade de acesso a produtos, transmissão, arquivos e a informação. Toda essa transformação acaba influenciando nossa economia, política e nosso cenário social. A criação destas novas tecnologias no setor da informática, segundo o mesmo autor, gerou um mercado cada vez mais competitivo e especializado do qual precisa estar sempre atualizado com os padrões de mercado.

Kohn (2007) ainda afirma que uma rede de comunicação entre pessoas e empresas se tornou algo essencial, pois temos a possibilidade de acessar uma grande quantidade de informações e trocar estes dados em tempo real. Por meio dos computadores, os serviços foram melhorados e tivemos uma substituição de uma porcentagem da mão de obra pela tecnologia que proporcionou novas profissões no setor da informática e das comunicações.

2.2 O DESENVOLVIMENTO DE SISTEMAS *WEB*

Yoshiura (2015) afirma que nas últimas décadas houve uma grande transformação nos meios de comunicação, caracterizada pela chegada da Internet que ocasionou grandes mudanças no desenvolvimento de sistemas da informação.

Por meio do avanço tecnológico, o uso da Internet apenas como meio de divulgação de informação foi transformado em uma plataforma de comunicação e desenvolvimento.

Deshpande (2002) relacionou este contexto com o surgimento de um novo modelo de desenvolvimento de sistemas: a criação de soluções *web*. As primeiras aplicações *web* eram constituídas de páginas com conteúdo estático, institucional e textual. Diferente dos sistemas atuais onde encontramos soluções maiores e complexas, disponibilizando dados dinâmicos, permitindo acesso a banco de dados e se comunicando com outras aplicações.

As aplicações *web* necessitam de uma estrutura para a apresentação de suas informações, e para isso é usada a *HyperText Markup Language* (HTML). Costa (2007) define HTML como uma linguagem padrão para acessar as páginas da Internet. Suas linhas de código são analisadas pelo navegador que apresenta o resultado para o usuário. A linguagem HTML é composta por textos e por códigos especiais chamados de *tags*. Para visualizar os resultados de um código HTML basta se ter um navegador para interpretar o código.

Assim como o HTML é uma linguagem importante para o desenvolvimento *web*, Costa (2007) apresenta o JavaScript como sendo uma linguagem de *script*¹ capaz de auxiliar na criação de páginas interativas. A partir dela é possível exibir mensagens e efeitos dinâmicos em uma página *web*. Para utilização do JavaScript é necessário apenas de um navegador que tenha suporte para essa linguagem.

Outra ferramenta essencial no desenvolvimento *web* é o uso de páginas dinâmicas. Segundo Niederauer (2004), PHP é uma das linguagens mais utilizadas na Internet, e seu principal destaque está na capacidade de atuar no meio *web*, transformando páginas estáticas em dinâmicas.

O autor explica que o que diferencia as páginas estáticas, é que elas utilizam apenas a linguagem HTML, sendo necessário alterar manualmente cada nova informação. Enquanto as páginas utilizando a linguagem PHP fazem este processo automaticamente, sendo necessário apenas a criação de um banco de dados em que será armazenado estas informações. Além disso, PHP é uma linguagem gratuita, e o seu arquivo de instalação pode ser encontrado na Internet, junto de uma vasta documentação sobre o seu funcionamento.

¹ Sequência de comandos e tarefas que devem ser executados.

Outra característica apontada por Niederauer (2004), é que o PHP é executado no lado servidor, não utilizando os recursos do computador do cliente. Ao acessar uma página em PHP por meio de um navegador, todo o código fonte é executado no servidor, que envia em seguida o resultado para exibição no navegador.

Contudo, o desenvolvimento *web* pode ser facilitado por meio do uso de *frameworks*, os quais serão explicados a seguir.

2.3 FRAMEWORKS PARA DESENVOLVIMENTO WEB

Um *framework*, segundo Alvim (2010), é uma coleção de classes que trabalham em conjunto para facilitar o desenvolvimento de uma solução. Evitando utilizar diversas soluções para um mesmo tipo de problema.

Como uma das vantagens no uso de *frameworks*, Minetto (2007) cita a utilização das mesmas convenções, classes e bibliotecas, tornando assim a manutenção do sistema muito mais fácil, mesmo que o código tenha sido desenvolvido por outra equipe. Outra vantagem descrita por Minetto (2007) é a automatização das tarefas para o reaproveitamento de um mesmo conteúdo.

2.3.1 Laravel

Laravel é um *framework* PHP *open-source*² para desenvolvimento de aplicações *web*. Trabalha com um padrão de arquitetura MVC (*Model-View-Controller*) e tem como destaque a construção de soluções seguras e de alto desempenho, de forma rápida e simplificada (LARAVEL, 2018).

Laravel utiliza o padrão de arquitetura MVC que, segundo Verma (2014), é usado para aumentar a modularidade de sistemas da informação, sendo dividida em três camadas:

- *Model*: Responsável por manipular os dados do sistema, realizando o intermédio do banco de dados e a aplicação.

² Sistema computacional que pode ser executado, copiado, modificado e redistribuído pelos usuários gratuitamente.

- *View*: Faz a comunicação com o usuário por meio de interfaces, exibindo os dados que o *controller* recebe do *model*.
- *Controller*: Controla o fluxo de solicitações dos usuários por meio da *view* para atualização de dados. Responsável pelo intermédio entre o *model* e a *view*.

2.3.2 Angular

O Angular é um *framework* JavaScript de código aberto, desenvolvido pela Google para facilitar a criação de aplicativos na *web*, possibilitando a implantação em diversas plataformas, como, páginas *web*, aplicativos móveis nativos e *desktop* nativo (ANGULAR, 2018).

2.4 FRAMEWORK PARA DESENVOLVIMENTO DE APLICATIVOS HÍBRIDOS

Prezotto (2014) relata que graças a um grande crescimento de dispositivos móveis ao longo dos últimos anos, a diversidade de equipamentos também cresceu. Causando um aumento no número de ambientes e linguagens de programação para atender a demanda destes dispositivos. Tal situação dificulta o desenvolvimento de uma aplicação, já que é necessário conhecimento da linguagem de cada sistema operacional, além de ser necessário desenvolver um aplicativo diferente para cada plataforma específica.

A partir disso, Prezotto (2014) afirma que com a utilização de *framework* de desenvolvimento híbrido, é possível criar um único aplicativo que será utilizado em diversas plataformas. Estes *frameworks* são responsáveis por empacotar o código-fonte para várias plataformas, disponibilizando um arquivo para ser executado pelo dispositivo. Além da possibilidade de acessar recursos nativos, como a câmera e a localização do dispositivo.

Para Gonçalves (2017), a escolha de um *framework* de desenvolvimento *mobile* é de grande importância e o levantamento de requisitos do projeto são imprescindíveis antes da escolha do *framework*, visto que, alguns produtos possuem menos recursos do que outros.

Gonçalves (2017) destaca que uma aplicação híbrida atende grande parte dos requisitos exigidos em uma aplicação para dispositivos móveis, principalmente por meio dos seus recursos nativos. Gonçalves (2017) demonstra também que aplicativos híbridos, construídos com os recursos de uma solução *web*, podem ser compartilhados em lojas de aplicativos como se fossem nativos.

O código desenvolvido em uma aplicação híbrida é executado dentro de um componente chamado *webview*, que é uma espécie especial de navegador. Porém essa execução da *view* não é percebida pelo usuário, passando uma impressão que o aplicativo é nativo. O funcionamento de uma aplicação híbrida consiste na junção do código nativo para empacotamento e distribuição da aplicação com o código não nativo (HTML, CSS e JavaScript), responsável pela interface e funções da aplicação (GONÇALVES, 2017).

2.4.1 Ionic

O Ionic é um *framework* de desenvolvimento de aplicativos híbridos que segundo Gonçalves (2017), está em alta no mercado devido à sua grande produtividade e facilidade na utilização de seus recursos, já que grande parte dos programadores conhecem as linguagens exigidas pelo *framework* (HTML, CSS, Javascript). Por meio destas linguagens é possível desenvolver um código que será exportado como aplicativo para ser executado em plataformas como Android, IOS e Windows Phone a partir do Ionic.

2.5 SISTEMAS DISTRIBUÍDOS

Coulouris (2013) define sistemas distribuídos como sendo aqueles em que os componentes de *hardware* e *software*, presentes em computadores de uma mesma rede, se comunicam por meio de troca de mensagens para realizar suas ações. Sua definição possui as seguintes consequências para o seu funcionamento:

Concorrência: onde vários processos ocorrem de forma concorrente, compartilhando recursos como páginas *web* e arquivos dentro de uma mesma rede entre vários usuários.

Inexistência de relógio global: Quando os programas se comunicam por meio da troca de mensagens, é necessário uma noção compartilhada de tempo em que estas ações acontecem. Porém, existem limites em relação a precisão em que estes computadores se sincronizam em uma rede. Não havendo uma percepção correta do tempo em que as comunicações ocorrem (COULOURIS, 2013).

Falhas independentes: Durante uma falha na rede, os programas podem não perceber imediatamente a existência de uma falha ou a perda de comunicação com um determinado componente. As falhas ocorrem independentemente e não prejudicam o funcionamento dos demais.

2.5.1 JSON

Miletto (2014) define JSON como sendo um formato responsável pela troca de dados e de notação simples. Seus dados são formados com pares de chaves e separadas por vírgulas, exceto para a definição de *arrays* que são acrescentados colchetes delimitadores para a sua representação. Por dessa notação, os dados em JSON podem reproduzir coleções inteiras de objetos.

2.5.2 Protocolo HTTP

O protocolo HTTP segundo Coulouris (2013), é responsável por estabelecer de qual forma os navegadores e outros tipos de clientes se comunicarão com os seus servidores. Ele destaca as principais características como sendo:

Interações requisição-resposta: Em que o cliente envia uma mensagem para o servidor a partir de uma URL solicitando algum recurso, em seguida o servidor busca pelo nome do caminho e responde por meio de uma mensagem. Caso ele não encontre este caminho, ele retorna uma mensagem de erro.

Tipos de conteúdo: Coulouris (2013) descreve que os navegadores não possuem a capacidade de manipular qualquer tipo de arquivo, sendo assim, quando o navegador requisita algum recurso, ele o seleciona a partir de uma lista dos tipos de conteúdo que ele suporta. Estas ações são usadas para determinar qual o tipo de

conteúdo pode ser configurado, e os leitores precisam fazer a verificação dessas configurações nos navegadores.

Um recurso por requisição: Dentro do protocolo HTTP cada cliente seleciona um recurso por requisição e o navegador tende a enviar sempre requisições separadas (COULOURIS, 2013).

2.5.3 *Web Service*

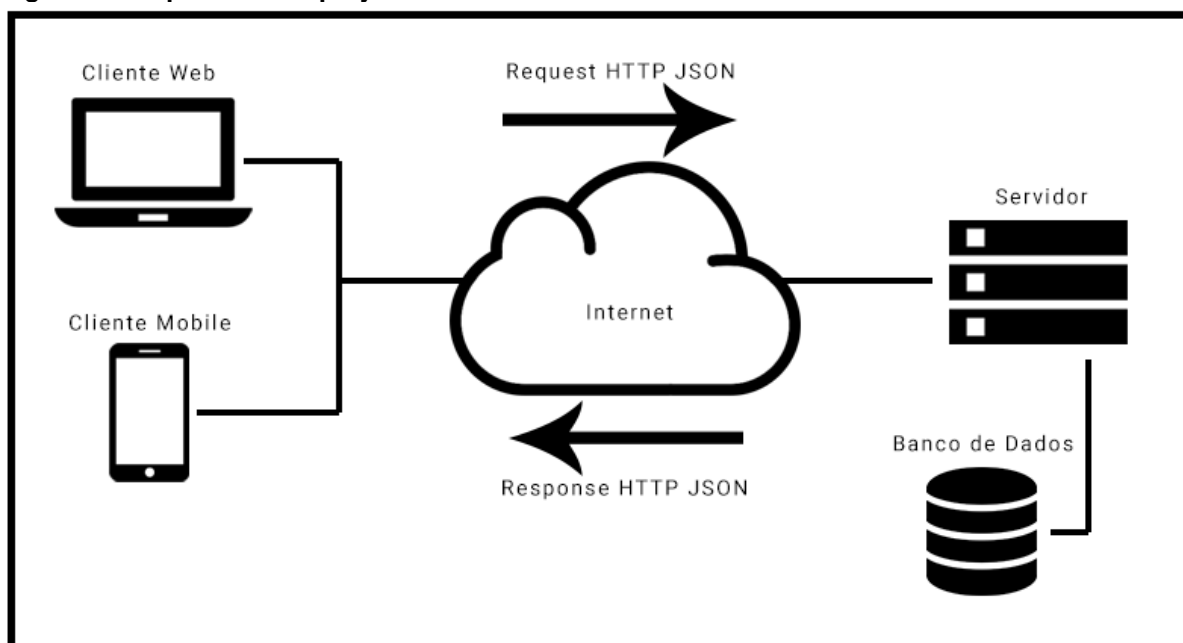
Coulouris (2013) define *web service* como sendo um fornecedor de interface de serviço que permite aos clientes se comunicarem com os servidores. Estes clientes solicitam operações a partir de *requests*³ normalmente transmitidas por meio do protocolo HTTP para sua comunicação.

2.5.4 Arquitetura REST

Segundo coulouris (2013), REST pode ser definido como uma forma de estilo arquitetural em que seus clientes utilizam URLs e métodos HTTP (GET, PUT, DELETE e POST) para realizar suas comunicações, utilizando recursos de dados no lugar de interfaces. Assim que um recurso é criado, ele obtêm uma nova URL para ser acessada por um cliente que recebe um estado inteiro deste recurso em vez de apenas um pedaço. A Figura 1 apresenta a arquitetura REST utilizada no desenvolvimento do projeto. No qual os clientes *web* e *mobile* realizam requisições para o servidor, que busca os dados a partir do banco de dados e retorna os dados que foram solicitados para cada cliente.

³ Informações enviadas para o servidor.

Figura 1 - Arquitetura do projeto



Autoria própria

2.5.5 Análise de sistemas similares

Foi realizada uma pesquisa a partir de um *website* de buscas para localizar sistemas similares a este projeto. Como resultado, encontramos algumas soluções semelhantes, porém com uma quantidade inferior de funcionalidades em relação ao sistema proposto. Além disso, alguns sistemas não possuem uma versão móvel. Sendo assim, a criação de uma solução com versões *web* e *mobile*, abrangendo vários tipos de profissionais foi reforçada.

O aplicativo Boa Consulta⁴ é uma solução com uma versão *web* e *mobile* que permite ao usuário consultar apenas médicos e dentistas. Por meio dele é informado o endereço e o telefone para contato. Para realizar o agendamento é necessário escolher os horários disponíveis pelo profissional e assim que selecionado, é enviado uma mensagem via SMS para confirmar o agendamento por parte do profissional. A Figura 2 ilustra o aplicativo Boa Consulta.

⁴ <https://www.boaconsulta.com>

Figura 2 - Aplicativo Boa Consulta

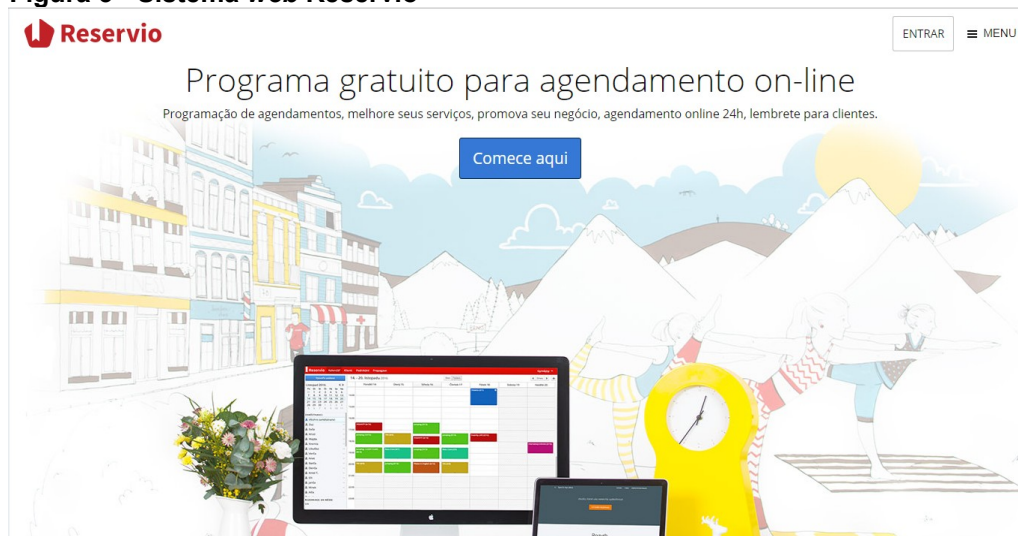


Fonte: Autoria própria

O sistema Reservio⁵ é uma solução *web* de agendamentos que atende diversas áreas de profissionais, possuindo diversos planos para a utilização do sistema e em destaque um plano gratuito. Além de contar com uma interface visualmente pouco amigável, a mesma não possui uma versão móvel, dificultando o acesso por parte dos usuários de *smartphones*. A Figura 3 apresenta o sistema *web* Reservio.

⁵ <https://www.reservio.com>

Figura 3 - Sistema web Reservio



Fonte: Autoria própria

A solução AgendaAi⁶, assim como o Reservio, é um sistema de agendamentos *online* para diversos tipos de profissionais. Possui uma interface bastante amigável e de fácil entendimento, porém não possui uma versão para dispositivos *móveis*. A Figura 4 exibe o sistema web AgendaAi.

Figura 4 - Sistema web AgendaAi



Fonte: Autoria própria

Tendo em vista esta pesquisa informal simulando uma busca por soluções semelhantes percebeu-se a necessidade de desenvolver um sistema com o objetivo

⁶ <https://www.agendaai.com.br>

de melhorar o atendimento de vários profissionais e, ao mesmo tempo, facilitar a procura por parte de seus clientes, tornando o processo mais simples e rápido.

O Quadro 1 realiza uma comparação das funcionalidades entre as três soluções similares encontradas.

Quadro 1 – Comparação entre as funcionalidades

Funcionalidade	Boa Consulta	Reservio	AgendaAi
Versão móvel	Sim	Não	Não
Agendamento por meio de SMS	Sim	Não	Não
Agendamento por meio do sistema	Não	Sim	Sim
Designer agradável	Sim	Não	Sim
Versão gratuita	Sim	Sim	Não
Abrange vários tipos de profissionais	Não	Sim	Sim

Fonte: Autoria própria

Todas as soluções citadas a cima possuem funcionalidades diferentes das demais. O sistema Boa Consulta é a única solução que é compatível com dispositivos móveis e realiza seus agendamentos através do envio de SMS. Porém abrange apenas profissões como médicos e dentistas.

A solução Reservio, assim como a solução AgendaAi realiza seus agendamentos para diversos tipos de profissionais e o processo de agendamento acontece através de suas aplicações. Porém, o sistema AgendaAi não possui uma versão gratuita para a sua utilização e a solução Reservio possui um designer pouco agradável que dificulta a usabilidade do usuário.

2.5.6 Restrições do projeto

Ainda que o *framework* Ionic por meio da ferramenta Apache Cordova possibilite a compilação de aplicativos para diversas plataformas móveis, este

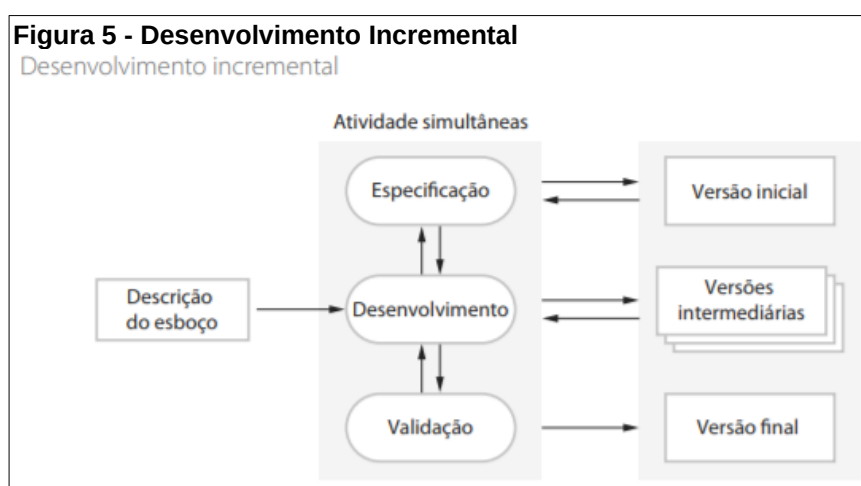
projeto terá como limitação o desenvolvimento apenas para a plataforma Android. Já que, segundo Kasperbauer (2013), o IOS exige uma licença de desenvolvedor para executar aplicativos e, além disso, seu SDK ser executado somente em computadores Mac, dificultado o seu desenvolvimento.

A plataforma Android possui um sistema operacional de código-fonte aberto, não exigindo uma licença para desenvolvimento ou um computador com sistema operacional específico (DE MENDONÇA; BITTAR; DE SOUZA DIAS, 2011).

Considerando-se as restrições em relação aos objetivos deste projeto, serão desenvolvidas apenas as regras de negócios em relação ao agendamento entre os profissionais e seus clientes, não sendo implementado nenhuma forma de pagamento *online*. Essa restrição foi aplicada levando em consideração o tempo disponível para o desenvolvimento do sistema.

2.5.7 Metodologia de desenvolvimento

Para o planejamento deste projeto foi utilizado o modelo de desenvolvimento incremental conforme mostra a Figura 5, que segundo SOMMERVILLE (2011), consiste na criação de novas versões do projeto até que o sistema ideal seja atendido.



Fonte: Autoria própria

SOMMERVILLE (2011) descreve o desenvolvimento incremental de software como uma parte fundamental das abordagens ágeis. Ideal para quando não se tem

uma solução do problema com antecedência. Sendo necessário buscar a solução a passo a passo e recuar ao perceber erros.

Durante o desenvolvimento de um sistema de forma incremental é mais eficiente fazer mudanças durante o estágio de desenvolvimento. A cada passo é incrementado funcionalidades ao sistema. As funcionalidades mais importantes são desenvolvidas nos incrementos iniciais, possibilitando assim ao cliente no início do desenvolvimento, analisar se os requisitos exigidos estão sendo atendidos. Caso uma funcionalidade precise ser modificada, apenas o incremento que estiver em desenvolvimento no momento deverá ser alterada e uma nova funcionalidade deverá ser definida nos próximos incrementos.

O desenvolvimento incremental não é ideal para os sistemas grandes e complexos, onde existem várias equipes desenvolvendo diversas partes do sistema, pois todo o planejamento deve ser feito com antecedência e não de forma incremental.

3 ANÁLISE DE REQUISITOS

O desenvolvimento de *software* engloba várias fases, como a definição dos requisitos, modelagem, desenvolvimento, implantação e manutenção. Por meio desses passos é possível organizar de uma melhor forma o desenvolvimento do sistema e atender a seus objetivos (SOMMERVILLE, 2011).

3.1 DEFINIÇÃO DOS REQUISITOS

Requisitos são descrições das funcionalidades que o *software* deve desempenhar, além de suas características mínimas para o funcionamento (SOMMERVILLE, 2011).

O principal objetivo do sistema proposto é facilitar a procura de serviços ofertados por profissionais e realizar o agendamento dos seus atendimentos por meio de uma interface *online*. A partir dessa premissa foi realizada uma análise do problema e identificado os seguintes requisitos:

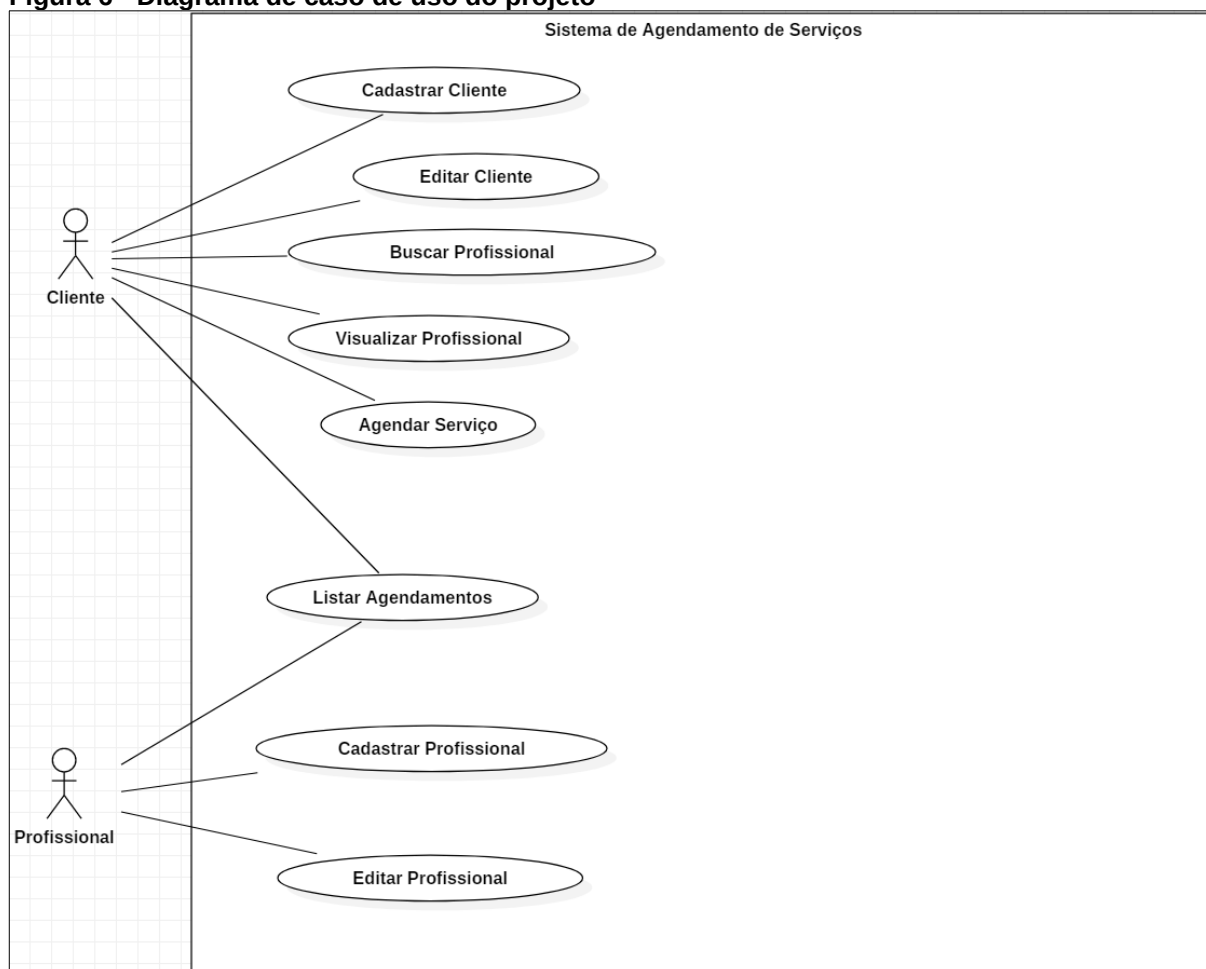
- Clientes deveriam poder realizar o seu cadastro no sistema, a edição de suas informações de perfil e efetuar buscas por serviços e profissionais.
- Ao acessar o perfil de um profissional, o cliente deveria poder visualizar os serviços ofertados, além de realizar o agendamento dos mesmos.
- Assim que o agendamento fosse realizado, o cliente deveria possuir a opção de visualizar as informações do agendamento e de realizar o cancelamento caso seja necessário.
- Profissionais deveriam poder cadastrar seus perfis e editar suas informações, acrescentando ou removendo serviços conforme a necessidade.
- Assim que um cliente solicitasse o agendamento de um serviço, deveria ser exibido uma notificação deste serviço na página inicial do profissional, ficando a critério do mesmo a confirmação ou o cancelamento do agendamento.
- Todo o agendamento que fosse solicitado deveria ser confirmado pelo profissional para que seja validado, caso contrário o agendamento ficaria pendente de confirmação.
- Durante um agendamento tanto o cliente como o profissional deveriam ter a possibilidade de cancelamento, caso fosse necessário.

- Assim que a data e horário de um agendamento fosse atingido, o sistema deveria solicitar a confirmação por parte do cliente, para que o *status* do agendamento fosse concluído.

3.2 DIAGRAMA DE CASO DE USO

Os diagramas de caso de uso facilitam o entendimento das funcionalidades e características do sistema por parte do usuário. Ele representa de que forma o usuário interage com o sistema e estabelece os passos necessários para realizar uma tarefa específica (PRESSMAN, 2011). A Figura 6 apresenta o diagrama de caso de uso do sistema proposto.

Figura 6 - Diagrama de caso de uso do projeto



Fonte: Autoria própria

O Quadro 2 a seguir apresenta a descrição resumida das funcionalidades de cada caso de uso do sistema.

Quadro 2 – Descrição resumida do diagrama de caso de uso do sistema

Caso de uso	Descrição resumida do caso de uso
Cadastrar Cliente	Permite que o cliente realize o cadastro no sistema.
Editar Cliente	Permite que o cliente realize a edição do seu perfil.
Buscar Profissional	Permite que o cliente realize buscas por profissionais que estão cadastrados no sistema.
Visualizar Profissional	Permite que o cliente visualize o perfil de um profissional.
Listar Agendamentos	Permite que um profissional ou cliente consulte todos os seus agendamentos realizados no sistema.
Agendar Serviço	Permite que o cliente realize o agendamento de um serviço oferecido por um profissional.
Cadastrar Profissional	Permite que o profissional realize o cadastro no sistema.
Editar Profissional	Permite que o profissional realize a edição do seu perfil.

Fonte: Autoria própria

4 DESENVOLVIMENTO

Este capítulo é destinado à descrição dos passos realizados para o desenvolvimento do projeto, bem como, os resultados obtidos. Como descrito anteriormente, o método de desenvolvimento abordado foi o incremental, onde as funcionalidades foram agrupados por diferentes conjuntos de incrementos.

4.1 PLANEJAMENTO DAS FUNCIONALIDADES

Antes da codificação do projeto, é necessário um planejamento das metas a serem alcançadas para a criação do sistema. Sendo assim, elaborou-se o planejamento das funcionalidades, onde foram definidas as tarefas a serem desempenhadas, agrupadas por incrementos e ordenadas por prioridade, como mostra o Quadro 3.

Quadro 3 – Planejamento das funcionalidades

Nº	Meta	Descrição	Incremento
1	Interface do modulo de clientes na <i>web</i> e <i>mobile</i> .	Criação da interface de cadastro, edição, busca e perfil de clientes.	1
2	Interface do modulo de profissionais na <i>web</i> e <i>mobile</i> .	Criação da interface de cadastro, edição, busca e perfil de profissionais.	1
3	Interface do modulo de agendamento na <i>web</i> e <i>mobile</i> .	Criação da interface de criação, edição e conclusão de agendamentos.	1
4	Interface de login no sistema <i>web</i> e <i>mobile</i>	Criação da interface de <i>login</i> no sistema.	1
5	Construção do servidor e banco de dados.	Criar a estrutura de comunicação entre os clientes e o servidor e banco de dados.	2
6	Função de <i>login</i> no sistema <i>web</i> e <i>mobile</i>	Criação da função de <i>login</i> no sistema.	2
7	Funções do modulo de profissionais na <i>web</i> e <i>mobile</i> .	Criação das funções de cadastro, edição, busca e perfil de profissionais	3
8	Funções do modulo de clientes na <i>web</i> e <i>mobile</i> .	Criação das funções de cadastro, edição, busca e perfil de clientes.	4
9	Funções do modulo de agendamento na <i>web</i> e <i>mobile</i>	Criação das funções de criação, edição e conclusão de agendamentos.	5

Fonte: Autoria própria

4.2 Incremento 01 – INTERFACES WEB E MOBILE

No primeiro incremento foram construídas as interfaces básicas que são alimentadas pelos dados vindos do servidor. Além disso, foi criada toda a estrutura de pastas para o funcionamento do *front-end web*, assim como o *mobile*.

4.2.1 Funcionalidades do Incremento

Para completar as metas do primeiro incremento foram realizadas as tarefas conforme o Quadro 4.

Quadro 4 – Funcionalidades do 1º. Incremento

1º Incremento	
Metas	Tarefas
Interface do modulo de clientes na <i>web</i> e <i>mobile</i> .	Formulário de cadastro de clientes
	Formulário de edição de clientes
	Perfil do cliente
	Busca de clientes
Interface do modulo de profissionais na <i>web</i> e <i>mobile</i> .	Formulário de cadastro de profissional
	Formulário de edição de profissional
	Perfil do profissional
	Busca de profissionais
Interface do modulo de agendamento na <i>web</i> e <i>mobile</i>	Agendar serviço
	Editar agendamento
	Finalizar agendamento
Interface de <i>login</i> no sistema <i>web</i> e <i>mobile</i>	Página de <i>login</i>

Fonte: Autoria própria

4.2.2 Desenvolvimento

Durante o desenvolvimento foram criados dois clientes para consumir os dados fornecidos pelo servidor. O cliente *web* que poderá ser acessado pelos usuários de navegadores e o cliente *mobile* que poderá ser acessado por meio de um aplicativo em *smartphones* ou *tablets*.

Para o cliente *web* foi elaborado o *front-end* dentro do *framework* JavaScript Angular, utilizando HTML 5, CSS 3 e o *framework* Bootstrap 4.

Para a construção do cliente *mobile* utilizou-se do *framework* híbrido Ionic 3 que também utiliza HTML 5, CSS 3 e JavaScript para a construção das interfaces. Estas interfaces serão utilizadas pelos usuários clientes e pelos profissionais que utilizarão o sistema para o agendamento dos serviços.

4.2.3 Interfaces *Web*

Neste incremento foram desenvolvidas todas as interfaces básicas que serão utilizadas pelos usuários para a manipulação do sistema. Além disso, foi criada por meio do *framework* Angular, a estrutura de pastas do cliente *web* como as rotas, *controllers* e *views* para a comunicação dos módulos propostos no incremento.

Para facilitar a organização dos arquivos, no diretório *controllers* foi criada uma pasta para cada módulo e dentro dela, os *controllers*, que contém as funções de validação da *view*.

Na pasta *controllers/cliente* foram criados os arquivos JavaScript responsáveis por este módulo, como o formulário de cadastro de um novo cliente, página do seu perfil, edição de seus dados e busca por profissionais. Assim como na pasta *clientes*, a pasta *controllers/profissional* contém todos os arquivos JavaScript responsáveis pela manipulação de profissionais, como cadastro, edição e perfil do profissional.

O diretório *controllers/agendamento* possui os arquivos responsáveis pelas funções de criação, edição e finalização de um agendamento de serviço por parte dos clientes para os profissionais. A pasta *utils* é destinada para os arquivos JavaScript responsáveis pelo controle de *login* e menu do sistema *web*.

O arquivo *rotas.js* é encarregado de definir todas as rotas das páginas do cliente *web*, além de realizar a conexão entre os arquivos *views* e os arquivos *controllers*. A Figura 7 apresenta o diretório contendo todos os arquivos *controllers* do cliente *web*.

Figura 7 - Diretório de *controllers* do cliente *web*

```

/idoWeb/
app
  controllers
    cliente
      cadastrar-cliente-controller.js
      editar-cliente-controller.js
      listar-cliente-controller.js
      perfil-cliente-controller.js
    profissional
      cadastrar-profissional-controller.js
      editar-profissional-controller.js
      listar-profissional-controller.js
      perfil-profissional-controller.js
    agendamento
      agendar-servico-controller.js
      editar-agendamento-controller.js
      finalizar-agendamento-controller.js
    utils
      login-controller.js
      menu.js
  app.js
  config.js
  rotas.js
  bower_components
  public
  bower.json

```

Fonte: Autoria própria

Para a codificação do HTML 5 e CSS 3 foi utilizado como complemento o *framework* Bootstrap 4 para manter a responsividade do sistema na *web*, além de proporcionar um visual agradável.

A pasta *public* contém todos os arquivos relacionados a *view*, como folhas de estilo, fontes externas e imagens, conforme mostra a Figura 8. O arquivo *index.html* é o arquivo principal, responsável por reunir todas as declarações dos componentes, *views* e *controllers* que são utilizados dentro do sistema.

A pasta *utils* possui o arquivo *login* e o arquivo menu que estão presentes em todas as páginas, como cabeçalho e rodapé do sistema. Assim como a pasta *controllers*, a pasta *páginas* possui uma pasta para cada módulo e dentro delas, os arquivos HTML que são utilizados para a interação dos usuários com o sistema. A Figura 8 mostra o diretório *public* do cliente *web* responsável por armazenar todos os arquivos html, folhas de estilo, imagens e arquivos de tipografia.

Figura 8 - Diretório de arquivos HTML do cliente web

```
/idoWeb/  
app  
bower_components  
public  
  css  
  fonts  
  images  
  js  
  paginas  
    cliente  
      cadastrar-cliente.html  
      editar-cliente.html  
      listar-cliente.html  
      perfil-cliente.html  
    profissional  
      cadastrar-profissional.html  
      editar-profissional.html  
      listar-profissional.html  
      perfil-profissional.html  
    agendamento  
      agendar-servico.html  
      editar-agendamento.html  
      finalizar-agendamento.html  
    utils  
      login.html  
      menu.html  
index.html  
bower.json
```

Fonte: Autoria própria

4.2.4 Interfaces *Mobile*

O *framework* Ionic possui um padrão de organização de seus arquivos onde os arquivos responsáveis pelas páginas do aplicativo são agrupados por pasta, contendo um arquivo com extensão *.html* que é responsável por estruturar a interface e possibilitar a interação com o usuário. O arquivo *.scss* é o encarregado de modificar o estilo da interface e proporcionar melhorias no *design*. Por fim o arquivo *.ts* controla a regra de validação das páginas e é por meio dele que é feita a comunicação com os *providers*.

Os arquivos *providers* são classes fornecidas pelo Ionic para gerenciar serviços que podem ser compartilhados em diversas páginas. Por meio destes arquivos é feita a comunicação com o servidor para consumir os dados.

Sendo assim, foi criado um arquivo *provider* para cada módulo, tornando a estrutura do projeto mais organizada, conforme mostra a Figura 9.

Figura 9 - Diretório dos arquivos do Ionic

```
/idoMobile/src/  
app  
assets  
pages  
  cliente  
    cadastrar-cliente  
      cadastrar-cliente.html  
      cadastrar-cliente.scss  
      cadastrar-cliente.ts  
    editar-cliente  
    buscar-cliente  
    perfil-cliente  
  profissional  
    cadastrar-profissional  
    editar-profissional  
    buscar-profissional  
    perfil-profissional  
  agendamento  
    agendar-servico  
    editar-agendamento  
    finalizar-agendamento  
  utils  
    login  
    menu  
providers  
  cliente.ts  
  profissional.ts  
  agendamento.ts  
theme  
index.html
```

Fonte: Autoria própria

4.2.5 Resultados

Com a finalização deste incremento foi possível compreender de que forma o sistema será utilizado, tornando assim o desenvolvimento do *back-end* mais rápido. Desta forma, foram levantadas quais informações seriam enviadas para alimentar as interfaces e de que forma o banco de dados deveria ser construído.

4.3 INCREMENTO 02 – *WEB SERVICE*

Com a conclusão do primeiro incremento foi possível determinar quais dados seriam fornecidos ao *front-end* da aplicação. Sendo assim, durante o segundo

incremento foi construído o servidor, responsável por fornecer essas informações para a interface por meio da arquitetura REST.

4.3.1 Funcionalidades do Incremento

Para completar as metas do segundo incremento foram realizadas as tarefas conforme o Quadro 5.

Quadro 5 – Funcionalidades do 2º. Incremento

2º Incremento	
Metas	Tarefas
Construção do servidor e banco de dados	Criação dos <i>controllers</i> e <i>models</i>
	Criação do banco de dados
Função de <i>login</i> no sistema <i>web</i> e <i>mobile</i>	Acesso do tipo cliente
	Acesso do tipo profissional
	Acesso do tipo administrador

Fonte: Autoria própria

4.3.2 Desenvolvimento

Nesse incremento foram construídas as funcionalidades básicas para o desenvolvimento do servidor.

Utilizando o *framework* Laravel, foi criada a estrutura do banco de dados MYSQL, por meio das *migrations* foi possível a criação e manipulação de todo o banco de dados, além de fornecer uma série de recursos para facilitar a sua construção e refatoração. Ainda usando o Laravel, foi possível criar todas as *models* necessárias para o funcionamento correto dos módulos do sistema.

Para que seja possível a comunicação da *Model* com a *View*, foram construídos os *Controllers* que atuam como intermediadores e responsáveis por toda a regra de negócio do sistema conforme o modelo MVC, como mostra a Figura 10.

São os *controllers* os responsáveis por toda ação realizada dentro do sistema como cadastros, edições e atualizações de dados.

Figura 10 - Diretório dos *Controllers* e *Models* do servidor

```
/idoWs/  
app  
  Console  
  Exceptions  
  Http  
    Controllers  
      Auth  
      AgendaController.php  
      ClienteController.php  
      Controller.php  
      ProfissaoController.php  
      ProfissionalController.php  
      ServicoController.php  
    Middleware  
      Kernel.php  
  Providers  
  Models  
    Agenda.php  
    Cliente.php  
    Profissao.php  
    Profissional.php  
    Profissional_Profissao.php  
    Profissional_Servico.php  
    Profissional_Agenda.php  
    Servico.php  
    Servico_Area.php  
    User.php  
bootstrap  
config  
...
```

Fonte: Autoria própria

A Figura 11 ilustra o diretório do arquivo *routes* que é responsável pelas rotas que se comunicam com o lado cliente. Cada rota é direcionada até uma funcionalidade específica, que é construída dentro de um dos *controllers* e tem o objetivo de executar uma tarefa solicitada pelo cliente.

Figura 11 - Diretório do arquivo de rotas do servidor



```
/idoWs/  
app  
bootstrap  
config  
database  
public  
resources  
routes  
  api.php  
  channels.php  
  console.php  
  web.php  
storage  
tests  
vendor  
...
```

Fonte: Autoria própria

4.3.3 Resultados

Com a finalização deste incremento foi possível realizar a comunicação entre os clientes e o servidor. A partir disto, foi necessário criar as funcionalidades para que o sistema pudesse ser utilizado pelos seus usuários.

4.4 INCREMENTO 03 – FUNCIONALIDADES DOS PROFISSIONAIS

Para a conclusão desse incremento foi necessário desenvolver as funcionalidades do módulo de profissionais para que fosse possível a divulgação e agendamento de serviços por meio da aplicação.

4.4.1 Funcionalidades do Incremento

Para completar as metas do incremento 3, foram realizadas as tarefas conforme o Quadro 6.

Quadro 6 – Funcionalidades do 3º. Incremento

3º Incremento	
Metas	Tarefas
Funções do modulo de profissionais	Cadastrar profissional
	Editar profissional
	Buscar profissional
	Perfil do profissional

Fonte: Autoria própria

4.4.2 Desenvolvimento

Durante a incremento 3 foram desenvolvidas todas as funcionalidades básicas referente ao módulo de profissionais.

A função de cadastro de um novo profissional exige diversas informações, dentre elas, os dados básicos do usuário, descrição da área em que atua, os serviços ofertados, local e horário de atendimento.

Assim como a funcionalidade de cadastro, a edição de um profissional possui o mesmo formulário, porém a função deve primeiramente consultar os dados referentes aquele profissional e preencher o formulário para que o usuário possa modificar conforme a necessidade.

Foi elaborado também a função de perfil do profissional, que lista todos os serviços ofertados com suas descrições, valores e duração. Além disso, foi criada a estrutura de exibição dos agendamentos pendentes que serão fornecidas nos próximos *incrementos*.

4.4.3 Resultados

A finalização desse incremento possibilitou o cadastro de perfis de profissionais dentro da aplicação, bem como a possibilidade de acrescentar e remover serviços ofertados por um profissional selecionado, por meio da edição de seu perfil.

A Figura 12 apresenta o cadastro de profissionais do cliente *web* onde é necessário acrescentar dados pessoais do profissional e da sua empresa.

Figura 12 - Cadastro de profissional

i-do

Seja bem-vindo a nossa comunidade

Dados Pessoais do Proprietário da Empresa

Nome do profissional: Seu nome completo
Sobrenome do profissional: Seu nome completo
Sexo: Selecione
Nascimento: dd/mm/aaaa

Estado: PR
Cidade: Ponta Grossa
Endereço:

Telefone: _____
Celular: _____
CPF: _____
RG: _____

Dados da Empresa

Razão Social: _____
CNPJ / CPF: _____

Estado: _____
Cidade: _____
Endereço: _____

Fonte: Autoria própria

A Figura 13 ilustra o cadastro de serviços de um profissional no cliente *web* onde é possível adicionar vários serviços e sua duração, valor e descrição. Além disso é possível alterar o horário de atendimento do profissional.

Figura 13 - Cadastro de serviços oferecidos pelo profissional

Fonte: Autoria própria

A Figura 14 mostra o perfil de um profissional cadastrado no sistema. Em que é possível visualizar informações de endereço, serviços disponibilizados e solicitar agendamento.

Figura 14 - Perfil do profissional

Fonte: Autoria própria

A Figura 15 apresenta o cadastro de um novo profissional a partir do cliente *mobile*.

Figura 15 - Cadastro de Profissional

🏠 Criar Conta

👤 DADOS PESSOAIS

Nome do Profissional

Sobrenome

RG

CPF

Endereço

🏠 ENDEREÇO

Estado ▼

Cidade ▼

Fonte: Autoria própria

4.5 INCREMENTO 04 – FUNCIONALIDADES DOS CLIENTES

O objetivo desse incremento foi o desenvolvimento de todas as funções do módulo cliente no servidor. Essas funções são responsáveis pela interação do usuário cliente dentro da solução.

4.5.1 Funcionalidades do Incremento

Para completar as metas do quarto incremento foram realizadas as tarefas conforme o Quadro 7.

Quadro 7 – Funcionalidades do 4º. Incremento

4º Incremento	
Metas	Tarefas
Funções do modulo de clientes	Cadastrar cliente
	Editar cliente
	Buscar profissionais e serviços
	Perfil do cliente

Fonte: Autoria própria

4.5.2 Desenvolvimento

A funcionalidade de cadastro de clientes é um formulário que solicita dados pessoais e o endereço do usuário cliente para a conclusão do cadastro.

Assim como a funcionalidade de cadastro, a edição de um cliente é semelhante ao cadastro, exceto na forma de exibição dos dados, pois ao editar um cliente, será enviado uma requisição para o servidor com o código do cliente que será editado. A partir disso o servidor responde com todos os dados deste cliente para permitir a manipulação.

Para o funcionamento da busca de profissionais foi necessário validar os filtros de nomes, cidade, e serviços para retornar os resultados que se encaixem com a requisição.

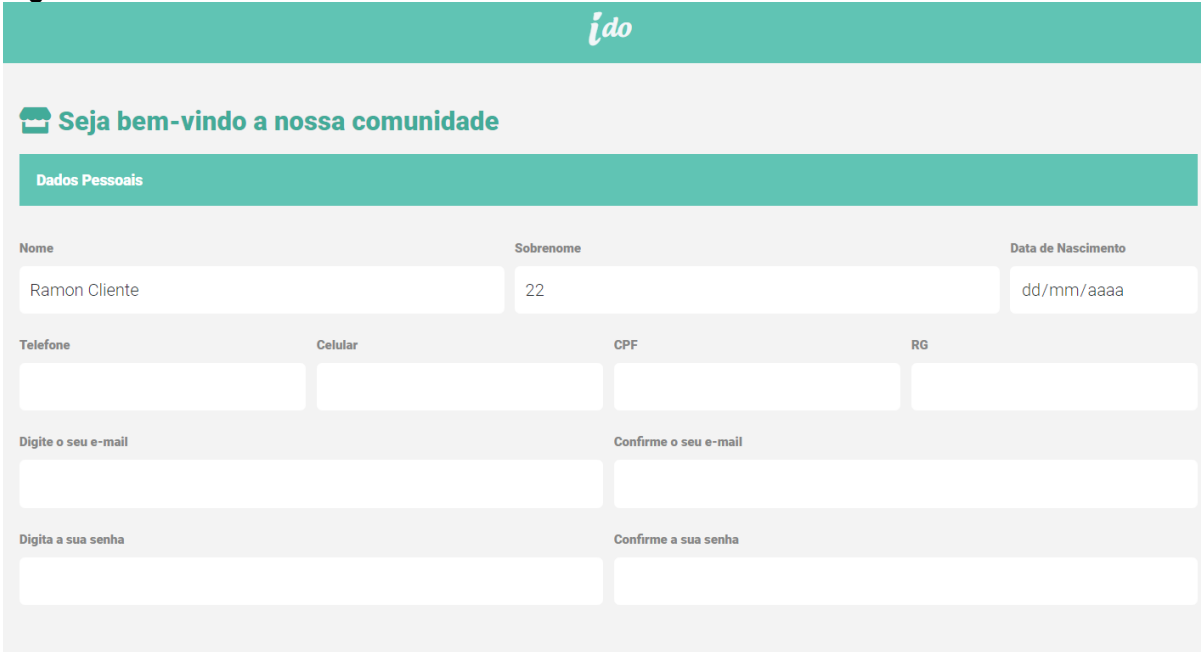
No desenvolvimento do perfil do usuário, além de buscar os dados pessoais do cliente, será retornado também os próximos agendamentos pendentes pelo profissional, ordenados por data. Além disso, será retornado um histórico de todos os profissionais que esse cliente já solicitou serviços, para facilitar a busca por parte do usuário cliente.

4.5.3 Resultados

Nesse incremento foram finalizadas as funções do módulo de cliente dentro do servidor. Sendo assim, será possível que um usuário cliente se cadastre no sistema, edite e visualize seu perfil. A funcionalidade responsável pela busca de profissionais também foi concluída.

A Figura 16 apresenta o formulário de cadastro de um novo cliente a partir do cliente *web*.

Figura 16 - Cadastro de cliente



ido

Seja bem-vindo a nossa comunidade

Dados Pessoais

Nome: Ramon Cliente Sobrenome: 22 Data de Nascimento: dd/mm/aaaa

Telefone: Celular: CPF: RG:

Digite o seu e-mail: Confirme o seu e-mail:

Digita a sua senha: Confirme a sua senha:

Fonte: Autoria própria

A Figura 17 ilustra a interface *web* do perfil de um cliente cadastrado no sistema.

Figura 17 - Perfil do cliente



ido

Carlos da Silva
✎ Editar Perfil

PROFISSIONAIS AGENDAMENTOS

Atendimentos Pendentes

Você não possui nenhum atendimento com este profissional.

Fonte: Autoria própria

A Figura 18 apresenta a página de pesquisa por profissionais através da interface *web*.

Figura 18 - Pesquisa por profissional e serviço

Fonte: Autoria própria

4.6 INCREMENTO 05 – FUNCIONALIDADES DE AGENDAMENTO

Nesse incremento foi desenvolvido o módulo de agendamento, o qual é responsável pela regra de negócio do sistema, possibilitando um cliente buscar profissionais e realizar o agendamento de seus serviços ofertados em seu perfil.

4.6.1 Funcionalidades do Incremento

Para completar as metas do quinto incremento foram realizadas as tarefas conforme o Quadro 8.

Quadro 8 – Funcionalidades do 5º. Incremento

5º Incremento	
Metas	Tarefas
Funções de agendamento	Realizar um novo agendamento
	Visualizar agendamento
	Cancelar agendamento
	Concluir agendamento

Fonte: Autoria própria

4.6.2 Desenvolvimento

Por meio da funcionalidade de agendamento, ao visualizar o perfil do profissional, o cliente terá a opção de selecionar um dos serviços disponíveis e, em seguida, selecionar a data e horário do agendamento.

Assim que um cliente solicitar o agendamento de um serviço, este é enviado para o profissional responsável para que confirme o agendamento ou cancele até um dia antes da data agendada. Caso o agendamento seja cancelado por um dos envolvidos, uma notificação é enviada ao outro para que seja informado do cancelamento.

Este atendimento também permanecerá no perfil do cliente para que ele possa visualizar os agendamentos pendentes e optar pelo cancelamento caso seja necessário.

4.6.3 Resultados

Ao final deste incremento foi possível a busca e divulgação de serviços realizados por profissionais que possuem a necessidade de um agendamento prévio. O gerenciamento dos atendimentos poderá ser visualizado no perfil dos usuários possibilitando a manipulação dos mesmos.

A Figura 19 exibe a escolha de um serviço durante o cadastro de um novo agendamento no cliente *web*.

Figura 19 - Agendamento de serviço

João Aparecido
Tiradentes 285 - Centro | Ponta Grossa, PR

Barbeiro Massagista

SOBRE SERVIÇOS AGENDAR

Agendar Serviço

Selecione um serviço

Como Serviço

Barbeiro Corte Simples

Corte Simples
Duração: 15min
Valor: R\$ 20
Descrição: Corte de cabelo masculino simples

Fonte: Autoria própria

A Figura 20 ilustra a seleção da data e horário do cadastro de um novo agendamento no sistema *web*.

Figura 20 - Agendamento de serviço

Corte Simples
Duração: 15min
Valor: R\$ 20
Descrição: Corte de cabelo masculino simples

Selecione uma data

Data do atendimento

11/04/2019

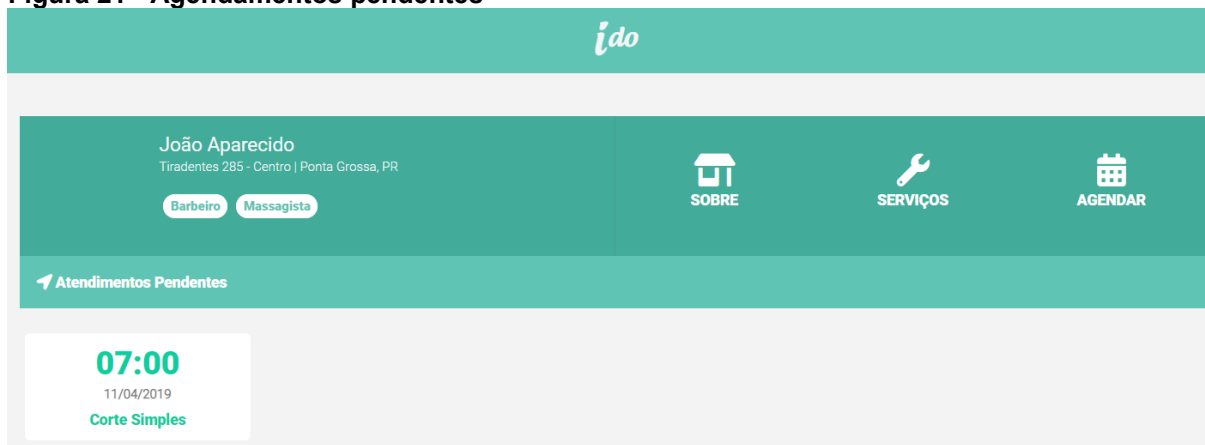
Selecione um horário

07:00 07:30 08:00 09:00

Fonte: Autoria própria

A Figura 21 exibe os agendamentos pendentes no cliente *web*.

Figura 21 - Agendamentos pendentes



Fonte: Autoria própria

A Figura 22 apresenta os detalhes de um agendamento e a opção de cancelar o agendamento no cliente web.

Figura 22 - Detalhes do agendamento web



Fonte: Autoria própria

A Figura 23 ilustra os agendamentos pendentes no cliente *mobile*.



Fonte: Autoria própria

5 CONCLUSÃO

Este trabalho propôs o desenvolvimento de uma aplicação *web* e *mobile* com o objetivo de melhorar o atendimento e a procura por profissionais que necessitam de um agendamento prévio para realização de seus serviços, tornando o processo mais simples e ágil.

A partir disso, foi desenvolvido um sistema capaz de pesquisar profissionais por nome, serviços disponíveis e cidade onde atua, possibilitando a comparação de valores, detalhes dos serviços e disponibilidades de horários. Facilitando assim a procura por parte do cliente que deseja realizar um agendamento.

A utilização de *frameworks* contribuiu de forma positiva para o desenvolvimento do sistema. Apesar da necessidade de estudar o funcionamento dos *frameworks*, a curva de aprendizagem foi baixa. Além disso, foi possível desenvolver uma aplicação de forma produtiva, por meio de bibliotecas e funcionalidades disponibilizadas pelos *frameworks*, o que facilitou o desenvolvimento e possibilitou a conclusão do projeto em um menor intervalo de tempo.

Ainda que o sistema não tenha sido testado com um usuário final para confirmar seu desempenho, todos os objetivos estabelecidos foram alcançados. Porém, para que a solução possa ser comercializada e utilizada por profissionais, é necessário uma atenção especial nas funcionalidades de autenticação e segurança aos quais não foram concluídos devido ao curto tempo disponível.

5.1 TRABALHOS FUTUROS

Este projeto permite a criação de novas funcionalidades que em função do tempo não foram implementadas. Como por exemplo, a disponibilidade de realizar pagamentos online entre o cliente e o profissional por meio da aplicação. Utilizando de APIs que fornecem este tipo de integração entre serviços de pagamentos digitais.

As tecnologias utilizadas para o desenvolvimento móvel foram limitadas apenas ao desenvolvimento da plataforma Android, já que o IOS exige uma licença para seu desenvolvimento e um computador com sistema operacional Mac para que seu SDK seja executado. Sendo assim, apresenta-se interessante a expansão deste projeto para o sistema operacional IOS.

REFERÊNCIAS

ALCANTARA, C. A. A.; VIEIRA, A. L. N. **Tecnologia móvel: uma tendência, uma realidade**. 2011. Disponível em:

<<http://arxiv.org/ftp/arxiv/papers/1105/1105.3715.pdf>>. Acesso em: 09 jun. 2018.

ALVIM, P. **Tirando o Máximo do Java EE 6 Open Source com jCompany Developer Suite**. 3. ed. Belo Horizonte: Pearson, Powerlogic Publishing, 2010.

ANGULAR. **What is angular?** Disponível em: <<https://angular.io/docs>>. Acesso em: 09 jun. 2018.

BORGES, A; JOIA, L. **Executivos e smartphones: uma relação ambígua e paradoxal**. Organizações & Sociedade, Salvador(BH), v. 20, n. 67, p. 585-602, out. 2013.

BRASSCOM. **Mapa de Conectividade**. Set. 2013. Disponível em

<<http://www.brasscom.org.br/brasscom/Portugues/detInstitucional.php?codArea=3&codCategoria=48>>. Acesso em: 08 abr. 2018.

COSTA, C. J. **Desenvolvimento para web**. Lisboa: Lusocrédito, 2007.

COULOURIS, G; DOLLIMORE, J; KINDBERG, T; BLAIR, G. **Sistemas Distribuídos: Conceitos e Projeto**. 5. ed. Porto Alegre: Bookman Editora, 2013

DE MENDONÇA, Vinicius Rafael Lobo; BITTAR, Thiago Jabur; DE SOUZA DIAS, Márcio. **Um estudo dos sistemas operacionais Android e iOS para o desenvolvimento de aplicativos**.

DESHPANDE, Y; MURUGESAN, S; GINIGE, A; HANSEN, S; SCHWABE, D; GAEDKE, M; WHITE, B. **Web Engineering. Journal of Web Engineering**. v. 1, n. 1, p. 3-17, mai. 2002.

FRANCO, P. A. A. **SERVIÇOS DE INFORMAÇÃO PARA DISPOSITIVOS MÓVEIS: usos e possibilidades**. ENGI/2014 – Encontro nacional de gestão, políticas e tecnologias de informação.2014, Goiânia: UFG, 2014. p7.

GABARDO, A. C. **Laravel para ninjas**. 1. ed. São Paulo: Novatec, 2017.

GONÇALVES, A. J. R. **Desenvolvimento de Aplicativos Híbridos com o Ionic Framework**. Picos (PI), v. 1, n. 1, p. 500-515, jun. 2017.

HAYDEN, T; WEBSTER, T. **The Mobile Commerce Revolution: Business success in a wireless world**. 1. ed. Indianapolis: Que Publishing, 2014.

KASPERBAUER, M et al. **Chronos Mobi: uma aplicação móvel multiplataforma para o gerenciamento de projetos**. Revista Brasileira de Computação Aplicada, Passo Fundo (RS), v. 5, n. 1, p. 84-97, abr. 2013.

KOHN, K; MORAES, Cláudia H. **O impacto das novas tecnologias na sociedade: conceitos e características da Sociedade da Informação e da Sociedade Digital**. In: XXX Congresso Brasileiro de Ciências da Comunicação. 2007.

MILETTO, E.M; CASTRO BERTAGNOLLI, S. **Desenvolvimento de Software II: Introdução ao Desenvolvimento Web com HTML, CSS, JavaScript e PHP**. 1. ed. Porto Alegre: Bookman Editora, 2014

MINETTO, E. L. **Frameworks para Desenvolvimento em PHP**. 1. Ed. São Paulo: Novatec, 2007.

NIEDERAUER, J. **Desenvolvendo websites com PHP**. 3. ed. São Paulo: Novatec, 2017.

PRESSMAN, R. S. **Engenharia de Software: Uma Abordagem Profissional**. 7. ed. Porto Alegre: Amgh, 2011

PREZOTTO, E D; BONIATI, B B. **Estudo de *frameworks* multiplataforma para desenvolvimento de aplicações *mobile* híbridas.** 2017.

ROYCE, W. **Managing the development of large software systems: Concepts and techniques.** In: Proc. IEEE WESCOM. IEEE Computer Society Press, Los Alamitos. 1970.

SOMMERVILLE, I. **Engenharia De Software.** 9. ed. São Paulo: Pearson, 2011.

VERMA, Archit. **MVC Architecture: A Comparative Study Between Ruby on Rails and Laravel.** Indian Journal of Computer Science and Engineering (IJCSE), v. 5, n. 5, p. 196-198, 2014.

YOSHIURA, V. T. **Desenvolvimento e Implantação de um Sistema Web para Monitoramento da Rede de Atenção em Saúde Mental.** 2015. 202 f. Dissertação (Mestrado) - Programa de Pós-Graduação Interunidades Bioengenharia - Escola de Engenharia de São Carlos / Faculdade de Medicina de Ribeirão Preto / Instituto de Química de São Carlos da Universidade de São Paulo. São Carlos, 2015.