

AGENDA+**DESENVOLVIMENTO DE UM APLICATIVO MÓVEL PARA AGENDAR SERVIÇOS ONLINE DE FORMA INTELIGENTE**

Jean Carlo Alves Ferreira
Graduando em Sistemas de Informação – Uni-FACEF
jean.carlo.ferreira@outlook.com

Marcelo de Andrade Silva
Graduando em Sistemas de Informação – Uni-FACEF
marcelosilva473@hotmail.com

Leandro Borges
Mestre em Desenvolvimento Regional – Uni-FACEF
leandro.borges@facef.br

Resumo

O projeto busca desenvolver um sistema que facilite o dia a dia das pessoas que não possuem tempo para realizar determinadas tarefas como agendamentos de serviços. Assim como problemas com tempo de atendimento ao cliente na vida de profissionais, sem um meio inteligente de realizar agendamentos sem que precise de uma mão de obra humana. Desta maneira, surgiu como objetivo, desenvolver um sistema a fim de que, em um único lugar, ter acessos a diversos estabelecimentos que possuem todos os serviços prestados e através deles ter acesso a agenda do profissional podendo realizar seu agendamento independente de horário de funcionamento do estabelecimento, escolhendo seu melhor horário para sua rotina com informações de tempo e valor. E também ao lado dos estabelecimentos tendo uma agenda virtual com encaixe automático de serviços. O aplicativo foi desenvolvido para os sistemas operacionais Android e IOS utilizando a Flutter SDK, atendendo os principais usuários de dispositivos móveis hoje presente na sociedade, e um sistema web para controle interno dos profissionais da organização. A conclusão do projeto se mostrou positiva no que tange aos principais objetivos propostos neste trabalho. Desta forma podemos concluir que com o uso do sistema, facilitou a rotina diária dos profissionais e de seus clientes, proporcionando simplicidade ao realizar um agendamento.

Palavras-chave: agendamento. aplicativo. profissionais. serviços.

Abstract

The project seeks to facilitate the daily lives of people who do not have the time to perform certain tasks such as scheduling services. As well as problems with customer service time in the life of professionals, without a clever way of scheduling without the need for human labor. Thus, the objective was to develop a system in order to have in one place access to several establishments that have all the services provided and through them have access to the professional's schedule and can make their independent scheduling of opening hours of the establishment, choosing your best time for your routine with time and value information. And also next to the establishments having a virtual agenda with automatic docking services. The application was developed for the Android and IOS operating systems using the Flutter SDK, serving the main users of mobile devices present in society, and a web system

for internal control of the organization's professionals. The conclusion of the project was positive regarding the main objectives proposed in this work. Thus we can conclude that with the use of the system, it facilitated the daily routine of professionals and their customers, causing simplicity when making a schedule.

Keywords: *scheduling. app. professionals. services.*

1 Introdução

O cotidiano das pessoas tem se tornado cada vez mais sobrecarregado e o tempo cada vez mais escasso para realizar tantas atividades (NOGUEIRA, 2016), desta forma é indiscutível a importância da tecnologia na vida das pessoas e principalmente na vida dos profissionais, que além de trazer novos meios de trabalhar, tem como seu maior objetivo a automatização de tarefas, trazendo a eles mais produtividade (LEUCOTRON, 2019).

A utilização de *smartphones* na busca dos estabelecimentos tem se tornado gradativamente comum, principalmente pelo acesso à Internet e busca de informações. Tecnologias para desenvolvimento de aplicações *mobile* também tem se popularizado entre os desenvolvedores, principalmente para o ambiente Android, considerando a grande popularização dessa plataforma entre os usuários.

Segundo o IBGE (2018), 138 milhões de brasileiros possuem um *smartphone*, utilizando diversos aplicativos em seu dia a dia como meio de comunicação e compras realizadas na internet em 2016.

Com base nessas questões que se passam com os clientes de serviços que necessitam de agendamento prévio, como por exemplo, salões de beleza, o artigo tem como objetivo propor a criação de um aplicativo de forma simples e rápida, o agendamento e controle de horários. A proposta é também facilitar a vida dos profissionais, aperfeiçoando de forma inteligente os horários personalizados de seus clientes.

A pesquisa tem como metodologia a revisão de literatura fundamentada em autores que tratam sobre tema, assim como em bases de dados do Google acadêmico, *Scielo* e outras fontes relevantes. Também será utilizada a metodologia exploratória de forma que o problema de pesquisa será levado a uma solução tomando como base o desenvolvimento e a análise de prototipação de um mínimo produto viável (MVP).

Em entrevista realizada e disponível em Github (2019) com uma empreendedora e administradora de um salão de beleza, foi constatado que os profissionais perdem muito tempo atendendo telefones e respondendo mensagens de clientes a fim de realizar agendamentos de serviços prestados pela empresa, apresentando conflitos e encaixe de horários, prejudicando o profissional e atrasando seus serviços. Nesse sentido, os próximos clientes seriam penalizados com esses atrasos. Desse modo, é comum que sejam causados constrangimentos aos clientes e ocasione desistências para com o estabelecimento.

Em um levantamento feito com os clientes da empreendedora sobre o aplicativo, foi identificada a falta de tempo para entrar em contato com o estabelecimento a fim de agendar algum de seus serviços. Com a disponibilização da

agenda dos profissionais para qualquer usuário, os mesmos poderão realizar agendamentos a qualquer horário que lhe for mais cômodo, mesmo que o estabelecimento já tenha encerrado seus serviços naquele dia.

Assim, é possível ajudar os empreendimentos que realizam agendamento a otimizar a agenda de seus profissionais e ainda fidelizar os clientes que aproveitam a praticidade do agendamento via aplicativo.

No capítulo 2, apresentamos o Referencial Teórico, em que revelará as tecnologias utilizadas no desenvolvimento do aplicativo, assim como tecnologias de desenvolvimento para Android e IOS que são os dois sistemas operacionais mais utilizados atualmente em nossa sociedade. Tratará também tecnologias na qual busca dinformações em banco de dados.

Em seguida, o capítulo 3 mostra os processos que constituem o projeto, como toda a documentação exigida na Engenharia de Software como levantamento de requisitos, funcionais e não funcionais, regras de negócio, diagrama de caso de uso e BPMN (*Business Process Modeling Notation*).

Apresentaremos os conceitos de empreendedorismo que são relacionados ao projeto no 4º capítulo, com a apresentação do modelo canvas, significado de startups junto com os conceitos de startup enxuta.

Na sequência, o capítulo 5 aborda os resultados do desenvolvimento do sistema web e aplicativo do Agenda+ (Agenda Plus) com figuras das interfaces além de relatar o funcionamento.

Por fim, o capítulo 6, apresenta as considerações finais, contendo o objetivo atingido e possíveis melhorias e ideias de como aperfeiçoar a experiência do usuário.

2 Referencial Teórico

Este capítulo apresenta todos os conceitos e definições dos fundamentos ligados ao projeto, com o intuito de orientar o leitor sobre o assunto.

2.1 Desenvolvimento de Aplicativos Móveis

Os dispositivos móveis oferecem conectividade e poder de uso a qualquer lugar e em qualquer momento, tornando-se importantes tanto para uso pessoal, quanto profissional.

Com o uso demasiado de dispositivos móveis, o número de plataformas e ambientes de desenvolvimento aumentou proporcionalmente. Para Aquino (2007), a escolha de uma plataforma ideal para o desenvolvimento de um projeto significa optar por uma solução que forneça os melhores benefícios em termos de custos, eficiência e tempo de desenvolvimento esperados para finalização do projeto.

2.1.1 Android

Segundo Pereira e Da Silva (2009), Android é uma plataforma para tecnologia móvel completa, envolvendo um pacote com programas para celulares, já com um sistema operacional, aplicativos, *middleware* e interface de usuários. Foi desenvolvida com base no sistema operacional Linux. Por ser *open source*, pode ser sempre adaptada a fim de incorporar novas tecnologias. A plataforma estará sempre

em evolução, já que a comunidade de desenvolvedores estará trabalhando em conjunto para construir aplicações móveis inovadoras.

O Android é uma tecnologia que foi anunciada em 2007 pela Google sendo de código aberto que consiste em vários componentes permitindo que desenvolvedores e fabricantes de celulares trabalhem independentemente, na qual pode ser dividido em 5 pontos principais: aplicativos, estruturas de aplicativos, bibliotecas nativas, tempo de execução do Android e o *kernel* do Linux (KRAJCI e CUMMINGS, 2013).

Segundo Krajci e Cummings (2013) a primeira versão do Android chamada de Astro foi lançada em 2008 no celular HTC Dream trazendo recursos como Android Market uma loja digital de aplicativos que posteriormente foi substituído pela Google Play sendo semelhante com a App Store do IOS.

O IDE oficial do Android é o Android Studio no qual foi baseado no *IntelliJ IDEA* criado especificamente para acelerar o desenvolvimento e ajudar na criação de aplicações de alta qualidade para qualquer dispositivo Android (ANDROID, 2019).

Segundo o Android (2019) ele é o sistema operacional móvel mais popular do mundo sendo encontrado em dispositivos de celulares, relógios, tablets e veículos automotivos e se encontra na versão *Pie* que traz um conjunto de aperfeiçoamentos que proporciona melhorias na experiência do usuário.

2.1.2 IOS

IOS é a abreviatura para *iPhone Operation System*, desenvolvido para *Apple*. Baseado no sistema operacional MAC OS X e projetado para atender somente aparelhos móveis desenvolvidos pela *Apple* (APPLE DEVELOPER, 2019).

O *IOS* é um sistema operacional para *iPhone*, *iPod*, *iPad* e a sua primeira aparição foi com o lançamento da primeira geração de *iPhone* em 2008 revolucionando a maneira como os telefones celulares seriam criados a partir deste momento (MORRISSEY e CAMPBELL, 2011).

Uma grande atualização para o *iOS* foi a inclusão do App Store que é um serviço de distribuição digital de aplicativos que permite os usuários navegar e baixar aplicativos para seus dispositivos e com o lançamento do SDK (*Software Development Kit*) para ajudar desenvolvedores a criar seus aplicativos para ser disponibilizado na App Store foi um grande benefício para a *Apple* que excedeu 2 milhões de vendas e downloads de aplicativos (MORRISSEY e CAMPBELL, 2011).

Segundo *Apple* (2019) o *IOS* está na versão 12, foi desenvolvido para que a experiência do usuário seja ainda mais rápida, responsiva e agradável com o *iPhone* e o *iPad* com avanços e recursos como 70% mais rápido para acessar a câmera, 50% mais rápido para exibir o teclado. *FaceTime*, aplicativo capaz de realizar chamadas de vídeo e chamadas de áudio com até 32 amigos ao mesmo tempo, *ARKit 2* que permite que os desenvolvedores de realidade aumentada criem experiências ainda mais envolventes.

2.2 Node JS

Node.js é um interpretador da linguagem *JavaScript* para o ambiente de servidor, é baseado na *engine* "V8" desenvolvida pela Google e gravado na linguagem C e C ++, enquanto está *engine* é usado para *JavaScript* no navegador, o *Node* tem

como meta o suporte de processos de servidor com o foco no desempenho e o menor uso de memória possível (TILKOV e VINOSKI, 2010).

Segundo Node Js (2019), a tecnologia foi idealizada para ser utilizada em aplicações de rede escalonáveis, pois ela não é baseada em rede de *threads* e sim em um modelo entrada/saída de dados sem bloqueio, orientado a eventos assíncrono que tem suporte para retornos de chamada o que torna leve, eficiente e essencial para aplicações que tem muitos acessos simultaneamente.

O Node.js suporta vários processos, mas sendo executados em núcleos divergentes além de ser um ambiente *single thread* que em segundo plano executa *multiple threads* em código assíncrono (TILKOV e VINOSKI, 2010).

A partir da versão 0.6.0 o Node recebeu como gerenciador *default* de pacotes o NPM (*Node Package Manager*). Para HERRON (2013) o que faz o Node ter um crescimento muito rápido é suportar ferramentas de terceiros. No centro disso está o NPM que traz recursos que é estruturado em seu arquivo "*package.json*" que contém os principais atributos para descrever um módulo de como será sua aplicação em ferramentas.

2.2.1 GraphQL

De acordo com GraphQL (2019), o GraphQL é uma linguagem de consulta (*query language*) para APIs (*Application Programming Interface*) que retorne dados de acordo com a necessidade do cliente, assim diminuindo a quantidade de tráfego pela rede e aumentando o poder do desenvolvedor sobre as consultas.

GraphQL é gerenciado pelo Facebook, mas é de código aberto, sendo uma ferramenta de tempo de execução. É uma camada de execução que a aplicação de servidor pode usar para representar e responder a quaisquer solicitações feita com a linguagem GraphQL, além de funcionar bem com outras linguagens de *back-end* isso mostra que podemos implementar o GraphQL como uma camada sobre qualquer lógica de servidor existe (BUNA, 2016).

Um diferencial do GraphQL para REST (*Representational State Transfer*) que ele pode ser apenas um *endpoint* que manipula todas as solicitações do cliente e pode dar o poder de personalizar essas solicitações, dessa forma, os clientes podem controlar os dados que buscam e podem facilmente evitar os problemas de buscas excessiva e insuficiente (BUNA, 2016).

2.3 Flutter

Conforme Flutter (2019), Flutter é um SDK (*Software Development Kit*) desenvolvido pela a Google para a criação de aplicações portáteis e nativas para os sistemas operacionais IOS, ANDROID e FUCHSIA utilizando a linguagem Dart e seus próprios objetos de interface.

Como o Flutter fornece o seu próprio conjunto de widgets, entrada/saída de arquivos e rede, os projetos são escritos na linguagem Dart e faz compilação nativa para a plataforma desejada fazendo com que entregue um alto nível de personalização para os desenvolvedores que é incomparável por outros SDK (DAGNE, 2019).

Flutter processa todos os componentes de visualização usando seu próprio mecanismo de renderização, além de suportar *hot-reload* durante o

desenvolvimento que é essencialmente atribuído pela injeção do código fonte atualizado na máquina virtual do Dart (WU, 2018).

A linguagem Dart foi escolhida para o Flutter por possuir uma compilação VM (Virtual Machine) com JIT (*just-in-time*) e um compilador AOT (*ahead-of-time*) para produzir código de máquina perfeita para o recurso *hot-reload*, além de ser mais rápido que o *JavaScript*. Assim o Dart permitiu criação de interface dinamicamente no código e como resultado a personalização de interface é mais fácil no Flutter (DAGNE, 2019).

2.4 React

Segundo React (2019), o React é uma biblioteca *JavaScript* que contém recursos para criação de soluções de interfaces de usuários com o propósito de ser interativa e de fácil criação utilizando componentes encapsulados que gerenciam seu próprio estado.

O React foi idealizado pelos engenheiros do Facebook para ser utilizado em uma abordagem de *Model-View-Controller* ao lado do cliente para resolver os desafios enfrentados ao criar soluções de interfaces de usuários com uma coleção de dados que mudam dinamicamente com o uso de novas estruturas *JavaScript*, introduzindo paradigmas e mudanças de status que é fundamental para criação de aplicativos e interfaces escaláveis e de fácil manutenção (GACKENHEIMER, 2015).

O React propõe resolver um único problema no *design* da interface e do usuário no desenvolvimento do *front-end* no qual é a criação de interfaces em grande escala com dados que mudam o tempo utilizando um mecanismo para substituir o estado apenas quando os dados mudavam (GACKENHEIMER, 2015).

De acordo com React (2019), graças a utilização de diversas técnicas capazes de minimizar a quantidade de operações difíceis de DOM que são necessárias para modificar a UI (Interface de Usuário) a aplicação ganha performance diferente de aplicação que utiliza página única.

2.5 Banco de Dados

Um banco de dados é uma coleção de dados relacionados que em alguns momentos representa aspectos do mundo real e pode ser de qualquer tamanho e de complexidade variável, além que pode ser gerado e mantido manualmente ou pode ser automatizado por um grupo de aplicativos feitos especialmente para esta tarefa (ELMASRI e NAVATHE, 2005).

Banco de dados pode ser considerado um sistema computadorizado de manutenção de registros ou um armário de arquivamento para uma coleção de arquivos de dados onde o usuário pode realizar operações de *INSERT* para salvar, *UPDATE* para modificar, *DELETE* para excluir, *SELECT* para consultar dados (DATE, 2004).

2.5.1 Banco de Dados Relacional

Um banco de dados relacional é baseado no modelo relacional que usa um conjunto de tabelas com colunas para representar os dados e as relações entre eles (SILBERSCHATZ, SUNDARSHAN e KORTH, 2016), ainda segundo Silberschatz, Sundarshan e Korth (2016) um modelo relacional é baseado em registro que cada um define um número fixo de campos, ou atributos onde o formato é fixo de vários tipos.

2.5.2 MySQL

O MySQL é um servidor relacional de licença dupla (sendo uma delas de *software* livre) e de código aberto, inicialmente foi projetado para aplicações pequenas ou de médio porte, entretanto isso mudou e atende inclusive de grande porte (MILANI, 2007).

O MySQL é um banco de dados relacional que armazena os dados em uma tabela separa e ainda existe a possibilidade de escolher o tipo em cada situação, assim atribui velocidade e flexibilidade e por estar sempre em constante desenvolvimento, o servidor MySQL oferece uma grande variedade de conjunto de funções como conectividade, velocidade e segurança que o torna um servidor altamente adequado para desenvolvimento web (WIDENIUS, AXMARK e ARNO, 2002).

As principais características do MySQL é ser um SGBD (Sistema Gerenciador de Banco de Dados), fornece sua API para linguagem como Java, Python, PHP, Perl, C, C++, entre outras, além de ser *multithreads* e ter várias formas de armazenamento (MILANI, 2007).

3 Engenharia de Software

Neste item iremos relatar a documentação de engenharia de *software* do Agenda+ (Agenda Plus), para visualizar os documentos completos que serão mencionados nesta divisão acessar em GITHUB (2019).

Como menciona Pressman e Maxim (2016) a engenharia de *software* é uma área voltada á especificação em camadas que engloba processos que é a liga que mantém as camadas para que o projeto seja entregue da melhor forma possível, utilizando métodos para o fornecimento de técnicas para a execução e também ferramentas que entrega para os métodos e processos, suporte automatizado ou semiautomatizado em uma abordagem que possa ser aplicada por uma equipe de desenvolvimento.

Segundo Rezende (2006) engenharia de *software* é a metodologia de desenvolvimento e manutenção de sistemas divididos em partes distintas que possuem diversas características sendo algumas como efetivação de padrões de qualidade, produtividade e efetividade.

3.1 Requisitos

3.1.1 Requisitos Funcionais

Requisitos funcionais são as funcionalidades que a aplicação deve realizar conforme a interação do usuário como entradas de dados, comportamentos em determinadas situações e o que ele não deverá fazer.

Na figura 1, apresentamos os principais requisitos funcionais do sistema Agenda+ (Agenda Plus).

Figura 1 – Requisitos funcionais do projeto Agenda+

| RF | DESCRIÇÃO | CATEGORIA/ PRIORIDADE | INFORMAÇÕES | REGRAS |
|---------------------------|--|--|---|---|
| Cadastrar Cliente | O sistema deve cadastrar novos clientes | Categoria: Evidente Prioridade: Altíssima | Nome do cliente, data de nascimento, cpf, e-mail, contato, cep, rua, número, bairro, cidade, estado, serviços | O sistema não poderá deixar a inclusão de cliente com o mesmo CPF, além de ter a possibilidade de alterar e desativar o cliente. |
| Agendar Horário | O sistema deve realizar agendamentos no aplicativo móvel | Categoria: Evidente Prioridade: Altíssima | Estabelecimento, profissional, serviço, data, hora, cliente (inclusão automática pelo sistema) | Caso o usuário escolha um dia que não haverá expediente no estabelecimento o sistema retornará uma mensagem de aviso, mas se houver expediente e escolher um determinado serviço que o tempo de execução seja maior que o disponível não aparecerá caso contrário ele poderá fazer o agendamento. |
| Cadastrar Serviços | O sistema deve realizar cadastro de serviços | Categoria: Evidente Prioridade: Altíssima | Descrição, valor padrão e tempo padrão | O sistema deverá alertar caso tenha um serviço com o mesmo nome ou parecido com o informado pelo usuário. |
| Mostrar Perfil | O sistema deve mostrar a tela de perfil. | Categoria: Evidente Prioridade: Altíssima | Mostrar informações com o nome, cpf, data de nascimento, sexo, e-mail, celular, cidade | O sistema deverá mostrar a tela de perfil caso o usuário esteja logado. |
| Visualizar Agenda | O sistema deve mostrar os horários já agendados no painel administrativo | Categoria: Evidente Prioridade: Altíssima | Mostrar informações para o usuário final e dashboard do cliente | A visualização da agenda deverá conter filtros para melhor experiência para o cliente e mostrar informações com data, horário, nome do cliente do estabelecimento e valor. Para o usuário final deverá aparecer o nome do estabelecimento, horário e data e valor. |

Fonte: Os autores

3.1.2 Requisitos Não Funcionais

Requisitos não funcionais são restrições as funcionalidades oferecidas pela aplicação, características e limites no processo de desenvolvimento.

Na figura 2, apresentamos os principais requisitos não funcionais do sistema Agenda+ (Agenda Plus).

Figura 2 – Requisitos não funcionais do projeto Agenda+

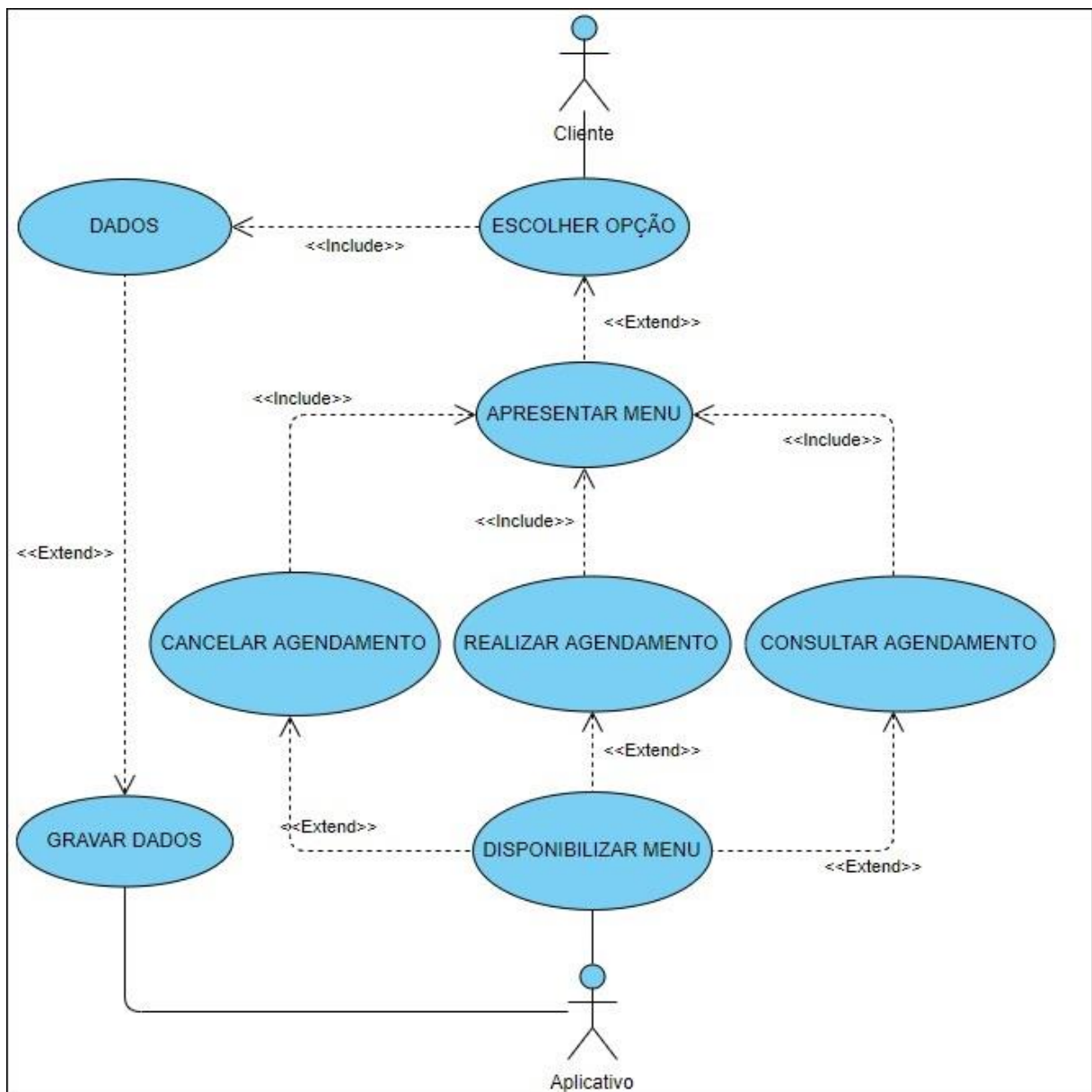
| | RNF | CATEGORIA | OBRIGATORIEDADE | PERMANÊNCIA | RESTRIÇÃO |
|----------------------------------|-----------------------------|------------------|------------------------|---|------------------|
| Sistema Operacional Móvel | Portabilidade | Obrigatório | Permanente | O software deve ser desenvolvido com tecnologia que possa ser utilizado tanto para ANDROID e IOS o mesmo código | |
| Sistema Modularizado | Manutibilidade / Desempenho | Obrigatório | Permanente | O software deve proporcionar módulos para ter um reparo fácil. | |
| Uso de Componentes | Reusabilidade | Obrigatório | Permanente | O software deve ser criado com objetos, funções e componente que possa ser reutilizado em diversas partes do sistema. | |
| Conexão com Componentes | Segurança | Obrigatório | Permanente | O sistema deve ser criado apenas para usuário com internet para evitar erros no agendamento. | |
| Tempo Limite de Resposta | Segurança | Obrigatório | Permanente | Inserir tempo limite de resposta da requisição/resposta da API para redução de erros no servidor (20 segundos). | |

Fonte: Os autores

3.3 Diagrama de Caso de Uso

Este diagrama é o responsável por documentar o sistema desenvolvido pelos olhos do usuário sem aprofundar em termos especificamente técnico, na figura 3 representamos especificamente a parte de agendamento do sistema Agenda+.

Figura 3 – Caso de uso do agendamento do aplicativo Agenda+



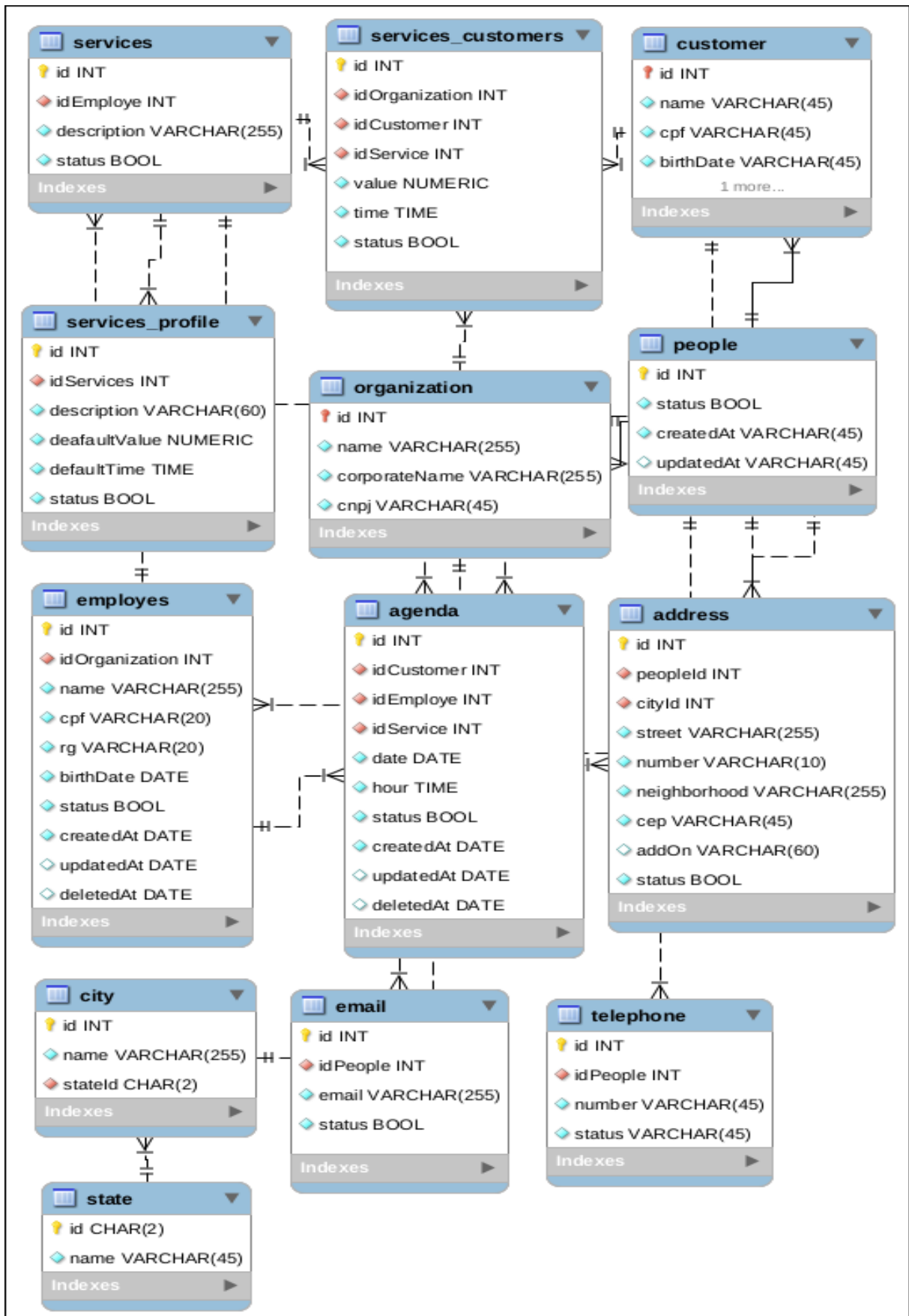
Fonte: Os autores

3.4 Modelo Entidade Relacionamento

Para Joel (2014) o Modelo Entidade Relacionamento, é um conceito utilizado na engenharia de *software* que representa de forma abstrata aos objetos que terá o banco de dados da aplicação.

Na Figura 4 mostramos o modelo entidade relacionamento que foi utilizado para a criação do banco de dados da aplicação do sistema do Agenda+ (Agenda Plus), as entidade foram nomeadas com nomes concretos e abstratos utilizando entidades associativas, entidades fracas e fortes, além de inserir relacionamentos “um para um”, “um para muitos”, “muitos para muitos”.

Figura 4 – Modelo entidade relacionamento do Agenda+



Fonte: Os autores

4 Contexto Empreendedor do Projeto

Na atualidade a concorrência em diversos setores está sempre aumentando sua competitividade sendo assim as empresas sofre mudanças rapidamente em um curto período. Para superar essas modificações as organizações estão utilizando o empreendedorismo para criar ideias como produtos ou serviços a fim de fidelizar seus clientes ou conseguir novos de ter um diferencial contra seus concorrentes através de um risco calculado.

De acordo com o Bueno (2016):

Empreendedorismo é a capacidade que uma pessoa tem de identificar problemas e oportunidades, desenvolver soluções e investir recursos na criação de algo positivo para a sociedade. Pode ser um negócio, um projeto ou mesmo um movimento que gere mudanças reais e impacto no cotidiano das pessoas. (BUENO, 2016).

Para Hisrich, Peters e Shepherd, (2014) o empreendedorismo tem um trabalho importante na criatividade e no desenvolvimento dos negócios e na prosperidade de países e regiões e pode começar em um ponto onde o empreendedor acredita o que pode ser uma oportunidade.

Conforme Dornelas, (2008) descreve em seu livro “Empreendedorismo”:

O movimento do empreendedorismo no Brasil começou a tornar forma na década de 1990, quando entidades como Sebrae (Serviço Brasileiro de Apoio às Micro e Pequenas Empresas) e Softex (Sociedade Brasileira para Exportação de Software) foram criadas. Antes disso, praticamente não se falava em empreendedorismo e em criação de pequenas empresas. Os ambientes político e econômico do país não eram propícios, e o empreendedor praticamente não encontrava informações para auxiliá-lo na jornada empreendedora (DORNELAS, 2008, p, 10).

Segundo Macedo, (2015) o Brasil de 2014 a 2015 na Taxa de empreendedorismo em estágio inicial (TEA) saltou no rank de países, agrupados por fase de desenvolvimento de décimo lugar para o oitavo lugar em um grupo composto por 31 países e existem várias tese que firma está ideia sendo uma delas com um país com economia mais evoluída maior a taxa de oportunidade para empreender.

Conforme menciona SBCoaching (2018) o termo *startup* significa uma organização inicial de baixo capital e com crescimento constante com características de inovação, escalabilidade, flexibilidade e trabalho em equipe que busca um modelo altamente lucrativo e escalável sempre buscando a inovação em um cenário de alto risco.

“Uma startup é uma instituição humana projetada para criar novos produtos e serviços sob condições de extrema incerteza” (RIES, 2012).

Conforme Ries (2012) conta em seu livro “A startup enxuta” para aumentar a probabilidade de ter uma *startup* de sucesso o empreendedor tem que atribuir em sua organização um conjunto de práticas que envolve técnicas difundidas como *Lean Thinking*, *Customer Development*, *Agile Methods*, *Minimum Viable Product*.

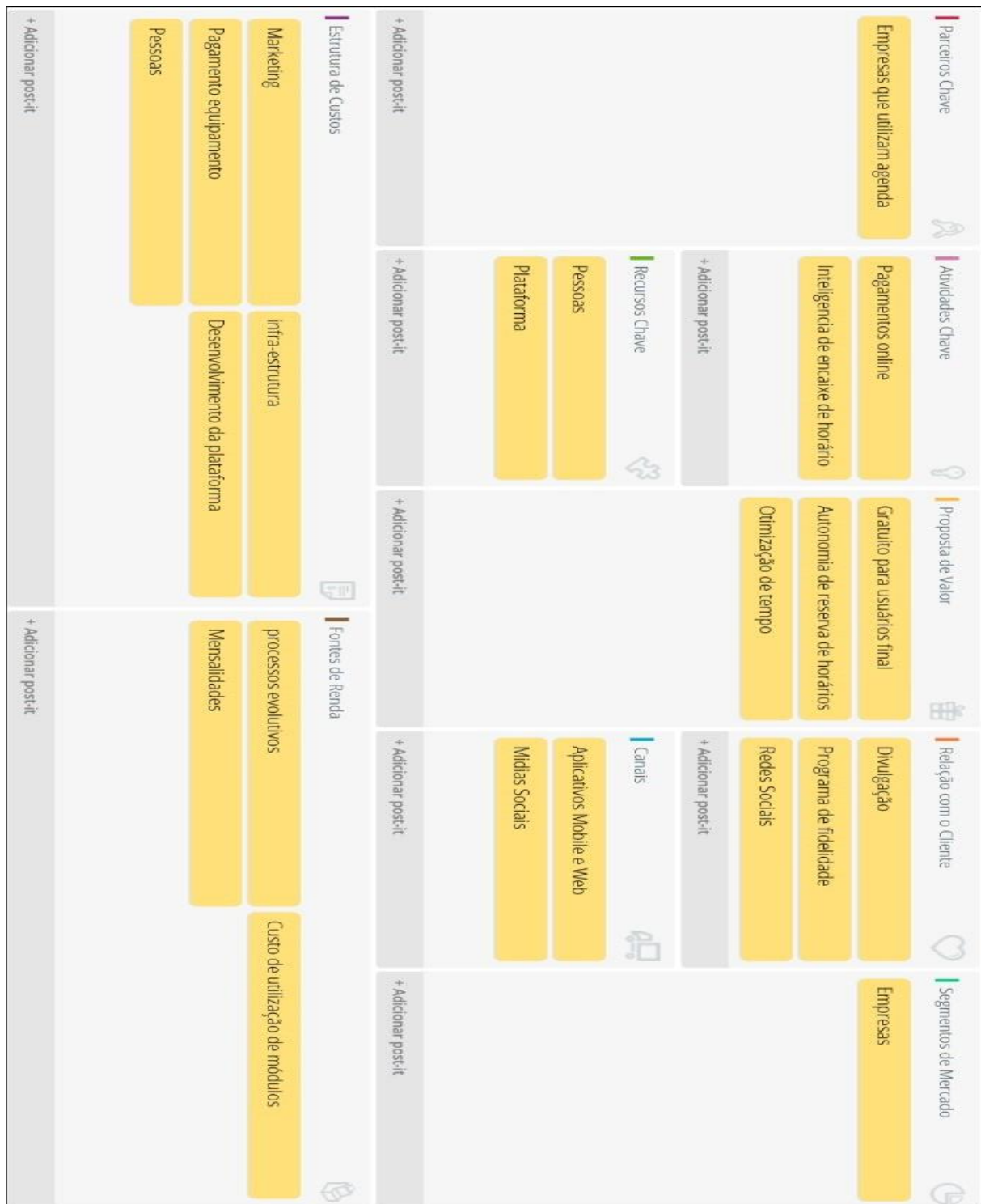
“O quadro (canvas) é uma nova maneira de pensar o modelo de negócios e é a essencial do *Business Model Generation* (BMG) ou *Business Model*

Canvas (BMC). O BMC é uma nova forma de gerar o modelo de negócios” (DE SOUSA NETO, 2014, p. 10).

Segundo Mota, (2019) a metodologia *Canvas* foi criada pelo suíço Alex Osterwalder em sua tese de doutorado, esta metodologia permite a geração de um esquema visual que possibilita as pessoas criarem modelos de negócios com elementos como estruturas de custos, fluxo de receitas, segmentos de clientes, canais de distribuição, atividades, parcerias-chaves, proposta de valor, recursos e relacionamento de clientes.

A figura 5 representa o *canvas* do Agenda+:

Figura 5 – Quadro *canvas* do Agenda+



Fonte: Os autores

O quadro canvas do Agenda+ possui nove blocos para o planejamento estratégico que inclui quais segmentos de clientes serão foco de nosso sistema, custos relevantes para que a proposta funcione, formas de obter receita por meio de propostas de valores entre outros.

5 Desenvolvimento

Nesta divisão iremos descrever o sistema Agenda+ para compreensão de todas as etapas de funcionamento da aplicação móvel que será destinada ao usuário final e também da aplicação *web* na qual é focada para os colaboradores da organização.

Através do Github (2019), poderá ser visualizado todas as interfaces de ambos os sistemas com qualidade alta além de conter todas as telas do sistema *web* que será mostrado no capítulo 5.2.

5.1 Sistema Móvel

5.1.1. Autenticação

A figura 6 representa a tela de autenticação do aplicativo do Agenda+ (Agenda Plus). Para realizar o acesso, o usuário deverá preencher os campos de CPF e *Password* corretamente, além desta funcionalidade, é possível que um novo usuário possa navegar até o local de realização de seu cadastro.

Figura 6 – Tela de autenticação



Fonte: Os autores

5.1.2. Navegação

Após passar pelo sistema de autenticação o usuário será levado para a página que está representada na figura 7, teremos quatro possibilidades de navegação pelo aplicativo, identificando a tela de novo agendamento; lista de estabelecimentos disponíveis para agendamento online; lista de agendamento já realizados; tela de perfil do usuário.

5.1.3. Agendamento

Conforme a imagem representada pela figura 7, o usuário terá a possibilidade de realizar um novo agendamento. Para isso, é preciso selecionar o estabelecimento, profissional, serviço, data e hora disponíveis na agenda do profissional.

O usuário precisará escolher os estabelecimentos que tem interesse, selecionando os favoritos na tela de estabelecimentos com apenas o toque no ícone de estrela conforme a figura 7.

Figura 7 – Tela de agendamento



A imagem mostra a tela de um aplicativo móvel intitulada "Novo Agendamento". No topo, há uma barra de status com o horário 20:33 e ícones de bateria, sinal e Wi-Fi. Abaixo, o título "Novo Agendamento" está em uma barra rosa. O formulário contém campos para: "Estabelecimento" (dropdown com "Selecione o Estabelecimento"), "Profissional" (dropdown com "Selecione o Profissional"), "Serviço" (dropdown com "Selecione o Serviço"), "Data" (campo de texto), e "Hora" (dropdown com "Selecione o Horário"). Um botão verde "Agendar" está na base do formulário. Na parte inferior, há uma barra de navegação rosa com ícones de casa, lista, calendário e perfil.

Fonte: Os autores

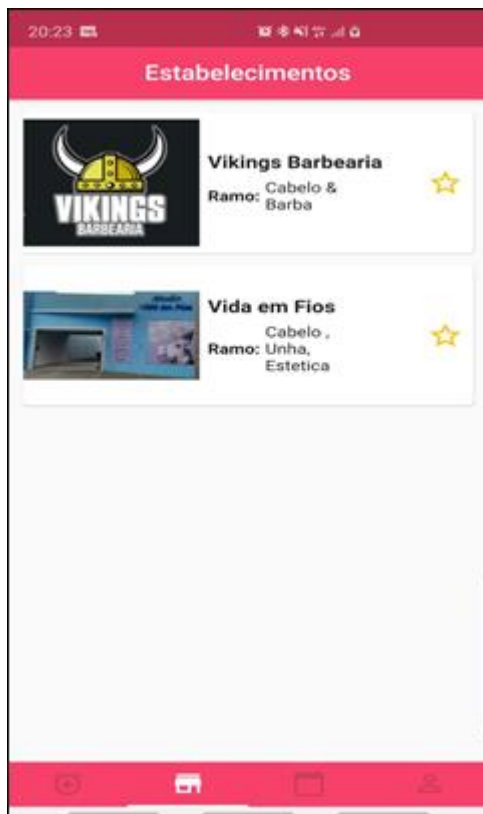
5.1.4. Estabelecimentos

A tela de estabelecimentos encenado pela figura 8, lista todos os estabelecimentos disponíveis na plataforma do Agenda+ (Agenda Plus), o usuário tem

a possibilidade de selecionar o estabelecimento como favorito em apenas com um toque na estrela ao lado direito do banner do estabelecimento.

Quando o usuário dar dois toques no estabelecimento serão exibidos os detalhes conforme figura 9.

Figura 8 – Tela de estabelecimentos



Fonte: Os autores

Figura 9 – Tela de detalhe estabelecimento



Fonte: Os autores

5.1.5. Agendamentos

Nesta tela será listado todos os agendamentos já realizados pelo o usuário trazendo informações como nome do estabelecimento, data, hora, serviço e tempo de duração do serviço.

No canto inferior direito o sistema possui um botão que implementa a funcionalidade de levar o usuário para a página de novo agendamento, todos esses detalhes podem ser visualizados na figura 10.

Figura 10 – Tela de agendamentos



Fonte: Os autores

5.1.6. Perfil

Em perfil o aplicativo recupera todas as informações do usuário durante o processo de cadastro e disponibiliza visualmente conforme imagem na figura 11.

Figura 11 – Tela de perfil



Fonte: Os autores

5.2 Sistema Web

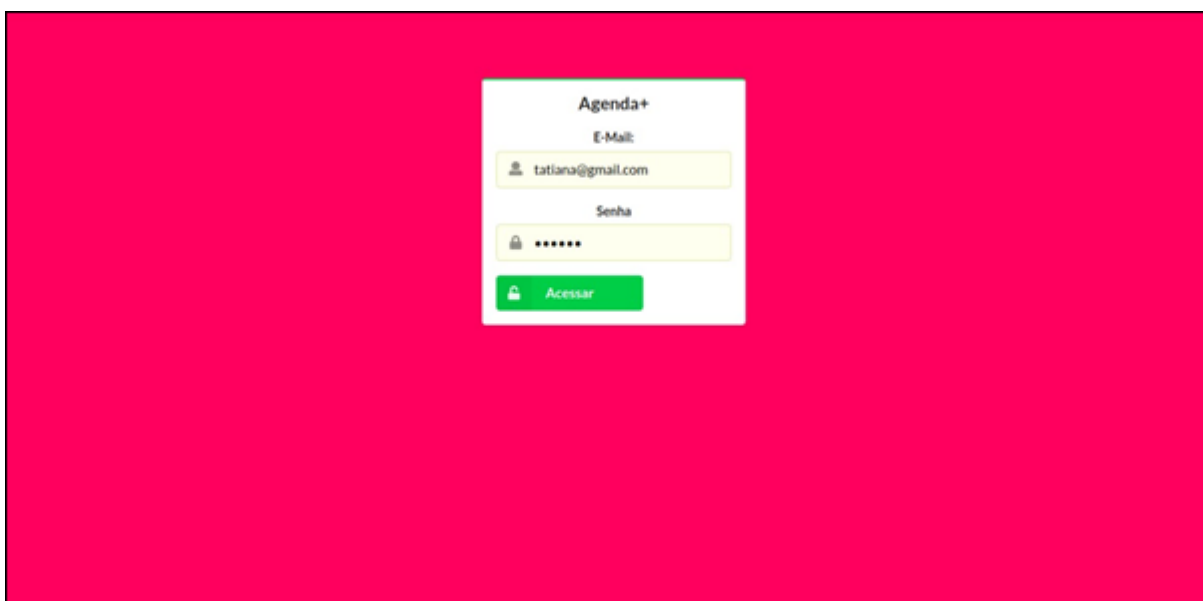
Sistemas web são soluções desenvolvidas para facilitar processos manuais de uma forma mais simples e intuitiva, podendo ser acessado através de navegadores em celulares e computadores de qualquer lugar a qualquer momento.

Ao lado do profissional o Agenda+ (Agenda Plus) disponibilizará um sistema web para auxílio de cadastro de clientes e também um melhor acesso de sua agenda podendo consultar através de filtros horários marcados.

5.2.1. Autenticação

A figura 12 representa a página de autenticação do sistema *web*, o usuário administrador da organização deverá inserir os campos de e-mail e senha posteriormente clicar no botão "Acessar".

Figura 12 – Tela de perfil

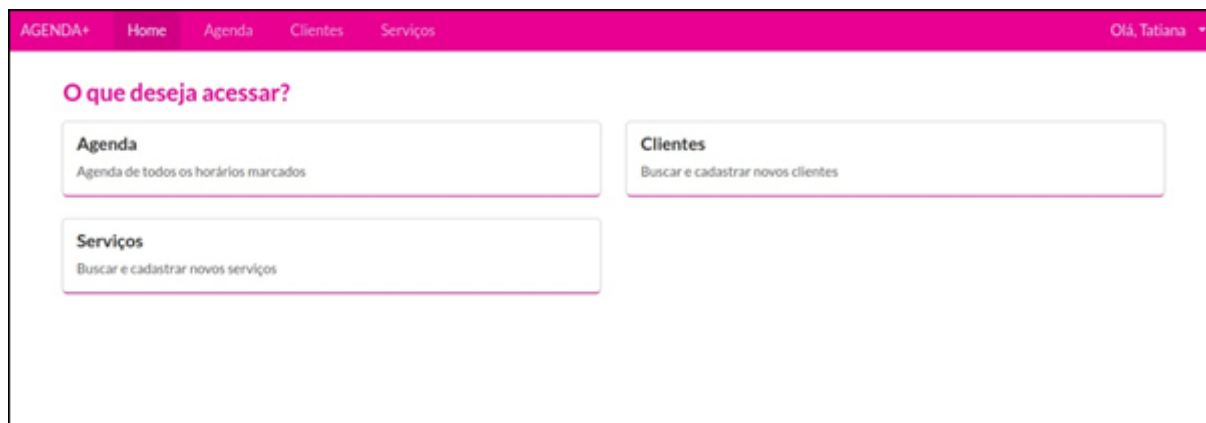


Fonte: Os autores

5.2.2. Página Inicial

Após a autenticação ser validada o usuário será direcionado para a tela inicial que contaram com três opções sendo agenda para visualizar todos os horários marcados e realizar agendamento; clientes para visualizar todos dos clientes cadastrados além de realizar novos cadastros; serviços para realizar buscas e cadastro, conforme figura 13.

Figura 13 – Tela de menu



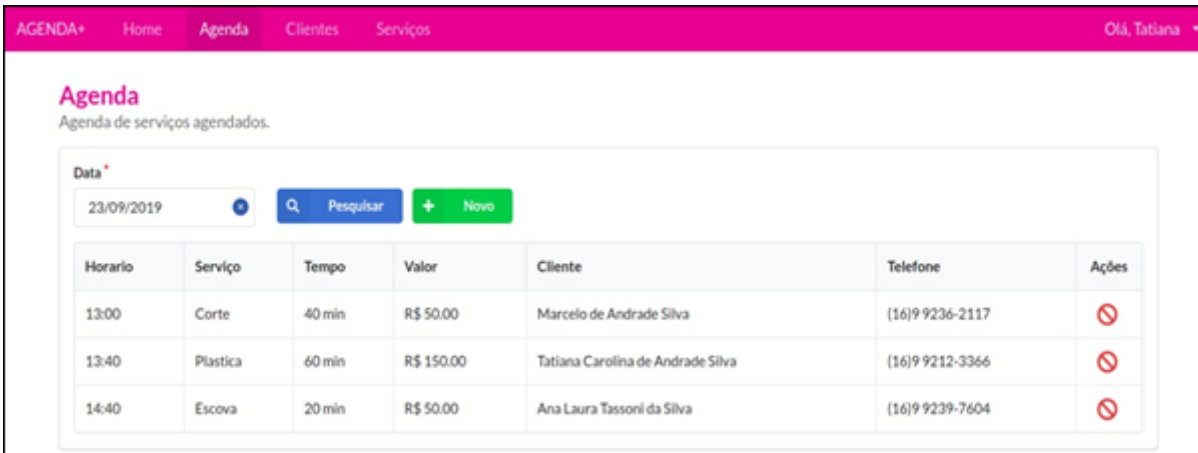
Fonte: Os autores

5.2.3. Agenda




Na tela de agenda serão listados todos os agendamentos pelo aplicativo móvel e pelo sistema *web* conforme representado pela figura 14.

A interface ainda conta com funcionalidades de filtrar agendamentos por datas, realizar um cancelamento de um horário já reservado e um botão de novo agendamento que fará que o usuário seja redirecionado para a página de realizar um novo agendamento.

Figura 14 – Tela da agenda



The screenshot shows the 'Agenda' screen with a pink header containing navigation tabs: 'AGENDA+', 'Home', 'Agenda', 'Clientes', and 'Serviços'. The user's name 'Olá, Tatiana' is visible in the top right. Below the header, the title 'Agenda' is followed by the subtitle 'Agenda de serviços agendados.' A date selector is set to '23/09/2019' with a search icon and a '+ Novo' button. Below this is a table with the following data:

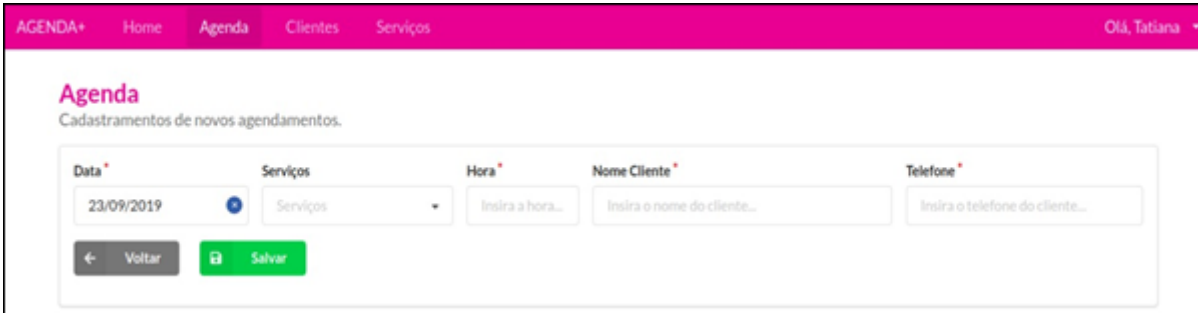
| Horario | Serviço | Tempo | Valor | Cliente | Telefone | Ações |
|---------|----------|--------|------------|-----------------------------------|-----------------|---|
| 13:00 | Corte | 40 min | R\$ 50.00 | Marcelo de Andrade Silva | (16)9 9236-2117 |  |
| 13:40 | Plastica | 60 min | R\$ 150.00 | Tatiana Carolina de Andrade Silva | (16)9 9212-3366 |  |
| 14:40 | Escova | 20 min | R\$ 50.00 | Ana Laura Tassoni da Silva | (16)9 9239-7604 |  |

Fonte: Os autores

5.2.4. Cadastro de Agendamento

A interface entrega a funcionalidade de realizar um cadastramento de agendamento assim facilitando a manipulação da agenda e evitando a dependência do empreendedor com os usuários do aplicativo, conforme figura 15.

Figura 15 – Tela de cadastro de agendamento



The screenshot shows the 'Cadastro de Agendamento' screen with the same pink header. The title 'Agenda' is followed by the subtitle 'Cadastramentos de novos agendamentos.' The form includes fields for 'Data' (set to 23/09/2019), 'Serviços' (a dropdown menu), 'Hora' (with placeholder 'Insira a hora...'), 'Nome Cliente' (with placeholder 'Insira o nome do cliente...'), and 'Telefone' (with placeholder 'Insira o telefone do cliente...'). At the bottom, there are 'Voltar' and 'Salvar' buttons.

Fonte: Os autores

Após realização de cadastro nesta interface o sistema irá preencher a agenda conforme os dados inseridos além de retirar no aplicativo a data e horário desde agendamento para evitar cadastro duplicado.

6 Conclusão

De uma forma geral, conseguimos alcançar o propósito principal do trabalho, desenvolver um aplicativo móvel para as plataformas ANDROID e IOS e um sistema web que pode ser acessado em qualquer navegador utilizando internet, ambos com um *design* bonito e intuitivo.

O presente projeto possui uma estrutura desenvolvida com o objetivo de criar um MVP (*Minimum Viable Product*) a fim de solucionar um problema e facilitar o agendamento de serviços. Consequentemente no decorrer deste desenvolvimento, foram analisadas melhorias e utilidades que ao serem desenvolvidos, melhorará a experiência de ambos os lados ao utilizarem o Agenda+ (Agenda Plus). Sendo assim novas funcionalidades como lembretes de agendamentos realizados em momentos antes do horário marcado, pagamento antecipado de serviços como garantia para o profissional evitando cancelamentos de última hora e modos de divulgação de novos serviços ou promoções que os estabelecimentos possam oferecer.

Referências

ANDROID. Disponível em: https://www.android.com/intl/pt-BR_br/. Acesso em: 10 ago. 2019, 17:07:13.

APPLE DEVELOPER. Disponível em: <https://developer.apple.com>. Acesso em 21 ago. 2019, 08:41:02.

APPLE. Disponível em <https://www.apple.com/br/>. Acesso em 21 ago. 2019, 09:13:05.

AQUINO, Juliana França Santos. Plataformas de Desenvolvimento para Dispositivos Móveis. 2007. Disponível em: <http://www-di.inf.puc-rio.br/~endler/courses/Mobile/Monografias/07/Android-Juliana-Mono.pdf>. Acesso em: 15 ago. 2019, 11:54:39.

BUENO, Jefferson. Mas afinal, o que é empreendedorismo?. 2016. Disponível em: <https://blog.sebrae-sc.com.br/o-que-e-empreendedorismo/>. Acesso em: 9 set. 2019, 08:12:39.

BUNA, Samer. Learning GraphQL and relay. Birmingham: Packt Publishing Ltd, 2016.

DAGNE, Lukas. Flutter for cross-platform App and SDK development. 2019. Disponível em: <https://www.theseus.fi/bitstream/handle/10024/172866/Lukas%20Dagne%20Thesis.pdf?sequence=2>. Acesso em: 5 set. 2019, 15:20:25.

DATE, Christopher J. Introdução a sistemas de bancos de dados. Tradução de Daniel Vieira. Rio de Janeiro: Elsevier Brasil, 2004.

DE SOUSA NETO, Manoel Veras. Gerenciamento de projetos: project model canvas (PMC). Rio de Janeiro: Brasport, 2014.

DORNELAS, José Carlos Assis. Empreendedorismo. 3. ed. Rio de Janeiro: Elsevier Brasil, 2008.

ELMASRI, Ramez; NAVATHE, Shamkant B. Sistemas de banco de dados. Tradução de Daniel Vieira. 7. ed. São Paulo: Pearson Universidades, 2005.

FLUTTER. Disponível em: www.flutter.dev. Acesso em: 11 ago. 2019, 14:44:19.

GACKENHEIMER, Cory. Introduction to React. Nova York: Apress, 2015.

GITHUB. Disponível em: https://github.com/jeancarloferreira/tcc_agenda_plus. Acesso em 16 ago. 2019, 13:13:39.

GRAPHQL. Disponível em: <https://graphql.org>. Acesso em: 11 ago. 2019, 09:17:57.

HERRON, David. Node Web Development. Birmingham: Packt Publishing Ltd, 2013.

HISRICH, Robert D.; PETERS, Michael P.; SHEPHERD, Dean A. Empreendedorismo. Tradução de Francisco Araújo da Costa. 9. ed. Porto Alegre: AMGH Editora, 2014.

IBGE. Acesso à internet e à televisão e posse de telefone móvel celular para uso pessoal 2016. 2018. Disponível em: <https://biblioteca.ibge.gov.br/visualizacao/livros/liv101543.pdf>. Acesso em: 9 ago. 2019, 08:47:13.

JOEL. Modelo Entidade Relacionamento (MER) e Diagrama Entidade-Relacionamento (DER). 2014. Disponível em: <https://www.devmedia.com.br/modelo-entidade-relacionamento-mer-e-diagrama-entidade-relacionamento-der/14332>. Acesso em: 9 set. 2019, 12:48:27.

KRAJCI, Iggy; CUMMINGS, Darren. Android on x86: An Introduction to Optimizing for Intel Architecture. Nova York: Apress, 2013.

LEUCOTRON. Como a tecnologia auxilia na melhora de produtividade no trabalho? 2019. Disponível em: <https://blog.leucotron.com.br/como-a-tecnologia-auxilia-na-melhora-da-productividade-no-trabalho/>. Acesso em: 24 ago. 2019, 22:39:15.

MACEDO, Mariano. Empreendedorismo no Brasil: 2015. 2015. Disponível em: <http://cer.sebrae.com.br/wp-content/uploads/2015/12/7347.pdf>. Acesso em: 9 set. 2019, 13:17:45.

MILANI, André. MySQL-guia do programador. São Paulo: Novatec Editora, 2007.

MORRISSEY, Sean; CAMPBELL, Tony. iOS Forensic Analysis: for iPhone, iPad, and iPod touch. Nova York: Apress, 2011.

MOTA, Gleison. Canvas: o que é e para que serve?. 2019. Disponível em: <https://administradores.com.br/artigos/canvas-o-que-e-e-para-que-serve>. Acesso em: 11 set. 2019, 16:32:10.

NODE JS. Disponível em: <https://nodejs.org/en/>. Acesso em: 11 ago. 2019, 07:39:42.

NOGUEIRA, Ronaldo. A falta de tempo no mundo moderno. 2016. Disponível em: <https://freesider.com.br/tempo/falta-de-tempo-no-mundo-moderno/>. Acesso em: 24 ago. 2019, 22:17:13.

PEREIRA, Lucio Camilo Oliva; DA SILVA, Michel Lourenço. Android para desenvolvedores. Rio de Janeiro: Brasport, 2009.

PRESSMAN, Roger; MAXIM, Bruce. Engenharia de Software. Tradução de João Eduardo Nóbrega Tortello. 8. ed. Pinheiros: McGraw Hill Brasil, 2016.

REACT. Disponível em: <https://pt-br.reactjs.org>. Acesso em: 12 ago. 2019, 15:35:17.

REZENDE, Denis Alcides. Engenharia de software e sistemas de informação. Rio de Janeiro: Brasport, 2006.

RIES, Eric. A startup enxuta. Lauro de Freitas: Leya, 2012.

SBCOACHING. Startup: Definição, Tipos, exemplos e Dicas Para o Sucesso. Disponível em: <https://www.sbcoaching.com.br/blog/startup/>. Acesso em 9: set. 09:27:40.

SILBERSCHATZ, Abraham; SUNDARSHAN, S.; KORTH, Henry F. Sistema de banco de dados. Tradução de Daniel Vieira. Rio de Janeiro: Elsevier Brasil, 2016.

TILKOV, Stefan; VINOSKI, Steve. Node. js: Using JavaScript to build high-performance network programs. 2010. Disponível em: <https://pt.slideshare.net/hustwj/nodejs-using-javascript-to-build-highperformance-network-programs>. Acesso em 23 ago. 2019, 17:16:14.

WIDENIUS, Michael; AXMARK, David; ARNO, Kaj. MySQL reference manual: documentation from the source. Sebastopol: O'Reilly Media, Inc., 2002.

WU, Wenhao. React Native vs Flutter, Cross-platforms mobile application frameworks. 2018. Disponível em: <https://www.theseus.fi/bitstream/handle/10024/146232/thesis.pdf?sequence=1>. Acesso em 24 ago. 2019, 15:52:14.