

CENTRO UNIVERSITÁRIO DO ESTADO DO PARÁ
ÁREA DE CIÊNCIAS EXATAS E TECNOLOGIA
BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO

Fabício da Silva Fonseca

APLICAÇÃO MÓVEL PARA AGENDAMENTO DE AULAS DE
AUTOESCOLA - AGENDADOR

Belém

2017

CENTRO UNIVERSITÁRIO DO ESTADO DO PARÁ
ÁREA DE CIÊNCIAS EXATAS E TECNOLOGIA
BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO

Fabício da Silva Fonseca

APLICAÇÃO MÓVEL PARA AGENDAMENTO DE AULAS DE
AUTOESCOLA - AGENDADOR

Trabalho de Curso na modalidade Produto, apresentado como requisito parcial para obtenção do grau em Bacharelado em Engenharia de Computação do Centro Universitário do Estado do Pará – CESUPA, sob orientação do Professor MSc. Ricardo Melo Casseb do Carmo.

Belém

2017

Dados Internacionais de Catalogação-na-publicação (CIP)
Biblioteca do Cesupa, Belém - PA

Fonseca, Fabrício da Silva.

Aplicação móvel para agendamento de aulas de autoescola - Agendador /
Fabrício da Silva Fonseca; orientação de Ricardo Melo Casseb do Carmo, 2017.

Trabalho de Conclusão de Curso (Bacharelado em Engenharia da Computação)
– Centro Universitário do Pará, Belém, 2017.

1. Smartphone. 2. Sistemas de comunicação móvel. I. Carmo, Ricardo
Melo Casseb do (orient.). II. Título.

CDD. 23° ed. 004.6

Fabrcio da Silva Fonseca

**APLICAO MVEL PARA AGENDAMENTO DE AULAS DE
AUTOESCOLA - AGENDADOR**

Trabalho de Curso apresentado na modalidade produto, apresentado como requisito parcial para obtenao do grau em Bacharelado em Engenharia de Computao do Centro Universitrio do Estado do Par - CESUPA.

Data da Defesa: 14/12/2017

Banca Examinadora:

Prof. Orientador MSc. Ricardo Melo Casseb do Carmo - CESUPA

Prof. MSc. Johnny Marcus Gomes Rocha - CESUPA

Prof.^a. MSc. Alessandra Natasha Alcântara Barreiros - CESUPA

Belém

2017

A minha família, com todo amor e carinho.

AGRADECIMENTOS

Agradeço primeiramente, à Deus por estar ao meu lado, sempre me protegendo de todo mal, me guiando por caminhos certos e me dando forças quando não tenho, para enfrentar todas as dificuldades e vencê-las.

Agradeço a minha avó Bernadete Baia por todo amor, carinho e dedicação dados a mim para que eu pudesse alcançar todos os objetivos incluindo a minha graduação.

Agradeço ao meu avô Benedito Baia por me ensinar que, a melhor forma de alcançar nossos objetivos é o trabalho e que independentemente do trabalho devemos fazê-lo com total dedicação e amor.

Agradeço a minha mãe Luciana Baia por sempre acreditar na minha capacidade e confiar na minha competência mesmo que em situações difíceis. Agradeço por ter me ensinado a manter foco nos estudos, quando pequeno passava horas ao meu lado me ensinado e exercitando os conteúdos ensinados na escola.

Agradeço ao meu pai Jerson Fonseca, por me ensinar a trabalhar arduamente naquilo que queremos e que apesar das dificuldades sempre podemos alcançar os objetivos e que no fim todo esforço será recompensado. Agradeço também por ter me ensinado a ser uma pessoa justa e honesta que sempre busca ser melhor a cada dia.

Agradeço a esposa de meu pai, Clarice Melo, sempre que tinha oportunidade vinha até mim com mensagens de motivação, e com elogios a cada etapa concluída, além de ser tratado de forma especial com todo carinho e respeito.

Agradeço aos avós da minha namorada, Eulídio Silva e Zeferina Silva por todos os conselhos e motivações que recebi deles, agradeço por me tratarem como um filho, ambos são muito especiais em minha vida.

Agradeço a minha namorada Ivani Barros, que sempre acredita na minha capacidade de ser um ótimo engenheiro, que sempre está ao meu lado para me aconselhar e também me corrigir, nos momentos de fraqueza durante o curso, nossas conversas foram de total importância para eu seguir em frente. Obrigado por todo amor, carinho e companheirismo dado a mim, você além de namorada é uma excelente amiga.

Agradeço aos colegas de curso Andrey Favacho, Barbara Lobo, Bruno Ramos e Julia Guimarães, estes que junto comigo enfrentaram as dificuldades do curso, mas nunca desistiram. Desejo a todos os colegas, sucesso e que se tornem engenheiros renomados e reconhecidos por seus trabalhos.

Agradeço aos professores Itamar Brito, Johnny Rocha, Elton Rafael, Polyana Fonseca, Eliane Alves, Gandhi Aranha, Cláudio Serra, Cindy Fernandes, Michele Bitar e Alessandra Natasha por me ensinarem que a engenharia é uma área muito complexa, porém uma área que nos faz cada dia mais, querer solucionar problemas cada vez mais complexos. Obrigado a todos vocês professores, cada um de vocês contribuiu para a formação do meu perfil de engenheiro e por me mostrar esse universo fantástico chamado engenharia de computação.

Agradeço ao professor Ricardo Casseb por ser um gênio da programação, sua habilidade em programação é admirável, pois os conceitos e aplicações ensinados durante o curso já não são leituras de livros, mas sim parte do dia a dia do professor, o qual demonstra domínio absoluto sobre eles. Obrigado por me orientar durante todo este projeto, sua contribuição foi crucial para a conclusão do mesmo.

Por fim agradeço ao Cesupa, por fornecer a estrutura adequada para desempenhar todas as atividades do curso, agradeço por todos os colaboradores desde a portaria até ao reitor, sempre nos tratando com total respeito e oferecendo seus serviços com a melhor qualidade.

Pois todas as coisas foram criadas por
Ele, e tudo existe por meio d'Ele e para Ele.
Glória a Deus para sempre! Amém!

Romanos 11:36

RESUMO

Por causa do aumento do uso de smartphones para acessar a internet e a realização de tarefas cotidianas através de aplicativos, como pedir comida e realizar pagamentos, foi observada a possibilidade de criação de uma ferramenta para facilitar a realização de tarefas cotidianas. Agendar aulas dependendo da intervenção de terceiros se torna uma tarefa demorada e as vezes é difícil de concluir, várias vezes o aluno tem muito do seu tempo consumido devido a espera para realizar um agendamento, outro empecilho é a demora no atendimento através de telefone quando o aluno consegue atendimento por telefone. Falhas desta forma tornam um simples processo de agendar aula em algo complexo, demorado e estressante. Portanto este trabalho propõe uma solução desenvolvida com o framework Ionic integrado a uma base de dados para gerar uma aplicação móvel que realize a gestão de horários de aulas, agendamento de aulas através da aplicação, sem necessidade de confirmação por pessoas, apenas através do sistema, com a intenção de automatizar o processo, possibilitando que o aluno realize seu agendamento no menor tempo possível através de seu smartphone em qualquer lugar.

Palavras-chave: Smartphone. Tarefas Cotidianas. Agendamento. Aplicação Móvel.

ABSTRACT

Because of the increase in the use of smartphones to access an internet and the accomplishment of daily tasks through applications such as ordering food and making payments, a possibility of creating a tool to facilitate the accomplishment of daily tasks was observed. Scheduling classes depending on the third party intervention becomes a time consuming task and as times are difficult to complete, several times or a lot of time with your time consumed due to a wait to execute a schedule, another hindrance is a delay without answering by phone when Student can get phone service. Failures in this way make a scheduler process complex, time-consuming and stressful. This is a database development program to generate a mobile application that performs the management of class schedules, scheduling classes through the application, without need for confirmation by people, only through the system, with an intention to automate the process, making it possible for you to perform your schedule in the shortest time possible through your smartphone anywhere.

Keywords: Smartphone. Daily Tasks. Scheduling. Mobile Application.

LISTA DE FIGURAS

Figura 1 – Ranking de tempo consumido na utilização de smartphone	17
Figura 2 – Características dos processadores para SE.....	23
Figura 3 – Lista de Sistemas Embarcados	24
Figura 4 – Arquitetura básica para processadores de SE.....	24
Figura 5 – Utilização de sistemas operacionais em dispositivos mobile.....	27
Figura 6 – Diferenças entre paradigmas de desenvolvimento de aplicativos.....	30
Figura 7 – Tela de perfil de aluno	31
Figura 8 – Tela de progresso do aluno	32
Figura 9 – Histórico de aulas práticas	33
Figura 10 – Diagrama de funcionamento	34
Figura 11 – Tecnologias utilizadas.....	35
Figura 12 – Arquitetura de aplicação Cordova.....	37
Figura 13 – Exemplo de armazenamento de dados em JSON.....	41
Figura 14 – Modelo JSON para armazenar horário de aula	41
Figura 15 – Diagrama de caso de uso.....	42
Figura 16 – Diagrama de pacotes	44
Figura 17 – E-mail de confirmação de cadastro	45
Figura 18 – Tela de confirmação de e-mail.....	46
Figura 19 – Tela de login projetada.....	46
Figura 20 – Tela de login construída	47
Figura 21 – Tela de recuperação de senha projetada.....	47
Figura 22 – Tela de recuperação de senha construída	48
Figura 23 – E-mail de redefinição de senha	48
Figura 24 – Tela de redefinição de senha	49
Figura 25 – Tela inicial de aluno projetada	49
Figura 26 – Tela inicial de aluno construída	50
Figura 27 – Tela de agendamento projetada.....	50
Figura 28 – Tela de agendamento construída.....	51
Figura 29 – Tela inicial do administrador projetada.....	52
Figura 30 – Tela inicial do administrador construída.....	52
Figura 31 – Tela projetada.....	53
Figura 32 – Tela construída.....	53

Figura 33 – Tela do serviço de autenticação do firebase.....54

LISTA DE SIGLAS

ANATEL	Agência Nacional de Telecomunicações
API	<i>Application Programming Interface</i>
APP	Aplicativo
ASIP	<i>Application-specific instruction set processor</i>
CPU	Unidade Central de Processamento
CSS	<i>Cascading Style Sheets</i>
DETRAN-RJ	Departamento de Trânsito do Estado do Rio de Janeiro
DSP	<i>Digital signal processor</i>
E-MAIL	<i>Eletronic mail</i>
FGV-SP	Fundação Getúlio Vargas de São Paulo
GPS	Sistema de posicionamento global
HTML	<i>HyperText Markup Language</i>
JSON	<i>JavaScript Object Notation</i>
MDN	<i>Mozilla Developer Network</i>
PC	<i>Personal computer</i>
SE	Sistemas Embarcados
SO	Sistema operacional
ULA	Unidade Lógica e Aritmética
UML	Linguagem de Modelagem Unificada

SUMÁRIO

1	INTRODUÇÃO.....	16
1.1	OBJETIVOS	18
1.1.1	Objetivo geral.....	18
1.1.2	Objetivos específicos.....	18
1.2	JUSTIFICATIVA	18
1.3	ESTRUTURA DO TRABALHO.....	19
2	SISTEMAS EMBARCADOS	21
3	APLICATIVOS MOBILE	26
4	TRABALHOS RELACIONADOS.....	31
5	MATERIAIS E MÉTODOS	34
5.1	FUNCIONAMENTO	34
5.2	LISTA DE FERRAMENTAS.....	35
5.2.1	Engenharia e projeto	36
5.2.1.1	Astah Community.....	36
5.2.1.2	Balsamiq Mockups	36
5.2.1.3	Draw.io	36
5.2.2	Desenvolvimento híbrido	36
5.2.2.1	Node.js.....	36
5.2.2.2	Apache Cordova	37
5.2.2.3	Ionic	37
5.2.2.4	Angular	37
5.2.2.5	HTML 5	38
5.2.2.6	JavaScript	38
5.2.2.7	TypeScript	38
5.2.2.8	CSS.....	38
5.2.2.9	Google Firebase.....	38
5.2.2.10	Ionic Creator.....	39
5.2.2.11	Visual Studio Code.....	39
5.2.2.12	Ionic DevApp	39

5.2.2.13 Android SDK.....	40
5.2.2.14 Gradle	40
5.3 ENGENHARIA DE SOFTWARE.....	40
5.3.1 Modelagem do banco de dados.....	40
5.3.2 Diagrama de caso de uso	42
5.3.3 Diagrama de pacotes.....	43
5.4 FUNCIONALIDADES.....	45
5.4.1 Aluno.....	45
5.4.1.1 Cadastrar.....	45
5.4.1.2 Entrar	46
5.4.1.3 Recuperar senha.....	47
5.4.1.4 Visualizar horários agendados.....	49
5.4.1.5 Visualizar horários disponíveis	50
5.4.1.6 Agendar horários	51
5.4.2 Administrador.....	51
5.4.2.1 Entrar	51
5.4.2.2 Visualizar horários criados	52
5.4.2.3 Criar horários	53
6 RESULTADOS E DISCUSSÕES.....	54
7 CONSIDERAÇÕES FINAIS	55
7.1 DIFICULDADES ENCONTRADAS.....	55
7.2 PROPOSTA DE TRABALHOS FUTUROS	56
8 REFERÊNCIAS.....	57

1 INTRODUÇÃO

O uso de smartphones vem crescendo exponencialmente, de maneira que nos dias atuais as pessoas preferem o uso de aplicativos do que acessar sites ou fazer ligações. Isso ocorre devido a esses dispositivos possibilitarem a execução de simples tarefas do dia a dia como pedir comida, ouvir músicas, comunicar-se com amigos, verificar e-mails, fazer compras e outras tarefas que fazem parte do nosso cotidiano (DINO, 2017).

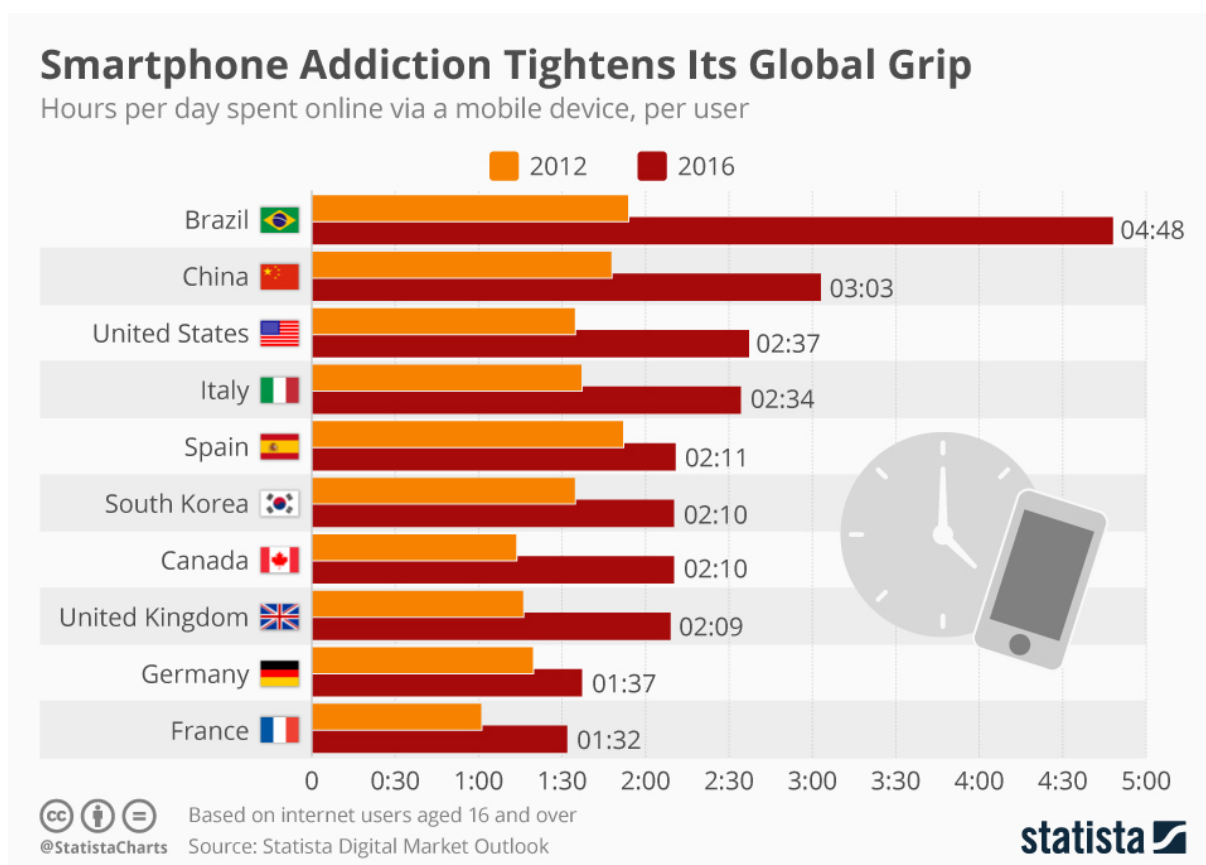
A Agência Nacional de Telecomunicações (ANATEL, 2017), registrou o aumento de mais de 150 milhões de linhas em operação. Esse aumento de linhas está relacionado diretamente ao uso de smartphones, pois, quanto maior o número de linhas operando maior é o número de smartphones sendo utilizados. A facilidade de comunicação e de execução de tarefas simples que um smartphone oferece, tem feito com que a maioria da população economize tempo com a utilização de aplicativos móveis.

Por exemplo atualmente é possível acompanhar notícias em tempo real, alguns anos atrás esse acompanhamento era feito apenas por rádio, televisão e jornal impresso. Outro exemplo que tem economizado o tempo das pessoas são os serviços bancários digitais, estes serviços oferecidos são visualização de saldo e extrato, transferência de valores para outras contas, pagamentos de boletos que podem ser lidos a partir da câmera do celular, recarga de celular e outros serviços oferecidos dependendo do banco. Segundo Ramos e Costa (2000) as principais vantagens em utilizar esses serviços são diminuição em filas e idas do cliente ao banco, redução de custos operacionais, manter o banco moderno e informatizado e agilidade na execução de alguns serviços como uma simples transferência de valores.

Esses são apenas alguns exemplos de como a utilização do smartphone juntamente com aplicativos facilitam a vida das pessoas. Porém algumas pessoas dificultam suas vidas com o uso da mesma tecnologia, Agrela (2017) menciona uma pesquisa realizada pela Fundação Getúlio Vargas de São Paulo (FGV-SP), onde está previsto que haverá cerca de 236 milhões de celulares em uso no ano de 2018 no Brasil. Agrela (2017) cita ainda uma outra pesquisa feita pelo Instituto Delete que se preocupa com bem-estar dos brasileiros, a pesquisa possui questionário que verifica se o usuário é um viciado em celular ou não. E por fim é apresentado um ranking feito pela Statista que demonstra o tempo de uso médio do celular nos 10 países que mais utilizam celulares no mundo.

O ranking tem o Brasil como líder em passar maior parte do tempo utilizando o celular, em segundo lugar tem-se a China e em terceiro e quarto lugar com uma pequena diferença estão Estados Unidos e Itália. O ranking foi feito com base em pessoas com 16 anos de idade ou mais. Pode-se verificar que um brasileiro permanece utilizando o celular em aproximadamente 5 horas por dia, porém este ranking foi feito com dados até o ano de 2016. Atualmente este número provavelmente aumentou ou pode até ter diminuído. Abaixo está ilustrado o ranking com a Figura 1 da pesquisa realizada pela Statista.

Figura 1 – Ranking de tempo consumido na utilização de smartphone



Fonte: Agrela (2017, online)

Tendo em vista as facilidades oferecidas para automatizar tarefas no dia a dia da população e o tempo consumido em utilização de smartphones. Este trabalho demonstra o estudo da utilização dos smartphones, para o desenvolvimento de uma aplicação mobile que ofereça ao usuário o agendamento de aulas de autoescola. Para a realização do

desenvolvimento do projeto foi utilizado o paradigma de desenvolvimento de aplicativos híbridos, com a ferramenta Ionic. Esta aplicação permitirá o usuário agendar suas aulas através da aplicação sem a interferência humana, em algumas autoescolas, as aulas ainda são agendadas por telefone, algumas já utilizam o aplicativo de comunicação WhatsApp, este, porém ainda tem a necessidade da intervenção de uma pessoa para atender o aluno.

1.1 OBJETIVOS

Os objetivos desta pesquisa são divididos em geral e específicos, de acordo com os tópicos abaixo.

1.1.1 Objetivo geral

Desenvolver um aplicativo híbrido para agendamento de aulas de autoescola.

1.1.2 Objetivos específicos

- Compreender a importância dos smartphones em facilitar a vida das pessoas no dia a dia.
- Identificar as formas de desenvolvimento de um aplicativo móvel.
- Desenvolver aplicativo a partir da plataforma Ionic para agendamento de aulas.
- Integrar a aplicação a um banco de dados de tempo real.

1.2 JUSTIFICATIVA

Como citado anteriormente cada vez mais os smartphones são utilizados para auxiliar em tarefas cotidianas, essa facilidade em automatizar as tarefas através de aplicativos de smartphones está ligada diretamente ao uso de internet. Na reportagem de Campos (2016) foi apresentado que o principal meio de acesso à internet é o celular, tal uso demonstra que o meio tradicional de acesso que era feito pelo microcomputador vem perdendo força, o principal fator é que o custo de um smartphone é bem menor que de um microcomputador, além de poder acessar a internet com apenas alguns valores das operadoras.

A facilidade de poder realizar várias tarefas com um simples smartphone e acesso à internet, tem obrigado empresas a se adaptar com as mudanças causadas pelo uso do smartphone, por exemplo a escola de inglês Minds possui um curso com horário flexível, onde o aluno pode agendar suas aulas através de várias formas como telefone, WhatsApp e

um sistema web (MINDS, 2017). A forma de horários flexíveis é também utilizada em autoescolas, pois os alunos não possuem horários fixos para realizar suas aulas práticas de direção. Atualmente a maioria das autoescolas fazem os agendamentos de aulas práticas presencialmente ou por telefone, esta maneira de realizar agendamento dificulta a agilidade do processo para alguns alunos, as pessoas possuem várias obrigações e tarefas a cumprir em seu cotidiano, elas nem sempre possuem disponibilidade para ir até a autoescola ou tempo para efetuar uma ligação e solicitar seu agendamento.

Baseado no contexto demonstrado, observa-se que uma forma simples e prática para realizar agendamento de aulas, é desenvolver uma solução para smartphones que permita o aluno agendar suas aulas sem precisar realizar uma ligação, ir até a autoescola ou enviar mensagens. Dessa forma, a motivação desse trabalho é criar uma ferramenta que automatize as tarefas cotidianas das pessoas, utilizando tecnologias que já são utilizadas pela população, que são smartphones, aplicativos móveis e internet.

1.3 ESTRUTURA DO TRABALHO

Além do capítulo introdutório citado, o trabalho é organizado da seguinte forma:

- Capítulo 1 – neste capítulo é apresentado o contexto, objetivos, motivação e a justificativa deste projeto.
- Capítulo 2 – trata da primeira parte do referencial teórico, descrevendo em detalhes o que é um sistema embarcado e definindo os tipos desses sistemas e apresentando os smartphones como um sistema embarcado.
- Capítulo 3 – trata da segunda parte do referencial teórico, descrevendo as aplicações móveis, os paradigmas de desenvolvimento de uma aplicação e as diferenças entre cada paradigma.
- Capítulo 4 – Trabalhos relacionados. Neste capítulo são apresentados produtos relacionados ao projeto deste trabalho, três produtos são demonstrados, porém nenhum possui a principal característica esperada por este trabalho.
- Capítulo 5 – neste capítulo é apresentado o funcionamento da aplicação, lista de ferramentas utilizadas para desenvolver a aplicação, a engenharia de software no processo de desenvolvimento e as funcionalidades do aplicativo.

- Capítulo 6 – neste capítulo são apresentados os resultados alcançados pela aplicação e quais facilidades são oferecidas ao desenvolvedor com base nos resultados.
- Capítulo 7 – neste capítulo são apresentadas as conclusões obtidas a partir da pesquisa e do produto deste projeto, dificuldades encontradas e trabalhos futuros.
- Capítulo 8 – apresenta as referências bibliográficas utilizadas para o estudo técnico de todos os pontos relevantes à pesquisa.

2 SISTEMAS EMBARCADOS

Os sistemas embarcados (SE) estão presentes em vários setores, podem ser encontrados em qualquer dispositivo eletrônico atualmente, tais sistemas têm aplicações em vários setores como a indústria automobilística por exemplo.

Sistemas embarcados são definidos como:

Creating a precise definition of such embedded computing systems, or simply *embedded systems*, is not an easy task. We might try the following definition: An embedded system is nearly any computing system other than a desktop, laptop, or mainframe computer (VAHID; GIVARGIS, 1999, p. 1).¹

Para encontrar a definição mais completa de SE devesse analisar as características comuns que esses sistemas possuem. A primeira característica de um SE é a funcionalidade única, Vahid e Givargis (1999) descrevem esta característica como sistemas que executam um único programa repetidamente. Um exemplo de SE de funcionalidade única é o pager, pois sempre será um pager executando repetidamente a mesma função.

Vahid e Givargis (1999) afirmam que todos os sistemas computacionais possuem restrições de projeto, porém em SE essas restrições são mais rígidas, como custo, tamanho, peso, desempenho, consumo de energia e etc... Todas essas restrições estão ligadas diretamente entre si, o principal desafio é agregar desempenho e consumo de energia ao tamanho, pois nem sempre o tamanho desejado é suficiente para construir um sistema que alcance as especificações desejadas. Esta é a segunda característica a ser analisada em um SE.

Uma terceira característica comum em SE que Vahid e Givargis (1999) descrevem são os SE reativos e de tempo real, neste caso os sistemas devem reagir continuamente as mudanças do ambiente e devem calcular certos resultados em tempo real sem demora, mais especificamente em um tempo determinado. Essa característica é aplicada principalmente no setor industrial, onde sistemas devem realizar tarefas em um tempo determinado, em caso de falha do sistema, uma fábrica inteira pode parar, conseqüentemente causando prejuízos gigantescos.

O principal componente de um SE é o microcontrolador, ele deve conter embutido em si unidade central de processamento (CPU) e outros periféricos, como: memória, conversor

¹ Criar uma definição precisa de tais sistemas de computação embarcados, ou simplesmente sistemas embarcados, não é uma tarefa fácil. Podemos tentar a seguinte definição: Um sistema embarcado é quase qualquer sistema de computação diferente de um computador desktop, laptop ou mainframe. (Tradução nossa).

analógico-digital (AD), conversor digital-analógico (DA), sensores, atuadores e um firmware, tudo em um mesmo chip, os processadores de computadores de propósito geral não possuem estes periféricos no mesmo chip (OSSADA; MARTINS, 2010).

Apesar de os SE possuírem uma arquitetura semelhante aos sistemas computacionais de propósito geral, eles possuem uma larga flexibilidade não só de software, mas também de hardware. SE podem ser implementados em plataformas de propósito geral e também podem ser projetados para hardware específico. Como dito anteriormente os SE possuem sensores e atuadores, estes geralmente não estão presentes em sistemas computacionais de propósito geral (BARROS; CAVALCANTE, 2010).

Como mencionado os SE possuem processadores, porém dependendo de qual finalidade se deseja aplicar um SE, é necessário definir qual tipo de processador será utilizado no projeto e na construção de um SE. A escolha do tipo de processador está ligada diretamente ao comportamento e ao custo do SE, além de influenciar em desempenho, consumo de energia e etc... Para saber qual tipo de processador será escolhido em um projeto, primeiro é necessário conhecer quais os tipos de processadores existentes para SE. Vahid e Givargis (1999) afirmam que existem três tipos de processadores que são: processadores de propósito geral, processadores de propósito único e processadores de aplicação específica.

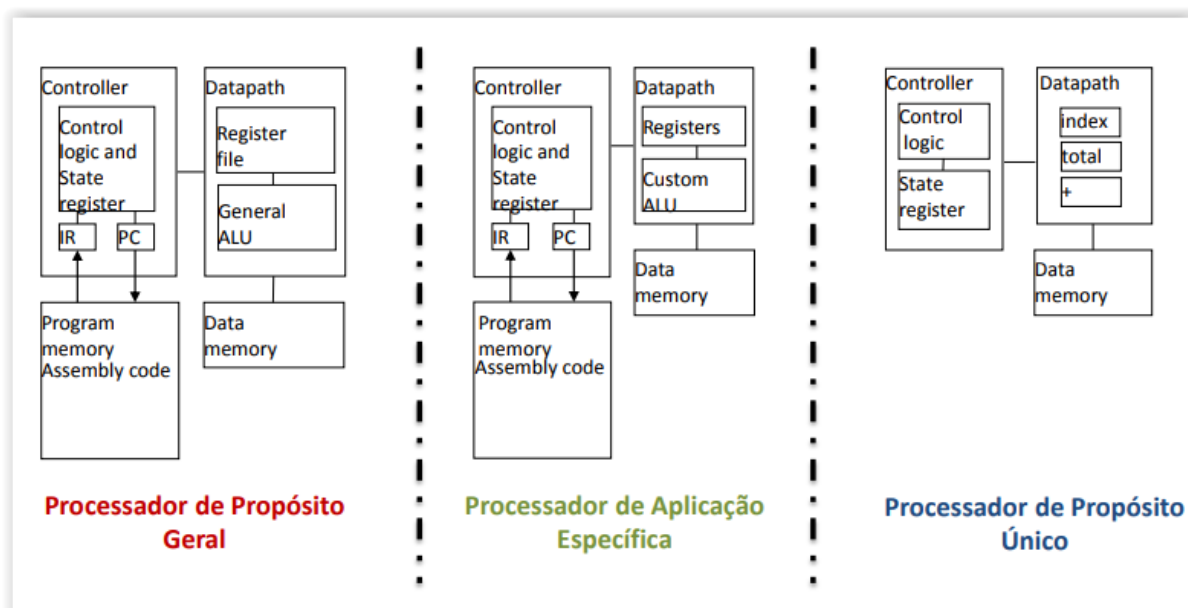
Processadores de propósito geral também são conhecidos como processadores de software, são construídos de forma que sejam adequados a uma variedade de aplicações, com o intuito de alcançar o máximo de vendas de dispositivos. Uma característica desse processador é sua memória de programa, esta memória é destinada a receber a programação através de software e não através de circuito digital, isso torna ele mais flexível, além disso ele possui um datapath que tem como finalidade lidar com uma variedade de cálculos, que tem tipicamente um grande arquivo de registro e uma ou mais unidades de lógica e aritmética (ULA) de uso geral (VAHID; GIVARGIS, 1999).

Vahid e Givargis (1999) afirmam que processadores de propósito único podem ser chamados de processadores de hardware, este processador é um circuito digital desenvolvido especificamente para executar uma tarefa apenas. A exemplo de uma câmera digital todos os componentes que não são um microcontrolador são processadores de propósito único, onde um codificador de imagem executa uma simples tarefa de comprimir e descomprimir frames de vídeo.

Um terceiro tipo de processador descrito por Vahid e Givargis (1999) são os processadores de aplicação específica (application-specific instruction-set processor ou ASIP), o ASIP é desenvolvido para uma classe de aplicações com características em comum, como processadores de sinal digital (Digital-signal processors ou DSPs), os DSPs são uma classe comum de ASIP. De forma geral um ASIP é um processador de aplicação específica para determinadas aplicações, para determinadas tarefas eles possuem maior desempenho, porém seu número de instruções é limitado.

Os processadores citados acima possuem características que os diferenciam uns dos outros, o processador de software ocupa maior espaço físico e sua ULA é de uso geral, o processador de hardware ocupa menor espaço físico e não possui ULA, por fim o processador de aplicação específica possui características semelhantes ao processador de software, porém contém uma ULA personalizada e menor número de instruções, ilustrado na Figura 2.

Figura 2 – Características dos processadores para SE



Fonte: Florencio (2016, online)

Como citado inicialmente os SE estão presentes em uma variedade de dispositivos eletrônicos, na maioria não se percebe que um determinado dispositivo possui um SE. Uma pequena lista de SE onde tem-se dispositivos que possuem SE e dispositivos que são SE é ilustrada na próxima página na Figura 3.

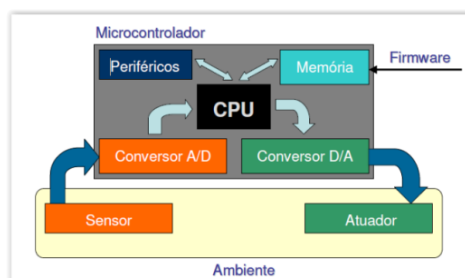
Figura 3 – Lista de Sistemas Embarcados

Freios ABS	Modems			
Cameras auto-foco	Decodificadores MPEG			
Lavadoras automáticas	Cartões de rede			
Briquedos	Roteadores			
Sistemas de Transmissão	Pagers			
Eletrônica de avião	Copiadoras			
Carregadores de bateria	Pontos de venda			
Filmadoras	Video games			
Telefones celulares	Impressoras			
Telefones sem fio	Telefones via satélite			
Sistema de navegação	Scanners			
Cameras digitais	Forno de micro-ondas			
Disk drives	Reconhecedores de voz			
Leitoras de cartão	Sistemas de som			
Instrumentos eletrônicos	Sistemas de teleconferencia			
Controle de Fábricas	Televisores			
Máquinas de Fax	Controladores de temperatura			
Identificadores de digitais	TV set-top boxes			
Sistemas de segurança	Video cassette e DVD's			
Instrumentos medicos	Lavadoras e secadoras			

Fonte: Barros ([entre 2010 e 2014], online)

Como mostrado na Figura 3 um dos SE listados são os telefones celulares, que evoluíram e chegaram na categoria de smartphones, essa categoria é considerada também como um SE, porém, com sua atual capacidade de executar diversas tarefas o torna mais semelhante a um sistema computacional de propósito geral, mas não deixa de ser classificado como um SE. A classificação do smartphone como um SE exige a demonstração da arquitetura do processador de um SE, essa arquitetura pode ser representada de três formas diferentes, devido ao fato de que existem três tipos de processadores de sistemas embarcados, cada um deles já foi descrito anteriormente, como sendo processadores com especificações diferentes e suas aplicações dependem de saber para qual finalidade serão utilizados. Abaixo é ilustrada a arquitetura básica de um processador de SE na Figura 4.

Figura 4 – Arquitetura básica para processadores de SE



Fonte: Rafael ([entre 2014 e 2016], online)

A figura ilustrada na página anterior representa a arquitetura básica de um SE, para demonstrar que um smartphone é um SE algumas características são apontadas, como processador e firmware. Bertoli (2014, p. 12) afirma que, “O smartphone pode ser definido como um telefone móvel com tecnologias avançadas, incluindo processadores semelhantes aos utilizados nos computadores pessoais, comumente chamados de PC ou Desktop”. Os smartphones possuem características como comunicação através da internet e ligação, além de recursos como sistema operacional, câmera e etc... Estas características e recursos se agregam as características de um SE.

Lenovo (2017, online) afirma que “Um smartphone é um telefone móvel ou celular que utiliza de um sistema operacional (SO) e funciona como um pequeno computador”.

Com base nas definições de SE e de smartphones, pode-se definir que um smartphone é um SE composto por um processador de propósito geral e com processadores de propósito único, além de seu firmware. Firmware é um programa que armazena as tarefas básicas a serem realizadas de um SE em um chip, dependendo da aplicação não é reprogramável (VAHID; GIVARGIS, 1999). O firmware de um smartphone é o seu sistema operacional, o diferencial do sistema operacional é que ele pode ser atualizado, porém as funções básicas designadas pelo firmware não são alteradas.

O sistema operacional de um smartphone permite o dispositivo instalar e executar aplicativos, conectar à internet, tirar fotos e gravar filmes e outros recursos (LENOVO, 2017). O principal recurso utilizado de um smartphone são os aplicativos, estes podem ser desde uma simples calculadora até um gerenciador de finanças.

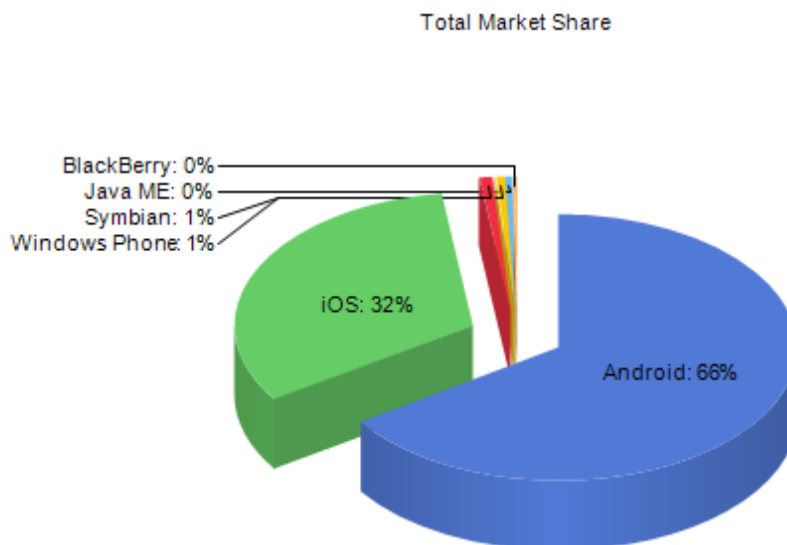
3 APLICATIVOS MOBILE

Os avanços tecnológicos de aparelhos celulares e as evoluções em hardware, ofereceram aos usuários vários recursos que vão além dos recursos básicos, e permitiram o desenvolvimento de sistemas operacionais avançados que possibilitaram o desenvolvimento de aplicativos com mais recursos e serviços (SILVA; SANTOS, 2014).

Com as facilidades oferecidas pelos aplicativos de smartphones, diversas aplicações foram geradas e continuam sendo geradas outras aplicações que solucionem os problemas do usuário. A disputa entre soluções cada vez melhores impulsionou o mercado de desenvolvimento de aplicativos, esse mercado gera ferramentas para todos os tipos de desenvolvimento, sejam elas mais simples ou mais complexas. Porém no desenvolvimento de um aplicativo é necessário escolher as ferramentas adequadas para tal, mas antes de escolher as ferramentas adequadas, primeiro se seleciona qual o paradigma de desenvolvimento para aquela aplicação.

Taurion (2013, online) afirma que, “Basicamente temos três tipos, cada um com suas potencialidades, mas também com restrições e cuidados especiais: os apps nativos, os apps web e os híbridos”. Esses tipos de aplicativos citados, possuem características distintas e a forma na qual são desenvolvidos também são distintas, além de utilizarem ferramentas específicas no desenvolvimento de cada um deles.

Aplicativos nativos são aplicativos construídos especificamente para uma plataforma (SILVA; SANTOS, 2014). Cada plataforma possui características próprias, essas plataformas para smartphones são os sistemas operacionais os quais utilizam. Para o desenvolvedor construir uma aplicação que alcance uma grande quantidade de usuários, é necessário que o mesmo tenha informações de quais são as plataformas mais utilizadas, caso o desenvolvedor construa uma aplicação para uma plataforma que possua uma pequena quantidade de usuários, sua aplicação não será compartilhada em larga escala, conseqüentemente o usuário deverá descontinuar o projeto ou desenvolver novamente a aplicação para outra plataforma. Na página seguinte é ilustrada a situação do mercado de sistemas operacionais para dispositivos móveis na Figura 5.

Figura 5 – Utilização de sistemas operacionais em dispositivos mobile

Fonte: Netmarketshare (2017, online)

Baseando-se na figura acima pode-se concluir que Android e IOS são as plataformas que disputam o mercado de aplicativos móveis. Para a construção de aplicativos nativos nessas plataformas são necessárias ferramentas específicas para cada uma delas. A linguagem de programação Java é utilizada para desenvolver aplicativos Android (DEITEL; DEITEL; DEITEL, 2015). A linguagem de programação utilizada para desenvolver aplicativos para IOS, Mac, Apple TV, Apple Watch é chamada de Swift, foi desenvolvida propriamente pela Apple com o intuito de dar mais liberdade e facilidade de código aos desenvolvedores (APPLE, 2017).

Os aplicativos nativos permitem ao usuário uma navegação mais imersiva e cheia de recursos que podem ser acessados pelo sistema operacional, recursos como GPS e câmera. A interface gráfica de um aplicativo nativo utiliza os componentes gráficos do sistema operacional. Aplicativos desenvolvidos nesse paradigma possuem melhor desempenho, um exemplo de aplicativo nativo é o Whatsapp, ele utiliza recursos de câmera, GPS, gravador de voz e microfone. Seu ótimo desempenho está ligado diretamente ao fato de ser um aplicativo desenvolvido no paradigma nativo.

Como esses aplicativos funcionam apenas na plataforma que foram desenvolvidos eles levam maior tempo para serem construídos e maior custo, pois disponibilizar em outras

plataformas levará maior tempo e profissionais capacitados para cada plataforma desejada (SILVA; SANTOS, 2014).

Com o alto custo para desenvolver aplicativos nativos para várias plataformas, surgiram ferramentas para alcançar várias plataformas através de uma única forma de desenvolvimento. Essas soluções já mencionadas antes são aplicativos web e aplicativos híbridos. Silva e Santos (2014) consideram que ambos são aplicativos web, o primeiro chamado web puro e o segundo chamado web híbrido.

O aplicativo web puro utiliza tecnologias Web Padrão como HTML, CSS e JavaScript, ele possui funcionamento semelhante ao nativo, porém não permite a navegação imersiva e utilização de todos os recursos do dispositivo. Este modelo de aplicativo é executado diretamente no navegador do celular, atualmente os smartphones possuem navegadores que suportam essas tecnologias (SILVA; SANTOS 2014). Os aplicativos web puro tem como principal característica seu funcionamento completamente no navegador do celular, outra característica é que o sistema operacional não tem controle sobre o conteúdo da aplicação, as alterações feitas na aplicação são disponibilizadas automaticamente para os usuários sem a intervenção do sistema operacional.

Outro paradigma de desenvolvimento para aplicativos móveis é o paradigma de aplicativos híbridos ou segundo Silva e Santos (2014) aplicativos web híbridos, que são caracterizados como uma variação de aplicativos web.

Aplicativos híbridos são similares aos web puro, os híbridos utilizam também tecnologias padrões da Web Padrão, HTML, CSS e JavaScript, a diferença é que estes utilizam recursos nativos do dispositivo através de uma API² JavaScript comum (SILVA; SANTOS, 2014).

Segundo Hartmann et al. (2011 *apud* SILVA; SANTOS, 2014, p. 165), “um aplicativo WEB Híbrido consiste em um aplicativo Web incorporado em um aplicativo nativo que fornece uma ponte para o sistema operacional e os serviços nativos do dispositivo”. Para o desenvolvimento de aplicativos híbridos existem várias ferramentas, porém será destacado o

² API em português significa Interface de programação de aplicativos, ela é um conjunto de rotinas e padrões de programação para acesso a um aplicativo ou plataforma baseado na Web (CIRIACO, 2009).

Ionic que é um framework³ open source. O Ionic permite que desenvolvedores web construam seus aplicativos com um único código base, e distribuam seu aplicativo a partir desse código base nas maiores lojas de aplicativos juntamente com aplicativos nativos (IONIC, 2017). Atualmente o Ionic encontra-se na versão 3, essa versão utiliza como base as seguintes tecnologias: Angular, JavaScript, TypeScript e HTML5.

Ele possui comandos CLI, que são linhas de comando de interface, esses comandos permitem gerar componentes para o desenvolvedor. Em sua versão atual o Ionic permite a utilização de TypeScript. Segundo Alves (2017) “O TypeScript é uma extensão do JavaScript criada para facilitar o desenvolvimento de aplicações”. Por sua característica de utilizar recursos nativos do dispositivo, os componentes gráficos tornam-se semelhantes aos componentes gráficos nativos da plataforma.

Para desenvolver um aplicativo, o desenvolvedor deve escolher entre os três paradigmas citados, a escolha do paradigma reflete diretamente em seu custo de desenvolvimento, manutenção e outras características de uma aplicação de software. Para escolher qual paradigma utilizar o desenvolvedor deve analisar situações como custo do aplicativo, quais plataformas vão utilizar o aplicativo? O aplicativo necessita de recursos nativos da plataforma? O aplicativo será publicado em lojas de aplicativos? Deverá possuir alto, médio ou baixo desempenho? Quais tecnologias são utilizadas no paradigma escolhido?

Na página seguinte são ilustradas as vantagens e desvantagens de cada paradigma na Figura 6. A figura demonstra os principais recursos que fornecem vantagens e desvantagens para cada paradigma, porém para a escolha deve-se conhecer ainda mais recursos oferecidos por cada paradigma, assim conhecendo completamente cada um, o desenvolvedor terá uma visão ampla daquilo que cada um oferece e dentre os três qual atenderá todas os requisitos de seu projeto, podendo então, determinar qual paradigma será utilizado no desenvolvimento do projeto. Para este projeto o paradigma escolhido foi o híbrido, pois ele atende aos requisitos do projeto e alcança o maior número de usuários, ele permite desenvolver uma única aplicação para várias plataformas sem a necessidade de reescrever ou alterar código para outra plataforma.

³ Framework é um conjunto de bibliotecas, componentes de software e diretrizes de arquitetura que permitem ao desenvolvedor uma série de ferramentas para criar um aplicativo completo para qualquer plataforma (Hartmann et al. (2011 *apud* SILVA; SANTOS, 2014, p. 164).

Figura 6 – Diferenças entre paradigmas de desenvolvimento de aplicativos

	Vantagens	Desvantagens
Apps Nativos	<ul style="list-style-type: none"> • Acesso às funcionalidades do aparelho (Câmera, GPS) • Apps baixados diretamente pelas lojas • Sistema mais específico • Funcionamento offline 	<ul style="list-style-type: none"> • Custo bem maior • Aprovação da loja / linguagem específica • Criação em diferentes plataformas
Web app	<ul style="list-style-type: none"> • Custo mais baixo • Página da web • Acessado de qualquer browser / Funciona para Todas as plataformas 	<ul style="list-style-type: none"> • Funcionamento apenas online • Não publica nas lojas • Não utilizam as funcionalidades do aparelho
Apps Híbridos	<ul style="list-style-type: none"> • Funcionalidades semelhante aos nativos • Custo mais baixo Funciona para todas as plataformas • Apps baixados diretamente pelas lojas • Opção mais barata para manter o app nas lojas 	<ul style="list-style-type: none"> • Performance e usabilidade um pouco inferior ao app nativo • Design mais restrito

Fonte: Ramires (2017, online)

4 TRABALHOS RELACIONADOS

Atualmente existem vários sistemas de gerenciamento e aplicativos com diferentes finalidades relacionados a autoescolas. Porém, aplicativos que permitam ao aluno agendar diretamente suas aulas não foram encontrados em lojas de aplicativos. Pode-se citar três soluções que existem no mercado relacionadas a autoescolas, a primeira solução é o aplicativo Desempenho de Aulas e Simulados Detran RJ, este aplicativo permite o aluno consultar diretamente o sistema do Departamento de Trânsito do Rio de Janeiro (DETRAN-RJ) as aulas teóricas e práticas realizadas na autoescola e a realização de simulados diretamente integrados ao DETRAN-RJ, a Figura 7 demonstra a tela com perfil do aluno e as opções disponíveis no aplicativo.

Figura 7 – Tela de perfil de aluno



Fonte: Autoescola.IO (2017, online)

A segunda solução é o aplicativo Autoescola APP, esta aplicação permite ao aluno a visualização de informações sobre datas e horários de aulas, além de resultados de provas, no aplicativo o aluno pode acompanhar o progresso de aulas de legislação com informações de data e horário, além do número total de aulas. A Figura 8 demonstra a tela de progresso do aluno.

Figura 8 – Tela de progresso do aluno



Fonte: Tadeu (2017, online)

A terceira solução encontrada é mais completa em relação as duas anteriores, é o sistema InoveCFC, esta solução possui ferramentas para gerar relatórios de aulas, consumo de veículos, realizar orçamentos, além de outras facilidades oferecidas, porém o sistema não permite que o aluno agende aulas através dele, o aluno pode apenas acompanhar seu

progresso de aula além de não ser um aplicativo, mas sim uma plataforma web. Abaixo a Figura 9 representa a tela de acompanhamento de aulas práticas. Nesta tela o aluno pode visualizar o horário da aula, data, nome do instrutor, veículo e o histórico das aulas anteriores, podendo assim visualizar a frequência.

Figura 9 – Histórico de aulas práticas



Qtd	Cat	Data	Hora	Instrutor	Veículo	Presença
21	B	05/07/2010	12:00	CARLOS RAMIRO SANTOS	FIAT - PALIO - AAA-0003	✓
22	B	11/01/2011	12:00	JULIANO BATISTA	FIAT - PALIO - AAA-0003	✗
23	B	12/01/2011	12:00	JULIANO BATISTA	FIAT - PALIO - AAA-0003	✓
24	B	13/01/2011	12:00	JULIANO BATISTA	FIAT - PALIO - AAA-0003	✗
25	B	17/01/2011	08:00	JULIANO BATISTA	FIAT - PALIO - AAA-0003	🕒
26	B	18/01/2011	08:00	JULIANO BATISTA	FIAT - PALIO - AAA-0003	🕒
27	B	19/01/2011	15:00	JULIANO BATISTA	FIAT - PALIO - AAA-0003	🕒
28	B	21/01/2011	15:00	JULIANO BATISTA	FIAT - PALIO - AAA-0003	🕒

Fonte: InoveCFC (2017, online)

A aplicação proposta neste trabalho, se comparada com as aplicações mostradas anteriormente, possui como diferencial a funcionalidade oferecida ao aluno para que ele realize agendamento de suas aulas diretamente pelo aplicativo de forma independente de intervenções humanas.

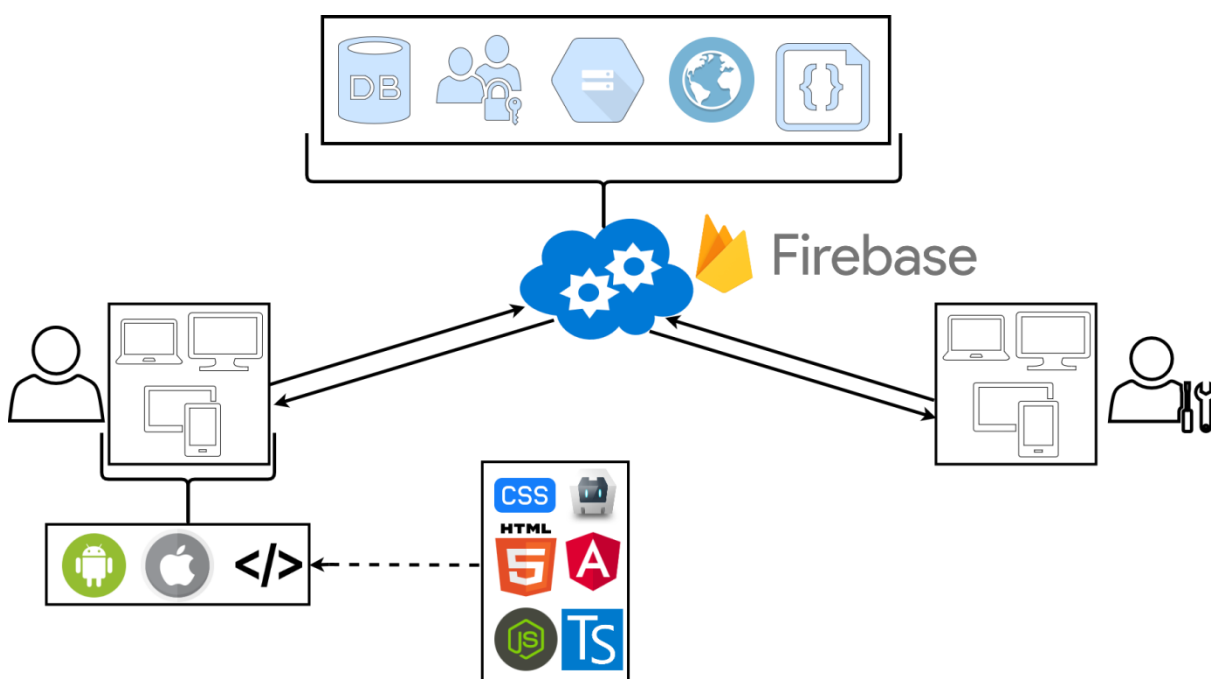
5 MATERIAIS E MÉTODOS

5.1 FUNCIONAMENTO

O funcionamento da aplicação inicia em sua conexão com o serviço de nuvem do Google firebase, onde é conectado aos serviços de autenticação e banco de dados e hospedagem, em seguida o usuário deve realizar seu login, caso não possua uma conta o mesmo pode realizar um cadastro simples e rápido, utilizando apenas seu e-mail e criando uma senha, após a realização do cadastro o usuário receberá um link de validação na sua caixa de e-mail. A aplicação possui dois tipos de usuários que são aluno e administrador, que possuem funcionalidades de agendar horários de aulas e criar horários de aula, respectivamente. As informações de usuários como e-mail, horários agendados, disponíveis e criados são armazenadas no banco de dados do firebase na nuvem, estes dados são sincronizados para todos os usuários quando alterados.

O diagrama abaixo representa o funcionamento da aplicação integrada ao serviço de nuvem do Google firebase. Mostra-se a seguir a Figura 10.

Figura 10 – Diagrama de funcionamento



Fonte: Autor (2017)

5.2 LISTA DE FERRAMENTAS

Abaixo é representada a lista de ferramentas utilizadas no desenvolvimento da aplicação na Figura 11, em seguida tem-se uma lista com a descrição das ferramentas e para quais finalidades do projeto foram utilizadas. A lista tem uma breve descrição de cada ferramenta e para qual finalidade cada ferramenta foi utilizada.

Figura 11 – Tecnologias utilizadas



Fonte: Autor (2017)

As ferramentas utilizadas na camada de engenharia e projeto, foram selecionadas para uso pelo fato de conhecer o funcionamento destas, pois foram utilizadas durante a graduação. Na camada de desenvolvimento híbrido as seguintes ferramentas Node, Apache Cordova, Angular, HTML 5, JavaScript, TypeScript, CSS, Ionic Creator, Ionic DevApp foram selecionadas porque são ferramentas obrigatórias para o desenvolvimento de aplicações através do Ionic.

Ainda na camada de desenvolvimento híbrido foi utilizada a ferramenta do Google Firebase, esta ferramenta foi selecionada devido o insucesso de conectar a aplicação a outras ferramentas de banco de dados. Em seguida a utilização do Visual Studio Code como aplicação para codificação do projeto, esta ferramenta foi selecionada por causa de ser própria para desenvolvimento de aplicações WEB e possuir funcionalidades de código como completar código baseado nas primeiras letras digitadas pelo desenvolvedor.

Por fim foram utilizadas as ferramentas para geração da aplicação para a plataforma Android, que são o Android SDK e o Gradle, tais ferramentas são obrigatórias para gerar a aplicação, pois possuem todos complementos necessários para a aplicação se comunicar com a plataforma e o hardware do dispositivo.

Abaixo é representada a lista de ferramentas com os detalhes de cada uma para suas finalidades sendo dividida em duas camadas, igual como foi demonstrado na Figura 11 na página anterior.

5.2.1 Engenharia e projeto

5.2.1.1 Astah Community

Ferramenta de modelagem UML (Unified Modeling Language), possui outras versões, a versão Community foi utilizada pelo fato de ser uma versão grátis, esta ferramenta foi utilizada na modelagem dos diagramas de caso de uso e pacotes (ASTAH, 2017).

5.2.1.2 Balsamiq Mockups

Aplicação desenvolvida pela companhia Balsamiq, essa aplicação é utilizada para desenvolver protótipos ou modelos, como telas de um sistema desktop, sistemas e/ou páginas web e modelagem de aplicações mobile (BALSAMIQ, 2017). Na modelagem de telas na etapa inicial do projeto, esta aplicação foi utilizada para representar um modelo de como seriam as possíveis telas do aplicativo.

5.2.1.3 Draw.io

Aplicação WEB gratuita, acessada a partir do navegador que permite a criação de diagramas, fluxogramas, esquemas, infográficos e etc. (DRAW.IO, 2017). No projeto a aplicação foi utilizada para o desenvolvimento do diagrama que demonstra o funcionamento da aplicação.

5.2.2 Desenvolvimento híbrido

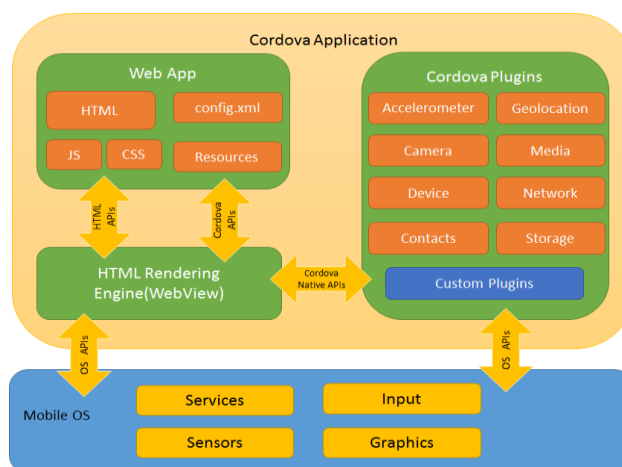
5.2.2.1 Node.js

Ambiente de execução multiplataforma em JavaScript que permite aos desenvolvedores produzirem aplicações de rede escaláveis (NODEJS, 2017). No projeto o Node foi utilizado para utilização de comandos como por exemplo o comando de instalação do Cordova, além de reconhecer comandos nativos do Ionic.

5.2.2.2 Apache Cordova

Framework para desenvolvimento de aplicações mobile que utiliza tecnologias web padrão (CORDOVA, 2017). Aplicações Cordova se comunicam com as aplicações mobile através de API's do sistema operacional do dispositivo. O Cordova possui plug-ins e API's nativos que também se comunicam com a aplicação mobile através de API's do sistema operacional do dispositivo. O Ionic exige a instalação do Cordova para utilização de plug-ins e API's que facilitam e agilizam o processo de desenvolvimento. Mostra-se a seguir a Figura 12 que demonstra a arquitetura de uma aplicação baseada em Cordova.

Figura 12 – Arquitetura de aplicação Cordova



Fonte: Cordova (2017, online)

5.2.2.3 Ionic

Framework de código aberto para construção de aplicações móveis utilizando tecnologias web como Angular, JavaScript, TypeScript e HTML5, o principal recurso oferecido aos desenvolvedores é a geração de aplicações híbridas que trabalham de forma similar as nativas, estas aplicações podem ser geradas para diferentes plataformas sem a necessidade de reescrever código (IONIC, 2017).

5.2.2.4 Angular

Angular é um framework mobile e desktop para desenvolvimento do front-end de aplicações web (ANGULAR, 2017). O framework foi utilizado no projeto para realizar a integração da aplicação web com os serviços do firebase. O Angular trabalha com tecnologias

web, sendo assim é um componente bastante importante para o desenvolvimento de aplicações com Ionic.

5.2.2.5 HTML 5

É uma nova versão da linguagem HTML (HyperText Markup Language), que é uma linguagem de marcação de hipertexto. A nova versão traz melhorias como descrição mais completa do conteúdo, utilização de temas mais sofisticados e outras melhorias (MDN, 2017). Na criação de formulários, tipos de entrada e manipulação dos dados informados nas páginas do aplicativo para os métodos TypeScript foi utilizado o HTML 5.

5.2.2.6 JavaScript

É uma linguagem interpretada, conhecida também como a linguagem de scripts para páginas web, também é utilizada em ambientes sem navegadores (MDN, 2017). Sua utilização no projeto foi na criação de funcionalidades para manipular dados e para funções que transmitem dados para o firebase.

5.2.2.7 TypeScript

Almeida (2017) afirma que a linguagem TypeScript não é uma linguagem nova, mas sim um superset do JavaScript ou uma linguagem que compartilha recursos do JavaScript. O TypeScript é base para desenvolvimento de aplicações através do Ionic, ele coleta os dados da parte HTML da aplicação, realiza as operações e devolve ao HTML. Porém os códigos são compilados em JavaScript pois é a linguagem que o navegador consegue interpretar. Para o projeto a linguagem foi essencial e obrigatória para desenvolvimento das funções de manipulações com os dados inseridos nas telas da aplicação.

5.2.2.8 CSS

A sigla significa em português: folhas de estilos em cascata, é uma linguagem responsável pela descrição e representação de elementos mostrados em tela, no papel, e outras mídias (MDN, 2017). Esta ferramenta é utilizada para representação de componentes das telas.

5.2.2.9 Google Firebase

Plataforma dedicada a construção de aplicações móveis nativas, híbridas e web, que oferece diversas funcionalidades aos desenvolvedores como banco de dados em nuvem,

serviço de autenticação de usuários, hospedagem, armazenamento, integração com serviços exclusivos do Google como analytics e etc. (FIREBASE, 2017). O serviço de autenticação permite ao usuário cadastrar e logar com e-mail, conta do Google, Facebook, Twitter e GitHub. O serviço de banco de dados armazena as informações através de JSON, este será explanado na engenharia de software do projeto, o serviço permite configurar regras de acesso para o banco, como ler e gravar dados apenas com usuários autenticados.

Outro serviço oferecido é o de hospedagem, este permite que a aplicação desenvolvida envie e receba os dados diretamente dos servidores do firebase, sem necessidade de contratar um serviço de hospedagem externo, sendo assim hospedando a aplicação de forma gratuita. O firebase oferece ainda o serviço de armazenamento, este permite aos desenvolvedores armazenar imagens, áudio, vídeo e outros conteúdos gerados pelo usuário. Por fim o serviço de funções, este permite que o desenvolvedor escreva funções e as armazene nos servidores da nuvem do Google, o serviço executa código back-end automaticamente em resposta a eventos acionados pelos recursos do firebase.

Neste projeto foram utilizados os serviços de autenticação para cadastro e login de usuários, banco de dados para armazenar os horários de aula criados e agendados, por fim o serviço de hospedagem foi utilizado para hospedar a aplicação permitindo que a mesma funcione online.

5.2.2.10 Ionic Creator

Ferramenta drag-&-drop de prototipação para criar a interface gráfica do aplicativo que será desenvolvido pelo framework Ionic (IONIC CREATOR, 2017). A ferramenta foi utilizada para modelar a interface gráfica do projeto.

5.2.2.11 Visual Studio Code

Editor de texto multiplataforma grátis para o desenvolvimento de aplicações web (VISUAL STUDIO CODE, 2017). Ferramenta utilizada para o desenvolvimento do código do projeto.

5.2.2.12 Ionic DevApp

Aplicação mobile grátis que permite executar aplicações baseadas em Ionic diretamente em um dispositivo Android ou IOS, exigindo apenas que o desenvolvedor

execute a aplicação em um computador e que o dispositivo móvel esteja conectado a mesma rede do computador para acessar a aplicação (IONIC, 2017).

5.2.2.13 Android SDK

O Android sdk é um kit de desenvolvimento de software para Android, ele permite o desenvolvimento de aplicações nativas para a plataforma Android, ele inclui exemplos de código-fonte, emuladores e bibliotecas necessárias para criar o aplicativo (ANDROID, 2017). O Android sdk foi utilizado no projeto para adicionar a plataforma Android no Ionic, pois sem o sdk é impossível gerar a aplicação para a plataforma Android.

5.2.2.14 Gradle

Gradle é um sistema de automação de compilação (GRADLE, 2017). Durante o processo de construção para gerar o aplicativo vários processos secundários são executados, como empacotamento, conversão de interfaces e outros processos, o Gradle automatiza todos esses processos fazendo com que a compilação do aplicativo seja mais leve, evitando a geração de arquivos desnecessários e entregando mais qualidade ao aplicativo. Para o processo de compilação do aplicativo para a plataforma Android é obrigatória a utilização do Gradle.

5.3 ENGENHARIA DE SOFTWARE

5.3.1 Modelagem do banco de dados

O processo de modelagem de banco de dados de uma aplicação é demonstrado a partir de um modelo conceitual representado por um diagrama de entidade relacionamento (DEVMEDIA, 2017). Porém para este projeto não será utilizada a modelagem conceitual devido ao banco de dados do firebase trabalhar com forma de armazenamento em JSON (JavaScript Object Notation - Notação de Objetos JavaScript) (JSON, 2017). O banco de dados do firebase funciona como uma árvore hospedada na nuvem (FIREBASE, 2017). Um exemplo de como funciona um JSON seria um aplicativo de bate-papo que os usuários armazenam um perfil básico e uma lista de contatos. Como demonstrado na Figura 13 na próxima página.

Figura 13 – Exemplo de armazenamento de dados em JSON

```
{
  "users": {
    "alovelace": {
      "name": "Ada Lovelace",
      "contacts": { "ghopper": true },
    },
    "ghopper": { ... },
    "eclarke": { ... }
  }
}
```

Fonte: Firebase (2017, online)

Neste projeto a utilização do JSON tem como finalidade armazenar as informações sobre os horários de aulas criados e disponibilizados para agendamento. Abaixo pode-se visualizar uma possível forma de armazenar os dados deste projeto.

Figura 14 – Modelo JSON para armazenar horário de aula

```
{
  "Horarios" : {
    "Data" : "11/11/2017",
    "Hora_Fim" : "16:40",
    "Hora_Inicio" : "15:00",
    "Status" : false,
    "Tipo" : "Carro"
  },
}
```

Fonte: Autor (2017)

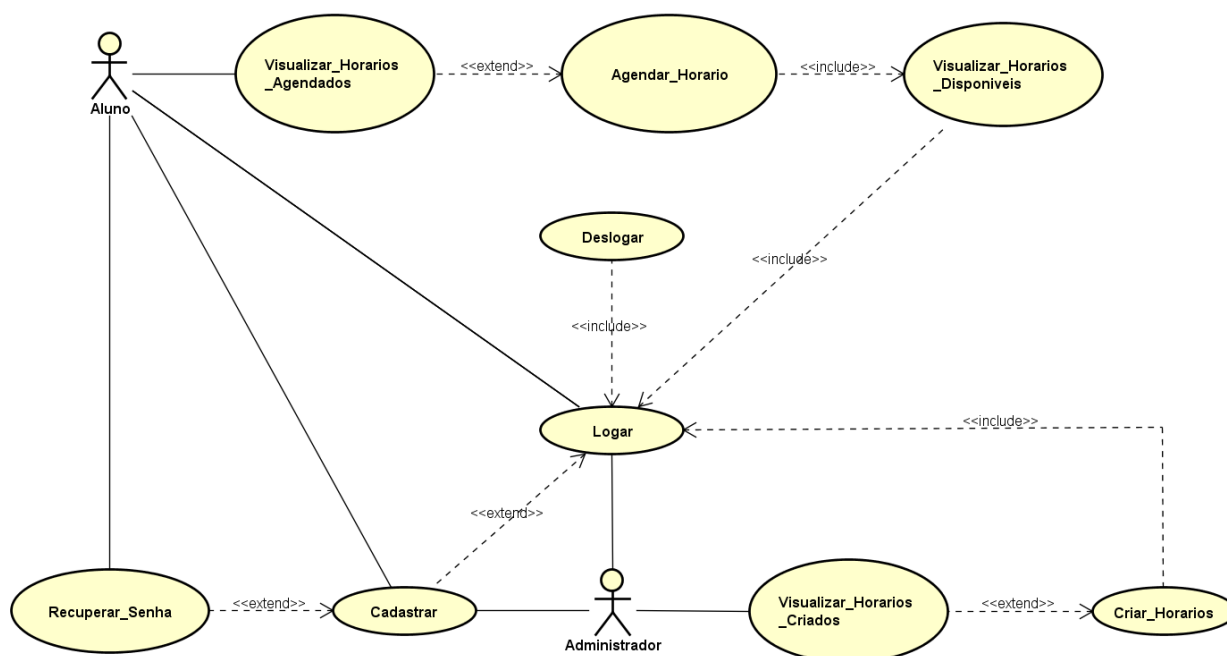
No código cima tem-se um objeto com nome "Horarios", o qual possui os dados de um horário de aula. Para evitar erros de codificação os nomes das variáveis não possuem acentos nem espaços. As informações após os dois pontos são os valores de suas respectivas variáveis.

5.3.2 Diagrama de caso de uso

Para descrição de funcionalidade proposta pelo sistema, foi utilizada a modelagem UML, a qual é utilizada antes da codificação do software (UML, 2017). A linguagem UML é utilizada como padrão internacional em modelagem de diagramas no processo de desenvolvimento de software. Para o desenvolvimento da aplicação foram definidas suas funcionalidades e modeladas conforme a Figura 15 do diagrama de caso de uso.

O diagrama abaixo representa funcionalidades para o desenvolvimento da aplicação. A funcionalidade de visualização de horários criados, realizada pelo administrador só pode ser executada se o usuário estiver conectado a aplicação com o perfil de administrador e se o mesmo tiver realizado a criação de pelo menos um horário, caso contrário a visualização de horários criados será vazia. Para a funcionalidade de visualização de horários agendados é obrigatório que o usuário esteja conectado a aplicação com o perfil de aluno, além de ter realizado pelo menos um agendamento que também exige autenticação de perfil de aluno, caso não tenha realizado nenhum agendamento a visualização de horários será vazia. Diagrama de caso de uso demonstrado na Figura 15 abaixo.

Figura 15 – Diagrama de caso de uso



Fonte: Autor (2017)

5.3.3 Diagrama de pacotes

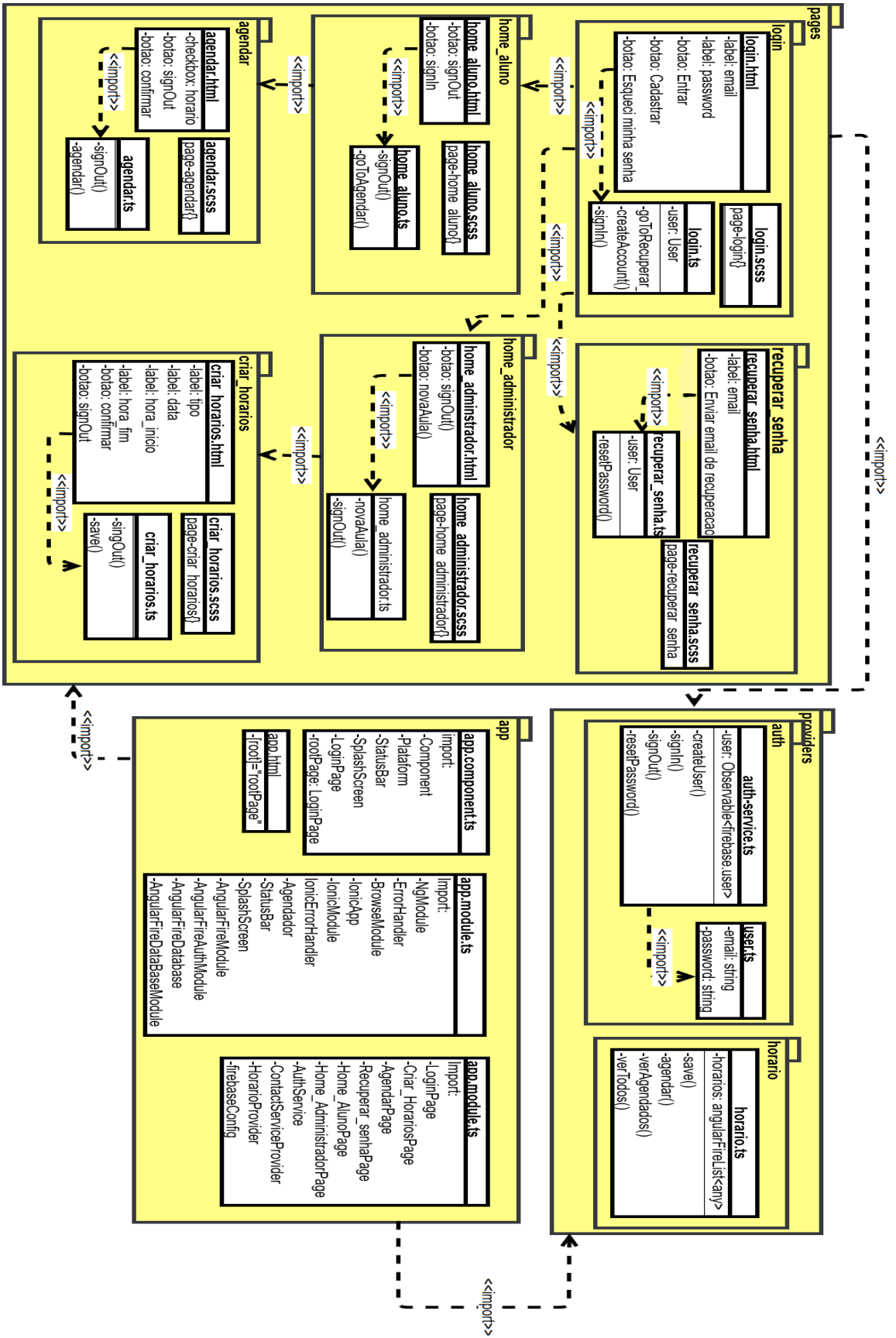
Para descrever como está dividida a estrutura do código do aplicativo, utiliza-se o diagrama de pacotes, este diagrama demonstra quais pacotes o código possui, o aplicativo deste projeto trabalha basicamente com três pacotes que são, app, providers e pages.

O pacote de pages contém outros pacotes que podem ser nomeados de pacotes internos, esses pacotes contém informações das telas do aplicativo e as funcionalidades dessas telas. Porém para que as páginas executem suas funções é necessário que o pacote pages importe informações do pacote providers. O pacote providers contém dois pacotes internos chamados auth e horário, para executar funções de autenticação e funções de agendamento respectivamente.

O terceiro pacote é o pacote app, este pacote contém todas as importações de páginas, módulos de frameworks como Angular, Ionic e importações do conjunto de ferramentas do Google firebase. É neste pacote que os serviços de autenticação, banco de dados e hospedagem da aplicação está localizado.

Mostra-se na página seguinte a Figura 16 que ilustra o diagrama de pacotes.

Figura 16 – Diagrama de pacotes



Fonte: Autor (2017)

5.4 FUNCIONALIDADES

No aplicativo cada funcionalidade depende do perfil de usuário, a aplicação possui dois perfis de usuários que são aluno e administrador, tais perfis oferecem diferentes funcionalidades, demonstra-se abaixo o detalhamento de cada perfil de usuário e suas funcionalidades suportadas de acordo com o perfil.

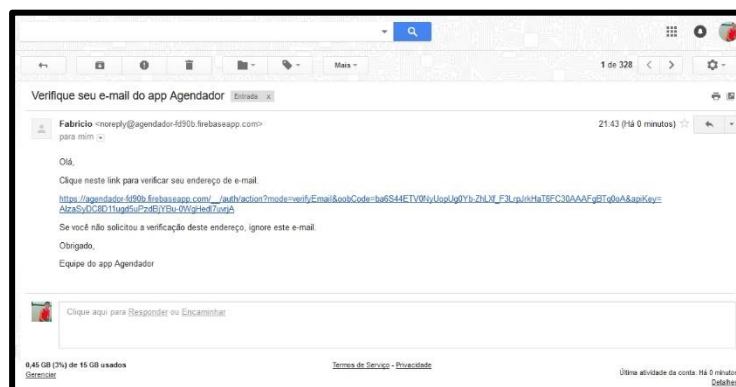
5.4.1 Aluno

O perfil de aluno da aplicação, permite que o usuário autenticado com esse perfil visualize horários disponíveis para agendamento, agende seus horários, visualize os horários agendados. A aplicação ainda oferece um serviço de recuperação de senha através de e-mail. Verifica-se abaixo as funcionalidades citadas e suas respectivas telas, comparadas as telas modeladas no início do projeto.

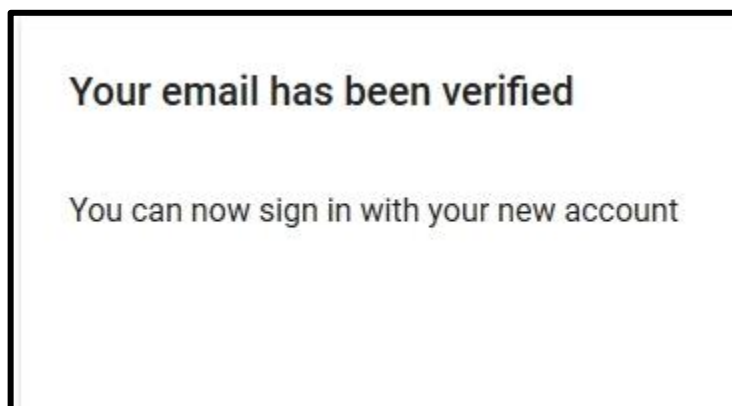
5.4.1.1 Cadastrar

Em seu primeiro acesso o aluno deverá realizar um cadastro, que solicita um e-mail e criação de uma senha de acesso que deve conter no mínimo seis caracteres. Após realização do cadastro o aluno é automaticamente conectado ao sistema e recebe uma mensagem com link de confirmação de cadastro em seu e-mail. O procedimento de cadastro é realizado na mesma tela de login, utilizando os mesmos formulários sem a necessidade de redirecionamento de tela, apenas inserindo as informações e clicando no botão cadastrar. Mostra-se a seguir a tela com recebimento do e-mail de confirmação e a tela após a realização da confirmação de e-mail.

Figura 17 – E-mail de confirmação de cadastro



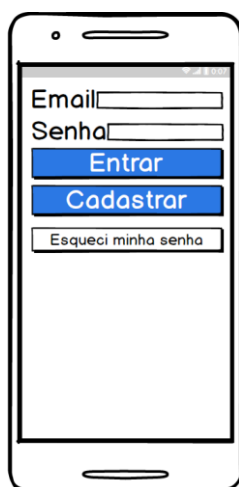
Fonte: Gmail (2017, online)

Figura 18 – Tela de confirmação de e-mail

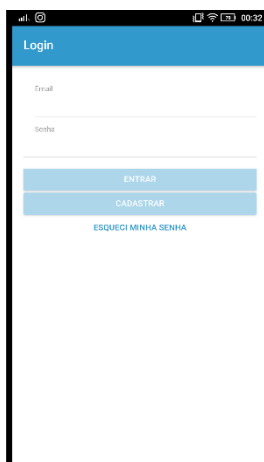
Fonte: Firebase (2017, online)

5.4.1.2 Entrar

Para que o usuário se conecte ao sistema é necessário um cadastro no primeiro acesso como foi citado anteriormente, o usuário necessita se cadastrar apenas uma vez após isso ele poderá se conectar ao sistema quantas vezes desejar. A tela de login é a mesma tela que permite o cadastro como já foi citado também anteriormente, a diferença é que para o usuário realizar login ele deve inserir seu e-mail e senha nos formulários e clicar no botão de entrar, feito isso o usuário realizará login. Mostra-se a seguir a comparação da tela projetada com a tela construída.

Figura 19 – Tela de login projetada

Fonte: Autor (2017)

Figura 20 – Tela de login construída

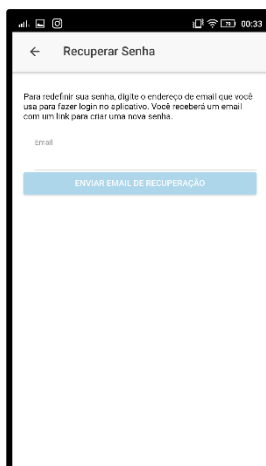
Fonte: Autor (2017)

5.4.1.3 Recuperar senha

Em caso de perda de senha o usuário tem como recurso oferecido pelo aplicativo a recuperação de senha, para isso o usuário deve clicar no botão esqueci minha senha que está localizado na tela inicial do aplicativo, após o click o usuário será redirecionado para a tela de recuperação de senha, nesta tela o usuário deverá informar o seu e-mail cadastrado no aplicativo e clicar no botão enviar e-mail de recuperação. Após a solicitação de recuperação de senha o usuário receberá uma mensagem em seu e-mail com um link para que ele insira uma nova senha. Representa-se abaixo a comparação entre as telas de recuperação de senha.

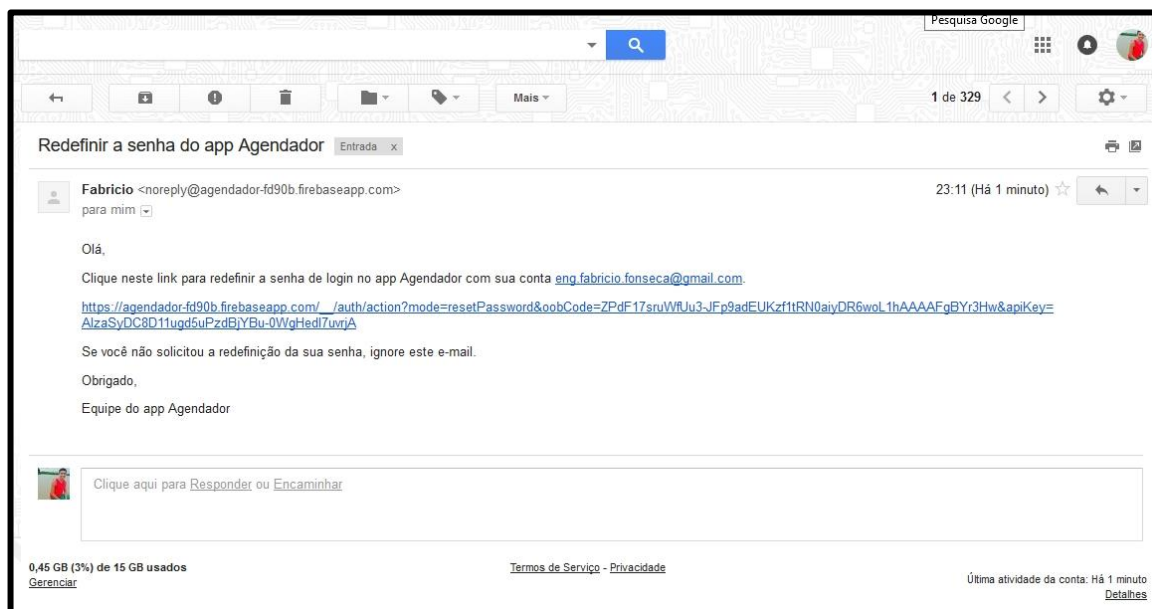
Figura 21 – Tela de recuperação de senha projetada

Fonte: Autor (2017)

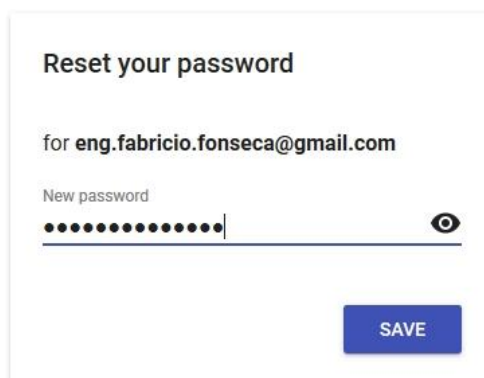
Figura 22 – Tela de recuperação de senha construída

Fonte: Autor (2017)

O processo de recuperação de senha envia um e-mail para o usuário com um link, que redireciona o usuário para uma página, que permite o mesmo criar uma nova senha para sua conta. Verifica-se a seguir as imagens que ilustram o e-mail de recebido e a tela de troca de senha.

Figura 23 – E-mail de redefinição de senha

Fonte: Gmail (2017, online)

Figura 24 – Tela de redefinição de senha

Reset your password

for eng.fabricio.fonseca@gmail.com

New password

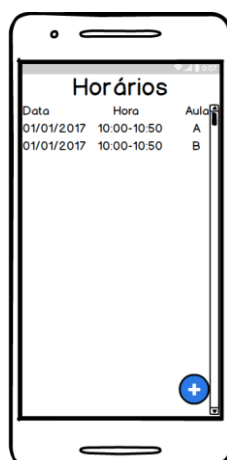
.....|

SAVE

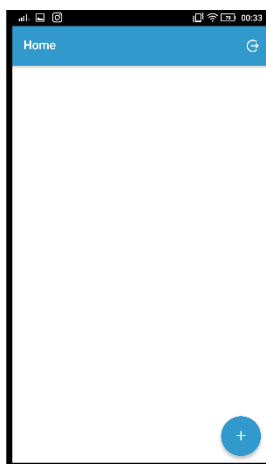
Fonte: Firebase (2017, online)

5.4.1.4 Visualizar horários agendados

Após a realização do login o aluno será redirecionado para sua página inicial, nesta página o aluno poderá visualizar os horários que ele agendou, desde que tenha realizado pelo menos um agendamento, caso ainda não o tenha feito ele visualizará a lista de horários agendados em branco. Independentemente de ter realizado um agendamento ou não, o usuário também contará nessa tela com um botão de adicionar no canto inferior direito do seu dispositivo, após o click desse botão o aluno será redirecionado para a página de horários disponíveis. Abaixo pode-se visualizar a comparação entre a tela modelada e a tela construída. O usuário também contará com um botão de logout no canto superior direito do dispositivo, com apenas um click neste botão o usuário terá sua conta deslogada.

Figura 25 – Tela inicial de aluno projetada

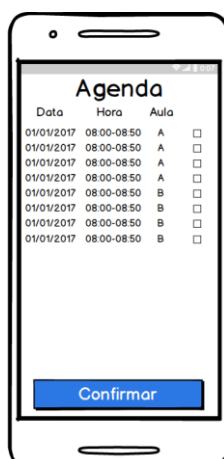
Fonte: Autor (2017)

Figura 26 – Tela inicial de aluno construída

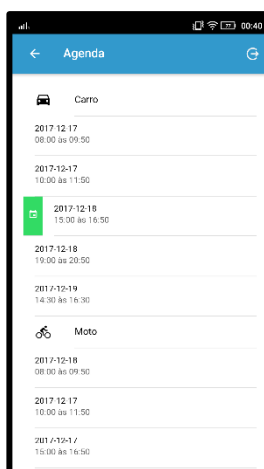
Fonte: Autor (2017)

5.4.1.5 Visualizar horários disponíveis

Para visualizar os horários disponíveis o aluno deverá efetuar login, em seguida clicar no botão adicionar, e então o aluno será redirecionado para a tela de agendamento, onde ele poderá visualizar os horários disponíveis. Demonstra-se a seguir a tela de visualização de horários disponíveis comparada a tela modelada.

Figura 27 – Tela de agendamento projetada

Fonte: Autor (2017)

Figura 28 – Tela de agendamento construída

Fonte: Autor (2017)

5.4.1.6 Agendar horários

A principal funcionalidade do aplicativo é a realização do agendamento, para isso aluno deve seguir os seguintes passos, efetuar login, clicar no botão adicionar para visualizar os horários disponíveis, quando aluno visualizar os horários ele deverá marcar os horários arrastando o horário desejado da esquerda para a direita, após arrastar o usuário deve clicar no botão verde para concluir o agendamento. Caso o usuário não clique no botão o agendamento não será realizado. A mesma tela de visualização de horários é usada para o agendamento.

5.4.2 Administrador

O perfil de administrador da aplicação permite ao usuário entrar no sistema, visualizar horários criados e criar horários. Abaixo serão apresentadas estas funções detalhadas e suas telas de uso em fase de modelagem e versões finais.

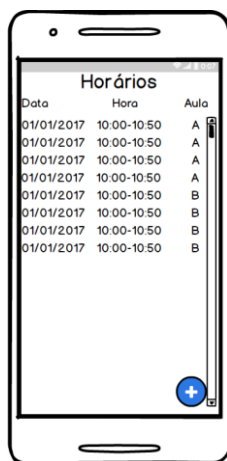
5.4.2.1 Entrar

A tela que permite ao usuário se conectar como administrador é a mesma tela para conexão do perfil de aluno, o usuário é redirecionado para o ambiente de administrador através do e-mail configurado no código fonte da aplicação. O administrador realiza login igualmente ao usuário aluno, porém é redirecionado para seu ambiente próprio.

5.4.2.2 Visualizar horários criados

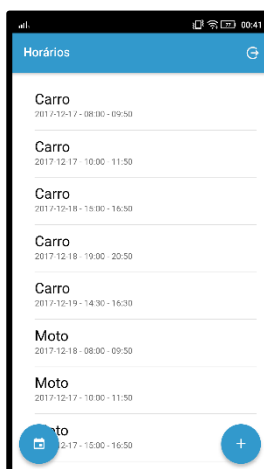
Quando o usuário efetua login como administrador ele é redirecionado a sua página inicial de administrador, esta página é semelhante a página inicial do aluno, porém o administrador irá visualizar horários criados por ele mesmo, caso tenha criado pelo menos um horário, também contará com botão de logout localizado na mesma região que o aluno e botão de adicionar também localizado na mesma região que o aluno, porém este botão redireciona para a página de criação de horários. Representa-se abaixo a comparação da tela modelada e a tela construída para o aplicativo.

Figura 29 – Tela inicial do administrador projetada



Fonte: Autor (2017)

Figura 30 – Tela inicial do administrador construída



Fonte: Autor (2017)

5.4.2.3 Criar horários

A principal funcionalidade oferecida pelo aplicativo com o perfil de administrador é a criação de horários, esta funcionalidade permite que o administrador crie os horários que serão disponibilizados aos alunos para agendamento. Para realizar a criação de horários o usuário deverá estar conectado como administrador, em seguida deverá clicar no botão adicionar localizado na tela inicial do mesmo, para que seja redirecionado a tela de criação de horários, nesta tela o administrador deverá informar o tipo de aula, data, hora em que se inicia a aula e hora na qual a aula é encerrada. Após a inserção dessas informações o usuário deverá clicar no botão confirmar para finalizar a criação de um horário. Caso o usuário não clique no botão de confirmação o horário não será criado, logo não estará disponível ao aluno. Abaixo é representada a comparação entre a tela modelada e a tela construída para criação de horários.

Figura 31 – Tela projetada



Diagrama de uma tela projetada para criação de horários. O título é "Criar Horarios". Abaixo dele, há quatro campos de entrada rotulados "Tipo", "Data", "Hora Inicio" e "Hora Fim". Na base da tela, há dois botões: um verde rotulado "Confirmar" e um vermelho rotulado "Sair".

Fonte: Autor (2017)

Figura 32 – Tela construída

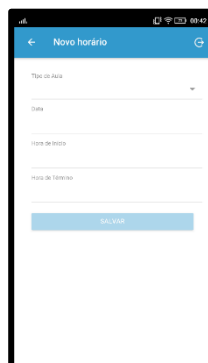


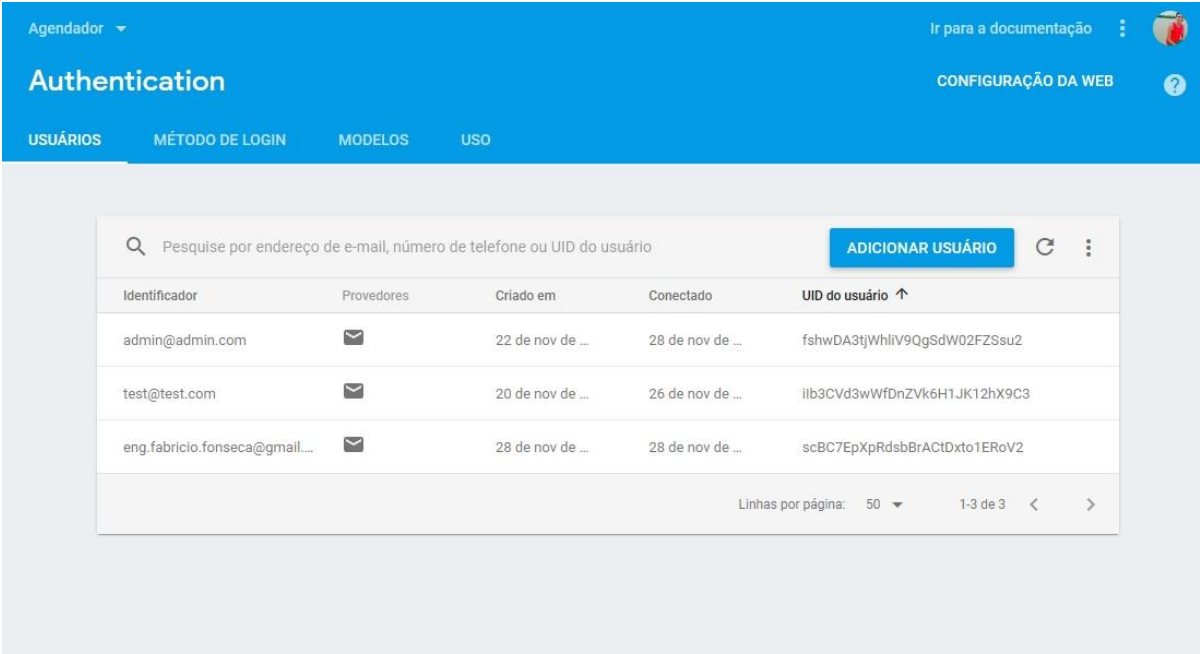
Imagem de uma tela construída para criação de horários. O título é "Novo horário". Abaixo dele, há quatro campos de entrada rotulados "Tipo de Aula", "Data", "Hora de Inicio" e "Hora de Término". Na base da tela, há um botão azul rotulado "Salvar".

Fonte: Autor (2017)

6 RESULTADOS E DISCUSSÕES

O aplicativo alcançou bons resultados, o serviço de autenticação do firebase teve pleno funcionamento, o banco de dados em nuvem foi suficiente para armazenar os dados, a hospedagem da aplicação através também do firebase foi realizada com sucesso. Dentre esses resultados vale destacar o primeiro, o serviço de autenticação foi eficaz, para comprovação de que os usuários cadastrados neste serviço foram cadastrados no serviço de nuvem verifica-se a seguir a figura que ilustra os usuários cadastrados na aplicação com seus endereços de e-mail utilizados no cadastro.

Figura 33 – Tela do serviço de autenticação do firebase



The screenshot shows the Firebase Authentication console interface. At the top, there is a navigation bar with the title 'Authentication' and a 'CONFIGURAÇÃO DA WEB' button. Below the navigation bar, there are tabs for 'USUÁRIOS', 'MÉTODO DE LOGIN', 'MODELOS', and 'USO'. The main content area displays a table of users with the following columns: 'Identificador', 'Provedores', 'Criado em', 'Conectado', and 'UID do usuário'. The table contains three rows of user data. At the bottom right of the table, there is a pagination control showing 'Linhas por página: 50' and '1-3 de 3'.

Identificador	Provedores	Criado em	Conectado	UID do usuário ↑
admin@admin.com	✉	22 de nov de ...	28 de nov de ...	fshwDA3tjWhIv9QgSdW02FZSsu2
test@test.com	✉	20 de nov de ...	26 de nov de ...	iIb3CVd3wWfDnZV6H1JK12hX9C3
eng.fabricio.fonseca@gmail...	✉	28 de nov de ...	28 de nov de ...	scBC7EpXpRdsbBrACTDxt01ERoV2

Fonte: Firebase (2017, online)

Pode-se visualizar na imagem acima o e-mail cadastrado, quando o usuário criou sua conta e data da última conexão realizada pelo usuário. Todas essas informações são organizadas pelo próprio firebase, cabe apenas ao desenvolvedor utilizar o serviço de autenticação. Isto oferece ao desenvolvedor a redução de tempo de desenvolvimento, pois esta estrutura já vem totalmente completa, apenas aguardando dados para serem inseridos. Vale ressaltar que o serviço oferece autenticação através de contas do Facebook, Google, Twitter, GitHub e outros.

7 CONSIDERAÇÕES FINAIS

Considerando o projeto proposto com um aplicativo móvel que realize agendamento de aulas, pode-se dizer que o objetivo foi concluído, os resultados obtidos foram satisfatórios e demonstraram como a aplicação pode ser uma solução viável para autoescolas. Um ponto importante do projeto é a base de dados da aplicação, com o serviço do firebase a autoescola que desejar implementar este projeto não terá necessidade de construir uma infraestrutura física para manter seu banco de dados, o firebase oferece esse serviço em nuvem, com planos gratuitos e pagos, dependendo da quantidade de acessos e dados trafegados, o cliente deverá optar por um destes planos. Outro ponto importante é o agendamento, com a utilização do aplicativo, o aluno realizará o agendamento de suas aulas em poucos minutos ou até segundos, trazendo então mesmo que através de software um processo de automatização ao agendamento.

7.1 DIFICULDADES ENCONTRADAS

Durante o desenvolvimento do projeto várias dificuldades foram surgindo e contornadas. Inicialmente, surgiu a dificuldade na falta de conhecimento de programação web, a qual seus fundamentos são essenciais para o desenvolvimento de aplicativos híbridos. Em seguida surgiu a dificuldade de qual banco de dados seria utilizado e como se executaria esse processo, pois a falta de conhecimento sobre banco de dados também dificultou o desenvolvimento. Por fim a principal dificuldade encontrada foi como integrar o aplicativo ao conjunto de ferramentas firebase, essa dificuldade foi a mais complexa dentre as outras, pois é um serviço novo que apesar de conter uma boa documentação não foi suficiente para realizar a integração.

Após estudos sobre fundamentos de HTML, CSS, JavaScript, TypeScript e Angular as dificuldades sobre desenvolvimento web foram transpassadas. Em seguida mais estudos sobre integração de Angular com firebase, manipulação de dados com JSON, através de tutoriais, fóruns de discussão e vídeo aulas, todas as outras dificuldades foram contornadas.

Apesar das dificuldades encontradas devido ao curso não ofertar a disciplina de programação web, as disciplinas de programação ofertadas pelo curso facilitaram no processo de aprendizado na programação web, as disciplinas de programação aplicada em C++ e engenharia de software, serviram de base para o aprendizado de manipulação de dados em JavaScript e modelagem do gráfico de pacotes.

7.2 PROPOSTA DE TRABALHOS FUTUROS

No desenvolvimento desta aplicação, foi observado que a implementação do aplicativo em várias autoescolas é algo a se fazer, porém como algumas autoescolas já possuem seus sistemas próprios o desafio será integrar o banco de dados do firebase aos bancos existentes nas autoescolas. Observando isso, uma possibilidade foi encontrada, transformar este aplicativo em uma plataforma. A plataforma deve conter um sistema interno onde o administrador de cada autoescola possa criar o ambiente de sua autoescola personalizado, sendo assim não será necessário o desenvolvimento de uma aplicação para cada autoescola, mas sim uma única plataforma que permita seus administradores personalizarem seus ambientes.

8 REFERÊNCIAS

AGREALA, Lucas. **Brasileiros estão cada vez mais viciados no celular.** Mai, 2017. Disponível em: <<https://exame.abril.com.br/tecnologia/brasileiros-estao-cada-vez-mais-viciados-no-celular/>>. Acesso em: 29 out. 2017.

ALVES, Bruno César. **Republic:** aplicação mobile para divulgar e procurar vagas em repúblicas universitárias. 2017. Trabalho de Conclusão de Curso (Graduação em Sistemas de Informação) – Universidade Federal de Uberlândia, Uberlândia, 2017. Disponível em: <<http://repositorio.ufu.br/bitstream/123456789/19599/1/RepublicRepublicAplicacaoMobile.pdf>>. Acesso em: 25 out. 2017.

ALMEIDA, Flávio. **TypeScript, favoreça o uso de Type Definitions.** Mai, 2017. Disponível em: <<http://blog.caelum.com.br/typescript-favoreca-o-uso-de-type-definitions/>>. Acesso em: 27 nov. 2017.

ANATEL. **Telefonia móvel registra aumento de 156.155 linhas em agosto.** Out, 2017. Disponível em: <<http://www.anatel.gov.br/dados/destaque-1/283-telefonia-movel-registra-aumento-de-156-155-linhas-em-agosto>>. Acesso em: 29 out. 2017.

ANDROID. **Android 8.0 Oreo.** Disponível em: <<https://developer.android.com/index.html>>. Acesso em: 26 nov. 2017.

APPLE. **Swift. Uma linguagem aberta e poderosa para todo mundo criar apps incríveis.** Disponível em: <<https://www.apple.com/br/swift/>>. Acesso em: 24 out. 2017.

ASTAH. **Astah is Modeling.** Disponível em: <<http://astah.net/>>. Acesso em 23 nov. 2017.

AUTOESCOLA.IO. **Desempenho de Aulas e Simulados Detran RJ.** Disponível em: <<https://play.google.com/store/apps/details?id=com.ionicframework.detran746437>>. Acesso em: 07 nov. 2017.

BALSAMIQ. **Unleash Your Creativity.** Disponível em: <<https://balsamiq.com/>>. Acesso em: 22 nov. 2017.

BARROS, Edna; CAVALCANTE, Sérgio. **Introdução aos Sistemas Embarcados.** 2010. Disponível em: <<https://pdfs.semanticscholar.org/3634/dfa05a583cf4050c693dae8293e5135b3cb8.pdf>>. Acesso em: 17 out. 2017.

_____. **Arquitetura de Sistemas Embarcados.** Pernambuco: Universidade Federal de Pernambuco. [entre 2010 e 2014]. 51 slides, color. Disponível em: <<http://slideplayer.com.br/slide/5744158/>>. Acesso em: 18 out. 2017.

BERTOLI, Sandra Maria de Souza. **Guia de segurança para dispositivos móveis:** hardware, software e comportamento. 2014. 77 f. Trabalho de Conclusão de Curso (Graduação) – Universidade Tecnológica Federal do Paraná, Curitiba, 2014. Disponível em: <http://repositorio.roca.utfpr.edu.br/jspui/bitstream/1/6696/1/CT_COINT_2013_2_11.pdf>. Acesso em: 19 out. 2017.

CAMPOS, Ana Cristina. **IBGE:** celular se consolida como o principal meio de acesso à internet no Brasil. Dez, 2016. Disponível em:

<<http://agenciabrasil.ebc.com.br/geral/noticia/2016-12/ibge-celular-se-consolida-como-o-principal-meio-de-acesso-internet-no-brasil>>. Acesso em: 05 nov. 2017.

CORDOVA, Apache. **Mobile apps with HTML, CSS & JS**. Disponível em: <<https://cordova.apache.org/>>. Acesso em: 26 nov. 2017.

CIRIACO, Douglas. **O que é API?** Mar, 2009. Disponível em: <<https://www.tecmundo.com.br/programacao/1807-o-que-e-api-.htm>>. Acesso em: 25 out. 2017.

DEITEL, Paul; DEITEL, Harvey; DEITEL, Abbey. **Android como programar**. Tradução João Eduardo Nóbrega Tortello. 2. ed. Porto Alegre: Bookman, 2015.

DEVMEDIA. **Conceitos Fundamentais de Banco de Dados**. Disponível em: <<https://www.devmedia.com.br/conceitos-fundamentais-de-banco-de-dados/1649>>. Acesso em: 21 nov. 2017.

DINO. **Aumento no consumo de smartphones provoca mudanças nas redações**. Ago., 2017. Disponível em: <<https://exame.abril.com.br/negocios/dino/aumento-no-consumo-de-smartphones-provoca-mudancas-nas-redacoes/>>. Acesso em: 29 out. 2017.

DRAW.IO. Disponível em: <<https://www.draw.io/>>. Acesso em: 21 nov. 2017.

FIREBASE. **O Firebase ajuda você a criar apps melhores e desenvolver seus negócios**. Disponível em: <<https://firebase.google.com/>>. Acesso em: 21 nov. 2017.

FLORENCIO, Heitor Medeiros. **Sistemas Digitais: Introdução**. Rio Grande do Norte: Universidade Federal do Rio Grande do Norte. 2016. 46 slides, color. Disponível em: <<https://www.dca.ufrn.br/~heitorm/aulasDCA/dca0119/DCA0119-01-Introducao.pdf>>. Acesso em: 18 out. 2017.

EMAIL. Disponível em: <<https://www.google.com/intl/pt/gmail/about/>>. Acesso em: 18 nov. 2017.

GRADLE. **Accelerate developer productivity**. Disponível em: <<https://gradle.org/>>. Acesso em: 26 nov. 2017.

INOVECFC. **MINHAS AULAS PRÁTICAS**. Disponível em: <https://www.inovecfc.com.br/img/facebook_4.jpg>. Acesso em: 09 nov. 2017.

IONIC. **The top open source framework for building amazing mobile apps**. Set, 2017. Disponível em: <<https://ionicframework.com/>>. Acesso em: 25 out. 2017.

IONIC CREATOR. **From idea to app store with Creator**. Disponível em: <<http://docs.usecreator.com/>>. Acesso em: 21 nov. 2017.

JSON. **Introdução ao JSON**. Disponível em: <<https://www.json.org/json-pt.html>>. Acesso em: 21 nov. 2017.

LENOVO. **Smartphone FAQs**. 2017. Disponível em <<https://www3.lenovo.com/br/pt/faqs/smartphone-faqs/que-e-um-smartphone/>>. Acesso em: 19 out. 2017.

MDN. **Recursos de desenvolvedores para desenvolvedores.** Disponível em: <<https://developer.mozilla.org/pt-BR/>>. Acesso em: 26 nov. 2017.

MINDS. Disponível em: <<http://www.mindsidiomas.com.br/>>. Acesso em: 05 nov. 2017.

NETMARKETSHARE. **Mobile/Tablet Operating System Market Share.** Set, 2017. Disponível em: <<https://www.netmarketshare.com/operating-system-market-share.aspx?qprid=8&qpcustomd=1&qpsp=224&qnp=2&qptimeframe=M>>. Acesso em: 24 out. 2017.

OSSADA, Jaime Cazuhiro; MARTINS, Luiz Eduardo Galvão. **Um Estudo de Campo sobre o Estado da Prática da Elicitação de Requisitos em Sistemas Embarcados.** 2010. Disponível em: <https://s3.amazonaws.com/academia.edu.documents/34778956/ossada.pdf?AWSAccessKeyId=AKIAIWOWYYGZ2Y53UL3A&Expires=1508269675&Signature=xAqARGMMbcsWX8gQui8CkPZLEzU%3D&response-content-disposition=inline%3B%20filename%3DUm_Estudo_de_Campo_sobre_o_Estado_da_Pra.pdf>. Acesso em: 17 out. 2017.

RAFAEL, Elton. **Introdução aos Sistemas Embarcados.** Pará: Centro Universitário do Estado do Pará. [entre 2014 e 2016]. 43 slides, color. Disponível em: <www.cesupa.br>. Acesso em: 15 out. 2017.

RAMIRES, Thammy. **Qual a diferença entre web app, app nativo e app híbrido.** Jan, 2017. Disponível em: <<http://fabricadeaplicativos.com.br/aprender/qual-diferenca-entre-web-app-app-nativo-e-aplicativo-hibrido/>>. Acesso em: 25 out. 2017.

RAMOS, Anália Saraiva Martins; COSTA, Fabrício de São Paulo H. A. Rubim. Serviços bancários pela internet: um estudo de caso integrando a visão de competidores e clientes. **RAC.** Curitiba. v. 4, n. 3, p. 133-154, Set-Dez, 2000. Disponível em: <<http://www.scielo.br/pdf/rac/v4n3/v4n3a08.pdf>>. Acesso em: 29 out. 2017.

SILVA, Marcelo Moro da; SANTOS, Marilde Terezinha Prado. Os paradigmas de Desenvolvimento de Aplicativos para Aparelhos Celulares. **T.I.S.** São Carlos. v. 3, n. 2, p. 162-170, Mai-Ago, 2014. Disponível em: <<http://revistatis.dc.ufscar.br/index.php/revista/article/view/86/80>>. Acesso em: 23 out. 2017.

TADEU, Paulo Henrique Alves. **Autoescola APP.** Disponível em: <https://play.google.com/store/apps/details?id=solucaosci.autoescolaapp&hl=pt_BR>. Acesso em: 09 nov. 2017.

TAURION, Cezar. **Desenvolvimento de apps – Parte 2: híbrido, nativo ou web?** 15 jul. 2013. Disponível em: <https://www.ibm.com/developerworks/community/blogs/ctaurion/entry/desenvolvimento_de_apps-parte_2_hibrido_nativo_ou_web?lang=en>. Acesso em: 23 out. 2017.

UML. **WHAT IS UML?** Disponível em: <<http://www.uml.org/what-is-uml.htm>>. Acesso em: 23 nov. 2017.

VAHID, Frank; GIVARGIS, Tony. **Embedded System Design: A Unified Hardware/Software Approach.** Riverside: Fall, 1999.

VISUAL STUDIO CODE. **Code Editing. Redefined.** Disponível em:
<<https://code.visualstudio.com/>>. Acesso em: 21 nov. 2017.