

CENTRO UNIVERSITÁRIO DO PARÁ - CESUPA
ÁREA DE CIÊNCIAS EXATAS E TECNOLOGIA - ACET
CURSO DE BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO

BÁRBARA RAMOS LOBO

APLICAÇÃO MOBILE PARA ACADÊMICOS DO ENSINO SUPERIOR

BELÉM

2017

BÁRBARA RAMOS LOBO

APLICAÇÃO MOBILE PARA ACADÊMICOS DO ENSINO SUPERIOR

Trabalho de Curso na modalidade produto,
apresentado como requisito parcial para
obtenção de título de Bacharel em
Engenharia de Computação do Centro
Universitário do Estado do Pará –
CESUPA

Orientador: Professor Msc. Ricardo Melo
Casseb do Carmo.

BELÉM

2017

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

Lobo, Bárbara Ramos.

Aplicação mobile para acadêmicos do ensino superior / Bárbara Ramos Lobo;
orientação de Ricardo Melo Casseb do Carmo, 2017.

Trabalho de Conclusão de Curso (Bacharelado em Engenharia de Computação) -
Centro Universitário do Estado do Pará, 2017.

1. Software – Desenvolvimento. 2. Aplicativos híbridos. I. Carmo, Ricardo melo
Casseb do (orient.). II. Título.

BÁRBARA RAMOS LOBO

APLICAÇÃO MOBILE PARA ACADÊMICOS DO ENSINO SUPERIOR

Trabalho de Curso na modalidade produto,
apresentado como requisito parcial para
obtenção de título de Bacharel em
Engenharia de Computação do Centro
Universitário do Estado do Pará –
CESUPA

Data da Aprovação: 14 / 12 / 2017

Banca Examinadora

Prof. Professor Msc. Ricardo Melo Casseb do Carmo-CESUPA
Orientador e Presidente da banca

Prof. Itamar Jorge Vilhena de Brito-CESUPA
Examinador interno

Prof. Eudes Danilo da Silva Mendonça-CESUPA
Examinador interno

AGRADECIMENTOS

Agradeço a Deus pela oportunidade oferecida, aos meus pais por terem me ajudado tanto nessa etapa e estarem sempre comigo. Ao meu noivo Apoena Nogueira, por ter me ajudado durante todo esse tempo em que estive me formando enquanto profissional e pessoa.

Agradeço a todos os meus familiares por sempre estarem presentes durante toda essa caminhada. Ao meu irmão, que mesmo de longe sempre se fez presente.

Agradeço a todos os professores que acompanharam minha jornada enquanto universitária e foram essenciais à minha formação como profissional e, além disso, minha evolução como pessoa. De modo particular ao meu professor orientador por todo apoio a mim oferecido.

Por fim, agradeço a todos os meus amigos e colegas de curso que estiveram comigo nessa jornada, principalmente ao meu colega Fabricio Fonseca, o qual me ajudou nas horas de dificuldades deste trabalho.

RESUMO

O objetivo deste projeto é desenvolver um aplicativo móvel para acadêmicos do ensino superior que funciona como uma agenda acadêmica, proporcionando ao usuário um ambiente adequado para o registro de suas atividades, no qual tem duas opções de realizar o armazenamento de tarefas, que podem ser através de notas textuais ou através de imagens, além de outras aplicabilidades que o mesmo possui. O aplicativo também pode ser usufruído por estudantes de diferentes níveis de escolaridade. É uma aplicação híbrida, na qual pode ser usada em plataformas Android e IOS ou pode ser acessada pela web. Este trabalho utilizou como principais tecnologias o *framework* Ionic e a plataforma Google Firebase para o desenvolvimento dos serviços multiplataforma.

Palavras-Chave: Ionic. Google Firebase. Aplicativos Híbridos. Serviços Multiplataforma.

ABSTRACT

The objective of this project is to develop a mobile application for high school academics that works like an academic diary, providing an appropriate environment to record their student's activities, which has two options to perform the task storage and can be done by text notes or images, besides others applications. Different schooling levels users can also use the application. It is a hybrid application that runs on Android and IOS platforms or can be accessed with a web browser. This article used as main technologies Ionic Framework and Google Firebase database to build multiplatform services.

Keywords: *Ionic.Google Firebase.Hybrid Applications.Multiplatform Services*

LISTA DE FIGURAS

Figura 1- Diferença entre o Microprocessador e o Microcontrolador.....	16
Figura 2-Lógica do Circuito Integrado de um Smartphone.....	18
Figura 3- Aplicações e Suas Tecnologias	19
Figura 4- Agenda Estudante	23
Figura 5- Agenda Escolar.....	24
Figura 6- Agenda Universitária.....	24
Figura 7-Infográfico do <i>Agenda Acadêmica</i>	25
Figura 8- Ionic.....	26
Figura 9-Node.js.....	26
Figura 10-Apache Cordova	27
Figura 11-Angula.Js	27
Figura 12-Atom.....	28
Figura 13-Tela de Edição do Atom.....	29
Figura 14-Draw.IO.....	30
Figura 15- Google Firebase.....	30
Figura 16-JSON	31
Figura 17-Interface do Justinmind Prototyper.....	32
Figura 18-Tela do Database do Google Firebase	33
Figura 19- Tela de Authentication do Google Firebase	34
Figura 20- Tela do Storage do Google Firebase.....	35
Figura 21-Diagrama de Caso de Uso do Agenda Acadêmica	36
Figura 22-Diagrama de Pacote do Agenda Acadêmica.....	37
Figura 23-Diagrama de Estado do Agenda Acadêmica	38
Figura 24-Versão Inicial da Tela de Login e Versão Final.	39
Figura 25-Tela de Cadastro da Versão Inicial e Final.....	40
Figura 26-Wireframe Inicial e Versão Final da Tela Redefinir Senha.....	40
Figura 27-Prototipagem Inicial e a Versão Final da Tela de Tarefa	41
Figura 28-Prototipagem Inicial e a Tela Final da Página Adicionar	42
Figura 29-Prototipagem Inicial e a Tela final do Menu	43
Figura 30-Prótotipagem Inicial e Versão Final da Tela de Agenda de Tarefas por Foto.....	43

LISTA DE SIGLAS

App- *Aplicativo*

API - *Application Programming Interface*

CSS - *Cascading Style Sheets*

HSDPA- *High-Speed Downlink Packet Access*

HTML- *HyperText Markup Language*

IDE - *Integrated Development Environment*

JSON - *JavaScript Object Notation*

NoSQL- *Not Only SQL*

NPM -*Node Package Manager*

SDK - *Software Development Kits*

UID -*User Identification*

UML-*Unified Modeling Language*

SUMÁRIO

1 INTRODUÇÃO	12
1.1 OBJETIVOS	13
1.1.1 OBJETIVO GERAL	13
1.1.2 OBJETIVO ESPECIFICO	13
1.2 JUSTIFICATIVA.....	13
2 FUNDAMENTAÇÃO TEÓRICA.....	15
2.1 SISTEMAS EMBARCADOS	15
2.1.1 Smartphone	17
2.2 APLICATIVOS MÓVEIS.....	18
2.2.1 Aplicativos Nativos	19
2.2.2 Aplicativos Web	20
2.2.3 Aplicativos Híbridos	21
3 TRABALHOS RELACIONADOS	23
4 DESENVOLVIMENTO DO APLICATIVO AGENDA ACADÊMICA	25
4.1 TECNOLOGIAS UTILIZADAS	25
4.1.1 IONIC	25
4.1.1.1 Node.js.....	26
4.1.1.2 Apache Cordova.....	26
4.1.1.3 NPM	27
4.1.1.4 AngularJS.....	27
4.1.1.5 HTML	28
4.1.1.6 TypeScript	28
4.1.1.7 CSS	28
4.1.2 Atom.....	28
4.1.3 Draw.io.....	29
4.1.4 Google Firebase.....	30
4.1.5 JSON	31
4.1.6 Justinmind Prototyper	31
4.2 BANCO DE DADOS	32
4.3 ARQUITETURA.....	35

4.3.1 Diagrama de Caso de Uso	35
4.3.2 Diagrama de Pacote	36
4.3.3 Diagrama de Estado.....	38
4.4 FUNCIONALIDADES DO SISTEMA	38
4.4.1 Login	39
4.4.2 Cadastro.....	39
4.4.3 Redefinir Senha.....	40
4.4.4 Agenda de Tarefas por Foto	41
4.4.5 Adicionar Tarefa.....	41
4.4.6 Menu	42
4.4.7 Agenda de Tarefa por Foto	43
5 CONSIDERAÇÕES FINAIS.....	44
5.1 DIFICULDADES ENCONTRADAS	44
5.2 TRABALHOS FUTUROS	44
6 REFERÊNCIAS	46

1 INTRODUÇÃO

Com o surgimento do smartphone que se deu no ano de 1994, o primeiro a ser comercializado foi o Simon, desenvolvido pela IBM. Nele apenas havia algumas funções como: enviar e-mail, SMS, calendário, podia fazer anotações, etc. Naquela época ele não era conhecido pelo nome de smartphone, e sim, de celular. A partir dele, foram surgindo outros, só que cada vez mais desenvolvidos. (O GLOBO, 2014).

Segunda uma pesquisa realizada pela GSMA Intelligence (GSM,2017), o número de pessoas que possuem *smartphones* no mundo é de 5 bilhões. A china é o país que lidera com mais de 1 bilhão de pessoas possuem smartphones.

De acordo com o site Mobile Time, cerca de 1.495 bilhões de smartphones foram vendidos no ano de 2016 em todo o mundo, onde houve um acréscimo de 5% em relação a 2015. (MOBILE TIME,2017).

A Agência Nacional de Telecomunicações (Anatel), registrou 241.1 milhões de celulares em setembro de 2017 no Brasil. (ANATEL,2017).

O crescimento de vendas de smartphones, se dá pelo fato de ser um aparelho que se tornou necessário no dia a dia das pessoas, devido suas funcionalidades. O principal motivo de possuir um dispositivo deste é por conta de aplicativos que podem ser armazenado nele ou até mesmo ser acessado através de seu navegador. Os aplicativos auxiliam as pessoas em várias atividades rotineiras.

Através dos aplicativos dos smartphones, tudo pode ser feito como: Visualização da situação do trânsito, analisar a conta bancária, usufruir das diversas redes sociais, para atividades profissionais, educativas, entretenimento, baixar vídeos e músicas, dentre muitas outras. (SMARTPHONES TUDO, 2014).

Visando o mercado de aplicativos, este produto tem como objetivo oferecer comodidade aos alunos do ensino superior, mas ele não se restringe apenas nestes, também poderá ser usado por outros estudantes de diferentes níveis de escolaridade, visando uma organização em suas tarefas. Por ser um aplicativo que faz o registro de tarefas tanto por texto quanto por foto, ele servirá para qualquer pessoa, além de estudante e, pode ajudar profissionais de diferentes áreas a organizar suas atividades do dia a dia.

Neste trabalho, foi utilizado o *framework* Ionic e a plataforma Google Firebase, a qual faz a armazenagem do banco de dados na nuvem, além de outros serviços. E foram usadas outras tecnologias para complementar o *framework*.

1.1 OBJETIVOS

1.1.1 Objetivo Geral

Desenvolver um aplicativo híbrido de agenda para atividades acadêmicas.

1.1.2 Objetivo Especifico

- Analisar o mercado de aplicativos móveis;
- Criar aplicativo a partir de framework híbrido;
- Utilizar banco de dados Real-time NoSQL para sincronização entre usuários;

1.2 JUSTIFICATIVA

Este projeto se justifica pelo desenvolvimento de um aplicativo mobile para acadêmicos do ensino superior, onde ele vai fazer a sincronização do estudante com as suas tarefas acadêmicas, pois, é um ambiente onde o aluno pode realizar suas anotações, coisa que atualmente é feita em algum outro ambiente, como: aplicativo de bloco de anotações ou em caderno, sendo que nem todas as notações são feitas no mesmo lugar, desta forma, permite que o aluno acabe esquecendo-se de seus afazeres.

Este produto tem o objetivo de fazer com que o aluno coloque suas informações acadêmicas em apenas um aplicativo, para que as mantenha organizadas e em dia. Conforme isto, não será preciso fazer anotações em diversos lugares.

Além disso, o aplicativo fornece comodidade ao usuário, pois funciona como uma agenda, nele irá conter todas as informações das tarefas acadêmicas, mas para isso, o mesmo deve ser alimentado, e quem deve fornecer essas informações é o próprio usuário. Ao adquiri-lo ele deve realizar seu cadastro e, após isso, pode ser usado para registrar suas atividades acadêmicas, cada uma com sua destinação adequado, podendo ser registrada por anotações simples ou por foto.

1.3 CAPÍTULOS DO PROJETO

Neste tópico será descrito a distribuição deste trabalho.

- Capítulo 1 – Introdução. Iniciando com uma contextualização com informações atuais, apresentando os objetivos e a justificativa do projeto.
- Capítulo 2 - Fundamentação Teórica. Neste capítulo possui os fundamentos teóricos usados para o desenvolvimento do projeto.

- Capítulo 3 – Trabalhos Relacionados. Este capítulo apresenta os tipos de aplicativos relacionados a este produto e, as diferenças entre eles.
- Capítulo 4 – Desenvolvimento do Aplicativo Agenda Acadêmica. Este capítulo é iniciado com uma breve explicação sobre o produto, dando continuidade as tecnologias utilizadas, as funcionalidades, a arquitetura e sua interface gráfica.
- Capítulo 5 – Considerações Finais. Neste capítulo é apresentada as conclusões obtidas a partir do desenvolvimento deste projeto, as dificuldades encontradas e trabalhos futuros.
- Capítulo 6 – Neste capítulo consta as referências utilizadas como estudo para o desenvolvimento do projeto.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 SISTEMAS EMBARCADOS

Sistemas embarcados (*Embedded Systems*) que também é chamado de sistema embutido, são projetados para serem dedicados a desempenhar uma determinada tarefa de um equipamento, que normalmente possui um microcontrolador e um software. Pode se dizer que é um sistema microprocessado no qual o computador é completamente encapsulado ou dedicado ao dispositivo ou sistema que ele controla. “Outros aspectos relevantes dependem dos tipos de sistemas, modos de funcionamento e itens desejados em aplicações embarcadas” (CHASE,2007).

De acordo com Pereira et al. (2015, p.3).

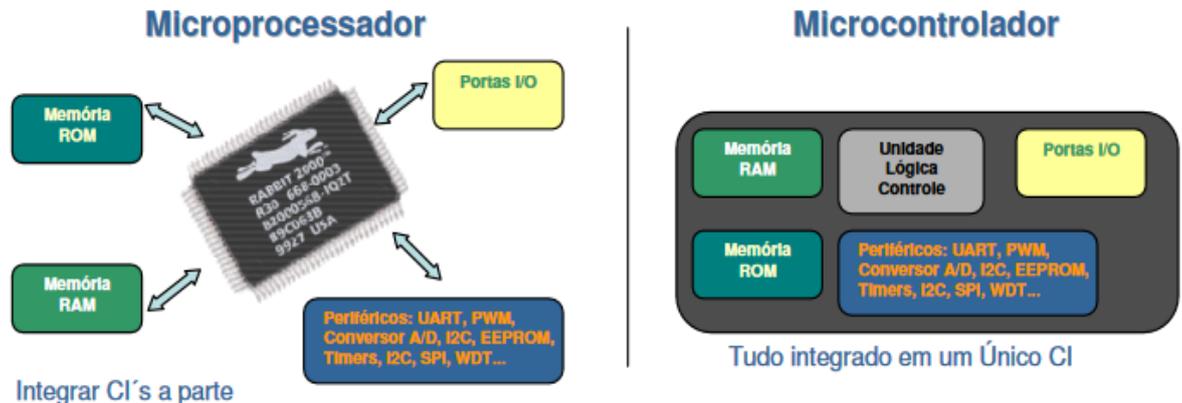
Para entender como funcionam esses sistemas e a necessidade de utilizá-los, é preciso conhecer a forma como trabalham os processadores onde os mesmos são armazenados. Os processadores podem ser divididos em duas categorias: microprocessadores e micro-controladores. Os Microprocessadores são circuitos integrados capazes de executar instruções a este submetidas, agem sob o controle de um programa armazenado na memória, executando operações aritméticas, lógica booleana, além de controlar a entrada e saída de dados, permitindo a comunicação com outros dispositivos e periféricos. Os microprocessadores estão presentes em sua quase totalidade em computadores.

Os microprocessadores necessitam de outros dispositivos para seu funcionamento, tais como memória e controladores de entrada e saída de dados, com base nisto, o mercado enxergou a necessidade de criar um sistema de processamento, que já incorporasse os dispositivos necessários para seu funcionamento, desta forma, surgiram os microcontroladores. (PEREIRA et al.,2015,p.3).

O microcontrolador é uma espécie de computador em um chip, pois em um chip são integrados: processador, memória e funções de entrada/saída. Nesses chips são acoplados os softwares (*firmware*), que é armazenado em uma memória ROM ou memória flash, pois é ele que comanda o equipamento eletrônico, sendo assim, o sistema não pode ser alterado pelo usuário final, apenas poderá ser feito o *upgrade*, isso, quando é liberado para o usuário uma nova versão. (PEREIRA et al.,2015, p.3).

Na figura abaixo será ilustrado os processadores.

Figura 1- Diferença entre o Microprocessador e o Microcontrolador



Fonte: Pereira et al. (2015, online).

Segundo Vahid e Givargis(1999), os sistemas embarcados possuem algumas características comuns:

- Funcionamento único: normalmente um sistema executa apenas um programa repetidamente. O desktop é um tipo de sistema que executa vários programas ao mesmo tempo.
- Reativo e em tempo real: alguns sistemas podem sofrer mudanças no seu ambiente, e ao ser alterado, ele deve reagir e calcular determinados resultados em tempo real.
- Muito restrita: os sistemas computacionais têm restrições sobre o design métricas, mas as de sistemas embarcados podem ser especialmente apertadas. Um design a métrica é uma medida dos recursos de uma implementação, como custo, tamanho, desempenho e poder. Os sistemas embutidos devem caber em um único chip, mas, devem ser rápidos para processar as informações em tempo real.

Existem casos em que um programa de um sistema integrado sofre atualização de uma nova versão. Um exemplo disso são os celulares, eles podem ser atualizados. Durante essa atualização, são trocados dentro e fora do sistema alguns programas devido a restrição do tamanho.

2.1.1 Smartphone

Smartphone (Telefone Inteligente) é um tipo de dispositivo móvel, que por muitas pessoas é conhecido pelo termo celular, por ser um modelo de celular, só que, mais desenvolvido.

Smartphone é um celular com capacidade avançada, que executa um sistema operacional identificável permitindo aos usuários ampliar suas funcionalidades com aplicações terceiras que estão disponíveis em uma loja de aplicativos, De acordo com esta definição, os smartphones devem incluir um hardware sofisticado com: a) capacidade de processamento avançada (CPUs modernas, sensores) b) Capacidade de conexões múltiplas e rápidas (Wi-Fi, HSDPA) e c) tamanho de tela adequado e limitado. Além disso, seu Sistema Operacional deve ser claramente identificável, como Android, Blackberry, Windows Phone, Apple`s IOS, etc. (THEOHARIDOU; MYLONAS; GRITZALDIS, 2012,p. 3. Tradução do autor)¹.

Pelo fato de ser um sistema embarcado, também possui um *firmware*, que é o sistema operacional, mas o seu pode ser atualizado quando surge uma nova versão. Hoje em dia, essa atualização pode ser feita pelo próprio usuário final, sendo que, é necessário que haja uma liberação para que o próprio usuário possa vim a fazer está atualização. Na figura abaixo será ilustrado a lógica do circuito integrado.

¹ Smartphone is a cell phone with advanced capabilities, which executes an identifiable operating system allowing users to extend its functionality with third party applications that are available from an application repository. According to this definition, smartphones must include sophisticated hardware with: a) advanced processing capabilities (e.g. modern CPUs, sensors), b) multiple and fast connectivity capabilities (e.g. Wi-Fi, HSDPA), and (optionally) c) adequately limited screen sizes. Furthermore, their OS must be clearly identifiable, e.g. Android, Blackberry, Windows Phone, Apple`s iOS, etc. (THEOHARIDOU; MYLONAS; GRITZALDIS, 2012,p. 3)

ferramentas já é possível desenvolver um aplicativo mobile uma única vez e poder utilizá-lo em multiplataformas de dispositivos móveis. (SILVA; SANTOS, 2014).

Com a quantidade de aplicativos sendo desenvolvidos constantemente, surgiram várias formas de desenvolvê-los. No mercado, estão sendo usados os tipos: nativos e multiplataformas, que são os híbridos e web, como mostra na figura abaixo.

Figura 3- Aplicações e Suas Tecnologias

Tipo de Aplicação	Linguagem, Bibliotecas, Framework, SDK, IDE.
WEB	HTML5, JQUERY, CSS3, JAVASCRIPT, JSON, NOTEPAD++.
NATIVA	OBJECTIVE-C, JSON, XCODE, IOS SIMULATOR.
HÍBRIDA	OBJECTIVE-C, XCODE, IOS SIMULATOR, CORDOVA, HTML5, JQUERY, CSS3, JAVASCRIPT, JSON.

Fonte: Silva, Pires, Neto (2015, online)

Conforme a figura acima, vimos um pouco das tecnologias que são necessárias para o desenvolvimento dos aplicativos, algumas delas serão explicadas abaixo, como:

- *Linguagem*: é definido como um conjunto de sintaxe e semântica, onde é interpretada por um compilador.
- *Biblioteca*: é um conjunto de funções pré-compiladas, que é escrita por outros programadores.
- *Framework*: é composto pelo conjunto de bibliotecas, elementos de software e diretrizes de arquitetura que fornecem um conjunto de ferramentas.

2.2.1 Aplicativos Nativos

Os aplicativos nativos são desenvolvidos para serem executados em uma plataforma específica de um aparelho celular.

Segundo Silva e Santos (2014, p. 2).

Essas plataformas são compostas de diversas tecnologias, tais como: sistema operacional, linguagens de programação e IDEs (*Integrated Development Environment*). O sistema operacional é responsável por gerenciar diversos recursos do aparelho celular, as linguagens de programação são utilizadas na programação do

aplicativo e o IDE, também conhecido como ambiente de desenvolvimento integrado, fornece ferramentas que auxiliam na criação do aplicativo.

“Cada um desses sistemas operacionais requer que aplicativos nativos sejam criados usando uma linguagem de codificação específico. Para aqueles que tomam notas, é o Swift para IOS, Java para Android, e normalmente C ++ ou C # para Windows”.(WHITE, 2013.Tradução do autor)².

Devido as plataformas serem diferentes umas das outras, existem variações de ferramentas, permitindo ou não o desenvolvido do aplicativo para uma determinada plataforma.

Os fabricantes de sistemas operacionais fornecem um conjunto de protocolos necessários para o desenvolvimento do aplicativo. AS APIs (*Application Programming Interface*) disponibilizadas pelos fabricantes podem ser usadas pelos programadores, para que possam ter acesso aos recursos do dispositivo, como câmeras, antenas GPS,etc .(WHITE, 2013).

Outro aspecto importante dos aplicativos nativos é a independência de comunidades responsáveis por versões de frameworks de desenvolvimento de aplicativos para celular, pois em um aplicativo nativo as atualizações de API do próprio sistema operacional do aparelho celular serão utilizadas no desenvolvimento de novas funcionalidades do aplicativo nativo, portanto a partir do momento que a plataforma é atualizada, já é possível incorporar as novidades existentes da plataforma para desenvolver para o aplicativo nativo, sem a dependência da atualização de tecnologias adjacentes, como por exemplo, algum framework particular.(SILVA; SANTOS, 2014,p.2).

2.2.2 Aplicativos Web

O desenvolvimento de aplicativos para dispositivos móveis é possível a utilização de aplicações web, porem com limitações se comparadas com as aplicações nativas, mas possui suas vantagens pois não necessita estar instalado no aparelho, ele se utiliza da ligação da internet para acessar os servidores que contém as aplicações, salvando apenas fragmentos no dispositivo

² Each of these operating systems requires that native apps be built using a particular coding language. For those taking notes, it is *Objective-C* for iOS, *Java* for Android, and typically *C++* or *C#* for Windows.”.(WHITE, 2013).

móvel e acumulando a carga de informação nos servidores. Para a seu desenvolvimento é necessário o uso do *HyperText Markup Language* (HTML) e *Cascading Style Sheets* (CSS), podendo ser programado em JavaScript.

Este aplicativo é aconselhado para projetos cujo custo e tempo sejam reduzidos e a necessidade de várias plataformas seja um requisito essencial. Por ser uma aplicação interpretada pelo navegador do dispositivo, a mesma passa a ser visualizada em sistemas operacionais diferentes.

White (2013), define este tipo de aplicativo como sites otimizados para dispositivos de telas menores, que é o caso dos telefones móveis. O sistema operacional do dispositivo não tem domínio sobre seu conteúdo ou funcionalidades. Os desenvolvedores podem fazer alterações a qualquer momento, com efeito imediato aos usuários.

2.2.3 Aplicativos Híbridos

A existência de aplicativos compatíveis para vários sistemas operacionais é possível graças aos aplicativos híbridos, pois em sua concepção ele é desenvolvido com ferramentas tanto web quanto nativo, ou seja, o aplicativo funciona utilizando os dados da web quanto a capacidade do aparelho, em muitos casos não sendo necessário estar online para ter o funcionamento do mesmo.

Segundo Charland e Leroux (2011), uma das semelhanças dos sistemas operacionais usados nos smartphones é o fato deles utilizarem navegadores móveis para acessar o código nativo de cada plataforma. Por sua vez concede a permissão ao navegador para abrir uma instância, denominada de *WebView*, que faz a interação com o seu código nativo. Por meio desta abordagem, é possível desenvolver um aplicativo híbrido usando as linguagens HTML, CSS e JavaScript, e ainda usa o API JavaScript, para ter acesso aos recursos e sensores de dispositivos nativos.

Este modelo pode ser empacotado de acordo com a plataforma de destino, após isso, é possível serem adicionados em uma loja de aplicativos como os nativos, para que possam ser comercializados para o usuário final.

Pelo fato dos aplicativos híbridos utilizarem uma parte da linguagem nativa e outra parte do app web, tem um custo baixo para o seu desenvolvimento. Existem algumas tecnologias que desenvolvem esse tipo de aplicativo, como o React JS, Alpha Anywhere, Ionic, entre outros.

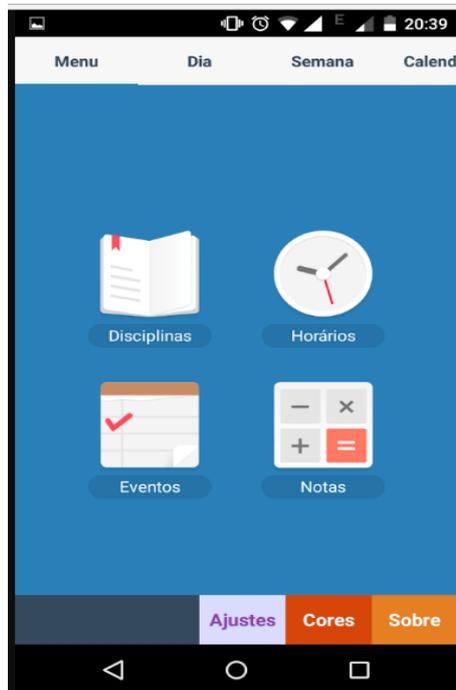
3 TRABALHOS RELACIONADOS

Os aplicativos mobile que serão mostrados mais abaixo, foram desenvolvidos para estudantes, os mesmos buscam a organização dos alunos diante a vida acadêmica. Possuem funcionalidades como: agenda, notas, eventos, horários, entre outras funcionalidades. O desenvolvimento destes foi direcionado a plataforma Android.

O *Agenda Acadêmica* se diferencia por ser um aplicativo híbrido, então pode ser usado em diferentes plataformas, como o Android e IOS, e, pode ser acessado pela web, outro diferencial é a câmera que contém nele, na qual o usuário pode fazer registros por foto.

Nas figuras 4, 5 e 6, serão mostrados os aplicativos Agenda Estudante, Agenda Escolar e Agenda Universitária.

Figura 4- Agenda Estudante



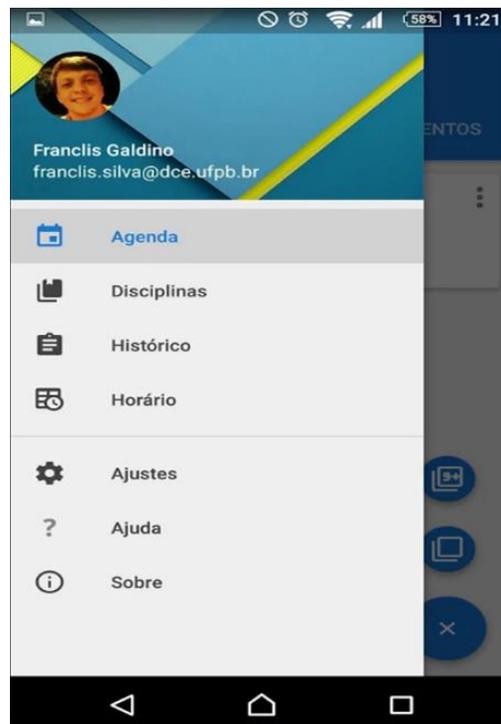
Fonte: Appsbuyout Dev(2017).

Figura 5- Agenda Escolar



Fonte: Cin, Andrea Dal (2017).

Figura 6- Agenda Universitária



Fonte: Galdino, Francis (2017).

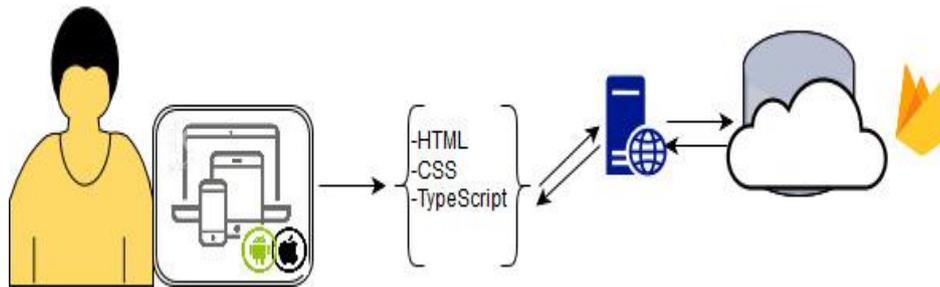
4 DESENVOLVIMENTO DO APLICATIVO AGENDA ACADÊMICA

A *Agenda Acadêmica* é um aplicativo mobile, desenvolvido para ser uma aplicação híbrida, que pode ser utilizado em plataformas *Android* e *IOS*, e também pode ser acessado através da web. Seu principal objetivo é fazer com que os estudantes tenham o controle de suas tarefas acadêmicas, possibilitando uma organização durante todo o seu semestre letivo.

Através de um dispositivo móvel o usuário pode ter acesso ao aplicativo e, os dados informados nele são armazenados na plataforma Google Firebase que faz a hospedagem de suas funcionalidades na nuvem, como por exemplo, o banco de dados.

A figura abaixo exemplifica o funcionamento do sistema.

Figura 7-Infográfico do Agenda Acadêmica



Fonte: Autor(2017).

4.1 TECNOLOGIAS UTILIZADAS

4.1.1 IONIC

O Ionic é um *framework* que tende a criação de aplicativos híbridos para dispositivos móveis. Onde realiza a emulação das diretrizes nativas do UI (*User Interface*) do aplicativo, onde é feita a interação do usuário com a máquina e usa SDKs nativos, trazendo os padrões de UI e recursos do dispositivo de aplicativos nativos, juntamente com a total potência e flexibilidade da web. (IONIC,2017?).

Figura 8- Ionic

Fonte: Ionic (2017?, Online).

Está foi a ferramenta utilizada para o desenvolvimento deste projeto, o qual também utiliza outras tecnologias como recurso, de acordo com o tipo de implementação que será feita. As tecnologias utilizadas juntas a ele foram o : Node.Js, Apache Cordova, NPM , AngularJS , HTML, CSS e *TypeScripte*.

4.1.1.1 Node.js

O Node.js é um interpretador de código *JavaScript*, onde trabalha ao lado de um servidor e faz uso de rede escalável.(NODE.JS,2017).

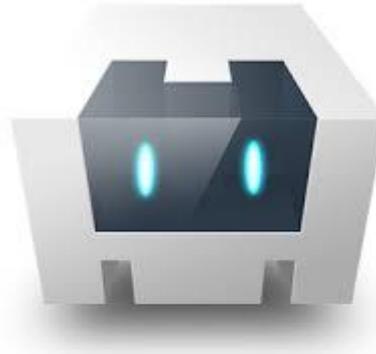
Figura 9-Node.js

Fonte: Node.js (2017).

4.1.1.2 Apache Cordova

O Apache Cordova é um *framework* de desenvolvimento móvel de código aberto. Permite a utilização de tecnologias web padrão - HTML5, CSS3 e JavaScript para o desenvolvimento de aplicativos para várias plataformas. Podendo serem executados dentro de wrappers direcionados a cada plataforma e dependem de ligações de API compatíveis para acessar as funções de cada dispositivo, como sensores, dados, status da rede, etc. (APACHE CORDOVA, 2017?).

Figura 10-Apache Cordova



Fonte: Apache Cordova (2017?,Online).

4.1.1.3 NPM

NPM (*Node Package Manager*) que também é conhecido com gerenciador de pacote para a linguagem JavaScript, que são as bibliotecas e aplicações de código aberto. Pode-se dizer que é um repositório online para publicação de projetos de código aberto para o Node.js e consiste em um utilitário de linha de comando que interage com este repositório online, que também é chamado de npm, e um banco de dados online que ajuda na instalação de pacotes, gerenciamento de versão e gerenciamento de dependências. (NodeBR,2016).

4.1.1.4 AngularJS

O AngularJS é uma plataforma que facilita na concepção de aplicativos na web, ele permite a expansão da linguagem HTML para a sua aplicação. É formado por um conjunto de ferramentas para a construção de um framework adequado, buscando obter um melhor desempenho para a criação de um aplicativo. Os recursos podem ser modificados ou substituídos para se adequar ao seu fluxo de trabalho de desenvolvimento exclusivo e necessidades de recursos. (AngularJS, 2017?).

Figura 11-Angula.Js



Fonte: Angular(2017?, Online).

4.1.1.5 HTML

É uma linguagem de marcação que é responsável pela estruturação de páginas web. Faz o uso de sintaxe simples, baseada em tags, que representam os diferentes elementos de uma página, como imagens e links. Ao acessar uma página web através de um navegador, ele possibilita a interpretação do código HTML, de forma compreensível para que o usuário final possa entender o que contém na página. (RODRIGUES, 2016?).

4.1.1.6 TypeScript

TypeScript é uma linguagem para desenvolvimento JavaScript em larga escala. Com ele pode-se escrever em código aberto usando uma estrutura tipada e ter o código compilado para JavaScript. Além disso, pode ser rodado em qualquer navegador. (HERMAN; MENDES; SCRIPT, entre 2012-2017).

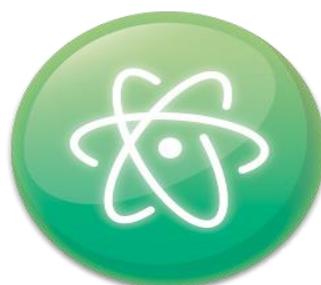
4.1.1.7 CSS

O CSS (folha de estilo cascata) é uma linguagem usada para descrever a apresentação de um documento escrito em HTML ou em XML. É responsável pela descrição de como os elementos serão mostrados na tela. (MDN, 2017).

4.1.2 Atom

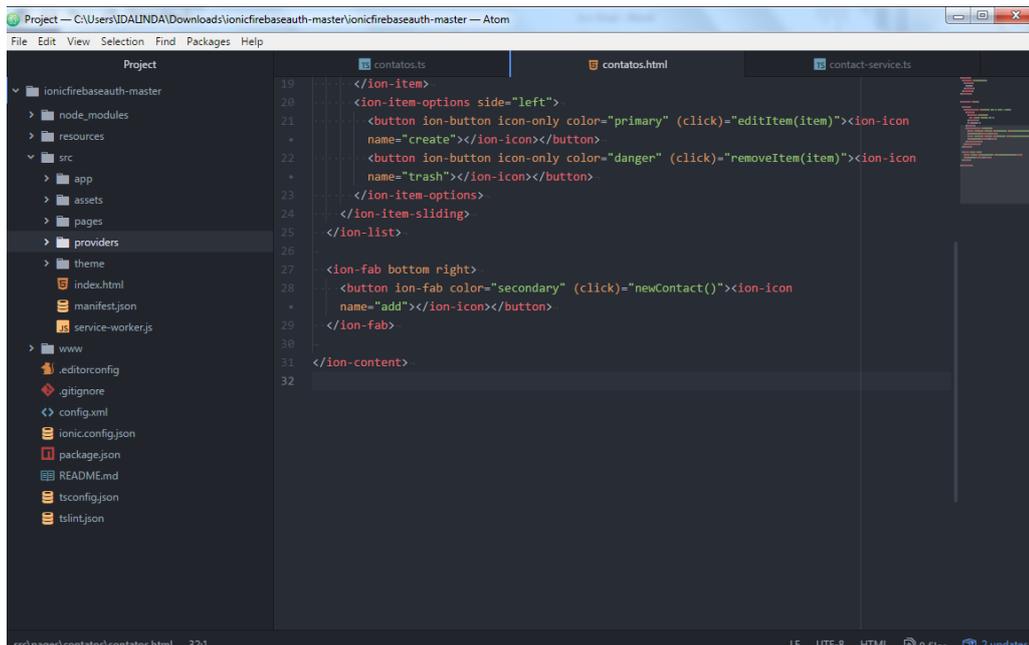
O Atom é um editor de texto de código aberto disponível para as plataformas Linux, macOS e Windows, que foi criada com integração HTML, JavaScript, CSS e Node.js. Ele é executado no Electron, uma estrutura para a construção de aplicativos de plataforma cruzada usando tecnologias da web. (ATOM, 2017?).

Figura 12-Atom



Fonte: Atom(2017?,Online).

Figura 13-Tela de Edição do Atom



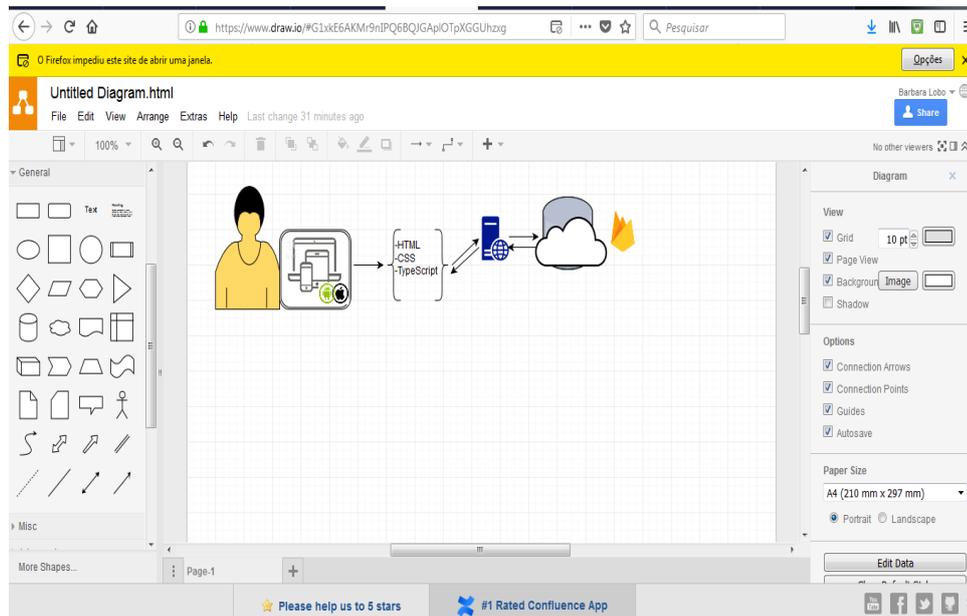
Fonte: Autor(2017).

4.1.3 Draw.io

O Draw.io é um editor gráfico online no qual é possível desenvolver diagramas entre outros tipos de desenhos, uma parte das funcionalidades é votada a arquitetura da informação. Se ajusta na plataforma do Google Drive. (FERREIRA,2015).

Está ferramenta foi utilizada para elabora o infográfico e os diagramas UML(*Unified Modeling Language*) deste trabalho.

Figura 14-Draw.IO



Fonte: Autor(2017).

4.1.4 Google Firebase

É uma plataforma da Google, que surgiu no mercado em 2014, onde possui alguns serviços free e suas funcionalidades são todas armazenadas na nuvem. É completa de recurso back-end tanto para o desenvolvimento mobile quanto web. Possui um SDK e um console para criar e gerenciar aplicações. Tem uma variedade de funcionalidades, que facilitam no desenvolvimento do sistema e suporta o desenvolvimento em linguagens como: C++, Java, Javascript, Objective-C, etc. (AVRAM, 2016).

Figura 15- Google Firebase



Fonte: Chat SDK(2016).

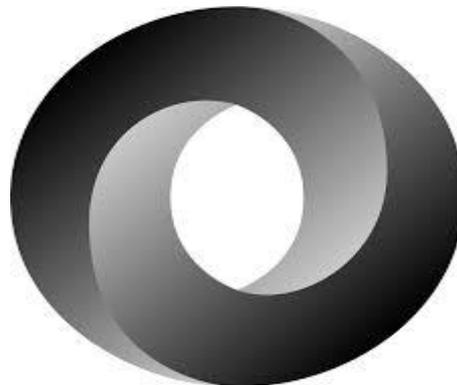
4.1.5 JSON

JSON (*JavaScript Object Notation*) utiliza a forma de texto na hora de realizar a troca de dados entre um navegador e um servidor e, é completamente independente de linguagem, pois usa convenções que são familiares às linguagens C, incluindo C++, C#, Java, JavaScript, Python, entre outras. Estas propriedades fazem com que ele seja um formato ideal de troca de dados. Possui dois tipos de estruturas que são:

- Uma coleção de nome e valor: é caracterizado como um object, record, struct, hash table, keyed list ou arrays associativas.
- Uma lista ordenada de valores: caracterizar-se como uma array, vetor, lista ou sequência.

A maioria das linguagens existentes atualmente as utilizam. Devido seu formato de troca de dados ser independente da linguagem. (JSON,2017?) .

Figura 16-JSON

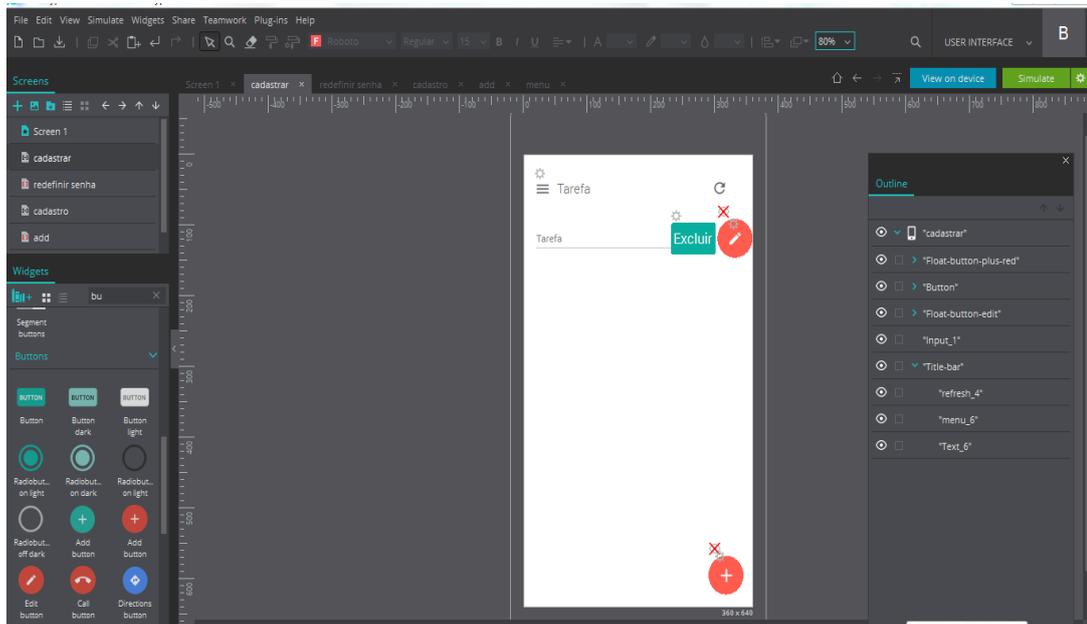


Fonte: JSON(2017?, Online)

4.1.6 Justinmind Prototyper

O *Justinmind Prototyper* permite a elaboração de prototipagem e *wireframes*. Sendo que, qualquer tipo de aplicativo pode ter sua prototipagem elaborada nele. O *wireframe* deste projeto foi elaborado neste software.

Figura 17-Interface do Justinmind Prototyper



Fonte: Autor(2017).

4.1.7 Android

O Android é um sistema operacional usado em dispositivos móveis, como smartphones e tablets Foi desenvolvido por um aliança de empresas liderado pelo Google. (BERGHER, 2017).

A plataforma Android foi utilizada para realizar o teste de funcionamento do aplicativo, onde foi necessário o SDK (*Software Development Kits*) do Android, o qual oferece um kit de ferramentas para ser utilizada no aplicativo.

4.2 BANCO DE DADOS

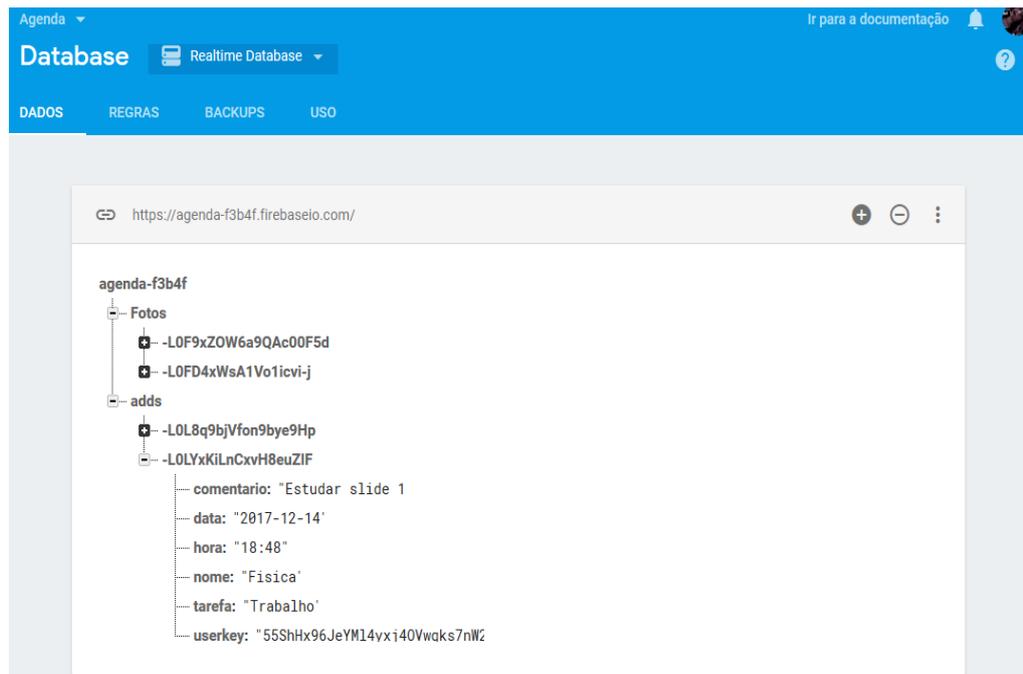
Foi utilizado a plataforma Google Firebase que faz a hospedagem de suas funcionalidades na nuvem, seu banco de dados possui a forma NoSQL (*Not Only SQL*), que utiliza a estrutura de árvore JSON. A plataforma faz a sincronização em tempo real com todos os usuários e a mesma possui serviço de segurança que foram definidas como públicas. O armazenamento dos dados só ocorrem quando os usuários estão conectados em uma rede de internet.

Para o banco de dados foi usado a funcionalidade *Database* da plataforma. O qual é responsável pelo armazenamento dos itens que foram preenchidos no aplicativo e pelas informações do registro por foto, cada vez que é salvo algum item no aplicativo, ele cria um

novo nó na árvore. Em cada nó possui nome, valor e a chave, essa chave é o endereço de cada usuário.

Abaixo será mostrado o ambiente do banco de dados e, como é salvo as informações nele.

Figura 18-Tela do Database do Google Firebase



Fonte: Autor(2017).

Foram definidas duas formas para salvar os dados, uma foi a de foto, a qual salva as informações da foto e a outra é a adds, que salva os dados que foram preenchidos como anotação de texto simples, como é ilustrado na imagem acima.

Além do banco de dados foram usadas outras funcionalidades desta plataforma. Abaixo será citado as funcionalidades que foram usadas no sistema, como:

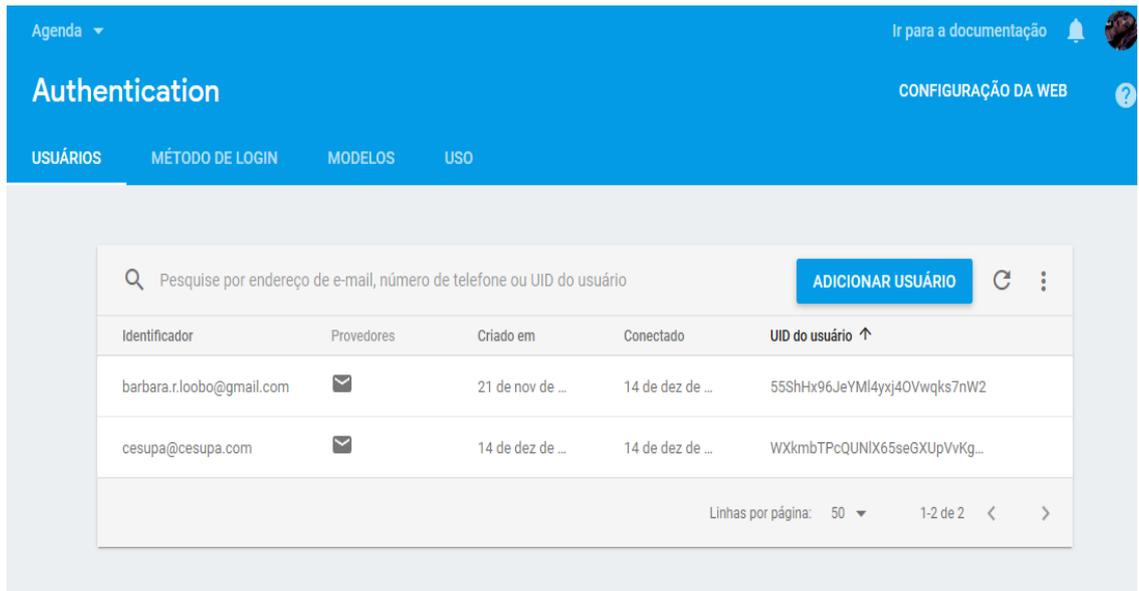
- *Authentication*: é responsável por fazer o armazenamento de cadastros realizados pelos usuários. O cadastro é efetivado com um e-mail e senha, sendo que o e-mail deve conter características definidas pela plataforma. Então todos os e-mails devem possuir “@” e o “.com”. Também é feita uma verificação no e-mail para ver se não possui um igual, se possuir, o cadastro não é validado.

Nesta funcionalidade também tem a opção de redefinir a senha caso o usuário queira mudar ou tenha esquecido, ele deve informar seu e-mail, e ao informar, a própria plataforma irá mandar uma mensagem para o e-mail com um link, esse link irá direcionar o usuário para uma página, onde poderá ser alterada a senha.

- *Storage*: é responsável pelo armazenamento de mídias, ele que realiza o armazenamento de fotos que foram registradas através do aplicativo.

Será ilustrado nas imagens abaixo a tela de *Authentication* e a do *storage*, onde são armazenados os dados pelos quais elas são responsáveis.

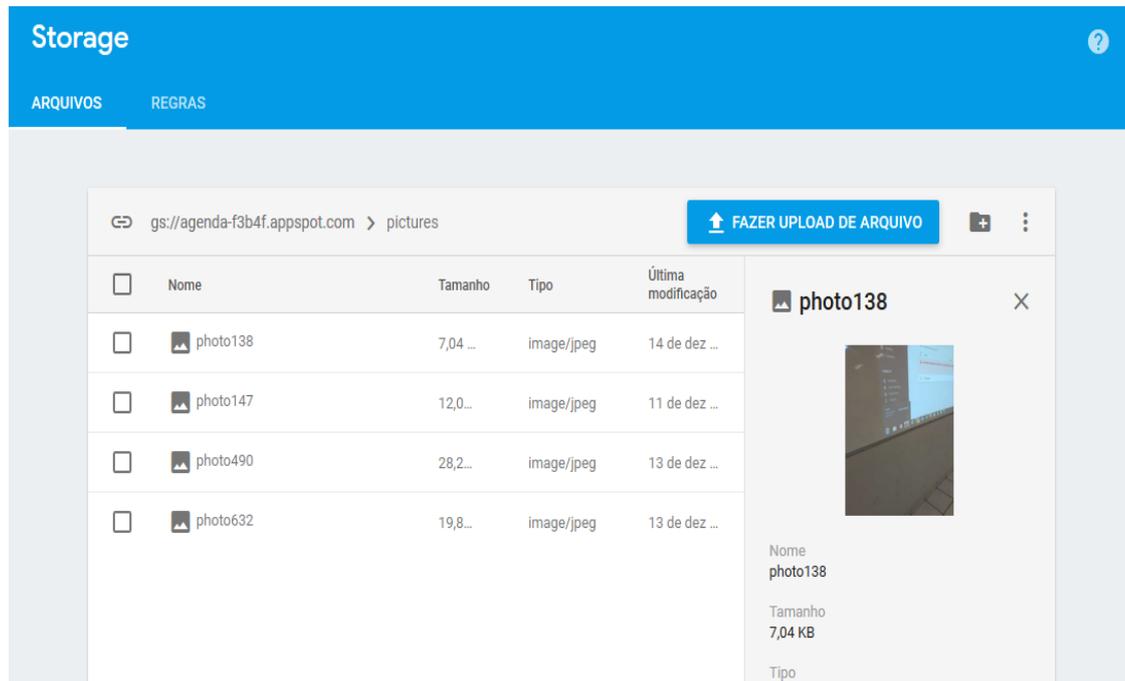
Figura 19- Tela de Authentication do Google Firebase



Fonte: Autor(2017).

Na figura acima é ilustrado como é salvo o e-mail de cada usuário, e, ao realizar o cadastro é fornecido um UID (*User IDentification*), que é a chave de identificação de cada usuário.

Figura 20- Tela do Storage do Google Firebase



Fonte: Autor(2017).

A imagem acima mostra as fotos registradas pelo aplicativo e que automaticamente foram salvas no *storage*.

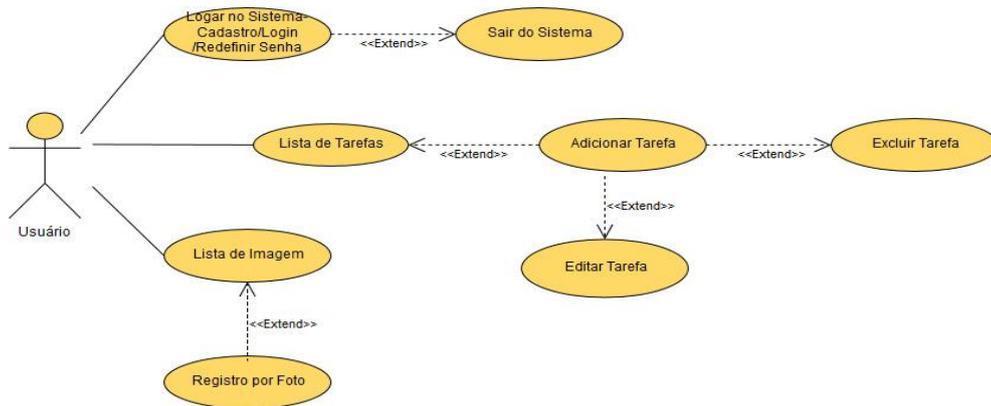
4.3 ARQUITETURA

4.3.1 Diagrama de Caso de Uso

O diagrama de caso de uso apresenta as principais funcionalidades do sistema e a interação delas em relação aos usuários, ou seja, é usado para facilitar a acessibilidade do sistema, mostrando como deve ser usado e quais funcionalidades possuem. (RIBEIRO, 2012?).

O diagrama de caso de uso será ilustrado na figura abaixo.

Figura 21-Diagrama de Caso de Uso do Agenda Acadêmica

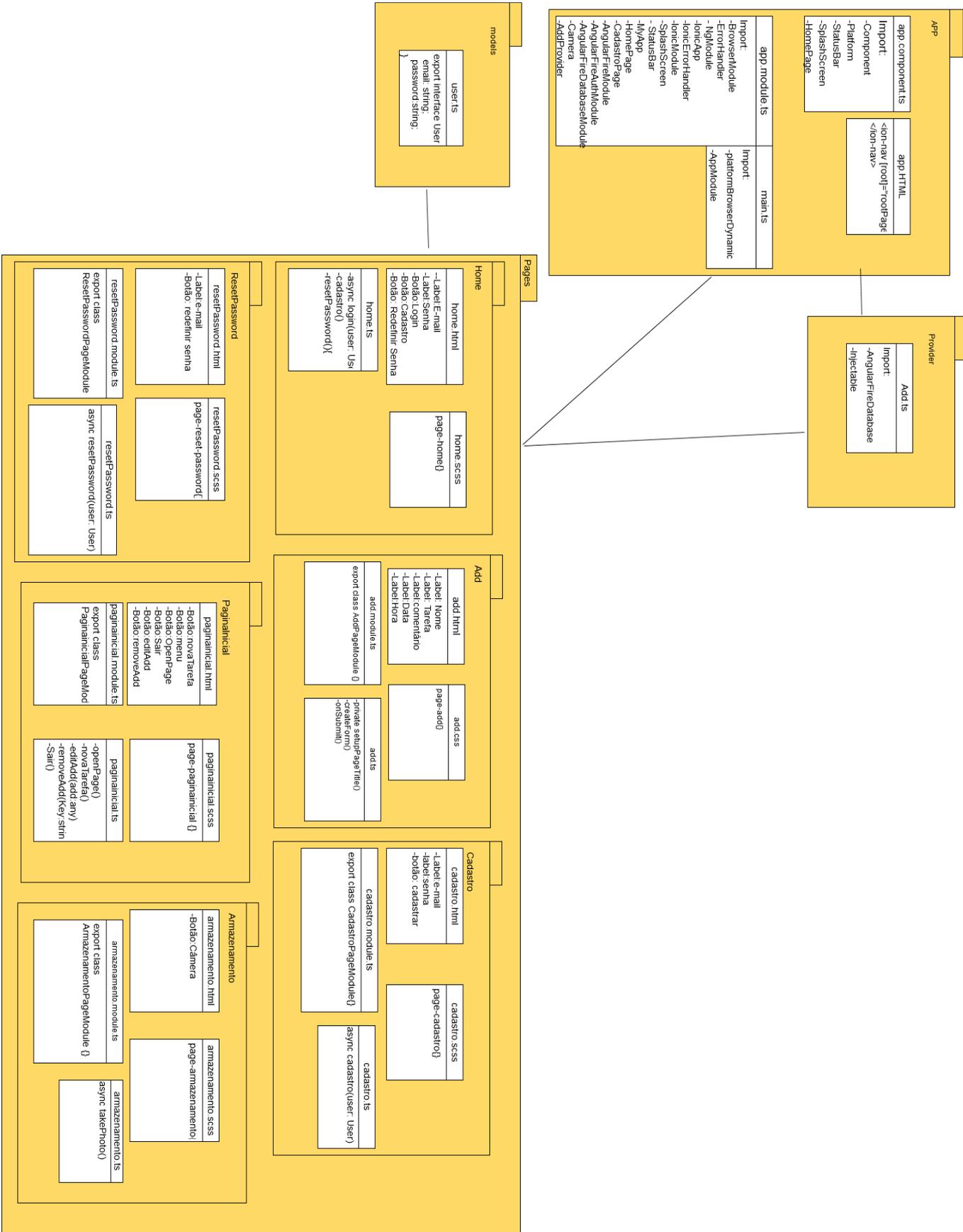


Fonte:Autor(2017).

4.3.2 Diagrama de Pacote

O diagrama de pacotes ou diagrama de módulos, descreve os pacotes do sistema, que são divididos em agrupamentos lógicos, mostrando as dependências entre eles. (OLIVEIRA,2010).

Figura 22-Diagrama de Pacote do Agenda Acadêmica



A figura acima exemplifica o funcionamento no Ionic, mostrando como é feita a definição dos pacotes. Foi utilizado quatro pacotes, sendo o *app*, o *providers*, o *models* e o *pages*.

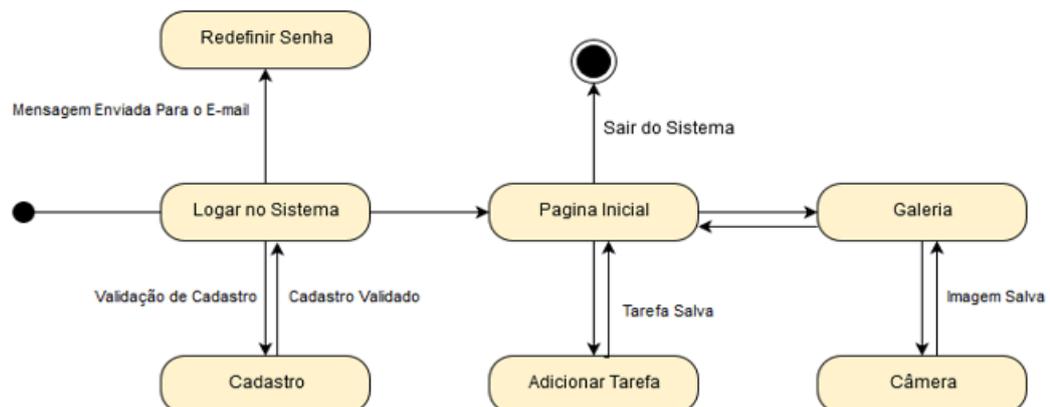
- *App*: possui sentenças que são responsáveis por todo o código elaborado.
- *Providers*: é onde está armazenado o código que faz a conexão com o banco de dados.
- *Models*: é onde está declarado o e-mail e password como string.
- *Pages*: é a definição das páginas que contem no aplicativo.

4.3.3 Diagrama de Estado

O diagrama de estado representa todos estados existentes no sistema, assim como quais eventos levam a passagem de um estado para outro. (ABDALA, 2016?).

A figura abaixo esclarece os estados do sistema.

Figura 23-Diagrama de Estado do Agenda Acadêmica



Fonte:Autor (2017).

A validação do cadastro e a redefinição de senha, são realizadas pela plataforma Google Firebase, pois ele é o responsável pela autenticação de e-mail dos usuários, o mesmo possui suas regras, como já foi citado mais acima na subseção 4.2 do capítulo 4, na página 32 .

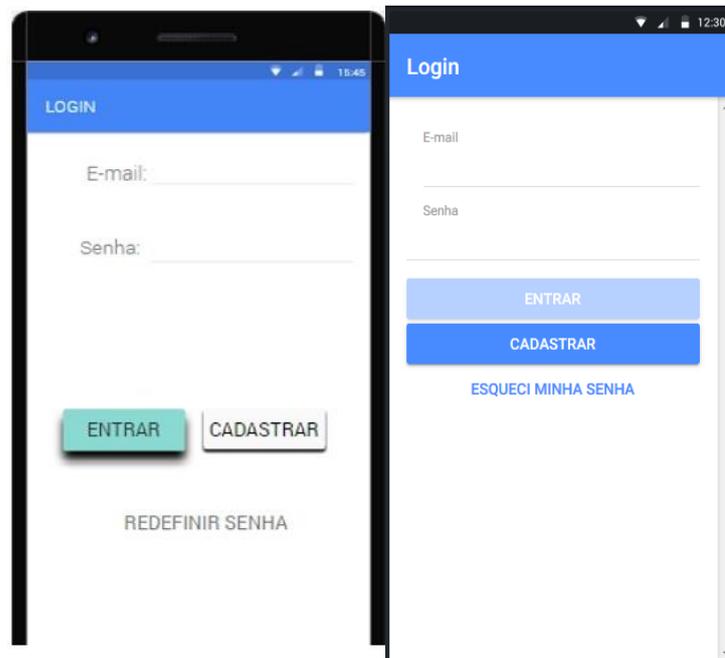
4.4 FUNCIONALIDADES DO SISTEMA

Neste item é apresentado o *wireframe* da *Agenda Acadêmica*, mostrando o planejamento inicial, suas funcionalidades e como provém a comunicação da interface com o usuário. A prototipagem inicial foi realizada no *software Justinmind*. As prototipagens situadas à esquerda são os *design* iniciais e à direita é versão final.

4.4.1 Login

Está página possui como campos iniciais e-mail e senha, o campo de cadastrar, de entrar no sistema, caso já houve o cadastro e de redefinir a senha.

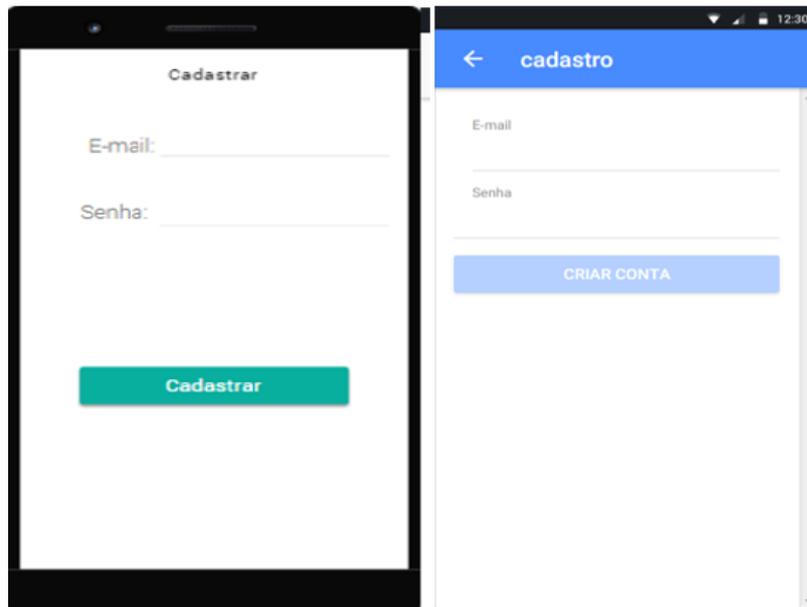
Figura 24-Versão Inicial da Tela de Login e Versão Final.



Fonte: Autor(2017).

4.4.2 Cadastro

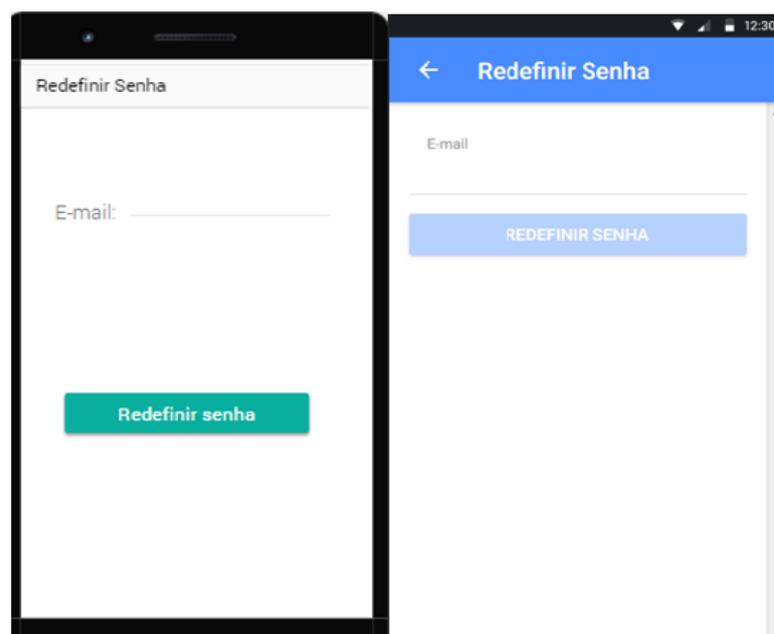
Nesta página ocorre o registro do usuário para que ele possa ter acesso as funcionalidades do aplicativo, e para efetuar o registro é necessário preencher os campos e-mail e senha, mas o usuário só poderá se registrar se caso não existir um e-mail igual ao dele e se possuir as características fornecidas pela plataforma Google Firebase, que já foram citadas mais acima na subseção 4.2 do capítulo 4, na página 33. O cadastro utiliza está plataforma para fazer a autenticação da conta do usuário.

Figura 25-Tela de Cadastro da Versão Inicial e Final

Fonte: Autor(2017).

4.4.3 Redefinir Senha

Esta página é responsável pela redefinição de senha do usuário, caso ocorra o esquecimento de senha. Para que haja a troca é necessário fazer a solicitação usando o e-mail, e uma mensagem será enviada para redefinir a senha.

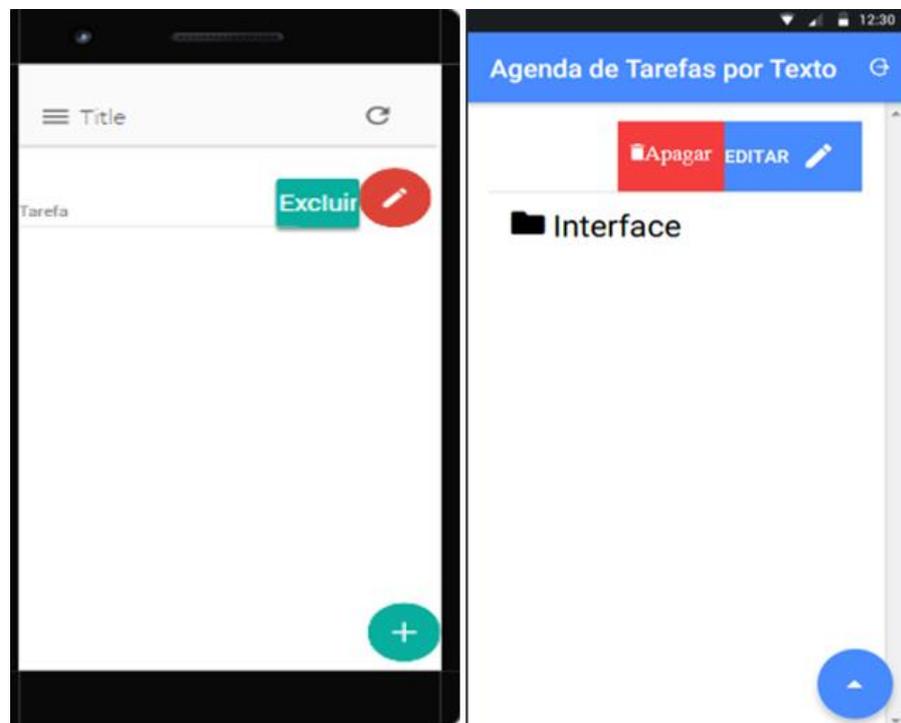
Figura 26-Wireframe Inicial e Versão Final da Tela Redefinir Senha

Fonte: Autor(2017).

4.4.4 Agenda de Tarefas por Foto

Está é a tela principal do aplicativo, através dela é possível ter acesso as funcionalidades do aplicativo. Nela contém um botão flutuante na parte inferior da tela que dá acesso a página de adicionar tarefa e a de agenda de tarefas por foto. As tarefas por texto após serem adicionadas são mostradas nesta tela, ao lado da tarefa possui o botão excluir e editar, para que a tarefa possa ser excluída ou reeditada e também possui o botão de sair, na parte superior da tela.

Figura 27-Prototipagem Inicial e a Versão Final da Tela de Tarefa

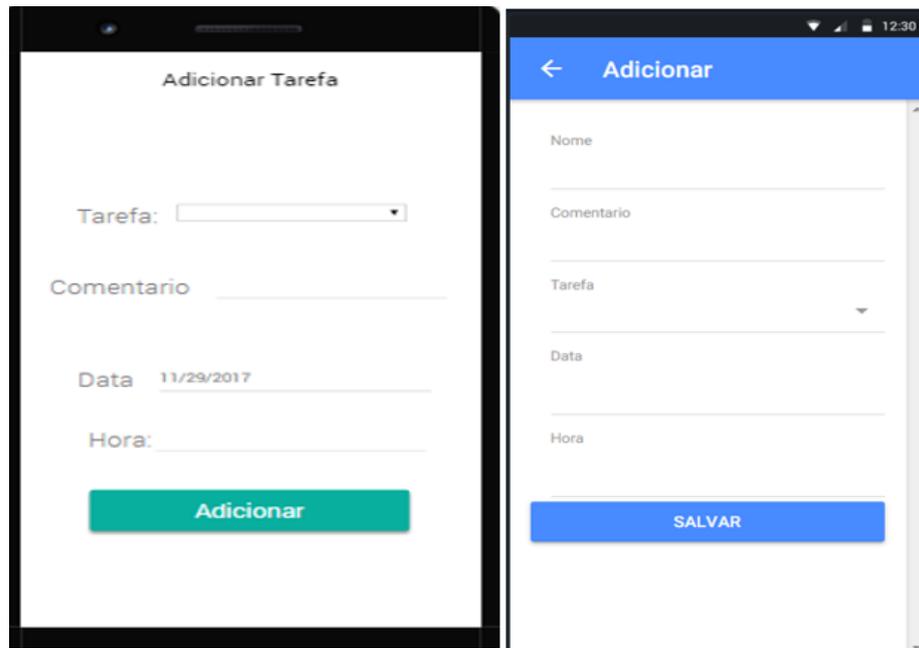


Fonte: Autor(2017).

A figura à direita ilustra a tela de agenda por texto com uma anotação, e, ao lado da anotação os campos de apagar e editar, para ter acesso à esses campos é necessário que deslize a anotação para o lado esquerdo da tela.

4.4.5 Adicionar Tarefa

Está página é o local onde pode ser feita as anotações, através dos campos nome, comentário, data e hora, podendo ser selecionado o tipo de tarefa para a determinada anotação, onde também pode solicitar a notificação do mesmo.

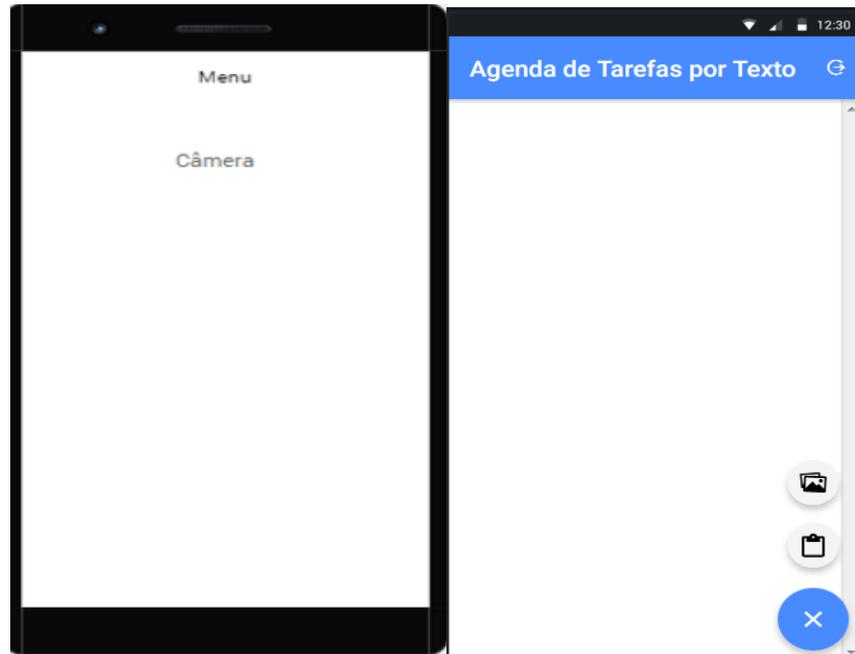
Figura 28-Prototipagem Inicial e a Tela Final da Página Adicionar

Fonte: Autor (2017).

4.4.6 Menu

O menu da prototipagem que está à esquerda, possui os botões de câmera e o de sair do sistema, já na versão final, o menu é um botão que fica na parte inferior da tela, que dá acesso as páginas de adicionar tarefas por texto e da agenda de tarefas por foto.

Figura 29-Prototipagem Inicial e a Tela final do Menu

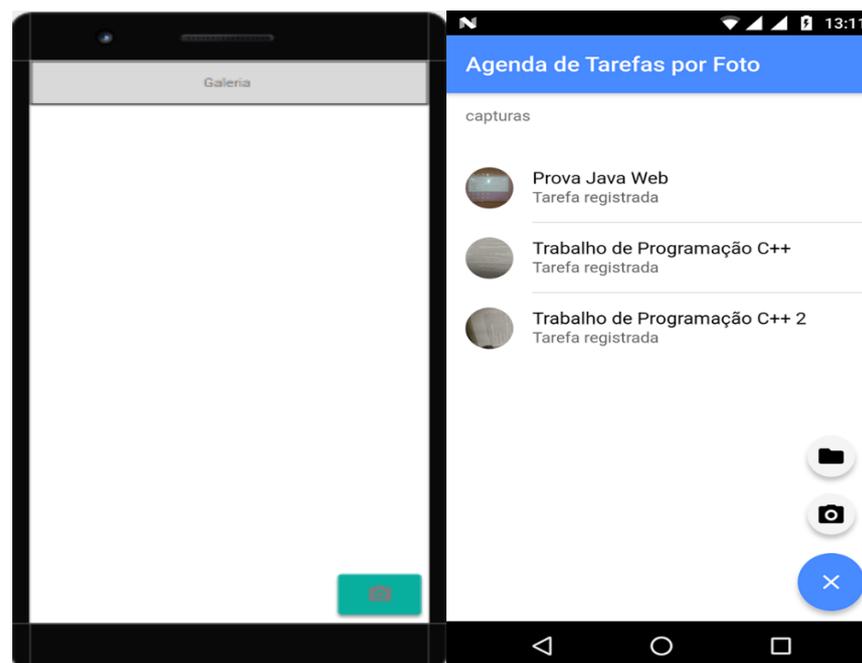


Fonte: Autor(2017).

4.4.7 Agenda de Tarefa por Foto

Esta tela mostra as fotos registradas pela câmera do aplicativo e, na parte inferior da tela possui o botão que dá acesso a câmera e a página de agenda de tarefas por texto.

Figura 30-Prótipagem Inicial e Versão Final da Tela de Agenda de Tarefas por Foto



Fonte: Autor(2017).

5 CONSIDERAÇÕES FINAIS

Este trabalho foi voltado para o desenvolvimento de um aplicativo híbrido, no qual foi projetado para acadêmicos do ensino superior, possibilitando o uso para outros estudantes. O produto desenvolvido, busca ajudar os alunos na organização de suas tarefas, proporcionando um melhor desempenho na sua vida acadêmica.

O desenvolvimento deste trabalho teve um grande ponto positivo, pois ele fez o uso de vários conhecimentos adquiridos durante a graduação, possibilitando um estudo aprofundado aos assuntos que foram envolvidos.

Pode-se concluir que o projeto atingiu seu objetivo, através do teste realizado na plataforma Android pude verificar se as funcionalidades do aplicativo estavam funcionando e se estava fazendo a conexão com o banco de dados, onde são armazenadas as informações que são registradas no aplicativo. Sendo que quase todos estavam funcionando como o esperado, apenas na parte da lista de fotos que houve um problema, as fotos não aparecem no aplicativo se forem puxadas manualmente através do código, pelo fato de não estarem fazendo o download automático.

5.1 DIFICULDADES ENCONTRADAS

No decorrer deste trabalho houve algumas dificuldades encontradas, que foram:

- Na escolha do banco de dados.
- Na autenticação com a plataforma Google Firebase, pelo fato de não conhecer a ferramenta.
- Na atualização das tecnologias, pois iniciaram com uma versão e finalizaram com outra, desta forma, houve uma perda de algumas funcionalidades.
- Na lista de fotos, pois as fotos não estão sendo salvas automaticamente no aplicativo, apenas no *storage* da plataforma Google Firebase.

5.2 TRABALHOS FUTUROS

Com a crescente demanda de aplicativos no mercado, o qual se tornou um fenômeno perante a sociedade, devido suas funcionalidades que facilitam muito durante o dia a dia das pessoas, há propostas para trabalhos futuros, como:

- Desenvolver uma versão para professores;
- Fazer a integração do Agenda Acadêmica ao dos professores, para que haja uma comunicação entre si, onde o professor poderá enviar para o aluno notificações de tarefas.

6 REFERÊNCIAS

ABDALA, Daniel D. **Diagrama de Estados**. Disponível em:<<http://www.facom.ufu.br/~abdala/DAS5312/Diagrama%20de%20Estados.pdf>>. Acesso em 21 de novembro de 2017.

AngularJS. **Extensibility**. Disponível em :< <https://angularjs.org/>>. Acesso em 21 de novembro de 2017.

APACHE CORDOVA. **Overview**. Disponível em :< <https://cordova.apache.org/docs/en/latest/guide/overview/index.html>>. Acesso em 21 de novembro de 2017.

APPSBUYOUT DEV. Agenda Estudante. GOOGLE PLAY. Disponível em :< <https://play.google.com/store/apps/details?id=com.clawdyvan.agendadigitalaluno>>. Acesso em 20 de novembro de 2017.

ATOM, **A Hackable Text Editor For The 21st Century**. Disponível em :< <https://atom.io/>>. Acesso em 21 de novembro de 2017.

AVRAM, Abel. **Google Firebase: back-end completo para aplicações web e mobile**. Disponível em:< <https://www.infoq.com/br/news/2016/07/google-firebase>>. Acesso em 21 de novembro de 2017.

BERGHER, Ricardo. **O que é um celular Android?**. Disponível em:< <https://www.zoom.com.br/celular/deumzoom/o-que-e-um-celular-android>>. Acesso em 23 de novembro de 2017.

BRASIL. Agencia Nacional de Telecomunicações(ANATEL). **Brasil Registra 241 Milhões de Linhas Móveis em Operação em Setembro de 2017**. Disponível em:< <http://www.anatel.gov.br/dados/destaque-1/283-telefonia-movel-registra-aumento-de-156-155-linhas-em-agosto>>. Acesso em :06 de novembro de 2017.

BRASIL. Mobile Time. **1,5 bilhão de smartphones vendidos no mundo em 2016**. Disponível em: < <http://www.mobiletime.com.br/16/02/2017/1-5-bilhao-de-smartphones-vendidos-no-mundo-em-2016/466463/news.aspx>>. Acesso em: 06 de novembro de 2017.

BRASIL. O Globo. **O Primeiro Smartphone do Mundo Completa 20 Anos**. Disponível em:< <https://oglobo.globo.com/sociedade/tecnologia/primeiro-smartphone-do-mundo-completa-20-anos-13630167>>. Acesso em: 06 de novembro de 2017.

BRASIL. Smartphones Tudo. **A Importância dos Smartphones no Dia a Dia**. Disponível em:< <https://smartphonestudo.wordpress.com/2014/02/26/a-importancia-dos-smartphones-no-dia-a-dia/>>. Acesso em : 06 de novembro de 2017.

CHASE, Otavio. **Sistemas Embarcados**. Disponível em: <<http://www.lyfreitas.com.br/ant/pdf/Embarcados.pdf>>. Acesso em: 22 de outubro de 2017.

CHARLAND, Andre ; LEROUX, Brian. **Mobile Application Development: Web vs. Nat. Communications of the ACM**, Vol. 54 No. 5, Pages 49-53. Disponível em:<

<https://cacm.acm.org/magazines/2011/5/107700-mobile-application-development/fulltext>>. Acesso em: 24 de outubro de 2017.

CHAT SDK. **Firestore**. Disponível em :< <https://chatsdk.co/direct-download/>>. Acesso em : 18 de dezembro de 2017.

CIN, Andrea Dal. **Agenda Escolar**. GOOGLE PLAY. Disponível em: < <https://play.google.com/store/apps/details?id=daldev.android.gradehelper> >. Acesso em:20 de novembro de 2017.

FERREIRA,Hélder. **Draw.IO – Desenhar diagramas nunca foi tão fácil**. Disponível em :< <https://pplware.sapo.pt/internet/draw-io-desenhar-diagramas-nunca-foi-tao-facil/> >. Acesso em : 20 de novembro de 2017.

GALDINO, Franclis. **Agenda Universitaria**. GOOGLE PLAY. Disponível em : <<https://play.google.com/store/apps/details?id=br.com.galdino.agendauniversitaria>>. Acesso em : 20 de novembro de 2017.

GSMA. **Number of Mobile Subscribers Worldwide Hits 5 Billion**. Disponível em :< <https://www.gsma.com/newsroom/press-release/number-mobile-subscribers-worldwide-hits-5-billion/>>. Acesso em : 07 de novembro de 2017.

HERMAN, Dave; MENDES, Gabriela, SCRIPT,Thiago. **Starts and ends with JavaScript**. Disponível em: < <http://www.typescriptlang.org/> >. Acesso em : 20 de novembro de 2017.

IONIC. **Native and Web Optimized**. Disponível em:< <https://ionicframework.com/>>. Acesso em: 23 de novembro de 2017.

JSON. **Introducing JSON**. Disponível em:< <https://www.json.org/>>. Acesso em: 23 de novembro de 2017.

MDN WEB DOCS.CSS. Disponível em:< <https://developer.mozilla.org/pt-BR/docs/Web/CSS>>. Acesso em: 23 de novembro de 2017.

NodeBR. **O que é a NPM do Node.JS**. Disponível em:< <http://nodebr.com/o-que-e-a-npm-do-nodejs/>>. Acesso em: 23 de novembro de 2017.

OLIVEIRA, Wellington. **UML - Diagrama de Pacotes**. Disponível em:< <https://pt.slideshare.net/softpalm/uml-diagrama-de-pacotes>>. Acesso em: 23 de novembro de 2017

PEREIRA, Luiz Arthur Malta et al. **Software Embarcado, o Crescimento e as Novas Tendências deste Mercado**. REVISTA DE CIÊNCIAS EXATAS E TECNOLOGIA. Disponível em: < <http://www.pgsskroton.com.br/seer/index.php/rcext/article/view/2308>>. Acesso em: 22 de outubro de 2017.

RODRIGUES, Joel. **HTML básico - códigos HTML**. Disponível em: < <https://www.devmedia.com.br/html-basico-codigos-html/16596>>. Acesso em: 22 de outubro de 2017.

RIBEIRO, Leandro. **O que é UML e Diagramas de Caso de Uso: Introdução Prática à UML**. Disponível em: < <https://www.devmedia.com.br/o-que-e-uml-e-diagramas-de-caso-de-uso-introducao-pratica-a-uml/23408> >. Acesso em: 23 de novembro de 2017.

SILVA, Leandro Luquetti B. et al. **Desenvolvimento de Aplicações para Dispositivos Móveis:** Tipos e Exemplo de Aplicação na plataforma iOS. Goiânia - GO, 29 de Maio de 2015. Disponível em: < <http://www.lbd.dcc.ufmg.br/colecoes/wicsi/2015/004.pdf>>. Acesso em 20 de outubro de 2017.

SILVA, Marcelo Moro; SANTOS, Marilde Terezinha Prado. **Os Paradigmas de Desenvolvimento de Aplicativos para Aparelhos Celulares.** Disponível em:< [https://pt.scribd.com/document/321459731/Artigo->](https://pt.scribd.com/document/321459731/Artigo-). Acesso em: 20 de outubro de 2017.

THEOHARIDOU, Marianthi; MYLONAS, Alexios; GRITZALIS, Dimitris. **A Risk Assessment Method for Smartphones.** Disponível em :< https://link.springer.com/content/pdf/10.1007/978-3-642-30436-1_36.pdf>. Acesso em: 22 de outubro de 2017.

VAHID, Frank; GIVARGS, Tony. **Embedded System Design:** A Unified Hardware/Software. Versão Preliminar. Califórnia: University of California,1999.

WHITE, James. **Going native (or not): Five questions to ask mobile application developers.** Disponível em< <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3575060/>>. Acesso em: 23 de outubro de 2017.