



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E
TECNOLOGIA DE MINAS GERAIS – CAMPUS FORMIGA
CURSO TÉCNICO INTEGRADO EM INFORMÁTICA

**ESTUDO DO FRAMEWORK IONIC 3 PARA DESENVOLVIMENTO DE
APLICATIVO HÍBRIDO**

Projeto Orientado de Curso – Técnico em Informática

Iago Brayham Cunha Santos

Orientador(a): Prof^a. Dr^a. Paloma

Maira de Oliveira Lima

FORMIGA

2018

IAGO BRAYHAM CUNHA SANTOS

**ESTUDO DO FRAMEWORK IONIC 3 PARA DESENVOLVIMENTO DE
APLICATIVO HÍBRIDO**

Relatório final apresentado como requisito parcial à obtenção do título de Técnico em Informática, Curso Técnico em Informática Integrado ao Ensino Médio do IFMG – Formiga.

Orientador(a): Profa. Dra. Paloma Maira de Oliveira Lima

Formiga – MG

2018

IAGO BRAYHAM CUNHA SANTOS

**ESTUDO DO FRAMEWORK IONIC 3 PARA DESENVOLVIMENTO DE
APLICATIVO HÍBRIDO**

Relatório final apresentado como requisito parcial à obtenção do título de Técnico em Informática, Curso Técnico em Informática Integrado ao Ensino Médio do IFMG – Formiga.

BANCA EXAMINADORA

Prof^a. Dr^a. Paloma Maira de Oliveira Lima
ORIENTADORA

Prof. Ms. Fernando Paim Lima
EXAMINADOR

Chrystian Rezende
EXAMINADOR

IFMG - FORMIGA
Formiga, 14 de novembro de 2018.

A todos aqueles que contribuíram para a realização deste trabalho.

AGRADECIMENTOS

Agradeço a minha mãe e irmãos, que me deram o apoio emocional para continuar no presente curso e finalizar o exposto projeto.

À minha orientadora, Profa. Dra. Paloma Maira de Oliveira Lima, que me guiou na realização deste trabalho e me deu todo o apoio acadêmico possível.

Aos avaliadores da banca por terem aceitado o convite de prestigiar, analisar e julgar o meu trabalho.

A todos os professores que fizeram com que eu tivesse o conhecimento necessário para dar início às pesquisas e desenvolvimento do projeto.

“A tecnologia move o mundo.”

Steve Jobs

RESUMO

O presente trabalho é um estudo, o qual visa a compreensão e utilização da ferramenta de desenvolvimento de aplicativos móveis híbridos, Ionic 3, juntamente à Angular 5, para o desenvolvimento de um aplicativo móvel multiplataforma. O aplicativo em questão é uma ferramenta para facilitar a comunicação entre os indivíduos de uma organização, não deixando de lado o sistema hierárquico ali constituído. Além disso, o aplicativo proposto também disponibiliza uma agenda de eventos, para a interação dos usuários com as ações planejadas nessas organizações (reuniões, eventos culturais, *etc.*) e a ele atribuídas. Ao final do trabalho, é apresentada uma análise do custo-benefício quanto à curva de aprendizagem das ferramentas utilizadas no desenvolvimento do projeto sob a ótica do autor.

Palavras-chave: Ionic 3; Angular 5; Aplicativos híbridos; Desenvolvimento móvel.

ABSTRACT

The present work concerns about the comprehension and use of the framework Ionic 3 alongside Angular 5, for the development of a real cross-platform mobile app. Such developed application works as a tool for helping with the communication between subjects from a specific group as well as considering that group's hierarchy. Besides that, the proposed app also brings an events calendar, so that its users can interact with previously planned actions of the group, such as meetings, cultural events and so on. At the end of this article, a cost-benefit assessment is discussed regarding the learning curve of these frameworks in the development of this application, during the present work, from its author's point of view.

Key-words: Ionic 3. Angular 5. Hybrid apps. Mobile development.

LISTA DE FIGURAS

FIGURA 1 – DIAGRAMA DE CASO DE USO – MÓDULO AGENDA	18
FIGURA 2 – DIAGRAMA DE CASO DE USO – MÓDULO MENSAGEIRO	19
FIGURA 3 – EXEMPLO DE HIERARQUIA EM UMA EMPRESA.....	20
FIGURA 4 – MODELO DE BANCO DE DADOS	22
FIGURA 5 – MODELO DE BANCO DE DADOS EM ESTRUTURA NOSQL APLICADA NO FIRESTORE.....	23
FIGURA 6 – GUI PRINCIPAL (FEED DE EVENTOS.....	24
FIGURA 7 – GUI DE CRIAÇÃO DE EVENTO.....	25
FIFGURA 8 – GUI DE DETALHES DO EVENTO	25
FIGURA 9 – GUI DE CONVERSAS	26
FIGURA 10 – GUI DE ENVIO DE MENSAGEM.....	26
FIGURA 11 – GUI DE LISTA DE CONTATOS.....	26

LISTA DE ABREVIATURAS E SIGLAS

APP – Aplicativo/aplicação

CSS – Folhas de Estilo em Cascata

CSU – Caso de Uso

DCSU – Diagrama de Caso de Uso

GPS – Sistema de Posicionamento Global

GUI – Interface Gráfica do Usuário

HTML – Linguagem de Marcação de Hipertexto

IDE – Ambiente de Desenvolvimento Integrado

JSON – Notação de Objeto JavaScript

JSX – Jakarta Stock Exchange

NOSQL – Linguagem de Consulta Não Relacional

POC – Projeto Orientado de Curso

SDK – Kit de Desenvolvimento de Software

SO – Sistema Operacional

SQL – Linguagem de Consulta Estruturada

UI – Interface do Usuário

UML – Linguagem Unificada de Modelagem

UX – Experiência do Usuário

WEB – Rede mundial de computadores (internet)

XML – Linguagem de Marcação Extensível

SUMÁRIO

1 INTRODUÇÃO	12
2 MATERIAIS E MÉTODOS	14
2.1 MODELAGEM DE DADOS.....	14
2.2 BANCO DE DADOS	14
2.3 CODIFICAÇÃO.....	15
2.4 CONTROLE DE VERSÃO	16
2.4 EQUIPAMENTOS UTILIZADOS	17
3 ANÁLISE E PROJETO	18
3.1 DIAGRAMA DE CASO DE Uso.....	18
3.2 REGRAS DE NEGÓCIO	19
3.3 BANCO DE DADOS.....	22
4 O APLICATIVO DESENVOLVIDO	24
5 CONSIDERAÇÕES FINAIS	27
6 REFERÊNCIAS BIBLIOGRÁFICAS	28

1 INTRODUÇÃO

De acordo com uma matéria publicada pela Gartner, Inc. - líder em consultoria de tecnologia relacionada a tomada de decisões no mercado tecnológico – em muitas empresas, a demanda por desenvolvimento móvel cresce 5 (cinco) vezes mais rápido do que as equipes de desenvolvimento de aplicativos conseguem entregar [NEWSROOM, 2015]. Ao se desenvolver aplicativos da forma convencional (que funcionam apenas para determinado sistema operacional/dispositivo), requer-se especialistas para escrever código nativo – além de construir um design, realizar testes e fazer sua manutenção; o que acaba demandando tempo, sendo muitas vezes caro e ineficiente. Caso precisasse de um aplicativo que rode em diferentes plataformas, será necessário desenvolvê-lo mais de uma vez, com uma base de código diferente. Para evitar os problemas acima mencionados, surge o desenvolvimento de aplicativos híbridos, ou seja, aplicativos *cross-platform* (multiplataforma). Tais aplicações possuem benefícios como [INTERON SOLUTIONS, 2018]:

- Única base de código multiplataforma;
- Utiliza-se design web responsivo;
- Possui acesso a recursos do sistema operacional do dispositivo;
- Maior capacidade de uso *off-line*.

Atualmente, no mercado duas tecnologias se destacam: o **React Native** – que utiliza JSX (pré-processador que adiciona XML – Linguagem Extensível de Marcação - à sintaxe do JavaScript - linguagem de programação baseada em *scripts*) [GITHUB PROJECT, 2018] e o **Ionic** – que utiliza TypeScript (um superset de JavaScript que permite a escrita de *scripts* com a utilização de tipagem estática, orientação a objetos, o qual facilita a escrita de código com uma sintaxe de fácil compreensão) [ONLINE PRESENTATION, 2018].

Neste contexto, este projeto orientado de curso (POC) tem **dois principais objetivos**:

- I. Estudar um *framework* que possibilite o desenvolvimento de aplicações híbridas. Para isso, foi selecionado o Ionic, por sua

sintaxe simplificada, como também pela disponibilidade de documentação e seus diversos componentes “pré” prontos e estilizáveis.

- II. Desenvolver um aplicativo híbrido usando o Ionic, com intuito de facilitar a comunicação entre os indivíduos de uma organização, não deixando de lado o sistema hierárquico ali constituído. Além disso, o aplicativo proposto também disponibiliza uma agenda de eventos, para a interação dos usuários com as ações planejadas nessas organizações (reuniões, eventos culturais, etc.) e a ele atribuídas.

A organização deste relatório está conforme a seguir: O Capítulo 2 apresenta os materiais e métodos utilizados, enquanto o Capítulo 3 apresenta a Análise e projeto do aplicativo proposto. Já o Capítulo 4, detalha o aplicativo desenvolvido e, finalmente, no Capítulo 5, são apresentadas as considerações finais.

2 MATERIAIS E MÉTODOS

Para o desenvolvimento do aplicativo proposto, várias tecnologias foram utilizadas, as quais são listadas a seguir.

2.1 Modelagem de Dados

- ADOBE EXPERIENCE DESIGN (5.0.X) – ADOBE: *Software* que integra *design* e prototipagem de aplicativos em uma aplicação desktop [TEIXEIRA, 2016]. Utilizado no projeto para desenvolvimento de *mockups* (visão de GUI's) do aplicativo, foi de suma importância para dar uma “cara” ao aplicativo, de modo inicial, e para discussões de projeto quanto à interação do usuário com o sistema.
- DIA – Aplicativo para desenhos de diagramas [DIA WEBSITE, 2018]. Foi utilizado para desenhar o DCSU's do aplicativo desenvolvido neste projeto.
- UML – A Linguagem Unificada de Modelagem é um padrão para visualização gráfica, especificação, construção e documentação de sistemas de software [BOOCH, 2006]. Foi utilizada para definir os planos de arquitetura e de negócio do projeto de desenvolvimento de aplicativo, exposto neste trabalho, visando o entendimento de ações entre usuário e sistema, além do banco de dados.

2.2 Banco de Dados

- MYSQL WORKBENCH - ferramenta para visualização de arquitetura de bancos de dados SQL, que permite a modelagem de dados [MCLAUGHLIN, 2013]. Neste trabalho, foi de suma importância para a arquitetura do banco de dados da aplicação desenvolvida e para visualização dos dados nela contidos.
- CLOUD FIRESTORE - FIREBASE (GOOGLE API'S): Ferramenta de banco de dados NoSql. O *Cloud Firestore* é um banco de dados flexível e escalável para desenvolvimento de dispositivos móveis, Web e servidores a partir do *Firebase* e do *Google Cloud Platform* [YAHIAOUI, 2017]. Foi utilizada no trabalho como servidor em tempo real para o banco de dados,

ou seja, fluxo de dados de todo o aplicativo no que tange à interação *on-line* e *off-line*, o que possibilitou a interação entre usuários e informações do sistema.

2.3 Codificação

- HTML - Linguagem de Marcação de Hipertexto: é uma linguagem utilizada na construção de páginas web, fazendo sua estruturação [RAMALHO, 1967]. Foi utilizada no projeto como base para a criação das páginas do aplicativo, juntamente à ferramenta Angular, sendo responsável pela UI do app.
- CSS - É um mecanismo que adiciona estilos a elementos de páginas *web* [RAMALHO, 1967]. Foi utilizado para controlar os estilos das páginas, como cores, tamanhos, animações e aspectos visuais das GUI's.
- JAVASCRIPT - *Linguagem* de programação de *scripts*, principal linguagem utilizada neste POC, pois codifica a execução do aplicativo desenvolvido [FERREIRA, 2018]. Apesar da linguagem em si utilizada no projeto ser TypeScript, a mesma é um *superset* da linguagem JavaScript, ou seja, ela é transformada em JavaScript para execução. Assim, foram utilizados conceitos de JavaScript para todas as ações feitas no aplicativos (codificação), de modo a controlar interações do usuário com o aplicativo, além de controlar, também, o *back-end* das operações com o servidor, ali travadas.
- NODE.JS (9.8.0) - é um interpretador de código JavaScript que funciona do lado do servidor. Ajuda na criação de aplicações de alta escalabilidade (como um servidor *web*) [RUBENS, 2017]. Foi utilizado no projeto, como parte do *framework* Ionic 3, dando suporte ao mesmo para gerenciamento de pacotes da aplicação, além de bibliotecas e *boiler-plate's*.
- IONIC (3.X.X) - *Framework* de desenvolvimento de aplicativos móveis, de código aberto, que facilita o processo utilizando tecnologias da *web* [RAVULAVARU, 2017]. Principal ferramenta do projeto, foi utilizado para criação, depuração e manipulação do projeto, oferecendo componentes pré-feitos que ajudaram na velocidade de codificação, além de seu site oficial oferecer diversos tutoriais e casos de uso como embasamento para

desenvolvimento. Desta forma, foi possível o desenvolvimento da aplicação com curva de aprendizagem média, tendo o autor do exposto projeto tido de estudar o *framework* e seu assessor Angular, para alcançar os objetivos do aplicativo.

- ANGULAR (5.2.8): Google - *Framework* JavaScript de código aberto, que auxilia na execução de aplicações *web*. Ionic 3 conta com o Angular no processo de desenvolvimento de *apps* [RAVULAVARU, 2017]. Foi utilizado no projeto como controle entre o código e a parte visual (HTML), além de ser responsável pela compilação do código JavaScript para comunicação com o sistema.
- APACHE CORDOVA (7.1.0): Adobe – *Framework* que utiliza Javascript para acessar recursos nativos do dispositivo. O Ionic foi construído em cima dele, que serve de camada inferior para transformação do aplicativo em pacotes para cada sistema operacional [ONLINE PRESENTATION, 2018]. Esta ferramenta é utilizada no projeto para acesso às API's do SO, como GPS, imagens, câmera, áudio, e outros recursos que normalmente seriam nativos.
- VISUAL STUDIO CODE (1.21.0): Microsoft - Editor de código-fonte [DEL SOLE, 2017]. Foi utilizado neste projeto como editor de todo o código fonte desenvolvido. Além disso, o seu terminal embutido foi utilizado para executar comando de execução do Ionic

2.4 Controle de versão

- GIT (2.16.2): Sistema de controle de versão distribuído e sistema de gerenciamento de código, utilizado para fazer conexão com o Gitlab [BEL, 2015]. Foi utilizado, então, para salvar uma versão *backup* do sistema em um servidor na nuvem.
- GITLAB: Gerenciador de repositório de *software*. [HETHEY, 2013]. Foi utilizado, neste projeto, como servidor para as operações em *git*, funcionando como repositório em nuvem para *backup* do projeto.

2.5 Equipamentos utilizados

Sobre os dispositivos que foram utilizados para o desenvolvimento e testes do aplicativo, destacam-se:

- DISPOSITIVO ANDROID PARA TESTES REAIS – Moto G5s Plus ou similar – Android 8.0 [MOTOROLA WEBSITE, 2018];
- DISPOSITIVO IOS PARA TESTES REAIS – iPhone 5s ou similar – IOS 11 [APPLE WEBSITE, 2018];
- NOTEBOOK – DELL INSPIRON 15 5000 – Processador: i7-7500U, x64; Memória RAM: 8GB; Capacidade: 1TB; Sistema Operacional: Windows 10 Pro [DELL WEBSITE, 2018].

3 ANÁLISE E PROJETO

Para o entendimento do escopo do aplicativo e planejamento das atividades, foram desenvolvidos: um diagrama de caso de uso (Seção 3.1), as regras de negócios (Seção 3.2) e o modelo lógico do banco de dados (Seção 3.3).

3.1 Diagrama de Caso de Uso

Como o projeto foi ideia de um cliente real, foram realizadas reuniões com esse cliente, para definição do escopo. A Figura 1 e 2 apresenta os diagramas de casos de uso (DCSU) desenvolvidos, nos quais constam as funcionalidades de alto nível do app desenvolvido.

Conforme observa-se nas Figuras 1 e 2, o sistema proposto possui três atores: Administrador, Líder e Participante. Ressalta-se que os usuários podem exercer tais papéis de acordo com o contexto ao qual está inserido. Isso é demonstrado através do relacionamento de herança entre os atores.

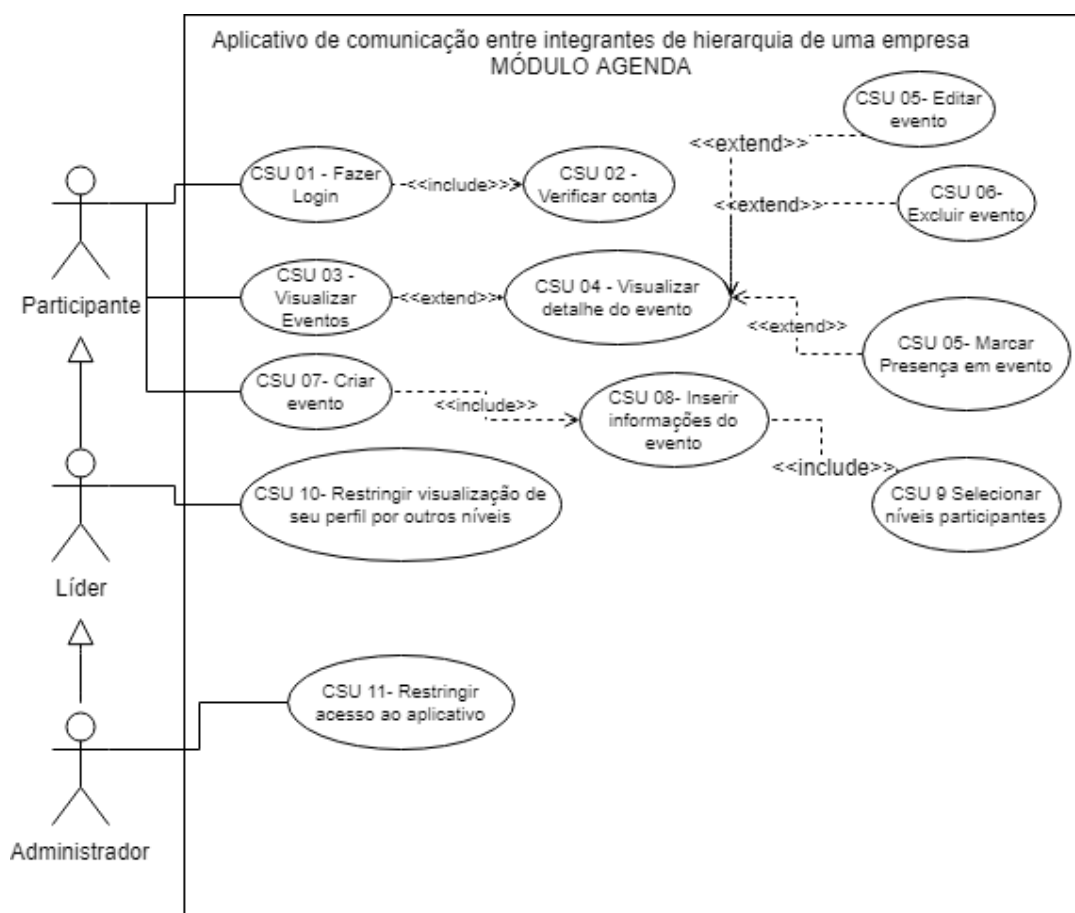


FIGURA 1 - Diagrama de caso de uso do módulo Agenda

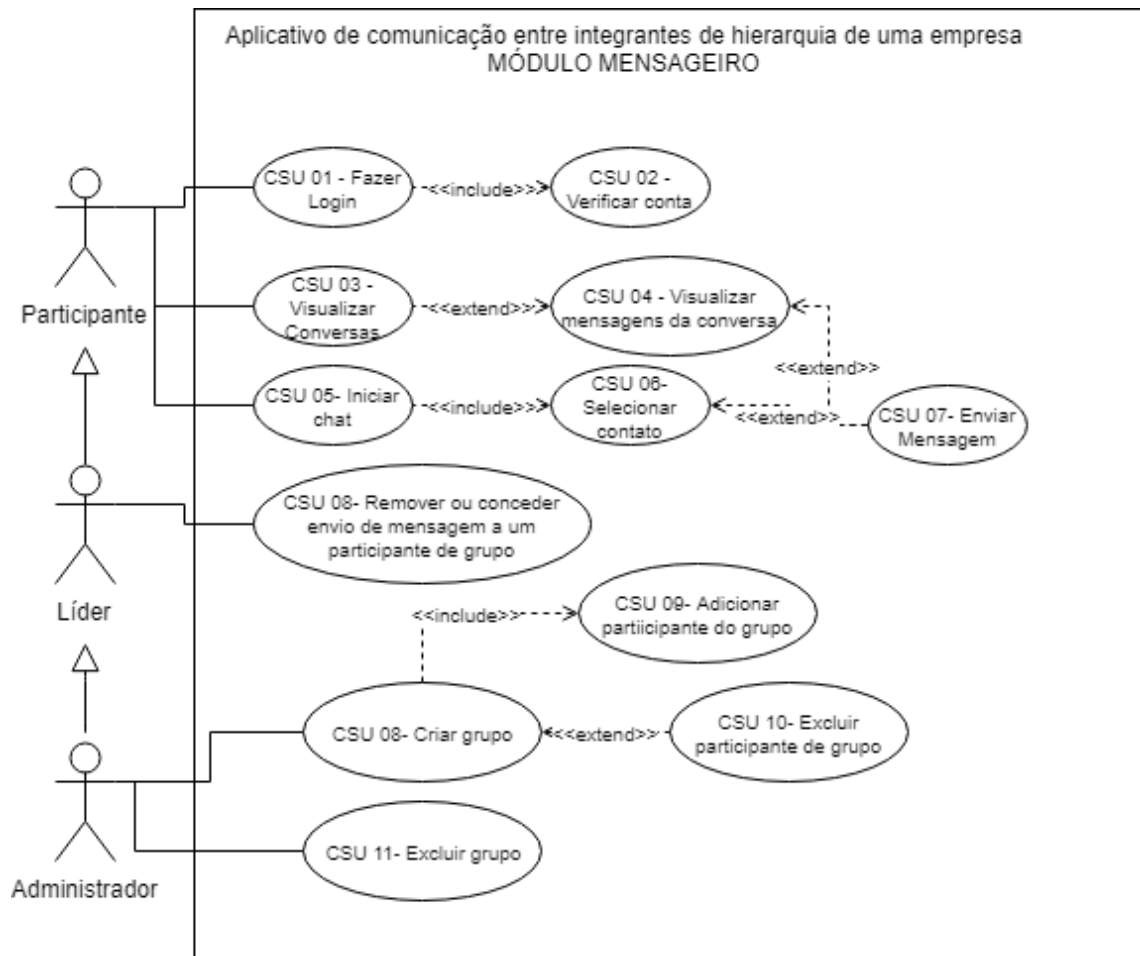


FIGURA 2- Diagrama de caso de uso do módulo Mensageiro

O primeiro DCSU (Figura 1) é composto por 11 casos de uso (CSU's), sendo funcionalidades relacionadas ao módulo agenda, enquanto que o segundo DCSU (Figura 2) é composto por 11 casos de uso (CSU's), sendo funcionalidades relacionadas ao módulo mensageiro.

3.2 REGRAS DE NEGÓCIO

Como mencionado anteriormente, o objetivo do trabalho é o desenvolvimento de um aplicativo que facilite a interação de integrantes de grupos nos quais uma hierarquia é constituída. Assim, foi definido dois módulos para o sistema, denominados: Mensageiro e Agenda. Ambos os módulos devem levar em consideração a hierarquia constituída na organização que irá utilizar o app. As subseções a seguir, detalham as funcionalidades (CSU's) de cada módulo.

3.2.1 MENSAGEIRO

Esse módulo foi inspirado em aplicativos como *Whatsapp*¹ e *Messenger*². Deve ser utilizado para troca de mensagens de texto em tempo real, além de vídeos, fotos e áudios, através de uma conexão com a internet. Frisa-se que o envio de arquivo é permitido desde que não ultrapasse o armazenamento permitido pelo banco de dados *Google Cloud Storage*.

O usuário contará com uma lista de contatos para os quais ele poderá se comunicar. Além disso, ele poderá visualizar as informações pessoais do contato selecionado, disponibilizadas pelo mesmo.

3.2.2 AGENDA

O módulo agenda foi inspirado no aplicativo *Google Calendar*. O usuário poderá, em um único app, visualizar os eventos disponíveis e destinados a ele, podendo marcar presença e verificar os detalhes do evento em questão. Além disso, ele poderá criar um novo evento, caso tenha tal permissão.

3.2.3 CONTEXTUALIZANDO

Para melhor entendimento do leitor do aplicativo proposto, levaremos em conta o cenário do IFMG - FORMIGA, onde uma hierarquia é constituída entre os funcionários da área de ensino, exemplificada pela Figura 3.

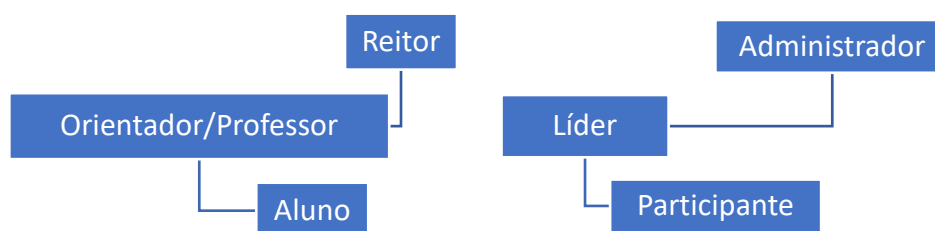


FIGURA 3 – Hierarquia do IFMG – Campus Formiga

Deste modo, percebe-se a possibilidade de interação entre esses níveis, que poderá ser feita através de um recurso mensageiro do aplicativo desenvolvido, o qual será explicado mais à frente. Tal interação é moderada pelo próprio usuário, por exemplo: o **reitor** pode não querer contato direto com os

¹ <https://www.whatsapp.com>

² <https://www.messenger.com>

alunos, cabendo a ele permitir, através de configurações no aplicativo, se seu perfil estará disponível para determinado nível. Caso o reitor queira esse contato, mas não deseje que suas informações pessoais como: telefone ou endereço estejam visíveis para usuários de determinado nível, assim poderá configurar, no próprio aplicativo.

Visto isso, pode-se fazer uma série de restrições quanto à comunicação entre os atores desta instituição, no sistema a ser desenvolvido, tornando-o parametrizável. Ou seja, tal afirmação é levada em conta: os integrantes dos níveis podem comunicar-se entre si, individualmente desde que o outro integrante permita tal comunicação.

Com acesso pleno, o nível master (Administrador), é quem poderá restringir o acesso ao aplicativo, ou seja, é quem permitirá o uso dos níveis e seus integrantes ao aplicativo.

Já a criação de eventos, poderá ser feita pelo ator permitido, seguindo o DCSU do módulo **agenda** (Figura 1), o qual poderá selecionar quais níveis poderão participar do evento criado, permitindo aos integrantes marcarem presença nesses eventos. Os usuários poderão visualizar tais eventos através de um *feed* (mural), para que se, convidados, possam ver os detalhes do evento e então marcar presença no mesmo. Poderão os Administradores convidados, conforme DCSU do módulo **mensageiro** (Figura 2), criarem grupos de conversa a partir de tal evento, podendo adicionar outros usuários, com acesso àquele evento, ao grupo. Além disso, poderão os líderes adicionados ao grupo, restringirem o envio de mensagens de participantes comuns, ou seja, permitir ou não o envio de mensagens por esses. Ressalta-se que, o criador do grupo será o seu moderador, de forma a poder adicionar ou excluir integrantes, bem como restringir o envio de mensagens de outros participantes do grupo.

Desta forma, presa-se a ideia inicial de interação entre pessoas, visando a hierarquia da organização e o controle de eventos ali desenvolvidos.

3.3 BANCO DE DADOS

3.3.1 MODELO DO BANCO DE DADOS

A Figura 4 apresenta o modelo lógico do banco de dados. Esse modelo foi criado para facilitar a visualização do relacionamento dos dados tratados no aplicativo, no que diz respeito aos usuários, mensagens e eventos. Assim, foi possível, a partir desse modelo, pensar em uma solução para a implementação dos dados em estrutura *NoSql*, utilizando a ferramenta *Cloud Firestore*.

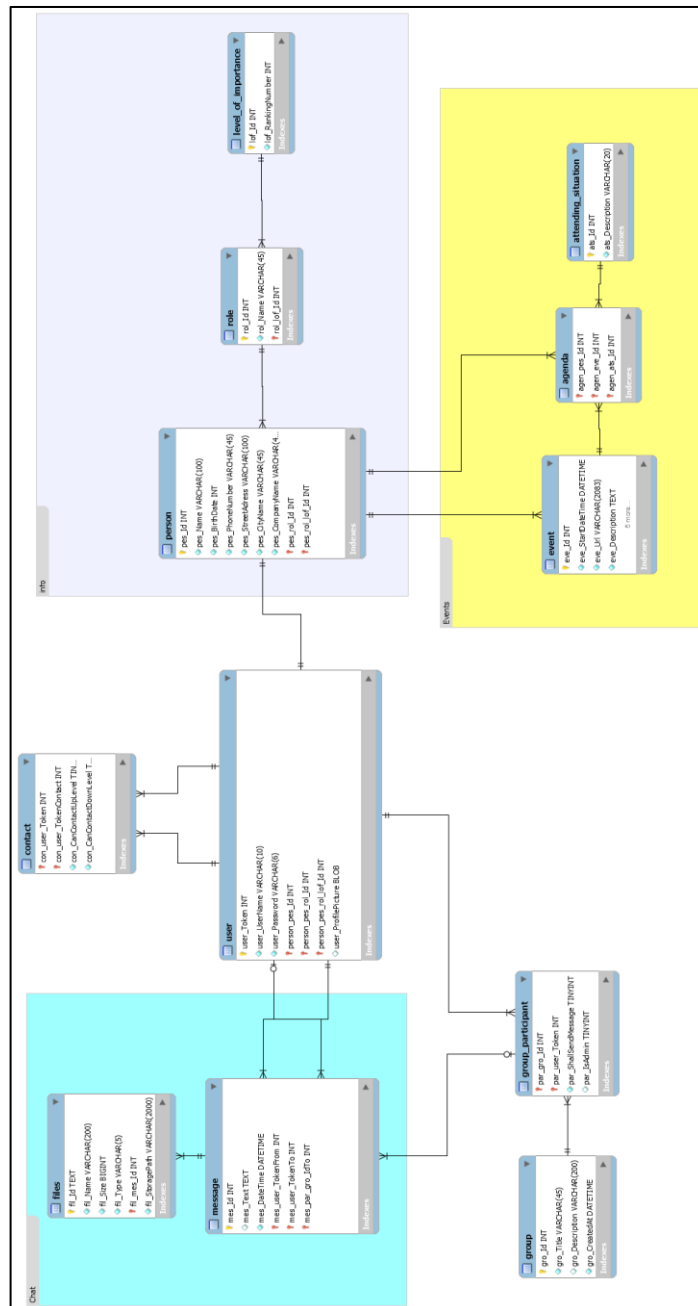


FIGURA 4 - Modelo de banco de dados

3.3.2 MODELO EM ESTRUTURA FIREBASE

Como haverá grande fluxo de dados (envio/recebimento), optou-se por utilizar o *Cloud Firestore*. A Figura 5 apresenta o modelo de banco de dados em estrutura NOSQL aplicada no *Firestore*. O uso do *Firestore* no presente projeto se mostrou relevante, pois, além de ser um servidor gratuito – enquanto projeto acadêmico com quantidade de dados permitida, possibilita uma maior flexibilidade no banco de dados. Os dados no *Cloud Firestore* são semelhantes a uma árvore JSON, mas são organizados em: coleção, documento, subcoleção, hierarquicamente, o que significa que dados complexos e hierárquicos são mais fáceis de organizar em escala. O formato em documento, possibilita uma facilitação de consultas, que, uma vez feitas, poderão trazer todas as informações dos documentos em forma aninhada, não sendo necessárias novas consultas, reduzindo o custo computacional. Ademais, tal ferramenta possui suporte à sincronização de dados em tempo real, o que significa que uma mudança feita em algum dado é persistida imediatamente no sistema e atualizada para todos aqueles conectados à rede, bem como oferece suporte *off-line* de armazenamento local de dados, para dispositivos móveis e WEB.

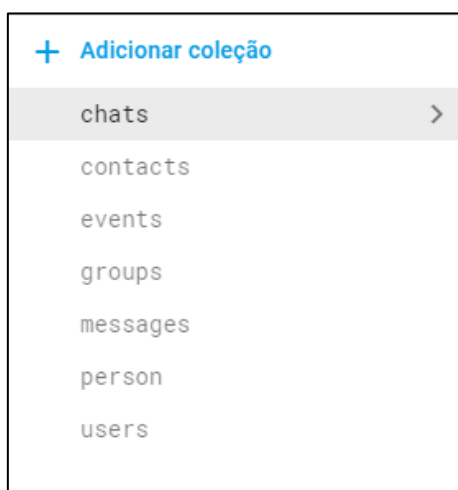


FIGURA 5 - Modelo de banco de dados em estrutura NOSQL aplicada no *Firestore*

4 O APLICATIVO DESENVOLVIDO


Com as especificações do aplicativo quanto à sua regra de negócio e recursos, apresenta-se as interfaces gráficas com o usuário (GUI) e sua finalidade.

O usuário poderá efetuar o *login* no aplicativo para acessar seus recursos, e então seus dados serão carregados e a GUI principal será apresentada, como mostra a Figura 6.



FIGURA 6 – GUI principal (*feed* de eventos)

Na GUI principal, são apresentados os eventos disponíveis na base de dados, de modo a mostrar eventos importantes no topo, eventos do dia (data), logo abaixo, e, a seguir, eventos nos quais o usuário marcou presença na lista.

Nessa GUI, também é apresentado o botão  para criação de um evento. Caso o usuário clique nele, a GUI de criação de evento será apresentada.

A Figura 7 apresenta os detalhes do evento: data, hora, título, local, bem como banner e níveis aos quais se destina devem ser inseridos/selecionados. Basta clicar no botão de conclusão para que esse evento seja publicado. Ao clicar sobre um evento na GUI principal (*feed*), será apresentada a GUI de detalhes, como mostra a Figura 8, a qual contém informações sobre o evento em questão, como: local, horário, data, criador e a lista de quem marcou presença.





FIGURA 7 – GUI de criação de evento



FIGURA 8 – GUI de detalhes do evento

Na GUI de detalhes (Figura 8) o usuário poderá marcar presença no evento, caso este seja destinado a algum dos níveis em que ele está inserido (hierarquia). Exemplo: a alunos, à reitoria, aos orientadores, etc. Caso o usuário seja o criador do evento, o botão a edição ou exclusão poderá ser feita.

Para iniciar uma nova conversa, basta clicar no botão , na GUI de chats (Figura 9). A lista de contatos será mostrada, bastando selecionar algum desses para iniciar uma nova conversa. Para acessar as mensagens, basta clicar no botão de menu inferior . A GUI de conversas (Figura 9) será mostrada. As conversas serão listadas, semelhante ao aplicativo WhatsApp, e, caso o usuário clique sobre uma conversa, a mesma será aberta (Figura 10). Logo, poderão ser enviadas mensagens de texto, imagens, *emojis* e áudio.

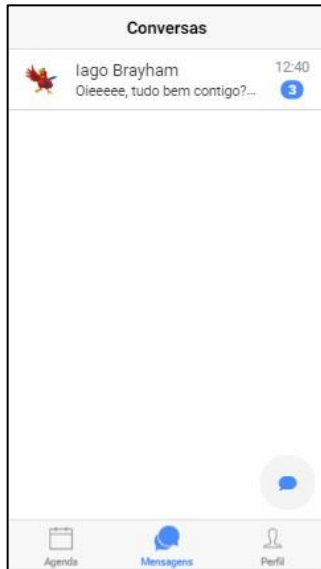


FIGURA 9 – GUI de conversas

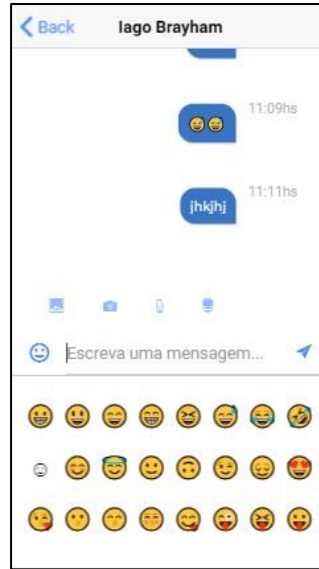



FIGURA 10 – Tela de envio de mensagem

Voltando à GUI de detalhes (Figura 8) de um evento, o participante com a devida permissão (Administrador), poderá iniciar um grupo de conversa para



discutir tal evento, através do botão , disponível na parte superior da GUI. Logo, a GUI para seleção de integrantes desse grupo será apresentada, semelhante a GUI de contatos apresentada anteriormente, conforme Figura 11.

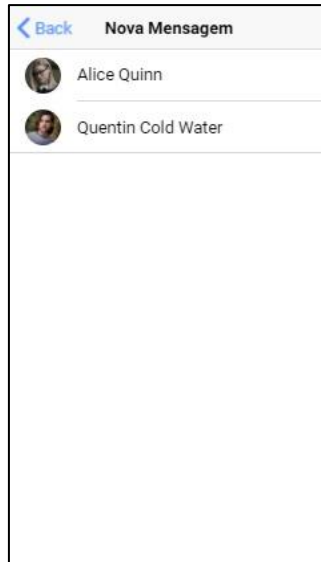


FIGURA 11 – GUI lista de contatos

Com os integrantes adicionados, basta clicar no botão de conclusão e o grupo será criado. Ao se dirigir à GUI de chats (Figura 9), será apresentado, também, o grupo criado, para troca de mensagens no mesmo. Lembrando que a regra de negócio se mantém.

5 CONSIDERAÇÕES FINAIS

A partir do estudo apresentado, e com base nas considerações do autor, observa-se que a utilidade do Ionic 3 depende das decisões de negócio e necessidades do cliente. Em 2018, várias são as ferramentas para desenvolvimento de aplicativos híbridos, como: Flutter, React Native, Ionic e outras; isso abre um leque de opções para empresas e desenvolvedores trabalharem nesse ramo. Assim, algumas considerações devem ser ressaltadas.

O código nativo, ou seja, aquele desenvolvido em IDE's e SDK destinadas a algum sistema operacional (SO) em específico, permite uma melhor performance dos aplicativos, visto que utiliza diretamente os recursos oferecidos pela API do sistema. Porém, a dificuldade de se manter dois códigos distintos para devidos SO's, continua, visto a utilização de recursos próprios (bibliotecas, API, GUI). Assim, caso a necessidade seja alcançar mais usuários, em várias plataformas, *e.g.* iOS, Android, levando em conta o tempo de codificação, que poderá ser feita uma só vez e destinada a qualquer dispositivo que possibilite o uso de *webview*³, é interessante procurar um framework como o Ionic.

A experiência do autor, ressalta que o Ionic oferece vários recursos (componentes) pré-existentes que já foram testados e podem ser usados para alcançar diversas plataformas, contemplando regras de *design*. O mais interessante é que tal recurso possibilita uma codificação que utilize tecnologias web, amplamente difundidas entre desenvolvedores, ou seja, oferece mais adaptabilidade para profissionais. Para citar uma desvantagem, o Ionic faz uso de uma "ponte" para acessar os recursos nativos do dispositivo, como câmera, GPS, fotos, e outros, com a "Cordova", ou seja, o acesso a tais tecnologias é "forçado", o que dificulta o processo de depuração.

Finalmente, conclui-se que, caso o tempo de desenvolvimento e a facilidade de uso sejam prezadas, o framework Ionic é uma ótima alternativa. Já, se a performance for mais relevante, como em desenvolvimentos de jogos e afins, os códigos nativos através das SDK's podem ser a melhor escolha.

³ *Webview*: Componente do SO que permite aplicativos a exibirem conteúdo da Web

6 REFERÊNCIAS BIBLIOGRÁFICAS

AMAZON WEBSERVICE. **O que é NoSQL.** Disponível em: <<https://aws.amazon.com/pt/nosql/>>. Acesso em 09 de março de 2018.

APPLE WEBSITE. **iPhone 5s - Especificações técnicas.** Disponível em: <https://support.apple.com/kb/SP685?locale=pt_BR&viewlocale=pt_BR> Acesso em: 09 de março de 2018.

BELL, Peter. **Introducing GitHub: A Non-Technical Guide.** 1. Ed. Estados Unidos: O'Reilly Media, 2014.

BOOCH, Grady. **UML: Guia do Usuário.** 12. Reimpressão. Rio de Janeiro: Elsevier, 2012.

DEL SOLE, Alessandro. **Visual Studio Code Succintly.** 1. Ed. Estados Unidos: CreateSpace Pub, 2017.

DELL WEBSITE. **Inspiron 15 5000.** Disponível em: <<http://www.dell.com/pt-br/shop/cty/pdp/spd/inspiron-15-5567-laptop>> Acesso em: 09 de março de 2018.

FERREIRA, Davi. **Getting Started.** Disponível em: <<http://tableless.github.io/iniciantes/manual/js/>> Acesso em: 09 de março de 2018

FIREBASE PRODUCTS WEBSITE, Google. **Produtos.** Disponível: <<https://firebase.google.com/products/>> Acesso em 09 de março de 2018.

GIT WEBSITE. **About.** Disponível em: <<https://git-scm.com/about>> Acesso em: 09 de março de 2018.

GITHUB PROJECT, Facebook Inc. **React Native.** Disponível em: <<https://github.com/facebook/react-native>> Acesso em: 09 de março de 2018.

GITLAB WEBSITE. **About us.** Disponível em: <<https://about.gitlab.com/about/>> Acesso em: 09 de março de 2018.

GOOGLE CHROME WEBSITE. **Saiba mais sobre o navegador Google Chrome.** Disponível em: <<https://www.google.com/chrome/browser/features.html>> Acesso em: 09 de março de 2018.

INTERON DIGITAL DESIGN AGENCY WEBSITE. **Hybrid App Development.** Disponível em: <<https://www.interon.co.za/services/app-development/hybrid-app-development/>> Acesso em: 02 de março de 2018.

M. HETHEY, Jonathan. **GitLab Repository Management.** 1. Ed. Estados Unidos: Lightning Source, 2013.

MCLAUGHLIN, Michael. **MySQL Workbench: Data Modeling & Development.** 1 Ed. Nova Iorque: McGraw-Hill/Osborne Media, 2013.

MOTOROLA WEBSITE. **Moto G5s Plus.** Disponível em: <<https://www.motorola.com.br/moto-g5s-plus/p>> Acesso em: 09 de março de 2018.

NEWSROOM, Gartner Inc. **Gartner Says Demand for Enterprise Mobile Apps Will Outstrip Available Development Capacity Five to One.** Disponível em: <<https://www.gartner.com/newsroom/id/3076817>> Acesso em: 02 de março de 2018.

NODE JS WEBSITE. **About Node.js.** Disponível em: <<https://nodejs.org/en/docs/guides/>> Acesso em: 09 de março de 2018.

ONLINE PRESENTATION. **Build Hybrid Apps with AngularJS and Ionic.** Disponível em: <<https://ionicframework.com/present-ionic/slides/#/>> Acesso em: 09 de março de 2018

RAMALHO, José Antônio. **HTML avançado.** 1 Ed. São Paulo: Makron Books, 1997

RAVULAVARU, Arvind. **Learning Ionic.** 2. ed. [S.l.]: Packt, 2017. 378 p.

TEIXEIRA, Fabrício. **Tudo sobre o Adobe Experience Design CC, a nova ferramenta de UX da Adobe.** Disponível em: <<https://brasil.uxdesign.cc/tudo-sobre-o-adobe-experience-design-cc-a-nova-ferramenta-de-ux-da-adobe-a24070ec6ca1>> Acesso em: 09 de março de 2018.

UFCG. **O que é um framework.** Disponível em: <<http://www.dsc.ufcg.edu.br/~jacques/cursos/map/html/frame/oque.htm>> Acesso em: 09 de março de 2018.

VISUAL STUDIO CODE DOCUMENTATION, Microsoft. **Getting Started.** Disponível em: <<https://code.visualstudio.com/docs>> Acesso em: 09 de março de 2018.

W3SCHOOLS WEBSITE. **HomePage.** Disponível em: <<https://www.w3schools.com/html/default.asp>> Acesso em: 09 de março de 2018.

WEBSITE DESIGN CULTURE. **O que é UI Design e UX Design.** Disponível em: <<http://designculture.com.br/o-que-e-ui-design-e-ux-design>> Acesso em: 09 de março de 2018.