

Thyerri Fernandes Mezzari

**UM FRAMEWORK PARA AUXÍLIO NO DESENVOLVIMENTO
DE SISTEMAS BASEADOS NO CONCEITO DE INTERNET DAS
COISAS**

Dissertação submetida ao Programa de Pós-Graduação em Tecnologias da Informação e Comunicação da Universidade Federal de Santa Catarina para a obtenção do Grau de Mestre em Tecnologias da Informação e Comunicação.

Orientador: Prof. Dr. Alexandre Leopoldo Gonçalves.

Coorientador: Prof. Dr. Robson Rodrigues Lemos.

Araranguá
2018

Ficha de identificação da obra elaborada pelo autor através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Mezzari, Thyerri Fernandes
UM FRAMEWORK PARA AUXÍLIO NO DESENVOLVIMENTO DE
SISTEMAS BASEADOS NO CONCEITO DE INTERNET DAS
COISAS / Thyerri Fernandes Mezzari ; orientador,
Alexandre Leopoldo Gonçalves, coorientador, Robson
Rodrigues Lemos, 2018.
118 p.

Dissertação (mestrado) - Universidade Federal de
Santa Catarina, Campus Araranguá, Programa de Pós-
Graduação em Tecnologias da Informação e Comunicação,
Araranguá, 2018.

Inclui referências.

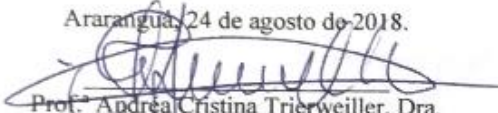
1. Tecnologias da Informação e Comunicação. 2.
Internet das Coisas. 3. Framework. 4. Android. I.
Gonçalves, Alexandre Leopoldo. II. Lemos, Robson
Rodrigues. III. Universidade Federal de Santa
Catarina. Programa de Pós-Graduação em Tecnologias da
Informação e Comunicação. IV. Título.

Thyerri Fernandes Mezzari

UM FRAMEWORK PARA AUXÍLIO NO DESENVOLVIMENTO DE SISTEMAS BASEADOS NO CONCEITO DE INTERNET DAS COISAS

Esta Dissertação foi julgada adequada para obtenção do Título de Mestre, e aprovada em sua forma final pelo Programa de Pós-Graduação em Tecnologias da Informação e Comunicação da Universidade Federal de Santa Catarina

Araranguá, 24 de agosto de 2018.



Prof. Andrea Cristina Trierweiler, Dra.
Coordenadora do Curso

Banca Examinadora:



Prof. Alexandre Leopoldo Gonçalves, Dr.
Orientador


Universidade Federal de Santa Catarina (UFSC)

Robson Rodrigues Lemos
Prof. Robson Rodrigues Lemos, Dr.
Coorientador


Universidade Federal de Santa Catarina (UFSC)



Prof. Juarez Bento da Silva, Dr.
Universidade Federal de Santa Catarina (UFSC)



Prof. Flavio Ceci, Dr.
Universidade do Sul de Santa Catarina (UNISUL)



Prof. Cristian Cechinel, Dr.
Universidade Federal de Santa Catarina (UFSC)

Este trabalho é dedicado aos meus amigos e à minha família que me apoiaram durante todos os momentos deste projeto.

AGRADECIMENTOS

Agradeço a minha família e, em especial, aos meus pais pelo incentivo à carreira acadêmica, pelo apoio, orientação e compreensão.

Aos colegas da Tecnnic Eletrônica®, que estiveram presentes nesta jornada, em especial ao Ricardo pelo conhecimento repassado e pelo abono das faltas em prol das disciplinas do mestrado.

Aos colegas da Innova Connect®, que estiveram presentes na etapa final desta caminhada, incentivando e motivando o projeto.

À minha companheira, Gabriela, por todo seu empenho em me manter focado e motivado, apoiando e incentivando a conclusão do projeto. Agradeço também pela sua compreensão em todas as vezes que precisamos abrir mão do tempo livre, do lazer e da diversão em prol deste trabalho.

Aos meus amigos, colegas e todos aqueles que de uma forma ou de outra, estiveram ao meu lado, compreendendo, orientando e apoiando durante todo o curso.

A Universidade Federal de Santa Catarina, por ofertar o curso de Tecnologias da Informação e Comunicação na região sul de Santa Catarina.

Em especial ao orientador Prof. Dr. Alexandre Leopoldo Gonçalves, pelo estímulo, dedicação e, principalmente, por ter acolhido a proposta, lapidando a ideia e transformando-a em um projeto.

“A única maneira de fazer um excelente trabalho é amar o que faz!”

(Steve Jobs)

RESUMO

A tecnologia está cada vez mais presente na vida das pessoas permitindo que tarefas do dia a dia sejam realizadas com maior facilidade, rapidez e eficiência. Os recentes avanços das tecnologias de redes e a consolidação da Internet das Coisas (IoT), vêm permitindo uma ampla gama de possibilidades quanto ao desenvolvimento de aplicações. A elaboração de sistemas desta natureza demanda esforços por parte dos desenvolvedores. Esses devem obter conhecimento a respeito de diversas ferramentas comuns em aplicações no contexto da IoT. Diante do exposto, este trabalho propõe um *framework* para o auxílio no desenvolvimento de sistemas baseados no conceito da IoT. O *framework* proposto permite agrupar, em uma única biblioteca, diversas funcionalidades comuns em aplicações na área de IoT. Entre as funcionalidades destacam-se a geolocalização, a manipulação de banco de dados, a análise da utilização da aplicação (*analytics*), e as notificações (*push notifications*). Para a demonstração de viabilidade, um protótipo foi desenvolvido na plataforma do sistema operacional móvel Android®, seguido de uma aplicação de teste, também na mesma plataforma. O protótipo desenvolvido foi disponibilizado a um grupo de especialistas seguido de um questionário com o objetivo de avaliar cada uma das funcionalidades do *framework*. A análise das respostas coletadas no questionário apontou que o *framework* proposto está em linha com o esperado e relatado na literatura. Demonstra também que o objetivo principal do trabalho foi atingido, auxiliando no desenvolvimento de sistemas voltados à Internet das Coisas.

Palavras-chave: Internet das Coisas, *Framework*, Android®.

ABSTRACT

Technology is increasingly present in people's lives, enabling day-to-day tasks to be performed more easily, quickly and efficiently. Recent advances in network technologies and the consolidation of the Internet of Things (IoT) are allowing a wide range of possibilities for application development. The development of this kind of systems demands efforts from developers. They should acquire knowledge about various common tools in applications in the context of IoT. Thus, this work proposes a framework aiming to help in the development of systems based on the IoT concept. The proposed framework allows to group, in a single library, several common functionalities in applications in the area of IoT. Features include geolocation, database manipulation, analytics, and notifications (push notifications). For the demonstration of feasibility, a prototype was developed on the Android® mobile operating system platform, followed by a test application also on the same platform. The developed prototype was made available to a group of experts followed by a questionnaire with the objective of evaluating each of the functionalities of the framework. The analysis of the answers collected in the questionnaire pointed out that the proposed framework is in line with the expected and reported in the literature. It also demonstrates that the main objective of the work was achieved assisting in the development of systems toward the Internet of Things.

Keywords: Internet of Things, Framework, Android®.

LISTA DE FIGURAS

Figura 1 – Elementos da Internet das Coisas.....	29
Figura 2 – Etapas do trabalho.....	33
Figura 3 – Produção científica para IoT na base Scopus, até o primeiro semestre de 2018.....	37
Figura 4 – Camadas da arquitetura IoT.....	38
Figura 5 - Plano de canais para o Bluetooth Low Energy.....	39
Figura 6 - Código de barras e RFID.....	42
Figura 7 – Sistema inteligente para desbloqueio de porta com fechadura eletrônica – diagrama de blocos.....	47
Figura 8 - Smart Cities e IoT: Tecnologias disponíveis versus desafios....	48
Figura 9 – Usuário interagindo com o sistema de realidade virtual.....	53
Figura 10 – Visão do usuário no sistema de realidade virtual.....	53
Figura 11 – Estrutura de um sistema IoT.....	57
Figura 12 – Framework proposto para o desenvolvimento de aplicações IoT.....	61
Figura 13 – Exemplo de uma aplicação IoT com dados de geolocalização	62
Figura 14 – Exemplo de painel do Firebase Analytics®.....	70
Figura 15 – Diagrama de interface para a inicialização do framework.....	73
Figura 16 – Diagrama de sequência de inicialização do framework.....	74
Figura 17 – Diagrama de interface para o serviço de autenticação.....	75
Figura 18 – Diagrama de sequência para o fluxo de login.....	75
Figura 19 – Diagrama da interface para manipulação de banco de dados...	76
Figura 20 – Diagrama de sequência sobre o fluxo de leitura e gravação....	77
Figura 21 – Diagrama de interface para o serviço de geolocalização.....	78
Figura 22 – Diagrama de sequência para o fluxo de coleta de dados com base na localização.....	78
Figura 23 – Diagrama de interface para o serviço de push notification.....	79
Figura 24 – Console do Firebase Cloud Message®.....	79
Figura 25 – Diagrama de sequência para o fluxo push notification.....	80
Figura 26 – Diagrama de interface sobre o serviço de analytics.....	80
Figura 27 – Diagrama de sequência do fluxo analytics.....	81
Figura 28 – Console do Firebase Analytics® - acesso público.....	81
Figura 29 – Console do Firebase Analytics® - aceitação.....	82
Figura 30 - Console do Firebase Analytics® - usuários ativos.....	82
Figura 31 – Etapas de desenvolvimento do framework na plataforma Android®.....	84
Figura 32 – Criando o projeto do framework IoT no Android Studio®.....	85
Figura 33 – Permissões do Usuário, no arquivo “Android Manifest”.....	86

Figura 34 – Informações pertinentes ao Facebook Account Kit® no arquivo “Android Manifest”	86
Figura 35 – Interface principal do framework IoT	87
Figura 36 – Classe principal do framework IoT	88
Figura 37 – Etapas para desenvolvimento do aplicativo de teste	89
Figura 38 – Interface gráfica do aplicativo de teste	90
Figura 39 – Método onClick() do botão “Add Object”	90
Figura 40 – Área de atuação dos entrevistados	92
Figura 41 – Tempo de atuação em IoT	92
Figura 42 – Funcionalidades presentes nos projetos de IoT na qual os entrevistados participaram	93
Figura 43 – Avaliação da proposta para a funcionalidade de fluxo de login	94
Figura 44 – Avaliação da proposta para a funcionalidade de banco de dados	94
Figura 45 – Avaliação da proposta para a funcionalidade de geolocalização	95
Figura 46 – Avaliação da proposta para a funcionalidade notificações (push notifications)	96
Figura 47 – Avaliação da proposta para a funcionalidade de analytics	96

LISTA DE QUADROS

Quadro 1 – Padrões 802.11	41
---------------------------------	----

LISTA DE ABREVIATURAS E SIGLAS

API – *Application Programming Interface*
AR – *Augmented Reality*
ARPANET – *Advanced Research Projects Agency Network*
BLE – *Bluetooth Low Energy*
CCS – *Context Sensitive System*
CSMA – *Carrier Sense Multiple Access*
DRY – *Don't Repeat Yourself*
DVR – *Digital Video Recorder*
GPS – *Global Positioning System*
GUI – *Graphical User Interface*
HTML - *HyperText Markup Language*
HTTP – *Hypertext Transfer Protocol*
ID – *Identity*
IoT – *Internet of Things*
ITS – *Intelligent Transportation System*
LAN – *Local Area Network*
OSD – *Open Source Definition*
OSI – *Open Source Initiative*
OTA – *Over The Air*
PPGTIC – *Programa de Pós-Graduação em Tecnologias da Informação e Comunicação*
PCB – *Printed Circuit Board*
RA – *Realidade Aumentada*
RDBMS – *Relational Database Systems*
RFID – *Radio Frequency IDentification*
RS – *Recommender Systems*
SDKs – *Software Development Kit*
SMS - *Short Message Service*
SO – *Sistema Operacional*
SoC – *System on a Chip*
SQL – *Structured Query Language*
TCP – *Transmission Control Protocol*
TIC – *Tecnologias da Informação e Comunicação*
UML – *Unified Modeling Language*
UX – *User experience*
VAE – *Virtual Assembly environment*
VR – *Virtual Reality*
WSN – *Wireless Sensor Network*

SUMÁRIO

1 INTRODUÇÃO.....	25
1.1 PROBLEMATICA.....	27
1.2 OBJETIVOS.....	28
1.2.1 Objetivo Geral.....	28
1.2.2 Objetivos Específicos.....	29
1.3 JUSTIFICATIVA E MOTIVAÇÃO.....	29
1.4 ESCOPO DO TRABALHO.....	30
1.5 ADERÊNCIA AO PROGRAMA DE PÓS-GRADUAÇÃO EM TECNOLOGIAS DA INFORMAÇÃO E COMUNICAÇÃO.....	31
1.6 PROCEDIMENTOS METODOLÓGICOS.....	32
1.7 ESTRUTURA DO TRABALHO.....	33
2 INTERNET DAS COISAS.....	35
2.1 INTRODUÇÃO.....	35
2.2 HISTÓRICO.....	36
2.3 TECNOLOGIAS DE COMUNICAÇÃO PARA A INTERNET DAS COISAS.....	38
2.3.1 <i>Bluetooth</i> ®.....	39
2.3.2 <i>Wi-Fi</i>	40
2.3.3 RFID e Código de Barras.....	41
2.4 SEGURANÇA E PRIVACIDADE.....	42
2.5 CENÁRIOS DA INTERNET DAS COISAS.....	45
2.5.1 <i>Smart Home</i>	45
2.5.2 <i>Smart City</i>	47
2.5.4 <i>Smart Healthcare</i>	49
2.6 SISTEMAS OPERACIONAIS MÓVEIS.....	51
2.7 SISTEMAS EMBARCADOS.....	52

2.8 <i>FRAMEWORKS</i>	54
2.8.1 <i>Frameworks</i> IoT	55
2.8.2 Trabalhos e <i>Frameworks</i> Relacionados	56
3 <i>FRAMEWORK</i> PROPOSTO	61
3.1 INTRODUÇÃO.....	61
3.2 AUTENTICAÇÃO (<i>LOGIN</i>)	63
3.3 BANCO DE DADOS	65
3.4 GEOLOCALIZAÇÃO	66
3.5 NOTIFICAÇÕES (<i>PUSH NOTIFICATIONS</i>)	68
3.6 COLETA E ANÁLISE DE DADOS (<i>ANALYTICS</i>)	69
3.7 REQUISITOS.....	70
3.7.1 Requisitos Funcionais	70
3.7.2 Requisitos não funcionais.....	71
3.8 ESPECIFICAÇÃO DO <i>FRAMEWORK</i>	71
3.8.1 Instalação e inicialização do framework	72
3.8.2 Autenticação (<i>Login</i>).....	74
3.8.3 Banco de Dados	76
3.8.4 Geolocalização.....	77
3.8.5 Notificações (<i>Push Notifications</i>).....	79
3.8.6 <i>Analytics</i>	80
4 IMPLEMENTAÇÃO E AVALIAÇÃO.....	83
4.1 DESENVOLVIMENTO DE <i>SOFTWARE</i>	83
4.1.1 Biblioteca Android®.....	84
4.1.2 Arquivo de manifesto Android®	85
4.1.3 Interface e Classe principal do <i>framework</i>	87
4.1.4 Disponibilização da biblioteca Android®	88
4.1.5 Aplicativo de teste.....	89

4.2 AVALIAÇÃO E RESULTADOS.....	91
5 CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS	97
5.1 CONSIDERAÇÕES FINAIS	97
5.2 TRABALHOS FUTUROS	98
REFERÊNCIAS.....	101
ANEXO A – LICENÇA	109
APÊNDICE A – RECURSOS DECLARADOS NO ANDROID® GRADLE E ANDROID® MANIFEST	111
APÊNDICE B – QUESTIONÁRIO DE AVALIAÇÃO DO FRAMEWORK	113

1 INTRODUÇÃO

A Internet das Coisas, assim como a Internet, tem a possibilidade de promover grandes avanços no contexto tecnológico atual. As pessoas estão mais habituadas com a tecnologia nas suas vidas, abrindo espaço para que novos produtos sejam criados (RUIZ-ROSETO et al., 2017).

Espera-se uma mudança de hábito dos usuários a partir do avanço da Internet das Coisas, que vem sendo conceituada e melhorada desde que foi citada pela primeira vez por Kevin Ashton em 1999 (ASHTON, 2009). Do inglês *Internet of Things* ou simplesmente IoT, o termo sugere que os mais variados tipos de objetos poderão estar conectados a rede mundial de computadores, a Internet.

Em um cenário idealizado com base no conceito IoT, um automóvel poderia coletar informações do trânsito e pensar no caminho ideal, enquanto uma pulseira eletrônica mede sinais vitais, temperatura e nível de glicose do usuário, informando-o para alimentar-se com base em uma dieta saudável. Ao perceber esta necessidade, a geladeira do usuário poderia informar a falta de alimentos saudáveis e o automóvel calcular um caminho alternativo passando pelo supermercado.

No cenário acima é possível verificar que os sistemas computacionais têm acesso a uma série de informações pessoais dos usuários, sendo esta uma das preocupações levantadas por autores em relação ao avanço da IoT. Ng e Wakenshaw (2017), por exemplo, lembram que as “coisas” precisarão trocar informações entre si, e estas trocas podem ocorrer em condições incertas. Para superar a falta de certeza destes sistemas eles acreditam no poder dos sistemas de gerenciamento de confiança.

Analisando o contexto da IoT é possível perceber que sua estrutura envolve, além dos sistemas computacionais, a interconexão de uma vasta gama de sensores. Tal estrutura, que é a base do conceito IoT, precisa ser robusta e responsável, garantindo a manipulação dos dados com segurança.

Também é importante lembrar que, em geral, as pessoas não costumam adquirir todas as suas “coisas” de uma mesma marca ou modelo, isto impacta em maiores dificuldades na comunicação entre estes sistemas. Para que sistemas das mais variadas marcas, modelos e plataformas interajam entre si, são necessários mecanismos que permitam tal comunicação. Estes mecanismos geralmente são protocolos e *frameworks* que os desenvolvedores adotam, permitindo que os dados sejam organizados e padronizados de forma que os sistemas possam

interagir entre si, consumindo estes dados de maneira adequada (MANIAPPAN, 2015).

Já a leitura e processamento dos sensores pode ser algo ainda mais complicado para os desenvolvedores, visto que demandam conhecimento das plataformas de *hardware*, das bibliotecas para leitura via *software* e do funcionamento dos sensores e suas grandezas para que os dados coletados possam ser transformados em informação (AMMAR; RUSSELLO; CRISPO, 2018).

Tais fatores demandam demasiados esforços dos desenvolvedores e tornam-se agravantes no desenvolvimento de sistemas IoT. Segundo Ng e Wakenshaw (2017), a complexidade das tecnologias e a fragmentação na pesquisa da IoT são dois obstáculos que impedem os desenvolvedores de obter o *know-how* adequado para projetar e implementar sistemas IoT. Programadores, analistas e engenheiros de *software* precisam aprofundar seus conhecimentos em diversas bibliotecas e APIs (*Application Programming Interface*), além de aprenderem sobre o contexto da aplicação e seus requisitos.

Essas dificuldades no desenvolvimento de sistemas IoT podem diminuir a robustez, qualidade e segurança dos mesmos. A maioria dos sistemas computacionais não estão completamente adequados ao conceito IoT, visto o demasiado esforço por parte dos usuários em fornecer informações, configurar parâmetros e preferências de uso (VIEIRA; TEDESCO; SALGADO, 2009), bem como, a imensa dificuldade em filtrar os serviços relevantes para cada perfil de usuário (MASHAL; CHUNG; ALSARYRAH, 2015).

O ideal seria que os sistemas percebessem as necessidades do usuário, executando suas funcionalidades sem que o usuário tenha de interagir explicitamente com o sistema, simplificando assim a presença da tecnologia na vida dos seres humanos.

As pessoas já estão habituadas com as tecnologias em seu cotidiano e, segundo Bicen e Arnavut (2015), quando essas pessoas são questionadas, destacam os computadores, a Internet e o telefone celular. A proposta da Internet das Coisas é exatamente tornar a tecnologia algo corriqueiro na vida das pessoas e, por este motivo, cabe aos sistemas computacionais serem o menos invasivo possível, flexíveis e fáceis de utilizar, e não dependentes da interação explícita do usuário para realização de suas funcionalidades.

1.1 PROBLEMATICA

Atualmente, as maiores dificuldades do universo IoT estão ligadas a interoperabilidade dos dispositivos. Muito trabalho tem sido despendido a fim de garantir a troca de informações entre os sistemas com segurança, rapidez e eficácia. Para que aplicações de IoT inteligentes e orientadas às pessoas surjam, a interoperabilidade nos níveis de dados e conhecimento torna-se necessária (AKPAKWU et al., 2017; MAZAYEV; MARTINS; CORREIA, 2017).

A interoperabilidade é um problema importante para as aplicações porque os dispositivos são construídos por diferentes fabricantes que, por falta de padronização, torna a interação entre dispositivos heterogêneos um grande desafio. Esta dificuldade na obtenção de comunicação entre dispositivos heterogêneos é causada, principalmente pela ausência de um *framework* que sirva de *middleware* exclusivo para as aplicações de IoT (AKPAKWU et al., 2017).

A complexidade das tecnologias de IoT e a fragmentação na pesquisa, levando em conta as dezenas de áreas onde a IoT pode ser aplicada, também são obstáculos que impedem os cientistas de obterem o conhecimento adequado para projetar e implementar sistemas de IoT (FERNANDEZ-GAGO; MOYANO; LOPEZ, 2017). Tais fatos podem conduzir a atrasos e falhas no processo de desenvolvimento de aplicações.

Outra questão importante é que a Internet das Coisas engloba uma grande variedade de dispositivos, equipamentos e plataformas, sendo necessário abordar questões de segurança em diferentes níveis. Khan e Salah (2018) lidam com a complexidade da segurança em sistemas IoT, aplicando soluções baseadas em *Blockchain*, uma tecnologia capaz de autenticar transações de ponta a ponta sem a necessidade de uma unidade central intermediadora.

Esta complexidade, somada ao grande crescimento das aplicações de IoT, cria muitos desafios científicos e de engenharia que exigem esforços de pesquisa tanto da academia como da indústria (KIM; RAMOS; MOHAMMED, 2017).

Mora, Gianni e Divitini (2016) lembram que desenvolvedores não especializados em Internet das Coisas, como designers, estudantes e até mesmo fabricantes de produtos IoT devem fazer parte de uma estratégia colaborativa, com objetivo de prover à ampla população todo o potencial que as ferramentas IoT podem oferecer.

Projetos estruturados de forma colaborativa são capazes de suprir a deficiência das ferramentas de desenvolvimento e a falta de

conhecimento em desenvolvimento de *hardware* e sistemas embarcados. Estas capacitações são geralmente vinculadas a tecnologias específicas e exigem uma curva de aprendizagem maior, dificultando o avanço do projeto.

Johnsson e Magnusson (2017) vão além e, de acordo com os conceitos da Internet das Coisas, defendem que o usuário deva ser proativo, participando na criação, refletindo suas necessidades e desejos. Tal contexto gerou o objetivo de seu trabalho, que visa aumentar o número de desenvolvedores e acelerar o desenvolvimento de GUIs (*Graphic User Interface*) para aplicações IoT. Para alcançar este objetivo, a ferramenta proposta pelos autores permite usuários comuns, participarem do processo de criação das interfaces gráficas.

É notável que o cenário de IoT carece de ferramentas que auxiliem os desenvolvedores no processo de criação destes sistemas, ou seja, ferramentas que acelerem o desenvolvimento da IoT. Estas ferramentas poderiam auxiliar no processo de coleta de dados dos sensores, no processo de acesso (*login*) por parte dos usuários, no armazenamento de dados em nuvem, entre outros.

Ammar, Russello e Crispo (2018) lembram que o desenvolvimento de aplicativos para a IoT é uma tarefa desafiadora devido a alta complexidade da computação distribuída, a falta de diretrizes ou *frameworks* que simplifiquem a implementação, linguagens de programação múltiplas e os vários protocolos de comunicação. Essa complexidade levou a uma rápida evolução em termos de introdução de *frameworks* de programação IoT que lidem com os desafios supracitados (AMMAR; RUSSELLO; CRISPO, 2018).

Com base na problemática exposta nesta seção apresenta-se a seguinte pergunta de pesquisa: **Como facilitar o desenvolvimento de sistemas voltados ao contexto da Internet das Coisas?**

1.2 OBJETIVOS

A seguir são descritos os objetivos geral e específicos.

1.2.1 Objetivo Geral

Desenvolver um *framework* que auxilie e facilite a implementação de sistemas voltados para a Internet das Coisas.

1.2.2 Objetivos Específicos

De modo a atingir o objetivo geral, os seguintes objetivos específicos são requeridos:

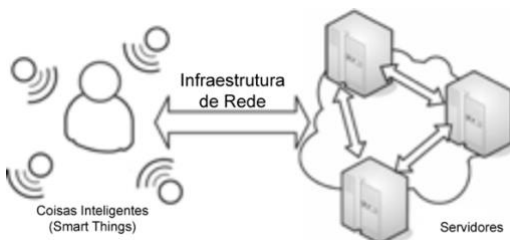
- Identificar os requisitos mais comuns em sistemas para a Internet das Coisas;
- Identificar ferramentas e bibliotecas que atendam de forma adequada os requisitos mais comuns necessários ao desenvolvimento de sistemas para a Internet das Coisas;
- Demonstrar a viabilidade do *framework* proposto através de um protótipo e de um questionário aplicado a um grupo de especialistas;
- Promover uma discussão dos resultados obtidos pela aplicação do questionário.

1.3 JUSTIFICATIVA E MOTIVAÇÃO

A justificativa deste trabalho refere-se à aceleração no processo de desenvolvimento de sistemas para IoT, principalmente em relação a interoperabilidade entre os sistemas, sensores, dentre outros requisitos presentes na maioria das aplicações IoT. Para vencer este desafio, são necessários padrões de desenvolvimento que forneçam suporte a estes novos sistemas.

As principais empresas do mundo que estão produzindo dispositivos inteligentes vêm trabalhando para alcançar uma interoperabilidade total, que garanta uma integração fácil com a Internet existente (RISTESKA STOJKOSKA; TRIVODALIEV, 2017). Contudo, faltam sistemas e estruturas computacionais capazes de explorar eficientemente os dados e aplicá-los na busca por conhecimento (ALI et al., 2017).

Figura 1 – Elementos da Internet das Coisas



Fonte: Adaptado de Lopez et al. (2017)

Lopez et al. (2017) levantam os aspectos que afetam a privacidade do usuário em sistemas IoT, apresentando-os em uma estrutura formada por três componentes principais (Figura 1): objetos inteligentes, servidores e infraestrutura de comunicação. Para vencer os desafios neste cenário e permitir a conexão dos objetos à Internet, torna-se fundamental a utilização de um ferramental que padronize a comunicação, garantindo robustez, eficiência e segurança, e reduzindo os riscos contra a privacidade do usuário.

Muitas vezes os padrões de tecnologia são estabelecidos de forma colaborativa, entre as empresas privadas e as instituições acadêmicas. Isso facilita o desenvolvimento e acelera a adoção do padrão, reduzindo os riscos para ambas as partes. Os padrões de tecnologia são determinantes no desenvolvimento e adoção de um novo paradigma como a Internet das Coisas. Até que um padrão dominante seja estabelecido, a tecnologia ainda será vista como emergente e terá vários vetores alternativos em seu desenvolvimento e difusão (KROTOV, 2017).

Um *framework* que auxilie o desenvolvimento de aplicações voltadas à Internet das Coisas é fundamental para acelerar o andamento de projetos, garantindo segurança e eficácia. O uso de frameworks durante o desenvolvimento, traz benefícios não só ao desenvolvedor, mas também para o usuário final, tendo em vista que melhores ferramentas podem construir melhores produtos (FERNANDEZ-GAGO; MOYANO; LOPEZ, 2017).

Espera-se com o *framework* proposto neste trabalho, que os projetos de IoT possam ser concluídos com mais rapidez, eficiência e eficácia, garantindo sistemas mais seguros e robustos. Procura-se também auxiliar a comunidade de desenvolvedores para que estes possam direcionar seu empenho em outros requisitos de seus projetos, criando sistemas cada vez melhores, mais competitivos e rentáveis.

1.4 ESCOPO DO TRABALHO

Este trabalho tem como principal intuito construir e avaliar um *framework* que auxilie no desenvolvimento de sistemas voltados para a Internet das Coisas. Desta forma, visa contribuir para o avanço da tecnologia de forma a acelerar o processo de desenvolvimento destes sistemas, reduzindo a curva de aprendizagem, evitando falhas e aumentando a confiabilidade.

Para a compreensão e delimitação do tema foram realizados estudos com foco em Internet das Coisas, suas áreas de atuação,

tecnologias e termos relevantes, além da busca por *frameworks* que atuam na área e que de alguma maneira pudessem auxiliar no desenvolvimento deste trabalho.

O *framework* desenvolvido auxilia os desenvolvedores a embutirem em seus projetos, ferramentas comuns em diversos sistemas como *login* e armazenamento de dados em nuvem, além de ferramentas de geolocalização que permitem recuperar dados relevantes com base em coordenadas cartesianas.

Não se encontra sob o escopo deste trabalho o estudo das técnicas para desenvolvimento de *frameworks*, uma vez que o foco principal se refere à contribuição no avanço para o desenvolvimento de aplicações de Internet das Coisas.

Além disso, o trabalho não visa a conversão do desenvolvimento de *software* em um produto final, mas sim como instrumento de estudo e avaliação de resultados obtidos.

O público alvo deste projeto é a comunidade de desenvolvedores de aplicações de pequeno porte, voltadas à Internet das Coisas. Este grupo de pessoas poderá se beneficiar com este trabalho, acelerando o desenvolvimento de seus projetos e aumentando a qualidade dos seus produtos.

1.5 ADERÊNCIA AO PROGRAMA DE PÓS-GRADUAÇÃO EM TECNOLOGIAS DA INFORMAÇÃO E COMUNICAÇÃO

Tecnologia Computacional, Tecnologia Educacional e Gestão e Inovação são as três linhas de pesquisa em que é dividido o Programa de Pós-Graduação em Tecnologias da Informação e Comunicação (PPGTIC).

Esta pesquisa visa a construção de um *framework* que procura auxiliar e acelerar o processo de desenvolvimento de aplicações voltadas para a Internet das Coisas. Atua assim, como ferramenta de inovação, conforme evidenciado no decorrer do texto. Por este motivo, este trabalho está alocado na linha de pesquisa em Tecnologia Computacional, cujo objetivo fomenta o desenvolvimento de ferramentas que auxiliem a resolução de problemas interdisciplinares (PPGTIC, 2018).

No âmbito da interdisciplinaridade o projeto contempla a linha educacional, fornecendo ferramentas e recursos para a implementação de protótipos IoT, desenvolvidos em laboratórios de escolas e universidades. Fornece conteúdo para o ensino-aprendizagem na área de IoT, com base em recursos presentes na grande maioria das aplicações

web e *mobile* atualmente disponíveis, fomentando o estudo e desenvolvimento de aplicações voltadas para a Internet das Coisas.

1.6 PROCEDIMENTOS METODOLÓGICOS

A metodologia expõe os procedimentos científicos levados a alcançar as respostas para os problemas.

Este trabalho caracteriza-se como uma pesquisa tecnológica e aplicada. Uma vez que visa resolver problemas técnicos a partir de estudo em bases científicas, a pesquisa tecnológica precisa de uma meta bem definida para comprovar os benefícios dos resultados (VARGAS, 1985). A pesquisa aplicada, segundo Silva e Menezes (2001, p.20), visa gerar conhecimentos para resolução de problemas específicos, com foco na aplicação prática, complementando a pesquisa tecnológica.

Deste modo, para atingir os objetivos propostos neste trabalho, os seguintes passos foram executados:

- Realização de uma pesquisa bibliográfica e estudo das tecnologias envolvidas no trabalho;
- Levantamento de requisitos com base na revisão bibliográfica, a fim de propor um *framework* para auxílio no desenvolvimento de sistemas computacionais baseados nas tecnologias estudadas;
- Desenvolvimento e disponibilização de um protótipo do *framework* proposto, visando uma determinada plataforma, dentre as opções disponíveis;
- Formulação e aplicação de um questionário junto a um grupo de especialistas na área de Internet das Coisas
- Avaliação do *framework* proposto por meio da análise e discussão dos resultados obtidos na aplicação do questionário;
- Apresentação das considerações finais e possíveis trabalhos futuros.

Buscando alcançar os objetivos propostos, foram executadas as etapas representadas na Figura 2. Uma vez definido o tema e o escopo do trabalho, procedeu-se a pesquisa bibliográfica com base em *Frameworks* e Internet das Coisas, no intuito de compreender os conceitos inerentes às tecnologias escolhidas. Conjuntamente, foi

realizado o desenvolvimento do *Framework* proposto, a fase mais importante e delicada do projeto.

Os testes, aplicação do questionário e avaliações do projeto desenvolvido procederam na etapa seguinte, concluindo o trabalho com a organização e análise dos dados obtidos.

Figura 2 – Etapas do trabalho



Fonte: Autor

1.7 ESTRUTURA DO TRABALHO

O presente trabalho é composto por 5 capítulos, estruturados da seguinte maneira:

- Capítulo 1: Capítulo introdutório, composto pela problemática, objetivos, justificativa, escopo do trabalho, aderência ao programa PPGTIC e a metodologia adotada.
- Capítulo 2: Neste capítulo apresentam-se os conceitos inerentes à Internet das Coisas, suas tecnologias, áreas e termos relevantes. Também é apresentada a pesquisa sobre *frameworks* voltados para Internet das Coisas.
- Capítulo 3: Discorre acerca dos detalhes do *framework* proposto.
- Capítulo 4: Neste capítulo são apresentados os resultados e discussões associadas ao *framework* proposto.

- Capítulo 5: Apresenta as conclusões e recomendações para futuras pesquisas.

2 INTERNET DAS COISAS

O conceito de Internet das Coisas objetiva, de modo geral, é a interconexão de uma série de objetos à Internet, ou uma aplicação que, de alguma forma, compartilhe determinado objeto na rede (HUANG; LI, 2010). Segundo Mazayev, Martins e Correia (2017), a IoT trará para a Internet todo o tipo de dispositivos e sensores, capazes de registrar observações físicas, que estarão acessíveis a qualquer momento, de qualquer lugar.

Neste capítulo é apresentado um histórico da IoT e alguns conceitos que apresentam sua evolução até o presente momento.

2.1 INTRODUÇÃO

A Internet das Coisas já é utilizada por 48% das empresas no mundo, segundo um levantamento feito pela Consultoria de Risco Marsh, realizado junto a 700 executivos de 60 empresas, incluindo o Brasil (MARSH, 2015).

As “coisas” ou objetos podem ser sensores, aparelhos de TV, eletrodomésticos, sistemas de segurança, sistemas de automação residencial, etc. (HUANG; LI, 2010), ou seja, tudo que está em volta das pessoas em seu dia a dia. A Internet das Coisas tem por objetivo interligar tudo na rede, permitindo assim, que computadores e “coisas”, ou ainda, aplicações da Internet e “coisas” se comuniquem compartilhando dados e comandos de forma a não ser necessário a intervenção humana.

As redes de sensores sem fio permitem aos objetos fornecerem informações sobre o ambiente, contexto e localização. Essas tecnologias inteligentes são capazes de permitir que coisas cotidianas interajam entre si realizando tarefas cada vez mais complexas. Por exemplo, na área da tecnologia vestível, os tecidos permitem incorporar componentes digitais e eletrônicos neles, além de relógios e pulseiras vestíveis, já disponíveis no mercado (NG; WAKENSHAW, 2017).

Do ponto de vista das aplicações, os sensores, ou as redes de sensores, são a porta de entrada para coleta de dados. Portanto, os sensores alimentam os sistemas. Em outras palavras, os sensores se tornaram a interface com o usuário, não sendo necessário aos usuários acionarem um botão ou enviar um comando para que a ação aconteça.

A ideia proposta pela Internet das Coisas pode ser resumida em uma interação natural entre seres humanos, computadores e dispositivos

em geral. Todos os equipamentos elétricos e eletrônicos de uso diário podem ser monitorados e controlados via rede.

Uma série de sensores de temperatura, por exemplo, podem ser capazes de transformar os dados físicos em dados digitais, enviando-os a um centro de controle que, por sua vez, pode monitorar mudanças climáticas em determinado ambiente e tomar ações como informar o supervisor técnico via *e-mail*, SMS (*Short Message Service*) ou, até mesmo, acionar um sistema de refrigeração que também está conectado à Internet das Coisas (SURESH; DANIEL; ASWATHY, 2014).

2.2 HISTÓRICO

Por volta de 1988, o experimento denominado ARPANET (*Advanced Research Projects Agency Network*), uma topologia de rede que interligava quatro universidades dos Estados Unidos, foi considerado como terminado. Após um período de transição até 1990, cedeu lugar à Internet, aplicando a ideia de uma rede internacional com conexões de redes regionais e nacionais nos Estados Unidos (ALMEIDA, 2005). Em outras palavras, as redes maiores passaram a servir de *backbones* (centrais de rede que interligam grandes cidades ou países), conectando redes regionais nos Estados Unidos com redes nacionais no exterior (KUROSE; ROSS, 2010).

A Internet foi, gradativamente, se transformando numa rede mundial pública que, segundo Almeida (2005), pode ser considerada uma “rede de redes”, onde qualquer pessoa ou computador, previamente autorizado, pode conectar-se para a troca de informações.

Na segunda metade da década de 1990, houve um período de grande crescimento e inovação para a Internet. Milhares de empresas com produtos e serviços começaram a surgir, sendo que Kurose e Ross (2010) destacam 4 (quatro):

- E-mail;
- Navegação *Web* e comércio pela *Internet*;
- Serviços de mensagem instantânea (*chat*);
- Compartilhamento *peer-to-peer* de arquivos.

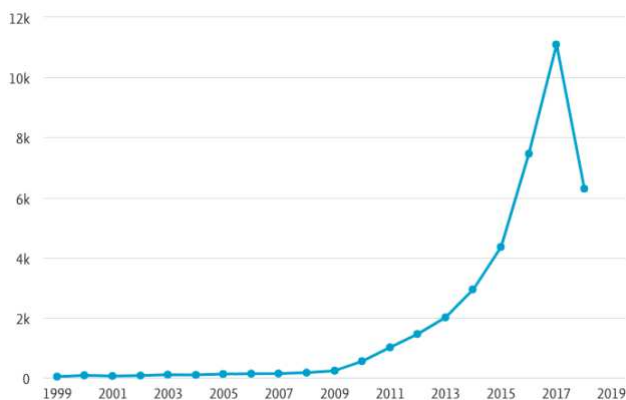
De 1995 a 2001 a Internet passou a influenciar o mercado financeiro e sendo que muitas empresas valiam bilhões de dólares na bolsa de valores, mesmo sem possuir um fluxo significativo de receita. Doravante, empresas como Google®, Yahoo®, Microsoft®, eBay®, Cisco® e Amazon® surgiram no mercado (KUROSE; ROSS, 2010).

O termo Internet das Coisas, do inglês *Internet of Things*, foi cunhado por Kevin Ashton em 1999 em uma apresentação à P&G®, grupo estadunidense responsável por grandes marcas de higiene e limpeza, onde apresentava a tecnologia RFID (*Radio Frequency IDentification*) (ASHTON, 2009).

Desde então, o termo Internet das Coisas vem agregando inúmeros conceitos, sendo a maioria deles adequados para o que a IoT realmente é. Não existe um único conceito ou uma única forma de se projetar um sistema IoT. Desde que o sistema colete informações de forma ubíqua, processe os dados e execute uma ação que contribua para o dia a dia do usuário, esta aplicação pode ser considerada uma aplicação de Internet das Coisas.

Na medida que as tecnologias avançam, contribuem também para a progressão da Internet das Coisas. O *Bluetooth*® e o RFID, por exemplo, são tecnologias de grande abrangência, que podem permitir que a IoT alcance seu total potencial (EVANS, 2011). Essas tecnologias vêm sofrendo atualizações e se adaptando ao universo IoT. Em 2010 foi introduzida a tecnologia *Bluetooth Smart*®, também conhecida como *Bluetooth Low Energy*® (BLE), onde tem por tradução literal, *Bluetooth Inteligente* ou *Bluetooth de Baixo Consumo*, com o objetivo de expandir o uso do *Bluetooth*® em aplicações de baixo custo e baixo consumo. Já o RFID está cada vez mais presente no dia a dia das pessoas, embutido em objetos como cartões e chaves de acesso, pulseiras para pagamentos instantâneos, bem como para o rastreamento de objetos.

Figura 3 – Produção científica para IoT na base Scopus, até o primeiro semestre de 2018



Fonte: Autor

Cerca de 15 anos após o termo Internet das Coisas ser citado pela primeira vez, a quantidade de produções científicas na base da Scopus retorna mais de 4000 documentos, sendo que o volume de produções tem aumentado vertiginosamente desde 2011, como pode ser visualizado na Figura 3.

2.3 TECNOLOGIAS DE COMUNICAÇÃO PARA A INTERNET DAS COISAS

Entre as tecnologias que promovem suporte para que a Internet das Coisas se torne realidade, destacam-se as tecnologias RFID, *Wi-Fi* e *Bluetooth®*.

Farahani et al. (2018) divide a arquitetura da IoT em quatro níveis distintos (Figura 4). A camada de detecção (sensores) se integra a todos os tipos de *hardware*, coletando dados ao se conectar ao mundo físico. Os dados são transferidos pela segunda camada, denominada camada de rede ou comunicação, entregando os dados à camada de serviços, que por sua vez, tem como objetivo cumprir os requisitos dos usuários. Na quarta e última camada (interface), as aplicações apresentam seus meios de interação com o usuário, entregando dados e recebendo novos comandos.

Figura 4 – Camadas da arquitetura IoT



Fonte: Adaptado de Farahani et al. (2018)

As tecnologias discutidas nesta seção encontram-se na segunda camada, a camada de rede. Seu principal papel é transportar os dados lidos pelos sensores, entregando-os às aplicações e aos serviços para que os dados sejam consumidos e disponibilizados ao usuário.

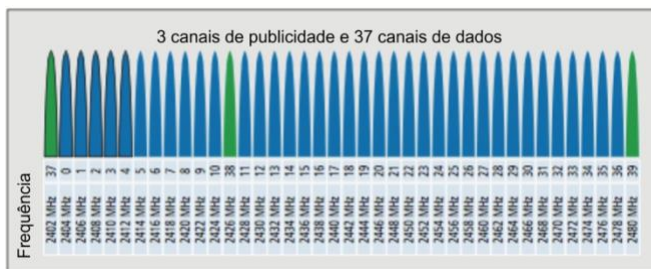
Cada uma destas tecnologias tem suas características próprias quanto ao consumo de energia, protocolo de comunicação, alcance e aplicações recomendadas. A seguir são apresentadas algumas das tecnologias que fornecem suporte a camada de rede da IoT.

2.3.1 Bluetooth®

Seguindo a tendência do desenvolvimento de tecnologias que visam facilitar o dia a dia das pessoas, o *Bluetooth*® foi criado com o intuito de interconectar dispositivos sem o uso de cabos. Com arquitetura simples e robusta, baixo custo e baixo consumo, esta tecnologia rapidamente foi aplicada em diversas áreas, interconectando dispositivos como aparelhos celulares, *notebooks*, fones de ouvido, entre outros (PRABHU; REDDI, 2004).

A tecnologia *Bluetooth*®, hoje conhecida como *Bluetooth Classic*®, foi uma grande evolução na interconexão de periféricos e dispositivos. Contudo, ela não oferece suporte para que a Internet das Coisas atinja seu potencial completo. Para tal, em 2010 foi introduzida a tecnologia *Bluetooth Smart*®, também conhecida como *Bluetooth Low Energy*®, em sua tradução literal, *Bluetooth* Inteligente ou ainda *Bluetooth* de Baixo Consumo, com o objetivo de expandir o uso do *Bluetooth* em aplicações de baixo custo e baixo consumo de energia.

Figura 5 - Plano de canais para o Bluetooth Low Energy



Fonte: Adaptado de Chang e Consulting (2014)

O BLE, apesar de ser derivado do *Bluetooth*® convencional, introduz um novo método de rádio comunicação. Os canais de comunicação da tecnologia *Bluetooth Classic*® são divididos por frequência, porém, no BLE, a largura dos canais é muito maior, diminuindo a interferência e permitindo que a intensidade da comunicação possa ser diminuída. Para organizar a utilização da faixa

de frequência disponível, canais de comunicação específicos para cada tipo de serviço foram determinados (Figura 5). O protocolo de comunicação é do tipo confiável (TCP – *Transmission Control Protocol*), desenhado em camadas e projetado especificamente para reduzir o consumo de energia (CHANG; CONSULTING, 2014).

Em resumo, pode-se dizer que os dispositivos BLE, se comparados aos dispositivos *Bluetooth*® comuns, se comunicam em uma menor quantidade de vezes entre si. Quando a comunicação acontece, os dados são entregues de forma confiável e esta entrega confiável é facilitada pelo baixo nível de interferência no meio de comunicação. Estes fatores permitem o baixo consumo de energia dos módulos *Bluetooth Low Energy*®, oferecendo potencial à tecnologia para o desenvolvimento de dispositivos IoT.

Outro ponto que coloca a tecnologia *Bluetooth*® a frente das outras no mercado IoT é o protocolo *Bluetooth Smart*®. A forma em que ele está estruturado permite que dispositivos sensores forneçam os dados lidos, também chamados de propriedades, encapsulados em perfis de atributos genéricos.

Isto permite, por exemplo, que qualquer aplicativo de celular faça a leitura de um termômetro ou uma pulseira leitora de batimentos cardíacos, pois os dados fornecidos pelo acessório *Bluetooth Smart*® estão encapsulados em um protocolo genérico que pode ser estudado e utilizado em aplicações por qualquer desenvolvedor (BLUETOOTH SIG, 2015).

2.3.2 *Wi-Fi*

Também conhecida como LAN (*Local Area Network*) sem fio IEEE 802.11, é o meio sem fio globalmente aceito para troca de dados, sinais e comandos (SURESH; DANIEL; ASWATHY, 2014). O *Wi-Fi* foi a tecnologia que se destacou dentre as diversas tecnologias que surgiram na década de 1990 como forma de interconectar dispositivos à Internet. Está presente em locais de trabalho, bares, restaurantes, cafês, aeroportos, praças, entre outros (KUROSE; ROSS, 2010).

É importante ressaltar que o *Wi-Fi* mesmo sendo uma tecnologia de comunicação sem fio, não pode ser totalmente considerada uma tecnologia móvel, já que este meio de comunicação possui um limite físico, que é o alcance do sinal *Wi-Fi*, cuja distância pode chegar, em geral, até 100 metros, dependendo dos obstáculos entre os aparelhos.

Por exemplo, esta tecnologia pode ser utilizada em um bar ou restaurante, para interconectar computadores portáteis a um ponto de acesso à *Internet*, sem utilização de cabos. Contudo, os usuários precisam ficar próximos ao ponto de acesso a fim de não perderem a conexão com a rede, indicando que nem sempre é possível associar o termo *wireless* (sem fio) ao termo mobilidade (SACCOL; REINHARD, 2007).

Das características da tecnologia *Wi-Fi*, destacam-se a capacidade de reduzir a taxa de transmissão para alcançar distâncias maiores, modos de conexão ponto a ponto, ou por meio de pontos de acesso, utilizando o protocolo de acesso ao meio CSMA (*Carrier Sense Multiple Access*) (KUROSE; ROSS, 2010). Das diferenças entre os padrões 802.11 (**b**, **a** e **g**), destacam-se algumas características da camada física. A Quadro 1 apresenta as diferenças de frequência e taxa de dados:

Quadro 1 – Padrões 802.11

Padrão	Faixa de frequência	Taxa de dados
802.11b	2,4 – 2,485 Ghz	Até 11 Mbps
802.11a	5,1 – 5,8 Ghz	Até 54 Mbps
802.11g	2,4 – 2,485 Ghz	Até 54 Mbps

Fonte: Adaptado de Kurose e Ross (2010)

No cenário da Internet das Coisas, a tecnologia *Wi-Fi*, permite que uma grande quantidade de dispositivos possa ser conectada a um único ponto de acesso, reduzindo a quantidade de cabos utilizados na instalação dos sensores. Todavia, deve ser considerado o alto consumo dos módulos *Wi-Fi* no desenvolvimento dos sensores, não sendo viável a utilização com baterias. O uso da faixa de frequência não licenciada (2,4 Ghz), disputando espaço com aparelhos *Bluetooth*®, telefones sem fio e controles remotos, provoca uma sobrecarga no ambiente, visto que a troca de dados via *Wi-Fi* necessita de uma conexão, mantendo o meio físico ocupado por mais tempo quando se compara a meios de comunicação como o *Bluetooth*® (SURESH; DANIEL; ASWATHY, 2014).

2.3.3 RFID e Código de Barras

O RFID e o Código de Barras possuem um conceito semelhante, onde basicamente um número identificador (ID - *Identity*) é utilizado para referenciar um determinado tipo de informação.

Os leitores de RFID e Códigos de Barras (Figura 6) são capazes de ler esta informação e, conseqüentemente, realizar uma busca na rede pela informação que é referenciada pelo ID.

Figura 6 - Código de barras e RFID



Fonte: Extraído de Suresh, Daniel e Aswathy (2014)

No código de barras, o ID é codificado em uma imagem que pode ser impressa em uma etiqueta de produto ou compartilhada digitalmente.

Um código de barras seria correspondente a uma *TAG* RFID. A *TAG* armazena os dados de identificação em um chip, que ao ser estimulado por um leitor RFID, responde informando o ID armazenado.

Os sistemas RFID podem ser usados para aplicações de sistema de localização em tempo real que permitem rastrear e localizar pessoas e objetos. Para a localização ocorrer, necessita-se de uma etiqueta RFID, que é a chave para muitos aplicativos de rastreamento e segurança. Exemplos de tais aplicações incluem um sistema de alarme que prevê situações perigosas, como uma criança que se aproxima de um forno quente. Outro exemplo é o uso de sistemas RFID nos hospitais para rastrear pacientes, equipes, equipamentos e itens. Esses aplicativos exigem alta precisão na determinação da localização do objeto rastreado (AL-KHATER; KUNHOTH; AL-MAADEED, 2017).

2.4 SEGURANÇA E PRIVACIDADE

A Internet das Coisas prevê um mundo conectado por bilhões de objetos (coisas) inteligentes e interativas capazes de oferecer todo tipo de serviços a entidades próximas e remotas. Os benefícios e o conforto que a IoT provocará são inegáveis, no entanto, estes podem ocorrer ao custo de uma perda de privacidade sem precedentes (LOPEZ et al., 2017).

Em ambientes inteligentes, segurança e privacidade são sempre as principais preocupações. Apenas pequenas quantidades de dados podem ser armazenadas em um dispositivo devido à capacidade de armazenamento limitada dos cartões de memória e é preciso garantir o

transporte e armazenamento destes dados para um local seguro. Pesquisadores já alertaram as indústrias sobre o "*hacking smart home*", termo que está sendo atribuído à quebra na segurança de sistemas IoT no contexto de *Smart Homes*, que é vista como uma ameaça real para a comunidade (AHMED et al., 2016).

Segurança e privacidade são requisitos muito importantes a serem considerados para o IoT. A heterogeneidade inerente da Internet conectada a objetos inteligentes, a capacidade de garantir a transmissão segura de informações confidenciais e os objetos físicos conectados a rede devem ser devidamente monitorados e controlados (AKPAKWU et al., 2017).

Segundo Lopez et al. (2017), os problemas de privacidade podem ser classificados em duas categorias: o usuário e a rede.

- O usuário pode ter sua privacidade violada pelo fato de estar cercado de sensores, monitorando tudo que ele está fazendo. A rede de sensores pode ser invadida e o usuário poderá ser espionado;
- Quanto à rede, uma entidade externa pode quebrar a criptografia dos dados e roubá-los, extraindo informações confidenciais. Algo que, nos dias atuais, já se tornou comum, apesar de não ser corriqueiro.

Alguns dos maiores fabricantes de *hardware* e *software*, como Amazon®, Samsung®, Apple®, Google®, entre outros, estão avançando no desenvolvimento de *frameworks* para IoT. Ammar, Russello e Crispo (2018) analisaram estes frameworks quanto aos padrões de segurança adotados e chegaram a conclusão que, apesar de cada *framework* encapsular sua lógica de segurança, todos seguem a mesma tendência e aplicam padrões semelhantes em alguns aspectos.

Quanto a privacidade, Jayaraman et al. (2017) apresentam modelos de segurança e privacidade em que os grupos de dados coletados são expressos aleatoriamente como uma soma de números, sendo recuperados e analisados em prol da IoT, sem que os dados reais dos usuários sejam expostos.

Em sua pesquisa sobre *Smart Healthcare*, Farahani et al. (2018) lembram do desafio que é manter um alto nível de segurança e privacidade neste cenário. As tecnologias baseadas em RFID e WSN (*Wireless Sensor Network*) podem ser utilizadas na identificação e rastreamento dos usuários. Os serviços devem oferecer autenticação, autorização, confidencialidade e integridade para todos os dados

coletados. Senhas e chaves devem ser autenticadas e duplamente verificadas com a codificação das *TAGs* RFID, maximizando o nível de proteção. Na camada de aplicação a identidade dos usuários e a privacidade do local precisam ser mantidas anônimas, minimizando o vazamento de informações pessoais.

Em uma entrevista com Tim Mackey, técnico da Black Duck Software® pertencente a Synopsys®, sinônimo em segurança de *software*, Mansfield-Devine (2018) contextualiza a IoT destacando pontos correlacionados a segurança nos sistemas de informação voltados para a Internet das Coisas.

Mansfield-Devine comenta sobre o senso de urgência, por parte dos fabricantes, em disponibilizar soluções IoT em seus produtos, beneficiando-se de ferramentas prontas, de baixo custo, fáceis e rápidas de serem implementadas. Esse senso de urgência prejudica a segurança dos *softwares* e, embora isso não seja um problema aparente, pode vir a causar grandes transtornos no futuro. Na entrevista, Mackey lembra os desastres na história e acrescenta:

"A história nos mostrou, praticamente desde o início do tempo da computação, que sempre há algum defeito de *software* que alguém poderá explorar eventualmente", diz ele. "Pode levar anos para aparecer, mas está lá."

Por exemplo, o DVR (*Digital Video Recorder*), um tradicional dispositivo para gravação de imagens em sistemas de segurança por vídeo, vem sendo adaptado para permitir a transmissão das imagens em tempo real, via Internet. Dispositivos como este são baseados em projetos antigos que não suportam, por exemplo, atualizações automáticas, também conhecidas como atualização OTA (*Over The Air*) deixando o aparelho vulnerável a pontos de quebra de segurança que futuramente possam ser encontrados. Sem possibilidade de atualização do *firmware*, estes produtos ficarão sempre vulneráveis enquanto estiverem em operação.

Em geral, os autores supracitados acreditam que as soluções existentes para proteção de dados sensíveis à privacidade na IoT, possuem pouco foco na segurança da informação e estrutura de rede. As falhas e aberturas expostas nos sistemas de segurança, junto ao precário trabalho realizado para proteger os dados, cria oportunidades para *hackers* e administradores mal-intencionados, permitindo roubo e divulgação de dados críticos e confidenciais. É preciso uma plataforma ou infraestrutura de IoT que garanta privacidade e segurança de ponta a ponta para seus usuários.

2.5 CENÁRIOS DA INTERNET DAS COISAS

A Internet das Coisas pode contemplar diferentes tipos de cenários. Dispositivos inteligentes podem ser utilizados em pacientes, coletando dados sobre seu estado de saúde, com base em sensores embutidos nas tecnologias vestíveis. As casas inteligentes detectam mudanças de temperatura, permitindo que sistemas de ar condicionado atuem em prol do conforto. Os sistemas de tráfego e transporte são monitorados pela IoT a fim de propiciarem cidades mais inteligentes. Sistemas de pagamento podem trabalhar juntamente com os bancos, possibilitando compras e pagamentos em segundos (FARAHANI et al., 2018).

Nesta seção são discutidos cenários IoT e algumas aplicações voltadas para a Internet das Coisas, suas características e potenciais.

2.5.1 *Smart Home*

A ideia de uma casa inteligente parte de um princípio onde os objetos domésticos estão conectados à Internet e podem tomar decisões de forma autônoma com base em informações provenientes de sensores, contribuindo para uma melhora no estilo de vida das pessoas (AKPAKWU et al., 2017).

Segundo Risteska Stojkoska e Trivodaliev (2017), a *Smart Home* é o último nível de controle da *Smart Grid*, que consiste em aplicar a IoT nas redes de energia elétrica, gerindo o consumo de energia, as fontes de energia renováveis e as linhas de transmissão. Ou seja, para a *Smart Grid*, as *Smart Home's* tem como papel fundamental contribuir para o consumo consciente de energia, utilizando as ferramentas de IoT para cumprir com este objetivo.

É importante salientar que esta infinidade de objetos inteligentes precisa conversar entre si. Os dispositivos e plataformas de serviços heterogêneos precisam ser totalmente interoperáveis, apoiando a execução conjunta e harmonizada das operações domésticas (TAO et al., 2018). Visando solucionar estas diferenças entre os aparelhos, podem ser utilizadas soluções baseadas em *frameworks* e computação em nuvem.

Os três pontos de vista sobre a *Smart Home* supracitados convergem em um universo impulsionado pelas TIC (Tecnologias da Informação e Comunicação), de forma que o desenvolvimento padronizado, com segurança e qualidade, produz sistemas mais eficazes

resultando em um ambiente inteligente com diversos objetos conectados. Ou seja, um mundo físico coberto por dispositivos sensores ubíquos, conectados, eficientes e perfeitamente incorporados ao cotidiano dos seres humanos (CURRY et al., 2018).

Ahmed et al. (2016) definem o termo “ambiente inteligente” como um pequeno mundo repleto de dispositivos conectados, dispositivos estes que cooperam entre si para aumentar a qualidade de vida do ser humano. O pequeno universo de atuadores, sensores e processadores acoplados aos objetos do dia-a-dia, provenientes da miniaturização da tecnologia, permite às “coisas” atuar de forma inteligente, ou seja, obter conhecimento e aplicá-lo de maneira a se adaptar as necessidades dos habitantes, melhorando sua experiência para com o ambiente.

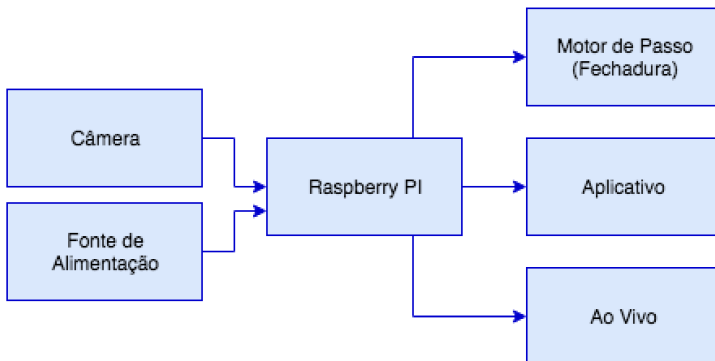
Curry et al. (2018) lembram também que a criação de uma experiência de usuário (*user experience* ou simplesmente *UX*) eficaz é um dos principais desafios em ambientes inteligentes, exigindo uma combinação de tecnologias, técnicas e habilidades interdisciplinares nas áreas de engenharia eletrônica, engenharia de dados e ciência de dados. Estes requisitos exigem uma maior curva de aprendizagem por parte dos desenvolvedores, mas em contrapartida, entregam sistemas mais eficientes e eficazes, tornando cada vez mais fácil a inclusão da Internet das Coisas no cotidiano das pessoas e, neste caso, nas *Smart Home*'s.

Para contribuir com a UX também podem ser utilizadas as redes de sensores sem fio, que prestam seus serviços sem interferir na rotina dos usuários, ou seja, são tecnologias não invasivas (JEBE MALAR; KOUSALYA; MA, 2017). Em sua pesquisa, Jeba Malar, Kousalya e Ma (2017) utilizam estas redes de sensores sem fio para monitorar a movimentação de pessoas, com foco em idosos e enfermos, em pequenos ambientes como casas e clínicas de recuperação, promovendo assim uma série de serviços inteligentes com base na localização destes usuários.

Dentro do cenário das *Smart Home*'s alguns cientistas costumam fazer alguma ligação com o termo *Home Security*, visto que muitas soluções são baseadas em “coisas” relacionadas à segurança, como por exemplo, os sistemas de monitoramento por câmera, os sistemas para controle da iluminação externa de forma automatizada, bem como, os controladores de acesso inteligentes. Isto parte da proposta da Internet das Coisas vista como uma necessidade de melhorar objetos existentes, tornando-os inteligentes, como demonstrado no projeto de Krishna, Reddy e Sandeep (2018), que propõem um sistema de reconhecimento

facial para o desbloqueio de fechaduras nas portas de uma casa inteligente.

Figura 7 – Sistema inteligente para desbloqueio de porta com fechadura eletrônica – diagrama de blocos



Fonte: Adaptado de Krishna, Reddy e Sandeep (2018)

A solução proposta por Krishna, Reddy e Sandeep é representada na Figura 7. Superficial e pouco detalhada, pode ser facilmente adequada a outras tecnologias e *frameworks*. O projeto utiliza um Raspberry Pi® para monitorar a câmera e fazer o controle da fechadura eletrônica. Por meio da aplicação desenvolvida o usuário pode visualizar as imagens ao vivo da câmera, bem como, bloquear e desbloquear a fechadura da porta. O acionamento da trava é comandado pelo Raspberry Pi®, que por sua vez comanda um motor de passo acoplado a fechadura.

2.5.2 Smart City

A IoT vem promovendo grandes mudanças nas indústrias, casas, na saúde e até mesmo na maneira como as cidades funcionam. O paradigma da Internet das Coisas está correlacionado à diversos domínios, como automação residencial, automação industrial, auxílio e assistência médica, assistência a idosos, gerenciamento inteligente de energia e redes, sistemas automotivos, gerenciamento de tráfego e muitos outros. No entanto, um campo de aplicação tão heterogêneo torna a identificação de soluções capazes de satisfazer os requisitos de todos um desafio formidável (ZANELLA et al., 2014).

Nesse cenário complexo, a aplicação do paradigma da IoT em um contexto urbano é de particular interesse, pois responde ao forte impulso

de muitos governos nacionais para adotar soluções de TIC na gestão de assuntos públicos, convergindo assim na chamada *Smart City* (Cidade Inteligente). O gerenciamento das cidades será tomado pelo gerenciamento de dados em tempo real, em todos os sistemas urbanos, incluindo a gerência de forma eficiente da água, da energia, do desperdício, dos serviços e dos meios de transporte entre cidades (KIM; RAMOS; MOHAMMED, 2017).

Nas *Smart Cities*, as tecnologias estão interligadas, garantido que os serviços básicos estão sendo oferecidos aos cidadãos, incluindo transporte, saúde, educação, monitoramento ambiental, monitoramento de tráfego, etc. (AKPAKWU et al., 2017).

Também é proposto um melhor aproveitamento dos recursos públicos, melhorando a qualidade dos serviços oferecidos aos cidadãos, enquanto uma redução de custos operacionais e administrativos pode ser esperada. A infraestrutura de comunicação pode oferecer acesso simples e unificado aos habitantes, contribuindo para a transparência política (ZANELLA et al., 2014).

Historicamente os grandes centros atraem a população rural e periférica, aumentando gradativamente a densidade demográfica das metrópoles. Visando atender o crescimento populacional e respeitar os requisitos das *Smart Cities*, Kim, Ramos e Mohamed (2017) sugerem o modelo representado pela Figura 8, apresentando os desafios e tecnologias encontrados em uma Cidade Inteligente:

Figura 8 - *Smart Cities* e IoT: Tecnologias disponíveis versus desafios



Fonte: Adaptado de Kim, Ramos e Mohamed (2017)

Segundo (Zanella et al. (2014), as *Smart Cities* enfrentam uma série de barreiras políticas, técnicas e financeiras. Do ponto de vista político, por exemplo, os líderes se mostram relutantes a descentralizar o poder de decisão, sendo necessário concentrar todo o processo de planejamento, decisão e execução em um único departamento.

Em uma *Smart City*, o setor dos transportes é crítico, uma vez que os sistemas de transporte inteligente devem contemplar algoritmos avançados para planejamento de rotas, sistemas de computação de alto desempenho e assim sucessivamente. Isto objetiva não só ajudar na resolução de congestionamentos de tráfego urbano, mas também garantir benefícios econômicos a cidade (YANG et al., 2017).

Os sistemas de transporte inteligentes, do inglês *Intelligent Transportation System* (ITS) são usados para garantir que a rede de transporte seja monitorada de forma eficiente e controlada. Grandes empresas do seguimento automotivo como a Ford®, Audi®, Mercedes-Benz®, empresas de tecnologia como a Google® e a Apple® ou empresas de ambos os seguimentos, como por exemplo, a Tesla®, estão revolucionando o mercado automotivo, com o desenvolvimento de carros autônomos (AKPAKWU et al., 2017).

Akpakwu et al. (2017) acrescentam ainda que, dentre os diversos sistemas presentes nestes veículos, como o GPS (*Global Positioning System*), o RFID e seus próprios sistemas embarcados para controle eletrônico do veículo, novos desenvolvimentos permitirão sistemas capazes de realizar comunicação veículo-veículo, abrindo um leque de possibilidades no âmbito da Internet das Coisas.

Dentre outros sistemas e serviços presentes nas Cidades Inteligentes, cabe citar o monitoramento da saúde estrutural dos edifícios, a gestão de resíduos, o monitoramento da qualidade do ar, além do monitoramento de ruído, monitoramento de tráfego, serviços de estacionamento inteligente, iluminação inteligente, além de muitos outros serviços e sistemas. Estas funcionalidades garantem a administração da cidade dezenas de benefícios, como por exemplo, melhor controle da emissão de poluentes e descarte correto de resíduos, atingindo a metas ambientais, promovendo vitórias econômicas e ecológicas (ZANELLA et al., 2014).

2.5.4 Smart Healthcare

Problemas como o envelhecimento da população mundial, doenças crônicas, mortalidade infantil, à falta de recursos naturais,

epidemias e condições sanitárias precárias, são desafios enfrentados em todas as regiões do mundo. Mesmo com o aumento da demanda por serviços médicos, as consultas e atendimentos continuam seguindo o modelo tradicional, onde os pacientes procuram as clínicas quando adoecem ou percebem anormalidades em sua saúde (FARAHANI et al., 2018).

A inserção da IoT no sistema médico e de saúde, vem sendo denominado *Smart Healthcare*, ou seja, Saúde Inteligente. Novos dispositivos sensores, baseados em tecnologia vestível (*wearable technology*) são conectados aos pacientes, coletando dados médicos e sinais vitais como, pressão sanguínea, temperatura corporal, nível de colesterol, frequência cardíaca, entre outros.

Isto diminui a distância entre pacientes e médicos, visto que os médicos podem realizar uma análise em tempo real de seus pacientes, acompanhando progressos, constatando anomalias e prevenindo doenças. Um grande avanço em estudos científicos na busca por diagnósticos e curas também pode ser esperado, considerando a quantidade incalculável de dados gerados por estes sensores (AKPAKWU et al., 2017).

Ainda em relação ao envelhecimento da população, doenças comuns em pessoas idosas, como Alzheimer, diabetes, doenças cardiovasculares, osteoartrite, etc., em geral provocam declínio progressivo das habilidades físicas e cognitivas, impedindo que estas pessoas vivam de forma independente em suas próprias casas. Com o surgimento dos ambientes inteligentes baseados em Internet das Coisas, esta parcela da população pode viver de forma independente por mais tempo, pois estes sistemas monitoram e avaliam qualquer condição de saúde que o idoso possa ter, além de monitorar como realizam suas atividades diárias. Outra vantagem é o alívio de pressão exercido sobre os serviços de saúde, permitindo à estes se tornarem mais sustentáveis (MSHALI et al., 2018).

Farahani et al. (2018) complementam destacando que a aprendizagem de máquinas (*machine learning*) e a análise de dados (*data mining*) podem ser usadas para prever condições de saúde, como cânceres, infecções e ataques cardíacos. O resultado esperado é que as pessoas sejam diagnosticadas com antecedência, ou que ao menos, sejam alertadas a procurar orientação médica aumentando as chances de cura.

Para alcançar um ambiente verdadeiramente inteligente do ponto de vista da saúde, existem ainda vários desafios. Cita-se o monitoramento remoto do ambiente, a tecnologia de comunicação

necessária, a existência de sistemas inteligentes para processamento (análise, tomada de decisões relevantes, etc.), e a prestação de serviços sensíveis ao contexto (MSHALI et al., 2018).

2.6 SISTEMAS OPERACIONAIS MÓVEIS

Sistemas operacionais representam um conjunto de *softwares* cujo objetivo é intermediar a camada de aplicação, operada pelo usuário, com a camada de *hardware*. São responsáveis por apresentar ao usuário uma experiência amigável, enquanto uma vasta quantidade de tarefas é distribuída ao *hardware*, buscando o máximo de eficiência possível.

Especificamente, segundo Afzal et al. (2017), um Sistema Operacional (SO) permite facilidade de uso de um determinado dispositivo, oferecendo portabilidade, suporte a operações envolvendo mais de um processo computacional e acesso a bibliotecas de desenvolvimento *online*.

Até o lançamento do primeiro iPhone®, em 2007, os sistemas operacionais eram conhecidos por sua presença em laptops e computadores de mesa, sendo os mais comuns o Windows®, Linux® e Mac OS®. Contudo, com a chegada do primeiro *smartphone* da Apple®, a primeira versão do iOS® foi apresentada, dando origem ao termo Sistema Operacional Móvel (*Mobile*) (NOVAC et al., 2017).

Para a Internet das Coisas, os SOs *mobile* são parte fundamental no processo de comunicação com os dispositivos (coisas). Por exemplo, um dos principais *frameworks* voltados para a IoT, o Home Kit®, está totalmente ligado ao funcionamento do iOS, sendo o SO da Apple® o principal responsável pela maioria das iterações com os dispositivos (AMMAR; RUSSELLO; CRISPO, 2018).

Os sistemas operacionais móveis mais comuns são o Android®, uma plataforma *Open Source* mantida pela Google® e seus parceiros, e o iOS®, uma plataforma privada e de propriedade da Apple®. Cabe citar também o Windows Phone®, Windows 10® e o Blackberry OS®, que apesar de não serem mais comercializados pelos fabricantes (CARVALHO, 2018; FILIPE GARRETT, 2017; JUNQUEIRA, 2016), detêm uma pequena e específica parcela do mercado (OUHBI et al., 2015).

Em uma análise comparativa, Novac et al. (2017) destacam vantagens e desvantagens dos SOs, comparando o Android®, o iOS® e o Windows Phone®. Pontos positivos como as lojas para *download* de aplicativos, a capacidade de receber e executar atualizações e sua

integração com outros aparelhos, além dos *smartphones*, são pontos determinantes para a IoT. A integração que estes sistemas propiciam com *Smart TVs* (TV inteligente), *Smartwatches* (relógio inteligente), caixas de som, equipamentos de segurança e iluminação, demonstram que a Internet das Coisas vem se consolidando.

Esta forte relação entre a Internet das Coisas e os Sistemas Operacionais Móveis também pode ser acompanhada nas bases de pesquisa alimentadas pela comunidade de cientistas e desenvolvedores. A base Scopus, por exemplo, retorna centenas de trabalhos relacionados aos termos “Android IoT” e “iOS IoT”, totalizando aproximadamente 500 documentos de 2009 a 2018.

Além de integrarem a Internet das Coisas à Sistemas Operacionais Móveis, os projetos citados nesta seção, assim como a maioria dos projetos baseados em IoT, também se enquadram no conceito de sistemas embarcados, que é discutido na seção a seguir.

2.7 SISTEMAS EMBARCADOS

A maioria das aplicações IoT são compostas por dispositivos restritos de recursos e com limitações quanto ao consumo de energia. Devido à sua menor capacidade computacional, esses dispositivos economizam custo e espaço, já que são incorporados neles somente os recursos de *hardware* específicos e extremamente necessários para aplicação. Estas características se assemelham com as características dos sistemas embarcados, sendo este cenário crucial para uma aplicação da Internet das Coisas (AFZAL et al., 2017).

Do inglês *embedded systems*, os sistemas embarcados são pequenos sistemas computacionais projetados para realizar tarefas específicas que contribuem para o funcionamento geral de um equipamento ou sistema maior. Normalmente, não são capazes de executar pacotes convencionais de *software*, não permitem a instalação de aplicativos e nem sequer possuem interface com o usuário. São utilizados de forma restrita e exclusiva para servir ao produto onde está embutido/embarcado (HELPS; ARMSTRONG, 2007).

Helps e Armstrong (2007) exemplificam: um automóvel, por exemplo, é repleto de sistemas embarcados. Responsáveis por gerenciar o sistema de combustão do veículo (injeção eletrônica), o painel de informações (velocidade, combustível, temperatura, etc.), o sistema de climatização, a central de entretenimento, etc., estes sistemas trabalham de forma independente, embarcados em um sistema maior (veículo) e contribuindo para o funcionamento geral deste.

Nos últimos anos, segundo Gadre et al. (2017), os sistemas embarcados estão sendo equipados com conectividade à Internet podendo ser denominado como um dispositivo IoT. Em outras palavras, isto representa a evolução dos sistemas embarcados em dispositivos inteligentes, com capacidade para tomada de decisões e interações com outros sistemas a níveis mais complexos.

É de suma importância que os sistemas embarcados continuem evoluindo para dispositivos IoT e contribuindo na propagação da Internet das Coisas. Com uma proposta semelhante à deste trabalho, krishnamurthy e Cecil (2018) também visam acelerar o desenvolvimento de aplicações voltadas para a Internet das Coisas, mais especificamente em relação aos sistemas embarcados. Eles estão trabalhando em um *framework* colaborativo baseado em óculos de realidade virtual (VR - *Virtual Reality*) para auxílio na montagem de placas eletrônicas, também conhecidas como placas de circuito impresso (PCB – *Printed Circuit Board*).

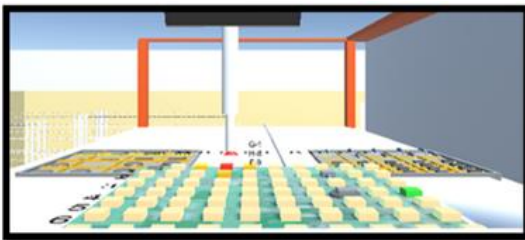
Com o auxílio da ferramenta chamada de VAE – *Virtual Assembly Environment*, engenheiros e técnicos tem acesso a visualizações privilegiadas da placa eletrônica, como demonstrado nas Figuras 9 e 10:

Figura 9 – Usuário interagindo com o sistema de realidade virtual



Fonte: Krishnamurthy e Cecil (2018)

Figura 10 – Visão do usuário no sistema de realidade virtual



Fonte: krishnamurthy e Cecil (2018)

Na medida que os sistemas embarcados se modernizam, beneficiando-se de ferramentas como a proposta por Krishnamurthy e Cecil (2018), estes dispositivos passam a ser denominados “dispositivos IoT” (GADRE et al., 2017), por possuírem atributos como conexão à Internet e interações inteligentes com pessoas e coisas.

2.8 FRAMEWORKS

Segundo Minetto (2007), um *framework* pode ser considerado uma base, uma coleção de códigos, classes, funções, técnicas e metodologias que dão suporte ao desenvolvimento de um novo *software*.

Outra definição de *framework* é apresentada como um conjunto de classes que colaboram, a fim de resolver uma necessidade da aplicação, uma abstração que une códigos comuns entre diversos projetos de *software*, provendo uma funcionalidade genérica (BEVILAQUA, 2013).

Ou seja, um *framework* pode ser resumido em um conjunto organizado de código, junto a diversas bibliotecas que se relacionam representando uma solução incompleta. Este esqueleto da aplicação, que pode ser customizado, está organizado e agrupado de forma a facilitar a progressão do projeto, garantindo maior segurança e organização (SILVA, 2011).

Do ponto de vista dos padrões de projeto, as vantagens de se utilizar um *framework* durante o desenvolvimento de um *software* estão ligadas a separação do código em parte lógica e apresentação (*interface* com o usuário), a maior facilidade durante os testes, a geração de documentação e também a aplicação do conceito denominado DRY, do inglês *Don't Repeat Yourself* ou, simplesmente, “não se repita”. Dessa forma, as funções que se repetem, ou se assemelham, podem ser embutidas em uma biblioteca, otimizando o código, a aplicação e reduzindo o tempo de desenvolvimento (MINETTO, 2007).

Silva (2011) afirma que existem muitas vantagens quanto a utilização de *frameworks*, entre elas, uma maior segurança, economia de tempo e suporte. As bibliotecas tendem a prevenir erros durante o desenvolvimento, evitando falhas em tempo de execução, quando o *software* já está sendo utilizado pelo cliente. Na maioria das vezes é disponibilizado aos programadores toda documentação de *software* da ferramenta, explanando suas funcionalidades e demonstrando suas características.

Portanto, utilizar um *framework* durante um projeto de *software* gera maior facilidade e agilidade no desenvolvimento, as aplicações resultantes são mais seguras, com melhor manutenibilidade e de qualidade superior (BEVILAQUA, 2013).

Por outro lado, a curva de aprendizagem para utilização de um *framework* pode ser elevada, uma vez que suas metodologias próprias para desenvolvimento podem exigir um pouco de esforço do programador durante o decorrer do projeto, sendo essas, algumas das desvantagens da utilização de *frameworks* (SILVA, 2011).

Dos diversos *frameworks* existentes, a maioria é voltado para aplicações *web*, devido a vasta demanda nesta área. Contudo, desde que o termo Internet das Coisas foi cunhado, muitos cientistas estão interessados em conceitualizar e desenvolver *frameworks* voltados para a IoT. Na seção a seguir será discutido sobre estas ferramentas e suas contribuições para a Internet das Coisas.

2.8.1 Frameworks IoT

O conceito de *Framework* IoT está ligado à coordenação e controle dos processos de elementos presentes na Internet das Coisas. Um *Framework* IoT é um conjunto de regras, protocolos e regulamentos que organizam a forma de processar os dados e trocar mensagens entre todas as partes envolvidas (por exemplo, dispositivos embutidos, nuvem e usuários finais). Ele também deve suportar a implementação de alto nível de aplicativos, ocultando a complexidade dos protocolos referente a infraestrutura de rede (AMMAR; RUSSELLO; CRISPO, 2018).

Segundo Maniappan (2015), é a evolução da Internet das Coisas para uma abordagem que vai além de dispositivos e soluções de conectividade, incluindo protocolos, suporte a diversos aplicativos e provedores em nuvem, habilitando o alcance da plataforma a diversos setores como a indústria, a medicina e o estilo de vida.

Os *Frameworks* IoT também possibilitam a prototipagem ágil, com foco em gerar protótipos funcionais a partir de inovações idealizadas por estudantes em laboratórios e universidades. Esta visão tem sido aplicada em cursos de engenharia e de acordo com as pesquisas realizadas as abordagens foram bem sucedidas (KARVINEN; KARVINEN, 2018).

A evolução dos *frameworks* voltados para a Internet das Coisas tem seguido duas abordagens básicas, uma com foco em protocolos, estruturas de rede e plataformas em nuvem, outra com foco em produtos, dispositivos e aplicativos. Essas abordagens são mencionadas

em Karvinen, Karvinen (2018), Ammar, Russello e Crispo (2018) e Maniappan (2015), mas também são citadas por diversos autores de forma indireta, sem defini-las, diferenciá-las ou compará-las.

A maioria das grandes empresas do ramo de tecnologia focam em estruturas que promovem suporte a protocolos de rede, plataforma em nuvem e ferramentas IoT enquanto que a comunidade científica, as universidades e as *Startups* apresentam abordagens com foco em dispositivos e ferramentas para o desenvolvimento de aplicativos. Os trabalhos e publicações encontrados, relacionados a *Frameworks* IoT, estão detalhados na seção a seguir.

2.8.2 Trabalhos e *Frameworks* Relacionados

Trabalhos de pesquisa, com foco em *frameworks* IoT, tem sido publicados e, em geral, mencionam as arquiteturas voltadas para a Internet das Coisas, oportunidades de mercado, protocolos e tecnologias de comunicação, além dos principais desafios. Nesta subseção, são comentados trabalhos e *frameworks* encontrados durante a pesquisa, abordando seus pontos principais com relevância para este trabalho.

A Allseen Alliance, uma comunidade formada por mais de 200 organizações, incluindo Microsoft®, LG® e Qualcomm®, possui foco em garantir a interoperabilidade de bilhões de dispositivos, aplicativos e serviços para a Internet das Coisas. Os estudos são baseados em um *framework* de código aberto (*open source*) construído em conjunto com a comunidade organizações filiadas (ALLIANCE, 2017).

Ammar, Russello e Crispo (2018) apresentam uma pesquisa que detalha o funcionamento dos *frameworks* desenvolvidos pelas principais empresas na área de tecnologia, incluindo Amazon®, Apple®, Microsoft® e Samsung®, com foco nos mecanismos de segurança por eles empregados.

A IoT League®, uma plataforma digital com objetivo de manter um repositório central de informações acerca da Internet das Coisas, publicou uma lista de 10 *frameworks* IoT, buscando capturar detalhes destas estruturas que visam aumentar a interoperabilidade entre os dispositivos (IOTLEAGUE, 2018).

De acordo com a pesquisa de Maniappan (2015), os *frameworks* devem oferecer suporte para a estrutura IoT que é representada pela Figura 11. Apesar da composição dos recursos variar ao longo do tempo, a fim de se adaptar às evoluções da Internet das Coisas, este escopo continua sendo mantido.

Figura 11 – Estrutura de um sistema IoT



Fonte: Adaptado de Maniappan (2015)

Os *frameworks* desenvolvidos pelas grandes empresas possuem uma infinidade de recursos, sendo muitos deles pagos, visando sistemas de grandes empresas na área de TI. Por exemplo, a empresa Microsoft® lançou o Azure IoT Suite®. A plataforma da Microsoft® contempla um conjunto de serviços que permite aos usuários finais a interação com seus dispositivos IoT, o recebimento de dados, a realização de diversas operações em relação aos dados e a visualização de forma adequada para as empresas. Oferece suporte a uma ampla gama de dispositivos, sistemas operacionais e linguagens de programação, abordando o desafio de ter uma estrutura IoT completa (AMMAR; RUSSELLO; CRISPO, 2018).

Para suportar diferentes dispositivos e plataformas da Internet das Coisas, a Microsoft® fornece vários SDKs (*Software Development Kit* ou Kit de Desenvolvimento de Software). Desta forma, os desenvolvedores são capazes de se conectar ao *hub* do Azure®, permitindo que os usuários gerenciem seus dispositivos. Estes SDKs podem ser utilizados para implementar aplicativos em diversos tipos de dispositivos, sendo que as linguagens de programação mais comuns e tradicionais já são suportadas.

Já a PTC®, uma provedora global de tecnologia na área de Internet das Coisas, Realidade Aumentada (*AR - Augmented Reality*), dentre outros serviços, como gerenciamento de ciclo de vida de produtos e serviços de projeto assistido por computador, mantém um framework voltado para Internet das Coisas denominado ThingWorxs®.

Segundo a PTC (2018), o ThingWorxs® é uma plataforma avançada de inovação industrial, projetada para fornecer rapidamente aplicativos de IoT e RA (Realidade Aumentada). Inclui ferramentas e tecnologias que permitem às empresas se desenvolverem, implantando e ampliando aplicativos direcionados a Internet das Coisas e a Realidade Aumentada.

Seu conteúdo está disponível em diversas línguas, incluindo o português brasileiro nas principais páginas. Possui uma poderosa

ferramenta de desenvolvimento com base em HTML (HyperText Markup Language), com suporte aos principais navegadores atuais e uma documentação completa, com artigos ilustrados e acompanhamento passo a passo.

O ThingWorxs® em si é um grande gestor de informações, permitindo uma avaliação de produtos e serviços IoT e total gerenciamento das aplicações. Possui ferramentas para executar modelagem de aplicativos de ponta a ponta e criar interfaces de usuário, e também possui um motor de execução direcionado à eventos, gerenciamento de dispositivos (coisas) e ferramentas de inteligência (MANIAPPAN, 2015).

Contudo, o desenvolvimento de aplicações IoT propriamente dito, fica a cargo de um repositório denominado ThingWorxs Marketplace®, que agrupa dezenas de parceiros provedores de ferramentas e *kits* de desenvolvimento. Essas ferramentas são mantidas por seus próprios desenvolvedores, sendo estes também responsáveis pelo suporte e documentação.

O Amazon AWS® é uma estrutura de back-end IoT pertencente à Amazon®, uma das maiores empresas de tecnologia na era da Internet. A plataforma permite fácil interação entre dispositivos e aplicativos podendo comportar bilhões de dispositivos e trilhões de mensagens (AMAZON WEB SERVICES, 2017). Permite processar e atuar sobre os dados de dispositivos, sendo que os aplicativos podem interagir com dispositivos mesmo quando estão *off-line* (MANIAPPAN, 2015).

O AWS® também traz maior facilidade para interconectar os diversos serviços oferecidos pela Amazon®, que em geral, já são utilizados na maioria das aplicações presentes na web. Fornece bibliotecas de código aberto e SDK para diversas plataformas de *hardware*, além de não restringir quanto a linguagens de programação ou plataformas de acesso (AMMAR; RUSSELLO; CRISPO, 2018). Dessa forma, o desenvolvedor pode escolher as plataformas que melhor se adaptam ao seu projeto, como aplicações mobile, desktop, Arduino®, Raspberry Pi®, dentre outros sistemas proprietários ou embarcados.

A Cisco®, uma das empresas mais tradicionais no ramo da tecnologia, principalmente no setor de redes, também está voltando parte de seus esforços para o mundo IoT, oferecendo produtos e consultoria, bem como, soluções para a indústria a fim de melhorar a produtividade e eficiência na manufatura, mineração, petróleo e gás, transporte e utilitários (MANIAPPAN, 2015).

O IoT Dev Center® é um portal mantido pela Cisco® (CISCO, 2018), onde são agrupadas as informações relacionadas aos produtos

oferecidos, documentação e suporte. É neste portal que os desenvolvedores têm acesso a informação necessária para interconectar seus produtos as ferramentas da Cisco®, visando acelerar o processo construtivo de aplicações voltadas para a Internet das Coisas.

Já a Google®, uma das maiores empresas do mundo, mantém um conjunto de plataformas denominadas Brillo® e Weave®, voltado à implementação acelerada de aplicações IoT. Brillo® e Weave® operam em conjunto, sendo o primeiro um Sistema Operacional baseado em Android®, e o segundo uma *interface* de comunicação para interações e troca de mensagens com os dispositivos. Juntos as duas partes se complementam, formando uma estrutura para a Internet das Coisas (AMMAR; RUSSELLO; CRISPO, 2018).

O Home Kit® da Apple® é uma plataforma dedicada a dispositivos IoT conectados em casa. Utilizando aparelhos com o sistema operacional iOS® (dentre outros sistemas proprietários da Apple®) é possível gerenciar e controlar os acessórios conectados na casa do usuário, por meio de um aplicativo denominado Home®.

Além disso, os usuários podem explorar os recursos de seus aparelhos Apple®, integrando-os com os dispositivos Home Kit®. Por exemplo, o usuário pode chamar a Siri® (serviço de comandos por voz da Apple®) e solicitar que se acendam as luzes ou, que toque uma música nas caixas de som da sala (APPLE, 2018).

Se as grandes empresas estão focadas em entregar um sistema computacional completo, complexo, fechado e caro, a fim suportar grandes sistemas IoT, com milhares de dispositivos interconectados, movimentando um imenso volume de dados, boa parte da comunidade de cientistas está focada em entregar um *framework* leve, prático e fácil de usar, com objetivo de acelerar a entrega das aplicações e facilitar o desenvolvimento.

Karvinen e karvinen (2018) apresentam uma configuração para prototipagem ágil em uma sala de aula, combinando as habilidades necessárias para desenvolver um sistema IoT. Operando dispositivos de prototipagem básica como o Arduino Uno® e um *software* na linguagem *Python*, os alunos não precisam se preocupar com a comunicação via Internet, troca de dados entre cliente e servidor, manipulação de dados, bem como outros requisitos básicos de um sistema para a Internet das Coisas.

Os alunos focam no protótipo real e seus atributos, otimizando o tempo e os custos do projeto. A proposta foi validada com duas turmas de aproximadamente 20 alunos e o método de prototipagem rápida utilizando ferramentas gratuitas e de fácil acesso foi considerado eficaz.

O trabalho de Krishnamurthy e Cecil (2018), também baseia-se no desenvolvimento ágil, voltado para produção em massa de *hardware*, utilizando óculos de realidade virtual para auxiliar no processo produtivo de placas de circuito impresso.

Ng e Wakenshaw (2017) propõem um *framework* para projetistas e desenvolvedores de *software* visando gerar benefícios para os usuários finais de IoT, de modo que estes se sintam mais confiantes experimentando sistemas de melhor qualidade.

Sendo assim, as plataformas IoT tem evoluído a partir de duas abordagens. Aquelas que fornecem dispositivos e suas soluções de conectividade em redes privadas que mudaram para a Internet/nuvem pública e outras, com foco nos protocolos IoT e no suporte à aplicativos do fornecedor da nuvem para aumentar o alcance das plataformas em variados setores industriais.

3 FRAMEWORK PROPOSTO

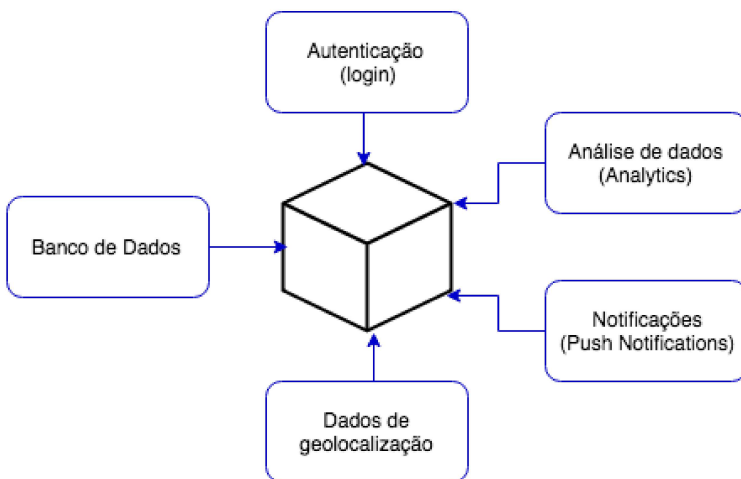
Neste capítulo é apresentado o *framework* proposto, descrevendo os requisitos, o processo de desenvolvimento, as ferramentas utilizadas e as etapas necessárias até a elaboração final do mesmo.

3.1 INTRODUÇÃO

O objetivo do *framework* consiste em auxiliar no desenvolvimento de aplicações multiplataforma, voltadas à Internet das Coisas. É composto por diversos métodos e blocos de códigos que visam otimizar o trabalho de desenvolvedores, deixando tarefas comuns em sistemas com foco em IoT a cargo do *framework*, podendo o desenvolvedor se concentrar nas regras de negócio da aplicação.

Serviços de *login*, banco de dados, notificações e geolocalização, por exemplo, são gerenciados pelo *framework*, ficando a cargo do desenvolvedor somente a inicialização e a invocação dos métodos que representam as funcionalidades. O *framework* inicializa e interage com os serviços de terceiros, entregando somente as informações necessárias à aplicação. Ao invés de interagir com diversos serviços e APIs, o programador irá interagir com uma única biblioteca (Figura 12) que engloba todos os serviços relevantes para uma aplicação de IoT.

Figura 12 – *Framework* proposto para o desenvolvimento de aplicações IoT

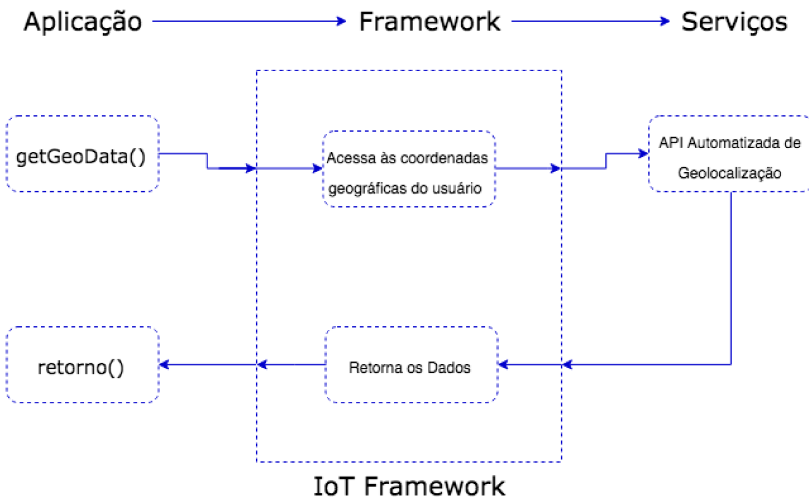


Fonte: Autor

Por exemplo, supondo que uma determinada aplicação móvel necessite descobrir informações relevantes com base nas coordenadas geográficas do usuário, o desenvolvedor faria uma simples chamada de método e o *framework* realizaria todo o trabalho, conversando com o sistema operacional para ler dados do GPS do aparelho, interagindo com a Google Maps API® para descobrir informações sobre a localização, interpretando e organizando os dados.

Após alguns instantes o *framework* entregaria à aplicação uma série de dados relevantes com base na localização do usuário a partir de uma simples chamada de método. Além de reduzir a incidência de erros e falhas, o desenvolvedor pode se preocupar com as regras de negócio da aplicação, reduzindo o prazo de entrega dos projetos e manutenção do sistema. O fluxo deste exemplo está representado na Figura 13:

Figura 13 – Exemplo de uma aplicação IoT com dados de geolocalização



Fonte: Autor

É importante ressaltar que, a estrutura proposta considera aplicações IoT multiplataforma, ou seja, aplicativos *web*, aplicativos mobile (Android®/ iOS®) e sistemas embarcados. Cada tipo de plataforma deverá receber uma implementação específica do *framework*, considerando as particularidades de cada ambiente.

O desenvolvimento do *framework* consiste em 3 etapas principais, sendo elas:

- Levantamento dos requisitos do *framework*;
- Documentação de *software*;
- Desenvolvimento de *software*.

Nas seções a seguir serão discutidas algumas das características presentes na maioria das aplicações IoT e que com possibilidade de serem agrupadas em um *framework*. Após este levantamento, são definidos os requisitos funcionais e não funcionais do *framework* proposto, seguidas pelas etapas de documentação e desenvolvimento de *software*.

3.2 AUTENTICAÇÃO (*LOGIN*)

Dhillon e Kalra (2017), consideram a autenticação do usuário uma forma incondicional para garantir a segurança do mesmo na IoT. Isto ocorre devido aos serviços fornecidos por inúmeros aplicativos voltados à saúde, segurança, vigilância, entrega de produtos, entre outros, que podem ser acessados pelo usuário de qualquer lugar. Sendo assim, os autores conceituam que é indispensável a autenticação do usuário nos aplicativos, a fim de torná-lo seguro e possibilitar sua privacidade, garantindo que os mesmos não sejam invadidos e evitando que as informações do usuário sejam expostas à terceiros. Nesta direção, os autores elaboraram um esquema de autenticação de usuário remoto baseado em biometria leve para serviços de IoT.

Ainda no entendimento de Dhillon e Kalra (2017), a autenticação do usuário pode ocorrer de três formas:

- Por objetos que o usuário mantenha consigo, como cartões, *TAGs* RFID, tokens, etc.;
- Por senhas, palavras passe e assinaturas digitais que o usuário memorize;
- Por autenticação biológica utilizando sensores biométricos, reconhecimento facial ou reconhecimento de voz.

Para os autores, senhas e/ou objetos que o usuário possua, com o objetivo de comprovar sua autenticação, são facilmente violáveis. Deste modo, defendem a autenticação por traços biométricos como uma forma mais eficiente de garantir a segurança e privacidade do usuário. Mostrando, portanto, a importância do *login* no âmbito da Internet das Coisas.

Além de Dhillon e Kalra muitos outros pesquisadores vêm trabalhando em ferramentas de autenticação visando produtos para a Internet das Coisas, como por exemplo, Usha Devi et al. (2015) e Bae e Kwak (2017). Contudo, existem ferramentas de autenticação prontas, seguras e robustas, desenvolvidas especificamente para isso, mantidas por grandes empresas como Google® e Facebook®.

O Firebase Authentication® é um dos serviços do pacote Firebase®, mantido pela Google®. O serviço consiste em uma plataforma popularmente conhecida como *back-end*, que realiza uma série de tarefas necessárias na grande maioria dos sistemas computacionais.

Em geral o *back-end* de uma aplicação é responsável por persistir e recuperar os dados do banco de dados, realizar tarefas em segundo plano ou disparar ações em um determinado momento. Também é responsável por aceitar requisições e efetuar respostas as requisições do *front-end*, que é a camada de interface com o usuário.

As bibliotecas e SDKs do Firebase® são muito bem documentadas e de fácil implementação. No caso do Firebase Authentication® elas promovem suporte à autenticação por meio de senhas, números de telefone e provedores de identidade como o Google®, Facebook®, Twitter®, entre outros (GOOGLE INC, 2017). Outra importante vantagem é que a autenticação realizada neste serviço é pré-requisito para o funcionamento de outros serviços fornecidos pelo Firebase®, ou seja, a comunicação com ferramentas como banco de dados, *push notifications* e *analytics* estão protegidas pela criptografia e autenticação do Firebase Authentication®.

Baseado no mesmo conceito, o Facebook Account Kit® apresenta praticamente as mesmas funcionalidades, com o diferencial de permitir a autenticação com base em um número de telefone e também de fornecer o código fonte para a interface gráfica da tela de *login*. Ou seja, otimiza e facilita ainda mais o desenvolvimento de aplicações que necessitem de ferramentas de autenticação, disponibilizando toda a programação necessária para o funcionamento da ferramenta.

Ao contrário do que a maioria acredita, o Facebook Account Kit® não é uma ferramenta para os usuários realizarem *login* com sua conta/perfil do Facebook®. Na verdade, basta que o usuário possua uma conta de *e-mail* ou um número de telefone celular ativo para realizar o *login*. Iniciando o fluxo de autenticação, seja por *e-mail* ou pelo número de telefone, o *back-end* do Facebook Account Kit® envia uma confirmação (por *e-mail* ou por SMS) para garantir a autenticação do

usuário. Ao receber a confirmação o acesso à aplicação é liberado (FACEBOOK, 2018).

Portanto, o trabalho realizado por estas ferramentas encapsula uma série de etapas necessárias ao desenvolvimento de um sistema, acelerando o desenvolvimento e permitindo mais espaço para o desenvolvedor explorar as regras de negócio da aplicação. Este fato motivou a seleção das ferramentas Firebase Authentication® e Facebook Account Kit® para o pacote de recursos presente no *framework* proposto.

3.3 BANCO DE DADOS

Algo muito importante a ser observado no projeto de sistemas de IoT, principalmente em sistemas para casas inteligentes, é o projeto do banco de dados. O desenvolvimento tecnológico de um sistema IoT certamente tem impacto na maior quantidade de dados que precisam ser armazenados e, assim, o projeto de um banco de dados eficiente será necessário para economizar espaço de armazenamento (ADIONO et al., 2017).

Existem algumas classes de sistemas de banco de dados, sendo a mais tradicional conhecida como RDBMS (*Relational Database Systems*) que se baseia no modelo relacional que utiliza como linguagem operacional o SQL (*Structured Query Language* ou Linguagem de Consulta Estruturada). Em função disso os sistemas desta classe também são chamados de sistemas SQL. Após a chegada da Internet, do lançamento das lojas de aplicativos e dos mercados digitais, nasceram as classes NoSQL e NewSQL, também conhecidos como banco de dados não relacional e banco de dados novo relacional, respectivamente (FATIMA; WASNIK, 2017).

Fatima e Wasnik (2017), também explicam a diferença entre as classes de banco de dados, lembrando que o SQL tem um esquema fixo e armazena os dados no formato tradicional de linha e coluna em uma tabela, lidando com a simultaneidade por meio de transações para garantir a consistência dos dados. Embora este modelo tenha algumas restrições quando aplicados a sistemas de alta demanda, é um modelo padrão eficiente e confiável. Contudo, devido a estas restrições, surgiu a classe NoSQL que é livre de esquemas e possui um projeto mais flexível, podendo armazenar uma variedade de dados, sendo mais apropriado para os sistemas IoT. Já o NewSQL é uma plataforma que une características do SQL e do NoSQL, mantendo a estrutura relacional e incorporando soluções desenvolvidas por sistemas NoSQL.

O Firebase® dispõe de dois sistemas de banco de dados NoSQL, o Realtime Database® e seu sucessor, o Cloud Firestore® que atualmente encontra-se em fase de desenvolvimento e testes. Estas ferramentas não são gratuitas, mas dependendo do volume de dados e transações realizadas disponibilizam parte de seus recursos de forma gratuita. Por exemplo, em projetos com até 100 conexões e no máximo 1Gb de armazenamento, é possível utilizar as ferramentas de forma gratuita (DEVELOPERS, 2018a).

Além de detalhar os modelos de banco de dados existentes, Fatima e Wasnik (2017) realizaram uma comparação entre as classes SQL, NoSQL e NewSQL, avaliando seu desempenho. De acordo com seus resultados os bancos de dados NoSQL e NewSQL são mais indicados para trabalhar com sistemas voltados à Internet das Coisas, principalmente quando envolvem grande volume de dados. Sendo assim, as ferramentas oferecidas pelo Firebase® são consideradas adequadas e selecionadas para o *framework* proposto.

O Cloud Firestore® é uma evolução do Realtime Database® provendo melhores resultados através de consultas mais rápidas e avançadas com melhor escalabilidade. É um banco de dados flexível e escalável para desenvolvimento de dispositivos móveis, Web e servidores a partir do Firebase®. Ele mantém seus dados em sincronia com aplicativos clientes em tempo real e, além disso, oferece suporte *off-line* para dispositivos móveis e Web para que você possa criar aplicativos responsivos que funcionem independentemente da latência da rede ou da conectividade com a Internet (DEVELOPERS, 2018a).

3.4 GEOLOCALIZAÇÃO

As aplicações que fazem uso de geolocalização consistem em detectar uma determinada posição geográfica e apresentá-la em um mapa ou visão de satélite, criando um relacionamento com lugares e destinos. Dispositivos com GPS executam essa função, determinando a posição do aparelho e disponibilizando em tempo real os dados de longitude e latitude (CADAVIECO, 2013).

No contexto da Internet das Coisas, um grande número de objetos e sensores são interconectados com o objetivo de determinar localização de algo, ou do usuário. Com mais pontos de referência, a localização de um objeto pode ser melhor determinada, uma vez que mais informações geográficas podem ser extraídas dos dados coletados (YING; PAHLAVAN; LI, 2017).

Sistemas baseados em geolocalização estão disponíveis em carros, barcos, aviões, *tablets*, *smartphones*, *notebooks* e até mesmo em *smartwatches* e não são utilizados somente para apresentar a posição geográfica em um mapa. As aplicações podem utilizar os dados de geolocalização para melhorar a compreensão do contexto em que o usuário está inserido, conectando-o com o ambiente e os objetos, bem como, coletando dados que permitem a evolução para a computação ubíqua (CADAVIECO, 2013).

A empresa Google®, oferece uma gama de serviços relacionados à geolocalização que contemplam a Google Maps API®. Estes serviços oferecem suporte a *interface* de mapas para Android®, iOS® e web, bem como, uma série de serviços para consulta de informações relacionadas a geolocalização, disponíveis por meio de Web Services, sendo um deles o serviço de processamento para geocodificação (DEVELOPERS, 2017a).

Os serviços de geocodificação são geralmente fornecidos por mecanismos de pesquisa na web e a maioria deles oferece uma API, que pode ser incorporada em aplicativos móveis por meio de SDKs e consultas HTTP (*Hypertext Transfer Protocol*). O uso é geralmente fácil e gratuito (com limites de solicitações), o que contribui para sua popularidade. A qualidade da geocodificação é variável, dependendo do tipo e da qualidade dos dados informados para que seja realizado o correto processamento (DAVID et al., 2016).

O Google Maps Geocoding API®, é um serviço que oferece geocodificação e geocodificação reversa de endereços (DEVELOPERS, 2015), possibilitando a conversão de um endereço em coordenada geográfica e também o processo reverso, a construção de um endereço a partir das coordenadas geográficas.

As ferramentas disponibilizadas pela Google Maps API®, fazem parte do *framework* proposto, trabalhando juntas para decodificarem uma determinada localização ou permitindo que a aplicação busque informações detalhadas sobre locais a partir da localização do usuário.

Assim como outras ferramentas do Google®, a Google Maps API® é disponibilizada gratuitamente com alguns limites de uso/consumo. Estes limites são suficientes para a maioria dos projetos, principalmente aqueles com cunho acadêmico que, por exemplo, dificilmente ultrapassarão o limite de 2.500 solicitações diárias.

3.5 NOTIFICAÇÕES (*PUSH NOTIFICATIONS*)

Os serviços de notificação por *push* ou *push notifications*, como são conhecidos, permitem o envio e recebimento de mensagens de forma confiável, eficiente e eficaz. Este método de comunicação é frequentemente utilizado em dispositivos móveis que usufruem dos serviços de notificação integrados aos sistemas operacionais. O Google Cloud Messaging® (GCM), por exemplo, é incorporado ao sistema Android®, podendo também ser incorporado em aplicativos para iOS® (KOLLMANN; BERESFORD, 2017).

Estes serviços são amplamente utilizados visando oferecer suporte à comunicação entre dois dispositivos, como por exemplo, o envio e recebimento de mensagens entre usuários, além de permitir a disseminação de informações em aplicativos de notícias, esportes e principalmente em redes sociais.

O Firebase Cloud Messaging® (FCM) é mais uma das soluções oferecidas pelo Firebase® e permite, de forma gratuita, enviar mensagens entre plataformas. Utilizando o FCM, a aplicação pode notificar um App cliente sobre informações coletadas de um determinado sensor, chamando a atenção do usuário para uma determinada ação. As notificações também podem ser enviadas para promover novas interações e a retenção de usuários (FIREBASE, 2017).

O FCM é na verdade a evolução do GCM e possui SDK para integração com as plataformas mais comuns (Android®, iOS® e Web), o que permite um aproveitamento para as aplicações voltadas à Internet das Coisas, que geralmente então presentes em todas estas plataformas.

As notificações por *push* habilitam que os desenvolvedores e administradores de aplicativos se comuniquem com os usuários mesmo que seu aparelho esteja desligado, pois o serviço é capaz de armazenar a mensagem e entregá-la quando o dispositivo acordar. As interações entre cliente e servidor são todas realizadas sobre uma única conexão TCP de longa duração, amenizando problemas de rede e de sobrecarga do aparelho. Outra vantagem é que o algoritmo que gerencia as *push notifications* pode agrupar diversas notificações, destinadas a diferentes aplicações de um mesmo *smartphone*, em uma única mensagem, otimizando o desempenho, reduzindo processamento e aumentando a vida útil da bateria (KOLLMANN; BERESFORD, 2017).

O Firebase® dispõe de uma ferramenta *web* para criar e disparar as *push notifications*. Esta ferramenta permite executar as funções básicas do FCM e é uma solução que permite acelerar o desenvolvimento e os testes. Contudo, deve ser considerado o

desenvolvimento de uma ferramenta extra que permita o gerenciamento e envio automatizado de notificações.

3.6 COLETA E ANÁLISE DE DADOS (*ANALYTICS*)

A medida que a tecnologia IoT avança, seu valor pode ser verificado analisando os dados gerados por “coisas” conectadas. O termo *Analytics* sugere a coleta e análise dos dados de uso, referente a uma determinada aplicação. A intersecção da Internet das Coisas com as ferramentas de coleta e análise de dados, evoluem para a *IoT Analytics*, considerada um dos propulsores para a quarta revolução industrial (KANG; YU; GOLEN, 2017).

Segundo Patel, Intizar Ali e Sheth (2017), a próxima onda de inovação da Internet das Coisas será impulsionada pela análise de dados. Os sensores enviam seus dados pelas redes de comunicação para plataformas baseadas em nuvem e aplicativos móveis, onde todo o processamento é realizado, permitindo as tomadas de decisões. Apesar desta lógica reduzir os custos de implementação e manutenção, visto o fato de a lógica de aplicação estar centralizada, as soluções baseadas em nuvem possuem limitações de banda, custos e alta latência.

O Google Analytics Firebase® é uma solução de análise ilimitada e gratuita, que está no núcleo do Firebase®, para avaliação de aplicativos. Nele são fornecidos *insights* sobre uso de aplicativos e engajamento do usuário. Com os relatórios do Google Analytics Firebase®, o gerenciador procura entender o comportamento dos usuários e pode tomar decisões mais bem informadas sobre quais áreas da aplicação o usuário tem mais interesse, quais são os produtos mais acessados, onde estão os pontos de falha e lentidão do sistema, quais são os períodos com picos de acesso, dentre outras informações (LASAOSA, 2016).

É muito importante coletar os dados de uso dos usuários e os dados lidos pelos sensores IoT, mas principalmente, é importante reconhecer em qual momento o dado está sendo coletado e qual análise está sendo aplicada, objetivando gerar reais *insights* do ecossistema da IoT (KANG; YU; GOLEN, 2017)..

Enquanto o usuário está utilizando o aplicativo, o SDK do Firebase® captura automaticamente uma série de eventos e propriedades do usuário, separados por tipo de evento, categoria, etc. Os dados capturados ficam disponíveis em um painel web (Figura 14) que pode ser acessado pelo gerenciador do aplicativo. O painel sumariza os dados coletados, desde dados resumidos como usuários ativos e informações

demográficas, até um detalhamento desses dados como a identificação das áreas mais acessadas da aplicação, ou o horário em que os usuários mais acessam o aplicativo.

Figura 14 – Exemplo de painel do Firebase Analytics®



Fonte: Firebase Analytics®

3.7 REQUISITOS

Com base na revisão bibliográfica exposta no capítulo 2 foi possível conhecer as necessidades dos usuários, as necessidades da comunidade de desenvolvedores, bem como as necessidades do próprio projeto, possibilitando a definição das tecnologias e funcionalidades de *software*, ou seja, os requisitos do sistema.

Segundo Sommerville (2007), requisitos de sistema são descrições dos serviços fornecidos pelo sistema, assim como, as suas restrições operacionais. Esses requisitos refletem as necessidades dos clientes de um sistema que ajudam a resolver algum problema.

Os requisitos de um sistema são frequentemente classificados em requisitos funcionais e não funcionais. Requisitos funcionais são declarações dos serviços que o sistema deve oferecer, como o sistema deve reagir a entradas específicas, e como o sistema deve se comportar em determinadas situações. Requisitos não funcionais são especificações que indicam como o sistema deve se comportar em relação ao desempenho, plataforma, interface, dentre outros atributos e/ou restrições de projeto (SOMMERVILLE, 2007).

3.7.1 Requisitos Funcionais

Conforme mencionado, os requisitos funcionais representam as necessidades do sistema. Nesta etapa, discutem-se quais funcionalidades

devem ser desenvolvidas para satisfazer as necessidades dos usuários, da comunidade de desenvolvedores e do *framework*, sendo:

- Realizar o *login* de forma simplificada e automatizada;
- Permitir a persistência de dados de forma simplificada e automatizada (banco de dados);
- Permitir a consulta de dados relevantes com base na localização do usuário (*geodata*);
- Permitir o envio e recebimento de notificações *push*;
- Permitir a análise de dados de uso da aplicação (*analytics*).

3.7.2 Requisitos não funcionais

Os requisitos não funcionais são aqueles que não dizem respeito diretamente às funcionalidades fornecidas pelo sistema. Podem estar relacionados a propriedades de sistemas emergentes, como confiabilidade, tempo de resposta, espaço em disco, desempenho e outros atributos de qualidade do produto, entre eles:

- Serviços para gerenciamento de *login*:
 - Facebook Account Kit®;
 - Firebase Authentication®;
- Serviço de banco de dados:
 - Firebase Realtime Database®;
- Serviço para coleta de dados baseados em geolocalização:
 - Google Maps API®;
- Serviço para envio e recebimento de notificações:
 - Firebase Cloud Messaging®;
- Serviço para coleta e análise de dados (*analytics*):
 - Firebase Analytics®.

3.8 ESPECIFICAÇÃO DO *FRAMEWORK*

Nesta seção são apresentadas as classes do *Framework* IoT e alguns passos para a implementação do mesmo. Para melhor expor a documentação de *software*, alguns dos diagramas foram construídos com base na linguagem UML (*Unified Modeling Language*).

Introduzida na segunda metade da década de 1990, a linguagem UML foi uma evolução da programação orientada a objetos. Suas origens estão na interseção de três metodologias populares nos anos 80 e início dos anos 90: o método de Booch, a técnica de modelagem de

objetos de Rumbaugh e a engenharia de *software* orientada a objetos de Jacobson (CICCOZZI; MALAVOLTA; SELIC, 2018).

Ciccozzi, Malavolta, Selic (2018) concluíram em sua revisão sistemática que há um crescente interesse científico na modelagem UML e consideram a linguagem adequada para documentação de *software*.

O conceito elaborado para o *framework* IoT é multiplataforma e pode ser aplicado em plataformas *mobile* e *web*. Cada plataforma é configurada de forma singular de acordo com suas características e particularidades, em cada um dos serviços utilizados pelo *framework*.

Para permitir a correta documentação de *software* também é necessário definir um paradigma de programação. Sendo assim, foi estipulado o paradigma de programação orientada a objeto.

As interfaces, na programação orientada a objetos, atuam como intermediadoras entre a classe de um determinado objeto e a aplicação principal. De modo geral, a interface define funcionalidades que são obrigatoriamente implementadas na classe (CAMARGO, 2010).

No *framework* IoT, as funcionalidades estarão definidas em uma *interface* principal que por sua vez, fará a ligação entre uma classe principal que implementa as funcionalidades e a aplicação principal, desenvolvida pelo programador.

As seções a seguir apresentam a documentação das funcionalidades do *framework*, ilustradas por meio da linguagem UML, com base no paradigma de programação orientada a objeto.

3.8.1 Instalação e inicialização do framework

Para o correto funcionamento do *framework* é necessário incluir no ambiente de desenvolvimento, todas as bibliotecas utilizadas, informar ao sistema operacional quanto aos recursos utilizados e por fim, inicializar a classe principal da ferramenta, que atuará intermediando à comunicação entre os recursos do *framework* e a aplicação principal.

O desenvolvedor que optar pela utilização do *framework* proposto, deverá acessar a *dashboard* de cada uma das ferramentas que estão embutidas no *framework* (Facebook Account Kit®, Firebase® e Google Maps API®), criando um projeto e realizando a configuração do mesmo com sua aplicação.

As bibliotecas possuem uma série de classes e métodos que permitem a correta execução de cada uma das funcionalidades do *framework*. Além da biblioteca que contém as classes pertencentes ao

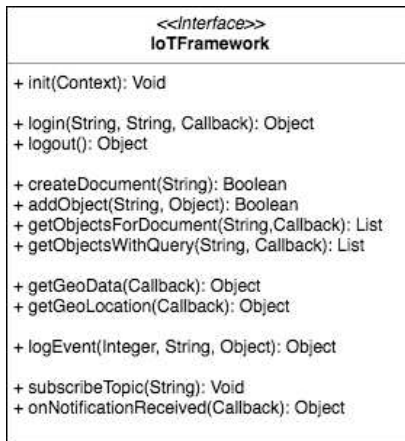
ferramental proposto, é necessário incluir as bibliotecas do Firebase®, Google Maps API® e Facebook Account Kit®. A lista completa com as bibliotecas está disponível no APÊNDICE A.

O segundo passo é declarar os recursos utilizados para que os sistemas operacionais permitam o acesso a estes. Esta prática é necessária em todas as aplicações que fazem uso de recursos de *hardware* ou coletam informações pessoais, por motivos de segurança ao usuário. Ao declarar a necessidade de um determinado recurso, o usuário é notificado que a aplicação faz uso do mesmo. Desta forma, o usuário pode permitir/negar aquilo que lhe convém e não corre o risco de ser monitorado sem seu devido consentimento.

Em geral todos os recursos utilizados pelo *framework* precisam ser declarados, exceto na plataforma web, em que recursos mais comuns como, por exemplo, o acesso à Internet, não precisa ser declarado. A lista completa de recursos a serem declarados está descrita no APÊNDICE A.

Com as bibliotecas importadas e os recursos declarados, o *framework* torna-se disponível para ser utilizado. Para acessar os métodos do *framework* é necessário que uma das classes da aplicação implemente a *interface IoTFramework* (Figura 15). Assim que o usuário inicia a aplicação, o método de inicialização pode ser acionado, disparando a sequência de eventos representada pelo diagrama de sequência UML da Figura 16.

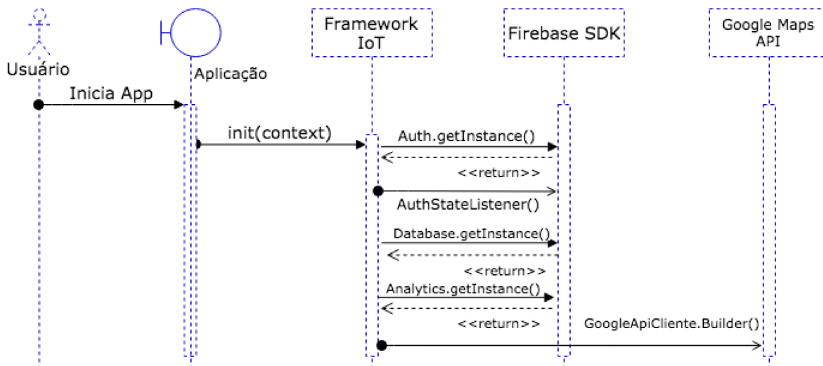
Figura 15 – Diagrama de interface para a inicialização do framework



Fonte: Autor

No método de inicialização, o *framework* se comunica com o SDK do Firebase® para recuperar a instância de objetos que são responsáveis por manter os recursos de autenticação, banco de dados, notificações e *analytics* em memória. Também declara o método que deverá ser disparado quando ocorrerem mudanças no estado de *login* (do usuário) e inicializa o SDK do GoogleMapsAPI®.

Figura 16 – Diagrama de sequência de inicialização do *framework*



Fonte: Autor

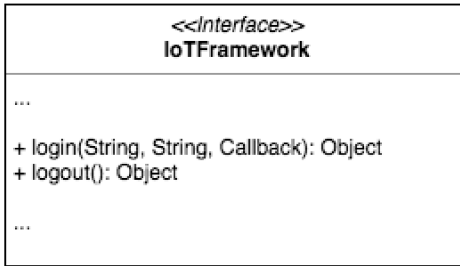
Ao finalizar a sequência acima o *framework* se torna disponível para ler/gravar dados no Realtime Database®, incluir eventos no Firebase Analytics®, receber notificações e consumir dados relacionados à Geolocalização®. Nesta etapa, o mais indicado é iniciar o fluxo de *login*.

3.8.2 Autenticação (*Login*)

Retardar a exigência de *login*, quando possível, é uma boa prática quando se trata de aumentar o engajamento dos usuários, permitindo que eles tenham uma prévia das funcionalidades do aplicativo antes que seja exigida a autenticação. Contudo, para utilizar os recursos disponibilizados pelo *framework* é necessário exigir a autenticação do usuário.

Os métodos relacionados na Figura 17 são parte da *interface* principal do *framework* e representam as possibilidades de autenticação disponibilizadas pelo *framework*. Todas as interações com o Facebook Account Kit® e o Firebase Autenticação® são realizadas pela estrutura desenvolvida, ou seja, o desenvolvedor precisa apenas disparar os métodos de *login* ou *logout*.

Figura 17 – Diagrama de interface para o serviço de autenticação

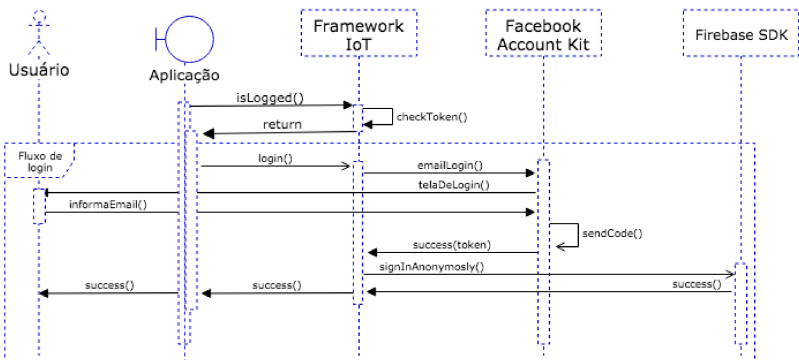


Fonte: Autor

Sempre que uma sessão de *login* é iniciada o Facebook Account Kit® fornece um *token* de acesso que é utilizado para autenticar a troca de informações com sua API. Este *token* também pode ser utilizado para validar outras transações na aplicação, como por exemplo, a troca de dados entre um servidor na nuvem. Neste caso, o *token* de acesso poderá ser utilizado para autenticar a comunicação entre o aplicativo (lado cliente) com os servidores do Firebase®.

Os SDKs do Facebook Account Kit® e Firebase® permitem que o token de acesso seja persistido na memória local, seja nos dados de aplicativos (Android® e iOS®) ou nos *cookies* do navegador (*web*). Isto facilita o processo de autenticação do ponto de vista do usuário, não sendo necessário realizar o *login* toda vez que a pessoa for utilizar o aplicativo.

Figura 18 – Diagrama de seqüência para o fluxo de *login*



Fonte: Autor

A seqüência de login representada na Figura 18 é iniciada pela conferência do *token* de autenticação, se o mesmo não existir ou estiver

sem validade (expirado) é iniciado o fluxo de *login* por *e-mail*, via Facebook Account Kit®. A partir disso seleciona-se o *e-mail* do usuário para qual é enviado um link/código de ativação. Ao selecionar o *link* recebido por *e-mail* o usuário é redirecionado para a aplicação, concluindo o fluxo de *login* e entregando ao *framework* o *token* de acesso.

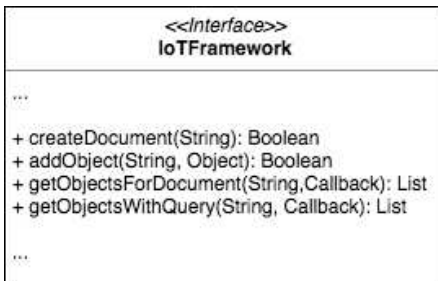
Por último, deverá ser realizada a autenticação simplificada no Firebase®, garantindo o correto funcionamento dos outros serviços presentes no *framework*, como o banco de dados, *push notifications* e *analytics*. Na conclusão da sequência de *login* o método **success()** é disparado na aplicação possibilitando ao usuário receber uma mensagem de saudação, informando que o mesmo foi autenticado com sucesso no sistema.

3.8.3 Banco de Dados

A persistência de dados está presente na maioria dos aplicativos disponíveis no mercado, salvando preferências do usuário e informações relevantes para as regras de negócio do sistema. Os métodos apresentados na Figura 19 relacionam as opções para interação com o banco de dados, disponibilizadas pelo *framework* IoT proposto.

O *framework* interage com as ferramentas para persistência de dados do Firebase®, que são baseadas em sistemas gerenciadores de banco de dados não relacional (*NoSQL*). Este tipo de banco de dados não exige a criação de tabelas para armazenamento dos dados, facilitando a automatização das operações de leitura e escrita.

Figura 19 – Diagrama da interface para manipulação de banco de dados

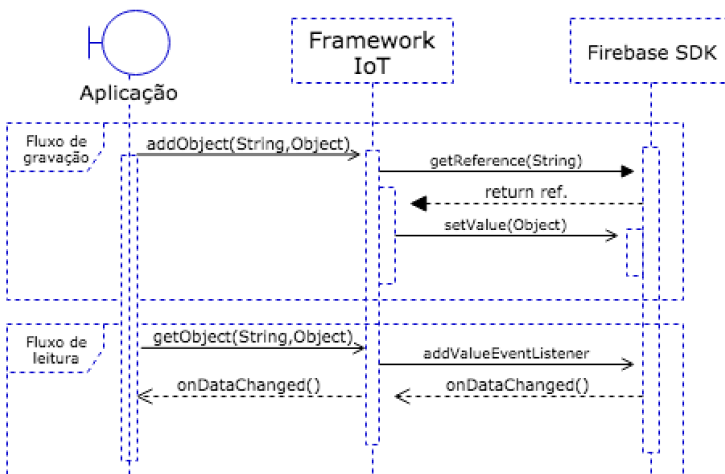


Fonte: Autor

Para organizar os dados e recuperá-los corretamente, o desenvolvedor deve criar um documento, também conhecido como

página, onde são armazenados os objetos (dados). Ao realizar a gravação, basta que o desenvolvedor envie por parâmetro o nome do documento onde o objeto deverá ser armazenado, disparando o fluxo de gravação representado na Figura 20.

Figura 20 – Diagrama de sequência sobre o fluxo de leitura e gravação



Fonte: Autor

A Figura 20 também expõe o fluxo de leitura que representa o acesso ao banco de dados do Firebase® realizado pelo *framework*, retornando a lista completa de objetos persistida em um determinado documento. Caso o desenvolvedor deseje filtrar os dados retornados pela busca, ou seja, aplicar consulta no banco de dados, deverá ser utilizado o método **`getObjectsWithQuery()`**.

Os serviços Realtime Database® e Cloud Firestore®, disponibilizados pelo Firebase®, também dispõem de ferramentas para ordenação e agrupamento dos dados, permitindo entregá-los para a aplicação de forma adequada e sem onerar recursos de processamento no dispositivo do usuário final. Estes recursos não estão incluídos nas funcionalidades do *framework* IoT, sendo sua implantação uma das sugestões para trabalhos futuros.

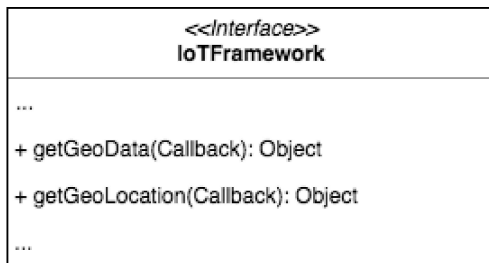
3.8.4 Geolocalização

Em geral, a maioria dos aplicativos consultam a localização do usuário para obter informações relacionadas à posição geográfica do

mesmo, seja para sugerir serviços, determinar ações dentro do sistema, produzir publicidade direcionada, entre outros.

Não é o foco do *framework* disponibilizar uma interface gráfica, exibindo a localização do usuário em um mapa ou realizar um monitoramento em tempo real da geolocalização do dispositivo.

Figura 21 – Diagrama de interface para o serviço de geolocalização

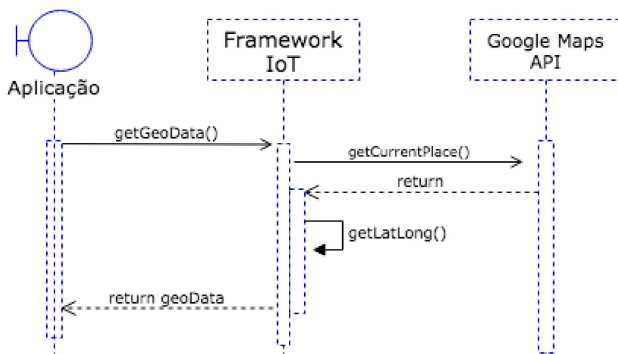


Fonte: Autor

Conforme representado na Figura 21, o *framework* IoT permite que o desenvolvedor requisite facilmente os dados de geolocalização, sendo que o método **getGeoLocation()** retorna à posição geográfica do dispositivo (latitude e longitude), enquanto que o método **getGeoData()** busca por informações detalhadas, com base na localização do dispositivo, nas bases e API's do Google Maps®.

O fluxo disparado pelo método **getGeoData()** é apresentado na Figura 22:

Figura 22 – Diagrama de sequência para o fluxo de coleta de dados com base na localização



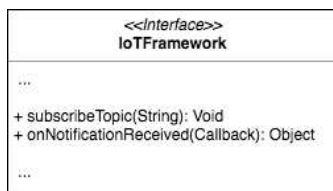
Fonte: Autor

3.8.5 Notificações (*Push Notifications*)

Os serviços de notificação por *push* ou *push notifications*, são utilizados para permitir a comunicação entre dois dispositivos, como a troca de mensagens entre usuários. Também serve como ferramenta de engajamento, permitindo que os administradores enviem mensagens aos usuários, no intuito de promover o retorno do mesmo ao aplicativo.

Conforme demonstrado na Figura 23, os métodos disponibilizados pelo *framework* IoT permitem apenas o recebimento de notificações. Estas, deverão ser enviadas por meio de um console (página *web*) disponibilizado pelo Firebase®. O envio de mensagens entre usuários poderá ser implementado em trabalhos futuros.

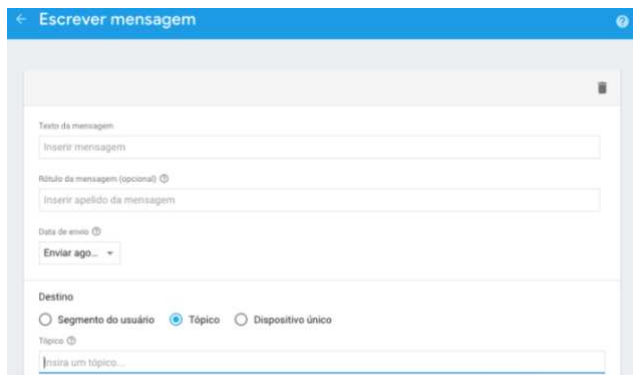
Figura 23 – Diagrama de interface para o serviço de *push notification*.



Fonte: Autor

Para receber uma notificação qualquer, o desenvolvedor deverá inscrever o dispositivo em um determinado tópico, disparando o método **subscribeTopic()**. Ao acessar o console do Firebase® o administrador poderá disparar notificações aos usuários, selecionando os tópicos que receberão esta notificação (Figura 24).

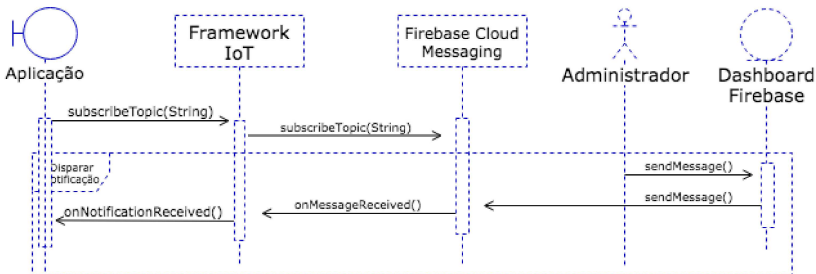
Figura 24 – Console do Firebase Cloud Message®



Fonte: Firebase Cloud Message®

Na Figura 25 o fluxo necessário para que uma notificação seja disparada na aplicação. O desenvolvedor pode inscrever o dispositivo em diversos tópicos diferentes, de acordo com as preferências do usuário, envolvendo, plataforma, versão do aplicativo, sistema operacional, entre outras.

Figura 25 – Diagrama de sequência para o fluxo *push notification*



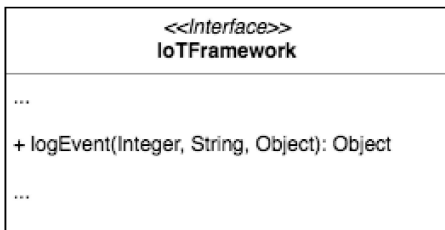
Fonte: Autor

Assim que o usuário estiver inscrito em um determinado tópico o administrador do sistema poderá selecioná-lo no console do Firebase® e disparar uma notificação. Quando o dispositivo recebe a mensagem, o método **onMessageReceived()** é acionado, permitindo que o desenvolvedor apresente uma saudação ao usuário, disparando o método **onNotificationReceived()**.

3.8.6 Analytics

A implementação comum de ferramentas analíticas envolve uma série de desenvolvimentos em diversas plataformas, necessitando de um banco de dados, do registro de eventos e de uma ferramenta gráfica para analisar as informações coletadas.

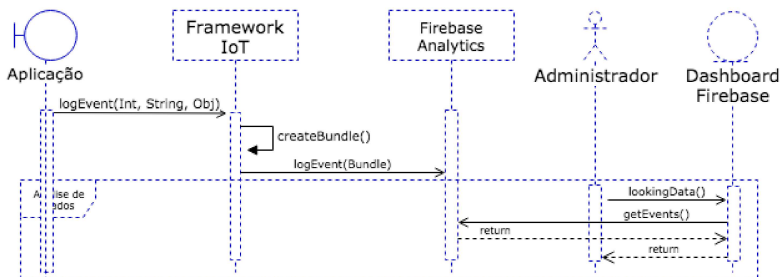
Figura 26 – Diagrama de interface sobre o serviço de *analytics*



Fonte: Autor

O *framework* IoT proposto simplifica a implantação de uma ferramenta/aplicação com este objetivo, ficando a cargo do desenvolvedor apenas a invocação do método **logEvent()** (Figura 26) quando desejar registrar um evento.

Figura 27 – Diagrama de sequência do fluxo *analytics*



Fonte: Autor

O processo para gerar um novo evento, representado na Figura 27, é automatizado pelo Firebase Analytics®, que dispõe de um painel *web* para visualizar todos os dados coletados.

Além dos dados registrados manualmente via implementação do desenvolvedor, o Firebase Analytics® registra automaticamente uma série de eventos relevantes conforme os usuários utilizam o aplicativo. Por exemplo, dados relacionados a região em que o aplicativo está sendo utilizado (Figura 28), dados relacionados as versões do aplicativo (Figura 29) e na Figura 30, dados sumarizados informando a quantidade de usuários ativos (usuários que efetivamente usam a aplicação). Estes dados são coletados automaticamente sem a necessidade de nenhuma implementação adicional.

Figura 28 – Console do Firebase Analytics® - acesso público



Fonte: Firebase Analytics®

Figura 29 – Console do Firebase Analytics® - aceitação



Fonte: Firebase Analytics®

Figura 30 - Console do Firebase Analytics® - usuários ativos



Fonte: Firebase Analytics®

4 IMPLEMENTAÇÃO E AVALIAÇÃO

Neste capítulo, é descrito o processo de implementação do protótipo do *framework* IoT proposto e a análise dos resultados obtidos com o questionário de avaliação.

4.1 DESENVOLVIMENTO DE *SOFTWARE*

O *framework* proposto neste trabalho tem como objetivo geral auxiliar e facilitar o desenvolvimento de sistemas voltados para a Internet das Coisas. Conforme apresentado nesta dissertação, a IoT abrange diversas áreas e ambientes, conectadas por meio de diferentes estruturas de comunicação e plataformas.

As ferramentas estudadas para incorporação no *framework* proposto são compatíveis com as plataformas *web* e *mobile* (Android®/iOs®/), podendo também operar com sistemas embarcados e dispositivos IoT que suportem as plataformas supracitadas.

Para a correta construção do *Framework* IoT, é necessário que o mesmo receba uma implementação específica em cada ambiente, considerando as linguagens de programação, ferramentas e APIs de cada plataforma. Todavia, com o intuito de otimizar o desenvolvimento e avaliação da estrutura proposta, este trabalho irá considerar o desenvolvimento de um protótipo do *framework* em uma única plataforma.

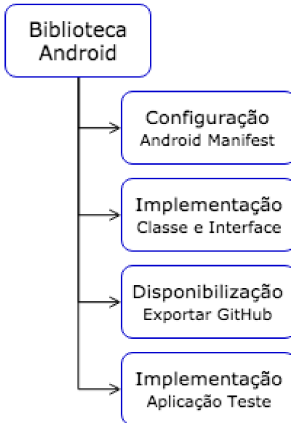
O protótipo foi implementado na plataforma Android®, cuja escolha ocorreu por conta da linguagem de programação (JAVA® ou Kotlin®), ambiente de desenvolvimento gratuito, extensa documentação das APIs, além da afinidade do autor com a plataforma. Para iniciar o desenvolvimento de aplicações voltadas para a plataforma Android®, basta instalar o Android Studio® em um computador comum (DEVELOPERS, 2018b).

A partir da definição da plataforma, as etapas de desenvolvimento, ilustradas na Figura 31, são:

- Estruturar a biblioteca Android®, onde ficará a interface e a classe principal do *framework*, com os métodos e rotinas desenvolvidas;
- Configurar o arquivo de manifesto e importar as APIs dos recursos utilizados no *framework*, na biblioteca Android®;
- Implementar na biblioteca Android®, a interface e classe principal, utilizando a linguagem JAVA;

- Exportar e disponibilizar o protótipo;
- Implementar um aplicativo Android® para teste das funcionalidades do protótipo.

Figura 31 – Etapas de desenvolvimento do framework na plataforma Android®



Fonte: Autor

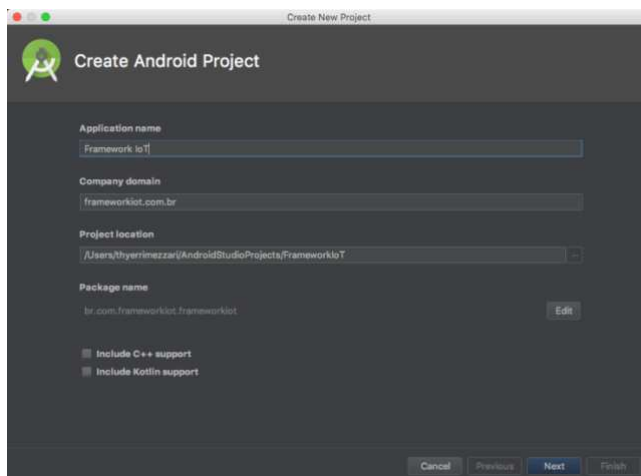
4.1.1 Biblioteca Android®

Para que o *framework* possa ser utilizado por aplicações Android®, é necessário que o mesmo seja construído no formato de uma biblioteca. Do ponto de vista estrutural, uma biblioteca Android® funciona como um aplicativo comum, podendo conter todos os elementos que uma App completa possui, incluindo código-fonte, imagens, mídias e outros arquivos de recursos, bem como, um arquivo de manifesto contendo especificações e configurações pertinentes a biblioteca.

Gerar uma biblioteca é útil ao trabalhar com diversos aplicativos que se utilizam de arquivos e componentes em comum, como classes e arquivos de *layout*. A biblioteca Android® é compilada em um arquivo denominado “*Android Archive*”, que pode ser incorporado como dependência em projetos de aplicativos (DEVELOPERS, 2018b).

Para criar uma biblioteca Android®, é necessário criar um projeto no Android Studio® (Figura 32) e alterar um dos arquivos de configuração, para que o compilador compreenda que este projeto se trata de uma biblioteca e não de um aplicativo Android® comum.

Figura 32 – Criando o projeto do framework IoT no Android Studio®



Fonte: Autor

O arquivo de configuração supracitado é o “build.gradle”, localizado na pasta App (frameworkiot/app/build.gradle). Este arquivo contém uma série de definições do projeto, incluindo a versão da biblioteca (para controlar as alterações e revisões), a versão mínima do Android® que a biblioteca oferecerá suporte, as dependências do projeto (bibliotecas de terceiros utilizadas nesse projeto, como as bibliotecas do Firebase®, Google Maps API®, etc), além de diversas outras configurações.

Com as configurações concluídas o Android Studio® irá solicitar a sincronização do projeto, para que ele possa importar todas as dependências, considerando as configurações realizadas no arquivo “build.gradle”, e compilar a biblioteca em um arquivo “Android Archive”.

4.1.2 Arquivo de manifesto Android®

Toda biblioteca ou aplicativo Android® precisa, obrigatoriamente, possuir um arquivo de manifesto, denominado “*Android Manifest*”. Ele fica no diretório raiz do projeto e documenta informações essenciais sobre o aplicativo, permitindo que o sistema operacional execute corretamente o código da App.

É no arquivo de manifesto que são declarados os recursos de *hardware* e *software* que o aplicativo irá utilizar, as diversas telas da aplicação, permissões para lidar com partes protegidas da API, dentre outros parâmetros que permitem a comunicação com outros aplicativos e recursos do sistema operacional (DEVELOPERS, 2017b).

O arquivo *Android Manifest* é criado pelo Android Studio® automaticamente durante a inicialização do projeto, contendo as configurações mínimas para que um aplicativo, ou biblioteca, possa ser compilado(a). Estas configurações complementam algumas das informações declaradas no arquivo “build.gradle”, adicionando dados quanto ao tema (*design* da App), quanto ao nome da aplicação, além de especificações referentes a cada uma das telas da aplicação.

Na biblioteca do *framework* IoT, é necessário incluir no arquivo de manifesto algumas permissões (Figura 33), informando ao sistema operacional que o sistema proposto fará uso de recursos de hardware acessando à internet e os dados de geolocalização do usuário.

Figura 33 – Permissões do Usuário, no arquivo “*Android Manifest*”

```

5      <uses-permission android:name="android.permission.INTERNET" />
6      <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />

```

Fonte: Autor

É comum que APIs de terceiros, como a Google Maps API® e o Firebase®, façam uso do arquivo de manifesto para declarar informações importantes para o funcionamento da aplicação. Este é o caso da API do Facebook Account Kit®, que exige a inclusão de algumas informações no “*Android Manifest*”.

Figura 34 – Informações pertinentes ao Facebook Account Kit® no arquivo “*Android Manifest*”

```

24      <activity
25          android:name="com.facebook.accountkit.ui.AccountKitActivity" />
26
27      <activity android:name="com.facebook.accountkit.ui.AccountKitEmailRedirectActivity">
28          <intent-filter>
29              <action android:name="android.intent.action.VIEW" />
30              <category android:name="android.intent.category.DEFAULT" />
31              <category android:name="android.intent.category.BROWSABLE" />
32              <data android:scheme="ak1857237500999654" />
33          </intent-filter>
34      </activity>
35
36      <meta-data android:name="com.facebook.accountkit.ApplicationName"
37          android:value="IoTFrameworkDemo" />
38      <meta-data android:name="com.facebook.sdk.ApplicationId"
39          android:value="1857237500999654" />
40      <meta-data android:name="com.facebook.accountkit.ClientToken"
41          android:value="dccc2e56e00867ef2e944943f7cc0caf" />

```

Fonte: Autor

A Figura 34 apresenta um trecho do arquivo de manifesto, com os dados requeridos pela API do Facebook Account Kit®. As informações incluídas entre as tag's *activity* implementam configurações pertinentes as telas de *login* utilizadas por esta API. Já as informações incluídas entre as tag's *meta-data*, são configurações relacionadas ao registro da aplicação nos servidores do Facebook®.

Após a configuração do arquivo “*Android Manifest*” a biblioteca já está pronta para receber a implementação dos métodos do *framework* IoT.

4.1.3 Interface e Classe principal do *framework*

As *interfaces* JAVA, presentes na plataforma Android®, permitem que sejam declarados uma série de métodos abstratos, garantindo benefícios de herança e polimorfismo, presentes nas linguagens orientadas a objeto, e protegendo contra os riscos relacionados a ambiguidades e falhas de implementação por parte da equipe de desenvolvedores (SIERRA; BATES, 2005).

Para criar uma *interface* no Android Studio®, basta seguir o assistente disponível no menu. Os métodos abstratos, presentes na *interface* do *framework*, foram listados na seção 3.8 deste trabalho, Figura 15. A seguir, a Figura 35 expõe a declaração dos métodos na *interface*:

Figura 35 – Interface principal do *framework* IoT

```
public interface IoTFrameworkInterface {

    void init(Context context);

    Object login(String login, String password);
    Object logout();

    Boolean createDocument(String databaseName);
    Boolean addObject(String databaseName, Object object);
    List getObjectsForDocument(String databaseName, Callback callback);
    List getObjectsWithQuery(String databaseName, String query, Callback callback);

    Object getGeoData(Callback callback);
    Object getGeoLocation(Callback callback);

    Object logEvent(Integer event, String text, Object object);

    void subscribeTopic(String topic);
    Object onNotificationReceived(Callback callback);

}
```

Fonte: Autor

A classe principal do *framework*, a **IoTFrameworkClass** deve implementar a *interface* supracitada. Por sua vez, os métodos declarados na *interface*, são implementados diretamente na classe. É nesta etapa

que são acionadas as rotinas presentes nas APIs do Firebase®, Facebook Account Kit® e Google Maps API®.

Figura 36 – Classe principal do *framework* IoT

```
public class IoTFrameworkClass implements IoTFrameworkInterface
{
    @Override
    public Boolean addObject(String databaseName, Object object) {
        DatabaseReference mDatabase;
        mDatabase = FirebaseDatabase.getInstance().getReference();
        if (mDatabase != null) {
            mDatabase.child(databaseName).setValue(object);
            return true;
        }
        else return false;
    }
}
```

Fonte: Autor

Por exemplo, a implementação do método **addObject()**, que tem por objetivo incluir um novo objeto no banco de dados do Realtime Database®, pode ser conferida na Figura 36. Ao ser invocado o método dispara uma sequência de chamadas na biblioteca do Firebase®, que por sua vez, disparam chamadas de API aos servidores do Firebase® e Realtime Database®.

4.1.4 Disponibilização da biblioteca Android®

Após finalizar a biblioteca Android® e compilá-la em um *Android Archive* é preciso disponibilizá-la em um ambiente público, para que outros desenvolvedores possam acessá-la e utilizá-la em seus projetos.

O ambiente escolhido foi a plataforma GitHub, um dos repositórios de código mais utilizados do mundo, (NIELEK et al., 2016), onde atualmente existem mais de 35 milhões de projetos hospedados (COSENTINO; LUIS; CABOT, 2016).

O GitHub, considerado a maior comunidade de código aberto do mundo, é baseado em um sistema concebido pelo criador do Linux, Linus Torvalds, denominado de Git (BLEIEL, 2016), que objetiva o controle de versões e alterações em códigos-fonte.

Contendo diversos recursos, permite que os desenvolvedores hospedem e compartilhem projetos ou parte do código-fonte. Bibliotecas, ferramentas e recursos desenvolvidos, que podem ser consumidos por diversos desenvolvedores. O principal fator que levou a popularização do GitHub foi sua facilidade para lidar com contribuições

em projetos de código aberto (*open source*), permitindo de forma ágil e fácil que desenvolvedores criem ramificações de projetos e submetam suas modificações para aprovação (BLEIEL, 2016).

Toda a biblioteca Android®, contendo os recursos do *Framework IoT*, bem como a interface **IoTFrameworkInterface** e a classe **IoTFrameworkClass** foi disponibilizada no GitHub, podendo ser acessada via <https://github.com/thyerrimezzari/iotframework>.

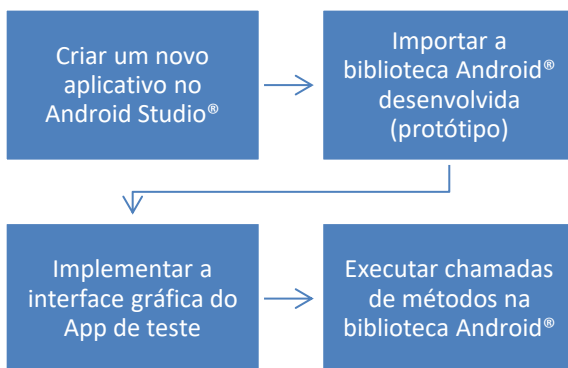
4.1.4.1 Licença

O projeto foi publicado sob os termos da licença MIT, permitindo que outros desenvolvedores possam alterar, sem nenhuma restrição, o código fonte do projeto, desde que o desenvolvedor original seja mencionado, salvaguardando o mesmo de quaisquer responsabilidades ou danos que o software possa ocasionar. A licença também permite que o software e seus derivados possam ser usados para fins comerciais. (GITHUB, 2017). O Anexo A contém a licença no seu formato integral, conforme também publicado no ambiente GitHub.

4.1.5 Aplicativo de teste

Visando avaliar o protótipo do *framework* IoT desenvolvido, foi implementado um simples aplicativo de teste. As etapas de desenvolvimento da aplicação de teste (Figura 37) são, em sua maioria, chamadas aos métodos implementados no protótipo (biblioteca Android®).

Figura 37 – Etapas para desenvolvimento do aplicativo de teste



Fonte: Autor

Após utilizar os assistentes do Android Studio® para criar um novo aplicativo e importar a biblioteca Android® desenvolvida, é necessário criar a interface gráfica do aplicativo de teste (Figura 38). A interface deverá conter um grupo de botões responsáveis por disparar chamadas de métodos no protótipo do *framework* IoT, validando seu funcionamento.

Figura 38 – Interface gráfica do aplicativo de teste



Fonte: Autor

Quando um botão é pressionado, na interface do aplicativo de testes, um método **onClick()** é disparado na classe responsável por tratar os eventos ocorridos neste tela da aplicação. Neste momento, é possível efetuar a chamada para os métodos da classe **IoTFrameworkClass**, que por sua vez irá se comunicar com os serviços correspondentes. A Figura 39 representa a ação realizada ao pressionar o botão “Add Object”:

Figura 39 – Método **onClick()** do botão “Add Object”

```

...
buttonAddObject.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        iotFrameworkClass.addObject( databaseName: "teste_database", new Object());
        iotFrameworkClass.logEvent( event: 1, text: "buttonAddObject", object: null);
    }
});
...

```

Fonte: Autor

O código completo do aplicativo de testes, contendo todas as classes e métodos também foi disponibilizado no GitHub, podendo ser acessado via <https://github.com/thyerrimezzari/iotframeworkexample>.

4.2 AVALIAÇÃO E RESULTADOS

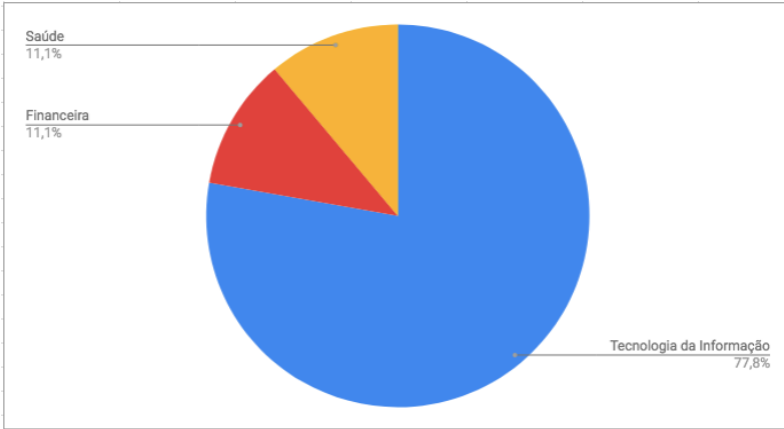
De modo a avaliar as soluções desenvolvidas e implementadas a partir do protótipo, foram convidados cerca de vinte desenvolvedores, cientistas da computação, pesquisadores e entusiastas, com experiência na área de Internet das Coisas, para analisar o *framework* de IoT proposto e responder um questionário de avaliação do mesmo.

Com base em suas experiências e conhecimentos os desenvolvedores avaliaram o código fonte do protótipo *framework* IoT, disponibilizado na plataforma GitHub, juntamente com a aplicação de teste, além dos os diagramas UML, contendo toda a documentação do *framework* proposto.

Do total de convidados, nove se auto consideraram aptos para responder um questionário contendo dez perguntas, a respeito da proposição realizada neste trabalho. Os convidados que não se sentiram confortáveis em responder as perguntas, alegaram não possuir conhecimento específico em alguma das áreas das perguntas. O questionário está disponível na íntegra no Apêndice B.

Dos convidados que aceitaram participar da pesquisa, quatro são graduados, um possui doutorado, outro uma especialização e outros três ainda cursam o nível superior. A maioria (77,8%), são desenvolvedores e analistas de sistemas atuando na área da Tecnologia da Informação. Por fim, um (11,1%) participante atua na área da saúde e outro (11,1%) na área financeira, conforme apresenta a Figura 40.

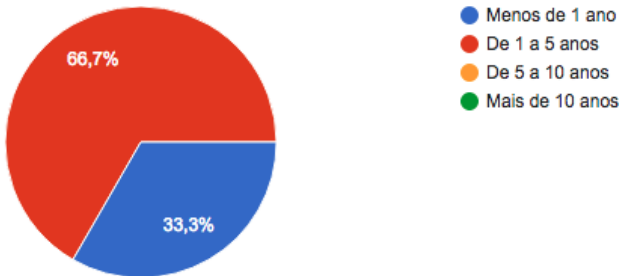
Figura 40 – Área de atuação dos entrevistados



Fonte: Autor

Os especialistas que aceitaram o convite para responder ao questionário, em sua maioria (66,7%), atuam de um a cinco anos em projetos envolvendo Internet das Coisas. Destes, três dos entrevistados (33,3%) iniciaram estudos recentes em trabalhos relacionados à IoT (Figura 41).

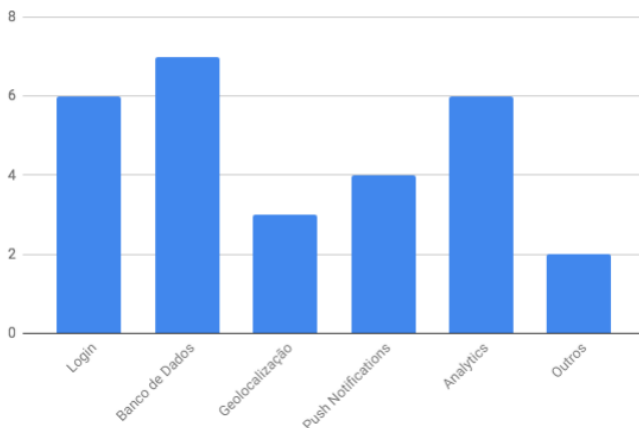
Figura 41 – Tempo de atuação em IoT



Fonte: Autor

Visando complementar a busca por funcionalidades relevantes que devem estar presentes em um *framework* IoT, os entrevistados foram questionados quanto as funcionalidades presentes nos projetos IoT em que trabalharam (Figura 42).

Figura 42 – Funcionalidades presentes nos projetos de IoT na qual os entrevistados participaram



Fonte: Autor

A maioria dos projetos possui as funcionalidades de Login, Banco de Dados e *Analytics*, enquanto que as funcionalidades de Geolocalização e *Push Notification* não foram tão significativas entre os convidados que responderam ao questionário. Dois dos entrevistados também citaram outras duas funcionalidades importantes:

- Envio / Recebimento de notificações por SMS;
- Automação e disparo de ações baseadas em tempo.

Após as questões iniciais, visando coletar informações quanto aos entrevistados, o questionário apresentou um conjunto de cinco perguntas, indagando sobre as soluções disponibilizadas no *framework* IoT.

Ao serem questionados sobre a solução proposta para a implementação de um fluxo de login, sete dos convidados (77,8%) responderam que a proposta é viável. Um desenvolvedor (11,1%) respondeu que prefere implementar sua solução manualmente e outro (11,1%) sugeriu preparar o *framework* para permitir outras formas de autenticação, desta forma, flexibilizando a solução. Conforme representado na Figura 43, nenhum dos entrevistados se posicionou contra o uso de ferramentas como o Facebook Account Kit® e o Firebase Authentication®.

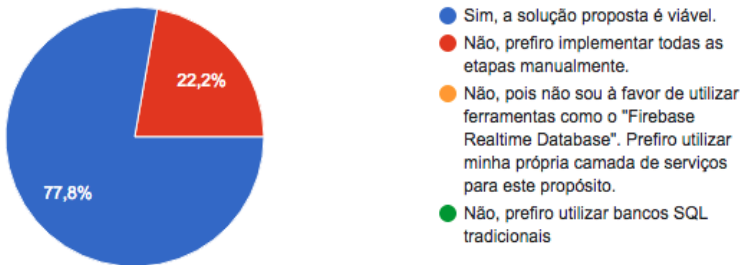
Figura 43 – Avaliação da proposta para a funcionalidade de fluxo de *login*



Fonte: Autor

Os resultados apresentados na Figura 44, apontam que a maioria dos entrevistados (77,8%) consideram viável a estrutura proposta para a funcionalidade de banco de dados. Apenas dois dos especialistas consultados consideram mais apropriado implementar manualmente, todas as rotinas de interação com o banco de dados. Nenhum dos entrevistados selecionou a opção que indica aversão a ferramenta Firebase Realtime Database®.

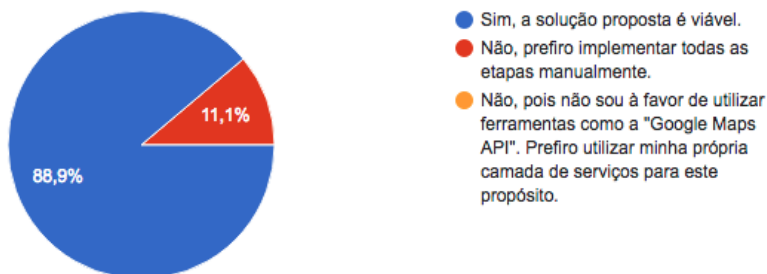
Figura 44 – Avaliação da proposta para a funcionalidade de banco de dados



Fonte: Autor

A solução proposta para capturar a localização do usuário, obter dados relevantes sobre sua localização e buscar dados relacionados a locais, construções e estabelecimentos, em aplicativos, foi muito bem avaliada entre os entrevistados. Nenhum especialista se posicionou contra a utilização da ferramenta Google Maps APIS®, sendo que oito (88,9%) dos nove especialistas consideram a solução viável e apenas um se sente mais confortável e seguro implementando todas as etapas de forma manual (Figura 45).

Figura 45 – Avaliação da proposta para a funcionalidade de geolocalização



Fonte: Autor

Quando questionados sobre a estrutura desenvolvida para a funcionalidade de notificações (*push notifications*), um dos entrevistados, desenvolvedor na área de IoT a cinco anos, sugeriu o desenvolvimento de uma plataforma customizada para lidar com o envio de notificações na plataforma do Firebase®. O *dashboard* fornecido pelo Firebase®, apresentado na seção 3.8.5 deste trabalho, permite a utilização dos recursos básicos oferecidos pelo Firebase Cloud Message®, ficando por conta dos desenvolvedores a criação de uma ferramenta mais avançada.

A sugestão do entrevistado, na íntegra, foi: *“Uma solução utilizando o FCM (Firebase Cloud Message)® em código e um dashboard customizado para o framework permitirá maior customização nas notificações push, pois a notificação enviada via Firebase Dashboard é bem limitada em comparação a versão em código. A principal desvantagem com o uso do FCM® é que o dashboard Web terá de ser construído do zero”*.

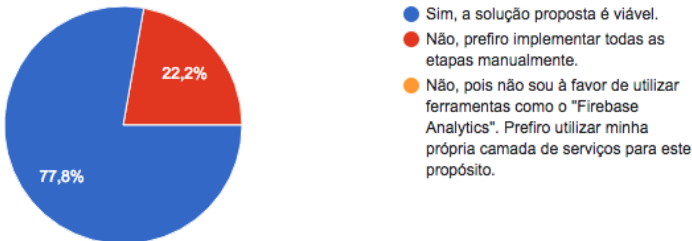
Contudo, conforme apresenta a Figura 46, para a maioria dos entrevistados (77,8%) a solução é viável e nenhum participante se posicionou contra a utilização de ferramentas como o Firebase Cloud Message®, ainda que careça de recursos mais avançados.

Figura 46 – Avaliação da proposta para a funcionalidade notificações (*push notifications*)



Fonte: Autor

Figura 47 – Avaliação da proposta para a funcionalidade de *analytics*



Fonte: Autor

A última funcionalidade do protótipo apresentada no questionário, foi a ferramenta para coleta e análise de dados de uso (*analytics*). Nesta questão, dois (22,2%) dos nove entrevistados afirmam preferir desenvolver toda a estrutura manualmente (Figura 47). Enquanto que os demais entrevistados (77,8%) consideram a solução proposta viável. Nenhum participante afirmou ser contra a utilização de ferramentas como o Firebase Analytics®.

Por fim, foi disponibilizada uma área de texto livre aos entrevistados, para comentários em geral sobre o protótipo. Um dos especialistas sugeriu a inclusão de uma camada de *Machine Learning*, prevendo ações a partir dos dados obtidos através da funcionalidade *analytics* e possibilitando a criação de aplicações mais inteligentes a partir do *framework* proposto.

5 CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

Nos últimos 20 anos, o avanço da tecnologia e sua presença na vida das pessoas vem proporcionando o surgimento de uma vasta gama de possibilidades quanto às aplicações baseadas no conceito de Internet das Coisas.

Neste capítulo são apresentadas as considerações finais em relação ao *framework* proposto neste trabalho, bem como, as pretensões para trabalhos futuros.

5.1 CONSIDERAÇÕES FINAIS

Neste trabalho foi proposto um *framework* e desenvolvido um protótipo voltado à implementação de novas aplicações no contexto da IoT. O *framework* contempla diversas ferramentas comuns em aplicativos, abstraindo dezenas de rotinas. Objetiva assim, auxiliar no processo de desenvolvimento.

As estruturas propostas para o *framework* são baseadas em bibliotecas e APIs de grandes empresas do ramo tecnológico, como Google® e Facebook®. Essas empresas fornecem uma série de funcionalidades e serviços gratuitos, barateando o custo de projetos em fase inicial, e aumentando as possibilidades para o desenvolvimento de aplicações IoT.

Com base no conteúdo do *framework*, os desenvolvedores possuem a disposição de suas aplicações os seguintes elementos: a) uma ferramenta de autenticação com fluxo de *login* completo; b) um banco de dados do tipo não relacional com métodos ágeis para leitura e gravação dos dados; c) uma ferramenta de geolocalização com capacidade para capturar a localização do usuário e retornar informações importantes quanto a sua posição; d) uma estrutura para análise de dados referente ao uso da aplicação (*analytics*) com um *dashboard* completo para visualização das informações; e e) uma funcionalidade para o envio e recebimento de notificações (*push notifications*).

A plataforma utilizada para o desenvolvimento do protótipo (*framework* IoT) foi o sistema operacional móvel Android®, que se destacou entre as outras opções, devido ao ambiente de desenvolvimento livre e a vasta documentação disponível.

O protótipo possui uma classe Java® principal, responsável por se comunicar com os servidores do Firebase®, Google Maps API® e Facebook Account Kit®, permitindo a troca de informações com a aplicação principal, lendo ou gravando dados, gerando o registro das

interações por meio das rotinas de *analytics* ou requisitando informações relevantes quanto à localização do usuário.

Na etapa final do trabalho, o protótipo foi disponibilizado a um grupo de especialistas na área de Internet das Coisas, juntamente com toda a documentação do sistema proposto e um questionário formado por dez perguntas a respeito do mesmo. A análise das respostas coletadas apontou que o *framework* IoT está em linha com o esperado e relatado na literatura. Aponta também que o objetivo principal do trabalho foi atingido, sendo que a estrutura proposta possui potencial para auxiliar no desenvolvimento de aplicações voltadas para a Internet das Coisas.

5.2 TRABALHOS FUTUROS

Durante o desenvolvimento deste trabalho, outras possibilidades de trabalhos futuros foram vislumbradas. Entre estas, pode-se mencionar a sugestão provida por um dos especialistas que participaram da avaliação do *framework* proposto neste trabalho: A disponibilização de técnicas e algoritmos de *Machine Learning* voltada à análise dos dados coletados pela ferramenta de *analytics* com o intuito de realimentar o sistema, permitindo maior autonomia e inteligência.

O *blockchain*, uma tecnologia capaz de rastrear, coordenar, realizar transações e armazenar informações de uma grande quantidade de dispositivos (SAGHIRI et al., 2018), habilita o desenvolvimento de sistemas que dispensam um servidor central intermediando a comunicação. A partir de estudos envolvendo a tecnologia *blockchain*, trabalhos futuros poderiam implementar, no *framework* proposto, uma rede de comunicação descentralizada, permitindo a comunicação de ponta a ponta entre sensores, máquinas e dispositivos.

Para garantir a usabilidade dos sistemas IoT, é muito importante que os desenvolvedores compreendam a crescente produção de dados gerados pelos sistemas computacionais e, principalmente, como os dados são tratados e expostos aos usuários. Este tipo de cenário pode obter suporte nas pesquisas advindas dos Sistemas de Recomendação (RS, do inglês *Recommender Systems*) (RESNICK; VARIAN; EDITORS, 1997).

Dessa forma, o *framework* proposto poderia compreender preferências de usuários, interesses, sentimentos e, a partir disso, disponibilizar ao desenvolvedor, recomendações que possam impactar no dia a dia dos usuários.

Outro fator determinante para os sistemas computacionais é a percepção do contexto em que o sistema está inserido. Na computação, o contexto serve de apoio na interação entre os sistemas e os usuários. Conhecer o contexto em que o sistema está inserido pode possibilitar uma melhor adaptação às necessidades dos usuários, eliminando uma série de informações que, sem a compreensão do contexto, teria de ser provida e configurada manualmente no sistema.

Os sistemas que utilizam esta abordagem em seus algoritmos são chamados de Sistemas Sensíveis ao Contexto (CSS, do inglês *Context Sensitive System*). Estes sistemas são complexos e baseados em uma série de entradas/sensores e, portanto, são também vislumbrados para uma possível implementação em trabalhos futuros.

Por fim, a Google lançou oficialmente em maio de 2018 o Android Things® (MÜLLER, 2018), um sistema operacional totalmente focado em Internet das Coisas. Inicialmente, a proposta é que este seja um sistema padronizado e fechado, com uma forte camada de segurança.

Em trabalhos futuros, o estudo e implantação das diretrizes presentes no Android Things®, permitiriam grandes avanços no *framework* IoT, melhorando sua segurança, inteligência e robustez. O Android Things® habilitaria também a expansão do *framework* para as plataformas que promovem suporte a dispositivos e sensores IoT, ou seja, as coisas (*things*).

Os dispositivos e sensores na IoT são geralmente baseados em processadores *System on a Chip* (SoC), onde uma única placa eletrônica ou *chip*, possui todos os componentes de *hardware* necessários para o desenvolvimento de um sistema embarcado voltado para a Internet das Coisas.

O desenvolvimento destes produtos também exige a utilização de diversas bibliotecas, APIs e ferramentas que são comuns em outros dispositivos e sensores voltados para a IoT. Os próximos estudos em relação ao *framework* proposto poderiam considerar as funcionalidades dos dispositivos SoC junto às características do Android Things®.

REFERÊNCIAS

- ADIONO, T. et al. Design of database and secure communication protocols for Internet-of-things-based smart home system. **TENCON 2017 - 2017 IEEE Region 10 Conference**, p. 1273–1278, 2017.
- AFZAL, B. et al. Enabling IoT platforms for social IoT applications: Vision, feature mapping, and challenges. **Future Generation Computer Systems**, 2017.
- AHMED, E. et al. Internet-of-things-based smart environments: State of the art, taxonomy, and open research challenges. **IEEE Wireless Communications**, v. 23, n. 5, p. 10–16, 2016.
- AKPAKWU, G. A. et al. A Survey on 5G Networks for the Internet of Things: Communication Technologies and Challenges. **IEEE Access**, v. 3536, n. c, p. 1–1, 2017.
- AL-KHATER, W.; KUNHOTH, S.; AL-MAADEED, S. A review on Radio Frequency Identification methods. **Department of Computer Science and Engineering**, n. 2, p. 1751–1758, 2017.
- ALI, M. et al. IoTFLiP: IoT-based Flip Learning Platform for Medical Education. **Digital Communications and Networks**, 2017.
- ALLIANCE, A. **Open Connectivity Foundation e AllSeen Alliance se fundem**. 2017. Disponível em: <<https://allseenalliance.org/>>. Acesso em: 24 maio. 2017.
- ALMEIDA, J. M. F. DE. Breve história da INTERNET. **Universidade do Minho. Departamento de Sistemas de Informação**, n. Ww li, 2005.
- AMAZON WEB SERVICES. **Internet das Coisas**. 2017. Disponível em: <<https://aws.amazon.com/pt/iot/>>. Acesso em: 22 fev. 2018.
- AMMAR, M.; RUSSELLO, G.; CRISPO, B. Internet of Things: A survey on the security of IoT frameworks. **Journal of Information Security and Applications**, v. 38, p. 8–27, 2018.
- APPLE. **Home Kit**. 2018. Disponível em: <<https://developer.apple.com/homekit/>>. Acesso em: 22 fev. 2018.
- ASHTON, K. That ' Internet of Things ' Thing. **RFID Journal**, p. 4986, 2009.
- BAE, W. IL; KWAK, J. Smart card-based secure authentication protocol in multi-server IoT environment. **Multimedia Tools and Applications**, p. 1–19, 2017.

BEVILAQUA, A. C. **Uma Comparação de Desenvolvimento Web entre a Tecnologia Servlets / JSP e o Framework Vraprot**. [s.l.] UNIVERSIDADE FEDERAL DE JUIZ DE FORA - UFJF, 2013.

BICEN, H.; ARNAVUT, A. Determining the effects of technological tool use habits on social lives. **Computers in Human Behavior**, v. 48, p. 457–462, 2015.

BLUETOOTH SIG. **Bluetooth Low Energy - Bluetooth Development Portal**. 2015. Disponível em: <<https://developer.bluetooth.org/TechnologyOverview/Pages/BLE.aspx>>. Acesso em: 30 out. 2015.

CADAVIECO, J. F. La interactividad de los dispositivos móviles geolocalizados, una nueva relación entre personas y cosas. **Ilu**, v. 18, p. 777–788, 2013.

CAMARGO, W. B. DE. **Interfaces Programação Orientada a Objetos**. 2010. Disponível em: <<https://www.devmedia.com.br/interfaces-programacao-orientada-a-objetos/18695>>. Acesso em: 24 jun. 2018.

CARVALHO, V. **É isso mesmo: a Microsoft acaba de lançar nova atualização cumulativa para o Windows 10 Mobile - Tudocelular**. 2010. Disponível em: <<https://www.tudocelular.com/windows-phone/noticias/n124419/microsoft-libera-atualizacao-windows-10-mobile.html>>. Acesso em: 17 jul. 2018.

CHANG, K.-H.; CONSULTING, C. Bluetooth: A Viable Solution for IoT? n. December, 2014.

CICCOZZI, F.; MALAVOLTA, I.; SELIC, B. Execution of UML models: a systematic review of research and practice. **Software & Systems Modeling**, 2018.

CISCO. **IoT Dev Center**. 2018. Disponível em: <<https://developer.cisco.com/site/iot/>>. Acesso em: 20 fev. 2018.

CURRY, E. et al. IoT-Enhanced User Experience for Smart Water and Energy Management. **IEEE Internet Computing**, n. February, p. 18–28, 2018.

DAVID, F. et al. **Smart geocoding of objects**. 2016 17th International Carpathian Control Conference (ICCC). **Anais...IEEE**, maio 2016

DEVELOPERS, G. **The Google Geocoding API**. 2015. Disponível em: <<https://developers.google.com/maps/documentation/geocoding/>>. Acesso em: 15 abr. 2018.

DEVELOPERS, G. **Firebase: Pricing**. 2016. Disponível em: <<https://firebase.google.com/pricing/>>. Acesso em: 15 abr. 2018a.

- DEVELOPERS, G. **Download Android Studio and SDK Tools**. 2016. Disponível em: <<https://developer.android.com/studio/index.html>>. Acesso em: 10 jul. 2018b.
- DEVELOPERS, G. **Map Coverage Details | Google Maps APIs | Google Developers**. 2017. Disponível em: <<https://developers.google.com/maps/coverage>>. Acesso em: 15 abr. 2018a.
- DEVELOPERS, G. **Manifesto do aplicativo**. 2017. Disponível em: <<https://developer.android.com/guide/topics/manifest/manifest-intro.html?hl=pt-br>>. Acesso em: 3 jul. 2018b.
- DEVELOPERS, G. **Cloud Firestore**. 2018. Disponível em: <<https://firebase.google.com/docs/firestore/>>. Acesso em: 15 abr. 2018a.
- DEVELOPERS, G. **Criar uma biblioteca Android - Android Developers**. 2018. Disponível em: <<https://developer.android.com/studio/projects/android-library>>. Acesso em: 2 jul. 2018b.
- DHILLON, P. K.; KALRA, S. A lightweight biometrics based remote user authentication scheme for IoT services. **Journal of Information Security and Applications**, v. 34, p. 255–270, 2017.
- EVANS, D. A Internet das Coisas Como a próxima evolução da Internet está mudando tudo A Internet das Coisas Como a próxima evolução da Internet está mudando tudo. 2011.
- FACEBOOK. **Visão geral - Account Kit**. 2018. Disponível em: <<https://developers.facebook.com/docs/accountkit/overview>>. Acesso em: 10 abr. 2018.
- FARAHANI, B. et al. Towards fog-driven IoT eHealth: Promises and challenges of IoT in medicine and healthcare. **Future Generation Computer Systems**, v. 78, p. 659–676, 2018.
- FATIMA, H.; WASNIK, K. Comparison of SQL, NoSQL and NewSQL databases for internet of things. **IEEE Bombay Section Symposium 2016: Frontiers of Technology: Fuelling Prosperity of Planet and People, IBSS 2016**, 2017.
- FERNANDEZ-GAGO, C.; MOYANO, F.; LOPEZ, J. Modelling Trust Dynamics in the Internet of Things. **Information Sciences**, v. 396, p. 72–82, 2017.
- FILIPE GARRETT. **Fim do Windows Phone_ Microsoft encerra suporte à versão 8**. 2017. Disponível em: <<https://www.techtudo.com.br/noticias/2017/07/fim-do-windows->

phone-microsoft-encerra-suporte-para-a-versao-81.ghtml>. Acesso em: 17 jul. 2018.

FIREBASE. **Firestore Cloud Messaging**. 2017. Disponível em: <<https://firebase.google.com/docs/cloud-messaging/>>. Acesso em: 16 abr. 2018.

GADRE, D. V. et al. Embedded Systems and Internet of Things (IoTs) - Challenges in Teaching the ARM Controller in the Classroom Embedded Systems and Internet of Things (IoTs) and Challenges in Teaching the ARM Controller in the Classroom Abstract. **ASEE Annual Conference and Exposition, Conference Proceedings**, v. Volume 201, 2017.

GOOGLE INC. **Firestore Authentication**. 2017. Disponível em: <<https://firebase.google.com/docs/auth/>>. Acesso em: 22 abr. 2017.

HELPS, C. R.; ARMSTRONG, J. Review of Current Embedded System Hardware, OS, Development Systems and Application Domains for Instructional Design. **ASEE Annual Conference and Exposition, Conference Proceedings**, 2007.

HUANG, Y.; LI, G. Descriptive models for Internet of Things. **International Conference on Intelligent Control and Information Processing**, p. 483–486, 2010.

IOTLEAGUE. **About Us**. 2018. Disponível em: <<http://www.iotleague.com/about-us/>>. Acesso em: 6 fev. 2018.

JAYARAMAN, P. P. et al. Privacy preserving Internet of Things: From privacy techniques to a blueprint architecture and efficient implementation. **Future Generation Computer Systems**, v. 76, p. 540–549, 2017.

JEBA MALAR, A. C.; KOUSALYA, G.; MA, M. Markovian model based indoor location tracking for Internet of Things (IoT) applications. **Cluster Computing**, p. 1–8, 2017.

JOHANSSON, B. A.; MAGNUSSON, B. Towards end-user development of graphical user interfaces for internet of things. **Future Generation Computer Systems**, 2017.

JUNQUEIRA, F. **É o fim! BlackBerry anuncia morte do sistema operacional próprio e usará Android - Tudocelular**. 2016. Disponível em: <<https://www.tudocelular.com/android/noticias/n69964/blackberry-10-morreu.html>>. Acesso em: 17 jul. 2018.

KANG, J. W.; YU, Q.; GOLEN, E. Teaching IoT (Internet of Things) Analytics. **Proceedings of the 18th Annual Conference on Information Technology Education - SIGITE '17**, p. 135–140, 2017.

- KARVINEN, K.; KARVINEN, T. IoT Rapid Prototyping Laboratory Setup. **International Journal of Engineering Education**, v. 34, p. 263–272, 2018.
- KHAN, M. A.; SALAH, K. IoT security: Review, blockchain solutions, and open challenges. **Future Generation Computer Systems**, v. 82, p. 395–411, 2018.
- KIM, T. HOON; RAMOS, C.; MOHAMMED, S. Smart City and IoT. **Future Generation Computer Systems**, v. 76, n. July 2014, p. 159–162, 2017.
- KOLLMANN, S. A.; BERESFORD, A. R. The cost of push notifications for smartphones using tor hidden services. **Proceedings - 2nd IEEE European Symposium on Security and Privacy Workshops, EuroS and PW 2017**, p. 76–85, 2017.
- KRISHNA, M. V.; REDDY, A. B.; SANDEEP, V. Facial recognition enabled smart door unlock system. **International Journal of Engineering & Technology Website: www.sciencepubco.com/index.php/IJET Research Paper Facial**, v. 7, p. 183–186, 2018.
- KRISHNAMURTHY, R.; CECIL, J. A next-generation IoT-based collaborative framework for electronics assembly. **International Journal of Advanced Manufacturing Technology**, p. 1–14, 2018.
- KROTOV, V. The Internet of Things and new business opportunities. **Business Horizons**, v. 60, n. 6, p. 831–841, 2017.
- KUROSE, J. F.; ROSS, K. W. Redes de computadores e a internet: uma abordagem top-down. In: **Redes de computadores e a internet: uma abordagem top-down**. [s.l: s.n.]. p. 614.
- LASAOSA, R. **Google Analytics para dummies – Rufino Lasaosa**. 2016. Disponível em: <<http://www.rufinolasosa.es/marketing/google-analytics-para-dummies/>>. Acesso em: 15 abr. 2018.
- LOPEZ, J. et al. Evolving privacy: From sensors to the Internet of Things. **Future Generation Computer Systems**, v. 75, p. 46–57, 2017.
- MANIAPPAN. **List of 10 IoT Frameworks And Platforms**. 2015. Disponível em: <<http://www.iotleague.com/list-of-10-iot-frameworks-and-platforms/>>. Acesso em: 6 fev. 2018.
- MANSFIELD-DEVINE, S. Open source and the Internet of Things. **Network Security**, v. 2018, n. 2, p. 14–19, 2018.
- MARSH. La gestión de riesgos en latinoamérica. **I BENCHMARK DE GESTIÓN DE RIESGOS EN LATINOAMÉRICA**, 2015.

MASHAL, I.; CHUNG, T. Y.; ALSARYRAH, O. Toward service recommendation in Internet of Things. **International Conference on Ubiquitous and Future Networks, ICUFN**, v. 2015–August, p. 328–331, 2015.

MAZAYEV, A.; MARTINS, J. A.; CORREIA, N. Interoperability in IoT through the Semantic Profiling of Objects. **IEEE Access**, p. 1–7, 2017.

MINETTO, E. L. **Frameworks para Desenvolvimento em PHP**. Novatec ed. [s.l: s.n.].

MORA, S.; GIANNI, F.; DIVITINI, M. RapIoT toolkit: Rapid prototyping of collaborative Internet of Things applications. **Proceedings - 2016 International Conference on Collaboration Technologies and Systems, CTS 2016**, p. 438–445, 2016.

MSHALI, H. et al. A survey on health monitoring systems for health smart homes. **International Journal of Industrial Ergonomics**, v. 66, p. 26–56, 2018.

MÜLLER, L. **Google lança oficialmente o Android Things 1**. 2018. Disponível em: <<https://www.tecmundo.com.br/software/130023-google-lanca-oficialmente-android-things-1-0-dia-i-o-2018.htm>>. Acesso em: 16 jul. 2018.

NG, I. C. L.; WAKENSHAW, S. Y. L. The Internet-of-Things : Review and research directions. **International Journal of Research in Marketing**, v. 34, n. 1, p. 3–21, 2017.

NOVAC, O. C. et al. Comparative study of Google Android, Apple iOS and Microsoft Windows Phone mobile operating systems. **2017 14th International Conference on Engineering of Modern Electric Systems, EMES 2017**, p. 154–159, 2017.

OUHBI, S. et al. Compliance of Blood Donation Apps with Mobile OS Usability Guidelines. **Journal of Medical Systems**, v. 39, n. 6, 2015.

PATEL, P.; INTIZAR ALI, M.; SHETH, A. On Using the Intelligent Edge for IoT Analytics. **IEEE Intelligent Systems**, v. 32, n. 5, p. 64–69, 2017.

PPGTIC. **Linhas de Pesquisa**. 2018. Disponível em: <<http://ppgtic.ufsc.br/linhas-de-pesquisa/>>. Acesso em: 19 fev. 2018.

PRABHU, C. S. R.; REDDI, A. P. **Bluetooth Technology: and its applications with JAVA and J2ME**. [s.l: s.n.].

PTC. **ThingWorx oferece inovação industrial**. 2018. Disponível em: <<https://www.ptc.com/pt/products/iot>>. Acesso em: 20 fev. 2018.

- RESNICK, P.; VARIAN, H. R.; EDITORS, G. Recommender Systems. **Communications of the ACM**, v. 40, n. 3, p. 56–58, 1997.
- RISTESKA STOJKOSKA, B. L.; TRIVODALIEV, K. V. A review of Internet of Things for smart home: Challenges and solutions. **Journal of Cleaner Production**, v. 140, p. 1454–1464, 2017.
- RUIZ-ROSETO, J. et al. Internet of things: A scientometric review. **Symmetry**, v. 9, n. 12, 2017.
- SACCOL, A. Z.; REINHARD, N. Tecnologias de informação móveis, sem fio e ubíquas: definições, estado-da-arte e oportunidades de pesquisa. **Revista de Administração Contemporânea**, v. 11, n. 4, p. 175–198, 2007.
- SAGHIRI, A. M. et al. A Framework for Cognitive Internet of Things based on Blockchain. **4th International Conference on Web Research**, p. 138–143, 2018.
- SIERRA, K.; BATES, B. **Use a Cabeça! Java - 2ª Edição**. [s.l.: s.n.].
- SILVA, P. E. **UTILIZAÇÃO DO FRAMEWORK CAKEPHP PARA DESENVOLVIMENTO DE WEBSITES EM PHP**. [s.l.] UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ – UTFPR CURSO, 2011.
- SOMMERVILLE, I. **ENGENHARIA DE SOFTWARE**. São Paulo ed. [s.l.] São Paulo: Pearson Education do Brasil, 2007.
- SURESH, P.; DANIEL, J. V.; ASWATHY, R. H. A state of the art review on the Internet of Things (IoT) History , Technology and fields of deployment. 2014.
- TAO, M. et al. Multi-layer cloud architectural model and ontology-based security service framework for IoT-based smart homes. **Future Generation Computer Systems**, v. 78, p. 1040–1051, 2018.
- USHA DEVI, G. et al. Mutual authentication scheme for IoT application. **Indian Journal of Science and Technology**, v. 8, n. 26, 2015.
- VIEIRA, V.; TEDESCO, P.; SALGADO, C. Modelos e Processos para o Desenvolvimento de Sistemas Sensíveis ao Contexto. **Challenges**, v. 20, n. 3, p. 381–431, 2009.
- YANG, J. et al. Optimization of real-time traffic network assignment based on IoT data using DBN and clustering model in smart city. **Future Generation Computer Systems**, 2017.
- YING, J.; PAHLAVAN, K.; LI, X. Precision of RSS-based Indoor Geolocation in IoT Applications. **2017 IEEE 28th Annual**

International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), 2017.

ZANELLA, A et al. Internet of Things for Smart Cities. **IEEE Internet of Things Journal**, v. 1, n. 1, p. 22–32, 2014.

ANEXO A – LICENÇA

MIT License

Copyright (c) 2018 THYERRI FERNANDES MEZZARI

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

APÊNDICE A – RECURSOS DECLARADOS NO ANDROID® GRADLE E ANDROID® MANIFEST

Gradle:

com.android.support.appcompat-v7:26.1.0

com.android.support.constraint:constraint-layout:1.1.2

com.facebook.android:account-kit-sdk:4.+

com.google.firebase:firebase-database:16.0.1

com.google.firebase:firebase-auth:15.0.0

com.google.firebase:firebase-core:15.0.0

com.google.firebase:firebase-messaging:15.0.0

Android Manifest:

android.permission.INTERNET

android.permission.ACCESS_FINE_LOCATION

APÊNDICE B – QUESTIONÁRIO DE AVALIAÇÃO DO FRAMEWORK

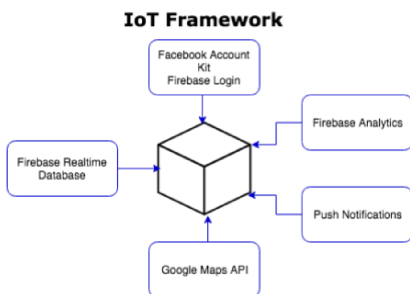
Framework para Internet das Coisas

Este questionário faz parte de uma pesquisa para a dissertação de mestrado de Thyerri Fernandes Mezzari, mestrando no Programa de Pós-Graduação em Tecnologias da Informação e Comunicação (PPGTIC) pela Universidade Federal de Santa Catarina, orientado pelo Dr. Alexandre Leopoldo Gonçalves e visa avaliar o framework proposto.

O projeto consiste em conceituar, desenvolver e implementar um framework que auxilie no desenvolvimento de aplicativos multi-plataforma (Android/iOS/Web) com foco na Internet das Coisas (IoT). Você já implementou um aplicativo e precisou de um sistema de login? Preciso de banco de dados? Preciso de geolocalização? Preciso analisar os dados de uso do aplicativo? Está é a ideia deste projeto, criar uma framework que contemple os requisitos mais comuns em um aplicativo IoT.

*Obrigatório

Visão geral do Framework



1. 1 - Escolaridade *

Marque todas que se aplicam.

- Ensino fundamental completo
- Ensino médio completo
- Ensino Superior Incompleto
- Superior completo
- Especialização
- Mestrado
- Doutorado

2. 2 - Área de Atuação *

3. 3 - Tempo de atuação em projetos IoT *

Marcar apenas uma oval.

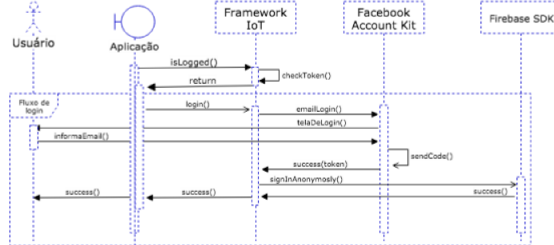
- Menos de 1 ano
- De 1 a 5 anos
- De 5 a 10 anos
- Mais de 10 anos

4. 4 - Assinale as opções que fizeram parte de algum projeto IoT que você participou/desenvolveu:

*
Marque todas que se aplicam.

- Fluxo Login e/ou autenticação
- Banco de Dados
- Dados de geolocalização como posição geográfica do usuário ou busca por informações de locais.
- Notificações (Push Notifications)
- Analytics - Coleta e análise de dados relacionados ao uso da aplicação
- Outro: _____

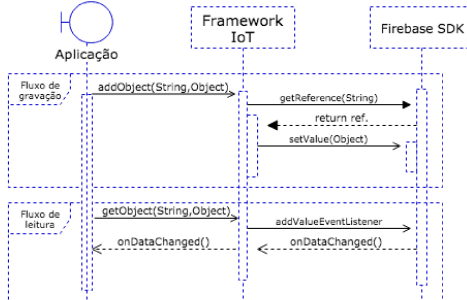
5. 5 - Considere as formas tradicionais para a implementação de uma estrutura de login em um aplicativo multiplataforma e compare com a sequência de login proposta (imagem), onde todo o fluxo de login é gerenciado pelo framework. Você considera a solução proposta viável? *



Marcar apenas uma oval.

- Sim, a solução proposta é viável.
- Não, prefiro implementar todas as etapas manualmente.
- Não, pois não sou à favor de utilizar ferramentas como o "Facebook Account Kit" e o "Firebase Authentication". Prefiro utilizar minha própria camada de serviços para este propósito.
- Outro: _____

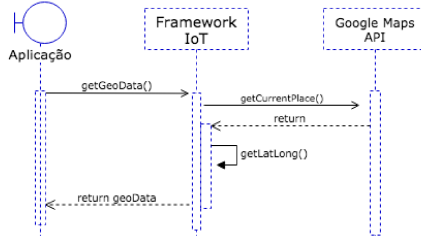
6. 6 - Considere as formas tradicionais para a implementação de uma estrutura de banco de dados em um aplicativo multiplataforma e compare com a sequência de leitura e gravação proposta (imagem), onde todas as operações são gerenciadas pelo framework. Você considera a solução proposta viável? *



Marcar apenas uma oval.

- Sim, a solução proposta é viável.
- Não, prefiro implementar todas as etapas manualmente.
- Não, pois não sou à favor de utilizar ferramentas como o "Firebase Realtime Database". Prefiro utilizar minha própria camada de serviços para este propósito.
- Não, prefiro utilizar bancos SQL tradicionais
- Outro: _____

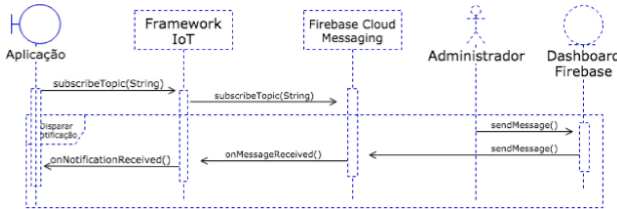
7. 7 - Considere as formas tradicionais capturar a localização do usuário, obter dados relevantes sobre sua localização e/ou buscar dados relacionados a locais, construções e estabelecimentos, em aplicativos multi-plataforma. Compare com a solução proposta (imagem) onde todas as operações são gerenciadas pelo framework. Você considera a solução proposta viável? *



Marcar apenas uma oval.

- Sim, a solução proposta é viável.
- Não, prefiro implementar todas as etapas manualmente.
- Não, pois não sou à favor de utilizar ferramentas como a "Google Maps API". Prefiro utilizar minha própria camada de serviços para este propósito.
- Outro: _____

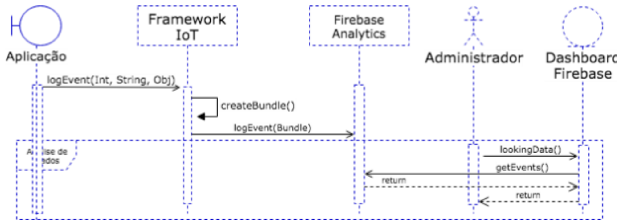
8 - Considere as formas tradicionais para envio e recebimento de notificações (Push Notifications) em aplicativos multiplataforma. Compare com a solução proposta (imagem) onde todas as operações são gerenciadas pelo framework. Você considera a solução proposta viável? *



Marcar apenas uma oval.

- Sim, a solução proposta é viável.
- Não, prefiro implementar todas as etapas manualmente.
- Não, pois não sou à favor de utilizar ferramentas como o "Firebase Cloud Message". Prefiro utilizar minha própria camada de serviços para este propósito.
- Outro: _____

9 - Considere as formas tradicionais para registro e análise de dados relacionados ao uso (Analytics) em aplicativos multiplataforma. Compare com a solução proposta (imagem) onde todas as operações são gerenciadas pelo framework. Você considera a solução proposta viável? *



Marcar apenas uma oval.

- Sim, a solução proposta é viável.
- Não, prefiro implementar todas as etapas manualmente.
- Não, pois não sou à favor de utilizar ferramentas como o "Firebase Analytics". Prefiro utilizar minha própria camada de serviços para este propósito.
- Outro: _____

10. 10 - Deixe seu comentário ou sugestão:
