

**INSTITUTO FEDERAL DE SANTA CATARINA**

**TAILOR ANDRES LÓPEZ MAFFINI**

**IOT CONN: APLICATIVO PARA COMUNICAÇÃO  
ENTRE ANDROID E DISPOSITIVO IOT VIA  
PROTOCOLO MQTT**



**TAILOR ANDRES LÓPEZ MAFFINI**

**IOT CONN: APLICATIVO PARA COMUNICAÇÃO ENTRE ANDROID  
E DISPOSITIVO IOT VIA PROTOCOLO MQTT**

Trabalho de Conclusão de Curso apresentado ao  
Curso Superior de Tecnologia em Análise e De-  
senvolvimento de Sistemas como requisito parcial  
para obtenção do grau de Tecnólogo em Análise e  
Desenvolvimento de Sistemas.  
Orientador: Andreu Carminati, Dr.

Ficha de identificação da obra elaborada pelo autor,  
por meio do programa de geração automática campus Gaspar, do IFSC

Maffini, Tailor Andres López  
IOT CONN: APLICATIVO PARA COMUNICAÇÃO ENTRE ANDROID  
E DISPOSITIVO IOT VIA PROTOCOLO MQTT / Tailor Andres López  
Maffini ; orientação de Andreu Carminati. Gaspar,  
SC, 2020.  
40 p.

Trabalho de Conclusão de Curso (TCC) - Instituto Federal  
de Santa Catarina, Campus Gaspar. Curso Superior  
de Tecnologia em Análise e Desenvolvimento de  
Sistemas. .  
Inclui Referências.

1. Android. 2. Internet das Coisas. 3. MQTT. 4.  
Código aberto. I. Carminati, Andreu. II. Instituto Federal  
de Santa Catarina. . III. Título.

## RESUMO

Em todos os lugares existe uma crescente demanda por dispositivos que auxiliem em tarefas diárias. Neste trabalho será apresentado um aplicativo de código aberto, que é uma ferramenta que tem como foco realizar a comunicação do *smartphone* com dispositivos e protótipos de Internet das coisas, em inglês *Internet of things* (IoT). Ele é responsável pelo envio e recebimento de mensagens de dispositivos e para realizar a troca de informações foi utilizado o protocolo de comunicação *Message Queue Telemetry Transport* (MQTT), e é permitido ao usuário adicionar vários dispositivos e ter os atalhos para cada um na tela principal do *software*. Também é possível criar uma conta para realizar a sincronização dos dados em nuvem, permitindo que estes sejam resgatados futuramente ou em outro *smartphone*. O aplicativo foi desenvolvido em linguagem Java utilizando as bibliotecas para Android e para uso do protocolo MQTT foi utilizada a biblioteca Eclipse Paho.

**Palavras-chave:** Android. Internet das Coisas. MQTT. Código aberto.



## ABSTRACT

The increased use of technology in everyday life is a global fact. Everywhere there is a growing demand for devices that assist with daily tasks. In this paper we present the open source application, which is a tool for communicating the smartphone with devices, focusing on Internet of things (IoT) devices and prototypes. It is responsible for sending and receiving messages from devices and the Message Queue Telemetry Transport (MQTT) communication protocol was used to exchange information, and is possible to the user to add multiple devices and have shortcuts for each of them on the main software screen. It is also be able to create an account to sync your data, allowing it to be redeemed in the future or on another smartphone. The application was developed in Java language using the libraries for Android and to use the MQTT protocol, the Eclipse Paho library was used.

**Keywords:** Android. Internet of Things. MQTT. Open source.





## LISTA DE ILUSTRAÇÕES

Figura 1 – Estrutura do Android . . . . .	17
Figura 2 – Internet das coisas em casa . . . . .	18
Figura 3 – Mensagem MQTT . . . . .	19
Figura 4 – Estrutura de base de dados do firebase . . . . .	20
Figura 5 – Google Home Mini . . . . .	21
Figura 6 – Amazon Echo . . . . .	22
Figura 7 – Diagrama de casos de uso do aplicativo . . . . .	26
Figura 8 – Diagrama de atividade . . . . .	28
Figura 9 – Diagrama de classes . . . . .	29
Figura 10 – Diagrama de sequência . . . . .	30
Figura 11 – Telas do sistema . . . . .	32
Figura 12 – Comunicação efetuada . . . . .	34



## LISTA DE TABELAS

Tabela 1 – Tabela de requisitos funcionais . . . . .	25
Tabela 2 – Tabela de requisitos não funcionais . . . . .	26



## LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
ART	Android Runtime
HAL	Hardware Abstraction Layer
XML	Extensible Markup Language
IFSC	Instituto Federal de Santa Catarina
IOT	Internet of Things
JSON	JavaScript Object Notation
MIT	Massachusetts Institute of Technology
MQTT	Message Queue Telemetry Transport
REST	Representational State Transfer
SGBD	Sistema de Gerenciamento de Banco de Dados
SO	Sistema Operacional
SOAP	Simple Object Access Protocol



## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>15</b>
1.1	Objetivos	15
1.1.1	Objetivo geral	15
1.1.2	Objetivos específicos	15
1.2	Justificativa	16
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>17</b>
2.1	Android	17
2.2	Internet das Coisas	18
2.2.1	Protocolo para troca de Mensagens MQTT	18
2.2.1.1	Eclipse Mosquitto	19
2.2.1.2	Eclipse Paho	19
2.3	Banco de Dados	20
2.3.1	Firestore Realtime Database	20
2.4	Trabalhos correlatos	20
2.4.1	Google Home	21
2.4.2	Amazon Alexa	21
2.4.3	MQTT Dash	22
2.4.4	IoT MQTT Panel	22
2.4.5	Comparativo	22
2.5	Resumo do Capítulo	23
<b>3</b>	<b>MATERIAIS E MÉTODOS</b>	<b>25</b>
3.1	Visão Geral do projeto	25
3.2	Requisitos funcionais	25
3.3	Requisitos não funcionais	25
3.4	Diagramas	26
3.4.1	Casos de uso	26
3.4.2	Diagrama de atividades	28
3.4.3	Diagrama de classes	28
3.4.4	Diagrama de sequência	30
3.5	Resumo do Capítulo	30
<b>4</b>	<b>RESULTADOS OBTIDOS</b>	<b>31</b>
4.1	Imagens	31
4.1.1	Telas do aplicativo	31
4.1.2	Comunicação MQTT	31
<b>5</b>	<b>CONCLUSÃO</b>	<b>35</b>
5.1	Trabalhos Futuros	35
	<b>REFERÊNCIAS</b>	<b>37</b>





## 1 INTRODUÇÃO

Segundo Santos e Freitas (2016), havia cerca de 0,07 dispositivos conectados para cada pessoa no mundo em 2003, um número que muda muito principalmente com a explosão dos *smartphones* no mercado. Já em 2010 haviam 1,84 dispositivos conectados por pessoa, sendo um total de 12,5 bilhões de dispositivos sendo maioria deles *smartphones*. E o crescimento não para, já que em 2020 foi atingido 25 bilhões de dispositivos e para 2022 a projeção é de quase 30 bilhões de dispositivos conectados (COSTA, 2019). Este crescimento representa cada vez mais dispositivos inteligentes interativos e realizando operações do nosso cotidiano de maneira automatizada. Internet das coisas é a conexão do mundo físico com o mundo digital, interligação entre a nossa vida e os diversos aparelhos eletrônicos que possuem conexão com a internet, facilitando e inovando a vida das pessoas (SILVA, 2017).

De acordo com Amazon (2020) já existem sistemas como o da Alexa, que servem como uma assistente dentro de casa, alguém que faz as coisas por você e se conecta com outros dispositivos que possuem suporte. Outro modelo parecido é o Google Home, e, conforme o Google (2020), este se propõe também a ser um assistente e fazer atividades pelo usuário dentro de sua casa e também se conectando com outros aparelhos dentro de casa que tenham suporte para tal. Para ambos os dispositivos, é necessário investimento na compra dos materiais, além de conhecimento do produto, sendo que ambos não possibilitam personalização por parte do desenvolvedor.

*Maker Culture* é o termo utilizado para um movimento onde as pessoas conseguem construir, consertar ou criar seus próprios objetos (PINTO, 2018). Este projeto tem como foco o *maker* que busca desenvolver um dispositivo voltado a IoT, que precise se comunicar com um *smartphone* e fornecer a ele um aplicativo para conexão e controle utilizando de protocolo MQTT. O aplicativo evita que este precise focar na parte de *software* e comunicação com o *smartphone*, dando para ele a solução de comunicação de forma mais simplificada possível.

### 1.1 Objetivos

A seguir serão apresentados o objetivo geral e os específicos deste projeto.

#### 1.1.1 Objetivo geral

Desenvolver um aplicativo gratuito e de código aberto para comunicação com dispositivos utilizando o protocolo MQTT.

#### 1.1.2 Objetivos específicos

- a) Fornecer uma ferramenta simplificada para conexão entre dispositivos IoT e o *smartphone* utilizando protocolo MQTT;
- b) Fornecer uma ferramenta que permita criar um usuário e salvar as informações sobre os

- c) Disponibilizar o código fonte em comunidades na Internet para que, se necessário, o usuário possa realizar suas modificações e incrementos.

## 1.2 Justificativa

De acordo com [Costa \(2019\)](#) existe uma crescente onda de novos dispositivos conectados na internet e a tendência é termos quase 30 bilhões de dispositivos conectados no planeta até 2022, e o aumento de dispositivos também acontece dentro das residências. Segundo [Santos e Freitas \(2016\)](#) existe uma crescente demanda no mundo de IoT e com isso surgem também novos projetos, porém muitos deles principalmente estudantis ou pequenos protótipos não necessitam de um *software* desenvolvido especificamente para ele para garantir seu funcionamento. Este projeto busca facilitar o trabalho do desenvolvedor de *hardware* e dispositivos eletrônicos automatizados, garantindo que o mesmo foque em desenvolver apenas o dispositivo de seu interesse e fornecer a ele um aplicativo para a comunicação do *smartphone* Android com seu *hardware* via protocolo MQTT.

O aplicativo permite a criação de um usuário para salvar todas as configurações inseridas na nuvem, para que em outro momento possam ser obtidas de volta, sem a necessidade de configurar tudo novamente. Ele tem a responsabilidade de apresentar ao usuário a situação atual do dispositivo e permitir que ele envie comandos ao mesmo. O dispositivo pode ser, por exemplo, um Arduino e este recebe ou envia mensagens ao *smartphone*. O sistema também tem uma fácil conectividade, permitindo que usuários criem seus próprios protótipos para se comunicar com ele, como por exemplo estudantes de automação controlando seu protótipo utilizando o sistema para comunicação entre o dispositivo e o *smartphone*.

O restante do trabalho está organizado da seguinte maneira: no Capítulo 2 será apresentada a fundamentação teórica; No Capítulo 3 são apresentados os requisitos, diagramas e casos de uso; O Capítulo 4 apresentará os resultados obtidos e o Capítulo 5 terá a conclusão e ideias para trabalhos futuros.

## 2 FUNDAMENTAÇÃO TEÓRICA

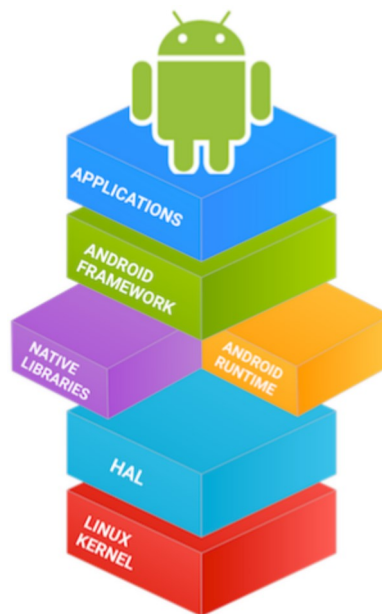
Neste capítulo, serão apresentados os assuntos necessários para o desenvolvimento desse projeto. Será apresentado conceitos sobre Android, IoT, MQTT e banco de dados. Em seguida serão apresentados também aprofundamentos sobre tecnologias e trabalhos correlatos.

### 2.1 Android

O sistema operacional Android foi desenvolvido pela Android, Inc., que foi adquirida pelo Google em 2005. Em 2013 81,3% dos *smartphones* e 56,5% dos tablets distribuídos pelo mundo continham Android instalado, no total havendo 1,5 milhões de dispositivos com o mesmo (DEITEL, 2015). Em 2019 a Google anunciou ter 2.5 bilhões de dispositivos no mundo utilizando o Android (BRANDOM, 2019). O Sistema Operacional (SO) contém diversas versões e cada versão com respectivos nomes como por exemplo 2.3 Gingerbread e 4.4 KitKat (DEITEL, 2015).

Conforme Android (2018), ele é um sistema de código aberto baseado no Linux, conforme demonstrado na Figura 1, é dividido em diversas camadas sendo elas: Kernel do Linux, *hardware* Abstraction Layer (HAL), Bibliotecas C/C++, Android Runtime (ART), Java API Framework e Aplicativos do sistema.

Figura 1: Estrutura do Android



Fonte: Diolinux (2017)

Android Runtime (ART), sucessora ao Dalvik, tem a responsabilidade de executar várias máquinas virtuais. Fica a cargo deste a execução dos aplicativos. Kernel é do Linux que, por exemplo, é confiado pelo ART para encadeamento e gerenciamento de memória de baixo nível (ANDROID, 2018).

A camada de abstração de *hardware* (HAL) que consiste em módulos de biblioteca como o módulo de câmera ou bluetooth. Na realização de uma chamada de uma Interface de Programação de Aplicativos, em inglês *Application Programming Interface* (API) para acessar um *hardware*,

Os serviços como ART e HAL dependem de bibliotecas nativas programadas em C ou C++. É possível, por exemplo, manipular gráficos em 2D e 3D acessando a OpenGL ES pela JAVA OpenGL API (ANDROID, 2018).

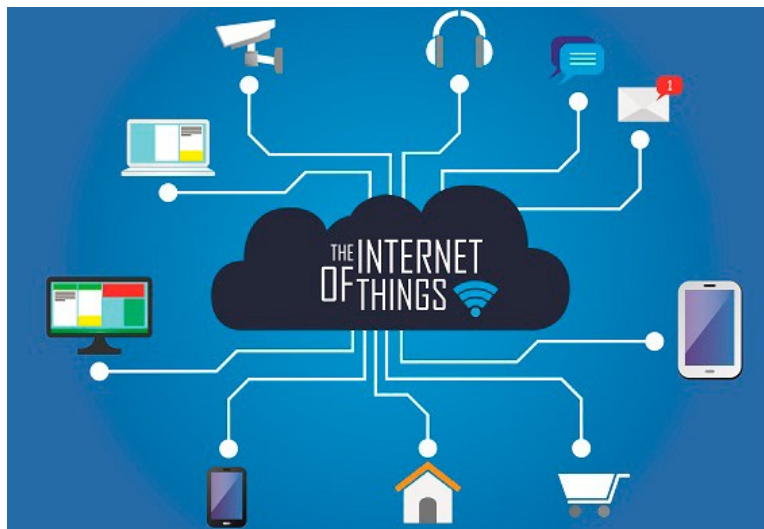
O Java API Framework fornece diversos instrumentos para o desenvolvimento de aplicativos em Java para Android, por exemplo, gerenciador de notificação e um rico sistema de visualização para criação de botões, caixa de texto e outros (ANDROID, 2018).

Aplicativos do sistema, são aplicativos nativos que já vem instalados no sistema operacional como calendário e navegador de internet. Estes aplicativos não tem status especial portanto permitindo um aplicativo terceirizado substituir ele como principal no sistema (ANDROID, 2018).

## 2.2 Internet das Coisas

Conforme explica Waher (2015), internet das coisas ou em inglês *Internet of Things* (IoT) é o nome que se dá à dispositivos eletrônicos não operados por humanos conectados na Internet conforme demonstrado na Figura 2. Segundo Santos e Freitas (2016), IoT é um dos assuntos de tecnologia mais comentados nos últimos anos e o mercado em cima do mesmo vem crescendo muito.

Figura 2: Internet das coisas em casa



Fonte: Max Dicas (2018)

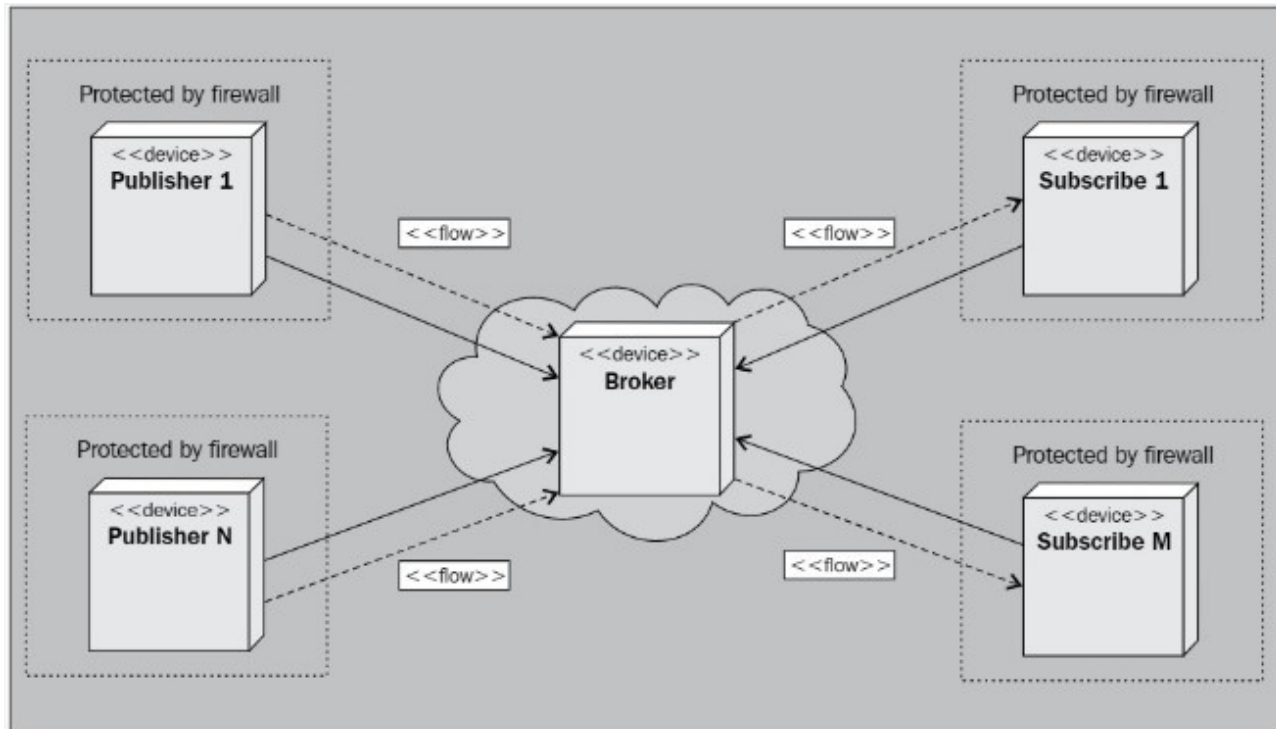
O termo IoT foi utilizado pela primeira vez por Kevin Ashton, pesquisador do Massachusetts Institute of Technology (MIT), em 1999, onde ele relata que gostaria de expressar algo como o encontro de duas redes diferentes: a rede de comunicação humana e o mundo real das coisas. Segundo Santos e Freitas (2016) o termo se popularizou em 2010 e a previsão é que para 2022 exista quase 30 bilhões de dispositivos conectados (COSTA, 2019).

### 2.2.1 Protocolo para troca de Mensagens MQTT

Segundo Waher (2015), o protocolo MQTT é um protocolo baseado em fila e permite

componentes sendo eles o *Publisher* que envia a mensagem ao *Broker*, esta mensagem pode ser formatada em qualquer formato de dados para carga útil. O *Broker* que é o servidor, fica responsável pela fila e também administra os *Publishers* e *Subscribers*, sendo cargo dele receber a mensagem do *Publisher* e enviar a mensagem ao *Subscriber* que por fim recebe a informação conforme demonstrado na Figura 3.

Figura 3: Mensagem MQTT



Fonte: [Waher \(2015\)](#)

[Yuan \(2017\)](#) explica em seu manual que MQTT é leve e permite implementação em dispositivos simples e locais onde a conexão é limitada, e também que o protocolo é flexível, possibilitando suporte a diversos cenários, sendo então um equilíbrio ideal para utilização em IoT. Na próxima etapa será apresentadas implementações do MQTT.

#### 2.2.1.1 Eclipse Mosquitto

O Eclipse Mosquitto é um gerenciador de mensagens (*broker*) da Eclipse Foundation, ele funciona baseado em protocolo MQTT. É um sistema open source leve, indicado para computadores de baixo processamento até servidores completos. O Mosquitto fornece uma instância de testes em um servidor online facilitando o teste de projetos, e uma comunidade ativa tanto para relatar problemas ou discutir sobre projetos e dúvidas ([ECLIPSE, 2019](#)).

#### 2.2.1.2 Eclipse Paho

O projeto Eclipse Paho faz parte de uma iniciativa na Eclipse Foundation que visa fornecer ferramentas e protocolos de código aberto para simplificar o desenvolvimento usando o MQTT. O projeto Eclipse Paho fornece um conjunto de bibliotecas que facilitam o desenvolvimento de

## 2.3 Banco de Dados

O Armazenamento de dados é uma parte importante dos *softwares*, até aplicações mais básicas necessitam salvar dados e acessá-los posteriormente. No passado cada aplicação realizava a administração dessas informações por conta própria, consumindo muito tempo e produtividade dos desenvolvedores, este problema foi amenizado com o surgimento do Sistema de Gerenciamento de Banco de Dados (SGBD) que consiste em realizar a administração do Banco de Dados (DEITEL, 2015) (DEVMEDIA, 2011). A seguir será apresentado sobre o Firebase.

### 2.3.1 Firebase Realtime Database

O Firebase Realtime Database é um banco de dados não relacional desenvolvido pela Google que armazena os dados como JSON conforme representado na Figura 4 e tem eles sincronizados em nuvem simultâneamente, portanto, garantindo que os dados estarão atualizados para qualquer usuário conectado. Caso não haja conexão com a internet no momento, os dados serão salvos em cache no dispositivo aguardando até a próxima conexão (GOOGLE, 2019).

Figura 4: Estrutura de base de dados do firebase



Fonte: elaborada pelo autor (2020)

## 2.4 Trabalhos correlatos

### 2.4.1 Google Home

Google home, representado na Figura 5, é um dispositivo existente no mercado que já obteve mais de 24 milhões de vendas nos Estados Unidos (UOL, 2018). Este que custa a partir de US\$39 nos EUA em sua versão mini, tem a aparência de uma caixa de som e interage com voz como um assistente particular. Pode se comunicar com dispositivos dentro de casa por tecnologia bluetooth ou wifi quando estes são previamente preparados para comunicação (GOOGLE, 2020).

Figura 5: Google Home Mini



Fonte: UOL (2018)

### 2.4.2 Amazon Alexa

Amazon Echo, representado na Figura 6 conta com a assistente Alexa, este que já teve mais de 100 milhões de produtos vendidos no mundo, é muito semelhante ao Google Home e tem o objetivo de ser uma assistente particular, ela pode controlar diversos eletrodomésticos e dispositivos compatíveis via bluetooth ou wifi. Sua versão mais barata pode ser encontrada a

Figura 6: Amazon Echo



Fonte: [Ribeiro \(2017\)](#)

#### 2.4.3 MQTT Dash

O aplicativo MQTT Dash que está disponível gratuitamente na Play Store, foi desenvolvido pela Routix *software* e tem o objetivo de fornecer soluções simples de comunicação MQTT oferecendo um painel de indicadores ou painel de bordo com possibilidade de inserir ícones e separar os dispositivos em blocos ([PEREIRA, 2018](#)). O MQTT Dash não oferece a opção de salvar os dados inseridos na nuvem para que possam ser resgatados em outro momento ou em outro *smartphone*, oferece uma grande liberdade para realizar configurações mas não é de código aberto para que o usuário possa realizar adaptações no aplicativo quando necessário ([PEREIRA, 2018](#)).

#### 2.4.4 IoT MQTT Panel

O aplicativo IoT MQTT Panel que está disponível em versão gratuita e paga na Play Store, foi desenvolvido pela Rahul Kundu e fornece soluções de forma intuitiva para comunicação MQTT com painéis e botões personalizados ([RAHUL KUNDU, 2017](#)). O IoT MQTT Panel oferece a opção de exportar e importar arquivo com as parametrizações porém não tem a possibilidade de realizar o salvamento em nuvem, oferece também assim como o MQTT Dash uma liberdade de configurações mas também não possui código aberto ([RAHUL KUNDU, 2017](#)).

#### 2.4.5 Comparativo

O Google Home e a Alexa da Amazon são semelhantes pois estes tem finalidade comercial, é necessário realizar um investimento para a compra de um dispositivo e ambos se propõem a ser um assistente pessoal por comandos de voz. O MQTT Dash e o IoT MQTT Panel são gratuitos e podem ser instalados em um *smartphone* com Android, estes funcionam apenas com comunicação MQTT de forma semelhante a este projeto porém nenhum destes oferece suporte



## **2.5 Resumo do Capítulo**

Neste capítulo foram levantadas informações técnicas que serão utilizadas no desenvolvimento do projeto como o protocolo MQTT, que é o protocolo de troca de mensagens que será utilizado. Também foram vistos conceitos sobre banco de dados e trabalhos correlatos, sendo eles o Google Home, Amazon Alexa, MQTT Dash e IoT MQTT Panel.



### 3 MATERIAIS E MÉTODOS

Neste capítulo será apresentada uma visão geral do trabalho, e então serão descritos os requisitos funcionais, requisitos não funcionais e Diagramas UML (caso de uso, atividade, classe e sequência) da aplicação.

#### 3.1 Visão Geral do projeto

Este trabalho se propôs em fornecer um aplicativo Android que realize conexão com dispositivos IoT utilizando o protocolo MQTT. O usuário tem acesso a todos os dispositivos adicionados podendo optar por um para realizar comandos nele ou verificar o status, cada dispositivo adicionado é tratado como uma conexão. Será necessário realizar um cadastro para salvar as informações e para sincronizar os dados em nuvem, para salvar os dados e realizar a autenticação do login será utilizado os serviços do Google Firebase. O software está padronizado para receber traduções e já tem inglês e português adicionados nos seus arquivos.

#### 3.2 Requisitos funcionais

Na Tabela 1 são apresentados os requisitos funcionais do projeto.

Tabela 1: Tabela de requisitos funcionais

ID	Descrição	Detalhamento
RF01	Incluir nova conexão.	O sistema deve incluir nova conexão pelo gerenciamento de conexões.
RF02	Editar uma conexão.	O sistema deve permitir editar uma conexão pelo gerenciamento de conexões.
RF03	Excluir uma conexão.	O sistema deve excluir uma conexão pelo gerenciamento de conexões.
RF04	Visualizar todas as conexões.	O sistema deve apresentar todas as conexões na tela inicial.
RF05	Enviar comandos para o dispositivo.	O sistema deve enviar comandos para o dispositivo através do broker.
RF06	Recebimento de informações do status dos dispositivos.	O sistema deve receber informações do status dos dispositivos quando estiver com a tela de dispositivo aberta.
RF07	Apresentar ao usuário status do dispositivo.	O sistema deve apresentar ao usuário status do dispositivo quando aberta tela de visualização do dispositivo.

Fonte: elaborada pelo autor (2019)

#### 3.3 Requisitos não funcionais

Tabela 2: Tabela de requisitos não funcionais

ID	Descrição	Detalhamento
RNF01	Desenvolvimento em Java com as bibliotecas do Android.	Sistema deve ser implementado utilizando Java com as bibliotecas Android.
RNF02	Utilização do Firebase para o armazenamento de dados.	Sistema utilizará banco de dados Firebase para salvar informações localmente e em nuvem.
RNF03	Utilização do Eclipse Paho.	Comunicação com os dispositivos via MQTT utilizando bibliotecas do Eclipse Paho.

Fonte: elaborada pelo autor (2019)

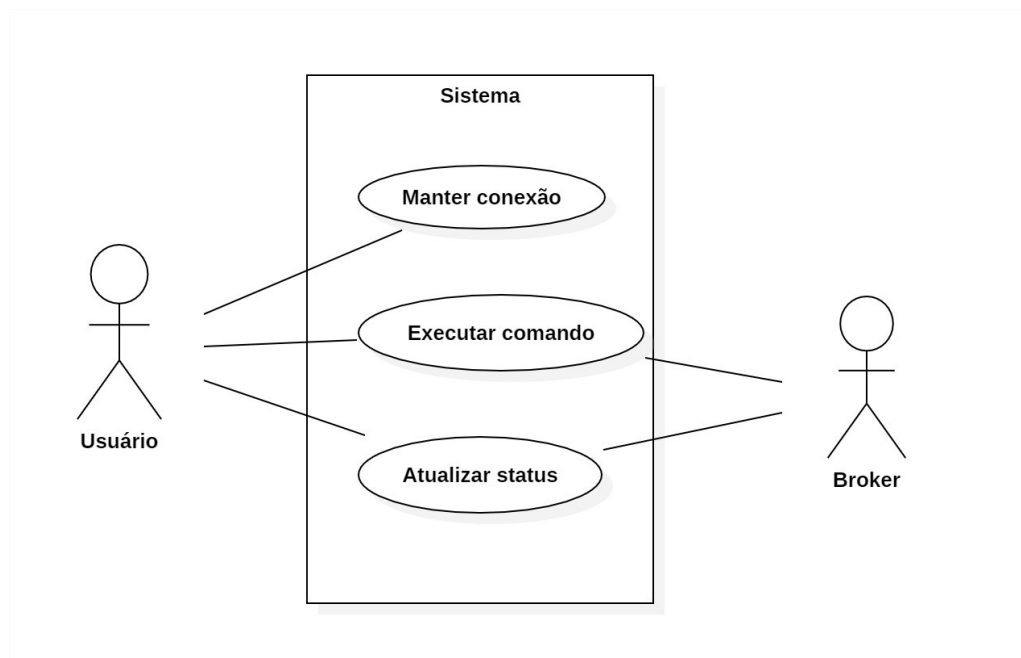
### 3.4 Diagramas

Nessa seção são apresentados os diagramas de caso de uso, classe, sequência e atividades referentes a este projeto.

#### 3.4.1 Casos de uso

Na Figura 7 é apresentado o diagrama de casos de uso e logo abaixo a descrição de cada caso.

Figura 7: Diagrama de casos de uso do aplicativo



Fonte: elaborado pelo autor (2019)

### 1. Manter conexão:

- Incluir conexão:
  - a) Usuário clica no botão “Nova conexão”;
  - b) Sistema solicita parâmetros de conexão;
  - c) Usuário completa os campos solicitados e salvar;
  - d) Sistema adiciona na tela principal a nova conexão;
- Editar conexão:
  - a) Usuário clica na conexão para abrir;
  - b) Sistema apresenta tela com informações da conexão;
  - c) Usuário clica em editar;
  - d) Sistema abre tela de edição da conexão;
  - e) Usuário altera informações e salvar;
  - f) Sistema salva informações e volta pra tela inicial;
- Apagar conexão:
  - a) Usuário clica na conexão para abrir;
  - b) Sistema apresenta tela com informações da conexão;
  - c) Usuário clica em apagar;
  - d) Sistema solicita confirmação;
  - e) Usuário confirma;
  - f) Sistema apaga conexão e volta pra tela inicial;

### 2. Realizar comando:

- Fluxo principal:
  - a) Sistema apresenta a tela do dispositivo;
  - b) Usuário clica na conexão para abrir;
  - c) Sistema apresenta tela com informações da conexão;
  - d) Usuário clica no comando a ser realizado;
  - e) Sistema atualiza status informando estado atual do dispositivo;
- Fluxo alternativo - erro retornado:
  - a) Sistema apresenta na tela inicial as conexões adicionadas;
  - b) Usuário clica na conexão para abrir;
  - c) Sistema apresenta tela com informações da conexão;
  - d) Usuário clica no comando a ser realizado;
  - e) Sistema não atualiza status e apresenta na tela o erro;

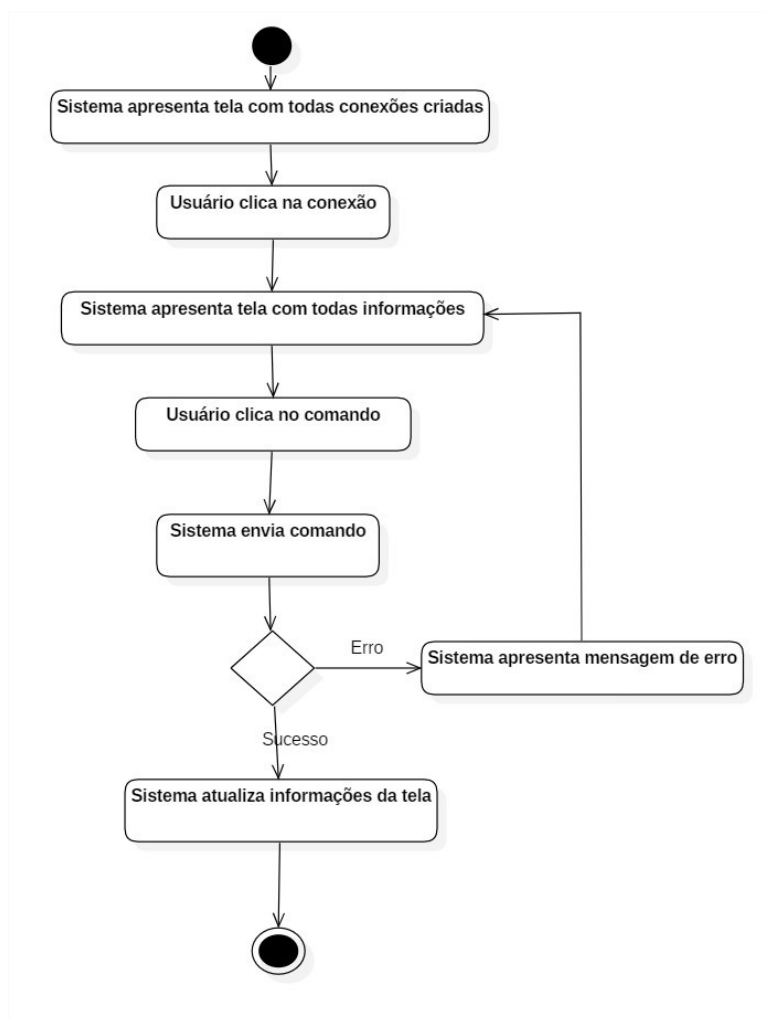
### 3. Atualizar status:

- a) Na tela do dispositivo o Usuário clica em “Atualizar”;
  - b) Sistema solicita informações de status e atualiza a tela;
- Fluxo alternativo - erro retornado:
    - a) Na tela do dispositivo o Usuário clica em “Atualizar”;
    - b) Sistema informa erro retornado;

### 3.4.2 Diagrama de atividades

No diagrama de atividades é apresentado o fluxo desde quando o usuário abre o dispositivo até receber um retorno do mesmo representado na Figura 8.

Figura 8: Diagrama de atividade

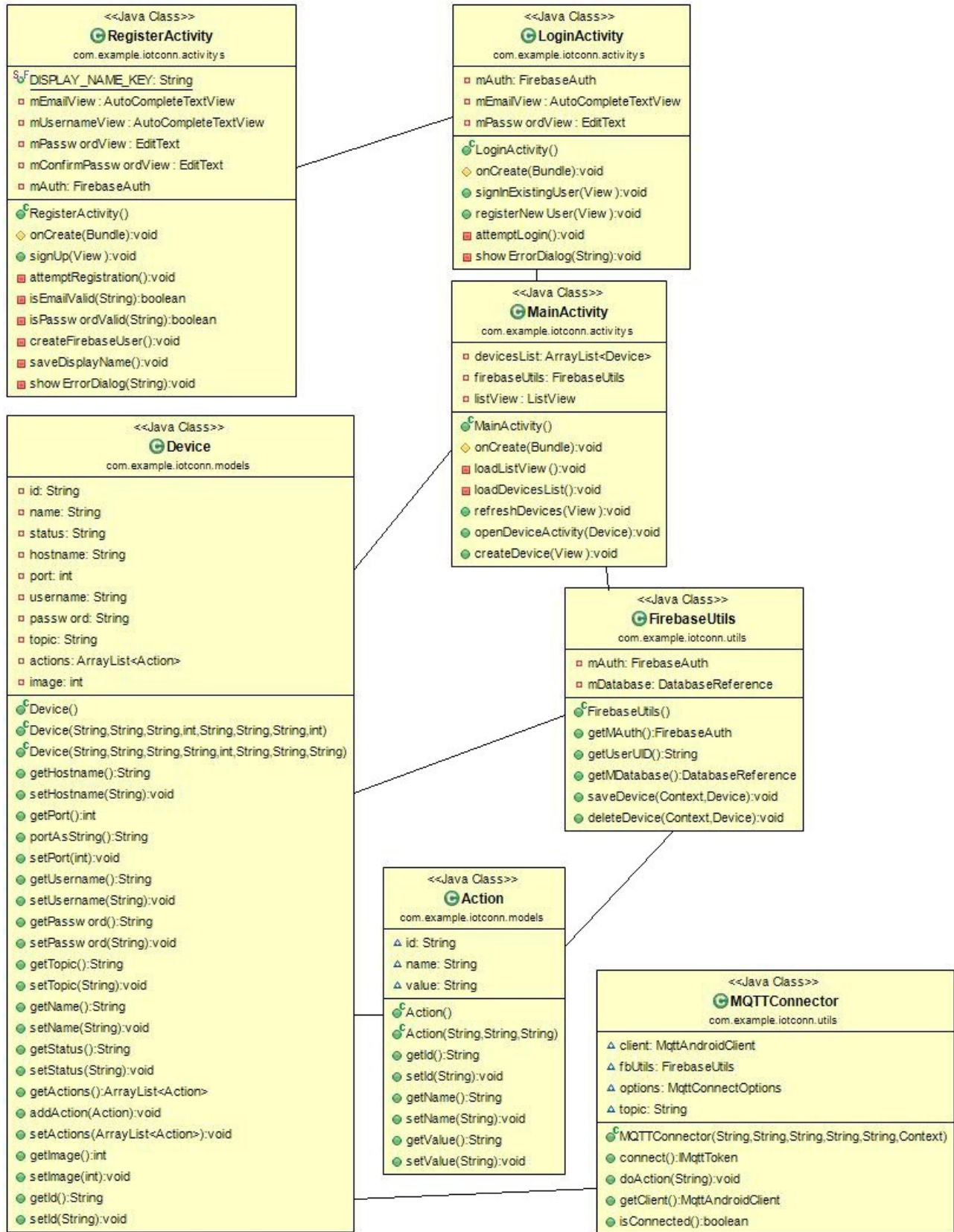


Fonte: elaborado pelo autor (2019)

### 3.4.3 Diagrama de classes

No diagrama de classes da Figura 9 é demonstrado as classes e relacionamentos entre elas. A classe principal carrega todos os dispositivos, e ao clicar em um deles é aberta a tela do mesmo com as informações e opções de comandos a serem realizados. Ao clicar em uma ação será enviado para a classe de conexão com o MQTT os dados de conexão do dispositivo

Figura 9: Diagrama de classes

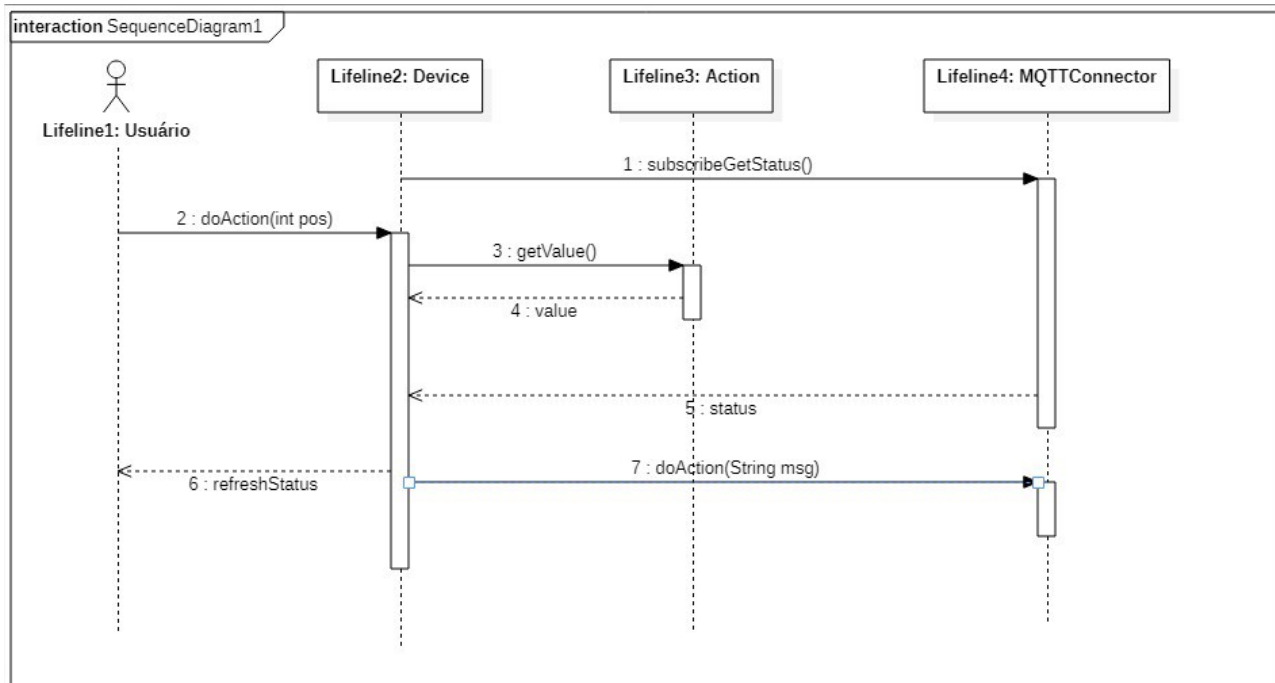


Fonte: elaborado pelo autor (2019)

### 3.4.4 Diagrama de sequência

No diagrama de sequência da Figura 10 é apresentado o fluxo do usuário selecionando um comando, este comando que é um botão de ação dentro da tela do dispositivo, até a resposta do dispositivo.

Figura 10: Diagrama de sequência



Fonte: elaborado pelo autor (2019)

## 3.5 Resumo do Capítulo

Neste capítulo foi apresentado uma visão geral da proposta e seus requisitos (funcionais e não funcionais), em seguida foi apresentado diagramas mapeando principais características do projeto. No capítulo a seguir serão apresentados os resultados obtidos levando em consideração a proposta e os objetivos.



## 4 RESULTADOS OBTIDOS

Foi desenvolvido um aplicativo que consegue realizar comunicação do *smartphone* com dispositivos utilizando o protocolo de comunicação MQTT evitando que *makers* tenham que se preocupar com esta parte da conexão de seus projetos. O sistema não necessita obrigatoriamente conhecimentos de programação para que o usuário possa utilizá-lo, mas também tem seu código fonte disponibilizado no GitHub<sup>1</sup>, para que entusiastas possam realizar suas alterações e incrementos. A seguir será apresentado algumas imagens do aplicativo em funcionamento:

### 4.1 Imagens

Nas próximas etapas serão apresentadas imagens do sistema.

#### 4.1.1 Telas do aplicativo

Na Figura 11a é apresentada a tela de login contendo também o botão que leva para a tela de registro, para criar um novo usuário. A Figura 11b apresenta a tela de registro permitindo ao usuário criar um novo cadastro. A tela inicial contendo todos os dispositivos adicionados, botão para adicionar novos dispositivos e botão para atualizar está na Figura 11c. Na Figura 11d está a tela de criar ou editar o dispositivo que é utilizada a mesma tela, porém quando em edição a mesma vem carregada com as informações do dispositivo. A Figura 11e apresenta a tela do dispositivo contendo as informações dele, botões de comando e opção para editar dispositivo, atualizar, deletar ou adicionar novos botões de comando. É apresentada na figura 11f a tela de criar um comando, nela existem apenas dois campos, o nome do comando e o valor que será enviado ao dispositivo. Na Figura 11g é apresentada a mensagem de confirmação para deletar um comando, que é apresentada ao dar um clique longo no comando. A Figura 11h apresenta a tela de confirmação para deletar um dispositivo, que é apresentada ao clicar em deletar no canto esquerdo inferior da tela do dispositivo.

#### 4.1.2 Comunicação MQTT

Na Figura 12 é demonstrado um teste realizado, onde foi enviada uma mensagem no *topic* "temp" e recebido uma resposta no *topic* "satus\_temp". Para realizar a comunicação e ter uma visualização melhor da comunicação foi utilizado o projeto *open source* MQTT Lens<sup>2</sup>, que é uma extensão para o Google Chrome que cria uma interface gráfica para trabalhar com o MQTT.

---

<sup>1</sup> Disponível em: <https://github.com/tailoralm/IoTConn>

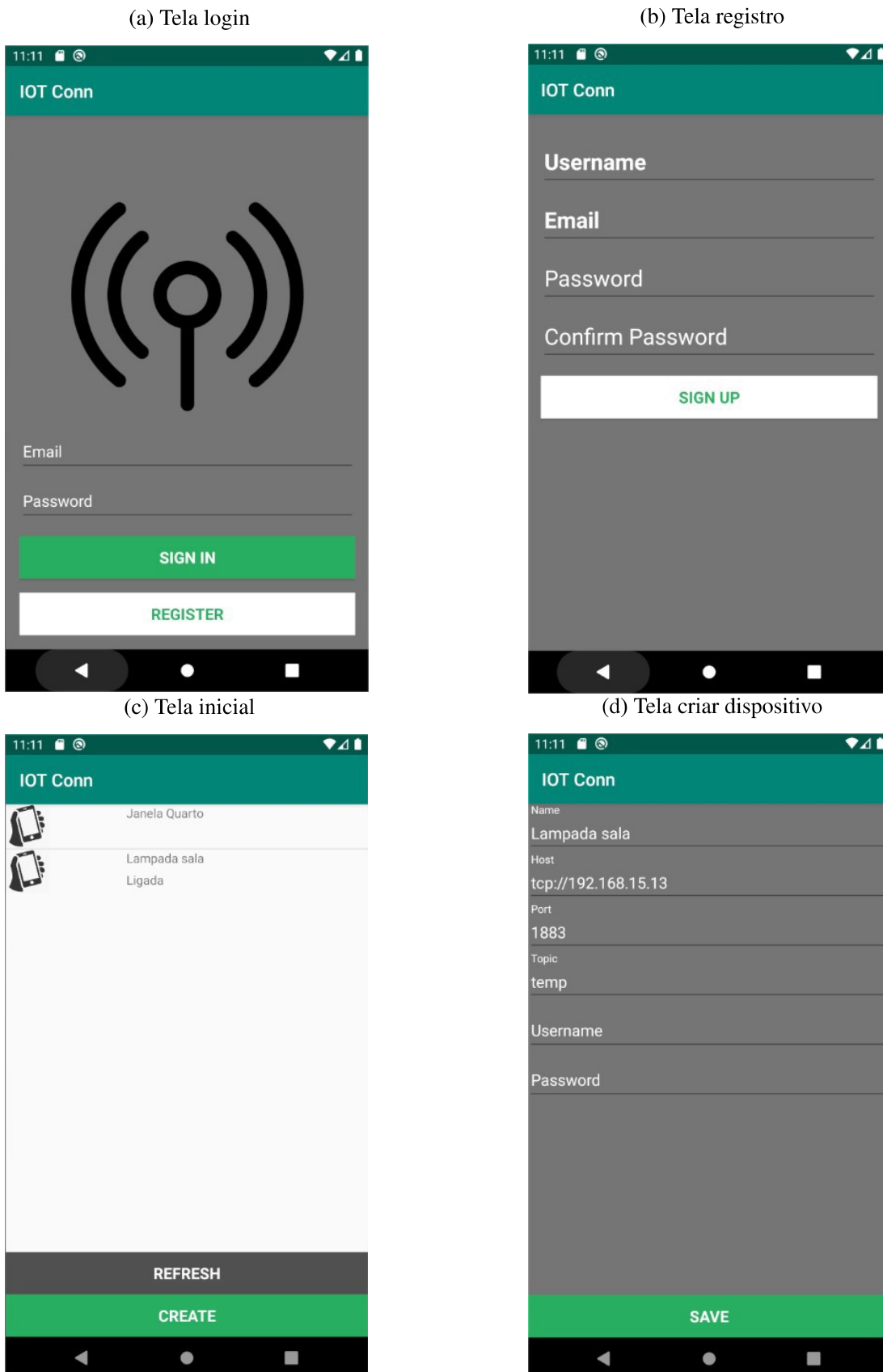
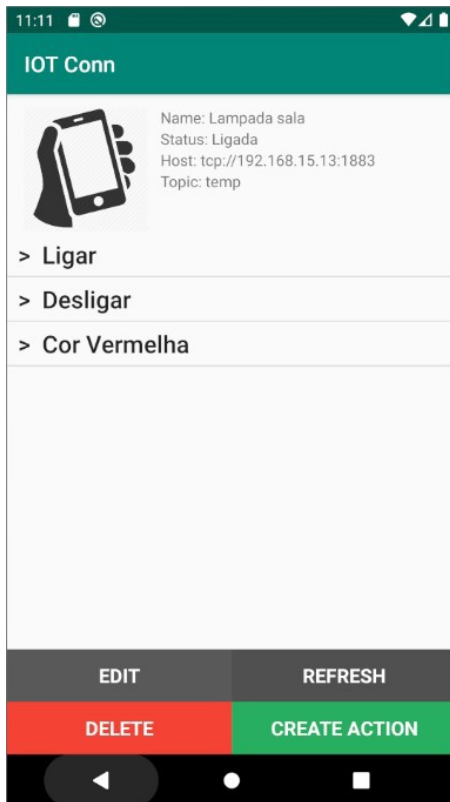


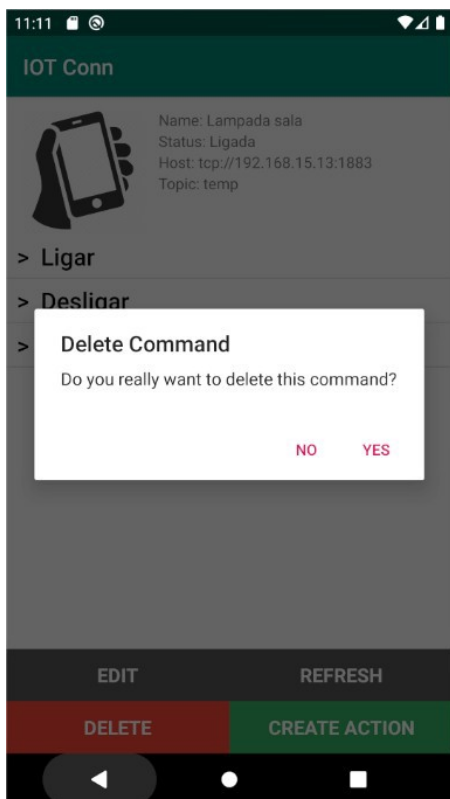
Figura 11: Telas do sistema

Fonte: elaborada pelo autor (2020)

(e) Tela dispositivo



(g) Tela deletar botão



(f) Tela criar comando



(h) Tela deletar dispositivo

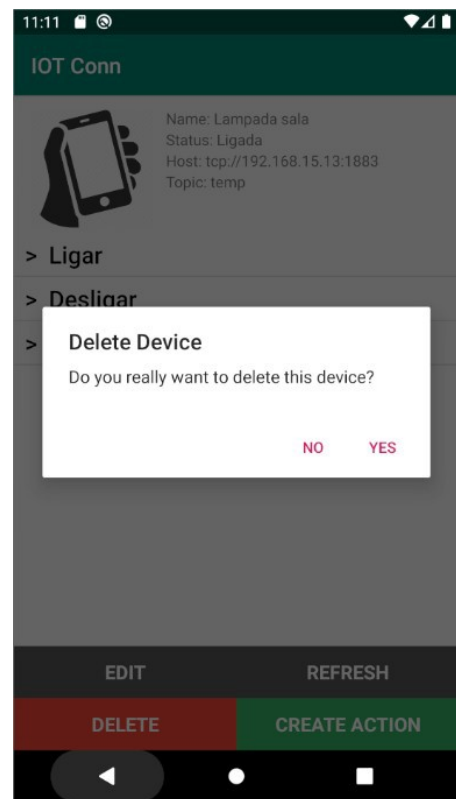
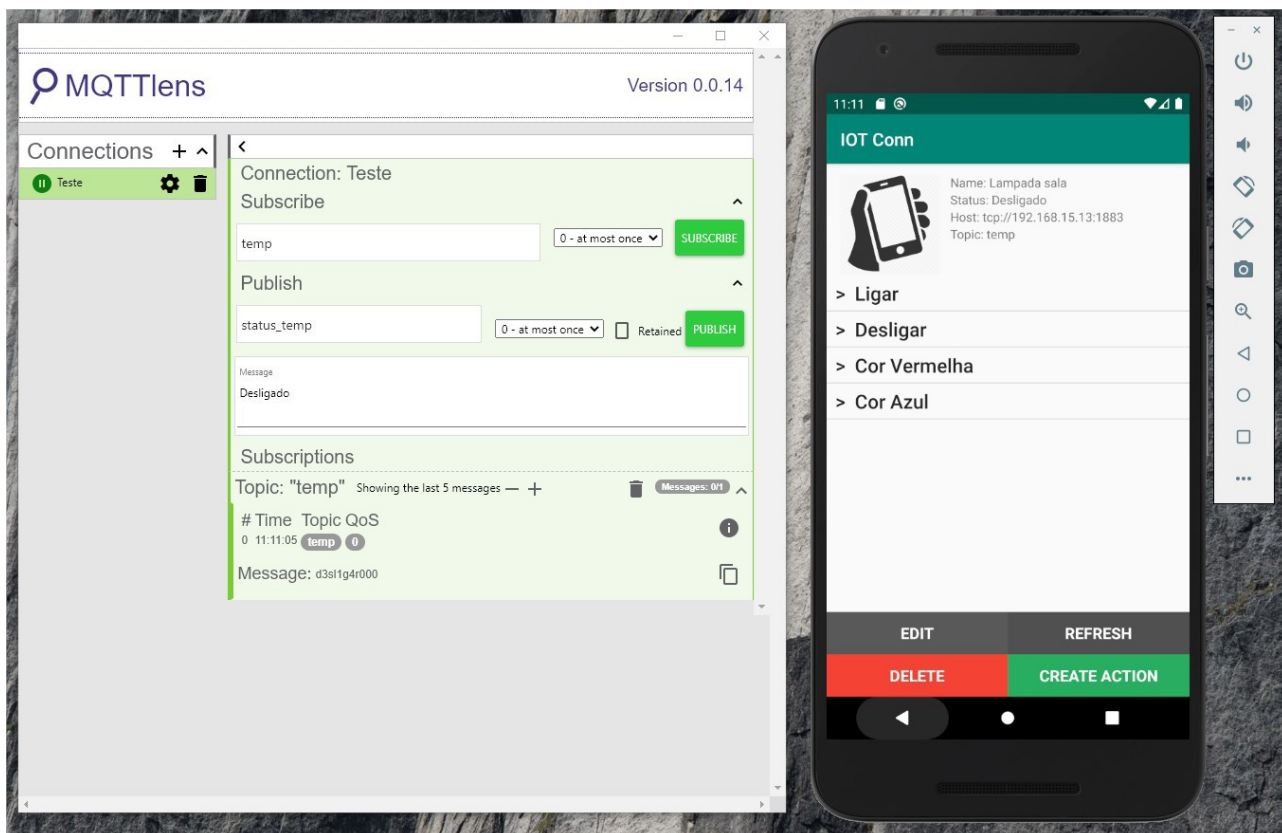


Figura 11: Telas do sistema

Fonte: elaborada pelo autor (2020)

Figura 12: Comunicação efetuada



Fonte: elaborado pelo autor (2020)

## 5 CONCLUSÃO

Foi possível notar que desde 2003 existe uma onda crescente de dispositivos conectados, de 2003 a 2020 existe uma perspectiva de um aumento de 0,07 para 6,5 dispositivos conectados por pessoa. Junto com esse número existe também uma tendência global de automatizar todas as funções do dia a dia, existindo já no mercado alguns dispositivos como Google Home e Alexa.

Este projeto teve como foco desenvolver um aplicativo Android de suporte para pessoas que desejam trabalhar com *hardware*, fornecendo ferramentas para realizar a conexão via protocolo MQTT. Os objetivos se caracterizam em fornecer para criadores um software que realize a comunicação com o *broker* via protocolo MQTT e que possa ser utilizado por quem não tem conhecimento de programação para Android, porém também ter seu código fonte disponibilizado na comunidade para que possa ser utilizado em projetos futuros, além de permitir a criação de usuários para salvar as informações. Os objetivos foram todos concluídos, com a utilização do Eclipse Paho foi possível realizar a comunicação via MQTT e utilizando o Firebase foi possível realizar autenticação de usuário e salvar os dados em nuvem. O projeto está disponível de forma pública no Github, onde quem tiver interesse pode realizar contribuições ou clonar para utilizar de base em seus projetos.

### 5.1 Trabalhos Futuros

Fica a principal sugestão para trabalhos futuros a inclusão de comandos por voz, podendo utilizar APIs como por exemplo o Watson da IBM. Sugere-se também o aprimoramento do aplicativo incluindo mais opções de tecnologias como outros protocolos de conexão e comunicação bluetooth. Por fim, sugere-se a criação de um leiaute aprimorado focando na inclusão de pessoas com dificuldades no contato com smartphones, como por exemplo deficientes visuais.



## REFERÊNCIAS

- AMAZON. *O que é Alexa?* 2020. Disponível em: <<https://developer.amazon.com/pt-BR/alexa>>. Acesso em: 30 jun. 2020. 15, 21
- ANDROID. *Arquitetura da plataforma*. 2018. Disponível em: <<https://developer.android.com/guide/platform/?hl=pt-br>>. Acesso em: 14 out. 2018. 17, 18
- BRANDON, R. *There are now 2.5 billion active Android devices*. 2019. Disponível em: <<https://www.theverge.com/2019/5/7/18528297/google-io-2019-android-devices-play-store-total-number-statistic-keynote>>. Acesso em: 06 mai. 2020. 17
- COSTA, M. d. F. N. *Internet das coisas: a proteção da privacidade em um mundo conectado*. Escola Nacional de Administração Pública (Enap) and Instituto Serzedello Corrêa (ISC), 2019. Disponível em: <<https://repositorio.enap.gov.br/bitstream/1/4125/1/MANUELLA%20DE%20FARIAS%20NARDELLI%20COSTA.pdf>>. Acesso em: 30 jun. 2020. 15, 16, 18
- DEITEL, P. J. *Android para programadores: uma abordagem baseada em aplicativos*. Porto Alegre: Bookman, 2015. 17, 20
- DEV MEDIA. *SQLite no Android*. 2011. Disponível em: <<https://www.devmedia.com.br/sqlite-no-android/19201>>. Acesso em: 21 out. 2019. 20
- DIOLINUX. *O projeto que quer integrar Apps de Android nas distros Linux de Desktop*. 2017. Disponível em: <<https://www.diolinux.com.br/2017/04/anbox-app-android-linux-desktop.html>>. Acesso em: 30 out. 2019. 17
- ECLIPSE. *Eclipse Mosquitto*. 2019. Disponível em: <<https://mosquitto.org/>>. Acesso em: 05 out. 2019. 19
- ECLIPSE. *Eclipse Paho*. 2020. Disponível em: <<https://wiki.eclipse.org/Paho>>. Acesso em: 06 mai. 2020. 19
- GOOGLE. *Firestore Realtime Database*. 2019. Disponível em: <<https://firebase.google.com/docs/database/?hl=pt-br>>. Acesso em: 09 out. 2019. 20
- GOOGLE. *Speakers with the Google Assistant*. 2020. Disponível em: <<https://assistant.google.com/platforms/speakers/>>. Acesso em: 30 jun. 2020. 15, 21
- MAX DICAS. *O que é Internet das Coisas?* 2018. Disponível em: <<https://www.canalmaxdicas.com.br/2018/05/internet-das-coisas.html>>. Acesso em: 08 out. 2018. 18
- PEREIRA, L. H. J. Monografia de graduação em engenharia de controle e automação. Universidade Federal de Ouro Preto, 2018. Disponível em: <[http://www.monografias.ufop.br/bitstream/35400000/922/1/MONOGRRAFIA\\_MoniotramentoConsumoEnergia.pdf](http://www.monografias.ufop.br/bitstream/35400000/922/1/MONOGRRAFIA_MoniotramentoConsumoEnergia.pdf)>. Acesso em: 22 out. 2018. 22
- PINTO, D. d. O. *O que é cultura maker e qual sua importância na educação?* 2018. Disponível em: <<https://blog.lyceum.com.br/o-que-e-cultura-maker/>>. Acesso em: 28 jun. 2020. 15
- RAHUL KUNDU. *IOT MQTT PANEL FAQ AND FEEDBACK*. 2017. Disponível em: <<http://www.snrelectronicsblog.com/iot/iot-mqtt-panel-user-guide/>>. Acesso em: 05 dez. 2019.

RIBEIRO, G. F. *Você vai falar com os eletrônicos da sua casa, e quem vai responder é Alexa*. 2017. Disponível em: <<https://tecnologia.uol.com.br/noticias/redacao/2017/01/09/voce-sabe-o-que-e-alexa-assistente-pessoal-pode-invadir-sua-vida-em-breve.htm>>. Acesso em: 20 ago. 2018. 22

SANTOS, D. d. O.; FREITAS, E. B. A internet das coisas e o big data inovando os negócios. *Revista Fatec Zona Sul*, Faculdade de Tecnologia da Zona Sul, v. 3, n. 1, 2016. Disponível em: <<https://doaj.org/article/6ccb87b61a3f485f9d0cfe3075be2822?frbrVersion=2>>. Acesso em: 21 mai. 2018. 15, 16, 18

SILVA, L. J. Trabalho de conclusão de curso: internet das coisas. Universidade do Sul de Santa Catarina, 2017. Disponível em: <<https://riuni.unisul.br/bitstream/handle/12345/3940-/TCC%20FINAL%20LEANDRO%20JAMIR%20SILVA.pdf?sequence=1&isAllowed=y>>. Acesso em: 03 set. 2018. 15

UOL. *Um Google Home é vendido por segundo; por que faz tanto sucesso?* 2018. Disponível em: <<https://tecnologia.uol.com.br/noticias/redacao/2018/01/11/em-crescimento-google-home-sera-o-proximo-melhor-amigo-da-sua-casa.htm>>. Acesso em: 27 ago. 2018. 21

WAHER, P. *Learning internet of things paperback*. Birmingham Mumbai: Packt Publishing Ltd., 2015. 18, 19

YUAN, M. *Conhecendo o MQTT*. 2017. Disponível em: <<https://www.ibm.com/developerworks-br/library/iot-mqtt-why-good-for-iot/index.html>>. Acesso em: 08 out. 2018. 19