Universidade Federal de Minas Gerais
Instituto de Ciências Exatas
Departamento de Ciência da Computação
Bacharelado em Ciência da Computação

DANIEL REIS SOUZA

MONOGRAFIA DE PROJETO ORIENTADO EM COMPUTAÇÃO II

Aplicativo Android Controlador de dispositivos Internet das Coisas

Belo Horizonte
2019 / 2º semestre

Universidade Federal de Minas Gerais
Instituto de Ciências Exatas
Departamento de Ciência da Computação
Curso de Bacharelado em Ciência da Computação

Aplicativo Android Controlador de dispositivos Internet das Coisas

por

DANIEL REIS SOUZA

Monografia de Projeto Orientado em Computação II

Apresentado como requisito da disciplina de Projeto Orientado Em Computação II do Curso de Bacharelado em Ciência Da Computação da UFMG.

Prof. Dr. *Daniel Fernandes Macedo*
Orientador

Belo Horizonte
2019 / 2º semestre

À minha família,
aos professores,
aos colegas de curso e
aos meus amigos,
dedico este trabalho.

RESUMO

A internet está cada vez mais ubíqua ao nosso redor e os dispositivos computacionais estão cada vez mais potentes e baratos, permitindo que a Internet das Coisas esteja cada vez mais acessível. A Internet das Coisas é um conceito em que dispositivos computacionais inteligentes e conectados, comunicam uns com os outros a fim de atingir algum objetivo em comum.

Neste projeto, o objetivo foi criar um aplicativo Android que permita à entusiastas da Internet das Coisas controlar seus dispositivos inteligentes através do celular.

O aplicativo permite ao usuário controlar dispositivos através de requisições HTTP a uma API REST. O usuário deve implementar o servidor REST e definir os comandos de sua API.

O uso da aplicação consiste em criar dispositivos e definir os comandos disponíveis para cada cada um deles. Um Dispositivo representa um equivalente dispositivo IoT físico, com a url raiz de sua API REST controladora. Para cada dispositivo, pode-se criar Comandos, que são a url específica de um endpoint da API, com headers e corpo de requisição. Cada comando permite a execução de uma requisição HTTP à API REST que fará com que o dispositivo realize uma ação.

Caso a API possua urls com partes dinâmicas, o usuário pode usar trechos de texto especiais, chamados “marcadores de interface”, para gerar interfaces amigáveis que preencham as partes dinâmicas da url de um comando. Pode-se criar um campo de entrada de texto, menus drop-down com opções ou um slider para escolher entre dois valores limites.

Assim, o entusiasta que possui dispositivos inteligentes mas não deseja aprender a desenvolver aplicações android, pode utilizar esta aplicação para controlar seus dispositivos através do seu celular.

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

ABSTRACT

The Internet is increasingly present in our environment, and computing devices are becoming increasingly powerful and cheaper, making the Internet of Things increasingly accessible. Internet of Things is a concept in which smart, connected devices communicate in order to achieve some common goal.

In this project, the goal was to create an Android app that allows IoT enthusiasts to control their smart devices through their mobile phone. To be as compatible as possible, the application allows the user to control devices through HTTP requests to a REST API. The user must implement the REST server and define its API commands. Using the application consists of creating devices and defining the commands available for each of them. A Device represents an equivalent physical IoT device with the root url of its controlling REST API. For each device, you can create Commands, which are the specific url of an API endpoint, with its headers and request body. Each command allows the execution of an HTTP request to the REST API that will cause the device to perform an action.

If the API has urls with dynamic parts, the user can use special text snippets, called “interface markers”, to generate friendly interfaces that fill the dynamic url parts of a command. You can create a text entry field, drop-down menus with options, or a slider to choose between two threshold values.

Thus, the enthusiast who owns smart devices but does not want to learn to develop android applications, can use this application to control their devices through their mobile phone.

Keywords: *Internet, Internet das Coisas, Android*

LISTA DE FIGURAS

Figura 1: Tela Principal da Aplicação. **12**

Figura 2: Tela de Comandos **13**

Figura 3: Tela de Criar/Editar Comando **14**

Figura 4: Tela de Comando **16**

Figura 5: Tela de Resultados **17**

LISTA DE SIGLAS

1. POC - Projeto Orientado em Computação, matéria obrigatória do curso de Ciência da Computação e fonte deste trabalho.
2. IoT - Internet das Coisas. Conceito de computação.
3. HTTP - HyperText Transfer Protocol, protocolo de comunicação pelo qual se navega em boa parte da internet.
4. API - Interface de Programação de Aplicações, Conjunto de rotinas e padrões estabelecidos para a utilização de um serviço.
5. REST - Padrão para criação de uma api através da comunicação HTTP.
6. URLS - Endereço de recursos web em uma rede.
7. NFC - Near-field communication, é uma tecnologia de comunicação por proximidade.
8. IDE - Ambiente de desenvolvimento de software.
9. TCP - Protocolo de comunicação entre dispositivos na internet.
10. SQL - Linguagem de execução de comandos para bancos de dados.
11. ENDPOINT - Url de uma API REST que permite executar uma rotina no servidor.
12. SOCKETS - Protocolo de comunicação entre dispositivos via rede.

SUMÁRIO

1. INTRODUÇÃO **8**

2. CONTEXTUALIZAÇÃO E TRABALHOS RELACIONADOS **9**

3. DESENVOLVIMENTO DO TRABALHO **10**

4. INTERFACE DE PROGRAMAÇÃO DA APLICAÇÃO (API) **11**

4.1. TELA PRINCIPAL **11**

4.2. TELA DE COMANDOS **12**

4.3. TELA DE ADICIONAR/EDITAR COMANDO **13**

4.3.1. MARCADORES DE INTERFACE **14**

4.4. TELA DE COMANDO **15**

4.5 TELA DE RESULTADOS **16**

5. BANCO DE DADOS **16**

5.1. TABELA DISPOSITIVOS **16**

5.2. TABELA COMANDOS **17**

5.3. TABELA HEADERS **17**

5.4. TABELA POSTDATA **17**

6. CONCLUSÃO E TRABALHOS FUTUROS **18**

7. REFERÊNCIAS **19**

8. APÊNDICE **20**

8.1. CÓDIGO FONTE **20**

8.2. DIAGRAMA DO BANCO DE DADOS **20**

1. INTRODUÇÃO

Desde sua invenção a Internet tem revolucionado o mundo que vivemos e nossas interações. Em junho de 2019, estima-se que 58.8% da população mundial são usuários da Internet [1]. Em 2017, 74,9% dos domicílios brasileiros possuíam conexão com a Internet [2]. Além disso, plataformas de desenvolvimento amador como o Arduino e Raspberry Pi tornaram a computação "Internet das Coisas" acessíveis. O raspberry zero é um exemplo de um computador moderno, conectável, com apenas 6cm de largura e custando apenas 5 dólares [3].

Na computação, Internet das Coisas descreve um grupo de dispositivos inteligentes e conectados que comunicam e realizam tarefas. Alguns exemplos de aplicação seriam uma casa inteligente, onde a porta se abre ao perceber a presença NFC de um celular; ou o ar condicionado que é ligado automaticamente quando o dono da casa estiver chegando do trabalho [4].

Neste projeto, o objetivo é desenvolver um aplicativo Android, que permite ao usuário controlar dispositivos IoT através da configuração de URLS de API REST que criará uma interface de usuário para utilizar estas APIS.

2. CONTEXTUALIZAÇÃO E TRABALHOS RELACIONADOS

A Internet das Coisas [4] é um conceito em que um grupo de dispositivos conseguem comunicar entre si para atingir objetivos comuns. A comunicação pode ser feita através de uma ou mais redes, de qualquer tipo. Em geral, o grupo é composto por sensores que comunicam estados a um controlador de dispositivos que reage de acordo com estes estados.

Assim, podemos criar dispositivos inteligentes, que respondem automaticamente a uma mudança de estado no ambiente. Alguns exemplos seriam carros que troquem informações de tráfego, tomadas que se desconectem automaticamente devido à previsão do tempo, entre outros. Portanto, a Internet das Coisas propicia diversas oportunidades de aplicações sociais e econômicas.

Uma API (Interface de Programação de Aplicação) descreve uma interface de interação com um sistema computacional. Uma API REST, define comandos de interação com um sistema através de URLs e o tipo de requisição HTTP.

A heterogeneidade de dispositivos que podem ser aplicados em Internet das Coisas cria a necessidade de um gerenciador inteligente que se adapte às características de cada dispositivo ao controlá-los.

Um trabalho relacionado, é a plataforma *Management for Internet of Things* (ManIoT), que é uma gerenciador de dispositivos IoT que abstrai suas respectivas características individuais e fornece uma API REST unificada que permite o controle destes dispositivos [5].

3. DESENVOLVIMENTO DO TRABALHO

O aplicativo foi desenvolvido na linguagem Kotlin, utilizando a IDE Android Studio. Foi utilizado a biblioteca de networking HTTP Volley, desenvolvida pela google. Ela abstrai o pedido de requisições, requisitos do protocolo TCP, controle de SOCKETS e fornece métodos nativos para executar requisições HTTP. Assim, nos permite lidar com as requisições em alto nível, lidando somente com a URL, corpo da requisição e seus headers.

Para o banco de dados, foi utilizado a implementação nativa SQLITE do Android, através dos métodos da biblioteca Room, da Google. Esta biblioteca fornece uma forma fácil de representação das tabelas do banco de dados como objetos nativos da linguagem. Além disso, ela fornece uma interface para criar um objeto que é capaz de executar as query`s SQL como se fossem métodos nativos.

Através da interface do aplicativo, o usuário poderá criar telas para dispositivos IoT e configurar a API REST de comunicação, sendo criados botões e campos que permitem controlar o dispositivo.

4. INTERFACE DA APLICAÇÃO

A interface do aplicativo é composta de 5 telas principais, pelas quais o usuário pode criar/editar dispositivos, executar um comando e ver o resultado da execução de um comando. As telas de interface são:

4.1. Tela Principal

Na tela principal temos as opções de “adicionar dispositivo”, e a lista de dispositivos cadastrados. Em cada dispositivo, temos os botões de editar e deletar. Um dispositivo representa qualquer API HTTP que forneça uma séries de ENDPOINTS de controle e comunicação. Ao clicar em um dispositivo, navegamos para a tela de comandos, onde é listado todos os comandos disponíveis para aquele dispositivo.

Figura 1: Tela Principal da Aplicação

4.2. Tela de Comandos

Para cada comando temos os botões de editar ou deletar. Ao clicar no botão de símbolo de "mais" ou em um botão editar, é aberto a tela de “Adicionar/Editar Comando”.

Um comando representa uma requisição HTTP a uns dos endpoints fornecidos pela API do dispositivo a qual o comando pertence.

Figura 2: Tela de Comandos

4.3. Tela de Adicionar/Editar Comando

Na tela de “Adicionar/Editar Comando”, podemos nomear o comando, escolher o seu tipo de requisição HTTP, configurar a url de sua requisição e adicionar headers a requisição.

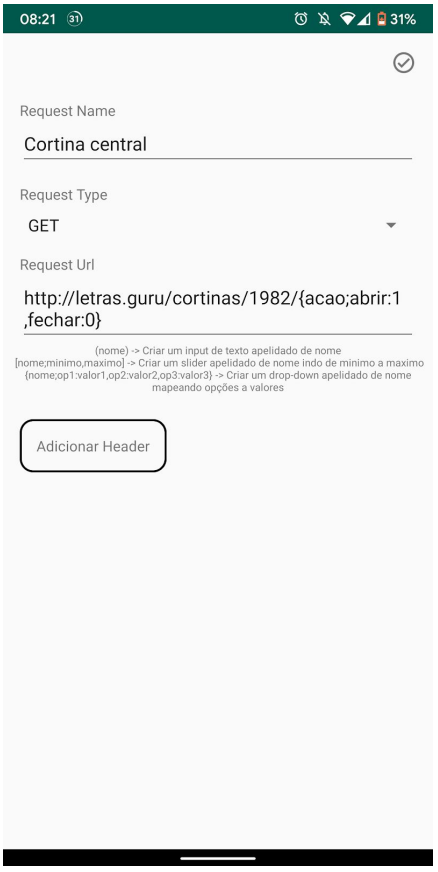

Figura 3: Tela de Criar/Editar Comando

4.3.1. Marcadores de Interface

Comumente, as urls de requisições REST possuem partes dinâmicas usadas para controlar funcionalidades na API. Para facilitar o manuseio de comandos que possuem URL dinâmicas, foi implementado uma pequena linguagem de marcadores de interface com três tipos de marcadores. São eles:

a) Entrada de Texto

Na url de requisição, o texto "(nome)" criará uma interface de entrada de texto com título "nome". O texto escrito nesta entrada substituirá o trecho do marcador na url.

b) Slider

O texto entre aspas "[nome;mínimo,máximo]" criará uma interface de slider de título "nome", cujos valores podem variar entre "mínimo" e "máximo".

c) Menu Drop-Down

O texto entre aspas "[nome;opção1:valor1,opção2:valor2,opção3:valor3]" criará uma interface de menu drop-down de título "nome", esta interface terá as opções "opção1", "opção2" e "opção3" para que o usuário possa escolher. Ao substituir o trecho do marcador na url, será utilizado valor correspondente à opção escolhida.

4.4. Tela de Comando

Ao clicar em um comando, abre-se a “Tela de Comando”, onde é mostrado a URL e os headers que serão incluídos na requisição.

Caso ao criar o comando tenha-se utilizado algum marcador de interface na URL, nesta tela será construído a interface correspondente e o usuário poderá interagir e escolher o valor desejado. Ao executar a requisição, os valores escolhidos pelo usuário substituem os marcadores de interface na URL da requisição e esta é executada, abrindo a “Tela de Resultado”.

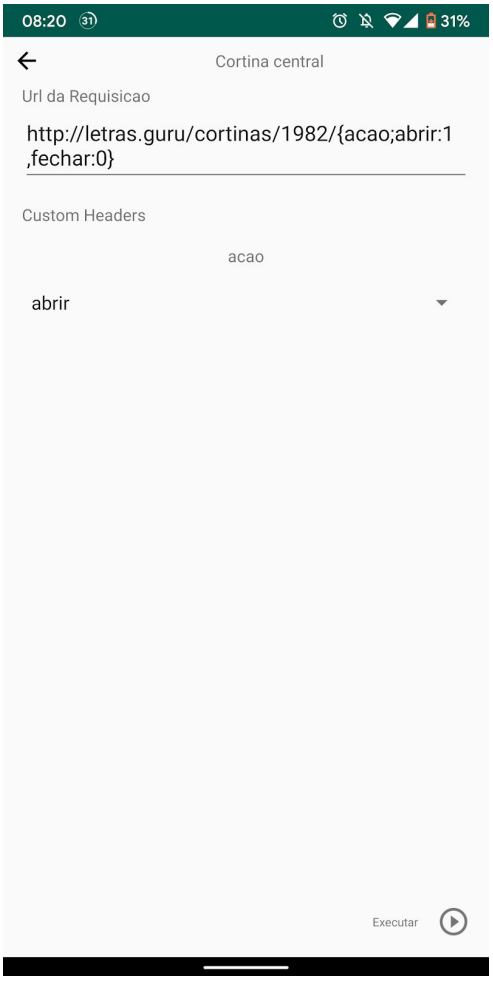

Figura 4: Tela de Comando

4.5. Tela de Resultado

Ao executar um comando, abre-se a tela de resultado, nela é mostrado se a requisição foi executada com sucesso ou não. Também é exibido a URL usada na requisição e o corpo da resposta HTTP recebida de volta.

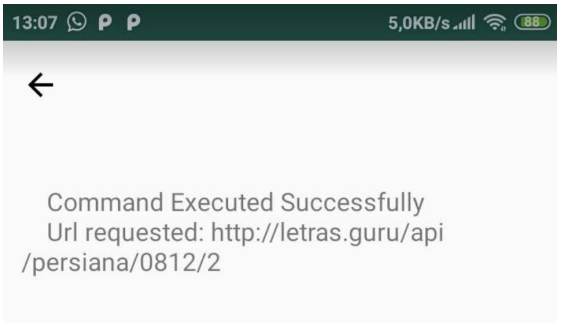

Figura 5: Tela de Resultados

5. BANCO DE DADOS

Para persistir todos os dados da requisição, foi implementado um banco de dados através da biblioteca Room desenvolvida pela Google. O banco de dados utiliza a implementação SQLITE nativa do android. Ele consiste das tabelas Dispositivos, Comandos, Headers e PostData.

O diagrama do banco de dados está desenhado no Apêndice B.

5.1. Tabela Dispositivos

Nesta tabela armazenamos os dados de nome e URL de um dispositivo. Sua chave primária é o nome dado ao dispositivo.

5.2. Tabela Comandos

Esta tabela persiste os dados de um comando, ela possui os campos nome, dispositivo, url e type. O campo nome é a chave primária que identifica um comando, o campo dispositivo é o nome e a chave estrangeira que identifica a qual dispositivo este comando pertence. O campo URL é a url da requisição e o campo type descreve qual tipo de requisição HTTP (GET ou POST) este comando executa.

5.3. Tabela Headers

Esta tabela persiste os dados de todos os headers que pertencem a um comando. A chave primária é um id auto gerado e um header possui o nome de um comando como chave estrangeiras. Os campos nome e valor dão a informação do header, que na requisição são adicionados usando o formato “nome: valor”.

5.4. Tabela PostData

Para um comando do tipo POST, é necessário armazenar também o seu “contentType” e o corpo da requisição. Esta tabela tem um id auto gerado como chave primária e um nome de um dispositivo como chave estrangeiro. O campo “contentType” é um header especial das requisições HTTP POST que informa ao remetente o tipo de estrutura de dados utilizada no corpo da requisição. O campo “body” é o corpo da requisição, escrito pelo usuário ao criar o comando.

6. CONCLUSÕES E TRABALHOS FUTUROS

Com a acessibilidade de microcomputadores e outros componentes, entusiastas podem criar dispositivos inteligentes e usá-los para qualquer fim. A ubiquidade da internet permite que estes dispositivos facilmente sejam controlados e que eles comuniquem entre si. Assim, a ‘Internet das Coisas’’ é uma realidade e está cada vez mais presente na nossa vida.

Para os entusiastas que não desejam investir em soluções comerciais e gostariam de controlar diferentes tipos de dispositivos, o aplicativo desenvolvido é uma boa solução para fornecer uma interface móvel de controle, bastando apenas possuir um telefone android. Sua meta-interface permite aos entusiastas configurar os endpoints de requisições em interfaces mais amigáveis e facilita o controle de seus dispositivos.

Como trabalho futuro, pode-se expandir a capacidade da meta-linguagem de “marcação de interface”, adicionando novas formas de controle e configuração. Além disso, poderia implementar o suporte à linguagem de descrição de dispositivos IoT da Samsung, a mesma usada para configurar e controlar os dispositivos Samsung SmartThings. Assim, um desenvolvedor acostumado com esta linguagem poderia usar facilmente o aplicativo, além de oferecer compatibilidade com estes dispositivos.

7. REFERÊNCIAS

[1] INTERNET WORLD STATS. **World internet users and 2017 population stats**, junho de 2019. Disponível em: <http://www.internetworldstats.com/stats.htm>. Acesso em: 01 de setembro de 2019.

[2] IBGE. **PNAD Contínua TIC 2017**, Dezembro de 2018. Disponível em: <https://agenciadenoticias.ibge.gov.br/agencia-sala-de-imprensa/2013-agencia-de-noticias/releases/23445-pnad-continua-tic-2017-internet-chega-a-tres-em-cada-quatro-domicilios-do-pais >. Acesso em: 02 de Setembro de 2019

[3] The Raspberry Pi Foundation. **Raspberry Pi Zero**, Disponível em: <https://www.raspberrypi.org/products/raspberry-pi-zero/>. Acessado em: 02 de Setembro de 2019

[4] Atzori, Luigi, Antonio Iera, and Giacomo Morabito. "The internet of things: A survey." *Computer networks* 54.15 (2010): 2787-2805.

[5] Antunes, J. B., Dénes, I. L., Santos, M., Castro, T. O., Macedo, D. F., & dos Santos, A. L. (2016). ManIoT: Uma PLataforma para Gerenciamento de Dispositivos da Internet das Coisas. *Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC)*, 14.

8. Apêndice

8.1. Código Fonte

O código fonte do aplicativo está disponível no GitHub no link https://github.com/danielrval/iothub sob licensa MIT.

8.2. Diagrama do Banco de Dados