

Computação Móvel Multisensorial

Marco Rodrigues

Mestrado em Engenharia de Software

2016-2017

**Index**

[1. Introduction 3](#_Toc482477955)

[1.1. Scope 3](#_Toc482477956)

[1.2. Goal 3](#_Toc482477957)

[2. Context 4](#_Toc482477958)

[2.1. Domotics, what is it? 4](#_Toc482477959)

[2.2. Potential Applications 4](#_Toc482477960)

[2.3. Domotic Apps 5](#_Toc482477961)

[3. System Developed 6](#_Toc482477966)

[3.1. Introduction 6](#_Toc482477967)

[3.2. Architecture and Used Technologies 7](#_Toc482477968)

[3.3. The A/C Control App 8](#_Toc482477969)

[3.3.1. Start Screen 8](#_Toc482477970)

[3.3.2. A/C State 9](#_Toc482477971)

[3.3.3. Historical Data 10](#_Toc482477972)

[3.3.4. About 10](#_Toc482477973)

[3.4. Raspberry Pi Tasks 11](#_Toc482477974)

[3.4.1. Server 11](#_Toc482477975)

[3.4.2. Sender 11](#_Toc482477976)

[4. Conclusion 12](#_Toc482477977)

# Introduction

# Scope

This project fits within the scope of Computação Móvel Multisensorial subject of the Mestrado em Engenharia de Software from the Escola Superior de Tecnologias e Gestão of Instituto Politécnico de Viana do Castelo, which aims to explore the integration of several technologies like mobile devices, hardware sensors, web services, shared databases and digital interfacing with Rasperry Pi.

Following the development and implementation of the project, the need for the creation of this document arises making it possible to register and describe all steps done to achieve the proposed goals for the project,the used technologies, the system architecture, the details of the communication methods, among other details.

# Goal

The main goal of the project is to create a system to manage and control the air conditioner physical devices of a group of divisions inside of an office.

This system will consist in an android app, a local database, a remote database, web services and a Raspberry Pi 2 to make the interface with the physical world.

The technical requirements for the success of the project are:

* Android app
* Usage of 1 hardware sensor from the android device
* Local Database for data persistence
* Remote Database to make it possible to share data with other users
* Web Services to get and set data remotely
* Raspberry Pi to turn possible the interface with the physical world

# Context

# Domotics, what is it?

It is a set of technologies responsible for the control and management of the housing resources. It came from the word “Domus” which means “house” mixed with “Robotics”, basically to allow us control something on an automated manner.

This automation concept appeared from the human need to do the less possible effort to achieve its goals. Therefore the domotics concept brings us quality of life and confort.

# Potential Applications

To clarify the “Domotics” concept here goes some practical examples for potential applications in common real world situations:

* Lights control
* Electric shutters control
* Swimming pool water quality control
* Air conditioner devices control
* Security cameras control
* Others..

All of them are perfectly possible by merging some technologies and giving each part a job.

# Domotic Apps

With the growing of the smartphones, tablets or even smartwatches usage it turned possible to connect these devices to others responsible for sensing and acting with the real world. This interaction is only possible due to a transduction process, where one form of energy is turned into another.

This means that the user of a mobile app can send a request to a Raspberry Pi and so the Raspberry can act with the physical world by sending a signal through its output pins. The opposite is also true, the raspberry might get “sensing data” coming from its input pins and send the data to a remote database which is being monitored by a mobile app.

Basically it is possibly to connect, get data and send data from “things” to our mobile devices, computers, etc. The image below illustrates this interaction.



Figura – Apps for IoT



# System Developed

# Introduction

To achieve the proposed goals successfully it was needed to integrate several technologies to make the full cycle complete. Starting with the user who detects high temperature in the room by using its device sensor within the A/C Control app, then sends a request through the app to the local database, then the app sends a request to the webservice which will comunicate with the remote database, and if the previous steps were successfull it will send a request to the Raspberry Flask API which will send the electrical signal through the output pin to the physical device. The opposite flow is also possible, so if we send an electrical signal through a button connected to the raspberry pi input pins, it will send a request to a remote webservice so it can insert data in the remote database.

This way the developed system will prove and materialize the following concepts:

* Android app with local database
* Android app detecting high temperature with the device’s sensor (it is beeing simulated with the light sensor instead of the temperature one)
* Communication between android app and php webservices
* Saving data remotely
* Raspberry Pi 2 acting as a server with a Flask API
* Android app comunicate with Raspberry Pi
* Raspberry Pi’s Flask API sends electrical signal through the output pin
* Raspberry Pi 2 gets electrical signals through input pin
* Raspberry Pi 2 sends HTTP requests to a remote web service

# Architecture and Used Technologies

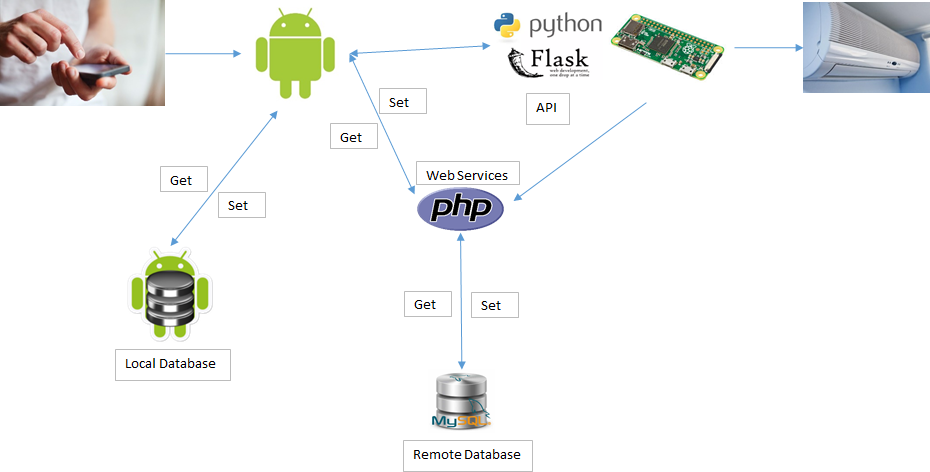


Figura 2 – System Architecture

As it is possible to see in the image above, the system was built with the following technologies:

* Android App
  + Version 4.1 or greater
* Android light sensor (simulating temperature sensor)
* Local SQLite database
* Remote MySQL database
* PHP WebServices
  + To allow the interaction with the remote database
* Raspberry Pi 2
  + To allow the interaction with the physical world, in this case by sending electrical signals to an air conditioner device
* Python + Flask API
  + Flask is a Python framework and it will act as a server inside the Raspberry Pi, meaning it will be listening in a certain port for a certain service, when request comes it will act accordingly (send the electrical signal)

# The A/C Control App

# Start Screen

When the user opens the app, the following screen will appear.

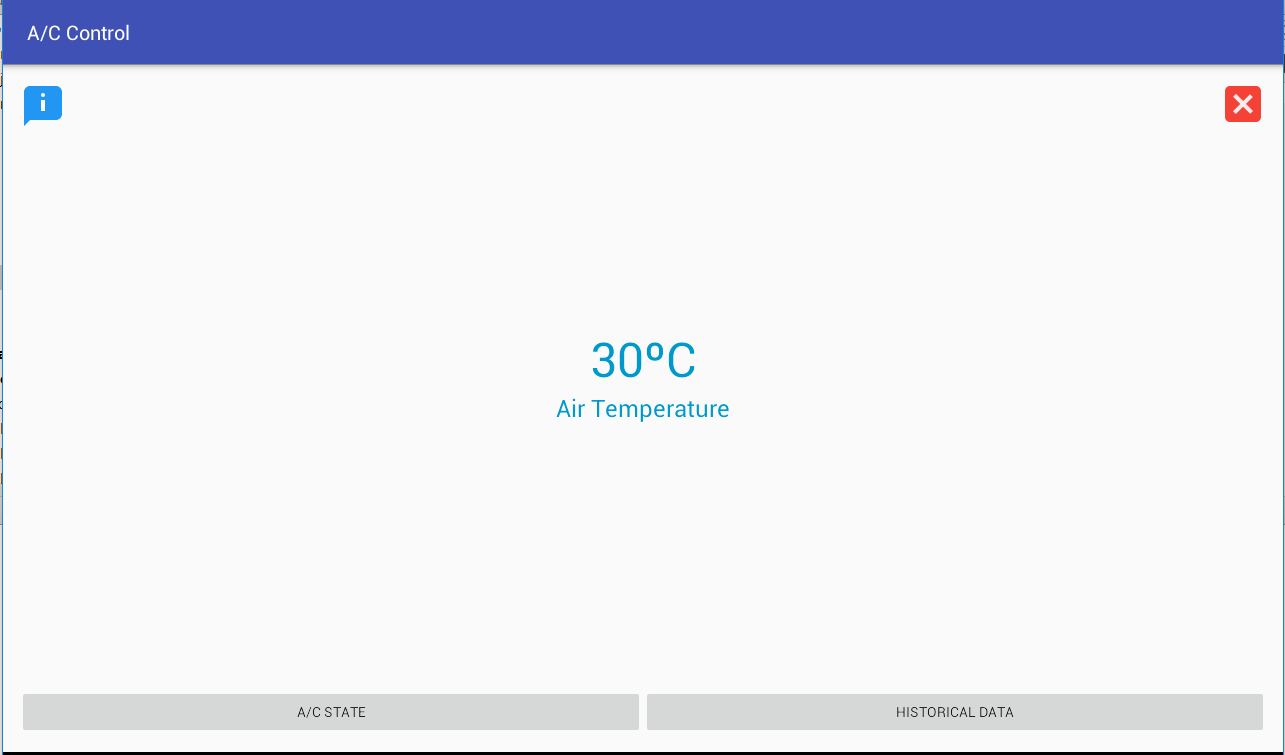


Figura 3 – Start Screen

In this screen, the app will automatically detect the air temperature and display it on the screen.

From here we can go to the A/C State screen, historical data screen, about or quit the app.

# A/C State

When the user selects the A/C State screen the following will show.

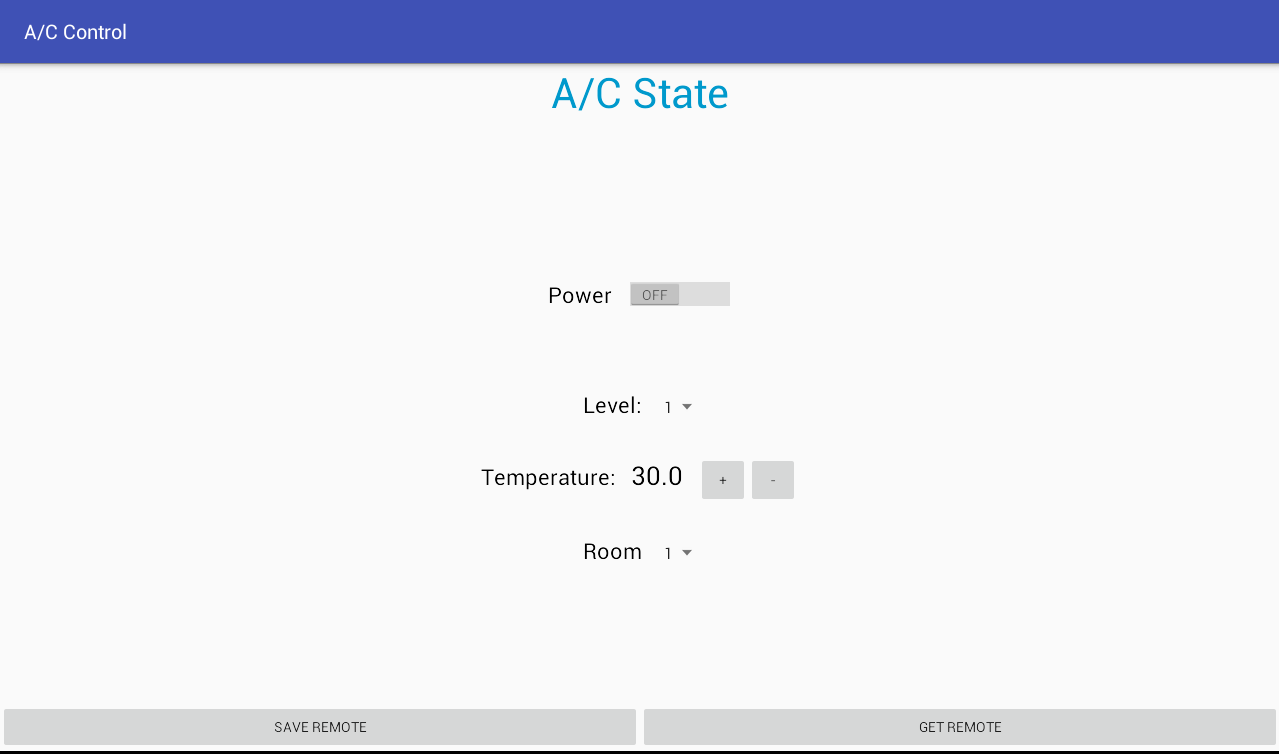


Figura 4 - A/C State Screen

Here the user is able to manage the air conditioner physical devices as well as get the last saved state.

When the user slides the power button, it will trigger the save event to the local database as well as the remote one with the data on screen, and also send the request to the Raspberry Pi.

If the user clicks “Save Remote” it will also trigger the save event mentioned above.

When the user clicks “Get Remote” it will get the last event saved from the remote database, showing all data in the respective controls.

# Historical Data

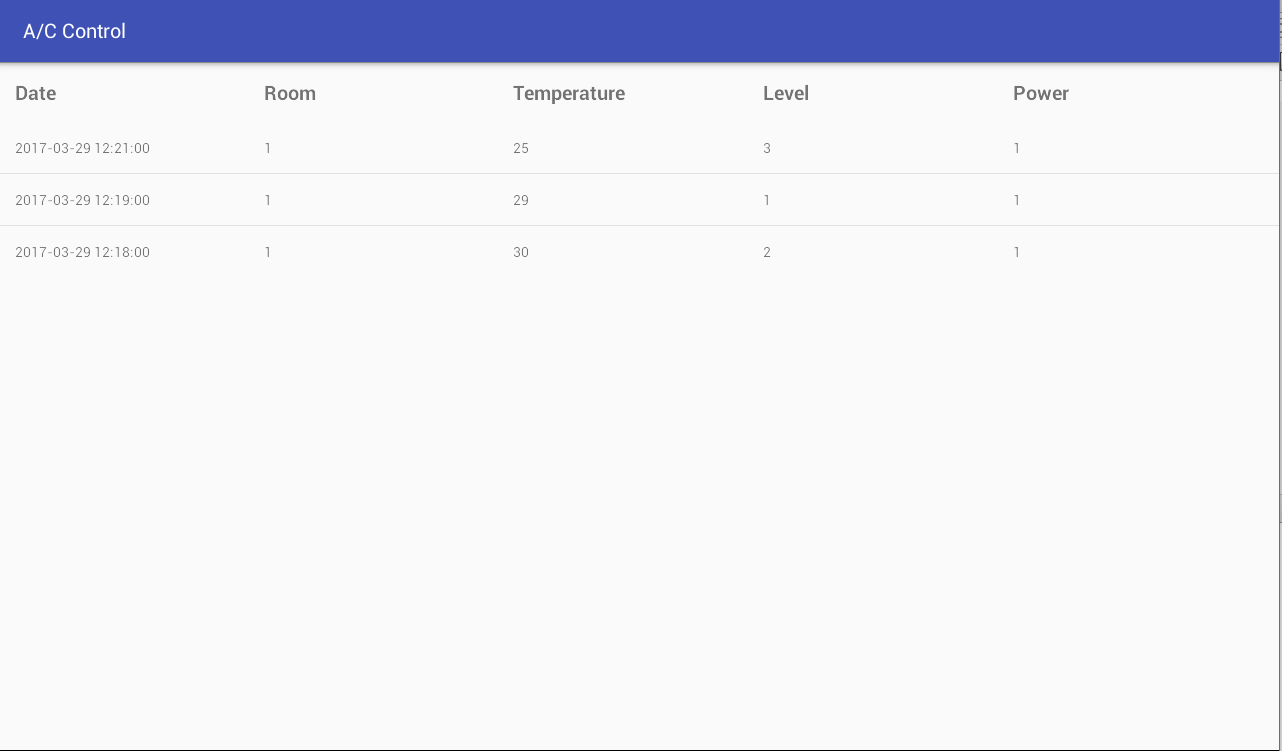


Figura 5 - Historical Data Screen

The Historical Data screen will present the user a list of all events saved in the local database.

# About

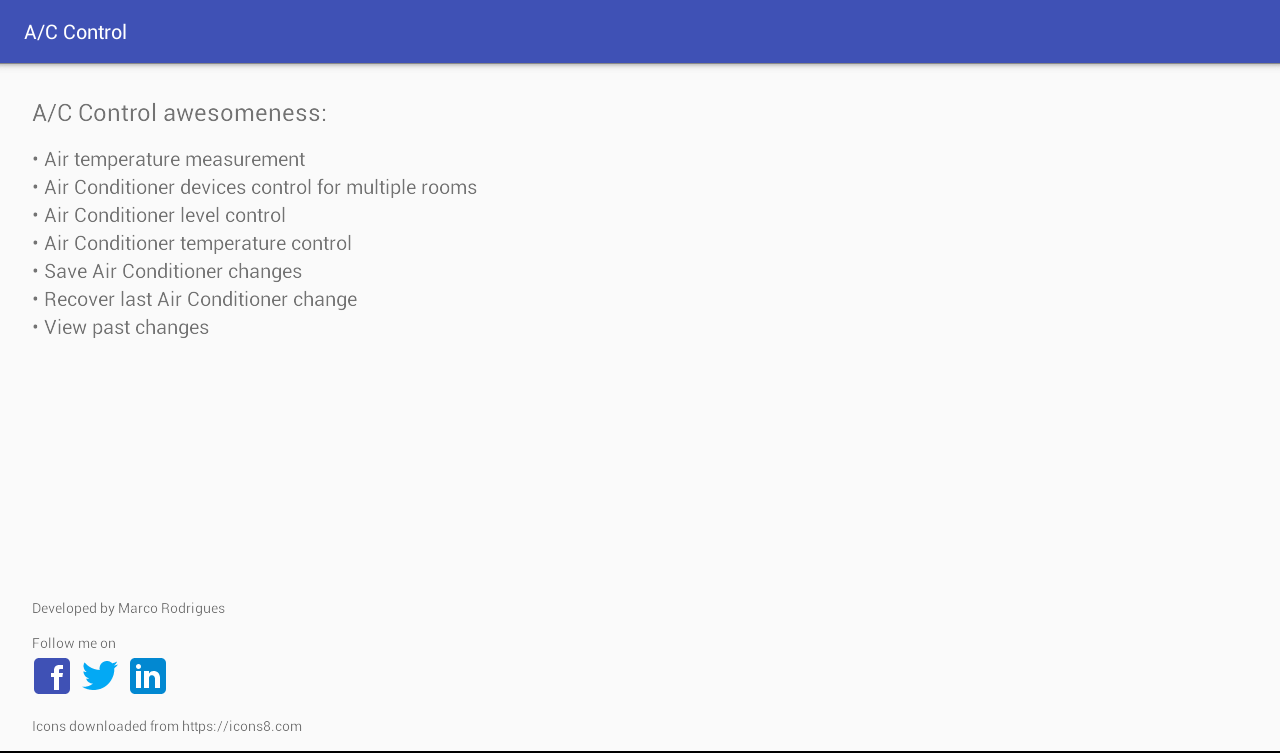


Figura 6 - About Screen

The About screen will display all functionalities of the app and some other information that might be interesting to the user like who the developer was, it’s contacts, etc.

# Raspberry Pi Tasks

# Server

The Pyhton’s Flask framework allowed me to develop a RESTful API and run it in the Raspberry, so it would behave as a server.

This way, when the user clicks on the “Save Remote” feature of the A/C Control app, data will be saved in the local and remote databases and a request will be sent to the Raspberry (which will be listening on a certain ip address/port) with the Room as a parameter.

Room possibilities are 1,2 or 3 and the Raspberry will send a number of electrical signals to the LED accordingly to the Room parameter, so if the selected room is 3 it will blink 3 times.

# Sender

Another feature of the Raspberry is to send an HTTP GET request to a web service so it can insert data in the remote database. This way, when the A/C Control app executes the “Get” feature, it might get the data sent by the Raspberry Pi.

With this feature it is possible to understand that the android app could have multiple users and also that it is possible to execute HTTP requests by a Raspberry Pi.

# Conclusion

This project allowed me to understand that the Domotics it is not the future, but the present. It is not a promise, but it is already a reality. So, with the dissemination of the microcontrollers based systems (like Arduíno) and/or microprocessors based systems (like Raspberry Pi) it is easy and cheap to build an intelligent “thing” or a human controlled “thing” through cutting-edge technologies. Android app development, remote databases and microprocessors, all of them can be integrated and connected. Smartphones, tablets, websites or apps, all of them can control a “thing” and make the connection of the cyber world with the physical one a reality.