

# *aWareHouse*

## Environment Control System for Warehouses



**FEUP** FACULDADE DE ENGENHARIA  
UNIVERSIDADE DO PORTO

MSc in Informatics and Computer Engineering  
Programming Paradigms  
EIC0065-2S

Duarte Nuno Pereira Duarte - 201109179 (ei11101@fe.up.pt)  
Hugo José Freixo Rodrigues - 201108059 (ei11086@fe.up.pt)  
João Pedro Matos Teixeira Dias - 201106781 (ei11137@fe.up.pt)

Faculdade de Engenharia da Universidade do Porto  
Rua Roberto Frias, sn, 4200-465 Porto, Portugal

May 8, 2015

## Abstract

We are in the rise of IoT (Internet of things). A world where everything is connected and we can, with simple tools, monitor and control everything. In this context, there is a lot of space for a more recurrent use of different programming paradigms because of the need of interaction with different layers of system architecture for a single application, since hardware to web.

Our application, *aWareHouse*, was designed with the objective of, with a simple interface, we can monitor a house or a warehouse in terms of environment conditions (temperature, humidity, sound and luminosity).

For accomplishing this was used a combination of hardware/software and some different programming languages, in a way that gave us a stable application that can be used for setting a alarm system when occurs changes in our environment, for taking decisions analysing the past conditions and the relations with external (meteorological) conditions or simple look at the current conditions inside our warehouse.

# Contents

<b>1</b>	<b>Introduction</b>	<b>4</b>
<b>2</b>	<b>System Description</b>	<b>5</b>
2.1	Conceptual Description . . . . .	5
2.1.1	Functionalities . . . . .	5
2.1.2	Architecture . . . . .	6
2.1.3	Programming Languages and Technologies . . . . .	7
2.2	Implementation Description . . . . .	7
2.2.1	Implementation Details . . . . .	7
2.2.2	Development Environment . . . . .	7
<b>3</b>	<b>Conclusion</b>	<b>8</b>
<b>4</b>	<b>Improvements</b>	<b>9</b>
<b>5</b>	<b>Resources</b>	<b>10</b>
5.1	<i>Bibliography</i> . . . . .	10
5.2	<i>Software</i> . . . . .	10
	<b>Appendices</b>	<b>11</b>
<b>A</b>	<b>Appendix</b>	<b>11</b>

# 1 Introduction

The *aWareHouse* project was developed for the Programming Paradigms course unit of the Master in Informatics and Computing Engineering at FEUP. The project was developed with the objective of combine, at least, three programming paradigms in the same application.

The base motivation for this project was the Internet of Things, as defined by *Gartner*:

“The Internet of Things (IoT) is the network of physical objects that contain embedded technology to communicate and sense or interact with their internal states or the external environment.”

So, we designed a low cost system capable of giving the user the possibility of, using a relative small hardware box, monitor the environment of a given place like a home or warehouse. This, associated with a system capable of maintain records of the past conditions of the environment (plus external weather conditions), results in one application that gives the user the capacity of taking decision, be aware of the environment status using alerts and see the current conditions.

As explained before the capabilities of this application make this a system useful in a lot of situations, for example, the monitoring of a warehouse and verify the relation between weather and internal conditions to make decision on what is the best settings for a refrigeration system and assure that the, for example, temperature is always below the maximum recommended. Other example is simple use it as a house control system and, for example, verify if someone forgot to turn any light off.

## 2 System Description

### 2.1 Conceptual Description

#### 2.1.1 Functionalities

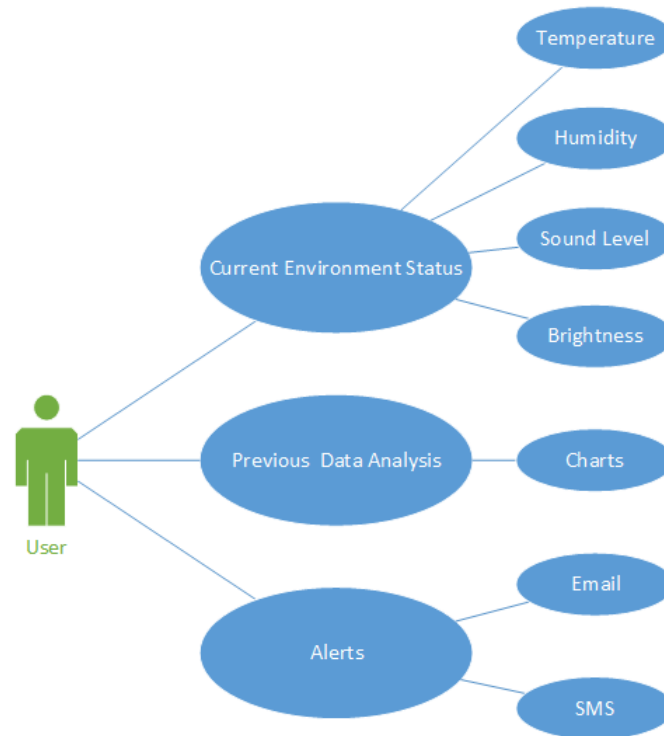


Figure 1: Use Case diagram.

### 2.1.2 Architecture

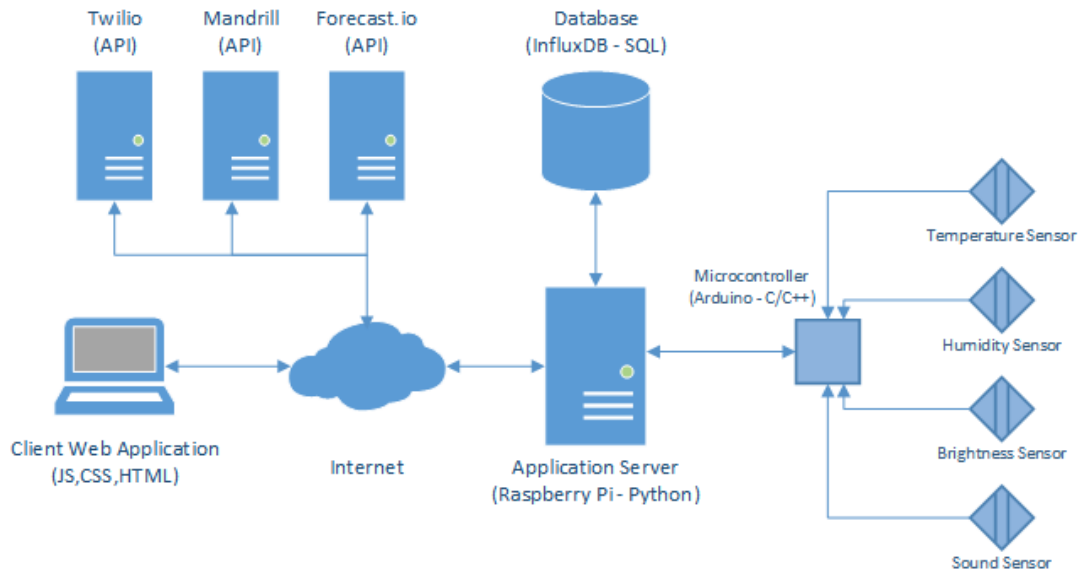


Figure 2: Architecture diagram.

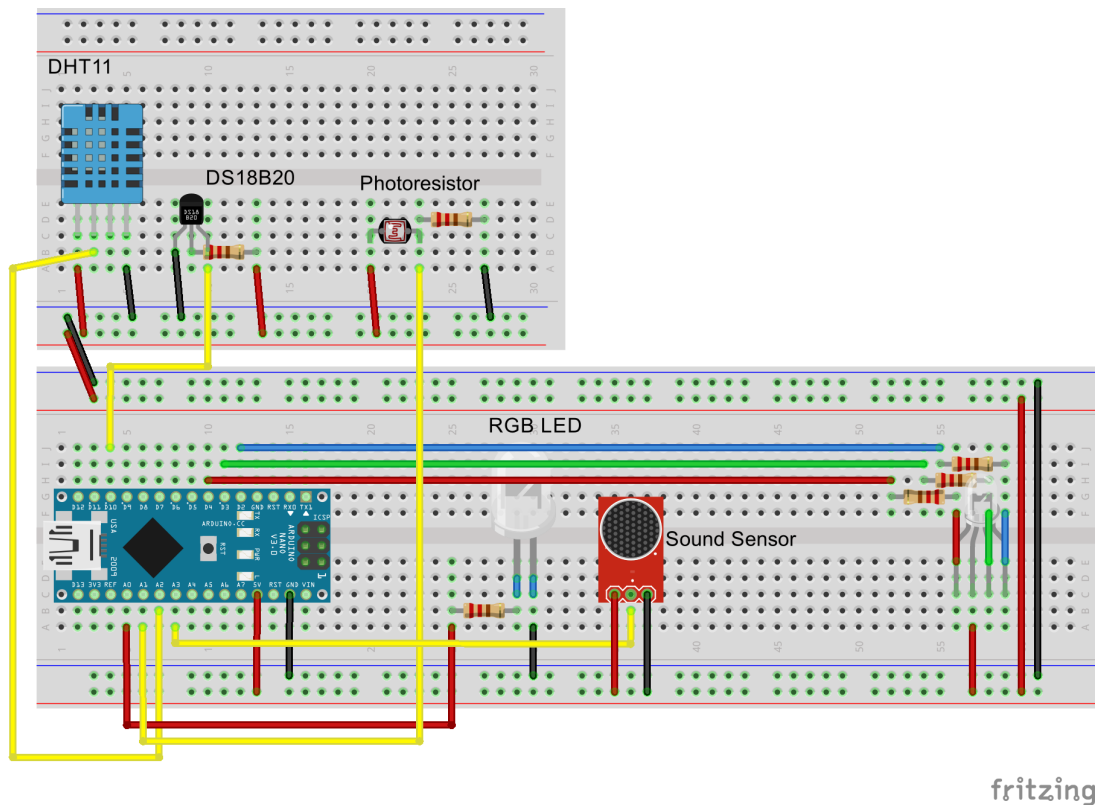


Figure 3: Circuit diagram.

### 2.1.3 Programming Languages and Technologies

In the context of realizing this project it was conciliated a diversity of technologies and programming languages that, working together in the architecture presented at fig. 2, made the application possible. Starting by enumerating the programming languages used and the reasons for choosing those languages is presented in the table 1.

Arduino	C/C++
Python	item 2
Web Languages	HTML,JS,CSS
SQL	InfluxDB

Table 1: Programming languages.

In addition to the programming languages we used a set of technologies as a part of our application as presented in table 2.

InfluxDB	
Grafana	
Web API's	Twilio, Mandrill, Forecast.io

Table 2: Technologies.

-Python lib's (?)

## 2.2 Implementation Description

### 2.2.1 Implementation Details

Socket communication over usb (Python - Arduino (c/c++))

HTTP REST protocol (web application - python server - InfluxDB - external API's - JSON)

### 2.2.2 Development Environment

Raspberry Pi, Arduino, Arduino IDE + Compiler, Atom, SSH, Python Compiler, ARM architecture , Version Control - Git

### **3 Conclusion**

almost ready to market (?), easy to expand or adapt, usefulness of programming paradigms, big data and internet of things



## 4 Improvements

Motion, better management of storage (slow), advice system (?), better safety (credentials)

## 5 Resources

### 5.1 *Bibliography*

Visual Studio 2013 Ultimate, Microsoft, <http://www.visualstudio.com/>.

### 5.2 *Software*

Visual Studio 2013 Ultimate, Microsoft, <http://www.visualstudio.com/>.

# Appendices

## A Appendix

The contents...