Pontifícia Universidade Católica do Rio de Janeiro

Marcio Luiz Coelho Cunha

Software of Places: Toward a Self-Learning Closed Plant Production System

Tese de Doutorado

Thesis presented to the Programa de Pós-Graduação em Informática of the Departamento de Informática, Centro Técnico Científico, PUC-Rio as partial fulfillment of the requirements for the degree of Doutor.

Advisor: Prof. Hugo Fuks

Rio de Janeiro
December 2017

Pontifícia Universidade Católica
do Rio de Janeiro

Marcio Luiz Coelho Cunha

Software of Places: Toward a Self-Learning Closed Plant Production System

Thesis presented to the Programa de Pós Graduação em Informática of PUC-Rio in partial fulfillment of the requirements for the degree of Doutor em Ciências - Informática. Approved by the undersigned Examination Committee

Prof. Hugo Fuks
Advisor
Departamento de Informática - PUC-Rio

Profa. Simone Diniz Junqueira Barbosa
Departamento de Informática - PUC-Rio

Prof. Alberto Barbosa Raposo
Departamento de Informática - PUC-Rio

Eng. Dra. Denise Del Re Filippo
UERJ

Prof. Fernando Reiszel Pereira
UERJ

Prof. Marcio da Silveira Carvalho
Vice Dean of Graduate Studies
Centro Técnico Científico – PUC-Rio

Rio de Janeiro, December 20th, 2017

Marcio Luiz Coelho Cunha

Graduated in Computer Engineering at PUC-Rio in 2008 and obtained his M.Sc. Degree in Computer Science from PUC-Rio in 2010.

Bibliographic data

Cunha, Marcio Luiz Coelho

Software of Places : toward a self-learning closed plant production system / Marcio Luiz Coelho Cunha ; advisor: Hugo Fuks. – 2017.
111 f. ; 30 cm

Tese (doutorado)–Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática, 2017.
Inclui bibliografia

1. Informática – Teses. 2. Fábrica de plantas com luzes artificiais. 3. Internet das coisas. 4. Fazendas verticais. 5. Análise das coisas. 6. Inteligência ambiental. I. Fuks, Hugo. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. III. Título.

CDD: 004

Acknowledgements

To those that contributed to this research.

Abstract

Cunha, Marcio Luiz Coelho; Fuks, Hugo(Advisor). **Software of Places: Toward a Self-Learning Closed Plant Production System.** Rio de Janeiro, 2017. 111p. Tese de Doutorado - Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

As the population grows, more food will need to be produced in the next four decades than has been in the past 10,000 years. However, the modern world still depends on high yield monoculture production which is increasingly threatened by unusual weather, water shortages, and insufficient land. In order to overcome these problems and feed the world, a practical path to provide quality fresh healthy food at scale with minimal weather dependency, water usage and reduced carbon footprint is necessary. One reasonable approach is to build vertical farms inside the cities in a close environment full of sensors and artificial lighting controlled by software for efficient production of food crops. This thesis proposes a model, entitled Software of Places Cycle (SoPC), that should be able to answer to environmental stimuli in a closed plant production system using artificial lighting in order to create a self-learning environment. This thesis describes the SoPC, the approaches and processes of implementing a mini Plant Factory using Artificial Lighting based on the discussion on five action-research cycles. The thesis main contribution is a conceptual model to guide the development and maintenance of a mini-PFAL (m-PFAL), a minor contribution is the deployment of the SoP, i.e., the very notion of having software dedicated to a specific place.

Keywords

Plant Factories with Artificial Lighting; Internet of Things; Vertical Farming, Controlled Environment, Analytics of Things, Ambient Intelligence.

Resumo

Cunha, Marcio Luiz Coelho; Fuks, Hugo. **Software dos Lugares: Em direção a um Sistema Fechadio para Produção de Plantas com Auto-Aprendizado.** Rio de Janeiro, 2017. 111p. Tese de Doutorado - Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

À medida que a população cresce, mais alimentos precisarão ser produzidos nas próximas quatro décadas do que nos últimos 10.000 anos. No entanto, o mundo moderno ainda depende da produção de monoculturas de alto rendimento, cada vez mais ameaçada por condições climáticas incomuns, escassez de água e terra insuficiente. A fim de superar esses problemas e alimentar o mundo, é necessário um caminho prático para fornecer alimentos frescos, com qualidade e em escala, com mínima dependência do clima e com uso de água e pegada de carbono reduzidos. Uma abordagem razoável é construir fazendas verticais dentro das cidades em um ambiente fechado repleto de sensores e iluminação artificial controlada por software para uma produção e gestão eficiente do plantio de alimentos. Esta tese propõe a instanciação de um modelo, chamado Ciclo do Software dos Lugares (SoPC), que é capaz de responder a estímulos ambientais em um sistema fechado de produção de plantas com iluminação artificial que possibilite a criação de ambientes com auto-aprendizagem. Esta tese descreve o SoPC, as abordagens e processos de implementação de uma mini fábrica de plantas com iluminação artificial com base na discussão em cinco ciclos de pesquisa-ação.

Palavras-chave

Fábrica de Plantas com Luzes Artificiais; Internet das Coisas; Fazendas Verticais, Ambientes Controlados, Análise das Coisas, Inteligência Ambiental.

Summary

List of figures

List of tables

1 Introduction

As the population grows, more food will need to be produced in the next four decades than has been in the past 10,000 years. Worldwide urban population will have a 72% increase by 2050[1]. Brazil already has a high level of urbanization: 82 out of every 100 Brazilians live in cities and more than half of this population are concentrated in just 5.6% of 5570 Brazilian cities[2]. Nevertheless, the modern world still depends on high yield monoculture production which is increasingly threatened by unusual weather, water shortages, and insufficient land. Food security, in context of availability, accessibility, utilization and stability, is expected to be a challenge in the near future. In order to overcome these problems and feed the world, a practical path to provide quality fresh healthy food at scale with minimal weather dependency, water usage and reduced carbon footprint is necessary.

One reasonable approach is to build vertical farms inside the cities in a close environment full of sensors and artificial lighting controlled by software for efficient production of food crops. These environments can be leveraged by the ongoing miniaturization of electronics, the commodification of bits [1] and the growing domination of software over materialized form, creating a new layer that brings behavior to everyday objects and to their surroundings, namely Software of Places (SoP) [2].

Currently, verticals farm can be Internet-enabled by the Internet of Things (IoT), linking it to additional computing power and analytics capabilities that make it aware and able to answer to pre-modeled environmental stimuli. However, there is a lack of a scalable model to develop and deploy applications atop such a heterogeneous collection of ubiquitous devices [3]. In order to provide a proper answer for different stimulus a Software of Places Cycle (SoPC) need to be developed for describing how sensor data feeds machine learning algorithms creating value that empower self-learning environments.

[1] http://www.fao.org/docrep/016/ap106e/ap106e.pdf

[2] https://oglobo.globo.com/brasil/ibge-mais-da-metade-dos-brasileiros-mora-em-apenas-56-dos-municipios-21763856

In this thesis, I propose to study the instantiation of a SoPC in a very specific environment, a Closed Plant Production System (CPPS) [4] to evidence the SoPC applicability in a Self-Learning CPPS (s-CPPS) [5]. This thesis describes the SoPC, the approaches and processes of implementing a Self-Learning CPPS using sensor fusion and data analysis with a focus on a practical aspects of a specific type of CPPS, a Plant Factory using Artificial Lighting (PFAL) [6]. The discussion in this thesis will focus on five action-research cycles, where I investigate technological issues regarding our prototype configuration and use. Our main contribution is a conceptual model to guide the development and maintenance of a domestic size PFAL, a mini-PFAL (m-PFAL). A minor contribution is the deployment of the SoP, i.e., the very notion of having software dedicated to a specific place.

1.1.Overview

There are many predictions over how ubiquitous IoT will be, but most of them indicate that the marketplace will host between 20 and 30 billion connected objects by 2020 [3], signaling novel challenges for hardware manufacturing, maintenance and software development. The early days of the IoT have largely been focused on sensor enablement and data collection. However, the real value comes not from connecting an object in order capture and store its data, but, from the Analytics of Things (AoT), i.e., by extracting meaning from this huge data volume.

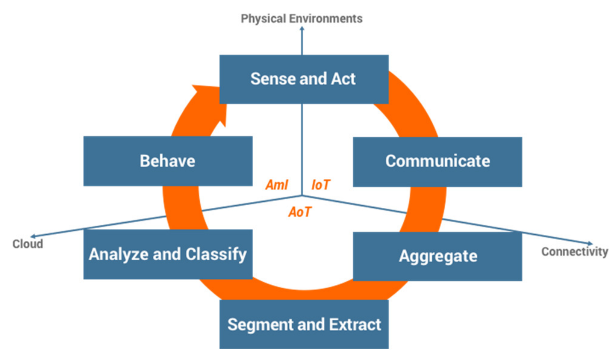

Figure 1. **Software of Places Cycle**

Modeling predictive behavior enabled by AoT seems to be the key to next generation IoT products. However, achieving a highly accurate model is not an easy task. The SoPC (Figure 1) helps to tame this complexity by dividing it into three

[3] http://www.gartner.com/newsroom/id/3598917

stages (IoT, AoT and AmI) comprising six steps, namely, Sense and Act, Communicate, Aggregate, Segment and Extract, Analyze and Classify, and Behave. To evidence the SoPC applicability, a m-PFAL was built (Figure 2) in the SecondLab at the *Núcleo de Informação Tecnológica* (NIT) from the Department of Informatics (DI) of Pontifical Catholic University of Rio de Janeiro (PUC-Rio).

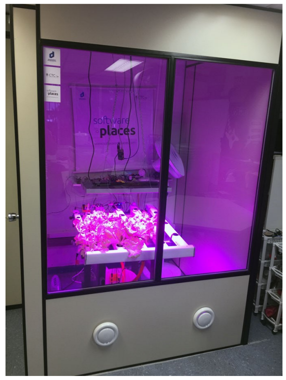

Figure 2. **Mini s-PFAL in SecondLab at PUC-Rio**

Technically, a m-PFAL is an indoor, advanced, and intensive form of hydroponic production system where the growing environment is controlled by software designed to optimize agriculture production by monitoring and actuating a desired climate recipe inside of a closed chamber [7]. In order to evidence the SoP model applicability, this thesis will address how to build a m-PFAL using the SoPC to guide the sensors choice, data segmentation and extraction to fine tune the system to fulfill a climate recipe that keeps the system at the following microclimate benchmark from Kozai et al. [6]:

- Air and water temperature between 22°C and 25°C,
- Humidity at 70%,
- Dissolved Oxygen (DO) at 9 PPM,
- A 18h photoperiod at the growth stage,
- Water Electrical Conductivity (EC) between 1.2 and 2.2 µS/cm,
- Water pH between 5.5 and 6.5, and
- Hydroponic nutrients feeding schedules following the Flora Series feeding charts from General Hydroponic[4].

[4] https://goo.gl/3uxFw1

1.2.Motivation and Significance of the Study

One of the motivations of this project lies in the concept of the IoT. By applying technology to everyday objects, the embedded software needs to be aware of its surroundings and interact with the world in the physical space. It needs to analyze data from sensors that have unique attributes related to their place and task. Architectural elements of physical space will frame and cue actions [2]. Sites of interaction involve geometric relationships and constraints at fine resolution making the embedded software and hardware unique to its place of action. Ultimately, the IoT asks for a new way of thinking software, where machines and other devices supplant humans as the primary means of collecting, processing and interpreting data [8] in a specific location. This data quickly becomes too complex [9].

1.3.Thesis Structure

The remainder of this thesis is structured as follows:

- **Chapter 2. Context.** This chapter provides a context for two important concepts of this thesis, Internet of Things and Plant Factories using Artificial Lighting.
- **Chapter 3. Related Work.** This chapter provides related work of the state-of-the-art of the following domains: Food Computing, Plant Factory R&D in the world, Artificial Light, Ambient Intelligence and Autonomic Computing.
- **Chapter 4. Software of Places.** This chapter describes our conceptual model, the Software of Places Cycle, the system design and architecture.
- **Chapter 5. Research Cycle.** This chapter describes our action research approach and reports the five research cycles I went through, following a typical action research cycle template.
- **Chapter 6. Research Cycles Dataset Wrap Up and Analysis.** This chapter I summarize the research cycles dataset and propose a new and untested prediction model to the system.

- **Chapter 7. Conclusion and Future Work.** This chapter presents my finals considerations, highlighting my main contributions and pointing to future work.

2 Context

In this section, I give a general overview about IoT and PFAL, as well as the motivations for investigating them. Despite the challenges and broadness of the scope, I explain the reasons why I think that approaching both of them together may bring significant benefits to the SoPC development, offering different perspectives from what is usually seen when each topic is researched in an isolated fashion. This is not a related work, but prepares the terrain for it, which will be presented on the following section (Section 3).

2.1. Internet of Things

Although IoT is in its infancy there are many different ways to understand and define it. In our view, IoT stands as the concept coined in 1999 by Kevin Ashton and his team at MIT's Auto-ID Center with the initial focus on developing universal electronic identification codes of products to support the widespread use of RFID (Radio-Frequency IDdentification) in commercial networks [8]. But it rapidly spread around encompassing a much broader concept thanks to the evolution of sensor, embedded electronics and mobile (wireless) communication technology.

The basic idea is the widespread presence of a variety of things – identified objects, sensors, actuators, computers, cell phones, etc. – that, through wireless communication and unique ID, are able to interact with each other and collaborate with its neighbors to achieve common goals [10]. Today, multiple definitions and facets of the Internet of Things are described by the scientific community. In this thesis, I am working with the following definition:

"In the context of the Internet, addressable and interconnected things, instead of humans, act as the main data producers, as well as the main data consumers. Computers will be able to learn and gain information and knowledge to solve real world problems directly with the data fed from things. As an ultimate goal, computers enabled by the Internet of Things technologies will be able to sense and react to the real world for humans." [11].

The key feature in IoT is, without any doubt, its impact on everyday life of potential users [12]. IoT has remarkable effects both in work and home scenarios, where it can play a leading role in the next future (assisted living, domotics, ehealth, smart transportation, etc.). Disruption is also expected for business (e.g. logistic, agriculture, industrial automation, transportation of goods, security, etc.).

According to these considerations, in 2008 IoT has been reported by US National Intelligence Council as one of the six technologies with potential impact on US interests towards 2025 [13]. Indeed, in 2011 the number of interconnected devices overtook the number of people [14]. In 2012, the number of interconnected devices was estimated to be 9 billion, and it was expected to reach the amount of 24 billion by 2020 [14].

2.2. Plant Factory using Artificial Lighting - PFAL

The term “Plant Factory using Artificial Lighting” refers to a plant production facility with a thermally insulated and nearly airtight warehouse-like structure [6]. Multiple culture shelves with LEDs on each shelf are vertically stacked inside to increases the efficiency of land use. Other necessary equipment and devices for running a PFAL are air conditioners, air circulation fans, CO_2 and nutrient solution supply units, and an environmental control unit. LEDs are increasingly being used in recently built PFALs owing to their compact size, low lamp surface temperature, high light use efficiency, and broad light spectra.

Figure 3. PFAL in Japan

The rapid development of PFALs has created new markets and business opportunities [15]. PFALs are already being used in Japan (Figure 3), Germany, Netherland and other countries for commercial production of leafy greens, herbs, and transplants. Indoor vertical farms, which is another term used in North America for concepts similar to PFALs, are also being built in the United States and Canada.

When growing plants in an open field, yield and quality are subject to weather conditions, and so a stable and reliable supply of plant-derived food is always in danger. Greenhouse production is not energy efficient because incident light is not regulated. Solar light intensity is often too low at dawn, sunset, and night, on cloudy and rainy days, and throughout the winter season, while it is too high around noon on sunny days [6].

The temperature and relative humidity inside a greenhouse are considerably affected by solar light intensity, and thus it is difficult to optimize the environment. In order to lower the temperature, greenhouses are often ventilated, but this allows insects and diseases inside the greenhouse. Furthermore, light quality and lighting direction are not controllable. Excessive agrochemicals are often used in greenhouse and open-field production and fossil fuels are needed for heating and cooling of greenhouses and for transportation of produce over long distances from production site to consumers.

On the other hand, the PFAL is an indoor, advanced, and intensive form of hydroponic production system where the growing environment is optimally controlled. PFAL is one form of “closed plant production system” (CPPS), where all inputs supplied to the PFAL are fixed by plants with minimum emission to the outside environment. If designed and managed properly, the PFAL has the following potential advantages over the conventional production system [16]:

a. It can be built anywhere because neither solar light nor soil is needed;
b. The growing environment is not affected by the outside climate and soil fertility;
c. Production can be year-round and productivity is over 100 times that of field production;
d. Produce quality such as concentrations of phytonutrients can be enhanced through manipulation of the growing environment, especially light quality;
e. Produce is pesticide-free and need not be washed before eating
f. Produce has a longer shelf life because the bacterial load is generally less than 300 CFU/g (colony-forming units per gram), which is 1/100 to 1/1000 that of field-grown produce;
g. Energy for transportation can be reduced by building PFALs near urban areas; and
h. High resource use efficiency (water, CO_2, fertilizer, etc.) can be achieved with minimum emission of pollutants to the outside environment.

Owing to the PFALs requirements mentioned above, in this thesis I prototyped a mini-PFAL (m-PFAL) to evidence the SoP Cycle. m-PFALs are characterized by [16]: (1) hydroponic or soilless culture, (2) plants are grown mostly under LED lamps, (3) easy to use and maintain, (4) no use of pesticides, and (5) lighting period, watering, temperature, and so forth are automatically controlled. The size, i.e., the air volume of the plant-growing space, of most mini-PFALs ranges from 2 m^3 (e.g., 2 × 0.5 × 2 m) to 30 m^3 (e.g., 2 × 5 × 3 m). m-PFALs are classified into three types [16]: Type (A) natural or forced ventilation through air gaps covered with fine mesh nets to prevent insects from entering, and without air conditioning; Type (B) air conditioned but some degree of natural ventilation and no CO_2 enrichment; and Type (C) an airtight structure equipped with an air conditioner and CO_2 enrichment unit.

3 Related Work

Various efforts have been made to allow people to benefit from the new opportunities on AmI, IoT and modern lighting systems. In this section, I will discuss about Food Computing and PFAL R&D around the world. Next, I will present some definitions of ambient intelligence and ubiquitous systems. Then, I will present current smart lighting systems. Finally, I will discuss related work about autonomic computing.

3.1. Food Computing

The Food Computer (FC) is a term for an agricultural technology platform proposed by Caleb Harper [17] that creates a controlled environment using robotic control systems and actuated climate, energy, and plant sensing mechanisms. FCs are designed to optimize agricultural production by monitoring and forcing a desired climate inside of a growing chamber. These climate recipes are composed by such variables as carbon dioxide, air and water temperature, humidity, dissolved oxygen, light intensity and radiation, water electrical conductivity, water pH and others ions-selective sensors to monitor hydroponic nutrients that feed the plants. These fused multi-sensors points are coupled with machine learning to analyze a group of recipes to be able to recommend new recipes that can grow plants in a desired time range and desired morphologic and physiologic expression [17].

Figure 4. A personal food computer

There are currently three scales of FCs being developed at MIT Media Lab OpenAG Initiative. The personal FC is a product-scale (2–10 sq. ft.) designed for a home user, hobbyist, or student (Figure 4).

Figure 5. The boutique production FC

The boutique production FC is a shipping-container scale (200–500 sq. ft.) designed for owner/operators or franchisees to sell small amounts of high-value produce into local markets, restaurants, or cafeterias (Figure 5).

Figure 6. The factory FC

The factory FC is a light industrial scale (+10,000sq. ft.) designed to operate in urban or peri-urban environments and distribute fresh produce into a regional supply chain or produce a large quantity of a very high-value crop (Figure 6). The ultimate goal of the OpenAG initiative and FCs project is to create an Open Phenome Project to be a crowdsource catalog of plants and their phenotypic traits correlated with the causal environmental variable. As the global network of FCs grows and begins to create and iterate over digital plant recipes, over time, FCs recipes would be optimized to decrease water, energy and minerals consumption, while increasing nutrient density, taste and other desirable characteristics. In this thesis, the FC bill of materials was used as a baseline to choose the m-PFAL components. The Open Phenome Project was also an inspiration that demonstrate how important the dataset and machine learning models will be the fourth agricultural revolution [17], or farming 4.0.

3.2. PFAL R&D in the World

This section describes the history, current status, and perspectives of PFALs in Japan, Taiwan, China, North America, and Europe, including research, development, and business:

I. Japan: The first commercial PFAL was established in 1983 in Shizuoka Prefecture. At that time, high-pressure sodium lamps were used as the light sources. Today, the largest PFAL is in Kyoto and produces 23,000 leaf lettuce heads daily [15]. By the end of 2016, the number of PFALs used for commercial production was one hundred and ninety one 191 and only 30% of PFALs are making profits, 50% are break-even, and 20% are losing money [15].

II. Taiwan: There are 56 organizations engaged in leafy green production using PFALs in Taiwan. Among these 56 organizations, there are 2 research institutes, 4 universities, and 50 private. Some companies have started to export and build turnkey PFALs abroad, mainly in China. Shops with a PFAL in the back, and with a restaurant or stand selling organic products, are a popular business model in Taiwan. Such shops are normally chain stores and are located throughout a city.

III. China: In China, studies on PFAL technologies began in 2002 and are mainly focused on hydroponic technologies and their control systems, supported by the Ministry of Science and Technology of China. Since then, R&D on PFAL in China has advanced rapidly. By 2015, about 35 plant factories had been built [15] and the project was joined by 15 universities. At Zhejiang University, the total area of the plant factory is 1600 m^2, and it has 10 layers of movable cultivation beds for efficiency reasons. Many research teams of Zhejiang University, including control science and engineering, light engineering, computer science and technology, agricultural engineering, biological engineering, and horticulture and plant nutrition, are working together to improve PFAL technologies.

IV. North America: In the USA and Canada, several large-scale commercial facilities were recently built to produce pharmaceutical protein products (antigens and antibodies) using PFALs. More recently, large commercial PFAL facilities were built close to large cities such as Chicago and New York (aerofarms.com and boweryfarming.com). NASA was the first research institute to initiate an Ecological Life Support System (ELSS) Program in 1980 [18], where plants were grown hydroponically under artificial (electric) lighting.

V. Europe: The Netherlands is the most advanced country regarding plant factories and protected glasshouse cultivation in the EU. Philips' Horticulture LED Solutions Group in the Netherlands has been developing LED lighting solutions for horticultural applications for more than 7 years and has recently started working on PFAL. In other parts of Europe, some small-scale vertical farms have been built, while construction has begun on some large-scale ones. In 2012, the Swiss company UrbanFarmers, a spin-off of the Zurich University of Applied Sciences (ZHAW), built a 260m^2 greenhouse farm on an industrial rooftop in Basel [15].

This section shows that most of the research in PFALs are being made by big companies and universities highly funded. This is a very expensive research and a big investment is necessary to produce several crops in parallel in order to accelerate the research for machine learning models, datasets, management models, feasibility and equipment (most researches are on grow lighting equipment). Most of these researchers are using industrial size PFAL with a multidisciplinary team and their research findings are not being published.

3.3. Smart Lighting

The 21 centuries saw lighting becoming a digital medium on the top of capturing, transporting and rendering information; it is providing settings for improving health, safety, wellbeing, productiveness and sustainability in many areas (e.g., smart industry, farming, ambient assisted living, offices, schools and healthcare) [19]. This new software based lighting paradigm is leveraging the advances in the Internet of Things, Machine Learning and Ambient Intelligence delivering value beyond remote control and automation. However, there is still a lack of research on how to adapt and tame in real-time this new exponential complexity of computing and communication at high dynamic, distributed, constraints resources and lossy environments [20].

One of the main smart lighting enablers has been the introduction and emergence of semiconductor based digital light sources such as LED (Light Emitting Diode) and LED technologies such as Organic Light Emitting Diodes, also known as OLEDs or Solid State Light (SLL) sources [21]. Besides the advantage of low consumption (range 3-12 volts), LEDs do not depend on the lamp/socket paradigm, are smaller, resistant, and are able to emit different light spectrums to suit the user and lit environments needs, directly affecting the health, humor and productivity [21]. LEDs can also deliver optical and data communications (Li-Fi) or Visible Light Communication (VLC), and are becoming a new option to scalable and secure wireless communication [22].

Today smart lights are remote controlled from smartphones, voice commands and programmable physical switches. Examples of such commercial product systems are LIFX, Philips Hue, Stack Alba, Samsung Smart Bulb and LG Smart Light. All these devices traditionally provide on/off, dimmer and color control on an app. Besides this, each individual bulb can be set into "scenes", where pre-defined color and intensity is applied to all bulbs in this group.

These smart bulbs communicate via ZigBee, Wi-Fi or Bluetooth to a dedicated bridge device that communicates via the internet or locally to the user's smartphone. This way the individual bulbs respond to direct lighting changes in terms of color and brightness initiated by its owner. Additionally, the same system allows for automatic behavior, based on the web service If This Then That (IFTTT). This service has a broad selection of triggers stemming from other services such as the weather, time of day, incoming mail, social updates or sport news. Other smart lights can communicate via BLE or Wi-Fi to enable remote control lights with voice commands. Currently, it is possible to integrate them with SIRI on iPhone (Apple Home Kit), Amazon Echo and Google Home.

Recently Sony released a demonstration video on its Japan based corporate web site about its multifunctional light[5]. This light communicates with other

[5] https://www.sony.co.jp/Products/multifunctional-light/

appliances like air conditioning units and thermostats using data from its humidity sensors, talks to TVs using infrared to turn it on when its motion sensors detect motion in the room, and act as a household intercom system equipped with speakers and a microphone. All of these functions are controlled via a smartphone app whenever the user is inside or away from home.

Smart lighting was the first project chosen for evidence de SoP Cycle [19][20][23][24]. However, to prototype and test it outside the lab is expensive, time consuming and has many ethical and privacy issues. During the previous research proposing the use the SoP Cycle in assisted living spaces helped to uncover that light is also a production tool and it is suitable to growing plants with artificial light which led to this research.

3.4. Ambient Intelligence, Sentient Computing

Ambient Intelligence (AmI) environments are ubiquitous systems that enhance physical environment using heterogeneous computational and wireless communication devices integrated and, at the same time, invisible to the inhabitant [25]. Hence, AmI applications need intelligent capabilities to adapt and respond to context in order to provide services. Sentient Computing (SC) is an approach that allows AmI to interact with their physical environment by becoming aware of their surrounding and respond upon them [26]. Awareness is achieved by means of a sensor infrastructure that helps to maintain a model of the world, which is shared between dwellers and applications [27]. Sentient artifacts have the ability to perceive the state of the surrounding environment, through the fusion and interpretation of data from diverse sensors [27].

AmI, SC and IoT include half a dozen different terms all synonymous or overlapping, all vying for relevance, citation and importance in the field. The same occurs with Ubiquitous and Pervasive Computing. Stajano [28] suggests the difference in names was often related to different research groups and a desire to claim antecedence in the concepts than any concrete differences: we can now regard ubiquitous and pervasive as synonymous. In this thesis, I will also consider that AmI and SC are synonymous and I will base myself on the following definition of

AmI [29]: “A mechanism that rules the behavior of the environment.”. This definition was the key to namely the third section of the SoPC.

3.5. Autonomic Computing

The complexity of software systems has reached a level where the operations needed to configure, maintain and keep them running are getting too costly and error prone. In this context, the term of autonomic computing was introduced [30] [31] to denote systems that manage themselves according to certain policies previously set by administrators. Thus, autonomic computing has emerged as a research area grouping together the efforts of building systems that exhibit selfmanagement properties: self-configuration, self-optimization, self-healing and selfprotection. Autonomic systems are often referred as self-adaptive and selfmanaging systems [30].

Each device in autonomic computing consist of sensors, actuators and processors. The sensors tasks are to monitor the behavior of the system, while the actuators are used to trigger any actions deemed necessary [32]. The process begins with the system collecting data originated from the sensors and, then, comparing the observed situation in the environment with what it is expected. Then, the system analyzes the data and makes decisions on how to act. If an action is required, it is performed and its effects are monitored, creating an autonomic feedback control loop Figure 7.

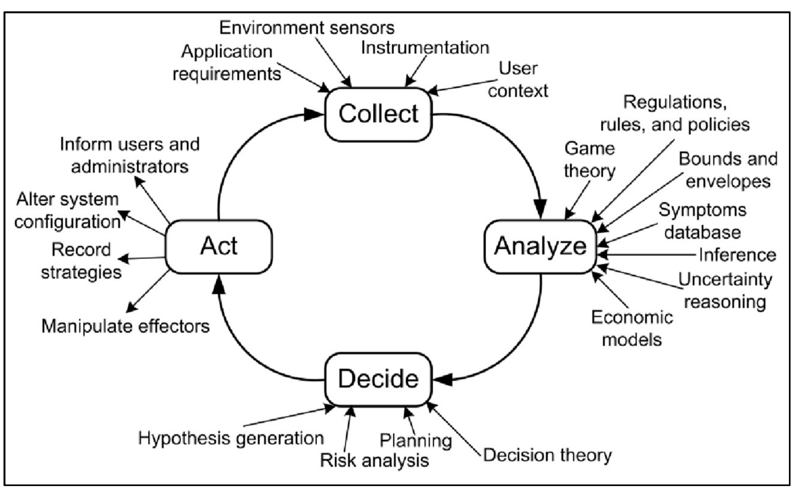

Figure 7. Autonomic Feedback Loop [31]

Autonomic computing also provides a reference knowledge base containing the system states, symptoms, references, rules and models to compare with the system observed behavior. The SoP Cycle broadens the autonomic feedback loop to be used as an IoT information value loop using AoT and AmI.

4 Software of Places

The definition of a computer is changing again. The continued evolution toward cheaper processors and faster networks has enabled a shift from desktop computers to mobile phones and, now, to everyday objects, to our body [33] (Figure 8) and to the environment itself. Almost any device and any place can become Internet-enabled by the IoT, linking it to additional computing power and analytic capabilities that make it “smart”.

Figure 8. Conductive hair as a mobile input device

These “smart” environments can be leveraged by the ongoing miniaturization of electronics, the commodification of bits [1] and the growing domination of software over materialized form, creating a new layer, namely Software of Places (SoP) [2]. In order to provide analytics capabilities that make this new layer of SoP aware and able to answer to pre-modeled environmental stimuli there is a need for understanding that sensors data have some unique attributes related to its place and task, so, related analytics are unique as well, thus creating the Analytics of Things (AoT).

Finally, to provide behavior using the SoP it is necessary to connect the IoT and AoT to the ambient intelligence ecosystem, where devices and environments behave in concert to support people in carrying out their activities and tasks in a “natural” way using data and intelligence that is hidden in the network connecting these devices [34]. In order to continuously tune and set the proper SoP acting behavior, a SoP Cycle is used to model the repetitive but shifting daily tasks activities.

4.1. SoP Cycle

Inspired on IBM's Autonomic Feedback Loop and Deloitte's IoT Information Value Loop[6], my proposal is to combine and broaden them to create the SoP Cycle (Figure 9).

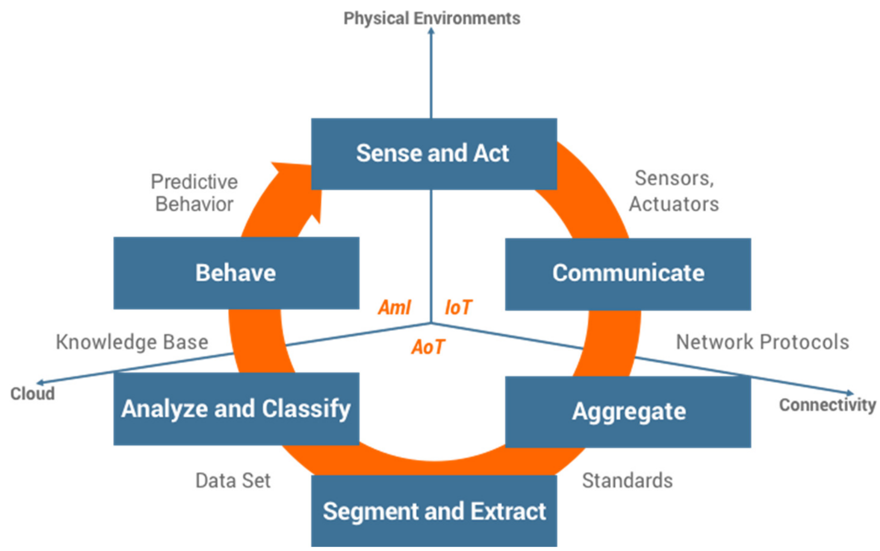

Figure 9. SoP Cycle

The SoP Cycle describes how data fusion applied to context oriented sensors creates value, and then, how extracting, classifying and analyzing these fusion increases that value, and then, how machine learning fine tunes this value to predict the desired shifting behavior. The cycle is multimodal and pervades three environments: the physical object within its environment, the local network (sometimes just the personal area network) and cloud computing. The cycle comprises six steps, namely, Sense and Act, Communicate, Aggregate, Segment and Extract, Analyze and Classify, and Behave, divided into three stages (IoT, AoT and AmI) described in detail in section 4.1.1, 4.1.2 and 4.1.3.

4.1.1.
IoT Stage

During the IoT stage, the system acquires data from sensors and then establishes a connection to transmit the data. In this stage, it is necessary to understand IoT enabling technologies. To that end, this section serves as a technical primer on

[6] https://dupress.deloitte.com/dup-us-en/deloitte-review/issue-17/value-creation-value-capture-internet-of-things.html

some of the technologies that currently drive the IoT. It meant to be used as a basic guide palette to choose the right IoT stack for a specific place of action.

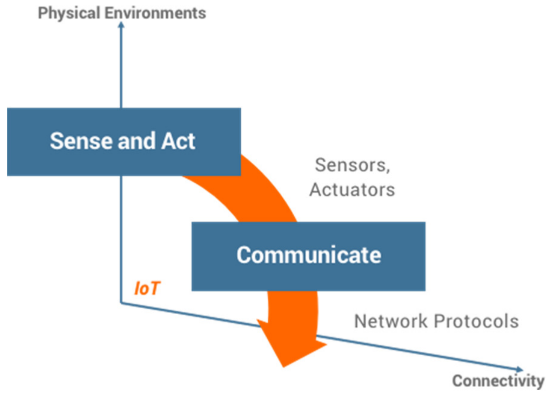

Figure 10. IoT Stage

The IoT Stage (Figure 10) comprises two steps: (1) Sense and Act, and (2) Communicate. These steps are described in detail in section 4.1.1.1 and 4.1.1.2.

4.1.1.1.
Sense and Act

Sensors have given the world a digital nervous system (Figure 11). Location data with GPS, beacons and RFID tags; eyes and ears using cameras and microphones; along with others sensory organs that can measure everything from temperature to pressure changes. Actuators are closing this loop by providing feedback and triggering action.

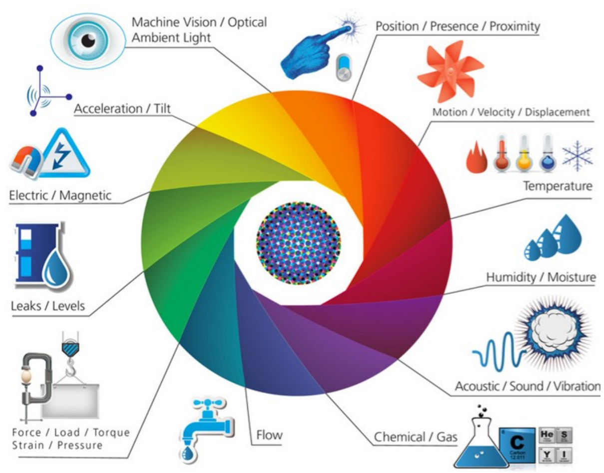

Figure 11. Sensors Palette (Postscapes IoT Infographic) [7]

[7] https://www.postscapes.com/what-exactly-is-the-internet-of-things-infographic/

The purpose of a sensor is to respond to some kind of a physical input property (stimulus) converting it into an electrical signal that is compatible with electronic circuits [35]. Sensors are classified into Contact or Noncontact, Passive (do not need any additional energy source and directly generates an electric signal in response to an external stimulus) or Active, Absolute or Relative measurement. In order to choose the right sensor for an IoT project it is important to understand its classification, specification, limitation, stimulus measurement and electrical signal output (Table 1) [35].

Table 1. Sensors Stimulus Description [35]

Stimulus	**Description**
Acoustic	Wave amplitude, phase, polarization Spectrum Wave velocity
Biological Chemical Electric	Biomass (types, concentration, states)
Chemical	Components (identities, concentration, states)
Electric	Charge, current Potential, voltage Electric field (amplitude, phase, polarization, spectrum) Conductivity Permittivity
Magnetic	Magnetic field (amplitude, phase, polarization, spectrum) Magnetic flux Permeability
Optical	Wave amplitude, phase, polarization, spectrum Wave velocity Refractive index Emissivity, reflectivity, absorption
Mechanical	Position (linear, angular) Acceleration Force Stress, pressure Strain Mass, density Moment, torque Speed of flow, rate of mass transport Shape, roughness, orientation Stiffness, compliance Viscosity Crystallinity, structural integrity
Radiation	Type Energy Intensity

Thermal	Temperature Flux Specific heat Thermal conductivity

A sensor does not function by itself; it is always a part of a larger system that may incorporate many other sensors, signal conditioners, signal processors, memory devices, microcontrollers, data recorders, and actuators. Table 2 [35] lists some types of sensors with representative description and example.

Table 2. Types of sensors with representative description [35]

Sensor Type	**Description**	**Examples**
Acoustic	Acoustic sensor measure sound levels and convert that information into digital or analog data signal	Microphone, Geophone and Hydrophone
Biosensors	Biosensors detect various biological elements such as organisms, tissues, cells, enzymes, antibodies and nucleic acids	Blood glucose biosensor, pulse oximetry, electrocardiograph
Chemical	Chemical sensors measure the concentration of chemicals in a system. When subjected to a mix of chemicals, chemical sensors are typically selective for a target type of chemical (CO_2, O_2, etc.)	Smoke Detector, Breathalyzer
Flow	Flow sensors detect the rate of fluid flow. They measure the volume or rate of fluid that has passed through a system.	water meter, anemometer, mass flow
Force	Force sensor detect whether a physical force is applied and whether the magnitude force is beyond threshold	force gauge, viscometer, tactile sensors
Humidity	Humidity sensor detect the amount of water vapor in the air.	Hygrometer, humistor, soil moisture
Light	Light sensor detects the presence of visible or invisible light	infrared sensor, photodetector, flame detector
Occupancy and motion	Occupancy sensor detect the presence of people and animals, motion sensors detect movement of people and objects	Electric Eye, LiDAR, Radar
Position	Position sensor measure the position of an object, the position sensors can be linear, angular or multi-axis	Potentiometer, proximity sensor
Pressure	Pressure sensors are related to force sensors and measure the force applied by liquids or gases	Barometer, piezometer
Radiation	Radiation sensor detect radiations in the environment	Geiger counter, neutron detector
Temperature	Temperature sensors measure the amount of heat or cold that is present in a system, can be contact or non-contact	Thermometer, calorimeter
Velocity and acceleration	Velocity sensor may be linear or angular, indicating how fast an object moves	Accelerometer, gyroscope

The technological complement to a sensor is an actuator, a device that converts an electrical signal into action, often by converting the signal to nonelectrical energy, such as motion. A simple example of an actuator is an electric motor that converts electrical energy into mechanical energy. Table 3 [35] lists some types of actuators with representative description.

Table 3. Actuators with representative description [35]

Name	Description
Brushless DC Servo	This synchronous electric motor features permanent magnet poles on the rotor, which are attracted to the rotating poles of the opposite magnetic polarity in the stator creating torque. It is powered by a DC current that has an electronically controlled commutation system instead of a system based on brushes. Current, torque, voltage, and rpm are linearly related. The advantages of a brushless motor include higher efficiency and reliability, reduced noise, longer lifetime (no brush erosion), elimination of ionizing sparks from the commutator, and an overall reduction of electromagnetic interference
Stepper	A type of brushless servo motor, this motor is generally electric and moves or rotates in small discrete steps. Stepper motors offer many advantages, such as dual compatibility with both analog and digital feedback signals. They can be used to easily accelerate a load because the maximum dynamic torque occurs at low pulse rates. Drawbacks of their use include low efficiency; much of the input energy is dissipated as heat and the inputs must be matched to the motor and load. The load should be carefully analyzed for optimal performance. Damping may be required when load inertia is exceptionally high to prevent oscillation.
Brushed DC Servo	The classic DC motor generates an oscillating current in a rotor with a split ring commutator, and either a wound or permanent magnet stator. A coil is wound around the rotor, which is then powered by a battery. The rotational speed is proportional to the voltage applied to it and the torque is proportional to the current. Speed control can be achieved by applying tape to the battery, varying the supply voltage, resistors, or electronic controls. The advantage to using a brushed motor over a brushless is cost. The brushless motor requires more complex electronic speed controls; however, a brushed DC motor can be regulated by a simple variable resistor, such as a potentiometer or rheostat. This is not efficient, but proves satisfactory for costsensitive applications.
AC Servo Motors	Used in applications that require a rapid and accurate response, these motors are basically two-phase, reversible induction motors that are modified for servo operation. AC Servo motors have a small diameter and high resistance rotors. This design provides low inertia for fast starts, stops, and reversals. AC Servo Motors can also be classified as asynchronous or synchronous.
Pneumatic	Powered by the conversion of compressed air, these actuators are used to control processes that require a quick and accurate response, but not a large amount of force. These compact and lightweight actuators are less energy efficient than electric motors.
Hydraulic	With the ability to convert hydraulic pressure and flow into torque and rotation, these actuators can be used when a large amount of force is needed. The most common example is a piston. This motor uses hydraulic fluid under pressure to drive machinery. The energy comes from the flow and pressure, not the kinetic energy of the flow.
Magnetic	Actuators which can be actuated by applying thermal or magnetic energy have been used in commercial applications. They tend to be compact, lightweight, economical and with high power density.

However, the choice of a specific sensor or actuator is not only a function of the signal to be measured (e.g., position versus motion sensors) or physical action to be taken. There are several generic factors that determine the suitability of a sensor or actuator for a specific application. These include, but, are not limited to the following:

- Accuracy: A measure of how precisely a sensor reports the signal or the actuator execute and action. For example, when the water content is 52 percent, a sensor that reports 52.1 percent is more accurate than one that reports it as 51.5 percent;
- Repeatability: A sensor's or actuator's performance in consistently reporting the same response when subjected to the same input under constant environmental conditions;
- Range: The band of input signals within which a sensor can perform accurately. Input signals beyond the range lead to inaccurate output signals and potential damage to sensors;
- Noise: The fluctuations in the output signal resulting from the sensor or the external environment;
- Resolution: The smallest incremental change in the input signal that the sensor requires to sense and report a change in the output signal; and
- Selectivity: The sensor's ability to selectively sense and report a signal. An example of selectivity is an oxygen sensor's ability to sense only the O2 component despite the presence of other gases.

Any of these factors can impact the reliability of the data received and therefore the value of the data itself or the action to be taken.

4.1.1.2. Communicate

Data created by sensors rarely attains its maximum value at the time and place of creation. The signals from sensors must be communicated to other locations for aggregation and analysis. This typically involves transmitting data over a network. However, sometimes the amount of data is so huge that transmitting it over a network is not a viable solution, it is necessary to evaluate whether the data must be processed in loco or over the cloud.

The choice of a network technology in an IoT project depends on the following features: geographical range to be covered, the amount of confidential data and the necessary security to transmit it, the data size and amount of power source available. When data have to be transferred over short distances (for example, inside a room), devices can use wireless personal area network (PAN) technologies such as Bluetooth and ZigBee. When data have to be transferred over a relatively bigger area such as an office, devices could use local area network (LAN) technologies. Examples of wired LAN technologies include Ethernet and fiber optics. Wireless LAN networks include technologies such as Wi-Fi. When data are to be transferred over a wider area beyond buildings and cities, an internetwork called wide area network (WAN). Examples of WAN technologies include 3G, 4G, LTE, Weightless, LoRa, SigFox (Figure 12).

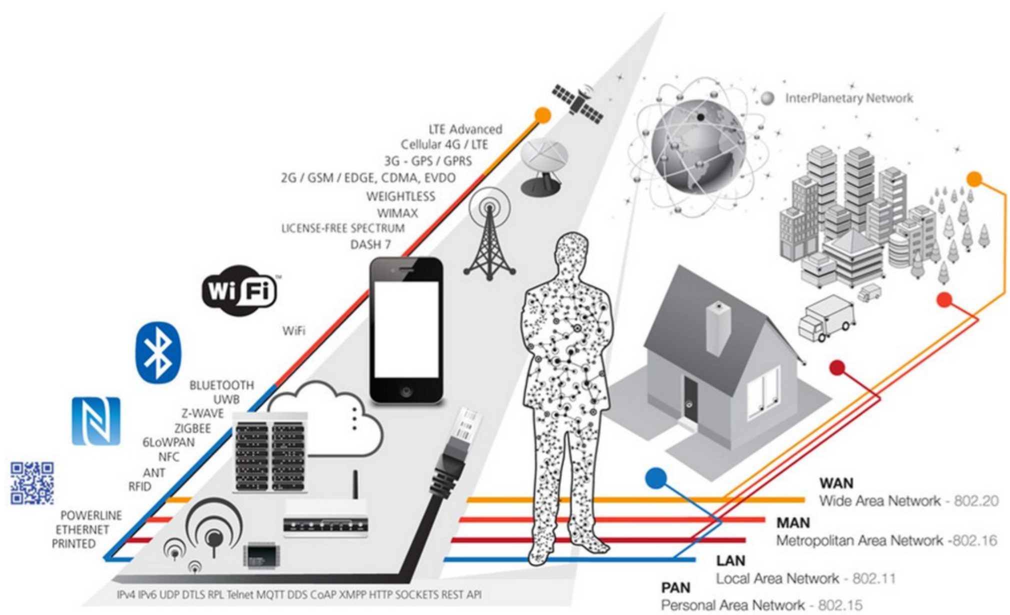

Figure 12. IoT network technologies (Postscapes IoT Infographic)[8]

The IoT network and protocols cover a huge range of industries and use cases that scale from a single constrained device up to massive cross-platform deployments of embedded technologies and cloud systems connecting in real-time. Tying it all together are numerous legacy and emerging communication protocols that allow devices and servers to talk to each other in new, more interconnected ways. At the same time, dozens of alliances and coalitions are forming in hopes of

[8] https://www.postscapes.com/what-exactly-is-the-internet-of-things-infographic/

unifying the fractured and organic IoT landscape. Table 4 (Postscapes IoT Protocols)[9] lists some protocols, transport layer and multi-layer frameworks available.

Table 4. Network Infrastructure, protocols and frameworks

Infrastructure	
IPv6	IPv6, is an Internet Layer protocol for packet-switched internetworking and provides end-to-end datagram transmission across multiple IP networks
6LoWPAN	6LoWPAN is an acronym of IPv6 over Low Power Wireless Personal Area Networks. It is an adaption layer for IPv6 over IEEE802.15.4 links. This protocol operates only in the 2.4 GHz frequency range with 250 kbps transfer rate.
UDP	A simple OSI transport layer protocol for client/server network applications based on Internet Protocol (IP). UDP is the main alternative to TCP and one of the oldest network protocols in existence, introduced in 1980. UDP is often used in applications specially tuned for real-time performance.
Discovery	
Physical Web	The Physical Web enables you to see a list of URLs being broadcast by objects in the environment around you with a Bluetooth Low Energy (BLE) beacon.
HyperCat	An open, lightweight JSON-based hypermedia catalogue format for exposing collections of URIs.
UPnP	Now managed by the Open Connectivity Foundation is a set of networking protocols that permits networked devices to seamlessly discover each other's presence on the network and establish functional network services for data sharing, communications, and entertainment.
Data Protocols	
MQTT	The MQTT protocol enables a publish/subscribe messaging model in an extremely lightweight way. It is useful for connections with remote locations where a small code footprint is required and/or network bandwidth is at a premium.
CoAP	CoAP is an application layer protocol that is intended for use in resource-constrained internet devices, such as WSN nodes. CoAP is designed to easily translate to HTTP for simplified integration with the web, while also meeting specialized requirements such as multicast support, very low overhead, and simplicity. CoAP is a RESTful protocol design to minimizing the complexity of mapping with HTTP, Low header overhead and parsing complexity, URI and content-type support, Support for the discovery of resources provided by known CoAP services. Simple subscription for a resource, and resulting push notifications, Simple caching based on max-age.
XMPP	An open technology for real-time communication, which powers a wide range of applications including instant messaging, presence, multi-party chat, voice and video calls, collaboration, lightweight middleware, content syndication, and generalized routing of XML data.
AMQP	An open standard application layer protocol for message-oriented middleware. The defining features of AMQP are message orientation, queuing, routing (including point-to-point and publish-and-subscribe), reliability and security.
Multi-layer Frameworks	

[9] https://www.postscapes.com/internet-of-things-protocols/

Alljoyn	An open source software framework that makes it easy for devices and apps to discover and communicate with each other
IoTivity	IoTivity is an open source project hosted by the Linux Foundation, and sponsored by the OIC.
IPSO Application Framework	This design defines sets of REST interfaces that may be used by a smart object to represent its available resources, interact with other smart objects and backend services. This framework is designed to be complementary to existing Web profiles including SEP2 and oBIX
Weave	A communications platform for IoT devices that enables device setup, phone-todevice-to-cloud communication, and user interaction from mobile devices and the web.

Even though network technologies have improved in terms of higher data rates and lower costs, there are challenges associated with interconnections, penetration, security, and power consumption:

- Interconnections: Metcalfe's Law [36] states that "the value of a network is proportional to the square of the number of compatibly communicating devices." There is limited value in connecting the devices to the Internet; companies may create enhanced value by connecting devices to the network and to each other. Different network technologies require gateways to connect with each other. This adds cost and complexity, which can often make security management more difficult;
- Network penetration: There is limited penetration of high-bandwidth technologies such as LTE and LTE-A, while 5G technology has yet to arrive. Currently, LTE accounts for only 5 percent of the world's total mobile connections. LTE penetration as a percentage of connections is 69 percent in South Korea, 46 percent in Japan, and 40 percent in the United States, but its penetration in the developing world stands at just 2 percent. In emerging markets, network operators are treading the slow-and-steady path to LTE infrastructure, given the accompanying high costs and their focus on fully reaping the returns on the investments in 4G technology that they made in the last three to five years;
- Security: With a growing number of sensor systems being connected to the network, there is an increasing need for effective authentication and access control. The Internet Protocol Security (IPSec) suite provides a

certain level of secured IP connection between devices; however, there are outstanding risks associated with the security of one or more devices being compromised and the impact of such breaches on connected devices. Maintaining data integrity while remaining energy efficient stands as an enduring challenge; and

- Power: Devices connected to a network consume power, and providing a continuous power source is a pressing concern for the IoT. Depending on the application, a combination of techniques such as power-aware routing and sleep-scheduling protocols can help improve power management in networks. Power-aware routing protocols determine the routing decision based on the most energy-efficient route for transmitting data packets; sleep-scheduling protocols define how devices can "sleep" and remain inactive for better energy efficiency without impacting the output.

4.1.2.
AoT Stage

While IoT Stage is all about embedding places and objects with sensors and actuators to connect them to the Internet, the AoT stage aggregates, and extracts what is necessary, classifies and analyzes the IoT descendant data. The AoT itself points out that data, must be processed and normalized before analyzed to be useful. It also suggests that analytics is place dependent, according to its sensors attributes, task and time.

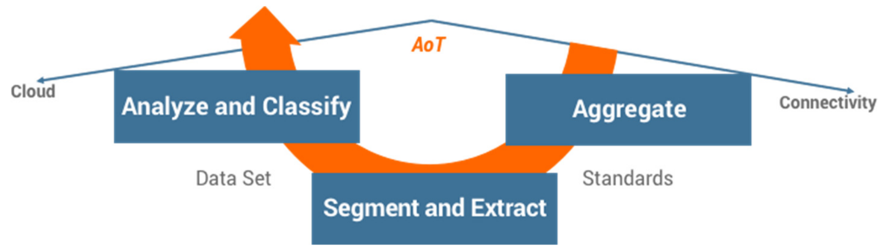

Figure 13. AoT Stage

The AoT stage (Figure 13) comprises three steps: (1) Aggregate, (2) Segment and Extract, (3) Analyze and Classify. These steps are described in detail in sections 4.1.2.1, 4.1.2.2 and 4.1.2.3.

4.1.2.1.
Aggregate

Aggregation refers to a variety of activities including data handling, processing, and storage. Data collected by sensors in different locations are aggregated so that meaningful conclusions can be drawn. Aggregation increases the value of data by increasing, for example, the scale, scope, and frequency of data available for analysis. Aggregation is achieved through the use of various standards depending on the IoT application at hand (e.g., data from multiple sources can be in different scales or time zones). Given that some sensors can provide multiple values (e.g., an acceleration sensor provides a 3D acceleration typically referred to as x, y, and z direction), or multiple sensors are jointly sampled, vector notation (1) is used to describe the sensor's output [37]:

$$(1)\quad \mathbf{s}_i = (d^1, d^2, d^3, \cdots, d^t), \text{ for } \quad i \quad = 1, \cdots, k$$

where **k** denotes the number of sensors and $\mathbf{d}^{\mathbf{i}}$ the multiple values at a time **t**. Each of the sensors is sampled at regular intervals, resulting in a multivariate time series. However, the sampling rates of different types of sensors can differ. For example, the typical sampling frequency for GPS is 5Hz, whereas acceleration is sampled at 25Hz and whether stations are sampled three or four times a day.

Sensors can also change their sampling frequency for other reasons, like power saving or due to OS requirements. In any case, t differs across $\mathbf{s}_i$, and synchronization across multimodal sensor data becomes a central technical issue. Moreover, raw sensor data can be corrupted by artifacts caused by a variety of sources (e.g., physical activity or sensor malfunction). AC power lines can cause electromagnetic interference with amplified electrical sensing techniques like Water Electro Conductivity (EC), pH, and so forth.

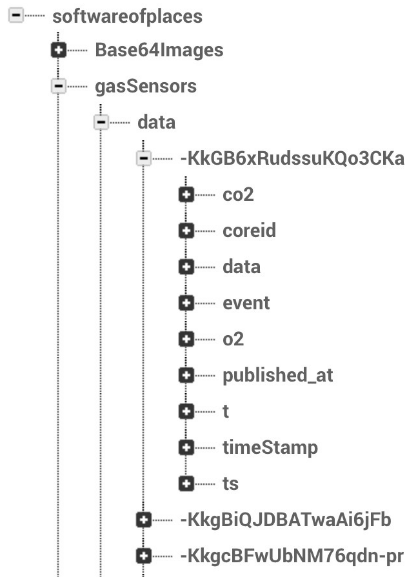

Figure 14. SoP Data Samples in a JSON tree structure

In this thesis, the time series was stored in a graph database (Figure 14) where each sample is a node with an associated hashed timestamp key in a JSON tree structure. A fetch is necessary in order to access $\mathbf{s_i}$ data at a location node $\boldsymbol{t}$ in the tree to retrieve all of its child nodes.

4.1.2.2. Segment and Extract

Segment and Extract is the process of selecting a subset of relevant features by reducing the signals from sensors into features that are discriminative. Features may be calculated automatically and/or derived based on expert knowledge. The first step is the preprocessing to synchronize time and to prepare the acquired signals for segmentation and feature extraction. This step transforms the raw multivariate and nonsynchronous time series data into a preprocessed time series ***D'*** (2) [37]:

$$(2)\ \boldsymbol{D}' = \begin{bmatrix} d'^{1}_{1} & \cdots & d'^{t}_{1} \\ \vdots & \ddots & \vdots \\ d'^{1}_{n} & \cdots & d'^{t}_{n} \end{bmatrix} = [\, d'_1, \cdots, d'_n]^T$$

where $\mathbf{di'}$ corresponds to one dimension of the preprocessed time series, **n** to the number of total data dimensions, and **t** to the number of samples. The transformation aims to enhance the robustness of the extraction by applying signal processing algorithms that reduce noise or filter out artifacts. At the same time, these algorithms need to preserve those signal characteristics that carry relevant information about the activities of interest. Preprocessing may involve calibration, unit conversion, normalization, resampling, synchronization, or signal-level fusion [38].

The second step is the data segmentation using a sliding window for identifying those segments of the preprocessed data streams that are likely to contain information about activities. In this approach, a window is moved over the time series data to “extract” a data segment for analysis. The window size directly influences the delay of the recognition system. The bigger the window size, the longer the system has to wait for a new segment to be available for processing. Also, the optimal (single) size is not clear a priori and can influence the recognition performance [39]. The size is subject to a tradeoff between segmentation precision and computational load. Although commonly used, a fixed-size sliding window is agnostic about the type and structure of the underlying time series data [37].

The third and last step is the feature extraction, for this thesis I am using the Signal-Based [40] features based on sensors spread in the m-PFAL. Signal-based features are mostly statistical features, such as the mean, variance, or kurtosis. These features are popular due to their simplicity as well as their high performance across a variety of activity recognition problems [37].

The higher the dimensionality of the feature space, the more training data is needed for model parameter estimation and, hence more computationally intensive the classification. Particularly for real-time processing on embedded systems, the objective is to minimize memory, computational power, and bandwidth requirements. It is therefore important to use a minimum number of features that still allow to achieve the desired target performance. Manual selection of such features is a difficult task. A large variety of methods for automatic feature ranking and selection has been developed [41]. Relevant features are automatically selected while ensuring generalization at the same time. After these tree steps, the result is a clean, structured and normalized dataset ready for analysis and classification.

4.1.2.3.
Analyze and Classify

Research in machine learning and computational statistics have been developing a large variety of inference methods. The choice of a particular method depends on the activity, environment and the phenomenon being measured. It is also necessary the understanding of what questions each method can answer.

Classification is the process of finding a set of models that describe and distinguish data classes or concepts, for the purpose of predicting the class of objects whose class label is unknown [42]. There are many methods to classify the data, including decision tree, regressions, K-Nearest Neighbor, Bayesian network, and support vector machines (Figure 15).

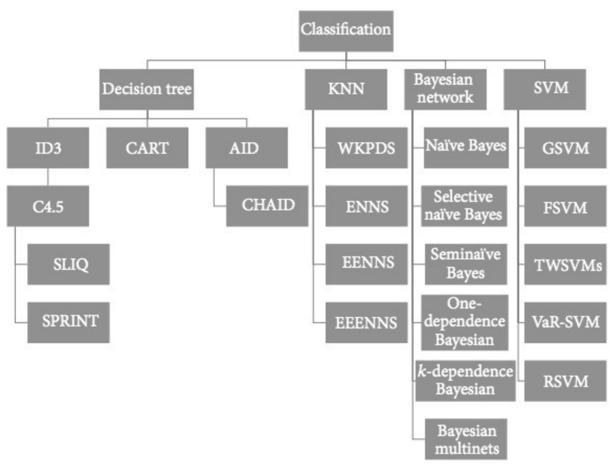

Figure 15. Research methods for Classification [42]

Table 5 [43] outline some machine learning methods for the IoT sensor data analysis with its application domains, pros. and cons.

Table 5. Classification Methods

Methods	Application Domains	Pros	Cons
Supervised Learning - Logistic Regression - Bayesian Networks, - Decision tree - Support vector machines	For situation where the feature set is easily identifiable, possible outcomes are known, and large data sets (for training as well) are available in numerical terms. (For example: activity recognition, missing value identification)	- Fairly accurate - Number of alternative models are available - Have mathematical and statistical foundation	- Require significant amount of data - Every data element need to be converted in to numerical values - Selecting feature set could be challenging - Can be more resource intensive (processing, storage, time) - Models can be complex - Difficult to capture existing knowledge

Unsupervised Learning - Clustering, - k-Nearest Neighbor	For situations where possible outcomes are not known (For example: unusual behavior detection, analyzing agricultural fields to identify appropriate location to plant a specific type of crop)	- No training data required - No need to know the possible outcome	- Models can be complex -Less semantic so less meaningful - Difficult to validate - Outcome is not predictable - Can be more resource intensive (processing, storage, time)

The choice for a particular method is subject to a tradeoff between computational complexity and recognition performance. With a view to classification on embedded systems with limited resources, the goal is to minimize computational complexity and memory requirements while still achieving high recognition performance. Feature selection allows one to identify contributing features during training and thereby reduce computational complexity during classification [44]. Therefore, machine learning methods are typically selected depending on the type of activity and the complexity of the feature space. They may also be selected based on other factors such as latency or online operation and adaptation.

4.1.3.
AmI Stage

AmI Stage shifts the system behavior based on the previous analyzes and programmed rules (Figure 16). It is the actuators triggering by analyzing the previous stage of the SoP Cycle. It also restarts the loop because action leads to creation of data.

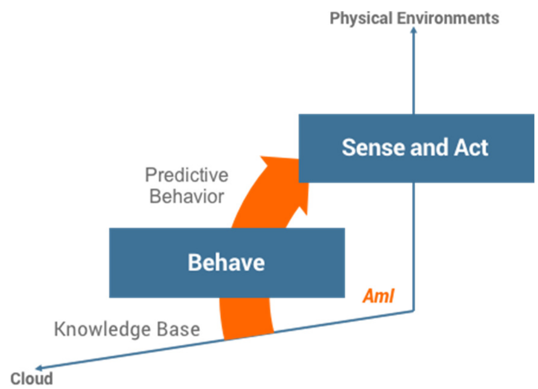

Figure 16. AmI Stage

Decision making in a predictive ambient intelligence environment is a key challenge in IoT. There are a variety of prediction and reasoning techniques, and a review of some techniques is presented below.

I. Case-Based Reasoning: Case-Based Reasoning (CBR) is a classification method that uses previous experiences to find a solution for current problem. Case-based reasoning is a type of analogy solution making. To use Case-Based-Reasoning, a four-step process module is necessary: Retrieve, Reuse, Revise and Retain.

   a) Retrieve: Given a target problem, retrieve from memory cases relevant to solving it. A case consists of a problem, its solution, and, typically, annotations about how the solution was derived.

   b) Reuse: Map the solution from the previous case to the target problem. This may involve adapting the solution as needed to fit the new situation.

   c) Revise: Having mapped the previous solution to the target situation, test the new solution in the real world (or simulation) and, if necessary, revise.

   d) Retain: After the solution has been successfully adapted to the target problem, store the result learning a new case in the knowledge base database.

II. Distributed Voting Approach: Due to the distributed nature of sensor networks in Ambient Intelligence, implementing distributed algorithms for learning approaches becomes attractive. Most of these algorithms use little computational power of individual sensors to construct a powerful learning approach in the whole network. The distributed voting algorithm proposed in [45] is one of these algorithms. In this algorithm, a tree structure of sensors as small computing devices and a powerful computing device as the root of this tree is constructed to solve a classification problem. Each sensor as a leaf of the tree uses neural network or decision tree approaches for local prediction. Due to the shortage of memory in sensory devices, all training data for different classes are stored in the tree root. During the learning process, each sensor receives training data from the root, then, each node can measure and classify one or more attributes in a local policy. Eventually, in a global prediction, the root receives local classification decisions from sensors and performs a global classification by applying a voting strategy. The distributed voting approach is categorized as a distributed approach. In spite of the distributed nature of this technique, a huge training data is stored in the root.

III. Reinforcement Learning: Reinforcement learning is a method that learns the relation between input and output with trial and error. In this method, a function called the reinforcement signal must be maximized [46]. Any significant difference between input signal and target signal is considered as a punishment; therefore, the value of the reinforcement signal decreases. On the other hand, a slight difference between input signal and target signal is considered as a reward; hence, the value of the reinforcement signal increases. As an example of reinforcement learning technique, [47] proposes an intelligent lighting control in which a multi-agent system controls lights. This technique concerns varying lighting preferences of different users for different tasks. An agent uses users' location and light readings as the state space for the learner and attempts to take actions that lead to appropriate light settings. For example, the absolute difference between the light intensity sensed by an agent before and after the user action is used as a negative reinforcement or punishment. Also, if an agent turns a light on and the user turns it off then the agent receives a negative reinforcement. In contrast, if a person does not change anything previously set by the agent, it receives a positive reinforcement as a reward.

IV. Fuzzy Rule Based Learning: Knowledge is represented by fuzzy rules and the learning process is an unsupervised algorithm [48]. In the learning process, input from sensors are sampled and transformed into fuzzy sets in a fuzzification phase. Then, the learning process compares the fuzzy inputs with stored fuzzy rules. Any significant difference between fuzzy inputs and stored fuzzy rules is considered as a punishment. On the other hand, a slight difference between fuzzy inputs and fuzzy rule is a reward to the fuzzy rule. For example, assume that an agent in the study room contains the following rule: If time is 8pm and the user is in study room then set the light intensity to 10. If, as a sample, the system recognizes that the user is in the study room and time is 8pm but the light of user's preference is 5, then the stored rule receives significant punishment, or it may be replaced with a new rule.

V. Manual Rules Database: This is the simplest and most straightforward methods of reasoning out of all of them. Rules are usually structure in an IF-THIS-THEN-THAT format. This is the most popular method of reasoning widely used at ifttt.com. It allows the generation of high level context information using low level context. IFTTT rules play a significant role in the IoT, being, in my own experience, the easiest and simplest way to model human thinking and reasoning in machines.

5 Research Cycle

In this section, I report on the five action research cycles I have run on the m-PFAL. However, before engaging in each action cycle research (Sections 5.3, 5.4, 5.5, 5.6 and 5.7) properly speaking, I needed to gain initial understanding on the action research methodology (Section 5.1) and the Plant Production Process (PPP) (Section 5.2) used in each cycle.

5.1. Action Research

Action Research allows the researcher to start from a particular problem identified in a real environment, and then the researcher acts and becomes part of the research environment, reporting his impressions of the problem and the solutions [49]. My aim is to advance the theory by practicing, which is done by acting in the context. The focus of this research is on understanding the problem and the actions taken to solve it within a particular real environment.

Action Research is typically performed in iterative cycles that successively refine the knowledge acquired in previous cycles. The execution of several cycles is seen as a way to increase the rigor of research, since it passes through new critical review in each cycle, that uncover errors, inconsistencies or biases previously not identified [50]. Action research cyclical process comprises five phases: diagnosing, action planning, action taking, evaluating, and specifying learning as observed in Figure 17.

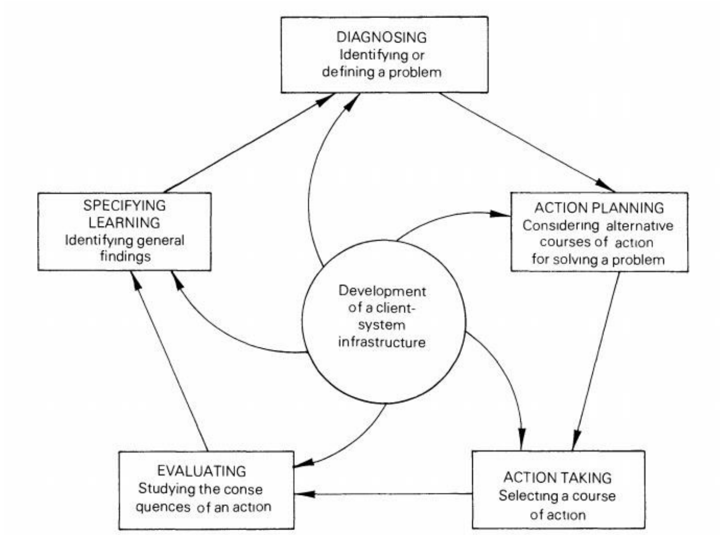

Figure 17. The cyclical process of action research [49]

Usually, all five phases are considered to be necessary for a comprehensive understanding of action research. However, action research projects may differ in the number of phases which are carried out depending on each research goal and circumstances [49]. The diagnosing stage involves the identification and definition of an improvement opportunity or a general problem to be solved in the client organization. The following stage, action planning, involves the consideration of alternative courses of action to attain the improvement or solve the problem previously identified. The action taking stage involves the selection and realization of one of the courses of action considered in the previous stage. The evaluating stage involves the study of the outcomes of the selected course of action. Finally, the specifying learning stage involves the study of the outcomes of the evaluating stage and, based on this study, knowledge building in the form of a model describing the situation under study. In the case of this thesis, the research was conducted between 2016 and 2017 and I carried out five action research cycles.

5.2. Plant Production Process (PPP)

PFALs are relatively new production systems, so the optimum production technologies have not yet been established [51], therefore, there is much room for improvement in the Plant Production Process (PPP). Each research cycle corresponds to one complete PPP. In this thesis, our PPP is based on Nunomura et al. [52]. A typical flow of operations in the plant production cycle is shown in Figure 18.

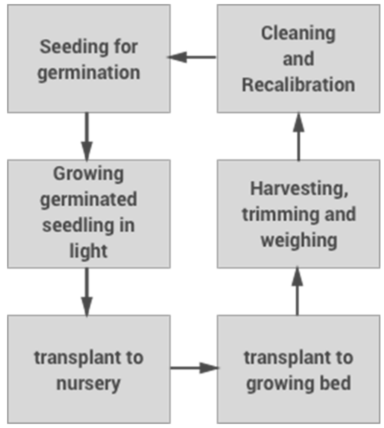

Figure 18. Plant Production Process Cycle

The cycle in Figure 18 starts with seeding for germination. Only lettuce (Lactuca sativa L. vari. crispa) was planted and a standard procedure [52] was adopted for producing lettuce seedlings in a PFAL, in order to achieve high percent seed germination and transplanting (98% or higher).

The seeding starts with the seed preparation, it is possible to choose between untreated or treated seeds. I chose to use treated coated seeds (Figure 19) for enhancing rapid and uniform germination.

Figure 19. Treated lettuce seeds

The seeds are placed in a phenolic foam seeding mat (30 × 46 × 2.0 cm) (Figure 20) consisting of 345 cubes (1.9 × 1.9 × 2.0 cm). The foam was embedded using water and in each cube a small hole was made (7mm in diameter, 1mm depth) using a 1cm syringe. The seed sits on the wet center of the hole. The foam is kept wet by four water drops connected to a peristaltic water pump and a moisture sensor for one week.

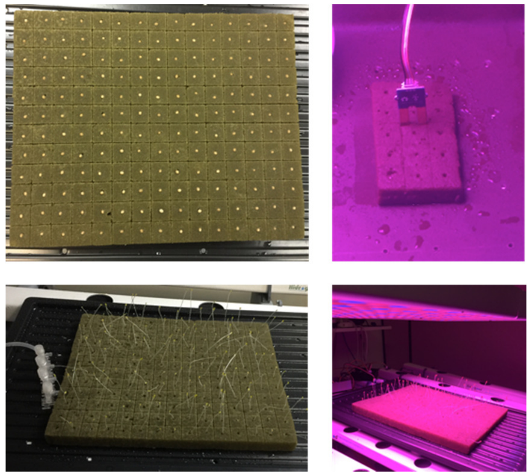

Figure 20. Phenolic foam seeding mat

After sitting three days in complete dark within a moist environment starts the seedling's photosynthesis process. Then it starts to receive light from the LED grow lights panel (Figure 21). At this stage, percent germination is expected to be 98% or higher. Otherwise, it is necessary to analyze what is causing the low percent germination in order to improve it.

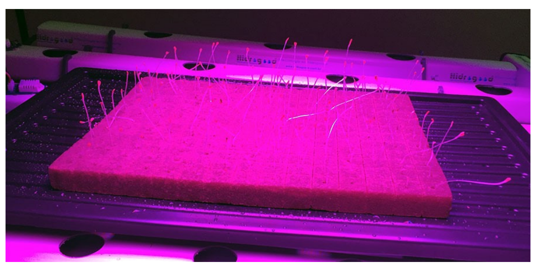

Figure 21. Lettuce seeds 96h after seeding

After the first week, the seedlings are suitable for their first transplanting. Each seedling within its cube is transplanted to the nursery culture panel for further growth (Figure 22). Then, the water and air pumps are turned on and the seedlings start to receive water and light for 18hs a day for 2 weeks.

Figure 22. Nursery culture panel

At the beginning of the third week, the seedlings are transplanted to their final and larger culture bed (Figure 23) to gain more space to grow. There remains for three or four weeks more.

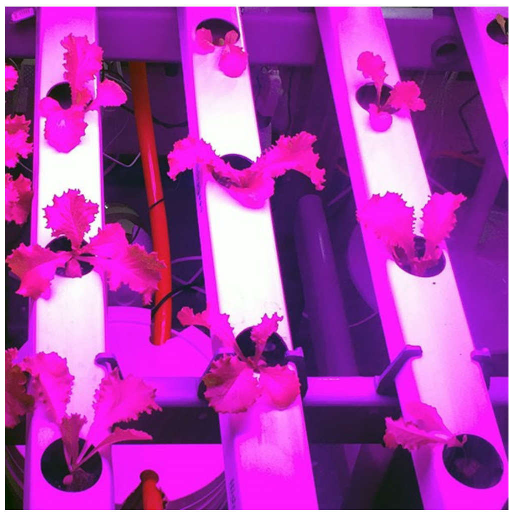

Figure 23. Culture panel last transplanting holes

During all this process, a special care with the water is needed. Every week the water must be changed and the water tank must be cleaned. Before replenished the hydroponic fertilizers, chlorine must be removed from the fresh water with a chlorine neutralizer, and then, the water pH must be adjusted to 6.

When the plants reach the desired size, before the harvest begins it is necessary to flush the system with water and a flush solution for five to seven days. Flushing exposes hydroponic plants and systems to fresh, clean water instead of the mineral-rich nutrient solution that they normally live on. It is recommended as a way of keeping the pump, pipes and other system parts free of an accumulation of excess mineral salts, dirt and other debris, such as algae. Flushing also allows plants to shed any buildup of salts they may have, making it easier for them to absorb the nutrients they need. After flushing, harvesting begins, roots are removed and some leafs are trimmed. Then, each plant is weighted, the culture panels and sensors are cleaned with chlorine and isopropyl alcohol and all sensors are recalibrated before restart the next action PPP.

5.3. Cycle 1 – Building the Prototype

On this first cycle, I aimed at building the first working prototype based on what I have learned from the plant production process from Nunomura et al. [52] and the PFAL design and management system from Sakaguchi et al. [53]. Hence, this was the longest research cycle, taking five months to build the m-PFAL and almost two months at the Plant Production Process (PPP).

5.3.1. Diagnosing

In order to prototype and manage a m-PFAL properly, it is necessary to understand some scientific and technical aspects related to PFALs. These aspects include hydroponics, environmental measurement and control, light sources and lighting system control, sanitation, plant ecophysiology and nutrition. In this thesis, I propose the following system structure (Figure 24) based on a simplified design and management system proposed by Sakaguchi et al. [53] for larger PFALs (the production management, plan and finance were removed).

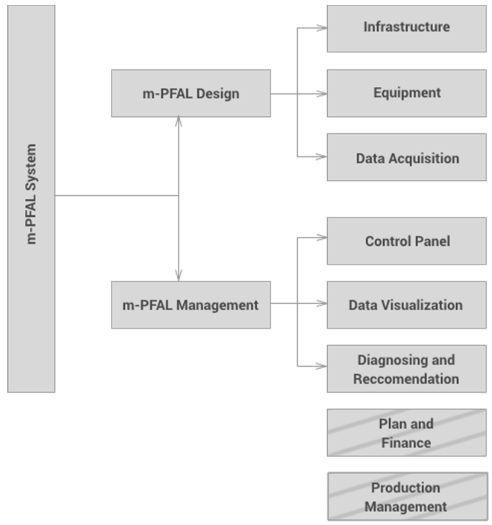

Figure 24. Structure of m-PFAL design and management system

The m-PFAL system comprises two parts: design and management. The mPFAL Design is further divided into three parts:

1. Infrastructure: related to building space, size, isolation, water, electricity and internet access,
2. Equipment: related to culture beds, pumps, electrical outlets, lighting system, fans, heaters and air conditions, and
3. Data Acquisition: related to environmental sensors measurement, air and water temperature, O_2 and CO_2, water pH and EC and Humidity.

The m-PFAL Management is divided into three parts:

1. Control Panel: related to control each equipment on the system,
2. Data Visualization: charts related to all acquired data, and
3. Diagnosing and Recommendation: related to improvements and required actions and interventions on the system.

5.3.2. Action Planning

To provide the necessary infrastructure to the system I set to build the prototype in the *SecondLab* at the *Núcleo de Informação Tecnológica* (NIT) building from the PUC-Rio Department of Informatics. The lab has a stabilized power grid, air condition and a Wi-Fi with internet access that can provide a public and static IPv4 IPs enabling worldwide access to the system. A corner of 5 m^3 (e.g., 1.6 × 1.4 × 2.3 m) was designated in the lab (Figure 25) to build the m-PFAL.

Figure 25. Second Lab reserved corner for the m-PFAL

Regarding the equipment and data acquisition, by revisiting the m-PFAL Type B requirements (air conditioned but some degree of natural ventilation and no CO_2 enrichment) on Section 2.2 and the SoP Cycle IoT Stage on Section 4.1 the following diagram was planned to meet the m-PFAL specifications (Figure 26):

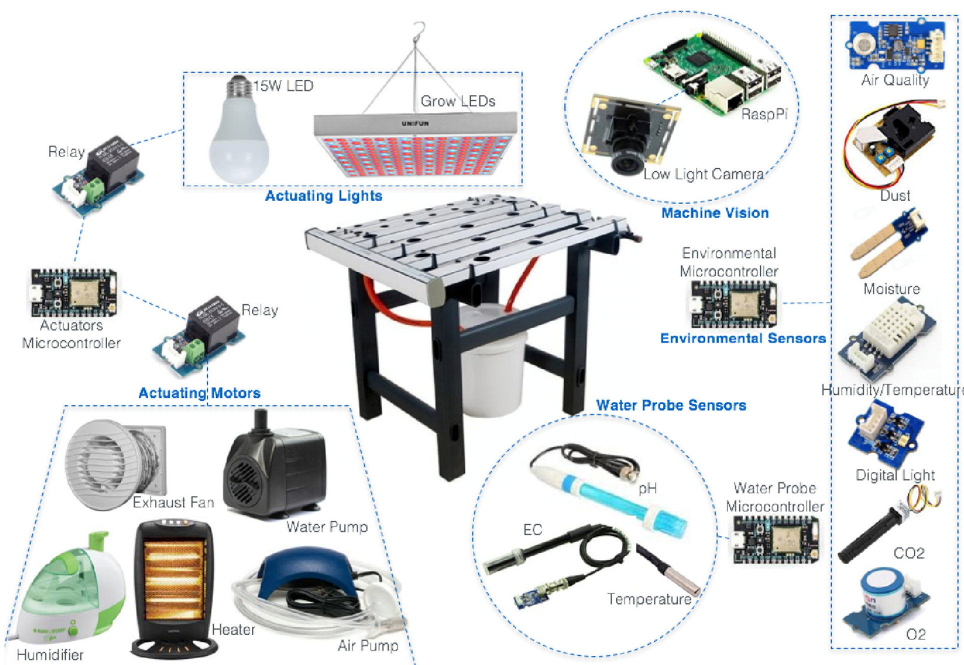

Figure 26. m-PFAL Prototype Diagram

Table 6 lists all devices, sensors and actuators bought to build the m-PFAL.

It also gives the price payed by each component in US Dollar in June of 2016.

Table 6. m-PFAL Bill of Materials

Device Name	Price	Description
Hydroponic Culture Bed	$170.00	one Hydrogood small culture bed measuring 90cm x 89cm x 79cm with room for fourteen seedlings in the nursery panel and fourteen at the large hole culture panel
Electricity Control	$107.20	eight regular electrical outlets ($2.00) were retrofitted by connecting them with eight relays (110V-250V at 10 amps), one Particle Relay Shield ($30.00) and two Particle Photon microcontroller ($19.00);
Lighting System	$79.98	two Erligpowht 45W LED Red Blue Hanging Light ($39.99) for indoor plant and one Philips LED 13.5W 6500K (19.99)
Water Pump	$10.50	Aleko G2950 submersible multi-function pump
Air Pump System	$12.78	one Tetra Whisper air pump for use in 10 gallon aquariums ($6.99) with an aquarium fish tank ceramic air stone diffuser - 40mm x 15mm ($2.49) and a Penn Plax standard airline tubing with 25 feet ($3.30)
Heater	$60.00	one oscillating halogen heater Cadence Aqc300
Humidifier	$32.00	one Black & Decker UC1000 humidifier
Exhaust Fan	$45.00	one Arkit exhaust fan with a 5/125mm duct and an air flow of 160m3/h
pH Sensor	$29.50	one Gravity Analog pH sensor (accuracy: ± 0.1 pH at 25°C, measuring range: 0 – 14 pH)
EC Sensor	$69.90	one Gravity Analog Electrical Conductivity sensor (accuracy: < ±10%, measuring range: 1 – 2000 micro-Siemens per centimeter or µS/cm)
O2 Sensor	$54.90	one Seeed Studio Grove O2 gas sensor (measurement range: 0-95%, Sensitivity: 0.05~0.15 mA)
CO2 Sensor	$89.00	one Seeed Studio Grove Carbon Dioxide Sensor - MH-Z16 (accuracy: 200PPM, measuring range: 0 – 2000 PPM)
Temperature & Humidity Sensor	$19.80	two Seeed Studio Grove temperature and humidity sensor pro - AM2302 (accuracy: ± 2%, measuring range humidity: 5%-99%, measuring range temperature: -40 – 80°C)
Waterproof Temperature Sensor	$6.90	one waterproof DS18B20 digital temperature sensor (accuracy: ± 0.5°C, measuring range: -55 – 125°C)
Digital Light Sensor	$19.80	two Seeed Studio Grove digital light sensor - TSL2561 (measuring range: 0.1 – 40000 LUX)
Moisture Sensor	$5.98	two Seeed Studio Grove moisture sensor (measuring range: 0 – 900)
Camera System	$110.00	one raspberry 2 and one USB Megapixel ELP 2.1mm wide lens night vision camera
Microcontroller	$79.60	four Particle Photon Wi-Fi microcontroller

Other equipment like hydroponic trays, structural frames, reservoir tubes, air stone, wires, insulating tape, hydroponic fertilizers, pH and EC calibrators fluids, manual pH probe had a total value of US$1800. The total cost of this prototype was around US$2800,00. Figure 27 show the planed m-PFAL electronics and connections diagram.

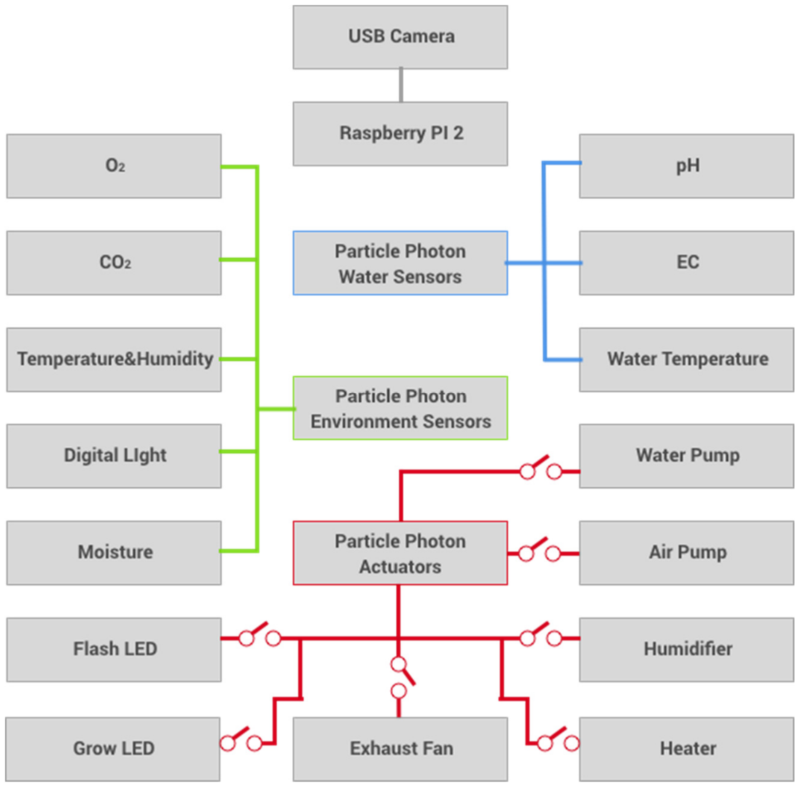

Figure 27. m-PFAL electronics and connections diagram

Regarding the m-PFAL management, in order to provide a control panel to operate the m-PFAL, the SoP website [10] should be used as web-based interface using the JavaScript language (AngularJS + NodeJS + Firebase).

5.3.3. Action Taken

A five-cubic meter greenhouse (1.6 × 1.4 × 2.3 m) was built (Figure 28). The hydroponic bed was bought in Brazil from Hidrogood website and after

[10] www.softwareofplaces.com

assembly, it was placed in the center of the greenhouse, and then, the water system was tested to find any leaks.

Figure 28. Greenhouse assembly

All sensors were bought abroad from Seeed Studio[11], Spark Fun[12] and DF Robot[13], taking three months to receive all of them. Because of the delivery delay, the first cycle initiated the PPP using only a temperature, humidity, digital light and moisture sensors (Figure 29).

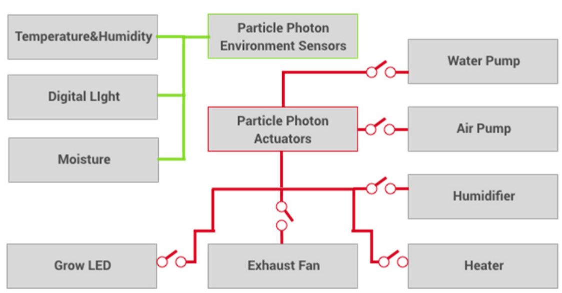

Figure 29. First Cycle electronics and connection diagram

All actuators (humidifier, fan, air and water pumps), except the lighting system were bought in Brazil. They were plugged in a custom-made power outlet using a copper cable (10mm X 1m), a relay (110V-250V at 10 amps) and a 3D printed enclosure, and then, one microcontroller (Particle Photon) was connected to all relays (Figure 30).

[11] www.seeedstudio.com

[12] www.sparkfun.com

[13] www.dfrobot.com

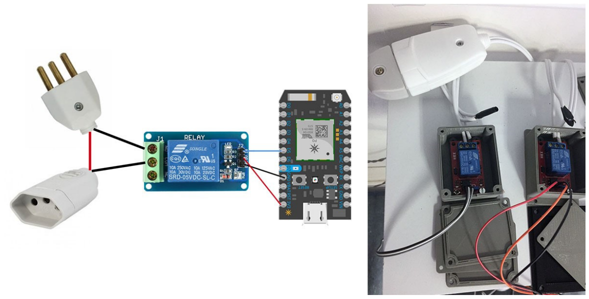

Figure 30. Power outlet electronics and connections diagram

In order to control all outlets and save the continuous sensors data stream the Particle Photon microcontroller was used as well as its proprietary cloud service. The Particle Photon connects to the Particle Cloud through its built-in Wi-Fi to publish data (pre-programed variables) and to subscribe functions to receive commands (e.g., turn on, turn off). The Particle cloud service is a web service that expose functions and variables to the web. It also allows one device to subscribe to other devices to receive their published data to trigger actions (Figure 31).

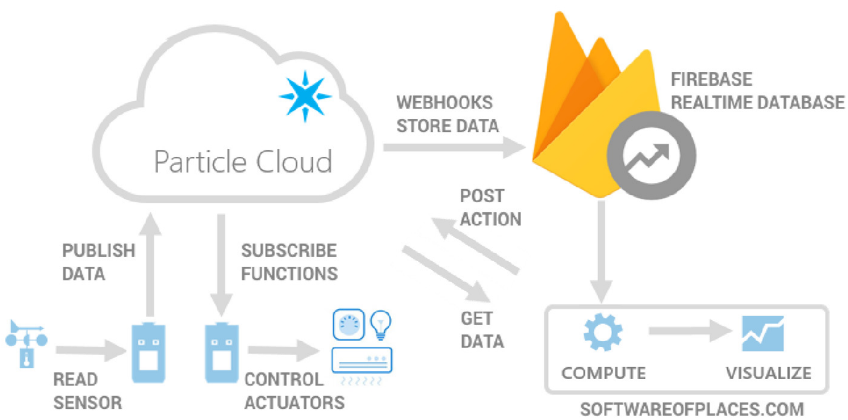

Figure 31. IoT Pub/Sub data model

Webhooks were programmed to store the sensors data stream in the Google Firebase Real-time Graph Database. To feed the SoP website a JSON semantic [54] [55] was written in order to describe to the website controller how to automatically recognize and control all sensors and actuators using POST and GET commands.

The JSON code in Figure 32 is an example of an actuator (exhaust fan) and an array of sensors (temperature, humidity and digital light).

```
[{
   name: 'Exhaust Fan', //DEVICE NAME
   description: 'Exhaust Fan at the door to remove hot air',//DESCRIPTION
status: 'Off',//DEVICE INITIAL STATE
   type: 'actuators', //DEVICE TYPE: SENSOR ||
ACTUATOR     commands: {//DEVICE COMMANDS LISTS
type: 'switch', // COMMAND TYPE        on: {//SWITCH ON
COMMAND            action: 'POST', //HTTP POST OR PUT
         url:'https://api.particle.io/v1/devices/{ID}/switch',//DEVICE URL
body: {//BODY TO SEND ON POST
            args: 'r0=on'//COMMAND TO BE SEND (turn on relay 0)
         }
      },
      off: {//SWITCH OFF COMMAND
action: 'POST', //HTTP POST OR PUT
         url:'https://api.particle.io/v1/devices/{ID}/switch',//URL
body: {//BODY TO SEND ON POST
            args: 'r0=off' //COMMAND TO BE SEND (turn off relay 0)
         }
      },
      status: { //GET DEVICE STATUS
action: 'GET', //HTTP POST OR PUT
         url:'https://api.particle.io/v1/devices/{ID}/relay0' //(GET relay 0 status variable)
      }
   }
},
{
   name: 'Environment Sensors', //SENSOR NAME
   dbKey: 'EnvirontmentSensorpack', //DEVICE KEY IN FIREBASE TO ACCESS DATA
type : 'sensors', //DEVICE TYPE
   description: Array of Sensors (Temperature, Humidity and Light)', //SENSORS
status : 'Off', //DEVICE INITIAL STATE
   sensors : [ //ARRAY OF SENSORS IN THIS DEVICE
      {
         name : 'Temperature', //SENSOR NAME
         description: 'Temperature Sensor', //SENSOR DESCRIPTION
parameter : 't', //READ PARAMETER
         scale : 'oC' //READ SCALE
      },
      {
         name : 'Humidity', //SENSOR NAME
         description: 'Humidity Sensor', //SENSOR DESCRIPTION
parameter : 'h', //READ PARAMETER
         scale : '%' //READ SCALE
      },
      {
         name : 'Digital Light', //SENSOR NAME
         description: 'Light Sensor', //SENSOR DESCRIPTION
parameter : 'l', //READ PARAMETER
         scale : 'lux' //READ SCALE
      },
   ],
   accesData : {//DIRECT ACCESS DATA ON THE MICROCONTROLLER
action : 'GET', //HTTP GET
      url : 'https://api.particle.io/v1/devices/{DeviceId}/AllSensorsData'
   }
}]
```

Figure 32. A JSON describing how to read and control an actuator and an array of sensors The AngularJS controller was programmed to read this JSON to create an interface in order to build a control

panel that describes the sensor/actuator, report its status and show available commands (Figure 33).

Figure 33. Dynamic Control Panel Created by the JSON Sensor/Actuator semantic

This control panel received an authentication page at the SoP web site in order to protect the manual control over all devices (Figure 33) and the real-time feed chart (Figure 34).

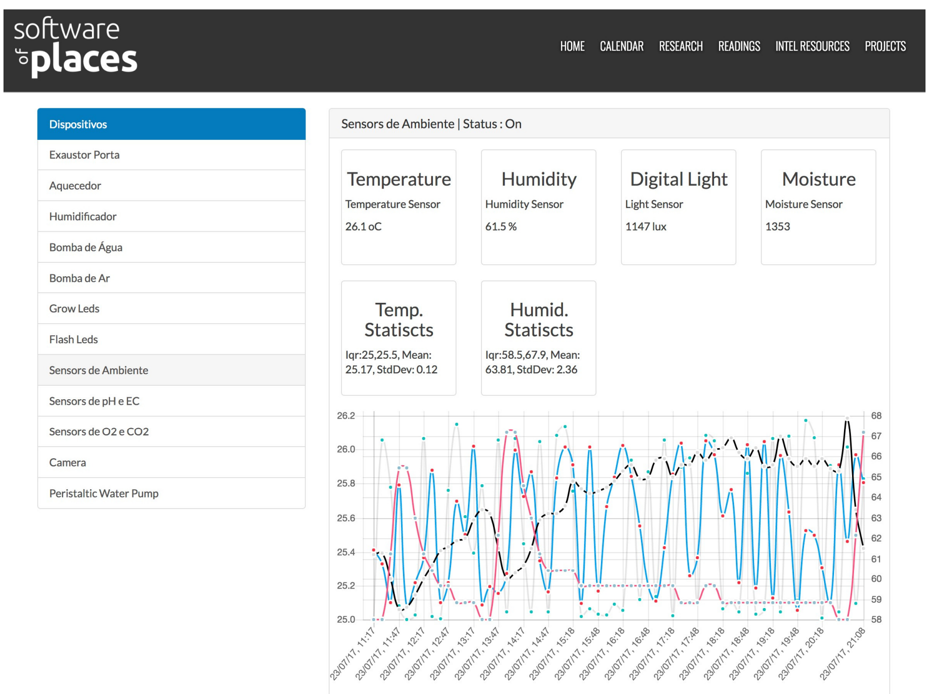

Figure 34. An example of the environment sensors feed chart

An IFTTT loop was programmed to run every ten minutes at the Particle Photon Relay microcontroller in order to the system follow the Plant Production Process (PPP) rules at certain points in time (e.g., LED grow lights and pumps must be ON between 5AM and 11PM, any other time must be OFF). Besides the lighting and pumps control, the IFTTT rules should keep the temperature between 22-25ºC and humidity above 70% (Figure 35).

```
function ifttttPPPRules () {
        const time = getTime();
        const temperature = getTemperature(); const humidity =
        getHumidity();
        // TIME RULES if(time >= '5:00'  || time <= '23:00') {
        setWaterPump('on');   setAirPump('on');
        setGrowLight('on');
        } else {
          setWaterPump('off');   setAirPump('off');
        setGrowLight('off');
        }
        // TEMPERATURE RULES if(temperature <
        22) {   setHeater('on');
        setExhaustFan('off'); } else if (temperature >
        25) {   setHeater('off');
        setExhaustFan('on');
        } else {
          setHeater('off');   etExhaustFan('off');
        }
        // HUMIDITY RULES if(humidity < 70) {
        setHumidifier('on');
        } else {   setHumidifier('off');
        }
}
//EXECUTE EVERY 10MIN setInterval(iftttPPPRules, 600000)
```

Figure 35. IFTTT Loop Rules from Cycle 1

At this point the m-PFAL was operational, the sensors used in this cycle did not need any calibration, all microcontrollers had a stable connection to the Internet, the system was saving its data every ten minutes and actuators were responding properly to the control panel, therefore, the first PPP was ready to start.

Table 7. Cycle 1 nutrients feed chart for a fourteen-liter water tank

WEEK #		**1**	**2**	**3**	**4**	**5**	**6**
GROWTH STAGE		Seedling	Early Growth	Late Growth	Transition	Early Bloom	Early Bloom
Base Nutrients	FloraMicro	8ml	30ml	40ml	32ml	32ml	32ml
	FloraGro	8ml	45ml	40ml	32ml	32ml	32ml
	FloraBloom	8ml	10ml	20ml	32ml	32ml	32ml

The seedling nutrition followed in this cycle was according General Hydroponic feed chart program[14] for recirculating nutrients. All hydroponic nutrients (FloraMicro, FloraGro and FloraBloom) were replenished every week (Table 7) after the entire water tank from previous week was discarded, filled with fresh water and the pH calibrated.

5.3.4. Evaluating

The cycle started on Saturday 17/12/2016, the Plant Production Process (PPP) took 45 days ending on Thursday 31/01/2017. Unfortunately, the seedlings died after three days from lack of water (Figure 36). That happened because the tube that connects the water pump to the culture bed lost its grip and popped out from the pump, halting the water circulation.

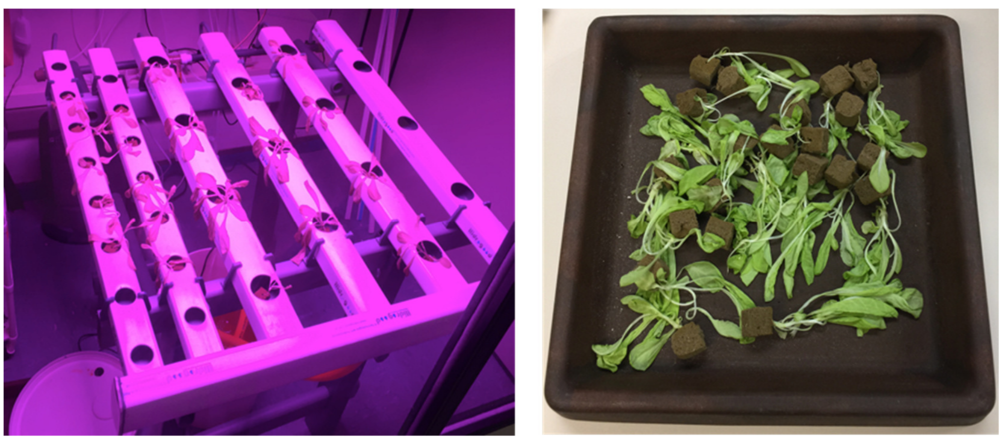

Figure 36. Seedlings from the first cycle died after 6 weeks

In this 45 days cycle the system did not disconnected from the Particle Cloud service at any time, all data were correctly saved in the database, there were no leaks on the water system, and the IFTTT loop was turning on the lights and pumps at the correct time. The water pH was measured manually with a swimming pool pH test kit and was stable at 6.5.

[14] https://goo.gl/3uxFw1

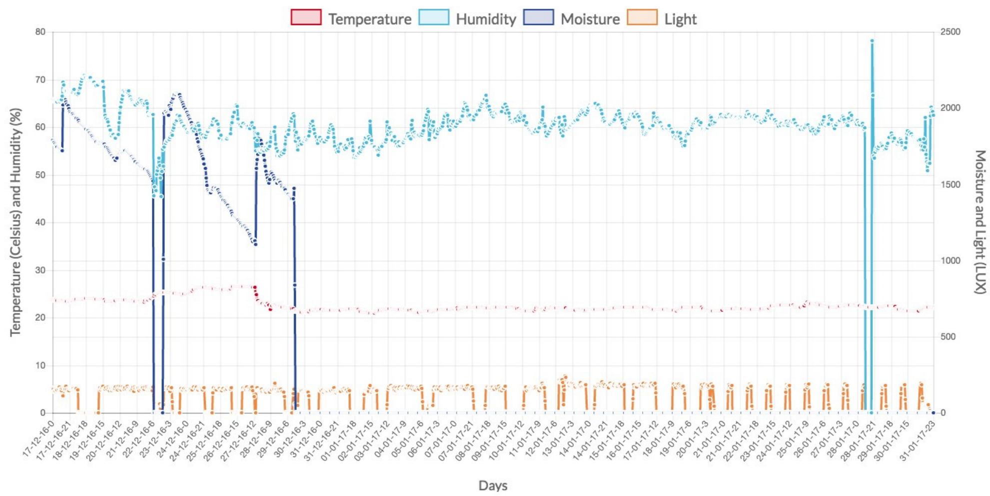

Figure 37. Environment Sensors chart (temperature, humidity, moisture and light) from Cycle 1 mean value by hour

Figure 37 plots the entire dataset generated during this cycle. It represents the mean value publish by hour from the temperature, humidity, moisture and digital light sensors in 45 days. In order to measure whether there is water flowing in the system, the moisture sensor was placed within the first culture panel hole. The temperature, humidity and light sensors were placed on the top right corner of the culture panel.

5.3.5. Specifying Learning

This first cycle was a successful failure. The IoT part of the system worked as expected (sensors were online, actuators were responding). The AoT part was aggregating the data from sensors as mentioned on Section 4.1.2.1 in a JSON tree at the graph database. The system behaved properly according to the IFTTT rules. A median filter was applied to remove outliers and to smooth the signal plotted on the graph by minifying the quantity of plots by hour to the day median value (Figure 38).

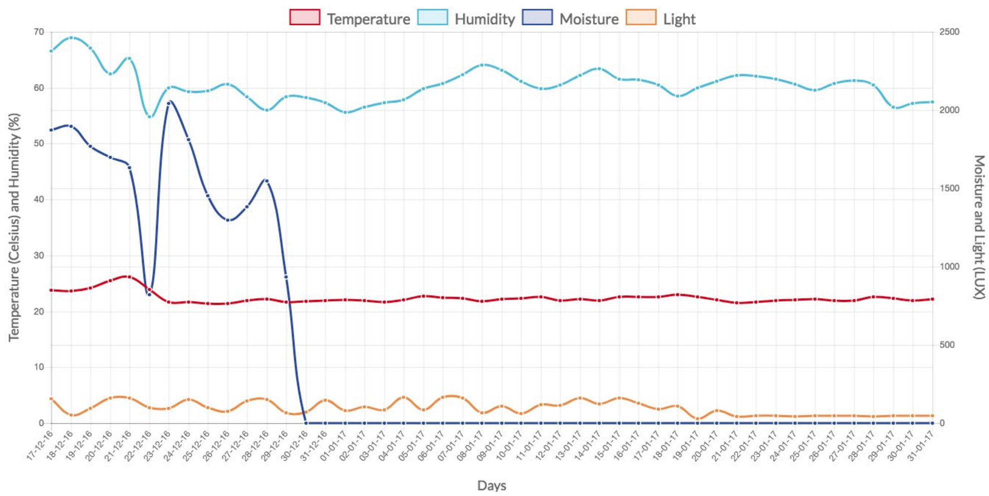

Figure 38. PPP environment sensors from Cycle 1 median value by day

By analyzing the outcome, I learned that the sensors were misplaced. The digital light sensor was too far from the LED grow lights panel, its value should be between 800 to 2000 lux. The water stream made the moisture sensor slide, making its probes losing contact with water. Because there was no direct human intervention during the cycle, just a manual water change every Wednesday, neither I nor the system could know whether there was a water stream. When the tube disconnected from the water pump I was abroad and, at the time, by looking at the real-time data stream at the SoP website (Figure 38) I could not do anything remotely by only operating with the tools I had at the time in the control panel, there was not enough data to make a decision.

This cycle was also a prove of concept, it was summer in Rio de Janeiro, the weather was very hot and stuffy, I was not sure that the prototype could grow anything by itself in a lab with a microclimate so distance from nature and I never had any experience in farming nor hydroponic farming. It was relief to know that with just a few changes the system could work properly.

5.4. Cycle 2 – Fortifying the Prototype

The second cycle started on Wednesday 15/03/2017, the Plant Production Process (PPP) took 70 days ending on Wednesday 24/05/2017. In this cycle, I wanted to strengthen all connections, welds and bolts. The sensors missed on first cycle (O_2, CO_2, pH, EC) arrived three weeks after the cycle begun. I decided to start the cycle without them and installed them as the shipment arrived.

5.4.1.
Diagnosing

In the previous cycle the system failed due to a weakness in the water pump tube connection. To prevent it to happen again all tubes, connections, weld and bolts needed to be fortified. The sensors also needed to be reallocated in other to capture the right stimulus from the LED grow lights panel and the water stream.

The camera had to be installed in order to save the system visual status in the dataset and to help remotely manage the system online. New sensors needed to be placed on the system in order to have more information to help diagnose what is happening with the seedlings.

5.4.2.
Action Planning

Insulation tape, cable ties, screws and double-sided tape needed to be bought in order to hold everything on its place. A camera and a water probe holder needed to be 3D printed in order to secure a proper and safe position for the sensors. The moisture sensor needed a backup sensor in order to double check whether there is a water stream. All sensors must be installed and placed over the hydroponic bed near the seedlings.

The Communication in the IoT Stage (Section 4.1.1.2) must be revisited to study a new architecture to manage the Raspberry Pi and the camera. Although the Raspberry Pi is a small single-board computer, it has to be treated as an IoT microcontroller. Its OS and running applications must be managed and modified as easy as sending new Arduino instructions to a microcontroller.

5.4.3.
Action Taken

The camera and water probe sensors (pH and EC) holders were downloaded from a 3D design community called Thingverse[15], I chose 3D models under the Creative Commons license, meaning that I could use and alter any design. The

[15] https://www.thingiverse.com

camera was placed on top of the LED grow lights panel, the water probes holder was placed within the water tank right over the water level (Figure 39).

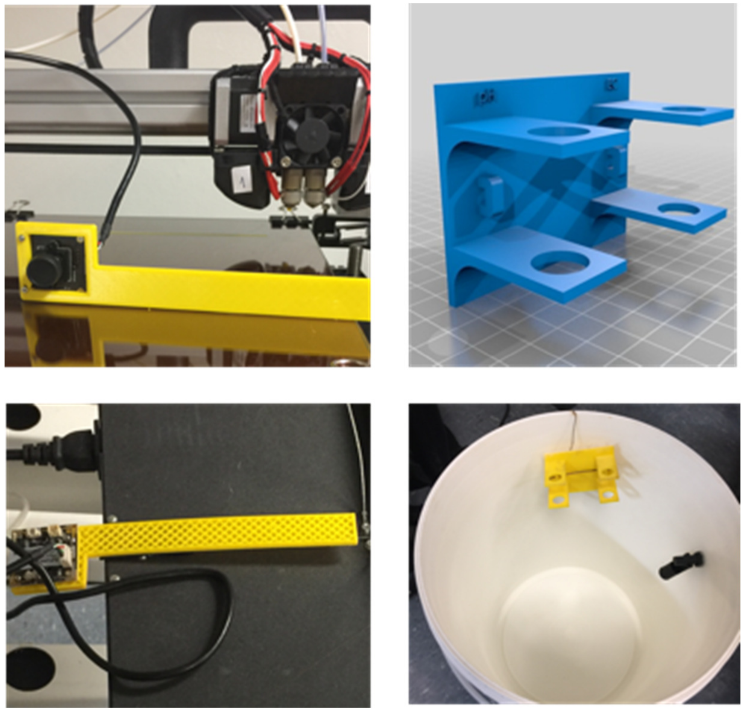

Figure 39. 3D Printed holders

Cable ties over the water tubes connected to the water pump were fastened, all 3D printed enclosures for the smart outlets and microcontroller were fixed using double-sided tape over a shelf. The SoP Cycle sensor pallet (Section 4.1.1.1) was revisited in order to look for a new water flow sensor.

An analog fluid sensor was bought to measure the water flow on the principle of the Hall Effect. According to it, a voltage difference is induced in a conductor transverse to the electric current and the magnetic field perpendicular to it. Here, the Hall Effect is utilized for measuring the flow using a small propeller shaped rotor which is placed in the path of the flowing liquid. The liquid pushes against the fins of the rotor, causing it to rotate. It is an arrangement of a current flowing coil and a magnet connected to the shaft of the rotor, thus a voltage is induced as this rotor rotates. In this flow meter, for every liter of liquid passing through it per minute, it outputs is about 4.5V.

When the pH and EC sensor arrived, calibration was needed with two specific chemical solutions for each one (Solutions with pH 4 and 7 for the pH sensor and solutions with 1413 and 12880 µS/cm for the EC sensor). To double check whether the pH sensor is calibrated a manual pH probe was bought for periodic testing. Seedlings are sensitive to water pH, water sensors need special attentions because they are analog, sensitive to temperature changes and easy to decalibrate due to minerals over its thin electrode membrane.

In order to make the wide-angle camera plugged into the Raspberry Pi to be controlled over the web, an operational system named ResinOS was installed in the Raspberry Pi. The ResinOS is optimized for running Docker containers on embedded devices in order to build a software architecture that allows iterative development over an IoT device, safely deploys, remote device management and monitoring (Figure 40).

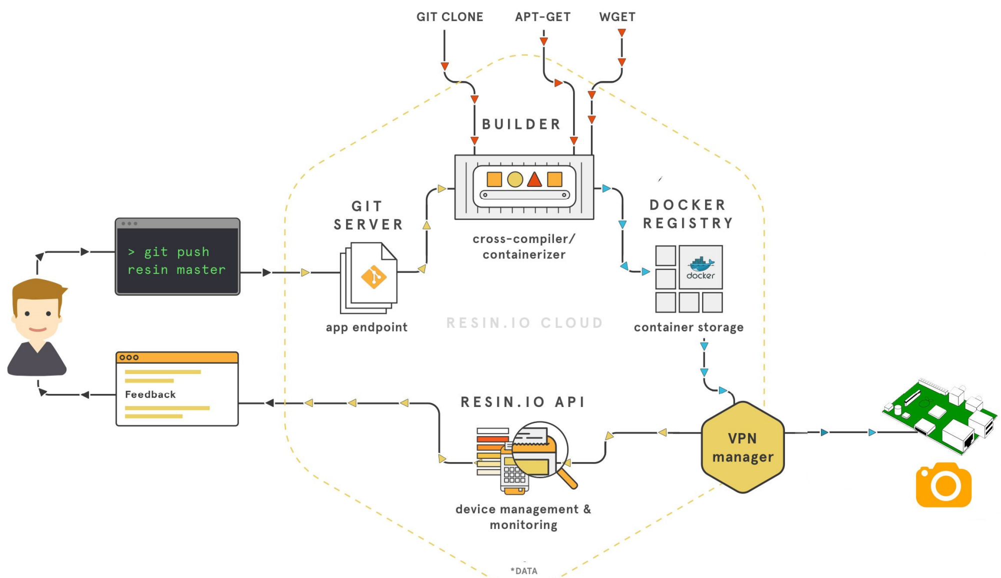

Figure 40. Resin.io Software Architecture

Resin.io is a platform that encompasses client, server, and device software. It lets the developer push its code using git, and then, the server receives the latest commits of the deployed code, and then, the codebase is passed into the builder, which build the code into a Docker container, and then, the container is stored in a container index. After that an agent running on the Raspberry Pi is alerted to download the application container image and to install it on the device. All this communication is running through a secure VPN between the device and the resin.io cloud giving the device a unique URL so it can be web-accessible from anywhere.

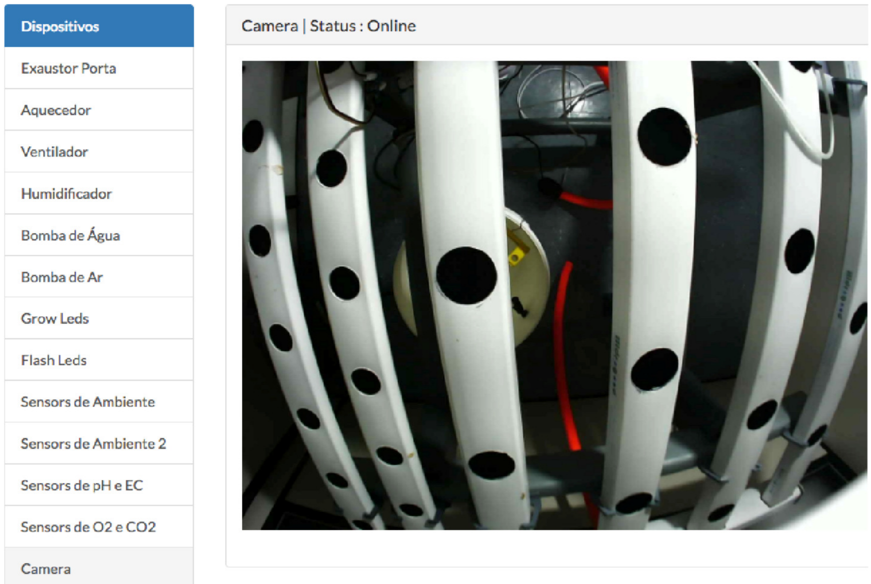

Figure 41. m-PFAL camera control panel

Using the Resin.io architecture the m-PFAL management system can send a POST command to the Raspberry Pi take a picture to save on the database or to show on the control panel (Figure 41). An auxiliary light (Philips 13W white LED) was installed to act as a flash light when the LED grow lights are off. The flash light goes on seconds before the camera takes a picture and goes off right after that.

Table 8. Cycle 2 nutrients feed chart for a fourteen-liter water tank

WEEK #		**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
GROWTH STAGE		Seedling	Early Growth	Late Growth	Transition	Early Bloom	Early Bloom	Mid Bloom	Mid Bloom
Base Nutrients	FloraMicro	8ml	30ml	40ml	32ml	32ml	32ml	32ml	32ml
	FloraGro	8ml	45ml	40ml	32ml	32ml	32ml	15ml	15ml
	FloraBloom	8ml	10ml	20ml	32ml	32ml	32ml	45ml	45ml

The seedlings were fed with hydroponic nutrients every week according to Table 8, at the last week they were fed only with fresh water and 7ml of Flawless Finish[16] flushing solution from Advanced Nutrition. This cycle had the same IFTTT loop from cycle one (Figure 35).

5.4.4. Evaluating

This cycle was successfully completed, fifteen lettuces were harvested with the average weight of fifty grams (Figure 42).

[16] http://www.advancednutrients.com/products/flawless-finish/

Figure 42. Lettuce harvested from cycle 2

When the cycle started, right on the first day, the system had a major water leak. The flow sensor spin rate dramatically decreased at the end of the first day making the output rate much slower than the input rate from the pump, causing a flood in the system. The fluid sensor was quickly removed and the PPP proceeded without it. Figure 43 represents the environment sensors mean value by hour during all cycle two. It shows that the temperature was stable, the light sensor position was adjusted during the cycle making the sensor value going over 2000lux.

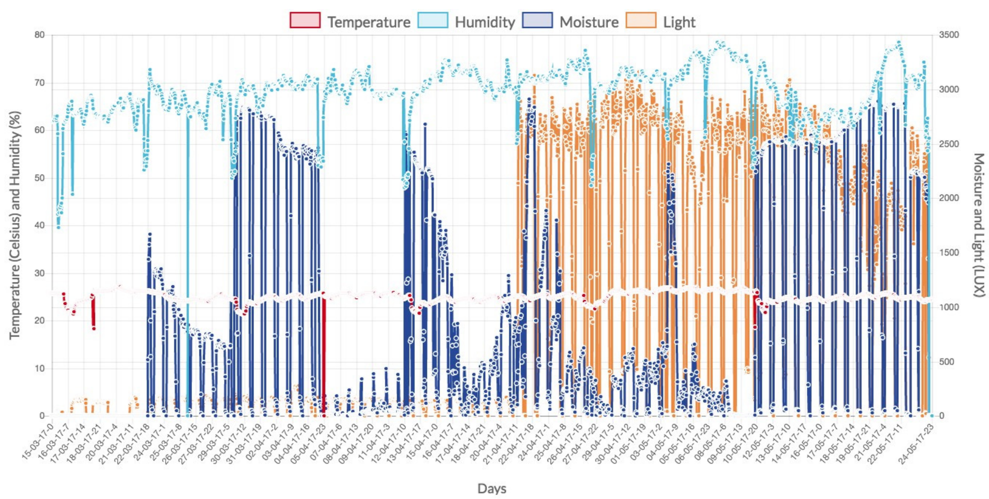

Figure 43. Cycle 2 environment sensors chart (temperature, humidity, moisture and digital light) mean value by hour

The water sensors (pH, EC and temperature) were installed within the water tank and places over the 3D printed holder. Their data are represented in Figure 44 showing that the pH and temperature were stable and with expected values but the EC sensors values was too low.

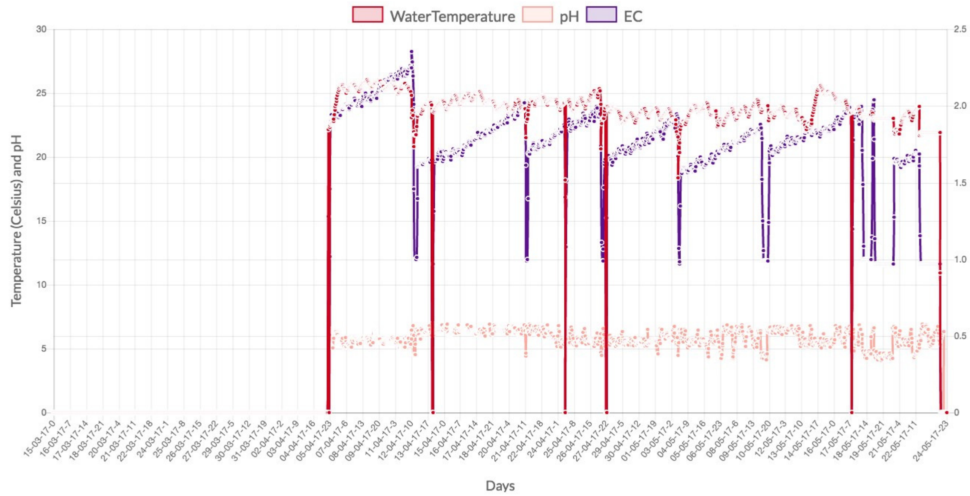

Figure 44. Cycle 2 water sensors chart (temperature, pH and EC) mean value by hour

The gas sensors (O_2 and CO_2) were also installed over the hydroponic bed, there were fixed with a double-side tape over a 3D printed enclosure near the seedlings. Figure 45 represents the gas sensors mean value by hour during all cycle two showing the O_2 concentration growing over the time and the CO_2 with drastic values changes (at the moment I did not know what that meant).

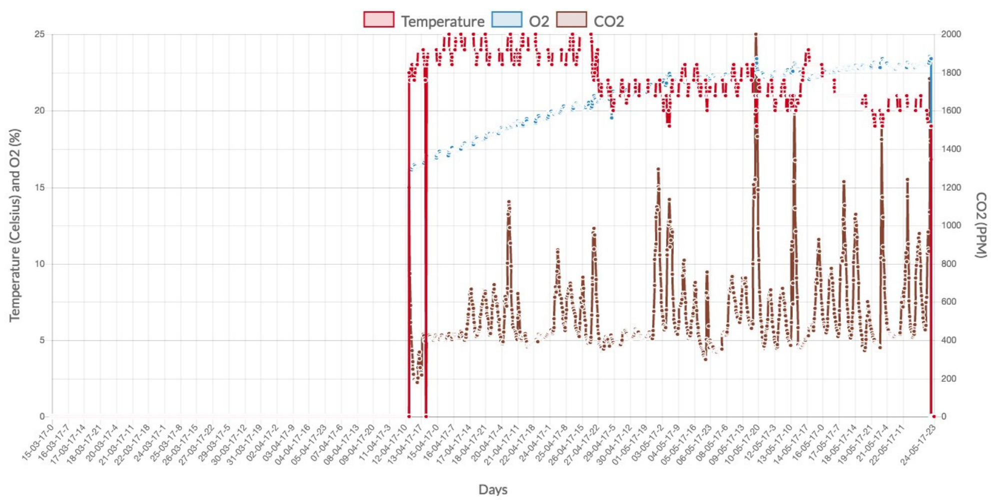

Figure 45. Cycle 2 gas sensors chart (temperature, O_2 and CO_2) mean value by hour

The CO_2 sensor has also a built-in temperature sensor. Its data was also being saved together with the CO_2 concentration. The camera was working well and all sensors were operational.

<table>
<tr><th>Date</th><th>Image</th><th>Temperature1</th><th>Temperature2</th><th>Water Temperature</th><th>pH</th><th>EC</th><th>O2</th><th>CO2</th><th>Humidity</th><th>Moisture</th><th>Light</th><th>Action</th></tr>
<tr><td rowspan="24">13-04-17</td><td rowspan="24"></td><td>H0 : 24 | 23.95</td><td>H0 : 22 | 22</td><td>H0 : 23.5 | 23.44</td><td>H0 : 6.28 | 6.11</td><td>H0 : 1.62 | 1.62</td><td>H0 : 98.6 | 98.56</td><td>H0 : 301 | 273</td><td>H0 : 65.59 | 66.15</td><td>H0 : 241 | 156.5</td><td>H0 : 0 | 0</td><td>Alive</td></tr>
<tr><td>H1 : 24 | 23.89</td><td>H1 : 22 | 22</td><td>H1 : 23.44 | 23.44</td><td>H1 : 6.25 | 5.94</td><td>H1 : 1.62 | 1.64</td><td>H1 : 98.56 | 98.53</td><td>H1 : 256 | 241</td><td>H1 : 66.25 | 66.4</td><td>H1 : 134 | 77</td><td>H1 : 0 | 0</td><td>2</td></tr>
<tr><td>H2 : 23.89 | 23.89</td><td>H2 : 22 | 22</td><td>H2 : 23.44 | 23.38</td><td>H2 : 5.93 | 5.88</td><td>H2 : 1.64 | 1.62</td><td>H2 : 98.56 | 98.56</td><td>H2 : 229 | 212</td><td>H2 : 66.59 | 66.95</td><td>H2 : 54 | 100</td><td>H2 : 0 | 0.5</td><td>Save Item</td></tr>
<tr><td>H3 : 23.89 | 23.89</td><td>H3 : 22 | 22</td><td>H3 : 23.38 | 23.38</td><td>H3 : 5.84 | 5.77</td><td>H3 : 1.63 | 1.63</td><td>H3 : 98.56 | 98.56</td><td>H3 : 201 | 191</td><td>H3 : 67.9 | 67.55</td><td>H3 : 89 | 93.5</td><td>H3 : 0 | 0</td><td></td></tr>
<tr><td>H4 : 23.89 | 23.89</td><td>H4 : 22 | 22</td><td>H4 : 23.38 | 23.38</td><td>H4 : 5.76 | 5.77</td><td>H4 : 1.63 | 1.63</td><td>H4 : 98.53 | 98.53</td><td>H4 : 183 | 177</td><td>H4 : 67.69 | 67.59</td><td>H4 : 94 | 62.5</td><td>H4 : 1 | 0</td><td></td></tr>
<tr><td>H5 : 23.89 | 24.1</td><td>H5 : 22 | 23</td><td>H5 : 23.38 | 23.38</td><td>H5 : 6.32 | 6.9</td><td>H5 : 1.63 | 1.63</td><td>H5 : 98.53 | 98.7</td><td>H5 : 175 | 183</td><td>H5 : 67.69 | 67.69</td><td>H5 : 2431 | 2409</td><td>H5 : 166 | 140</td><td></td></tr>
<tr><td>H6 : 24.29 | 24.39</td><td>H6 : 23 | 23</td><td>H6 : 23.5 | 23.56</td><td>H6 : 6.92 | 6.04</td><td>H6 : 1.63 | 1.62</td><td>H6 : 98.77 | 98.8</td><td>H6 : 191 | 183</td><td>H6 : 67.4 | 67.19</td><td>H6 : 2389 | 2376</td><td>H6 : 132 | 139</td><td></td></tr>
<tr><td>H7 : 24.39 | 24.39</td><td>H7 : 23 | 23</td><td>H7 : 23.63 | 23.75</td><td>H7 : 6.04 | 6.11</td><td>H7 : 1.61 | 1.62</td><td>H7 : 98.77 | 98.77</td><td>H7 : 181 | 178</td><td>H7 : 67.3 | 67</td><td>H7 : 2365 | 2345</td><td>H7 : 129 | 153</td><td></td></tr>
<tr><td>H8 : 24.5 | 24.5</td><td>H8 : 23 | 23</td><td>H8 : 23.81 | 23.88</td><td>H8 : 6.19 | 6.21</td><td>H8 : 1.62 | 1.62</td><td>H8 : 98.77 | 98.73</td><td>H8 : 182 | 178</td><td>H8 : 66.9 | 67</td><td>H8 : 2347 | 2335</td><td>H8 : 153 | 155</td><td></td></tr>
<tr><td>H9 : 24.5 | 24.5</td><td>H9 : 23 | 23</td><td>H9 : 23.94 | 23.94</td><td>H9 : 6.23 | 6.25</td><td>H9 : 1.63 | 1.64</td><td>H9 : 98.8 | 98.8</td><td>H9 : 181 | 181</td><td>H9 : 67.19 | 67.3</td><td>H9 : 2331 | 2326</td><td>H9 : 131 | 151</td><td></td></tr>
<tr><td>H10 : 24.6 | 24.6</td><td>H10 : 24 | 24</td><td>H10 : 24 | 24</td><td>H10 : 6.31 | 6.33</td><td>H10 : 1.64 | 1.64</td><td>H10 : 98.8 | 98.73</td><td>H10 : 182 | 227</td><td>H10 : 67.3 | 67.19</td><td>H10 : 2322 | 2321</td><td>H10 : 136 | 144</td><td></td></tr>
<tr><td>H11 : 24.6 | 24.6</td><td>H11 : 24 | 24</td><td>H11 : 24.06 | 24.06</td><td>H11 : 6.34 | 6.31</td><td>H11 : 1.63 | 1.64</td><td>H11 : 98.73 | 98.7</td><td>H11 : 238 | 275</td><td>H11 : 67.5 | 67.5</td><td>H11 : 2307 | 2307</td><td>H11 : 150 | 142</td><td></td></tr>
<tr><td>H12 : 24.6 | 24.7</td><td>H12 : 24 | 24</td><td>H12 : 24.13 | 24.13</td><td>H12 : 6.32 | 6.33</td><td>H12 : 1.65 | 1.64</td><td>H12 : 98.7 | 98.67</td><td>H12 : 330 | 327</td><td>H12 : 67.9 | 68.5</td><td>H12 : 2298 | 2304</td><td>H12 : 156 | 142</td><td></td></tr>
<tr><td>H13 : 24.7 | 24.7</td><td>H13 : 24 | 24</td><td>H13 : 24.13 | 24.19</td><td>H13 : 6.44 | 6.4</td><td>H13 : 1.64 | 1.65</td><td>H13 : 98.63 | 98.63</td><td>H13 : 336 | 342</td><td>H13 : 68.3 | 68.4</td><td>H13 : 2298 | 2287</td><td>H13 : 142 | 146</td><td></td></tr>
<tr><td>H14 : 24.79 | 24.79</td><td>H14 : 24 | 24</td><td>H14 : 24.19 | 24.25</td><td>H14 : 6.42 | 6.38</td><td>H14 : 1.65 | 1.65</td><td>H14 : 98.63 | 98.6</td><td>H14 : 317 | 289</td><td>H14 : 68.69 | 68.69</td><td>H14 : 2293 | 2277</td><td>H14 : 138 | 129</td><td></td></tr>
<tr><td>H15 : 24.79 | 24.79</td><td>H15 : 24 | 24</td><td>H15 : 24.25 | 24.31</td><td>H15 : 6.42 | 6.46</td><td>H15 : 1.65 | 1.65</td><td>H15 : 98.63 | 98.63</td><td>H15 : 267 | 250</td><td>H15 : 68.69 | 68.5</td><td>H15 : 2284 | 2277</td><td>H15 : 128 | 151</td><td></td></tr>
<tr><td>H16 : 24.79 | 24.79</td><td>H16 : 24 | 24</td><td>H16 : 24.31 | 24.31</td><td>H16 : 6.53 | 6.57</td><td>H16 : 1.65 | 1.65</td><td>H16 : 98.67 | 98.63</td><td>H16 : 236 | 231</td><td>H16 : 68.5 | 68.69</td><td>H16 : 2275 | 2267</td><td>H16 : 131 | 152</td><td></td></tr>
<tr><td>H17 : 24.79 | 24.79</td><td>H17 : 24 | 24</td><td>H17 : 24.31 | 24.38</td><td>H17 : 6.58 | 6.56</td><td>H17 : 1.65 | 1.65</td><td>H17 : 98.63 | 98.63</td><td>H17 : 224 | 218</td><td>H17 : 68.5 | 68.4</td><td>H17 : 2263 | 2265</td><td>H17 : 136 | 131</td><td></td></tr>
<tr><td>H18 : 24.79 | 24.79</td><td>H18 : 24 | 24</td><td>H18 : 24.38 | 24.38</td><td>H18 : 6.62 | 6.69</td><td>H18 : 1.66 | 1.66</td><td>H18 : 98.63 | 98.6</td><td>H18 : 218 | 215</td><td>H18 : 68.19 | 68.9</td><td>H18 : 2263 | 2254</td><td>H18 : 147 | 148</td><td></td></tr>
<tr><td>H19 : 24.79 | 24.79</td><td>H19 : 24 | 24</td><td>H19 : 24.38 | 24.38</td><td>H19 : 6.69 | 6.73</td><td>H19 : 1.66 | 1.66</td><td>H19 : 98.56 | 98.53</td><td>H19 : 439 | 442</td><td>H19 : 68.3 | 68</td><td>H19 : 2249 | 2249</td><td>H19 : 151 | 152</td><td></td></tr>
<tr><td>H20 : 24.79 | 24.79</td><td>H20 : 24 | 24</td><td>H20 : 24.38 | 24.38</td><td>H20 : 6.71 | 6.78</td><td>H20 : 1.65 | 1.65</td><td>H20 : 98.53 | 98.5</td><td>H20 : 437 | 433</td><td>H20 : 67.9 | 68.9</td><td>H20 : 2243 | 2245</td><td>H20 : 157 | 128</td><td></td></tr>
<tr><td>H21 : 24.79 | 24.7</td><td>H21 : 24 | 24</td><td>H21 : 24.31 | 24.31</td><td>H21 : 6.8 | 6.85</td><td>H21 : 1.64 | 1.64</td><td>H21 : 98.5 | 98.46</td><td>H21 : 430 | 431</td><td>H21 : 68.19 | 68.4</td><td>H21 : 2237 | 2231</td><td>H21 : 127 | 145</td><td></td></tr>
<tr><td>H22 : 24.7 | 24.65</td><td>H22 : 24 | 23</td><td>H22 : 24.31 | 24.31</td><td>H22 : 6.68 | 6.41</td><td>H22 : 1.63 | 1.64</td><td>H22 : 98.46 | 98.4</td><td>H22 : 427 | 423</td><td>H22 : 68.05 | 68.35</td><td>H22 : 1214 | 183.5</td><td>H22 : 79 | 0</td><td></td></tr>
<tr><td>H23 : 24.55 | 24.39</td><td>H23 : 23 | 23</td><td>H23 : 24.25 | 24.19</td><td>H23 : 6.38 | 6.09</td><td>H23 : 1.65 | 1.66</td><td>H23 : 98.36 | 98.23</td><td>H23 : 418 | 402</td><td>H23 : 68.6 | 68.65</td><td>H23 : 1265 | 286.5</td><td>H23 : 66.5 | 0</td><td></td></tr>
</table>

Figure 46. Aggregation, segmentation and classification of the data in a day

Now, it was time to start the data aggregation and segmentation to help with the manual classification dataset (AoT Stage – Section 4.1.2). Figure 46 shows all ten features segmented by day, hour and quarter of an hour, besides that it shows a daily picture and at the last column it asks for manual classification (which cycle this data belongs and if the seedlings are alive or dying), Section 6.1 will explain this dataset in more depth. The features on the data set are: temperature 1 (capture by the temperature and humidity sensor), temperature 2 (captured by the CO_2 sensor), water temperature, pH, EC, O_2, CO_2, Humidity, Moisture and Digital Light.

5.4.5. Specifying Learning

When the data from the environment sensors was filtered using the median filter by day, it was possible to see that the moisture sensor did not receive the same stimulus when its probes were in contact with water (Figure 47).

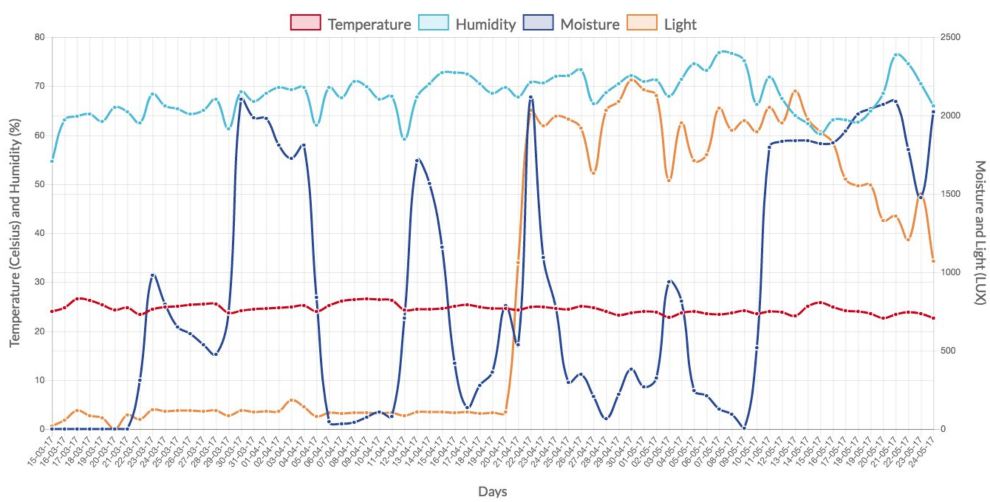

Figure 47. PPP environment sensors chart from Cycle 2 median value by day

This inconsistency is due to the fact that the Seeed Studio Grove moisture sensor is not hardened against contamination or exposure of the control circuitry to water and may be prone to electrolytic corrosion across the probes, given that the minerals from the hydroponic solutions accelerates this corrosion and creates a crust in the sensor. Due to its inaccuracy to decide whether there was a water stream on the system, the moisture sensor was considered as a binary sensor, if there was no water stream its value was always zero, otherwise its value was greater than one.

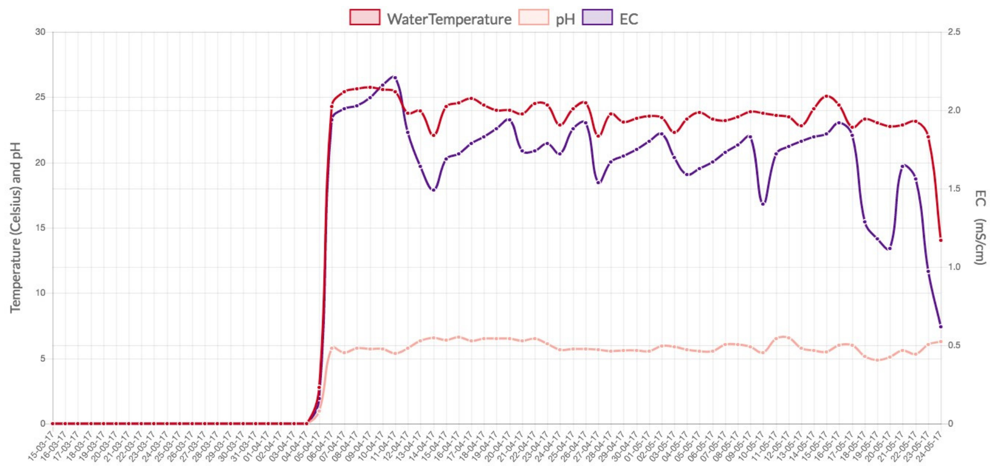

Figure 48. PPP water sensors chart from Cycle 2 median value by day

Figure 48 represents the water sensors median value by day. By analyzing this chart, it was possible to see that the water temperature and pH sensors were correctly calibrated and the EC sensor was not. The EC sensor values should be between 800 and 2000 µS/cm.

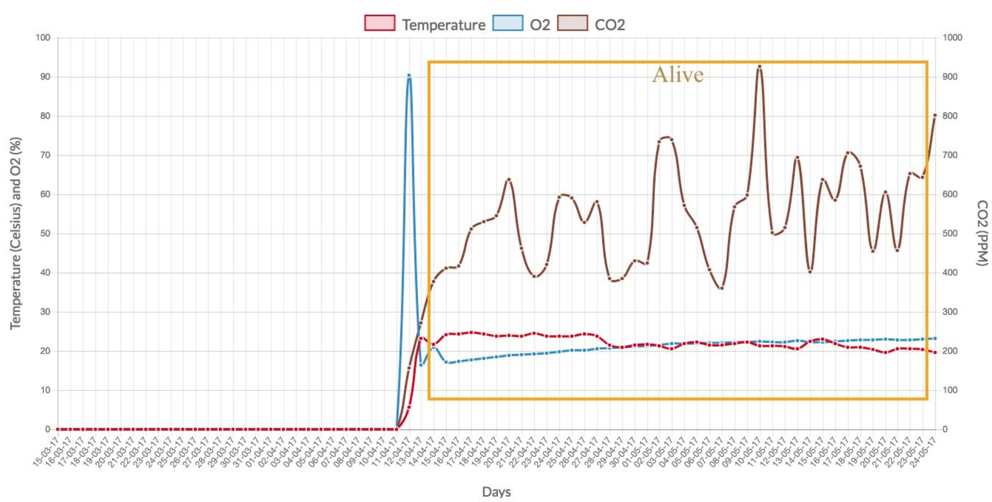

Figure 49. PPP gas sensor chart from Cycle 2 median value by day

Figure 49 represents the gas sensors median filter. By analyzing this chart, it is possible to see that the temperature sensor was between the acceptable range

(22-25°C), the O_2 sensors values were increasing along the cycle and the CO_2 sensor was volatile. The biological significance of this is that during daylight hours, plants take in carbon dioxide and release oxygen through photosynthesis; plants also release carbon dioxide through respiration. This behavior was classified as “Alive” (Figure 49) and was the first pattern that I would try to find in the next cycles. This was the first pattern to be added the knowledge base on the SoPC AmI Stage (Section 4.1.3).

Figure 50. Cycle 2 snapshot before harvest

At the end of the second cycle I received a visit from an agronomist from one of the largest organic producers in Rio de Janeiro (Vale das Palmeiras), named Danilo Pinto Silva. He harvested, weighed and analyzed some crops with me, his visit is register at the CTC/PUC-Rio IoT YouTube Channel (https://youtu.be/QefwEWXcYsE). He pointed out that the cycle two PPP took too long; the lettuce was too big (Figure 50), the stems were too thick causing a bitterness in its flavor; the cycle overall temperature could be higher (around 26°C) and the seedlings needed to receive more light.

5.5. Cycle 3 – Heating the Prototype

The third cycle started on Wednesday 21/06/2017, the Plant Production Process (PPP) took 35 days ending on Wednesday 26/07/2017. In this cycle, I wanted to make the seedling grow faster. I was too far from five to six weeks PPP benchmark proposed by Kozai et al. [4].

5.5.1.
Diagnosing

The grow lights intensity and the temperature were not maximized for the best plant growth. They needed to be increased in order optimize the PPP. According to Rachmilevitch et al. [56], it is important to regulate air temperature in PFALs, since the plant respiration may vary according to the temperature (Figure 51).

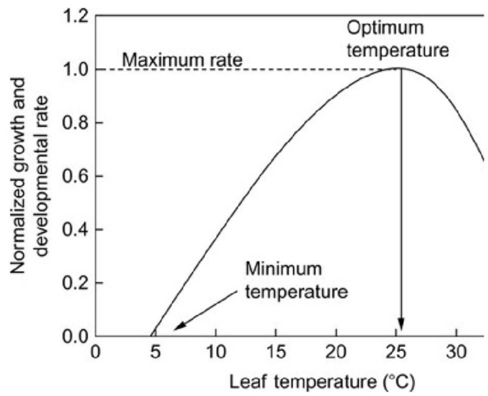

Figure 51. Plant growth and developmental rate as affected by temperature [57]

Understanding how environmental factors affects various aspects of plant growth and development is crucial for m-PFAL design and operation. Kubota et al. [57] list some factors affecting plant growth include : (1) temperature, (2) light intensity, (3) light quality, (4) humidity, (5) CO_2 concentration in the air, (6) air current speed, and (7) nutrient and root-zone environments. Factors 1, 2 and 7 needed to be reviewed in this cycle in order to increase the plants growth.

5.5.2.
Action Planning

To increase light quality and intensity the LED grow lights panel needed to be placed closer to the hydroponic bed. The temperature needed to be near as possible of 26°C, the heater and fans needed to be turned on altogether in order to keep the best seedlings development rate.

The IFTTT rules needed to be reviewed in order to maintain the room temperature at 26°C, instead of the range stablished in previous cycles (between 22 and 25°C). The moisture sensor also needed to be read every ten minutes by the IFTTT rules to check if there was a water stream on the roots.

5.5.3. Action Taken

A one meter chain was bought to act as an extension to decrease the LED grow lights panel distance by 30cm from the hydroponic bed (Figure 52).

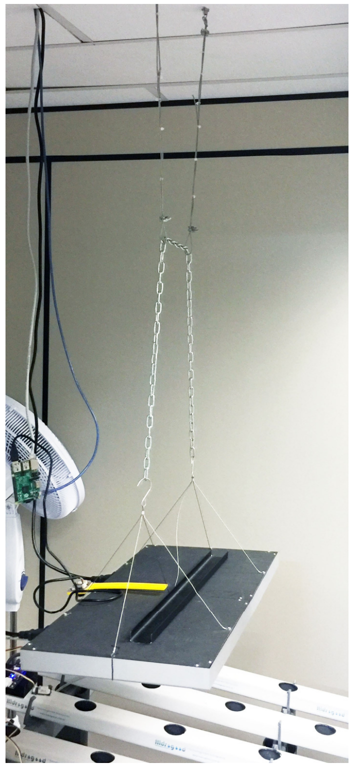

Figure 52. LED Grow Lights Panel chain extension

The temperature on the IFTTT Loop was changed to set the heater **ON** if the temperature is below 25°C and turn **OFF** the fan; if the temperature is greater than 27°C it should turn **OFF** the heater and turn **ON** the fan, any other value the heater and fan are **OFF** (Figure 53). A new moisture rule was also added to alert the control panel about the water stream status.

```
function iftttPPPRules () {
        const time = getTime();
        const temperature = getTemperature(); const
        humidity = getHumidity(); const moisture =
        getMoisture();
        // TIME RULES
        if(time >= '5:00' || time <= '23:00') {   setWaterPump('on');   setAirPump('on');
        setGrowLight('on');   // MOISTURE RULES   if(moisture >= 1)
        setMoistureStatus('water stream OK')   else setMoistureStatus('no water stream')
        } else {
          setWaterPump('off');   setAirPump('off');
        setGrowLight('off');
          // MOISTURE RULES
          if(moisture >= 1) setMoistureStatus('water stream must be off')   else setMoistureStatus('no
        water stream')
        }
        // TEMPERATURE RULES if(temperature < 25) {
        setHeater('on');   setExhaustFan('off');
        } else if (temperature >= 27) {   setHeater('off');
        setExhaustFan('on');
        } else {   setHeater('off');
        etExhaustFan('off');
        }
        // HUMIDITY RULES if(humidity < 70) {
        setHumidifier('on');
        } else {
          setHumidifier('off');
        }
}
//EXECUTE EVERY 10MIN
setInterval(iftttPPPRules, 600000)
```

Figure 53. IFTTT Loop Rules from Cycle 3

Table 9. Cycle 3 nutrients feed chart for a fourteen-liter water tank

WEEK #		1	2	3	4	5
GROWTH STAGE		Seedling	Early Growth	Late Growth	Transition	Early Bloom
Base Nutrients	FloraMicro	8ml	30ml	40ml	32ml	32ml
	FloraGro	8ml	45ml	40ml	32ml	32ml
	FloraBloom	8ml	10ml	20ml	32ml	32ml

The seedlings were fed with hydroponic nutrients every week according to Table 9 following the General Hydroponic recirculating feed chart program[17].

5.5.4.
Evaluating

Cycle three was not successful. The seedlings did not die but the environment got too hot, causing the appearance of algae in the water tank and the growth of a fungus that affected the roots (Figure 54).

Figure 54. Seedlings size and fungus appearance at the end of cycle 3

The seedlings were not developing. After five weeks, its size and appearance was like as seedlings at week two. In the last week of cycle three, when I made the weekly replenished of nutrients, I noticed that the m-PFAL was too hot and the heater was **ON** when it should be **OFF**.

I examined the relays microcontroller output and noticed that the environment temperature sensor was locked below 25°C making the heater always on. When I reset all microcontrollers, and decreased the temperature inside the mPFAL with the SecondLab air conditioner the hole system started to work again. Figure 55 represents the environment sensors mean value by hour during all cycle three. All sensors were acting as expected and their values were also in range.

[17] https://goo.gl/3uxFw1

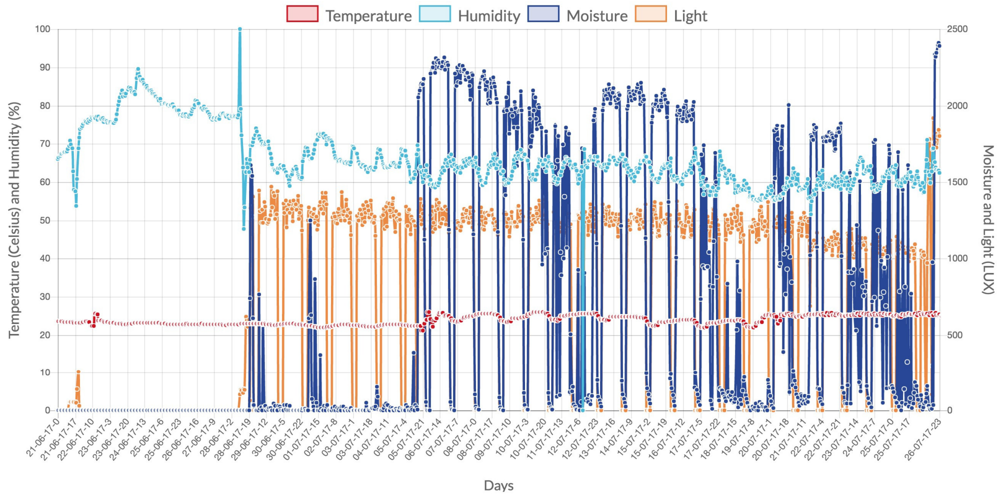

Figure 55. Cycle 3 environment sensors chart (temperature, humidity, moisture and digital light) mean value by hour

The water sensors (temperature, pH and EC) mean values by hour are represented in Figure 56. Analyzing this chart, it was possible to notice that the pH and EC values were as expected but the temperature was not (too stable).

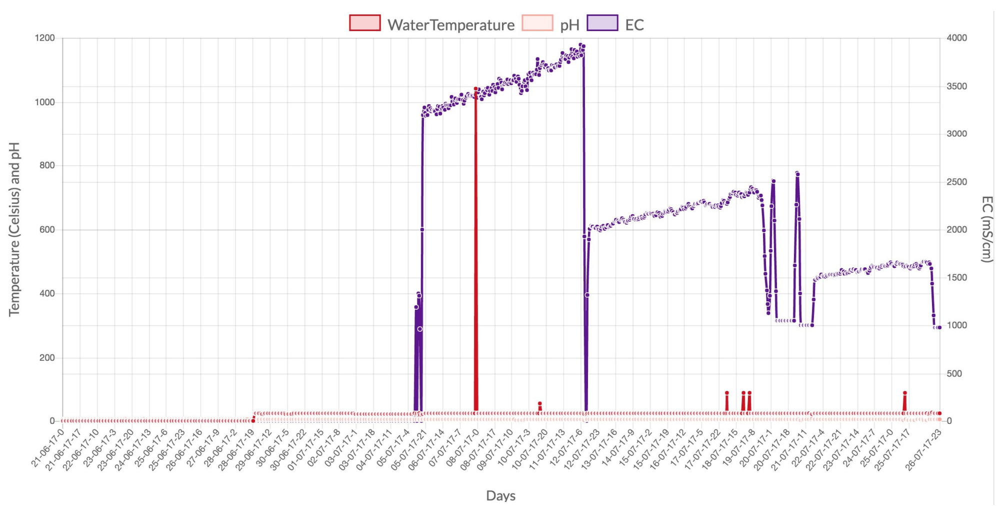

Figure 56. Cycle 3 water sensors (water temperature, pH and EC) mean value by hour

The gas sensors (temperature, O_2 and CO_2) mean value by hour during all cycle three are represented in Figure 57, in this chart it was possible to notice that the temperature was over 26°C throughout the cycle and the O_2 and CO_2 sensors were acting differently from previous cycle.

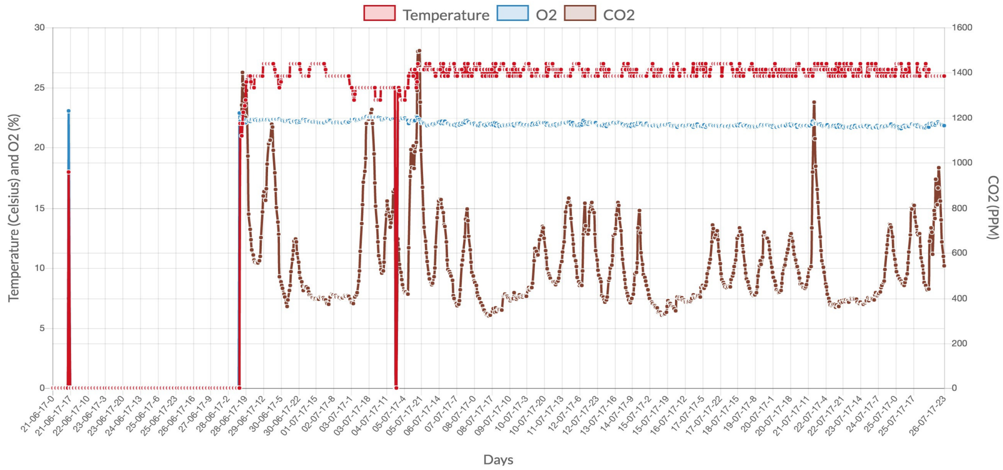

Figure 57. Cycle 3 gas sensors chart (temperature, O_2 and CO_2) mean value by hour

Because of the fungus appearance I did not try do adjust the temperature and continue with the third cycle. All seedlings were thrown away and the hydroponic bed and all sensors were cleaned with isopropyl alcohol and chlorine. The water system worked for three uninterrupted days with 80% water and 20% chlorine to remove fungi and algae from all tubes, gutters and pumps.

5.5.5. Specifying Learning

This incident made me notice that the IFTTT temperature control rule was dependent only on one sensor to control all actuators that influenced the m-PFAL temperature. In the next cycles the IFTTT loop should be changed to make redundancies by calculating the mean between all temperature sensors.

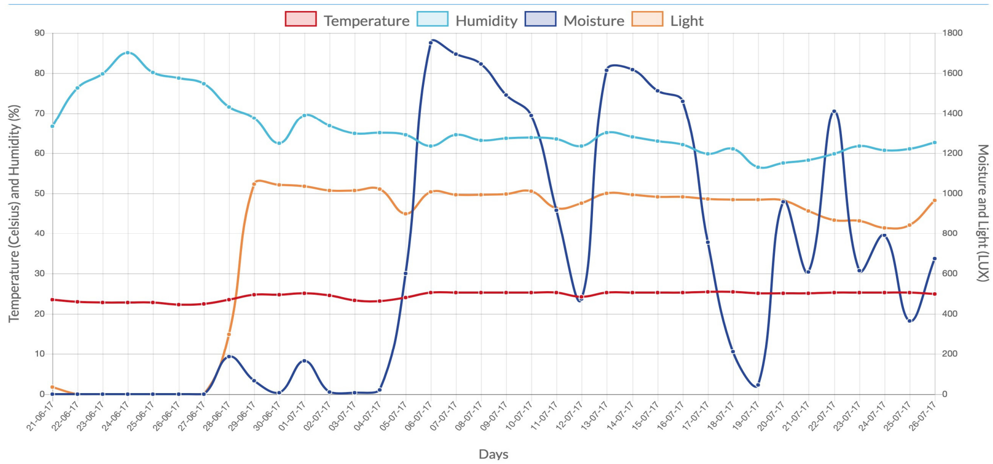

Figure 58. PPP environment sensors chart from Cycle 3 median value by day

Analyzing the environment sensors chart (Figure 58) it was possible to notice the humidity values dropping as the temperature got hotter, it is also possible to see that the temperature sensor stopped working for a period of time.

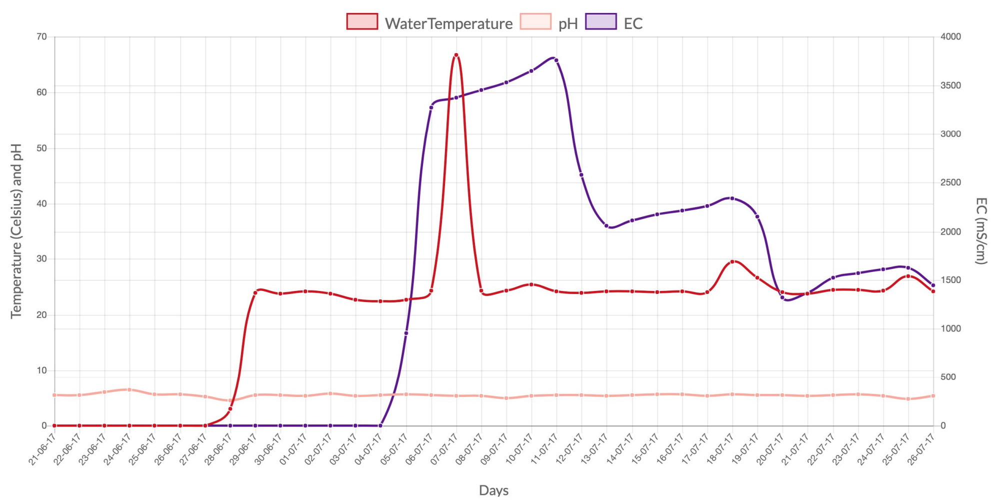

Figure 59. PPP water sensors chart from Cycle 3 median value by day

Analyzing the water sensors chart (Figure 59) it was possible to notice that the water temperature was above 26°C throughout the cycle and it even reach 28 and 30°C. According to the hydroponic fertilizer manufacturer, for the nutrient solution be most effective, the ideal water temperature should be between 22 and 25°C. That is why the algae appeared, algae blooms occur when the water temperature gets warm.

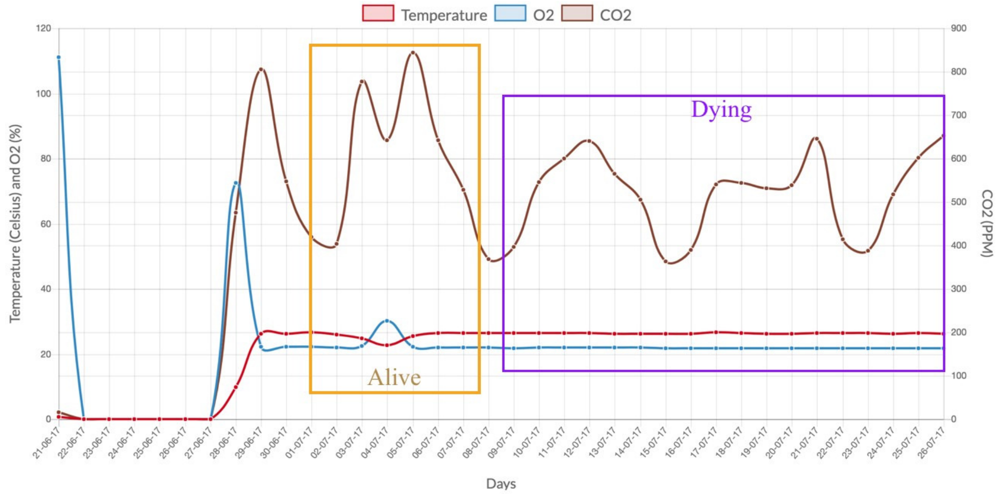

Figure 60. PPP gas sensors chart from Cycle 3 median value by day

Analyzing the gas sensors chart (Figure 60) it is possible to see that the temperature was above 26°C throughout the cycle and the O_2 sensor values were almost the same (21.9%), the seedlings were not growing properly and were not releasing sufficient O_2 during the photosynthesis that could be read by the sensor (Figure 60 – dying pattern). Looking at the CO_2 sensor it is possible to notice that the seedlings were not breathing like during Cycle 2 (Figure 49). This chart shows that the seedlings were alive but there was something wrong (it was like the plants were having difficulty breathing), this was the second pattern to be added to the knowledge base on the SoPC AmI Stage (Section 4.1.3). However, there is one period between 02-07-17 and 06-07-17 (Figure 60 – alive highlight). that is possible to notice the seedling breathing (releasing CO_2) and performing photosynthesis (converting CO_2 from the air and water into glucose and releasing O_2 as product), coincidentally this period was the coolest period on the entire cycle.

In this cycle, I learned that keeping the m-PFAL warm was not a good idea, it helped algae and fungus bloomed and dramatically decreased the seedlings growth. I also learned that the system exhaust fan was not good to decrease the temperature inside the m-PFAL, its main function was to maintain air circulation through the system.

5.6. Cycle 4 – Hibernating the Prototype

The fourth cycle started on Saturday 29/07/2017, the Plant Production Process (PPP) took 46 days ending on Wednesday 13/09/2017. In this cycle, I still wanted to make the seedling grow faster and reach the six weeks benchmark proposed by Kozai et al. [4].

5.6.1. Diagnosing

In previous cycle the m-PFAL got too warm. Due to the system inaccuracy having just one temperature sensor data was not enough to guarantee the triggering of the IFTTT rule that controls the system temperature. Besides that, the system had no other way to decrease its temperature apart from triggering the exhaust fan.

Furthermore, when LED grow lights panels were placed closer to the plants in previous cycle, leaves were absorbing significant amounts of thermal energy from the LEDS. Air circulation also helps mitigate leaf temperatures by facilitating convective heat transfer away from the plants. As air speed increases, more and more heat is dissipated by forced air movement. An air current speed of more than $0.5ms^{-1}$ at the leaf canopy is required to promote gas exchange [58].

5.6.2. Action Planning

The new IFTTT temperature rule should use an average of the sum of all online temperature sensors that were publishing data. To avoid the algae and fungus blooming the temperature range should be lower than 26°C. A stronger fan should be placed near the hydroponic bed and the water tank to rapidly decrease the temperature in case there was a need and to maintain an air current speed to promote gas exchange.

5.6.3. Action Taken

A commercial pedestal floor fan was bought and placed in a fixed position pointing the fan blades to the wall in order to blow an indirect wind to the hydroponic bed. The fan was also tilted up to blow the hot air up towards the exhaust fan (Figure 61).

Figure 61. m-PFAL cooling system

The IFTTT loop was changed to get the temperature from all temperature sensors in the system. The temperature range changed and was between 22 and 25°C. The pedestal fan was also included in the rules to blow its weakest wind when LED grow lights were **ON (**Figure 62**)**.

```
function iftttPPPRules () {
        const time = getTime();
        const temperature = getTemperature(); const temperature2 = getTemperature2();//FROM
        CO2 Sensor const wtemperature = getWaterTemperature(); const humidity = getHumidity();
        const moisture = getMoisture();
        // TIME RULES if(time >= '5:00' || time <= '23:00') {    setWaterPump('on');
        setAirPump('on');    setGrowLight('on'); setPedestalFan('on');    // MOISTURE RULES
        if(moisture >= 1) setMoistureStatus('water stream OK')    else setMoistureStatus('no
        water stream')
        } else {
          setWaterPump('off'); setAirPump('off');    setGrowLight('off'); setPedestalFan('off');    //
        MOISTURE RULES    if(moisture >= 1) setMoistureStatus('water stream must be off')    else
        setMoistureStatus('no water stream')
        }
        // TEMPERATURE RULES
        //GET TEMPERATURE MEAN
        const meanTemperature=mean([temperature,temperature2,wtemperature]) if(meanTemperature
        < 22) {    setHeater('on');    setExhaustFan('off'); } else if (meanTemperature > 25) {
        setHeater('off'); setExhaustFan('on');    setPedestalFan('on');
        } else {
          setHeater('off');    setExhaustFan('off');
        }
        // HUMIDITY RULES if(humidity < 70) {
        setHumidifier('on');
        } else {    setHumidifier('off');
        }
}
//EXECUTE EVERY 10MIN setInterval(iftttPPPRules, 600000)
```

Figure 62. IFTTT Loop Rules from Cycle 4

Table 10. Cycle 4 nutrients feed chart for a fourteen-liter water tank

WEEK #		1	2	3	4	5
GROWTH STAGE		Seedling	Early Growth	Late Growth	Transition	Early Bloom
Base Nutrients	FloraMicro	8ml	8ml	10ml	15ml	15ml
	FloraGro	8ml	15ml	10ml	20ml	20ml
	FloraBloom	8ml	8ml	10ml	7ml	7ml

The seedlings were fed with hydroponic nutrients every week according to Table 10 following the European General Hydroponic recirculating feed chart program[18]. The difference from this recipe to the recipe used in previous cycles was that the European version recommends less nutrients per liter.

5.6.4. Evaluating

Cycle four was successfully completed under five weeks reaching the benchmark proposed by Kozai et al. [4]. However, this cycle almost ended in failure. The water circulation receiving channel that drains the water back into the water tank partially detached itself from the hydroponic bed, causing a slow but constant water leak. The moisture sensor could not detect the leak until the water tank was complete empty.

Figure 63. Dry seedling in Cycle 4 due lack of water

The seedlings almost died from lack of water (Figure 63), seedlings under grow lights drink more water than usual and because of the LED heat over the plant it also gets dry very quickly. The seedlings stayed one day without water, when I saw the control panel warning I quickly acted to replenish the water and nutrients. Unfortunately, I got a bit late and five out of fifteen plants died. All those who died were right under the LED grow lights panel.

18 http://www.eurohydro.com/publications/publications/APPLICATION%20CHARTS/GB/CHART-FLORASERIES-GB.pdf

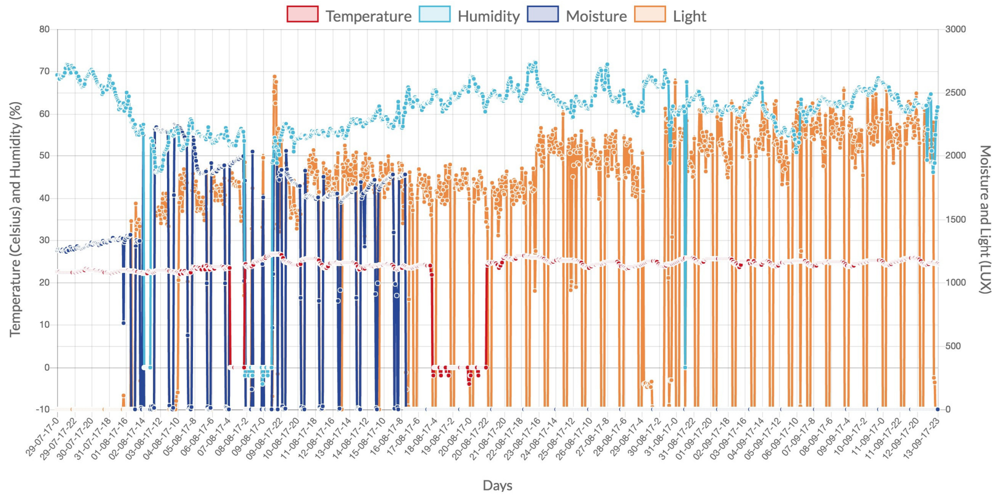

Figure 64. Cycle 4 environment sensors chart (temperature, humidity, moisture and digital light) mean value by hour

Figure 64 represents the environment sensors mean value by hour during all cycle four. All environment sensors values were as expected. The water sensors (temperature, pH and EC) mean values by hour are represented in Figure 65. The water temperature was very noise and published many outliers.

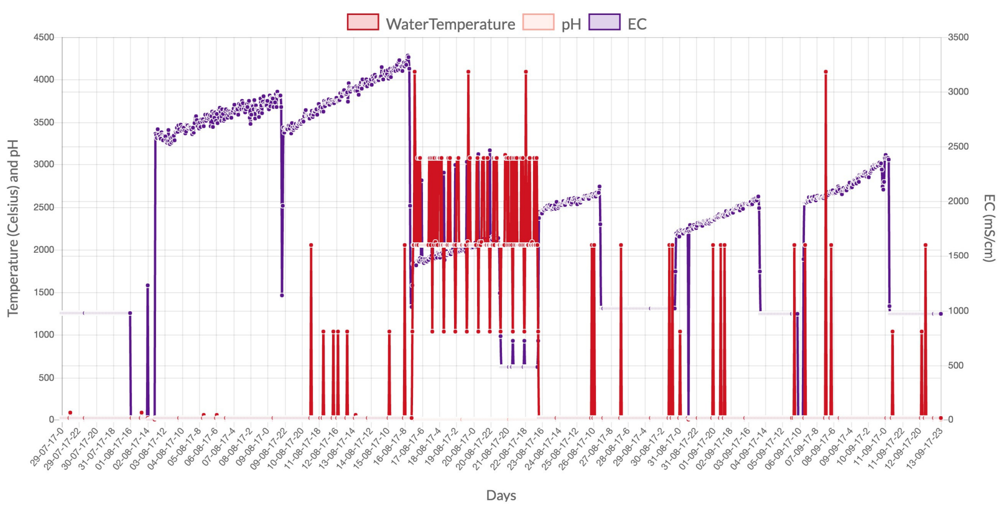

Figure 65. Cycle 4 water sensors (water temperature, pH and EC) mean value by hour

The gas sensors (temperature, O_2 and CO_2) mean value by hour during all cycle four are represented in Figure 66. All gas sensors were behaving as expected.

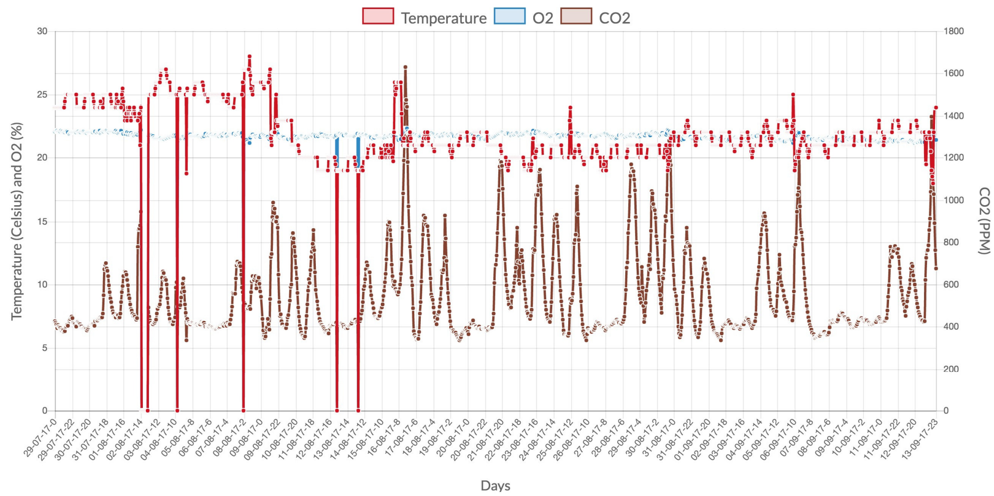

Figure 66. Cycle 4 gas sensors chart (temperature, O_2 and CO_2) mean value by hour

Despite the setback, the seedlings quickly recovered and in just three days they returned to normal. At the end of cycle ten lettuces were harvested with the average weight of forty-eight grams.

5.6.5. Specifying Learning

This new incident made me notice a simple but very important rule: if anything goes wrong or is out of pattern the system should be put to hibernate immediately. The hibernation that I proposed was to set the m-PFAL as dark and cooler as possible. All lights should be **OFF**, the heater should be **OFF**, all fans should be **ON**, humidifier should be **ON** and whenever possible the water pumps and air pumps should be **ON**, given that there is water in the water tank and there is a water stream running through the system, otherwise it should be **OFF** to not damage the pumps.

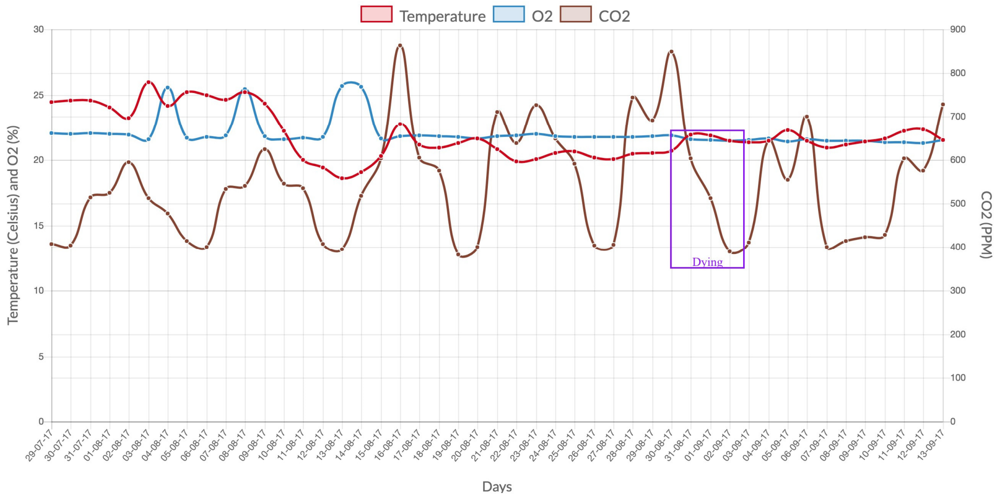

Figure 67. PPP gas sensors chart from Cycle 4 median value by day

Analyzing the gas sensors chart (Figure 67) it was possible to notice that seedlings breathed normally throughout the cycle but right after the lack of water (29-08-17) the seedlings were recovering and their next respiration reached lower level than the previous ones, if everything were normally it should be the other way around.

Another weakness exposed in this cycle was the lack of data to detect if there was a leak in the system. I only detected the problem when the water tank was complete empty. The Sensor and Act Section in the IoT Stage (Section 4.1.1.1) must be revisited in order to study a new sensor and where to install it to detect if the system is leaking.

5.7. Cycle 5 – Final Evaluation

The fifth cycle started on Saturday 16/09/2017, the Plant Production Process (PPP) took 50 days ending on Monday 05/11/2017. In this cycle, I just planned to repeat the previous cycle result without incidents.

5.7.1. Diagnosing

In case of lack of water, system malfunction or any unpredicted situation, to avoid losing any seedlings or damage equipment, a new pre-programmed routine should be implemented in order to set the m-PFAL to a damage control state. The AmI Stage should use its knowledge base and predict behavior to analyze patterns to make an autonomous decision to set the system in the damage control state. Besides that, a new sensor needed to be installed in the system in order to detect leaks.

5.7.2. Action Planning

Until now the leading causes of death were lack of water and excessive heat. For avoiding both death causes it would be better for the seedlings to enter in a hibernation state. Therefore, the first damage control state should be a hibernation state preparation procedure as follow: turn **OFF** lights and heater; turn **ON** all fans, humidifier and pumps. If there is no water in the water tank or water stream, pumps are **OFF**.

A good way to know whether there is a leak in the system is to constantly monitor the water level in the water tank. Whenever a critical water level in the tank was reached, the system should be alerted. There are two cheap sensors fit to the task (Section 4.1.1.1). The first one is a water level switch, comprising a float ball and a reed switch, the float ball rises or falls with the liquid to the level of the switch, a magnet within the float ball causes the reed switch to turn ON if they are in contact. When the float ball move away from the reed switch, it turns the switch OFF. The second sensor is a non-contact ultrasonic sensor to measure the distance from the top of the water tank to the water level. This sensor emits an ultrasonic wave and calculates the time the wave takes to reflect on the water and return to the sensor.

5.7.3. Action Taken

A water level switch was installed in a critical level within the water tank. This switch was simple to install and its data do not need to be saved, it works as an alert binary flag. The IFTTT loop was changed to set the m-PFAL in a hibernation state as follow on Figure 68.

```
function iftttPPPRules () {
            const time = getTime(); const temperature = getTemperature(); const temperature2 =
            getTemperature2();//FROM CO2 Sensor const wtemperature = getWaterTemperature();
            const humidity = getHumidity(); const moisture = getMoisture(); const waterInLevel =
            getWaterLevelSwitch();//Binary Data (1 is ok)
            // TIME RULES if(time >= '5:00' || time <= '23:00') {//LIGHT PERIOD    setWaterPump('on');
            setAirPump('on');    setGrowLight('on'); setPedestalFan('on');    // MOISTURE RULES
            if(moisture >= 1) setMoistureStatus('water stream OK');    else setMoistureStatus('no water
            stream');setDamageControl();
            } else {//REST PERIOD
              setWaterPump('off'); setAirPump('off');    setGrowLight('off'); setPedestalFan('off');    //
            MOISTURE RULES    if(moisture >= 1) setMoistureStatus('water stream must be off');    else
            setMoistureStatus('no water stream');
            }
            // TEMPERATURE RULES
            //GET TEMPERATURE MEAN
            const meanTemperature=mean([temperature,temperature2,wtemperature]) if(meanTemperature
            < 22) setHeater('on'); setExhaustFan('off'); else if (meanTemperature > 25) setDamageControl ();
            else setHeater('off'); setExhaustFan('off'); //HUMIDITY RULES if(humidity < 70) setHumidifier('on');
            else setHumidifier('off')
            //Check Water Tank Level
            if (!waterInLevel) setDamageControl();
}
//EXECUTE EVERY 10MIN setInterval(iftttPPPRules, 600000); //DAMAGE
CONTROL RULES (HIBERNATION) function setDamageControl () {
setHeater('off'); setExhaustFan('on'); setPedestalFan('on');
setHumidifier('on');
            if(moisture >= 1) setWaterPump('on'); setAirPump('on'); else
setWaterPump('off'); setAirPump('off'); }
```

Figure 68. IFTTT Loop Rules from Cycle 5

Table 11. Cycle 5 nutrients feed chart for a fourteen-liter water tank

WEEK # GROWTH STAGE		**1** Seedling	**2** Early Growth	**3** Late Growth	**4** Transition	**5** Early Bloom
Base Nutrients	FloraMicro	8ml	8ml	10ml	15ml	15ml
	FloraGro	8ml	15ml	10ml	20ml	20ml
	FloraBloom	8ml	8ml	10ml	7ml	7ml

The seedlings were fed with hydroponic nutrients every week according to Table 11 following the European General Hydroponic recirculating feed chart program[19]. After five weeks, the seedlings were fed only with fresh water and 7ml of Flawless Finish[20] flushing solution from Advanced Nutrition.

5.7.4.
Evaluating

Cycle five was successful completed under five weeks reaching the benchmark proposed by Kozai et al. [4] and without incidents. Figure 69 represents the environment sensors mean value by hour during all cycle five. All environment sensors were behaving as expected.

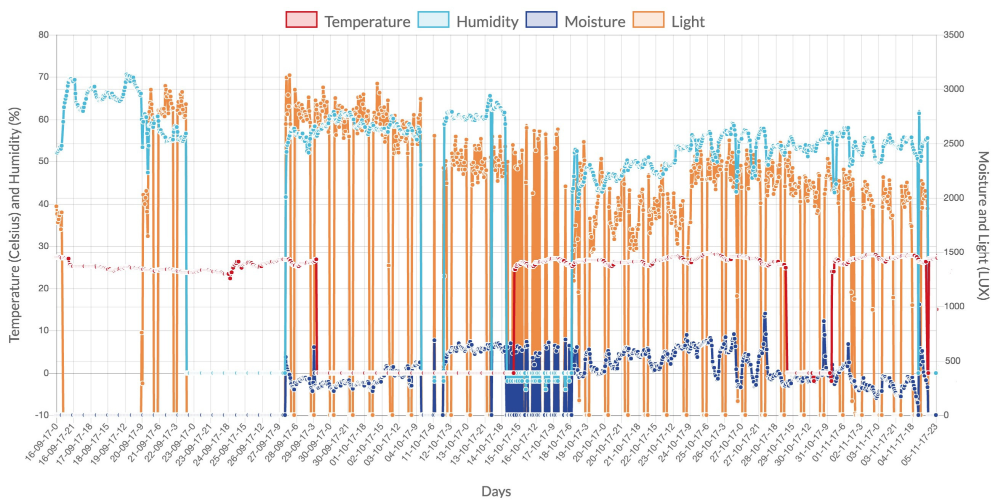

Figure 69. Cycle 5 environment sensors chart (temperature, humidity, moisture and digital light) mean value by hour

The water sensors (temperature, pH and EC) mean values by hour are represented in Figure 70. The water temperature sensors noise was getting worse through time, in this chart it is possible to notice all outliers published by the sensor.

[19] http://www.eurohydro.com/publications/publications/APPLICATION%20CHARTS/GB/CHART-FLORA-SERIES-GB.pdf

[20] http://www.advancednutrients.com/products/flawless-finish/

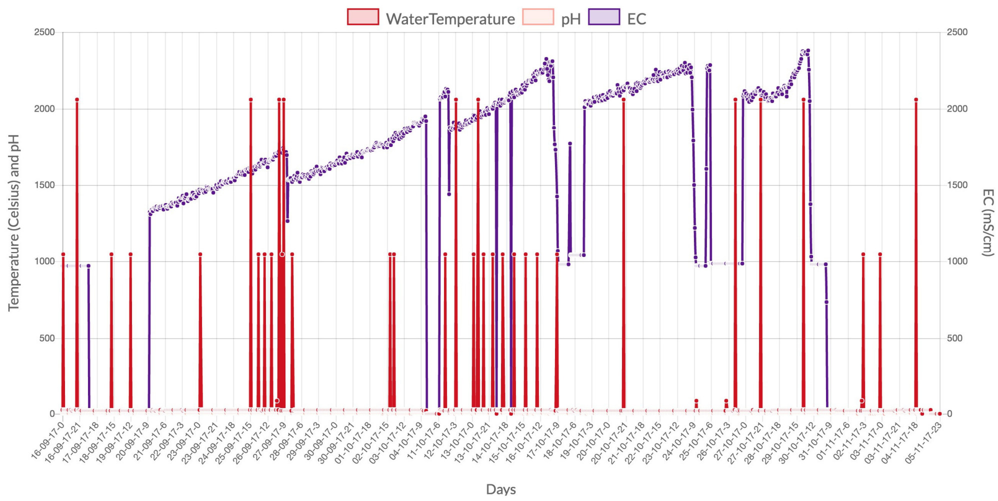

Figure 70. Cycle 5 water sensors (water temperature, pH and EC) mean value by hour

The gas sensors (temperature, O_2 and CO_2) mean value by hour during all cycle four are represented in Figure 71. All environment sensors were behaving as expected.

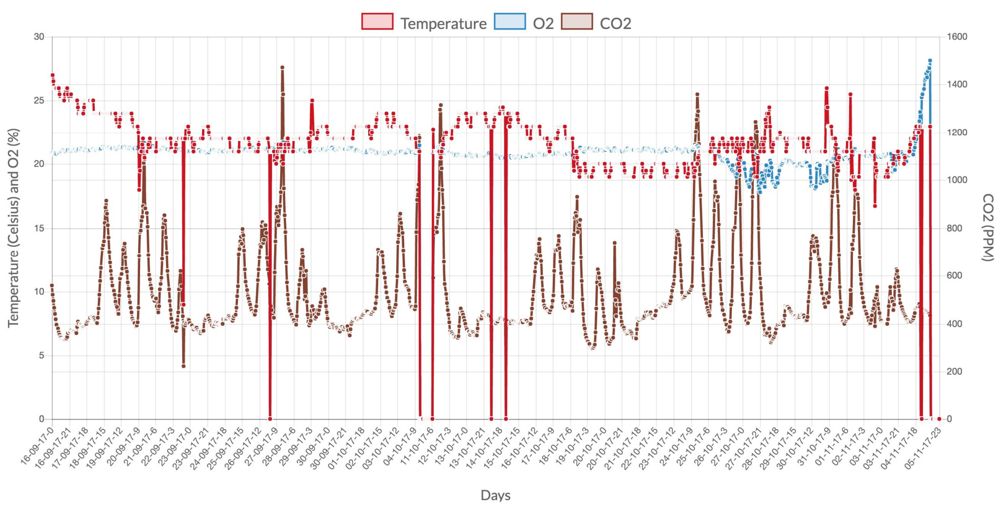

Figure 71. Cycle 5 gas sensors chart (temperature, O_2 and CO_2) mean value by hour

5.7.5. Specifying Learning

After five cycles the system was stable. There were two sensors (water temperature sensor and EC sensor) that were generating more noises and outliers than usual. I checked with the manufacturer and I learned that all water sensors used in this prototype had a one year life span due to corrosion in the electrodes by the hydroponic nutrients (Figure 72).

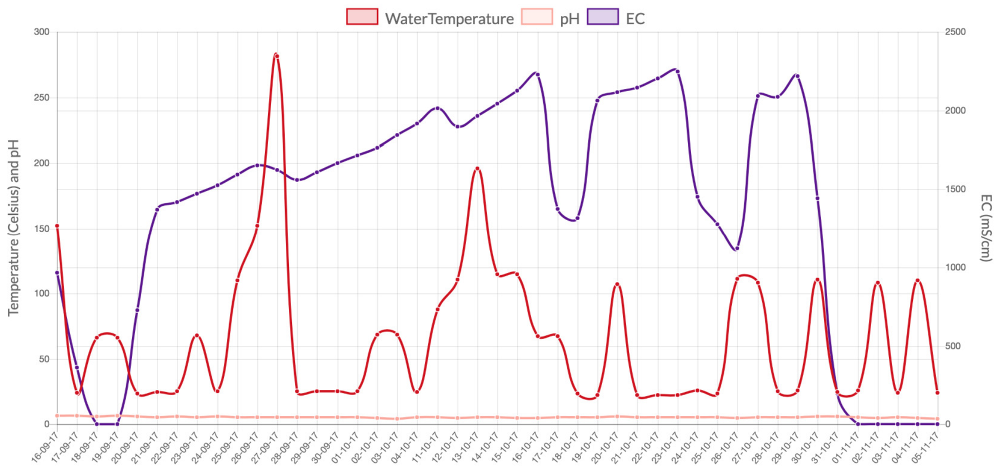

Figure 72. PPP water sensors chart from Cycle 5 median value by day

Besides the unexpected inaccuracy of the water sensors (water temperature, pH and EC), the gas sensors capture the expected system behavior (Figure 73) for seedlings respiration and photosynthesis.

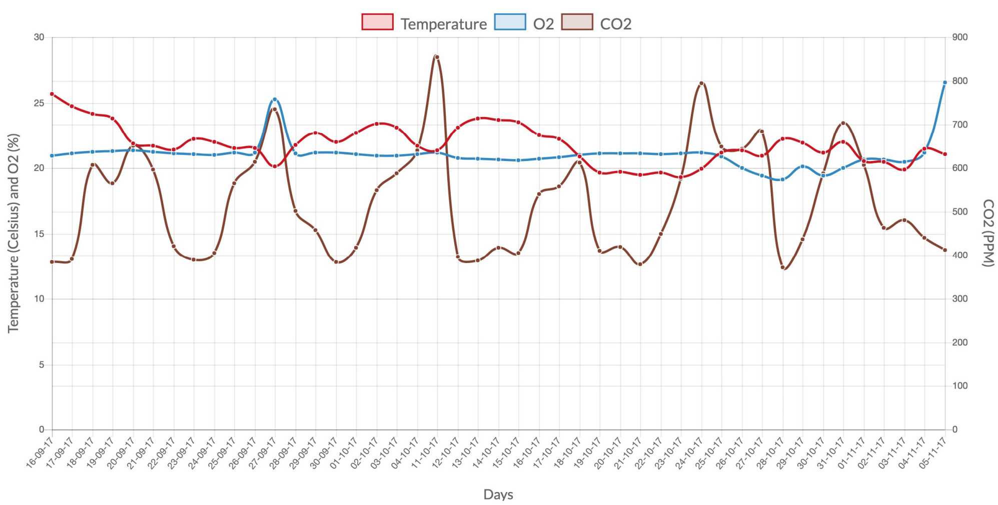

Figure 73. PPP gas sensors chart from Cycle 5 median value by day

The next step is to use the dataset generated by all five cycles to train a model to help the system predict its behaviors and maintenance.

6 Research Cycles Dataset Wrap-Up and Analysis

This section is a wrap-up the research cycles dataset findings and analysis.

6.1. Dataset Summary

The dataset was divided into three parts (aggregated, segmented and analytic) using two different types of NoSQL database (a graph database and a document database). The first part, the aggregated dataset, has 123430 records, containing the raw data published by each sensor aggregated by type. It was recorded into a graph database (Google Firebase) using webhooks as mentioned in Section 4.1.2. and Section 5.3.3. This database has a JSON tree structure where the first four nodes represent the aggregation type (environment sensors, gas sensors, water sensors and images). To access an aggregation type sample, it is necessary to navigate into its structure with an associated key (a hashed timestamp key). This architecture allows data to be quickly saved as a timeline log. Each aggregation type has its own JSON structure as show in Figure 74.

Figure 74. (a) Environment Sensors JSON, (b) Water Probes Sensors JSON, (c) Gas Sensors JSON, and (d) Images Base64 JSON.

The second part, the segmented dataset, has 717 records, containing the timeline log from previous dataset segmented by day, hour and half an hour. The segmented data is registered in a NoSQL document database (Google Firestore) by a serverless time-based web function that performs a parse script every day. For each type (environment sensors, gas sensors and water sensors) a collection (a set of JSON documents) was created, segmenting each document by day (e.g. DDMM-YY is the key to access the document). In each document a JSON tree was created segmenting the data by every hour of the day (e.g. H0, H1,~,H23) and by half an hour (e.g. Q0: first half an hour, Q1: second half an hour) (Figure 75).

Figure 75. Gas Sensor segmented by day, hour and half an hour

For each half an hour object, an array was saved to segment the data every 10 minutes. Therefore, each half an hour has an array with at most three elements (sensors publish data every ten minutes), and each hour has an object with two keys (representing the data on every thirty minutes) and, finally, each document has 24 keys (representing the data segmented for every hour of the day) (Figure 76).

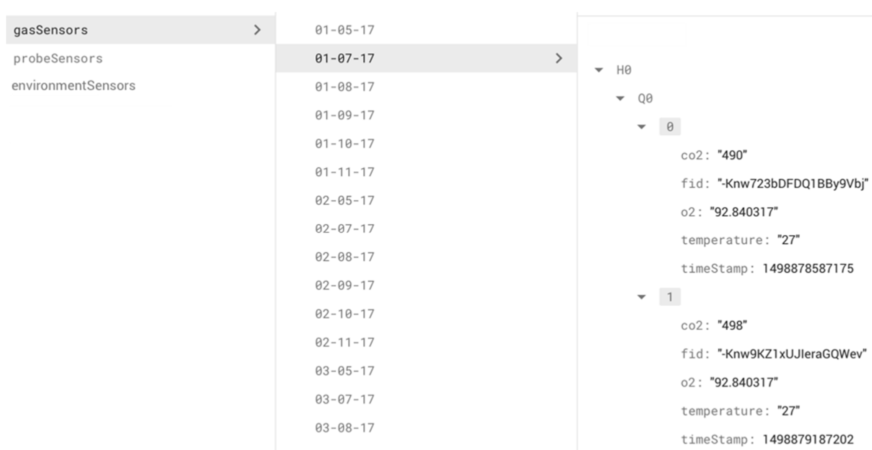

Figure 76. JSON document Array saved at every half an hour object

The third part, the analytic dataset, has 349 records. This dataset was also recorded in a NoSQL document database and it has the same data structure as the second part above mentioned: a key by day, hour and half an hour. However, in this case, the daily document has tree new keys: an image array (representing all photos taken in a day), a numerical value that represents the research cycle that this data belongs and the manual classification of the seedling condition (e.g. alive or dying) (Figure 77).

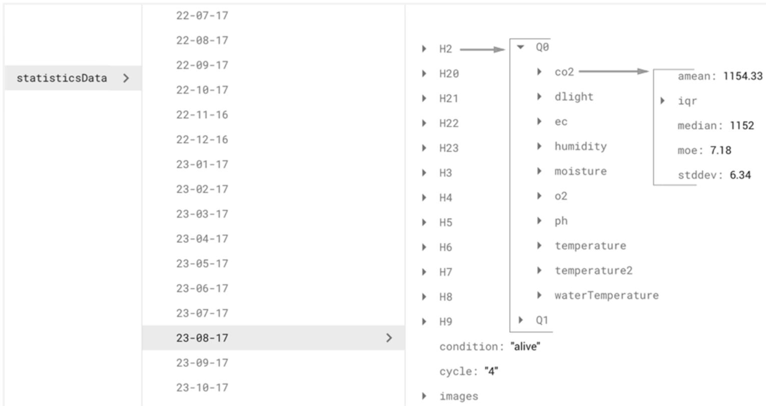

Figure 77. Statistic Dataset with extracted features, daily images and seedlings condition

The analytic dataset half an hour object has the features extracted from previous datasets. Their values are a statistical summary of the thirty-minute window (an arithmetic mean, a median, a standard deviation and margin of error) from the previous array. The analytic dataset was used to expose all data in a webpage in order to execute the manual cycle and seedlings condition classification (Figure 78).

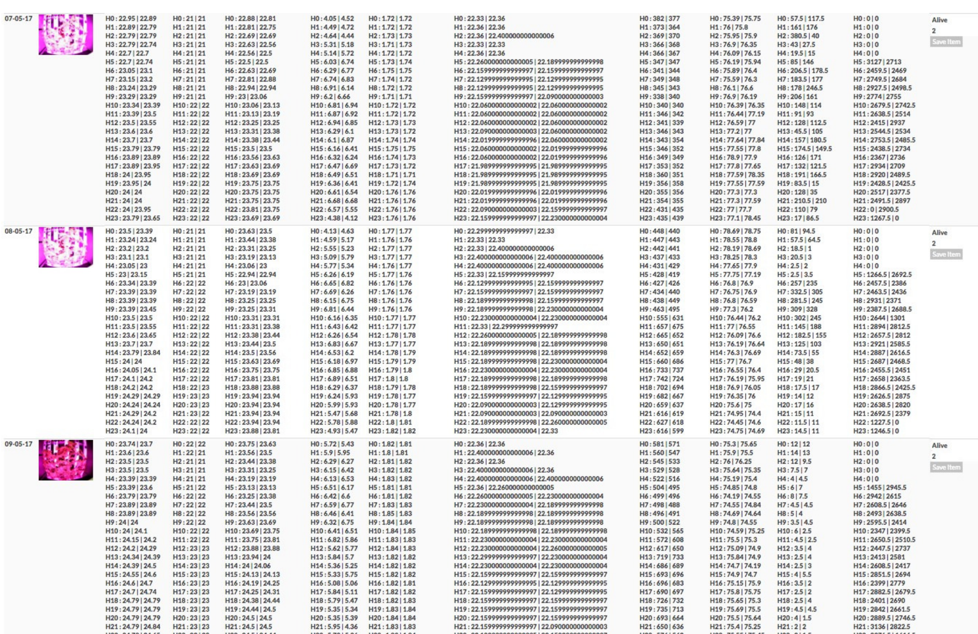

Figure 78. Analytic Dataset Web Form for manual analysis and classification

The analytic dataset was also used to plot all chart by hour and by day showed in Section 5. All charts were plotted using AngularJS and ChartJS at the SoP website control panel. It is possible to change between live real-time data from an ongoing cycle and the accumulated data from a specific cycle analytic dataset (Figure 79).

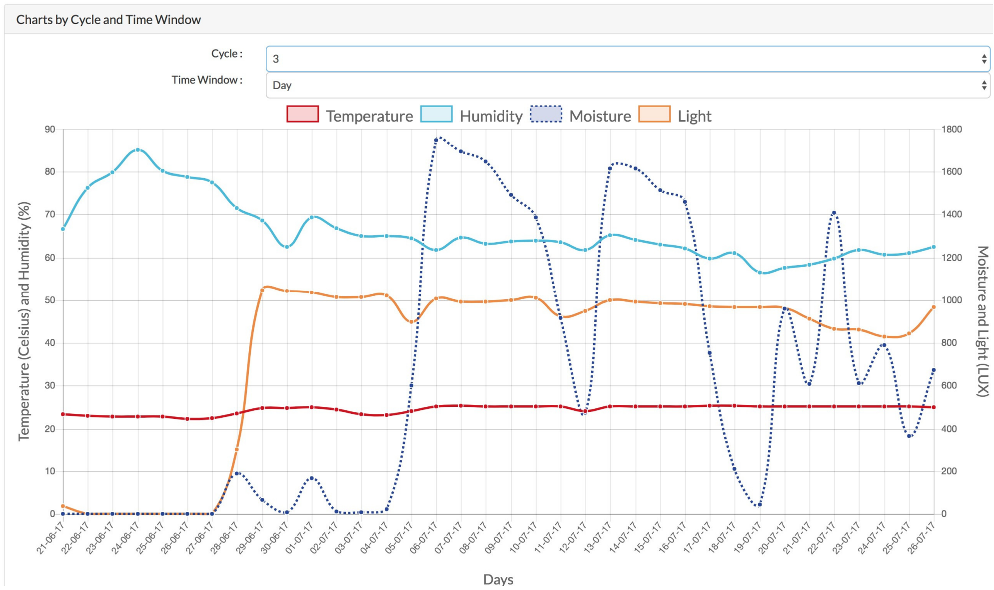

Figure 79. Cycle and Time Window Charts Selector

This real-time web based analytics was fundamental in all research cycles analyses, also comprising the SoP AoT stage (Section 4.1.2) and the m-PFAL management system (Section 5.2) in order to keep up with the system status and seedlings progress.

6.2. Machine Learning Analyses

After one year and five action research cycles, the analytic dataset ended with 12140 registers segmented in 349 days by hour and half an hour. Each day was manually classified in alive (61%) and dying (39%). Unfortunately for data analysis, all cycle one and the first three weeks from cycle 2 had to be discarded due to the lack of data from gas and water sensors, leaving the dataset with only 7990 registers.

An important observation is that the dataset cannot be random split for training and testing. The plants respiration, photosynthesis and growth are timebased events and have a strong influence on sensors values. Therefore, cycle 2 (with only living classification) and 3 (with alive and dying classification) were used for training and cycle 4 (with alive and dying classification) and 5 (with only alive classification) were used for test.

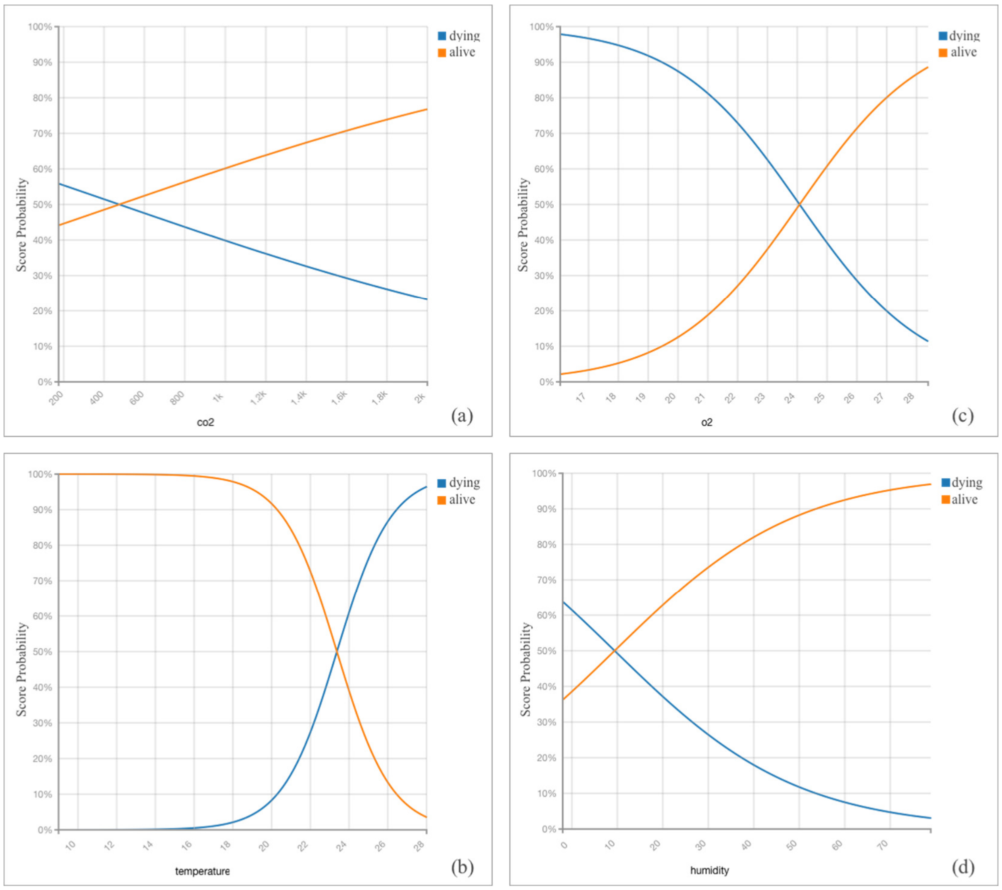

Figure 80. Logistic regression for: (a) CO2, (b) temperature, (c) O2 and (d) humidity

The dataset was analyzed using decision trees, logistic regression, time series and anomaly detection. All of them presented good accuracy but low precision. They cannot generalize beyond their training set because there is not enough data and the resulting dataset from each cycle are quite different. The algorithm that described best the system behavior was the logistic regression, the model trained with the logistic regression algorithm had an accuracy of 95.5% but its precision was only 48.11%. The aim of a logistic regression is to model the probability of an event that occurs depending on the values of the independent variables. The classification event was dying and alive, independent variables were CO_2, O_2, temperature and humidity. When breathing, plants emitted CO_2, and as they grow more CO_2 was emitted. The higher the CO_2 emission the more likely the plants are alive (Figure 80a).

Figure 80b shows that the higher the temperature, the more likely the plant will die, but given that in lower temperature the plants will stop to grow, it is necessary to find a balance. Figure 80c shows that if the oxygen is increasing over time, greater are the probability of the plants being alive, as evidence Figure 49 on Section 5.4.5. Finally, Figure 80d shows how important is to keep humidity over 70% as evidence Figure 55 on Section 5.5.5.

```
// To predict alive or dying fill the sensors data
// inputData: input values' object. Numeric fields are compulsory var inputData = {
        "temperature2": 23,
        "temperature1": 22.5,
        "o2": 22.3,
        "co2": 467,
        "humidity": 67
};
logisticregression.predict(inputData, function (error, data) { if(!error && data == 'dying')
setDamageControl(); });
```

Figure 81. Logistic Regression Prediction Function

This logistic regression model is a web function hosted in Azure and can be access using a HTTP POST within the IFTTT rules (Figure 81). This function should be used as tool in the knowledge base ensemble algorithms in order to help the system take action (SoPC AmI Stage - Section 4.1.3).

In the future, more SoP Cycles should be done in other to create more data that will improve this and other machine learning models in the knowledge base to improve the system accuracy and precision to recognize if the seedling is dying or alive with more confidence. The ultimate goal is to create as many datasets as possible for many types of crops to serve as a comprehensive catalog of epigenetic plants data. By applying machine learning in these datasets, it is possible to increase the plants nutrients, change the phenotypic expression that results in flavor, size and texture. The way to doing this is to set the plants in stressful environments such as drought, UV light, high temperature and chemical inductors that simulate predation from herbivorous [59]. The different combinations of these stressor can produce more nutrient and flavorfully plants as desired by the farmer, the dataset is the key for the next agriculture revolution to come.

7 Conclusion and Future Work

The aim of this thesis was to propose a model, entitled Software of Places Cycle (SoPC), that should be able to answer to environmental stimuli in a closed plant production system using artificial lighting in order to create a self-learning environment. This thesis describes the SoPC, the approaches and processes of implementing a mini Plant Factory using Artificial Lighting based on the discussion on five action-research cycles. The thesis main contribution is a conceptual model to guide the development and maintenance of a mini-PFAL (m-PFAL), a minor contribution is the deployment of the SoP, i.e., the very notion of having software dedicated to a specific place.

Section 1 gives an overview and the motivations for this research. Section 2, introduced the context for two important concepts of this thesis, Internet of Things and Plant Factories using Artificial Lighting. Section 3 provided related work to give the reader an overview of the state-of-the-art about what is currently happening in the world regarding to Food Computing, PFAL, AmI, Smart Lighting and Autonomic Computing. Section 4 outlined the Software of Places and the Software of Places Cycle. Section 5 reported the five research cycles I went through, following a typical action research cycle template. Section 6, summarized the resulting dataset from all five action research cycles.

The SoPC was extremely useful to build and manage the m-PFAL. The SoPC steps were revisited to prototype the m-PFAL, evaluate and evolve the system as demonstrated in each research cycle on Section 5. The research combination of plant factory using artificial lighting, its management and plant production system was a perfect fit to evidence the SoPC given the m-PFAL cyclical nature. The SoPC model also taught me that in order to pinpoint failure in the system, not just the software but also the hardware must be checked, sometimes the problem is not with the software, maybe one more sensor in the physical space is needed or the actual sensor is misplaced evidencing that the physical space is framing and cueing software behavior.

The proposed SoPC model actually supported the experiments in five plant production process with various combinations of cultivation environment, including light intensity, temperature, humidity, CO_2, pH, and EC, depending on

growth stages. It also provided support for various functions, including integrated environment and growth monitoring, device control, alarm system, trend analysis, and logging. The integrated data collected from the sensors in the system are useful for calculating optimal environment parameters, while the active control of devices improves plant growth and quality. I believe that the SoPC could also support the design of similar technologies in other environments.

Moreover, the SoPC is also useful for the teaching of a wide range of topics, including but not limited to IoT, Machine Learning, Distributed Systems, Data Base, Ubiquitous Computing, Wearables, and Ambient Intelligence. The SoP Cycle was slowly developed after three years teaching graduate students how to think software beyond computers and mobile phones. Above all, it helped teaching IoT and the requirements that physical spaces impose over the system. The SoPC Physical Environment and Connectivity axes represent the magnitude order between the physical area comprised by the system and the connectivity technologies necessary to encompass them. It looks like that physical space growth behaves in a non-linear way in terms of energy, connectivity and management complexity. The Connectivity, Cloud and Physical Environment axes represent the relations between the network protocol, data volume, predictive model and where they will reside, whether on the cloud or on the edge (or fog computing). Besides that, the m-PFAL is a complex engineer system that can be used not only as a research platform but also as an educational tool. This research findings should be used to design specifically balanced cultivation system for supporting plant growth for close urban agriculture environments.

For future work, the system needs to be constantly in production, given that more data need to be produce in order to improve the machine learning models. Using machine learning, it is not only possible to optimize the predictive system maintenance, its micro-climate to sustain plants life and optimum growth, but also change the phenotypic expression that changes flavor, size, texture, and production cost for a myriad of crops. Future work needs include developing an algorithm to analyze correlations between the integrated data using statistical methods, data mining, and neural networks, and to optimize the system through design patterns. In addition, computer vision algorithms should be introduced to measure leafs growth in order to help determine the plant mass and growth rate.

References

[1] W. Mitchell, “City of bits,” *City Bits*, pp. 159–173, 1995.

[2] M. McCullough, *Digital Ground: Architecture, Pervasive Computing, and Environmental Knowing*. 2004.

[3] J. S. Rellermeyer, M. Duller, K. Gilmer, D. Maragkos, D. Papageorgiou, and G. Alonso, “The software fabric for the Internet of things,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2008, vol. 4952 LNCS, pp. 87–104.

[4] T. Kozai, “Plant Production Process, Floor Plan, and Layout of PFAL,” in *Plant Factory: An Indoor Vertical Farming System for Efficient Quality Food Production*, 2015, pp. 203–212.

[5] T. Kozai and G. Niu, “Challenges for the Next-Generation PFAL,” in *Plant Factory: An Indoor Vertical Farming System for Efficient Quality Food Production*, 2015, pp. 387–393.

[6] T. Kozai and G. Niu, “Plant Factory as a Resource-Efficient Closed Plant Production System,” in *Plant Factory: An Indoor Vertical Farming System for Efficient Quality Food Production*, 2015, pp. 69–90.

[7] H. Shimizu, K. Fukuda, Y. Nishida, and T. Ogura, “Automated Technology in Plant Factories with Artificial Lighting,” in *Plant Factory: An Indoor Vertical Farming System for Efficient Quality Food Production*, 2015, pp. 313–319.

[8] K. Ashton, “That ‘Internet of Things’ Thing,” *RFiD J.*, p. 4986, 2009.

[9] M. McCullough, *Ambient Commons*, vol. 53, no. 9. 2013.

[10] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, “Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications,” *IEEE Commun. Surv. Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

[11] Y. Qin, Q. Z. Sheng, N. J. G. Falkner, S. Dustdar, H. Wang, and A. V. Vasilakos, “When things matter: A survey on data-centric internet of things,” *J. Netw. Comput. Appl.*, vol. 64, pp. 137–153, 2016.

[12] S. Li, L. Da Xu, and S. Zhao, “The internet of things: a survey,” *Inf. Syst. Front.*, vol. 17, no. 2, pp. 243–259, 2015.

[13] the N. I. Council, N. Nic, and S. C. B. Intelligence, “Disruptive Civil Technologies Six Technologies With Potential Impacts on US Interests Out to 2025,” *Natl. Intell. Counc.*, no. April, p. 48, 2008.

[14] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, “Internet of Things (IoT): A vision, architectural elements, and future directions,” *Futur. Gener. Comput. Syst.*, vol. 29, no. 7, pp. 1645–1660, 2013.

[15] T. Kozai, *PFAL Business and R&D in the World*. 2016.

[16] T. Kozai, S. Sakaguchi, T. Akiyama, K. Yamada, and K. Ohshima, “Design and Management of PFAL,” in *Plant Factory: An Indoor Vertical Farming System for Efficient Quality Food Production*, 2015, pp. 295–312.

[17] C. Harper and M. Siller, “OpenAG: A Globally Distributed Network of Food Computing,” *IEEE Pervasive Computing*, vol. 14, no. 4. pp. 24–27, 2015.

[18] R. M. Wheeler, “Horticulture for mars,” in *Acta Horticulturae*, 2004, vol. 642, pp. 201–215.

[19] M. Cunha and H. Fuks, “AmbLEDs collaborative healthcare for AAL systems,” in *Proceedings of the 2015 IEEE 19th International Conference on Computer Supported Cooperative Work in Design, CSCWD 2015*, 2015, pp. 626–631.

[20] M. Cunha and H. Fuks, “AmbLEDs Para Ambientes De Moradia Assistidos Em Cidades Inteligentes,” in *Proceedings of the 13th Brazilian Symposium on Human Factors in Computing Systems*, 2014, pp. 409–412.

[21] R. F. Karlicek, “Smart lighting - Beyond simple illumination,” in *2012 IEEE Photonics Society Summer Topical Meeting Series, PSST 2012*, 2012, pp. 147–148.

[22] F. Deicke, W. Fisher, and M. Faulwasser, “Optical wireless communication to eco-system,” *Futur. Netw. Mob. Submit*, pp. 1–8, 2012.

[23] M. Cunha, “AmbLEDs: Context-Aware I/O for AAL Systems,” *Proc. 20th Int. Conf. Intell. User Interfaces Companion*, no. Vlc, pp. 137–140, 2015.

[24] M. Cunha and H. Fuks, “AmbLEDs : Implicit I / O for AAL Systems,” in *Proceedings of the 20th International Conference on Intelligent User Interfaces - IUI ’15*, 2015.

[25] M. Weiser, “The computer for the 21st Century,” *IEEE Pervasive Comput.*, vol. 1, no. 1, pp. 19–25, 2002.

[26] D. López de Ipiña and S. Lai Lo, “Sentient Computing for Everyone,” in *The Third International Working Conference on New Developments in Distributed Applications and Interoperable Systems*, 2001, pp. 41–54.

[27] M. Addlesee *et al.*, “Implementing a sentient computing system,” *Computer (Long. Beach. Calif).*, vol. 34, no. 8, pp. 50–56, 2001.

[28] F. Stajano, “Security Issues in Ubiquitous Computing,” in *Handbook of Ambient Intelligence and Smart Environments*, 2010, pp. 281–314.

[29] J. C. Augusto, “Ambient intelligence: The confluence of ubiquitous/pervasive computing and artificial intelligence,” in *Intelligent Computing Everywhere*, 2007, pp. 213–234.

[30] M. C. Huebscher and J. A. McCann, “A survey of autonomic computing—degrees, models, and applications,” *ACM Comput. Surv.*, vol. 40, no. 3, pp. 1–28, 2008.

[31] J. O. Kephart and D. M. Chess, “The vision of autonomic computing,” *Computer (Long. Beach. Calif).*, vol. 36, no. 1, 2003.

[32] P. Horn, “Autonomic Computing: IBM’s Perspective on the State of Information Technology,” *IBM Corporation*, vol. 15. pp. 1–39, 2001.

[33] K. Vega, M. Cunha, and H. Fuks, “Hairware,” *Proc. 20th Int. Conf. Intell. User Interfaces - IUI ’15*, pp. 78–86, 2015.

[34] N. Shadbolt, “Ambient Intelligence,” *IEEE Intell. Syst.*, vol. 18, no. 4, pp. 2–3, 2003.

[35] J. Fraden, *Handbook of modern sensors: Physics, designs, and applications*. 2016.

[36] B. Metcalfe, “Metcalfe’s law after 40 years of ethernet,” *Computer (Long.*

Beach. Calif)., vol. 46, no. 12, pp. 26–31, 2013.

[37] A. Bulling, U. Blanke, and B. Schiele, “A tutorial on human activity recognition using body-worn inertial sensors,” *ACM Comput. Surv.*, vol. 46, no. 3, pp. 1–33, 2014.

[38] D. Figo, P. C. Diniz, D. R. Ferreira, and J. M. P. Cardoso, “Preprocessing techniques for context recognition from accelerometer data,” *Pers. Ubiquitous Comput.*, vol. 14, no. 7, pp. 645–662, 2010.

[39] T. Huynh and B. Schiele, “Analyzing features for activity recognition,” in *Proceedings of the 2005 joint conference on Smart objects and ambient intelligence innovative context-aware services: usages and technologies - sOc-EUSAI ’05*, 2005, p. 159.

[40] T. Plötz, N. Y. Hammerla, and P. Olivier, “Feature Learning for Activity Recognition in Ubiquitous Computing,” *Proceeding IJCAI’11 Proc. Twenty-Second Int. Jt. Conf. Artif. Intell.*, vol. Volume 2, pp. 1729–1734, 2011.

[41] I. Guyon and A. Elisseeff, “An Introduction to Variable and Feature Selection,” *J. Mach. Learn. Res.*, vol. 3, no. 3, pp. 1157–1182, 2003.

[42] F. Chen, P. Deng, J. Wan, D. Zhang, A. V. Vasilakos, and X. Rong, “Data mining for the internet of things: Literature review and challenges,” *International Journal of Distributed Sensor Networks*, vol. 2015. 2015.

[43] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, “Context aware computing for the internet of things: A survey,” *IEEE Commun. Surv. Tutorials*, vol. 16, no. 1, pp. 414–454, 2014.

[44] U. Blanke and B. Schiele, “Daily routine recognition through activity spotting,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2009, vol. 5561 LNCS, pp. 192–206.

[45] S. McConnell and D. Skillicorn, “A distributed approach for prediction in sensor networks,” *SIAM Int’l Conf. Data Min. Work. Sens. Networks*, 2005.

[46] M. J. Akhlaghinia, a Lotfi, C. Langensiepen, and N. Sherkat, “Occupant behaviour prediction in ambient intelligence computing environment,” *J. Uncertain Syst.*, vol. 2, no. 2, pp. 85–100, 2008.

[47] J. S. Sandhu, A. M. Agogino, A. K. Agogino, and others, “Wireless sensor networks for commercial lighting control: decision making with multiagent systems,” *AAAI Work. Sens. networks*, vol. 10, pp. 131–140, 2004.

[48] U. Rutishauser, J. Joller, and R. Douglas, “Control and learning of ambience by an intelligent building,” *IEEE Trans. Syst. Man, Cybern. Part ASystems Humans.*, vol. 35, no. 1, pp. 121–132, 2005.

[49] G. I. Susman and R. D. Evered, “An Assessment of the Scientific Merits of Action Research,” *Adm. Sci. Q.*, vol. 23, no. 4, p. 582, 1978.

[50] N. F. Kock, R. J. McQueen, and L. S. John, “Can action research be made more rigorous in a positivist sense? The contribution of an iterative approach,” *J. Syst. Inf. Technol.*, vol. 1, no. 1, pp. 1–23, 1997.

[51] J. K. Schueller, “Implications for plant factories from manufacturing and agricultural equipment technologies.,” in *Proceedings of Plant Factory Conference*, 2014, p. 5.

[52] O. Nunomura, T. Kozai, K. Shinozaki, and T. Oshio, “Seeding, Seedling Production and Transplanting,” in *Plant Factory: An Indoor Vertical Farming System for Efficient Quality Food Production*, 2015, pp. 223–235.

[53] S. Sakaguchi, T. Akiyama, K. Yamada, and K. Ohshima, “Chapter 22 – Design and Management of PFAL,” in *Plant Factory*, 2016, pp. 295–312.

[54] A. Sheth, C. Henson, and S. S. Sahoo, “Semantic sensor web,” *IEEE Internet Comput.*, vol. 12, no. 4, pp. 78–83, 2008.

[55] D. Pfisterer *et al.*, “SPITFIRE: toward a semantic web of things,” *Commun. Mag. IEEE*, vol. 49, no. 11, pp. 40–48, 2011.

[56] S. Rachmilevitch, H. Lambers, and B. Huang, “Root respiratory characteristics associated with plant adaptation to high soil temperature for geothermal and turf-type Agrostis species,” *J. Exp. Bot.*, vol. 57, no. 3, pp. 623–631, 2006.

[57] C. Kubota, “Growth, Development, Transpiration and Translocation as Affected by Abiotic Environmental Factors,” in *Plant Factory: An Indoor Vertical Farming System for Efficient Quality Food Production*, 2015, pp. 151–164.

[58] Y. Kitaya, J. Tsuruyama, T. Shibuya, M. Yoshida, and M. Kiyota, "Effects of air current speed on gas exchange in plant leaves and plant canopies," *Adv. Sp. Res.*, vol. 31, no. 1, pp. 177–182, 2003.

[59] C. Harper, "Open-source agriculture initiative-food for the future?," in *LED Lighting for Urban Agriculture*, 2016, pp. 37–46.