TEMP & CAP

Angel Garcia Calleja, Youssef Assbaghi Asbahi

Abstract— Throughout this report we will explain in detail the realization of the practice of the Internet of Things subject. The project to be developed consists of the creation of a system that will detect when someone enters a room and then take measures to reach the desired temperature previously set. In addition, the system will have a capacity counter, which will send a notification to the user once the maximum capacity has been reached. Through the development and implementation of this system, the aim is to help both people not to waste their money when paying the electricity bill and stores or restaurants to have the capacity controlled.

Index Terms—Cloud, Device, Edge, IA, Java, Server, Android System, Arduino, Flutter, BLE, AWS, AWS EC2, AWS S3, Router

1 INTRODUCTION

In this section of the report, we are going to detail the idea and motivation of the project, as well as to explain in depth what our project consists of.

One of the problems that Spaniards have been suffering in recent months has been the drastic increase in the cost of electricity.

This problem is also linked to the insufficient progress in the efficiency of energy consumption in the field of heating and cooling.

If we put these two problems together, we can see how we are paying a lot for something that is being used inefficiently.

Our project focuses on making sure that we are not wasting energy in places where it is not needed, for example, if we are working in our office, we do not need to have the heating of our room on.

Once we started with this idea, we realized that we could be more efficient with the resources we would use, since we were going to detect when a person enters or leaves a room, we could add a capacity counter for that place. This functionality would be useful for a larger number of potential customers as it would also be useful for places such as stores, nightclubs or restaurants, since in this day and age, many of these places have capacity restrictions because of Covid-19.

UAB Internet de les Coses

- *Youssef Assbaghi Asbahi E-mail: Youssef.assbaghi@e-campus.uab.cat*
- *Angel Garcia Calleja. E-mail: angel.garcia.calleja@e-campus.uab.cat*

The project has three distinct parts. Device, Edge and Cloud.

In the Device section we find the components that will generate the data. These data will be obtained through a thermal imaging camera from which we will obtain the information about the number of people in the room at all times, and through a temperature sensor, from which we will obtain the current temperature data.

This data will be passed via Wifi directly to the Cloud, which will consist of an AWS server contracted by us.

The Cloud will be a server where we will store the data of the temperature and capacity preferences of each user. The Cloud will consist of an AWS application which runs Linux that will receive the data via wifi from the Device in JSON format. Once here, the current temperature and capacity data received will be compared with the user's preferences. These comparisons can be observed in the Front End in an html webserver.

In the Front End we will have a system application, which will show all the data obtained from the Cloud.

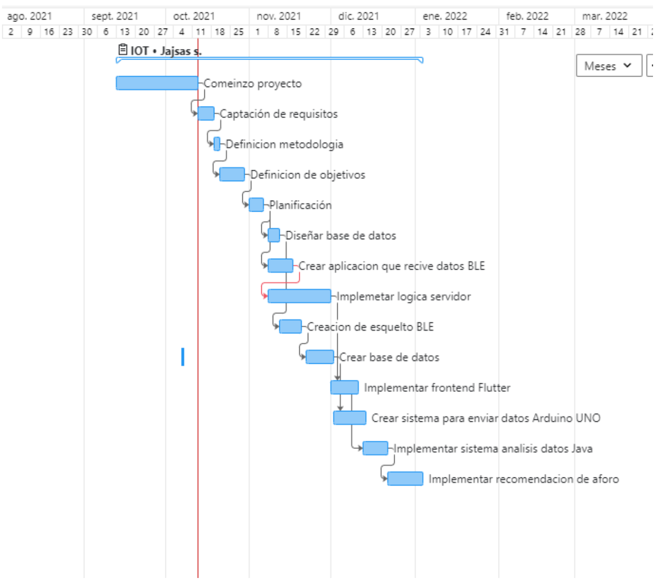
2 OBJECTIVES

The main objective of this project can be summarized as helping people not to waste their energy. Other objectives could be:

- Regulate the use of AC and heating.
- Monitor the capacity of the site
- Keeping the user informed of the data
- Designing the system with the lowest cost
- System design with acceptable performance

3. Planification

This is the first planifaction approach of this project. As we are two in the team we have to work very well in parallel. The firsts weeks are hard because we have to get a good design of the application and the format of the data because we have to work in different sides together at the same time. So, we have to proceed in a good communication system and a good designment and requirements. The GANTT diagram shows us the way that we will develop the project. The first weeks we have to work together to create a good design but after these weeks we can work in parallel to get a better performance. We have to mention that in the appendix A1 is the full GANTT diagram.



After that we decided to creat a small checklist of forseen risks and contingency plan.

RISK	CONTINGENCY PLAN
Can't synchronize all sensors together.	Remove one of the sensors and fous on an accurate one
Don't have time for the client side.	Change flutter ide with a simple html web
Problems with the algorithm of recommendation	Change the algorithm for a trained ML
Problems to connect with the server in the android system	Create an NGROK server and use that develop environment
Problems with compatibility of system Device-Edge-Cloud	Change the format of the data sended.

4 Market Analysis

We looked for all tools that detect temperature and count the number of people in a room and we found out that they are very expensive and not too accurate. Also, it's hard to find a machine that does both things and the one we found costs 2000€.

We found some objects that are quite similar to what we want like Controlador de temperature "Reference 1" which only controls the temperature of a house and costs 188€ without IVA. We found another one a bit cheaper, but it works analogically without Bluetooth and doesn't hace a capacity counter, it's the Interruptor de termostato "Reference 2" which only costs 11€.

In the other one we found some better products, but they are more expensive, like Control de aforo basado en inteligencia "Reference 3" which costs 269€ and Automatizacion de puertas con control de aforo "Reference 4" which cost 1850€.

So, as we didn't find any product which is a bit cheaper and has a good temperature measurement and also counts the number of people entering a room. So, we will create a product that allows us to get both things for a lot less money.

5 Specifications

Our product consists of a multiplatform product which allows to control the temperature and the number of persons in a room. It has 4 sides, the device, the cloud and the client.

5.1 Functional requirements

The connectivity between the different sensors and the device will be wired, while the connectivity between device, edge and cloud will be wireless.

We plan to have a power consumption range between 1.5A and 4.6A, and between 5W to 10W.

The device will be screwed to the wall near the door, so we will put the device in a box with IP62 protection to protect it from dust and small drops of water. Because of this, the dimensions of the device cannot exceed 10x10x5cm and 500g in weight.

For the user interfaces we have the following in mind for the various parts of the system.

Device: we will have a button to turn on/off the device with its corresponding LED to show the system status (off - looking for connection, blinking - and working - green light).

Front End: we will have a web appliaction where we will be able to connect to the Device. The web application will allow to select the temperature and the maximum capacity, apart from showing the current temperature and capacity values. And showing the current state of the site's domotic (AC and heating).

5.2 Perfomance

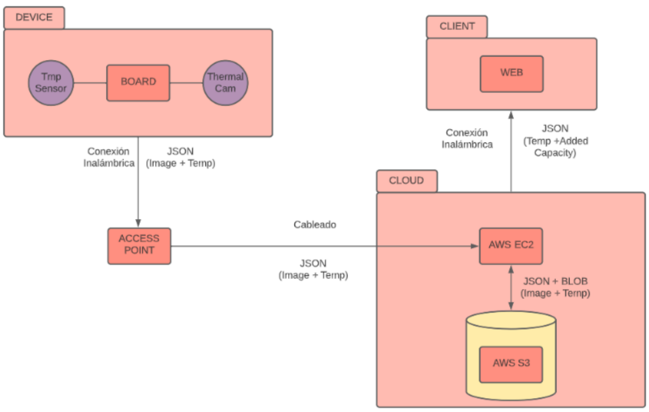
As for the power supply, the device will be connected to the power supply so as not to have to recharge the batteries, although we are also thinking of a wireless charging system as an alternative, which would have an autonomy of 26h.

As for the connection range between the Device and the Edge, it should not exceed 45m, in case of a larger establishment, the connectivity between the Device and the repeater will be wired.

The network topology between Device and Edge shall be linear, while between Edge and Cloud shall be a star topology.

The bandwidth frequency will be 2.4 GHz, with a latency of less than 10ms, giving a speed of 150 Mbps if there is no data loss.

6 System architecture



The device will consist of a board, a temperature sensor and a thermal camera. The temperature sensor will give us the value of the ambient temperature and the thermal camera will give us a picture of the door where we can observe people entering and leaving the room.

The sending of these two data between Device and Edge will be done through a wireless Wifi connection.

From the Edge they will pass immediately to the Cloud via wired connection, which means that the Edge only acts as a bridge to the Cloud with optic fiber.

Once in the Cloud (AWS), the data is sent to the database (AWS S3). The image will be saved for model learning. Using the algorithm in (AWS EC2), the image will be "translated" into a floating value.

Finally, the Cloud sends via Wifi wireless connection to the Client (Web Application) a JSON file with the current Temperature data and the variation of capacity.

The Client will calculate the new capacity with the data received from the Cloud and will launch the warnings according to the limits preset by the user.

6.1 System alternatives

Board:

Dispositivo			
Datos/Aparato	Arduino UNO Wifi	Raspberry PI3 A+	Arduino UNO R3
RAM	48 Kb Flash, 6 Kb Sram, 256 Kb Eeprom	512 MB LPDDR2	32 KB Flash (0,5 para bootloader), 2KB RAM y 1KB Eeprom.
Conectividad	WiFi	802.11 b/g/n/ac de doble banda (2,4 y 5 GHz) y Bluetooth 4.2	Con pines para conectar modulos
Dimensiones	68.6 x 53.4 mm	9.6 x 7.11 x 2.69 cm	8 x 5.51 x 2.49 cm
Alimentacion	7 - 12V	5 V / 2,5 A DC	: 7,5 - 12 v
Otros	Incorpora IMU y módulo criptográfico	OpenGL ES 1.1, gráficos 2.0	Necesita modulo Wifi y BLE
Peso	25g	50g	31.75 g
Precio	38,90 €	27,50 €	20 €

As we can see, the cheaper option doesn't come with a Wifi module, which would increase the cost.

The Raspberry P13 A+ comes with a really helpful image processor which would be really usefull for us.

Temperature sensor:

Sensor de temperatura y humedad		
Datos/Sensor	DHT22	DHT11
Precision humedad	±2%	±5%
Precision temperatura	±0.5ºC	±2ºC
Alimentacion	3,3V ~ 5,5V	3,3V ~ 5,5V
Precio	6,95	1,6

The DHT22 gives us more accurate results without increasing the cost by too much.

Thermal camera:

Cámara termica		
Datos/Sensor	Grid-EYE AMG88	FLIR Lepton 2.5
Pixels	8x8	80x60
Temperatura	-20ºC A 80ªC	-10ºC A 80ªC
Frame rate	10fps	8.6FPS
Angulo de visio	60º	63,5º
Precio	25,40 €	134 €

As we can see, both options have pros and cons, since it's a very important part of our project we will delay making the decision.

Actual size

Device – Edge – Cloud protocols:

PROTOCOLOS DEVICE-EDGE			
DATOS/PROTOCOLO	802.11ah	BLE	Zigbee
Conectividad	WiFi HaLow	Buetooth 5.0	Zigbee 3.0
Frecuencia	2.4Ghz	2.4Ghz	2.4Ghz
Latencia	<10ms	80-90ms	20ms-220ms
Ancho de banda	150kbps-346Mbps	1Mbps	250kbps
Rango	1km	30m-130m	10-100m
Duracion bateria	dias	hasta 2 años	6meses - 2años

For the communication protocols, 802.11aH gives us a really low latency and a far larger range.

Cloud:

Cloud			
Características/tipo	Privado	Publica (AWS)	Hibrida (Google Cloud
Escalabilidad	Limitado a los recursos de la empresa	Muy alto	Muy alto
Seguridad	Depende de la inversiopn en seguridad	Bastante seguro	Muy seguro
Rendimiento	Muy Bueno	Medio o bajo	Bueno
Disponibilidad	Alto	de Internet y Amazon	Media Alta
Coste	€€€	€	€€

A private Cloud offers us a really high availability and a very strong performance, and we also don't rely on third parties.

7. System solution

After considering the different alternatives of components, protocols and communications, the final system solution will be explained below.

7.1 Device

The Device contains all the components that will allow us to obtain the data for the correct operation of the system. We find a board, a temperature sensor and a thermal camera. With these components we will obtain the value of the current temperature, as well as images of the access to the enclosure to determine the number of people entering and leaving, therefore, we will be able to determine the number of people inside the enclosure at all times.

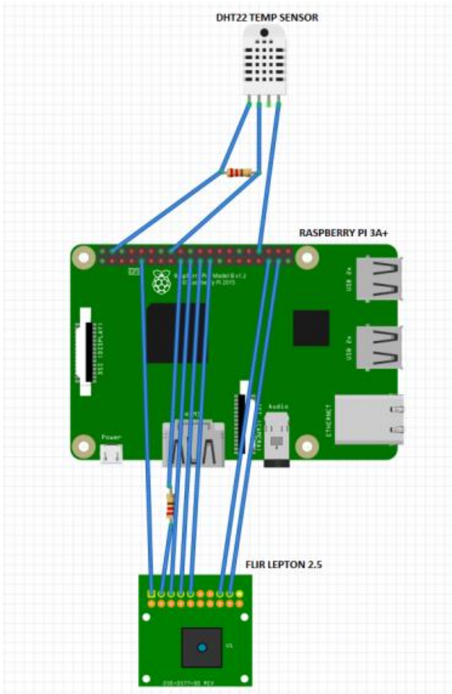
7.1.1 Components

First, we find the Raspberry PI3 A+ board [5]. We have chosen this board because it offers us a larger RAM than the other options we were considering, we don't need to add any extra modules and it comes with a very useful image processor for our project. All this without being very expensive.

For the temperature sensor we have decided to stick with the DHT22 [6], as it offers better accuracy without inflating the cost too much.

Finally, the thermal camera we have chosen has been the Teledyne FLIR Lepton 500-0763-01 [7]. With this camera we obtain images of 80x60 at 9 FPS, although the price is higher, we get a quality necessary for further processing of the images.

Here we can see the scheme of the different components.



7.1.2 Power supply

For the power supply, our Device will be connected to the mains via a power adapter, thus saving us the problem of batteries. Our Raspberry board needs 5V and 2.5A, so the adapter should have those characteristics. For the DHT22 temperature sensor you only need to connect the VCC power pin to 3-5V, the GND pin to Ground (0V) and the data pin to a pin of our board. Our Teledyne FLIR Lepton 500-0763-01 camera needs about 3V, so we should not have power supply problems in the circuit. To all this, it is recommended to use a resistor of about 5k-10k Ohms for the correct operation of the system.

In terms of consumption, we obtain the following values. According to the official source our Raspberry board works at 5V and provides a maximum current of 3A, but in reality, the intensity is about 1.3A at maximum stress, which is equivalent to 6.7W. The temperature sensor consumes about 10mW and our camera about 150mW, which in total adds up to 6,860W, so we can place the average consumption around 6.9W.

For our current system, there would be no need to add PCBs, unless the system needed to be scaled up.

7.1.3 Communication between components

The protocol used between our temperature sensor and the board will be PWM. For the communication between the board and the thermal camera we will use the I2C protocol. This is a bus with only two wires, one for clock and one for data with which we can make transmissions of up to 400kbits/s.

7.1.4 Data

For the correct operation of the system, we will need to send to the Edge and then to the Cloud the ID of the camera used (useful in case the system needs to be scaled), the images of our thermal camera and the temperature value collected by our sensor.

The bytes needed for the ID and the temperature we can determine will be an int and a float respectively, so 2 and 4 bytes.

To calculate the bytes of the images we need to do several calculations. First, we multiply the image resolution by the number of channels, in this case 80x60x8 = 38.800 bits. Next, we multiply that value by the FPS of the camera 38.800 bits x 9 FPS = 345600 bits, finally we pass the bits to bytes dividing by 8, which gives us 43,200 bytes, or what is the same, 43.2kB.

7.2. Communication Device-Edge

Given the amount of data we want to send, the communication between our Device and Edge will be done via Wifi. With Wifi we have a very low latency of up to less than 10ms and more than enough bandwidth for our data, so we will not cause a bottleneck. We will use the FTP application layer protocol. The dataframes of this protocol allow us, excluding headers, to send up to 239 bytes of data.

7.3 Edge

Our last decision is to remove an edge system. That's because the RaspBerry can create a connection directly with the server and it reduces our number of systems. So, we will need a router to act as a gatewall between the device and the server to connect both sides.

7.4 Communication Edge-Cloud

The connection between the router and the server will be wired with a fiber network. This will allow us to handle all the data as we have a in this connection almost 900Mbps of data transfer. The protocol will be the same as the device-edge, will be an http protocol wich will allow us to connect and acknowledge all the data sended.

7.5 Cloud

Finally, we have decided to add a public cloud offered by Amazon. This way we will have storage, computing resources and databases without the need for a very high CAPEX. Although the availability is not the highest, as our system is simple, we will not need large resources and unnecessary complexity. In addition to the OPEX is a little high as we ask enough resources to the cloud both computing and storage, so we can ensure optimal operation of the system. In this way Amazon Web Service S3 will be our cloud service provider with an estimated price of 47€ per month. As we can see in the image below it will offer us 2000GB of storage, 1500 PUT requests and COPY, 1500 GET requests and SELECT, and will return us about 200GB of data with a total scan of 500GB.

This in the part of storage. The computer platform that we will use is AWS EC2. This allows us to get a running server with high benefits and low CAPEX. Our configuration of this cloud will be very basic. The server will run a Linux operative system with 8 cores and 16GB of RAM. This server will have 30GB to store the coding of the server and the trained model (The images will be stored in S3). We calculate that the server will be running at 100% all the time and it will have 5 gigabit per second of internet, which means that the server will not have any bottleneck.

7.5.1 Back-end

The backend will receive data from the device and store it in the AWS database. This data will then be used by the ML algorithm to make its predictions and indicate whether people have entered or left the premises. Once the prediction is generated, it will be stored in the database so that the user can consult it every time he enters the application.

7.1.4 Front-end

The front end will be created by the cloud as an html web server. This allows us to take the data from the server and show it dynamically on the front-end.

8. Data analysis

The analysis of the temperature data and enclosure gauging is easy to analyze. The system will compare the values previously selected by the user with the values received from our Device. In the case of exceeding the values, a warning will be displayed in the front-end, where the user will be able to observe the data at all times.

For the subject of people recognition, we will use Machine Learning algorithms to identify people in the images, an example could be the unsupervised classification algorithm k-means, for the classification and processing of images.

9. System programming

The device will be programmed in Python. That's because is one of the best languages for RaspBerry Pi 3A+. It will use PWM to get the data from the Temperature sensor and the protocol I2c to take the data from the camera, for this communication we will use the simbus library which we send him and id of sensor and it return us the data of that sensor. After that we will use the ftplib to send the data to the server.

The edge doesn't have any code because it will be a router who acts as a gateway.

The cloud is programmed in java and has scripts in python to get a proper data analysis. Also, the fornt end will be coded in html, css and javascript.

10. Estimations

For the estimations we have used the data we have collected so far in order to obtain a first cost and scalability data.

10.1 System scalability

Regarding the scalability of the system, we have two extremes, the minimum and the maximum. Our aws cloud, as it is the basic service, we will be able to support about 3500 requests per second and each request with a size of 16MB/s. We have in mind that the client works with fiber optic wifi of minimum 100mb/s in order to ensure that there are no performance problems with the bandwidth, otherwise the system will work with delay.

The minimum scalability of the system consists of a board, a temperature sensor, a camera, and the cloud running on AWS. This system has no performance limi-tations since the board is sufficient to process the image and send it to the cloud and since each request weighs about 43.2kB/s we will have no problems with either the number of AWS requests or the size per request of the cloud throughput. Since it is a very small request size.

On the other hand, the maximum scalability is limited by the number of AWS cloud requests. Assuming that the maximum memory bandwidth for transferring im-ages is 6MB/s, this means that we can add a total of 120 cameras per device operating synchronously and 120 temperature sensors. Add that for this configuration we will have to add a cost overrun of GPIO pin expansion boards that will help us to connect all the systems. Therefore, again the limitation will be the cloud requests since the data transfer rate with this configuration will be 6.5MB/s, which with an adequate internet and cor-rect wifi the bottleneck will be formed by the 3500 re-quests of the aws that offers us.

10.2 Cost

Regarding the cost we divide it into two sections, the unit cost of the device and the cost of the cloud. We must emphasize that the cost of our device is of a min-imum scalability and that any configuration that you want to add will increase the cost of the product.

10.2.1 Unitary costs

The total cost of the device is 195,54€. We have to get conscious that the expansion board and the wires are an extra that they will be charged if the customer needs it.

- Device:
  - Raspberry PI 3A+ board: 26,5€.
  - Temperature Sensor DHT22: 6,95€.
  - Teledyne FLIR Lepton thermal camera: 139,13€
  - Power Adapter: 10,99€.
  - Board protection box: 3,80€.
  - Custom 3D print camera mount: €2,17
  - SD memory for 2MB board: 7,03€
  - SD memory for 6MB board: 5,03€
  - Board wires: 2m - 1€ 1€ 1m
  - Expansion board: 3,5€.

10.2.2 Cloud costs

As we know we will contract an amazon web server s3. With the actual configuration the total costs of the cloud will be 47€.

SERVIDOR AWS S3		
Producto	Cantidad	Precio (USD)
Almacenamiento de S3 Estándar	2000 GB	46
Solicitudes PUT, COPY, POST y LIST	1500	0,0075
Solicitudes GET, SELECT	1500	0,0006
Datos devueltos	200 GB	0,14
Datos escaneados	500 GB	1
	Total Mes	47,1481

The cloud with actual configuration will cost around 126,38€ which means all the cloud system will have a cost of 173,52€

Amazon EC2	
Caractiristica/Unidad	Unidad
Sistema Operativo	Linux
CPU	8 Cores
RAM	16 GB
Memoria SSD	30 GB
Red	5Gb/s
Precio Total	126,38 €

10.3 Planning

For the planning of the project, we have used the Microsoft Project tool to define the tasks, define the tasks dependent on each other, the times of the tasks, the personnel assigned to each task and the costs of each type of personnel.

In order to show the planning of our project with a Gantt Chart, we have shown the image of it in Appendix a1.

For the selection of tasks, we have based ourselves on projects similar to ours, as well as on our experience developing projects throughout our career.

Below is the list of all project tasks, with their respective durations, start and end dates and predecessor tasks:

Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
PROYECTO IOT GRUPO 6	**105 días**	**lun 10/01/22**	**vie 03/06/22**	
FASE 1 (Inicio)	**7 días**	**lun 10/01/22**	**mar 18/01/22**	
Captación de requisitos	5 días	lun 10/01/22	vie 14/01/22	
Definición objetivos	3 días	lun 10/01/22	mié 12/01/22	
Documento especificación sistema inicial	2 días	lun 17/01/22	mar 18/01/22	3
FASE 2 (Estudi)	**8 días**	**mié 19/01/22**	**vie 28/01/22**	**2**
Estudio de viabilidad	**5 días**	**mié 19/01/22**	**mar 25/01/22**	
Evaluación coste HW	3 días	mié 19/01/22	vie 21/01/22	
Definición presupuesto inicial	2 días	lun 24/01/22	mar 25/01/22	8
Definición planificación inicial	3 días	mié 26/01/22	vie 28/01/22	7
FASE 3 (Contratación)	**11 días**	**lun 31/01/22**	**lun 14/02/22**	**6**
Búsqueda, acuerdos y contratación de licencias SW	3 días	lun 31/01/22	mié 02/02/22	
Búsqueda, acuerdos y contratación de Cloud de almacenamiento	3 días	jue 03/02/22	lun 07/02/22	12
Búsqueda, acuerdos y contratación de personal	5 días	mar 08/02/22	lun 14/02/22	13
Búsqueda, acuerdos y contratación de proveedores	5 días	jue 03/02/22	mié 09/02/22	12
FASE 4 (Diseño y planificación)	**36 días**	**mar 15/02/22**	**mar 05/04/22**	**11**
Planificación final	3 días	mar 15/02/22	jue 17/02/22	
Definición presupuesto final	2 días	mar 15/02/22	mié 16/02/22	
Documento especificaciones de los sistemas	4 días	mar 15/02/22	vie 18/02/22	
Gestión del material	5 días	lun 21/02/22	vie 25/02/22	19
Diseño Cloud	**5 días**	**lun 21/02/22**	**vie 25/02/22**	**19**
Diseño BD	5 días	lun 21/02/22	vie 25/02/22	
Diseño Web	5 días	lun 21/02/22	vie 25/02/22	
Diseño Device	5 días	lun 21/02/22	vie 25/02/22	19
Diseño algoritmo personas	**27 días**	**lun 28/02/22**	**mar 05/04/22**	**21**
Normalización de datos	10 días	lun 28/02/22	vie 11/03/22	
Selección algoritmo personas	7 días	lun 14/03/22	mar 22/03/22	26
Entrenamiento del modelo	10 días	mié 23/03/22	mar 05/04/22	27
Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
FASE 5 (Construcción)	**14 días**	**mié 06/04/22**	**lun 25/04/22**	**16**
Construcción Device	**8 días**	**mié 06/04/22**	**vie 15/04/22**	
Montaje dispositivos Device	3 días	mié 06/04/22	vie 08/04/22	
Conexión dispositivos Device	3 días	lun 11/04/22	mié 13/04/22	31
Conexión Device - Edge	2 días	jue 14/04/22	vie 15/04/22	32
Construcción Cloud	**12 días**	**mié 06/04/22**	**jue 21/04/22**	
Configuración servidor	4 días	mié 06/04/22	lun 11/04/22	
Configuración Web	5 días	mar 12/04/22	lun 18/04/22	35
Conexión BD - Web	3 días	mar 19/04/22	jue 21/04/22	36
Implementación algoritmo personas	2 días	mar 12/04/22	mié 13/04/22	35
Conexión Device - Cloud	2 días	vie 22/04/22	lun 25/04/22	34
FASE 6 (Test)	**19 días**	**mar 26/04/22**	**vie 20/05/22**	**29**
Test Device	**7 días**	**mar 26/04/22**	**mié 04/05/22**	
Test Componentes	**4 días**	**mar 26/04/22**	**vie 29/04/22**	
Test placa	3 días	mar 26/04/22	jue 28/04/22	
Test DHT22	4 días	mar 26/04/22	vie 29/04/22	
Test cámara	3 días	mar 26/04/22	jue 28/04/22	
Test conjunto Device	3 días	lun 02/05/22	mié 04/05/22	42
Test Device - Edge	1 día	lun 02/05/22	lun 02/05/22	42
Test Device - Cloud	2 días	lun 02/05/22	mar 03/05/22	42
Test Cloud	**4 días**	**jue 05/05/22**	**mar 10/05/22**	**41**
Test BD	3 días	jue 05/05/22	lun 09/05/22	
Test Web	4 días	jue 05/05/22	mar 10/05/22	
Test algoritmo	4 días	jue 05/05/22	mar 10/05/22	
Test conjunto Cloud	3 días	mié 11/05/22	vie 13/05/22	49
Test conjunto Sistema	5 días	lun 16/05/22	vie 20/05/22	53
FASE 7 (Realización)	**9 días**	**lun 23/05/22**	**jue 02/06/22**	**40**
Redacción documentación final	**9 días**	**lun 23/05/22**	**jue 02/06/22**	
Redacción informe del proyecto	4 días	lun 23/05/22	jue 26/05/22	
Redacción documento costes del proyecto	4 días	vie 27/05/22	mié 01/06/22	57
Redacción manuales de usuario	5 días	vie 27/05/22	jue 02/06/22	57
FASE 8 (Final)	**1 día**	**vie 03/06/22**	**vie 03/06/22**	**55**
Lanzar al mercado	1 día	vie 03/06/22	vie 03/06/22	

Once we have planned the time to perform each task, we can see that the duration of the project is 105 days, from Monday, January 10 to Friday, June 3.

All this work must be done by people, so within the same Microsoft project, we have created the different worker profiles for our project. We have also assigned them their rate of euros per hour of work.

Nombre del recurso	Tasa
Programador	20,00 €/hora
Becario	8,00 €/hora
Director proyecto	35,00 €/hora
Especialista	25,00 €/hora
Técnico	18,00 €/hora

The next step has been to assign to each task the profile or profiles needed to be completed. Once all the tasks have been assigned from the beginning to the end of the project, these are the number of working hours for each profile, with the corresponding price to be paid to them:

Nombre del recurso	Trabajo	Costo
Programador	340 horas	6.800,00 €
Becario	326 horas	2.608,00 €
Director proyecto	152 horas	5.320,00 €
Especialista	208 horas	5.200,00 €
Técnico	358 horas	6.444,00 €

Once all these steps are done, we can create a column in our project to see the total cost of the project, as well as the amount of labor hours needed to complete the project.

Nombre de tarea	Duración
PROYECTO IOT GRUPO 6	**105 días**

Trabajadores				
Perfil	Horas totales	Tasa €/h	Costo	Total post impuestos
Director P.	152	35	5320	7120
Programador	340	20	6800	8600
Especialista	208	25	5200	7000
Técnico	358	18	6444	8244
Becario	326	8	2608	3108
TOTAL				34072

10.4 Selling price

Considering all the values of the previous sections, we will proceed to make a first estimate of the cost of sale of our product.

The unit cost of the system is 196.57€.

As a first approach, we can say that we want to have 240 devices in stock, so the cost of the first purchase of materials would amount to 47,176.8€.

The cost of assembly would be about 1,400€.

The price to be paid to the workers is 34.072€.

The monthly cost of the Cloud is 168€.

The cost of the first year of the project would amount to 84.664,80€.

1º AÑO		
Coste	Unidades	Total
Coste unitario	1 unidad	196,57 €
Coste producción	240 unidades	47.176,80 €
Coste montaje	240 unidades	1.400 €
Coste trabajadores		34.072 €
Coste Cloud	AWS S3 + EC2	168,00 €
Coste total		84.664,80 €

The cost of the following years would consist of paying both the AWS service, which would be about 2.000€ per year, and the purchase of more materials if needed. We can assume that we will not need to buy more materials, with the initial 240 devices should be enough.
Our sales forecast per month would be about 10.

2º AÑO		
Coste	Unidades	Total
Coste Cloud	AWS S3 + EC2	168 €
Coste total		2.016 €

The initial price we had in mind is €379, which would give us a profit of 45.480 € per year.
After 2 years, we would make a profit of 90.960 €, which covers the first year's investment.

3º AÑO		
Coste	Unidades	Total
Coste unitario	1 unidad	196,57 €
Coste producción	120 unidades	23.588,40 €
Coste montaje	120 unidades	1.400 €
Coste Cloud		168,00 €
Coste total		27.004,40 €

With these values, we know that if we continue at the same rate of sales we would be in positive, since in the third year we would obtain profits of 136.440€ while producing the 120 new devices for this year plus the cost of the Cloud service would be 27.000€, which added to the previous costs would add up to a total of 113.685,2€.

Therefore, the initial selling price would be 379€.

Coste - Beneficio		
Datos		Total
Coste 1º año		84.664,80 €
Coste 2º año		2.016 €
Coste 3º año		27.004,40 €
Ventas estimadas	1 año	120
Margen		20%
Precio estimado	1 unidad	379 €
Costes totales		113.685,20 €
Beneficios esperados		136.422 €
Diferencia		22.737,04 €

11. Conclusions

To finalize the project, we are going to check if we have fulfilled all the conditions and specifications we set at the beginning of the project.

First of all, we have kept the basic functionalities of our project, regulating the use of AC and heating by connecting to the domotics of the place, monitoring the capacity of the place and keeping the user informed of the current data.

As for the structure, in the end we decided to remove the Edge and simply use an access point.

On the cabling side, we decided to put fiber optics for the connection between the Router and the Cloud.

When choosing the device components, we took into account the initial specifications. We met the forecasts for power consumption as well as the device's dimensions and weight.

Due to the choice of our board, we have not added the LED to show the board status as the board does not include it.

Finally, our first approach was to connect the device to the mains, although we also considered batteries, but we ended up opting for the mains connection.

We have not met the frequency forecast, but with the Raspberry's 1.4 GHz we have no problem, the bottleneck is still the speed of the SD memory.

As for the price of the device, at first, we thought it would be relatively inexpensive, but as we got deeper into the project, we realized that it could not be that cheap.

At first, we estimated that the selling price would be around 50€, but by adding components and needing more quality, the selling price has risen to 379€.

ESPECIFICACIONES		
Tipo	Previsto	Final
Funcionalidades	· Regular el uso del AC y calefacción · Monitorizar el aforo del lugar · Mantener al usuario informado	Cumplidas
Estructura	· Device · Edge · Cloud	· Device · Cloud
Cableado	Todo wireless	Router - Cloud cable
Consumo	Entre 1.5A - 4.6A y 5W - 10W	5V, 1.3A - 3A y 6,9W
Dimensiones	No mayor de 10x10x5cm	· Placa 9,6x7,11x2,69cm · Cámara 1,15x1.27x0,7cm
Peso	No mayor de 500 g	· Placa 50 g · Cámara 0,9 g
Interfaz	· Device tendrá LED · Front End tendrá Web App	· Front End tiene Web App
Autonomía	Conectado a red eléctrica	Cumplido
Frecuencia	2.4 GHz	No cumplido
Precio	No mayor a 100€	Imposible

References

[1] Controlador de temperature plus con 4 sensores https://www.heiz24.de/Controlador-de-temperatura-diferencial-Resol-DeltaSol-CS-Plus-con-4-sensores-Nuevo?curr=EUR

[2] Interruptor de tempertatura de Ourleeme termostato https://www.amazon.es/OurLeeme-Controlador-temperatura-interruptor-termostato/dp/B07THY1411/ref=asc_df_B07THY1411/

[3] Control de aforo basado en inteligencia https://tsimplifica.com/control-de-acceso/544-control-de-aforo-basado-en-inteligencia-artificial-aforalia-0652217778113.html

[4] Automatizacion de puertas con control de aforo https://tecnoparking.com/tienda/accesorios-de-seguridad/920-automatizacion-y-control-sistema-de-aforo-para-puertas-de-cristal-y-accesos.html

[5] Raspberry PI3 A+ https://www.kubii.es/raspberry-pi-3-2-b/2334-raspberry-pi-3-modelo-a-kubii-652508442181.html

[6] Sensor Temperatura DHT22 https://www.tiendatec.es/arduino/sensores/790-sensor-dht22-temperatura-y-humedad-8472496009812.html

[7] Cámara térmica Teledyne FLIR Lepton 500-0763-01 https://www.mouser.es/ProductDetail/Teledyne-FLIR-Lepton/500-0763-01?qs=%2Fha2pyFadugxXYJdD9ymI-NAg6KNpoh7L%2FyHVWoYF5ytWbd88bYguWQ==

[8] Adaptador de corriente https://www.kubii.es/index.php?controller=order-opc

[9] Caja de protección placa https://www.kubii.es/cajas-y-soportes/2374-caja-para-raspberry-pi-3-modelo-a-kubii-3272496012981.html

[10] Calculadora precios Cloud https://calculator.aws/#/createCalculator/S3?nc2=pr

[11] Tarjeta Micro SD clase 4 https://www.amazon.es/dp/B00195G388?m=A1AT7YVPFBWXBL&tag=wwwidealoes-21&ascsubtag=Ktt0D2FXZ_6X24G2L0kNOw&th=1

[12] Tarjeta Micro SD clase 6 https://www.amazon.es/Cloudisk-Paquete-tarjetas-memoria-unidades/dp/B08L8T86TS/

[13] Camara profesional para medición de temperatura https://www.cdpsa.eu/producto/camara-profesional-portatil-para-medicion-de-temperatura-cdpr200/

APPENDIXES

a.1 Gantt Diagram

Below we can see the complete Gantt Chart of our project. It has a duration of 105 days from Monday, January 10 to Friday, June 3.