

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

Integración de sistemas de confort en ambientes inteligentes: control
térmico basado en reglas e inteligencia artificial

PROYECTO INTEGRADOR

Previo la obtención del Título de:

Ingeniero en Telemática

Ingeniero en Telemática

Presentado por:

Andrés Sebastian Cedeno Vinces

Angelo de Francisco Sánchez Condo

GUAYAQUIL - ECUADOR

Año: 2022

DEDICATORIA

Andrés Cedeño

Sin duda, las personas a las que más les debo quién soy, son mis padres. Una dedicatoria no me alcanza para describir lo agradecido que estoy por ustedes. Esto no hubiera sido posible sin su ayuda. A mi hermana, que siempre me alentó en todo proyecto que tengo. Por último, a todas las personas que se cruzaron en mi camino, ya sea para bien o para mal. Cada una apporto su granito de arena en mi vida universitario y sin todas esas experiencias yo no estaría aquí ni hubiera alcanzado este objetivo.

Angelo Sanchez

Este proyecto se lo dedico principalmente a mi madre que desde el cielo ha sido mi luz durante estos últimos años, a mi padre que me ha apoyado siempre incondicionalmente, mi hermana de igual manera, al resto de mi familia y amigos que siempre estuvieron ahí para apoyarme de una u otra manera y a la persona con la que pienso compartir el resto de mis días y a nuestro hijo/a que sin conocerlo/a aún no hay nada que no haría para darle siempre lo mejor.

AGRADECIMIENTOS

Andrés Cedeño

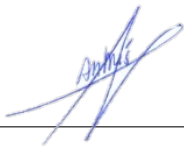
Gracias al PhD José Cordova por la paciencia que tuvo con nosotros y por estar siempre dispuesto a responder nuestras inquietudes y preguntas. Al Laboratorio de Sistemas Telematicos y sus colaboradores por la atención y la buena disposición para ayudar. Al al Ing. Ignacio Marín por sus clases y recomendaciones para la escritura de este documento. Finalmente, gracias a mi compañero Angelo Sanchez, por su dedicación y esfuerzo gigante para la realización y culminación de este proyecto.

Angelo Sanchez

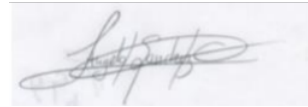
En primer lugar gracias a Dios por permitirme llegar hasta aquí, al PhD José Córdoba por la paciencia y la disposición de su persona hacia nosotros, a todos los profesores que he tenido a lo largo de la carrera estaré eternamente agradecido con cada uno de ustedes y sin duda alguna a mi familia que siempre estuvieron ahí para darle ese apoyo que necesitaba en los momentos más difíciles.

DECLARACIÓN EXPRESA

"Los derechos de titularidad y explotación, nos corresponde conforme al reglamento de propiedad intelectual de la institución; Andrés Sebastián Cedeño Vines y Angelo de Francisco Sánchez Condo damos nuestro consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual"



Ándres Cedeño Vines



Angelo Sanchez Condo

EVALUADORES

Ignacio Marín García
PROFESOR DE LA MATERIA

José Córdova García
PROFESOR TUTOR

RESUMEN

El confort térmico de los ocupantes en espacios cerrados y/o ambientes inteligentes es de vital importancia actualmente debido a que el confort es un aspecto clave en el rendimiento y productividad a nivel laboral o educativa. El presente proyecto tiene como objetivo mejorar el confort de los usuarios mediante el diseño e implementación de un sistema IoT (Internet of Things), el cual consta con una base de datos en tiempo real que permitirá atender las necesidades de los usuarios en términos de confort de manera inmediata en conjunto con dos (2) algoritmos que tienen el mismo objetivo, pero lo abordan de formas diferentes; el primero se basa en una serie de condiciones definidas por nosotros y el segundo utiliza un paradigma de inteligencia artificial conocido como aprendizaje por refuerzo y un aplicativo web que permitirá el monitoreo en tiempo real. La solución abarca varios módulos donde intervienen microcomputadoras y sensores infrarojos para la obtención de data en tiempo real. Por otro lado la comunicación entre los distintos módulos se basó principalmente en los protocolos HTTP (Hypertext Transfer Protocol) y MQTT (Message Queuing Telemetry Transport). El resultado más relevante es que efectivamente el algoritmo de aprendizaje por refuerzo, que es un tipo de inteligencia artificial, es considerablemente mejor para asegurar el confort. Con este resultado pudimos cumplir con el objetivo de construir un sistema que afecta positivamente en las actividades de los usuarios.

Palabras Clave: IoT, Confort Térmico, Aprendizaje por Refuerzo, Plataforma Web, Firebase.

ABSTRACT

The thermal comfort of occupants in enclosed spaces and/or intelligent environments is of vital importance today because comfort is a key aspect in performance and productivity at work or educational level. This project aims to improve the comfort of users through the design and implementation of an IoT (Internet of Things) system, which consists of a real-time database that will allow to meet the needs of users in terms of comfort immediately in conjunction with 2 algorithms that have the same objective, but approach it in different ways; The first one is based on a set of conditions defined by us and the second one uses an artificial intelligence paradigm known as reinforcement learning and a web application that will allow real-time monitoring. The solution comprises several modules involving microcomputers and infrared sensors to obtain data in real time. On the other hand, the communication between the different modules is mainly based on HTTP and MQTT protocols. The most relevant result is that indeed the reinforcement learning algorithm, which is a type of artificial intelligence, is considerably better at ensuring comfort. With this result we were able to fulfil the goal of building a system that positively affects the users' activities.

Keywords: IoT, Thermal Comfort, Reinforcement Learning, Web Platform, Firebase

ÍNDICE GENERAL

RESUMEN	i
ABSTRACT	iii
ABREVIATURAS	ix
SIMBOLOGÍA	xi
ÍNDICE DE FIGURAS	xi
ÍNDICE DE TABLAS	xiii
ÍNDICE DE CODIGOS DE PROGRAMA	xv
1 INTRODUCCIÓN	1
1.1 Descripción del problema	2
1.2 Justificación del problema	3
1.3 Objetivos	3
1.3.1 Objetivo General	3
1.3.2 Objetivos Especifico	3
1.4 Marco Teórico	4
1.4.1 Componentes de hardware	4
1.4.2 Componentes de software	5
1.5 Descripción de escenarios para la propuesta	6
2 METODOLOGÍA	9
2.1 Módulo administrador pesonalizado de confort	10
2.1.1 Frontend	10
2.1.2 Backend	12
2.2 Modulo de almacenamiento	13

2.2.1	Firestore	13
2.2.2	Realtime Database	14
2.3	Modos de operación del sistema de Confort en ambientes inteligentes . . .	15
2.3.1	Módulo de control térmico basado en reglas	15
2.3.2	Módulo de control térmico basado en Inteligencia Artificial	17
2.3.3	Módulo de control inteligente de aire acondicionado	18
3	PRUEBAS Y RESULTADOS	23
3.1	Escenario de pruebas	23
3.2	Control térmico basado en reglas	24
3.2.1	Prueba 1: Sistema basado en recomendaciones al usuario	24
3.2.2	Prueba 2: Sistema basado en votaciones del usuario	25
3.2.3	Análisis de resultados del control térmico basado en reglas	28
3.3	Control térmico basado en aprendizaje por refuerzo	30
3.3.1	Prueba 3: Implementación de Value Iteration	30
3.3.2	Análisis de resultados del control térmico basado en aprendizaje por refuerzo (value iteration)	33
3.4	Costos del proyecto	33
3.4.1	Costos de Firebase	34
3.4.2	Valor unitario del producto	36
3.5	Análisis Económico	37
3.5.1	Costos fijos	37
3.5.2	Costos Variables	38
3.6	Punto de equilibrio	38
3.7	Modelo de Negocio	39
3.8	Rentabilidad	40
4	CONCLUSIONES, RECOMENDACIONES Y LÍNEAS FUTURAS	43
4.1	Conclusiones	43
4.2	Recomendaciones	44
4.3	Líneas Futuras	44
	BIBLIOGRAFÍA	47

APÉNDICES	51
A Código de control térmico basado en reglas	53
B Código de control basado en aprendizaje por refuerzo	58
C Código de control de dispositivo Broadlink	63
D Encuentas acerca del funcionamiento del control térmico basado en value iteration	68
E Matriz de recomendaciones basada en reglas	70
F Matriz de recomendaciones basada en Aprendizaje por Refuerzo	71
G Proyección trimestral del flujo de caja para 2 años	72

ABREVIATURAS

ESPOL	Escuela Superior Politécnica del Litoral
OCDE	Organización para la Cooperación y el Desarrollo Económicos
IA	Inteligencia Artificial
UI	User Interface
HTTP	Hypertext Transfer Protocol
MQTT	Message Queuing Telemetry Transport
ASHRAE	American Society of Heating, Refrigerating and Air-Conditioning Engineers
HVAC	Calefacción, Ventilación y Aire Acondicionado
OCC	Control Centrado en el Ocupante
RL	Reinforcement Learning
LST	Laboratorio de Sistemas Telemáticos

SIMBOLOGÍA

mil	Milésima de pulgada
mg	Miligramo
pH	Potencial de Hidrógeno
m	Metro
CO ₂	Dióxido de carbono

ÍNDICE DE FIGURAS

2.1	Arquitectura de Sistema IoT de recomendaciones de confort para ambientes inteligentes	10
2.2	Diagrama de flujo del modo basado en reglas	17
2.3	Diagrama de flujo del modo de uso basado en aprendizaje por refuerzo . .	19
2.4	Mando infrarojo a distancia Broadlink RM4 Mini	20
3.1	Croquis del Laboratorio de Sistemas Telemáticos	24
3.2	Elección del confort según el usuario	25
3.3	Conteo de recomendaciones dadas al usuario	26
3.4	Conteo de votos durante el 15 de diciembre del 2022	27
3.5	Elección de temperatura del sistema por medio del sistema de votos durante el 15 de diciembre del 2022	27
3.6	Pregunta 1 del formulario de Google	28
3.7	Pregunta 2 del formulario de Google	29
3.8	Pregunta 3 del formulario de Google	29
3.9	Pregunta 4 del formulario de Google	30
3.10	Convergencia de un estados para distintos valores de Gamma	31
3.11	Número de visitas a los distintos estados durante un día completo	32
3.12	Comportamiento de estados en un día determinado	32
3.13	Punto de Equilibrio	39
1	Pregunta 1 de la encuesta	68
2	Pregunta 2 de la encuesta	68
3	Pregunta 3 de la encuesta	69
4	Pregunta 4 de la encuesta	69
5	Matriz de recomendaciones	70
6	Matriz de recomendaciones basada en aprendizaje por refuerzo	71
7	Flujo de caja	72

ÍNDICE DE TABLAS

2.1	Variables consideradas en el módulo basado en reglas	16
2.2	Estados del aire acondicionado	18
2.3	Periodos de tiempo	18
2.4	Especificaciones técnicas de Broadlink RM 4 mini	21
3.1	Costos de Hardware	34
3.2	Costos de desarrollo del software	34
3.3	Valores de lectura/escritura en Realtime Database	35
3.4	Precios de Firestore	35
3.5	Costo en el peor caso de Firebase	36
3.6	Precio de venta unitario	37
3.7	Costos fijos mensuales	37
3.8	Costos fijos mensuales	38
3.9	Valor de VAN y TIR	41

ÍNDICE DE CODIGOS DE PROGRAMA

2.1	Esquema de base de datos en tiempo real	14
1	Código de control térmico basado en reglas	53
2	Código de control térmico basado en value iteration	58
3	Código para realizar control del sensor Broadlink	63

CAPÍTULO 1

1. INTRODUCCIÓN

El confort térmico según el estándar ASHRAE 55 se define como la expresión de satisfacción en la mente del ser humano con el ambiente térmico [1], considerándose como una sensación psicológica de una persona que varía de acuerdo a las personas. Comprende un análisis de diversas variables que se encuentran relacionadas con el ambiente y los usuarios. A pesar de que el confort térmico tiene una relación muy particular con la temperatura también se debe tomar en cuenta otras variables como la humedad relativa, la calidad del aire y el nivel de Dióxido de carbono (CO₂) [2]. Por medio del confort térmico se busca conocer el nivel de aceptación térmica de los usuarios dentro de sus lugares de oficio y a su vez como mejorar ese nivel.

En ambientes cerrados como salones de clases, oficinas, laboratorios, etc, es muy importante brindar el confort térmico adecuado proporcionando entornos de vida y de trabajo de alta calidad [3]. Por el contrario, la falta de confort térmico afecta de forma directa en la productividad y la eficiencia de los adultos en sus trabajos, y en el nivel de concentración y aprendizaje de los estudiantes [4, 5]. Por esto, se debe tener en cuenta que la temperatura ambiente puede alterarse dramáticamente provocando una fluctuación en la temperatura de un espacio cerrado. Según [6] los trabajadores de oficinas prefieren temperaturas entre los 20°C a 24°C para tener un ambiente de trabajo fresco y con ello tener un óptimo rendimiento en sus actividades.

Con el objetivo de lograr el confort térmico se implementaron los sistemas de calefacción, ventilación y aire acondicionado (HVAC) como la principal forma de poder controlar el ambiente térmico en interiores. Sin embargo, a pesar de ser considerada una solución también puede generar preocupaciones como en el consumo de energía que en edificios oscila entre 20% y 40% del consumo total de un edificio [7], el ajuste inadecuado de la temperatura del sistema HVAC provocando una incomodidad a los usuarios con

temperaturas que pueden generar o mucho frío o mucho calor. Por lo tanto, es necesario tener en cuenta que el desafío presente es alcanzar el confort térmico óptimo en interiores sin comprometer el consumo de energía. Por esta razón la solución propuesta consiste en un sistema integrado para confort que tiene como finalidad dar al usuario varios modos de uso de control térmico dentro de un espacio cerrado.

La solución integra tres modos de uso. El modo manual o tradicional donde el usuario decide a que temperatura estar de acuerdo a su percepción. Un modo basado en reglas que funciona mediante dos parámetros de entrada (temperatura y confort) ofreciendo recomendaciones al usuario basado en hábitos y métricas de estado ambiental modificando la temperatura del aire acondicionado solo si el usuario acepta la sugerencia. Por ultimo un modo basado en inteligencia artificial mediante la utilización de un algoritmo de aprendizaje por refuerzo. Todos los modos de uso mencionados deben ser visualizados por el usuario mediante un aplicativo híbrido (web y móvil) para el control de los aires acondicionados dependiendo del modo de uso empleado. Finalmente, la acción que se haga sobre los aires acondicionados se realizará mediante sensores infrarojos según la decisión del usuario.

1.1 Descripción del problema

El confort dentro de un espacio cerrado es un tema muy importante debido a que ofrece a los usuarios el bienestar necesario evitando el impacto negativo en el rendimiento de ellos ya sea en un contexto laboral o académico.[8] Hoy en día la tecnología ofrece la posibilidad de solventar esto por medio de la implementación de ambientes inteligentes mediante un control térmico basado en reglas, el uso de la inteligencia artificial, mediante consumo de corriente eléctrica, entre otros.

A medida que se diseñan más opciones para solventar el confort surge la necesidad de implementar un sistema completo que permita integrar cada una de las soluciones propuestas. Sin embargo, los ambientes inteligentes a pesar de todas las posibilidades que ofrecen presentan un gran problema el cual es la utilización de diversas tecnologías para cada una de las posibles soluciones provocando una gran limitante a nivel de compatibilidad entre ellas que incluso perjudica su proyección a futuro.[9]

1.2 Justificación del problema

El confort en interiores tiene muchas implicaciones en diferentes ámbitos que pueden y deben ser abordados. El más importante: la productividad.

Por ejemplo, según la OCDE (Organización para la Cooperación y el Desarrollo Económicos) [10] en México se invierten en promedio 2.137 horas anuales a trabajar, es decir 43 horas a la semana. Sin embargo, México siendo el país que más horas trabaja, esta debajo del promedio de productividad., como indica el gráfico "PIB por hora trabajada" [11]. Esto es una muestra clara de que más horas trabajadas, no equivalen a mayor productividad.

Otro ejemplo con respecto a la temperatura del ambiente de trabajo es un estudio llevado a cabo por la Universidad de Cornell. El artículo apunta a que: "Los trabajadores con frío no solo cometen más errores, sino que las temperaturas más frías podrían aumentar el costo de mano de obra por hora de un trabajador en un 10 por ciento"[12].

Por último, en el mundo laboral y educativo actual, la mayoría de las actividades cotidianas se realizan en ambientes cerrados. Por todo lo anterior mencionado, se vuelve de altísima relevancia asegurar y maximizar el confort de los ocupantes de forma que su productividad aumente y otros factores no puedan afectarla.

1.3 Objetivos

El objetivo general de este proyecto integrador es:

1.3.1 Objetivo General

Desarrollar un sistema de confort en ambientes inteligentes para un control térmico que integre soluciones basadas en reglas e inteligencia artificial mediante un aplicativo web.

1.3.2 Objetivos Específico

A fin de alcanzar el objetivo general se determinaron tres (3) objetivos específicos. Estos objetivos específicos fueron:

- Diseñar un aplicativo web que facilite al usuario la utilización de los modos de uso para el control térmico en un espacio cerrado como son el control manual, basados en reglas e inteligencia artificial.
- Elaborar un sistema de asistencia inteligente mediante sensores infrarojos para el control de aires acondicionados dependiendo el modo de uso que decida el usuario.
- Realizar un análisis de la eficiencia del sistema antes y después de su implementación mediante la utilización de encuestas a los usuarios.

1.4 Marco Teórico

En esta sección se va a presentar y describir cada uno de los componentes de hardware y software necesarios para el desarrollo del proyecto integrador propuesto. Cabe recalcar la importancia de esta sección debido a que facilitará la comprensión acerca del desarrollo y funcionamiento de la presente solución.

1.4.1 Componentes de hardware

Shelly EM¹ es un medidor de energía operado por Wi-Fi con control de contactor. Permite monitorear el consumo de cualquier electrodoméstico, circuito eléctrico y equipo de oficina (luces, líneas eléctricas, sistemas de seguridad, calefacción y refrigeración, etc.) de forma individual, a través de cualquier de sus dos entradas de mediciones independientes de hasta 120 A. Además de la medición precisa de energía, Shelly EM permite el control por contacto de electrodomésticos de de bajo consumo y carga pesada. La combinación entre monitoreo y control del consumo facilita maximizar el ahorro. Este sensor nos permitirá obtener el consumo de los aires acondicionados del ambiente que se esté controlando. Con esta información se podrán tomar decisiones para poder elegir la temperatura que de mejor confort con respecto al consumo de energía.

Broadlink RM4C Mini² permite controlar dispositivos que posean comunicación

¹<https://shelly-api-docs.shelly.cloud/gen1/#shelly-em>

²<https://www.broadlink.ae/product-page/broadlink-rm4-mini>

infrarroja en cualquier momento desde cualquier lugar. Se lo conecta a internet para ser controlado con un dispositivo inteligente. Junto con Alexa o Google Assistant, brinda también control de voz a manos libres de sus aparatos conectados a él. Este dispositivo es base para nuestro proyecto y realmente útil, ya que un programa no puede ni sabe usar el control remoto del aire y, por tanto, necesita de otro medio para poder interactuar con él. Su uso principal será el control de los aires acondicionados a través de nuestros scripts.

El **Raspberry Pi** es una versión más pequeña de una computadora moderna capaz de realizar tareas de manera efectiva. Cuenta con el sistema operativo Raspbian la cual es una distribución de Linux[13]. El módulo utiliza varios tipos de procesadores; por lo tanto, solo puede instalar aplicaciones y sistemas operativos de código abierto, permitiendo al usuario utilizarlo para aplicaciones que se ejecutarían en un computador normal[14]. Raspberry Pi admite varios lenguajes de programación. En nuestro caso el que más nos interesa es Python, ya que con él corremos algunos scripts para la comunicación con los sensores.

1.4.2 Componentes de software

Desarrollamos una aplicación móvil con el framework de Javascript React Native³ que integre todas las funcionalidades esperadas de nuestro proyecto. Este framework está basado en la librería de JavaScript React para la creación de componentes visuales, reutilizándolas para correr sobre sistemas operativos móviles. Es decir, su uso está destinado para la creación de aplicaciones reales nativas para iOS y Android, indistinguibles de lo que se podría desarrollar con código en Objective-C o Java.

La librería principal para aplicar RL es pyqlearning⁴. Esta es una biblioteca de Python para implementar Reinforcement Learning Deep Reinforcement Learning, Q-Learning, Deep Q-Network y Multi-agent Deep Q-Network. Permite diseñar el algoritmo de búsqueda de información. Sin embargo, esta biblioteca está hecha específicamente

³<https://www2.deloitte.com/es/es/pages/technology/articles/que-es-react-native.html>

⁴<https://pypi.org/project/pyqlearning/>

para diseñadores, no para usuarios finales. La filosofía de los creadores es enseñar a diseñar los algoritmos, más no dar la solución en concreto.

Broadlink⁵ es una librería de Python que hace las veces de API (Application Programming Interface) y sirve para controlar dispositivos Broadlink que se encuentren en la red local. Este módulo será como nuestra aplicación hablará con el RM4C Mini para controlar remotamente el aire acondicionado. Mediante esta comunicación obtendremos los códigos binarios que corresponden a ciertas acciones de los aires como prender, apagar, subir la temperatura, entre otras.

Firebase⁶ es una plataforma de desarrollo de aplicaciones que ayuda a crear y hacer crecer aplicaciones. Brinda servicios como compilación, autenticación, alojamiento, almacenamiento en la nube, bases de datos en tiempo real, análisis y mucho más, lo que encaja perfectamente con nuestra implementación. También es importante tener en cuenta que, con un uso ligero, Firebase es completamente gratuito, lo que es una gran ventaja.

Una base de datos es un grupo de datos estructurados que pertenecen a un mismo contexto. **NoSQL** (Not Only SQL) plantea una amplia clase de sistemas de bases de datos que se desmarcan del modelo clásico (Relacional). Entonces las bases de datos NoSQL⁷ utilizan una variedad de modelos de datos para acceder y administrar datos. Estos modelos están optimizados para aplicaciones modernas que requieren grandes volúmenes de datos, baja latencia y modelos de datos flexibles. Firebase provee una base de datos NoSQL que usaremos para almacenar los datos de nuestra aplicación.

1.5 Descripción de escenarios para la propuesta

El proyecto en cuestión está orientado a brindar un confort óptimo a los usuarios mediante un sistema que permite la utilización varios modos de uso para el control térmico en

⁵<https://github.com/mjg59/python-broadlink>

⁶<https://firebase.google.com/?hl=es>

⁷<https://aws.amazon.com/es/nosql/>

edificios inteligentes. Dado a su variedad de modos de uso esta solución es adaptable a diversas áreas como la salud, la industria, la educación, entre otras. Permitiendo a los usuarios un ambiente más agradable y una mayor efectividad en la realización de sus actividades. De manera particular la solución mencionada tiene como escenario principal edificios de un campus universitario, en un ambiente de laboratorios informáticos, donde un control térmico adecuado es indispensable debido a la interacción de equipos pueden afectar en la temperatura interna del sitio y por ende una menor satisfacción del usuario con el confort térmico que provee el lugar.

CAPÍTULO 2

2. METODOLOGÍA

De manera general la solución propuesta presenta tres (3) modos de uso para el control térmico y que a su vez se encargará de ofrecer a los usuarios el confort térmico óptimo necesario para que los usuarios se sientan más productivos y optimistas a la hora de realizar sus actividades. Los modos de uso y arquitectura general se muestra en la figura tal.

El modo manual es el control total del usuario a la hora de elegir la temperatura a la cual se siente cómodo. Por otro lado, el control basado en reglas se compone de un módulo principal, es decir, el módulo de recomendaciones que brindará las recomendaciones de temperatura de acuerdo a métricas que se encontrarán almacenadas en una base de datos en Firebase para su posterior consumo. Las recomendaciones dadas por el usuario en este modo de uso serán visualizadas en un aplicativo móvil que será el frontend de la solución propuesta donde brindará al usuario la opción de tomar la sugerencia para mejorar su confort térmico o no. El último modo de uso, es decir, el control térmico mediante inteligencia Artificial, donde definimos a este modo de uso como un control centrado en el ocupante (OCC) para el confort térmico basado en Reinforcement Learning (RL) [15].

Por último se presenta el módulo de control de aire acondicionado el cual trabaja mediante un sensor infrarojo conectado a un servidor alojado en una Raspberry Pi 3 que mediante internet recibe la información del aplicativo móvil para la posterior acción dependiendo de lo que decida realizar el usuario final.

En la figura 2.1 se muestra a detalle cada uno de los módulos mencionados, que interactúan en la solución propuesta.

En esencia, el sistema funciona de la siguiente manera: la forma en la cual los usuarios participan en el sistema es a través de una UI en la cual ellos pueden votar por el estado

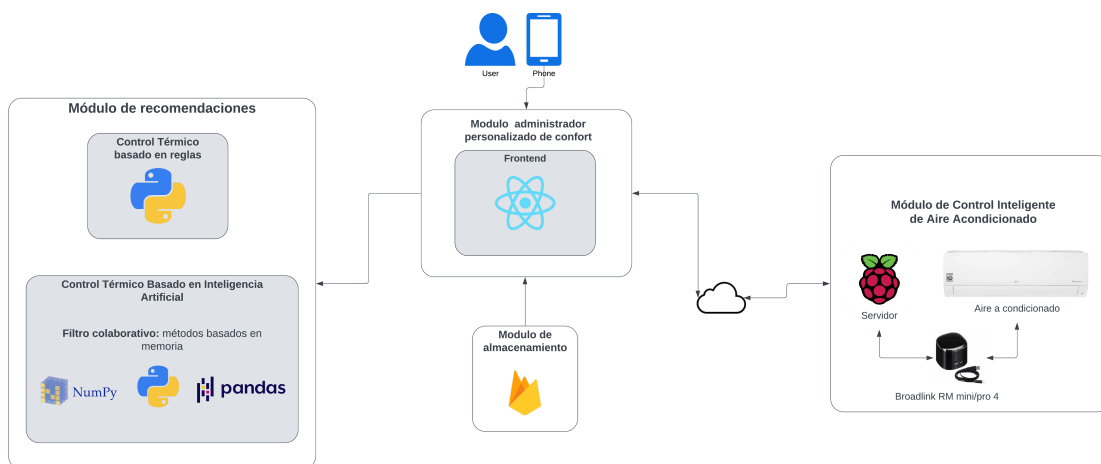


Figura 2.1: Arquitectura de Sistema IoT de recomendaciones de confort para ambientes inteligentes

de confort en el cual se encuentren. Esto se almacena en Firebase, nuestro módulo de almacenamiento. Alimentado de este input y de otras variables ambientales, nuestro modulo de recomendación analiza la mejor acción a tomar, y a través del módulo de control del aire, la ejecuta.

2.1 Módulo administrador personalizado de confort

El módulo de administración personalizado es una de las partes más importantes del proyecto debido a que permitirá al usuario interactuar de manera directa con la solución propuesta. Actualmente existen varias formas de lograr la interacción con el consumidor, particularmente se implementa un aplicativo web que le permitirá al usuario poder asignar el modo de operación de control térmico a cada instancia de aire acondicionado existente en la habitación y poder observar la temperatura en tiempo real y controlar el aire acondicionado desde su ordenador dependiendo del modo de uso asignado.

2.1.1 Frontend

En el desarrollo móvil existen dos modelos a seguir muy importantes como son las aplicaciones nativas y multiplataformas¹. Particularmente este proyecto se enfoca en el desarrollo multiplataforma ya que la idea es abarcar el mayor número de dispositivos

¹<https://www.netsolutions.com/insights/cross-platform-app-frameworks-in-2019/>

finales y no solo centrarnos en abarcar un público en particular. El desarrollo móvil multiplataforma sigue evolucionando día a día en el cual existen varios frameworks que facilitan su implementación brindando beneficios como alto rendimiento, bajo costos en desarrollo, código reutilizable para las diferentes plataformas móviles existentes². El tipo de framework a elegir depende de las características del producto final a implementar. Por ello se analizaron diferentes librerías para el desarrollo móvil multiplataforma de las cuales se priorizaron Flutter y React Native.

Las librerías, Flutter y React Native, han mostrado un crecimiento enorme en los últimos años convirtiéndose en los frameworks multiplataformas más famosos. Por un lado, React Native, es un framework open source diseñado por Facebook que trabaja con lenguaje javascript para el desarrollo de aplicaciones móviles, ofreciendo la reusabilidad de código para adaptarse a los distintos sistemas operativos como son Android y iOS. Flutter³ en cambio fue diseñado por Google y utiliza lenguaje Dart, al igual que React Native ofrece la reusabilidad de código para los distintos sistemas operativos, mayor rendimiento, menor tiempo de implementación, etc. A pesar de que Flutter está obteniendo mayor popularidad día a día en el mercado se tomó la decisión de implementar React Native para el desarrollo de la aplicación multiplataforma debido a factores como el uso del lenguaje Javascript, el cual es mayormente utilizado en el mercado tecnológico y a la amplia documentación y soporte por parte de la comunidad que este posee ⁴.

Luego de desarrollar la aplicación móvil y hacer las pruebas en el LST, los administradores y usuarios del laboratorio nos hicieron notar algunos inconvenientes con ella. El más importante fue la dificultad de distribución de la aplicación compilada a los usuarios. Esta fue la razón por la cual decidimos migrar nuestra UI una página web. De esta forma, basta con levantar un pequeño servidor web en una computadora del LST, para que se pueda acceder a la página en cualquier computador e incluso dispositivo móvil. Resolviendo de forma satisfactoria la limitación mencionada. Elegimos construir la nueva página con React, ya que React Native nace a partir de esta librería, es decir, comparten la misma sintaxis (Javascript) y lógica. Esto facilitó enormemente la migración y nos permitió cambiar de plataforma en poco tiempo.

²<https://www.bacancytechnology.com/blog/mobile-app-development-frameworks>

³<https://flutter.dev/>

⁴<https://reactnative.dev/>

2.1.2 Backend

En informática, específicamente el diseño de software, backend⁵ se refiere a la capa de acceso a datos, la parte del software que procesa la entrada del usuario que viene del frontend [16]. Esta esta no es accesible para los usuarios y esa es la razón de su nombre. El desarrollo de un backend robusto y que se pueda ajustar de buena forma a lo que necesitamos es bastante complicado y demorado. Es por esta razón que decidimos usar el servicio de Google Firebase. Este es plataforma BaaS (Backend as a Service) que viene ya levantado y se integra fácilmente a las necesidades que tenemos como guardar la métrica de los sensores para procesarlos y obtener las recomendaciones de confort. Nos interesa en 3 módulos de este servicio:

- **Authentication**⁶: Firebase Authentication proporciona servicios para autenticar a los usuarios en las aplicaciones. Admite la autenticación mediante contraseñas, números de teléfono, Google, Facebook y Twitter, entre otros más. Este es una funcionalidad importante ya que se necesitan conocer la identidad de un usuario, para guardar de forma segura los datos del usuario y brindar una experiencia de confort personalizada.[17]
- **Cloud Firestore**⁷: Base de datos NoSQL que ofrece Firebase. Flexible y escalable para el desarrollo móvil y web. Se necesita mantener sincronización de los datos entre las aplicaciones de los clientes y un espacio confiable para poder guardar las mediciones de los sensores que son parte de nuestro proyecto. Firestore cubre ambas necesidades de manera satisfactoria. Además, tiene integración con el servicio de Autenticación, lo que nos va a permitir tomar los datos de confort, mediciones y el perfil de la persona para realizar recomendaciones y cambios más personalizados a la temperatura de la habitación.
- **Realtime Database**⁸: Es una base de datos no relacional en formato JSON que permite el almacenamiento de datos y actualización en tiempo real para todos los usuarios. Esta base de datos será ideal para magnitudes como la temperatura actual de la habitación, consumo energético, nivel de confort, entre otros.

⁵<https://platzi.com/blog/que-es-frontend-y-backend/>

⁶<https://firebase.google.com/docs/auth?hl=en>

⁷<https://firebase.google.com/docs/firestore?hl=en>

⁸<https://firebase.google.com/docs/database?hl=en>

2.2 Modulo de almacenamiento

Tener un lugar en el que se guarden los datos para posteriormente acceder a ellos es fundamental en cualquier sistema o proyecto que combine hardware y software. El diseño y clase del módulo de almacenamiento depende de la naturaleza del proyecto y sus requerimientos. En la línea del tema de los módulos de almacenamiento, es importante también mencionar que las Ciencias de la Computación describe tres pilares claves para su desarrollo: la consistencia, disponibilidad y la tolerancia a la partición. Un sistema perfecto aseguraría estos tres rasgos, sin embargo, esto no puede suceder según el teorema CAP, el cual enuncia que: Es imposible garantizar que estas tres características existan simultáneamente en el almacenamiento de datos de un sistema distribuido.

En las bases de datos, la consistencia se alcanza cuando se devuelve el valor del último registro que se realizó mientras se realizan consultas a varios servidores, es decir, todos los cambios de datos deben reflejarse en los nodos restantes. La tolerancia de partición requiere que la información sea accesible incluso cuando algunos nodos del sistema no están disponibles. Por último, la disponibilidad se logra cuando el acceso a la información siempre devuelve una respuesta dentro de un período de tiempo razonable [cita]. Para nuestro proyecto es de alta importancia que la información este siempre accesible para poder alimentar nuestros algoritmos y que las mediciones dadas por los sensores sean consistentes a lo largo de todos los usuarios. También se requiere de una disponibilidad aceptable[18].

Con todo lo mencionado, se justifica de manera satisfactoria el uso de Firebase como módulo de almacenamiento, ya que sus dos productos Firestore Database y Realtime Database son considerados sistemas CP, lo cual se acopla a nuestros requerimientos. También ofrece ventajas como una disponibilidad excelente y una integración con aplicaciones web o móvil eficaz. Por otro lado, también se tienen desventajas como el almacenamiento en sí no nos pertenece y el uso más avanzado de Firebase es pagado. Aun así, consideramos que para un prototipo es ideal.

2.2.1 Firestore

En Firestore es donde vamos a persistir toda la información importante de nuestro proyecto. Será nuestra base de datos principal. Su mayor su mayor ventaja es que

permite recuperar datos realizando queries sencillos para filtrar los documentos, de acuerdo con ciertos parámetros. La base de datos almacenado 2 tipos de objetos: colecciones y documentos.[19] Las colecciones son como el directorio que almacena varios documentos, y estos utilizan el sistema clave-valor para guardar la información. Por ahora, planeamos tener 2 colecciones una para guardar las recomendaciones que se hacen para el confort otro para almacenar las mediciones de los sensores cada cierto tiempo.

2.2.2 Realtime Database

A diferencia de Firestore en Realtime Database los valores se sincronizan en tiempo real, por lo que esta base de datos no es ideal para persistir la información sino para tener el valor exacto de una variable al momento para que otros partes de nuestro proyecto puedan consumirla e incluso modificarla.[20] La importancia de la sincronización en tiempo real es que el sistema de control está constantemente escuchando a actualizaciones en el valor de la recomendación, para con este número poder decidir qué acción tomar, como encender o subir la temperatura. Estos comandos se realizan con el Broadlink a través de comunicación infrarroja. La base de datos en tiempo real esta estructurada en formato JSON. El Código 2.1 muestra la estructura de datos que tendrá nuestro proyecto:

Código 2.1: Esquema de base de datos en tiempo real

```
1 {'https://energy-coach-f5270-default-rtdb.firebaseio.com/':  
2   {'comfort':  
3     {'cold':0  
4       'neutral':0  
5       'warm':0  
6     'recomendation':  
7       {'rec': 1 }  
8   }  
9 }
```

2.3 Modos de operación del sistema de Confort en ambientes inteligentes

De manera general el sistema se basa en un sistema de recomendaciones personalizado, estos sistemas brindan al usuario una ayuda en la predicción de decisiones a futuro utilizando determinados métodos para ofrecer la mejor recomendación. Los métodos más utilizados son aquellos basados en contenido y métodos basados en filtros colaborativos. Los métodos basados en contenido permiten al usuario elegir opciones de recomendación que están centradas en las elecciones que previamente han escogido. En cambio los métodos que se basan en filtros colaborativos recomiendan distintas opciones que fueron escogidas previamente por usuarios análogos al usuario objetivo [21]. Debido a la facilidad de poder adquirir información de lugares similares con la finalidad de obtener la mejor recomendación para el usuario objetivo se tomó la decisión de enfocarnos en el método basado en filtros colaborativos.

El método basado en filtros colaborativos para recomendaciones presenta técnicas que se dividen en métodos basados en modelos y basados en memoria. El método basado en modelos se encarga de enseñar al módulo a partir de los datos de entrenamiento, es decir, de las preferencias que los usuarios escogieron previamente y con ello predecir recomendaciones diferentes a futuro, por otro lado, la técnica basada en memoria fue implementada en el sistema de filtros colaborativos la cual se encarga de usar cada una de las opciones elegidas previamente por los usuarios similares y con ello generar una predicción[22]. Con esta implementación el filtro colaborativo aprende de forma interactiva debido a la utilización y el análisis que se realizan a los datos y por ende las decisiones tomadas por los usuarios análogos en el pasado produciendo la mejor recomendación posible.

2.3.1 Módulo de control térmico basado en reglas

El módulo de control térmico basado en reglas es el primero de dos modos de uso que presenta el sistema completo. Este módulo es desarrollado mediante una estructura de programación que funciona por medio de ciertas reglas en donde se toma decisiones considerando variables ambientales, tiempo y estado del aire acondicionado como se

puede visualizar en la tabla 2.1. Se diseñó una matriz de recomendaciones que consta principalmente de estados y acciones. Los estados son cada uno de los condicionales establecidos previamente para los distintos valores que pueden tomar las variables mientras que las acciones se definen como las recomendaciones que puede recibir el usuario para mejorar su confort realizando un control térmico donde se considera: Encender el AC, apagar el AC, subir la temperatura del AC, bajar la temperatura del AC y mantener el estado actual del aire acondicionado.

Variables	Nomenclaturas
tiempo	t
Confort del usuario	Confort
Estado del AC	AC_state
Temperatura exterior	t_out

Tabla 2.1: Variables consideradas en el módulo basado en reglas

El Anexo E muestra cada una de las reglas que dependen de las variables mencionadas anteriormente donde se resalta que el tiempo debe pertenecer a cualquiera de los cuatro rangos establecidos en la matriz y a su vez analizando si la temperatura del interior es mayor a la del ambiente. El análisis de cada una de estas variables de acuerdo a las reglas que deben seguir va a sugerir bajar o subir la temperatura según sea el caso con el objetivo de mejorar el confort del usuario, adicionalmente se estudia el confort del usuario la cual representa la variable más importante porque al encontrarse con niveles de frío extremo o con calor excesivo el usuario estaría en un punto de comodidad inaceptable. También es importante tener en cuenta evitar dar recomendaciones del mismo estado, es decir, que se recomiende apagar el aire acondicionado cuando este ya se encuentra apagado, este tipo de recomendaciones se van a evitar con el uso de la variable estado del aire ya que de otra manera al ocurrir ese tipo de casos sería algo muy ineficiente.

El flujo del control térmico basado en reglas se realizará de acuerdo con la figura 2.3. En esta se observa como primer paso la recopilación de los valores de las variables descritas en la tabla tal donde posteriormente se determina el estado y la acción a realizar siempre y cuando el usuario decida aceptar la recomendación. En caso de que el usuario acepte la recomendación envía la acción a realizar al dispositivo Broadlink. Luego el sistema esperará una hora y repetirá el proceso. En caso de rechazar la recomendación

la acción a realizar es la de mantener el estado actual y al igual que en el caso afirmativo se procede a esperar 15 minutos para reiniciar el proceso de toma de datos.

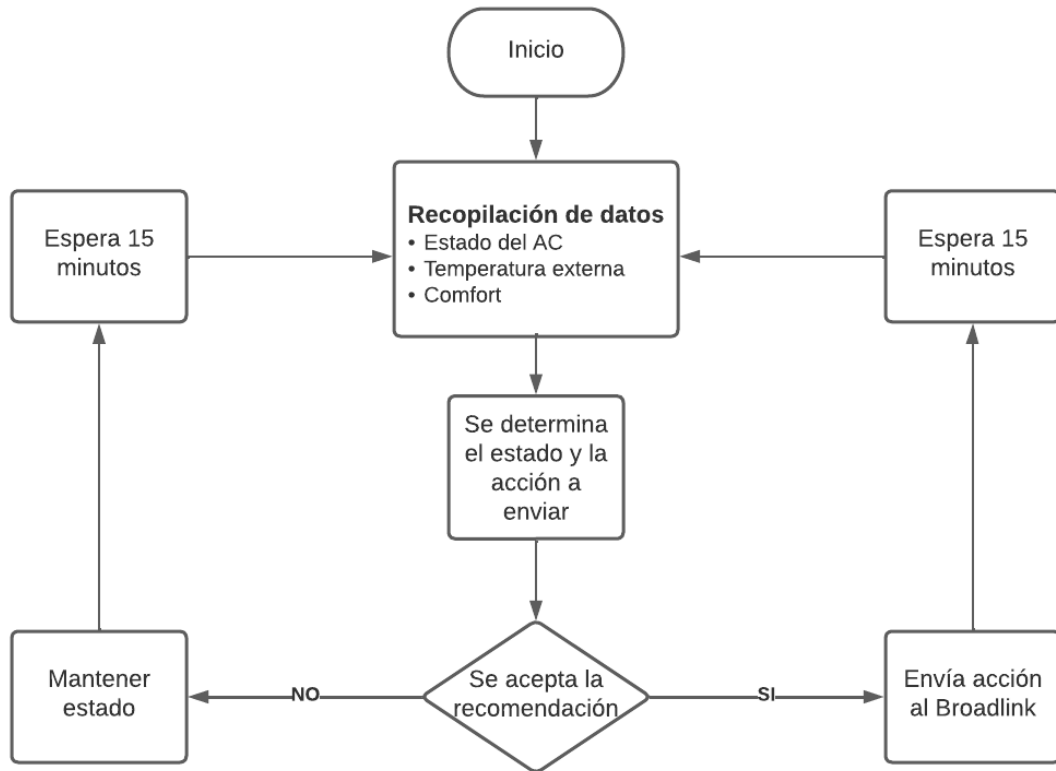


Figura 2.2: Diagrama de flujo del modo basado en reglas

2.3.2 Módulo de control térmico basado en Inteligencia Artificial

Este segundo y último modo de uso implementa aprendizaje por refuerzo con la finalidad de determinar la acción ideal que se deba sugerir al usuario dependiendo del estado en el que se encuentre. En este modo de uso se toma en cuenta cuatro variables: ocupancia, estado del aire, confort del usuario y el periodo de tiempo. Adicionalmente se presenta un valor de recompensa el cual tiene como tarea calificar la viabilidad de encontrarse en un estado. El Anexo F muestra los estados y las recompensas respectivas. La recompensa se divide en tres valores. El valor de -1 representa que el estado asociado a esa recompensa significa incomodidad en el usuario. EL valor de 0 hace referencia a que el usuario se encuentra cómodo, pero que podría mejorar. Mientras que el valor de +1 representa la comodidad óptima requerida por el usuario.

Con respecto al estado del aire acondicionado, este podrá tomar cuatro opciones, las

cuales son: Apagado, T1 , T2 y T3. La Tabla 2.2 muestra la definición de estos rangos. Por otra parte, los periodos de tiempo se dividieron en cuatro y están definidos como se indica en la Tabla 2.3.

Las acciones se aplicaban de acuerdo al diagrama de flujo en la Figura 2.5. El algoritmo empieza recopilando los datos en tiempo real de tiempo, estado del aire acondicionado, confort y ocupancia. Luego obtiene el estado actual de acuerdo a los datos recopilados. En caso de que el estado actual sea igual que el estado anterior, entonces se espera un minuto y se vuelven a recopilar datos. En caso de tener distintos estados, se determina la mejor acción, para luego determinar si el estado actual será distinto al siguiente estado considerando la mejor acción. En caso que sean iguales, se espera un minuto y vuelven a recopilar los datos. Por el contrario, si son distintos, se envía la acción al broadlink y luego se esperar un minuto, siguiendo con el flujo desde el inicio.

Variable	Estado
OFF	Apagado
T1	16-18°C
T2	19-21°C
T3	21-24°C

Tabla 2.2: Estados del aire acondicionado

Variable	Periodo
P1	07:00-12:00
P2	12:00-17:00
P3	17:00-23:59

Tabla 2.3: Periodos de tiempo

2.3.3 Módulo de control inteligente de aire acondicionado

El módulo de control inteligente se encarga de cambiar el estado del aire acondicionado de acuerdo a las ordenes dadas por el aplicativo móvil ya sea por acciones directas sobre el estado del AC o por las recomendaciones que el sistema le brinda al usuario.

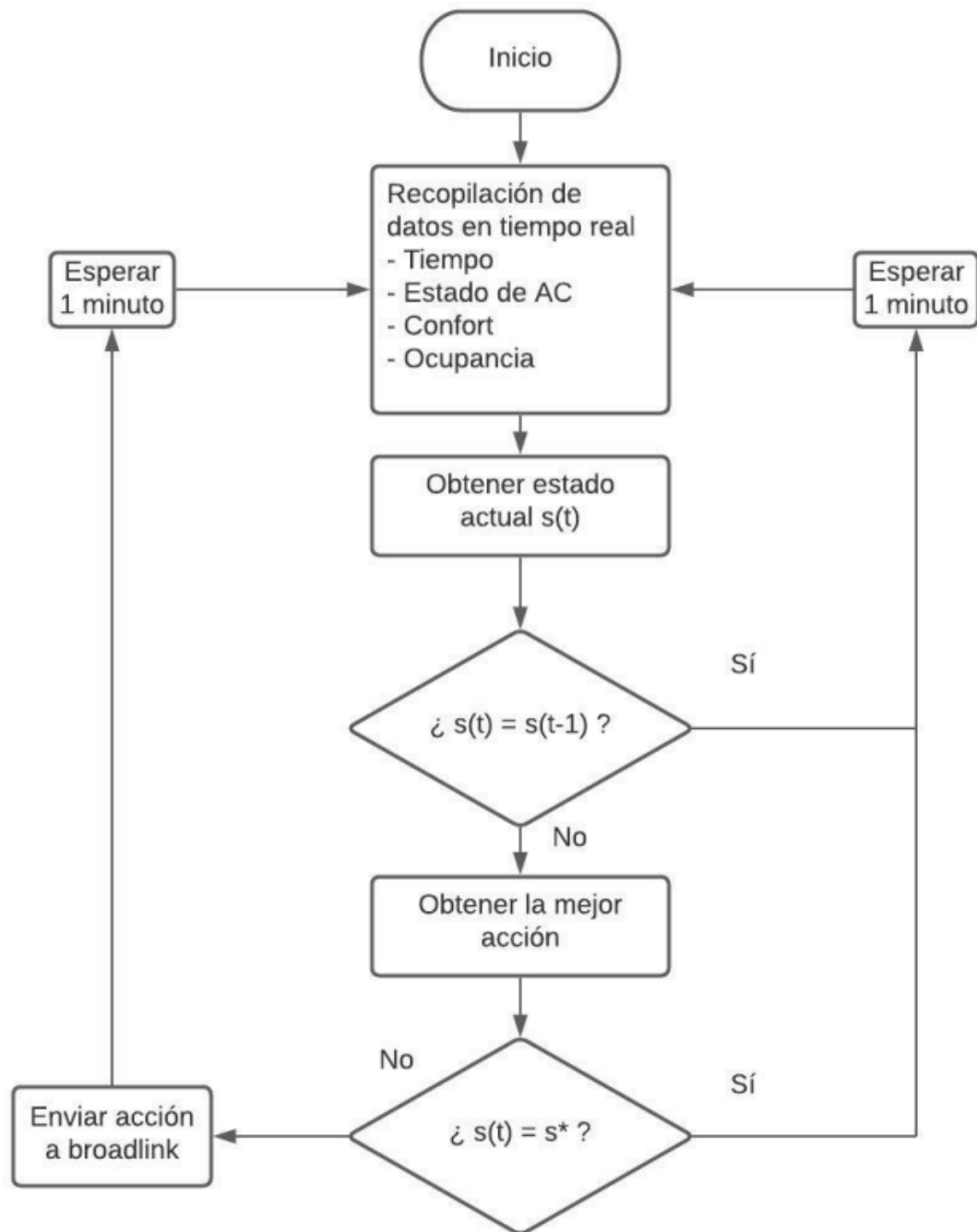


Figura 2.3: Diagrama de flujo del modo de uso basado en aprendizaje por refuerzo

El módulo está compuesto principalmente por el BroadLink RM 4 Mini⁹ (Figura 2.6) con las especificaciones técnicas según la Tabla 2.4¹⁰. Este dispositivo tiene incorporado un sensor infrarrojo permitiendo controlar a distancia el aire acondicionado mediante una comunicación bidireccional previamente establecida con el servidor. De esta manera es como el módulo de control inteligente se diseña, permitiendo recibir las instrucciones

⁹<https://www.broadlink.com.es/>

¹⁰<https://www.powerplanetonline.com/es/broadlink-rm4-mini-google-home-alexa>

dadas por el aplicativo móvil procesandolas en el servidor alojado en la Raspberry Pi 3 para finalmente enviar al mando Broadlink RM 4 mini. A diferencia de otros sensores infrarojos existentes en el mercado como el LED emisor de infrarojos, el Broadlink es un control universal¹¹ para cualquier dispositivo de aire acondicionado sin importar la marca.

La comunicación entre el sensor infrarojo y el aplicativo móvil se desarrolló mediante la utilización de una Raspberry Pi debido a que brinda la posibilidad de alojar un servidor que procese la información y envíe los datos hacia el aplicativo móvil y viceversa, utilizando el protocolo HTTP¹². Otra opción que se tomó en cuenta para la comunicación del módulo es el uso del protocolo MQTT mediante un Broker MQTT de código abierto basado en el sistema publicador/subscriptor para el envío y recepción de los datos[23] entre el sensor y el aplicativo móvil.



Figura 2.4: Mando infrarojo a distancia Broadlink RM4 Mini

¹¹<https://tecnomarket.ec/products>

¹²<https://developer.mozilla.org/es/docs/Web/HTTP>

Características	Especificaciones
Marca	Broadlink
Modelo	RM4 Mini
Conectividad	Wi-Fi 802.11 b/g/n (2.4GHz)
Humedad entorno instalación	<85%
HRConsumo	<0.69W
Voltaje	5V 1A
Frecuencia Infrarroja	38kHz
SO Compatible	iOS 9.0 / Android 4.0 y superior
Certificaciones	CE, RoHS, FCC ID
Dimensiones	48 x 48 x 43.2 mm
Peso	60 g

Tabla 2.4: Especificaciones técnicas de Broadlink RM 4 mini

CAPÍTULO 3

3. PRUEBAS Y RESULTADOS

El sistema integrado de confort para ambientes inteligentes se evalúa en un escenario en particular con la finalidad de obtener un análisis acerca de la solución propuesta. Durante este capítulo se revisará aspectos como escenario para línea base, resultados de las pruebas de los dos modos de uso en el escenario propuesto, estudio económico y los resultados de la interfaz gráfica donde el usuario va a interactuar con la solución dada.

3.1 Escenario de pruebas

El escenario de pruebas está localizado dentro de las instalaciones de ESPOL, específicamente en el Laboratorio de Sistemas Telemáticos de aproximadamente $54m^2$ dividido en cuatro mesas de trabajo, oficina del jefe de laboratorio y un baño, tal y como muestra la figura 3.1. Dentro del laboratorio, existen dos aires acondicionados de la marca Panasonic de 24000 BTU. Estos dispositivos fueron la pieza fundamental para la realización de las pruebas ya que con ellos determinamos el confort adecuado para el usuario dentro del Laboratorio.

Las pruebas consisten en primer lugar en la obtención de datos como: temperatura exterior, estado del aire acondicionado, votos del usuario sobre su confort actual y comportamiento de los estados según el modo de aprendizaje por refuerzo. Con esos datos y en conjunto con una encuesta realizada a los usuarios se contrastará y analizará la viabilidad del sistema integrado de confort para ambientes inteligentes.

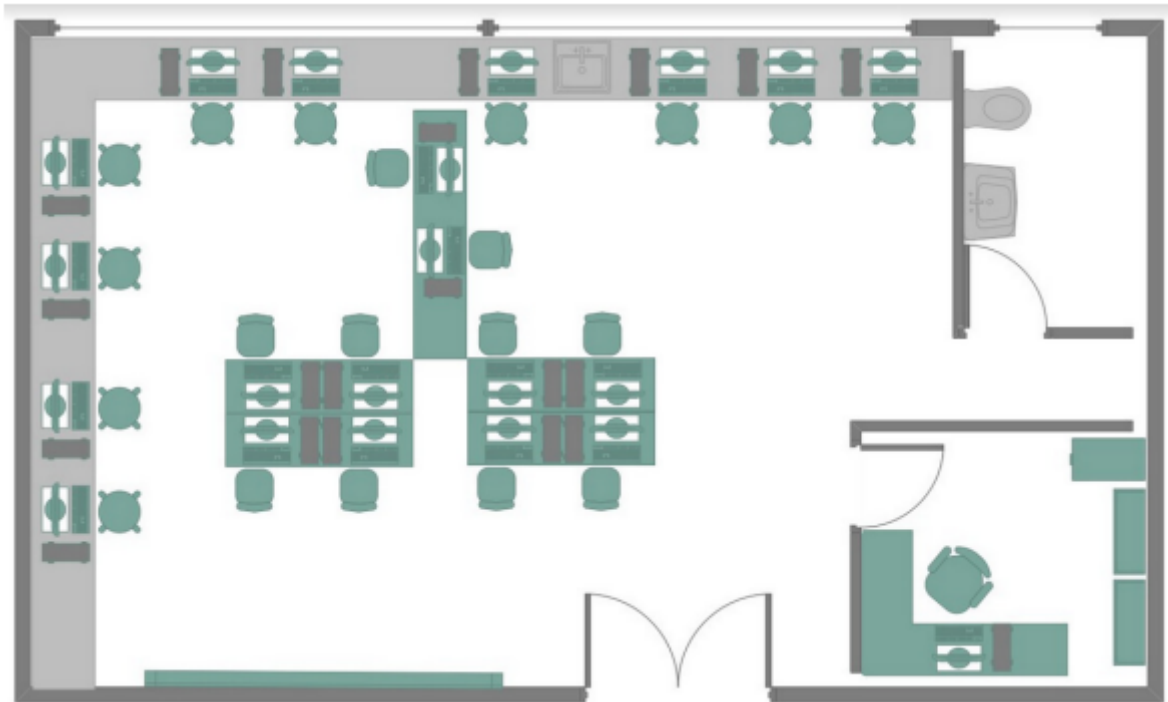


Figura 3.1: Croquis del Laboratorio de Sistemas Telemáticos

3.2 Control térmico basado en reglas

El control térmico basado en reglas se realizó en dos fases. En la primera fase se siguió la idea de un sistema de recomendaciones al usuario donde los datos fueron obtenidos mediante la recomendación de temperatura y la decisión del usuario de tomar o no la recomendación generada por el sistema. En la segunda fase se decidió no generar recomendaciones al usuario y en lugar de ello se empleó un sistema de votación que en conjunto con las reglas predefinidas envía directamente la nueva acción a realizar a los aires acondicionados.

3.2.1 Prueba 1: Sistema basado en recomendaciones al usuario

La primera prueba se realizó durante la semana del 21 al 25 de noviembre del 2022. Durante esa semana se recolectaron datos sobre la elección del confort y las recomendaciones que el sistema brindaba a los usuarios. Como se puede visualizar en la figura 3.2 se evidencia una mayor elección de la opción "cold" la cual indica que se encuentran con frío, es decir, el estado de confort actual. Por otro lado, las recomendaciones dadas por el sistema se visualizan en la figura 3.3 donde se puede

evidenciar la recomendación de "apagar" como la predominante en contraste con las demás.

La eficiencia del sistema se ve reflejada en las recomendaciones dadas al usuario. Partiendo del nivel frío (Elegido es más ocasiones) donde se espera una respuesta apropiada del sistema como un aumento de temperatura o apagar el aire acondicionado dependiendo de la temperatura actual. En la figura 3.3 "apagar" y las opciones de 20 y 23 grados Celcius fueron las recomendaciones mayormente dadas al usuario verificando un funcionamiento correcto del control térmico basado en reglas.



Figura 3.2: Elección del confort según el usuario

3.2.2 Prueba 2: Sistema basado en votaciones del usuario

La fase dos del control térmico basado en reglas consiste en el sistemas de votos. Esto surge como una alternativa para validar de mejor manera los resultados y ofrecer mayor eficiencia del sistema en cuestión. A pesar de que los resultados de la fase uno fueron muy buenos dando recomendaciones aceptables según el confort del usuario presentaba ciertos inconvenientes como la consideración solamente de la última elección de confort durante el periodo de tiempo dado (cada hora) lo cual era ineficiente ya que otro usuarios pueden sentir un nivel de confort distinto al dado por el sistema y adicionalmente el periodo de toma de datos era muy extenso para dar una recomendación de mejora de confort.

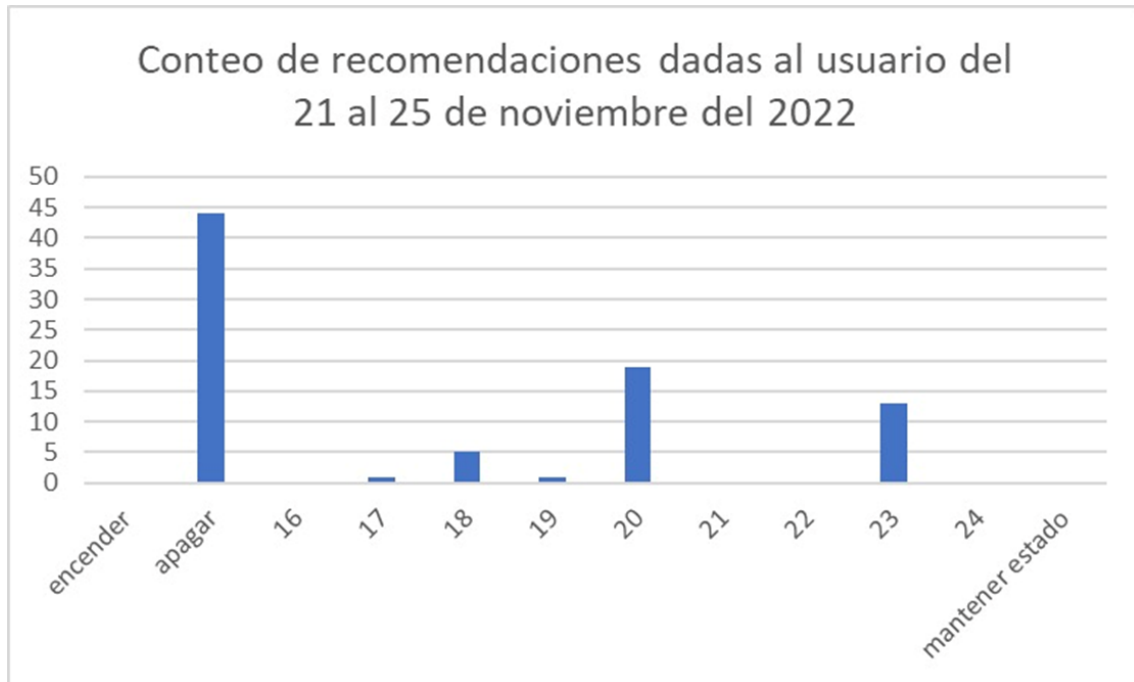


Figura 3.3: Conteo de recomendaciones dadas al usuario

Se realizaron pruebas desde el viernes 9 de diciembre hasta el 16 de diciembre sin contar los fines de semana debido a que nuestro escenario de pruebas se aloja dentro de una institución educativa. Particularmente se obtienen los datos de un día en específico cada 15 minutos debido a una mayor interacción con el sistema por parte de los usuarios.

Las elecciones de temperatura (figura 3.5) que el sistema tomó para mejorar el confort van de acuerdo a las reglas establecidas en el capítulo anterior y al nivel de confort ganador en base a los votos entre las tres opciones dadas por el sistema como se visualiza en la figura 3.4. Gracias al sistema de votos ahora si se consideran cada uno de las elecciones del usuario acerca de su confort a diferencia de la prueba 1. Durante este día se puede visualizar que las temperaturas elegidas por los usuarios se encuentra entre los 16 y 20 grados Celcius. Adicionalmente un dato importante son los niveles de confort que durante la mañana los usuarios elegían mayormente "calor" y a partir del medío día su elección cambió radicalmente a "frío" esto es causa a varios factores como la temperatura externa y el estado del aire acondicionado antes y después de cada clase. En la mañana los aire acondicionado se encuentran apagados por lo que al concentrarse un numero determinado de personas va a provocar sensación de calor entre ellos. A partir de la tarde ocurre lo contrario y esto se debe a que el laboratorio ya ha tenido interacción con

Votación de nivel de confort

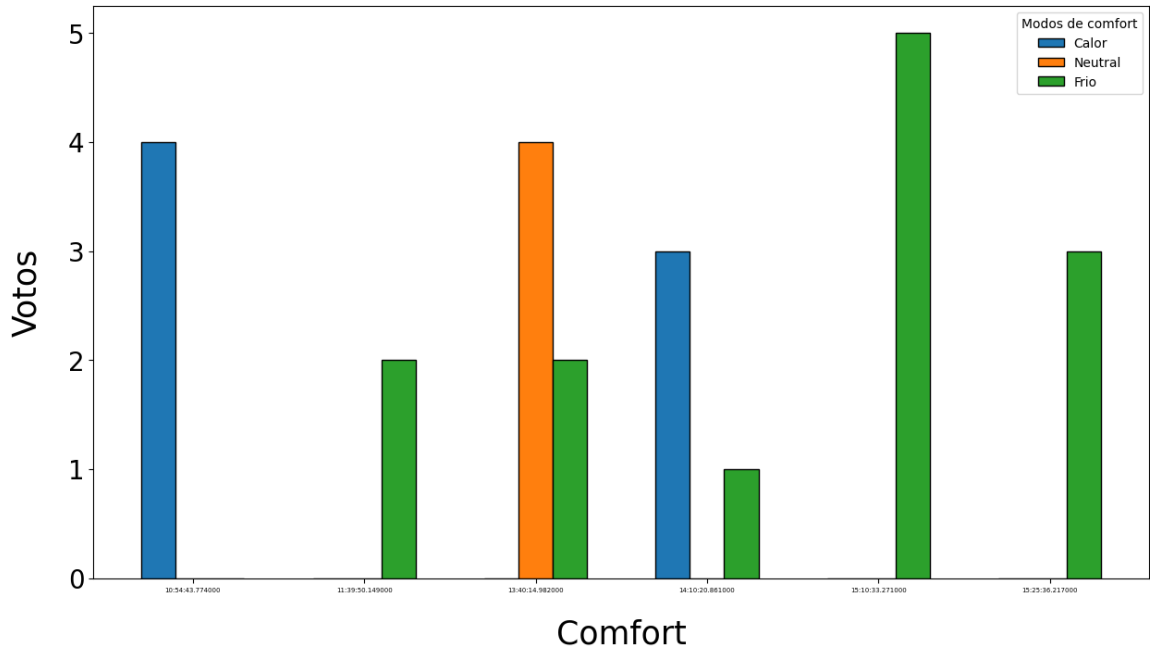


Figura 3.4: Conteo de votos durante el 15 de diciembre del 2022

Votación de nivel de confort

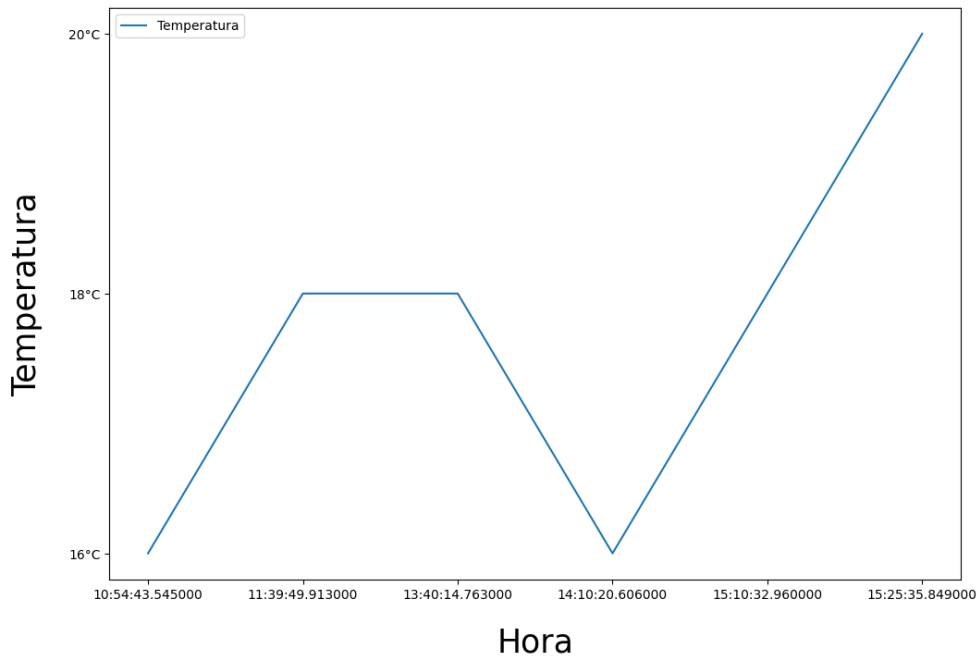


Figura 3.5: Elección de temperatura del sistema por medio del sistema de votos durante el 15 de diciembre del 2022

estudiantes y profesores por lo que el ambiente se encuentra en un estado de confort previamente establecido, por ende la sensación de frío se ve reflejada en los usuarios

que interactúan durante la tarde.

3.2.3 Análisis de resultados del control térmico basado en reglas

A pesar de que los resultados obtenidos en las pruebas realizadas fueron muy buenos desde un punto de vista lógico no es razón suficiente para cuantificar la eficiencia del sistema dentro del laboratorio de Sistemas Telemáticos. Por ello desarrollamos un formulario en Google Forms con el objetivo de preguntarle a los usuarios que estuvieron presentes en las fechas de las pruebas para este modo de control de térmico acerca del confort dentro del laboratorio.

El formulario que desarrollamos cuenta con preguntas muy puntuales acerca del confort del usuario que hacen referencia a como se encuentran en ese instante y como les gustaría sentirse igualmente en términos de confort.

¿Cómo se siente en este preciso momento?

16 respuestas

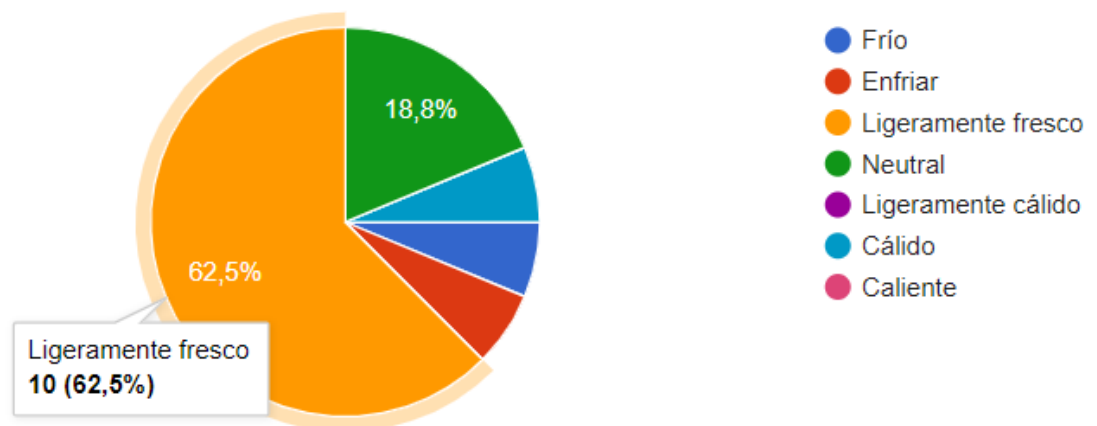


Figura 3.6: Pregunta 1 del formulario de Google

Las primeras dos preguntas del formulario tienen referencia al confort del usuario en el instante preciso de utilizar el sistema, donde podemos visualizar en la figura 3.6 que de el total de personas que realizaron el formulario un 62.5% ha elegido la opción de **ligeramente fresco** lo que nos da a entender que no se encuentran en un punto medio de satisfacción con respecto a la temperatura del lugar ya que la respuesta esperada es una opción de satisfacción plena respecto a la temperatura que sería la opción de **Neutral** la cual solo un 18.5% la ha elegido. En la figura 3.7 ya nos preguntan de

¿Qué tan cómodo con el ambiente se encuentra en este momento?

16 respuestas

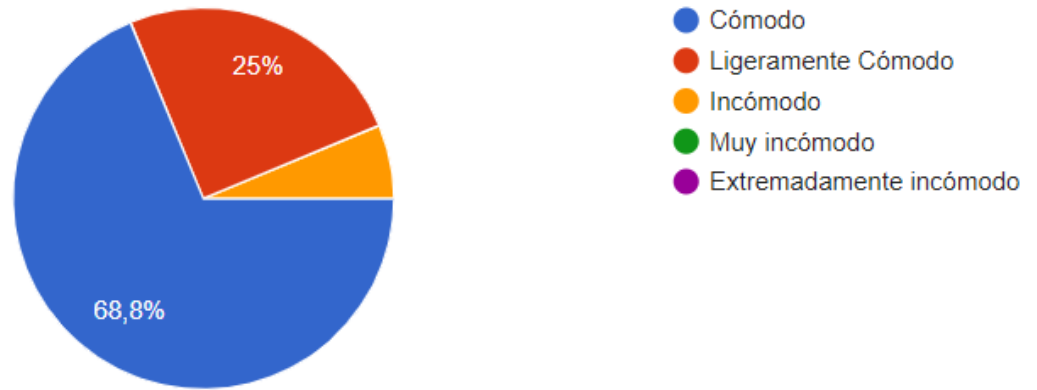


Figura 3.7: Pregunta 2 del formulario de Google

forma precisa acerca de la comodidad al utilizar el sistema con un resultado de 68.8% **Cómodo** coherente con el resultado de la figura 3.6 donde **Ligeramente fresco** también es sinónimo de comodidad en términos generales y un 25% que indica **Ligeramente cómodo** dando a entender que no sienten incomodidad, pero que se podría mejorar dicho confort.

¿Cómo preferiría sentirse usted en este instante?

16 respuestas

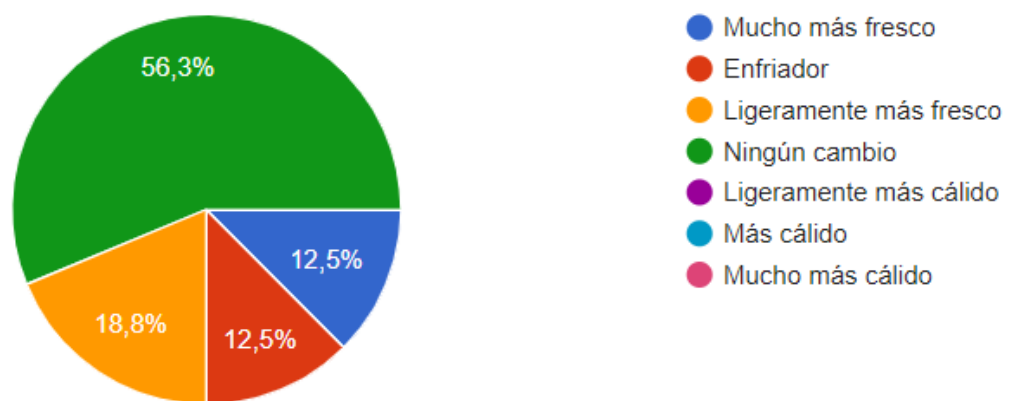


Figura 3.8: Pregunta 3 del formulario de Google

Para responder a lo mencionado al final del párrafo anterior surgen las preguntas tres y cuatro del formulario debido a que no solo es importante conocer el estado de confort en ese preciso instante sino tener una retroalimentación de como preferirían o esperarían

¿Qué tan aceptable es para usted el ambiente térmico en estos momentos?

16 respuestas

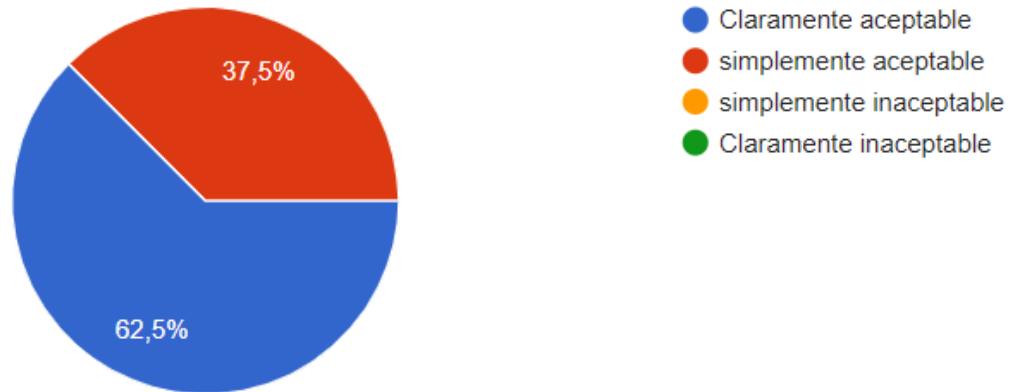


Figura 3.9: Pregunta 4 del formulario de Google

sentirse en término del confort con el sistema propuesto. La figura 3.8 nos dice que el 56.3% de las personas eligieron **Ningún cambio**, mientras que el resto del porcentaje se reparten entre **ligeramente más fresco**, **enfriador** y **Mucho más fresco**. Dando a notar que aproximadamente el 50% no se encuentra conforme con el sistema y preferían una acción distinta sobre la temperatura del lugar y finalmente la figura 3.9 indica que un 62.5% de las personas aceptan el sistema con claridad.

3.3 Control térmico basado en aprendizaje por refuerzo

El segundo modo de control térmico de nuestra solución usa un algoritmo de aprendizaje por refuerzo [24] denominado **Value Iteration**[25] el cual nos permite determinar el mejor estado posible y la mejor acción basado en distintos parámetros como el confort (votaciones de los usuarios), temperatura externa, periodo del día y el estado actual del aire acondicionado.

3.3.1 Prueba 3: Implementación de Value Iteration

La implementación de Value Iteration requiere definir algunos parámetros como la matriz de probabilidades de transición de estados, el valor gamma y la tolerancia. El valor de gamma requiere de un análisis previo para definir su valor, para ello se realizaron pruebas

con distintos valores de gamma. En la figura 3.10 se puede observar una tendencia menos lisa para valores gamma menores que 0.7 con un valor máximo de 3.3, a diferencia de gamma igual a 0.9 que a pesar de tener un número 329 iteraciones alcanza valores muy cercanos a 10 con una curvatura mucho más lisa. Finalmente para la realización de la prueba respectiva se consideró el valor de gamma igual a 0.9.

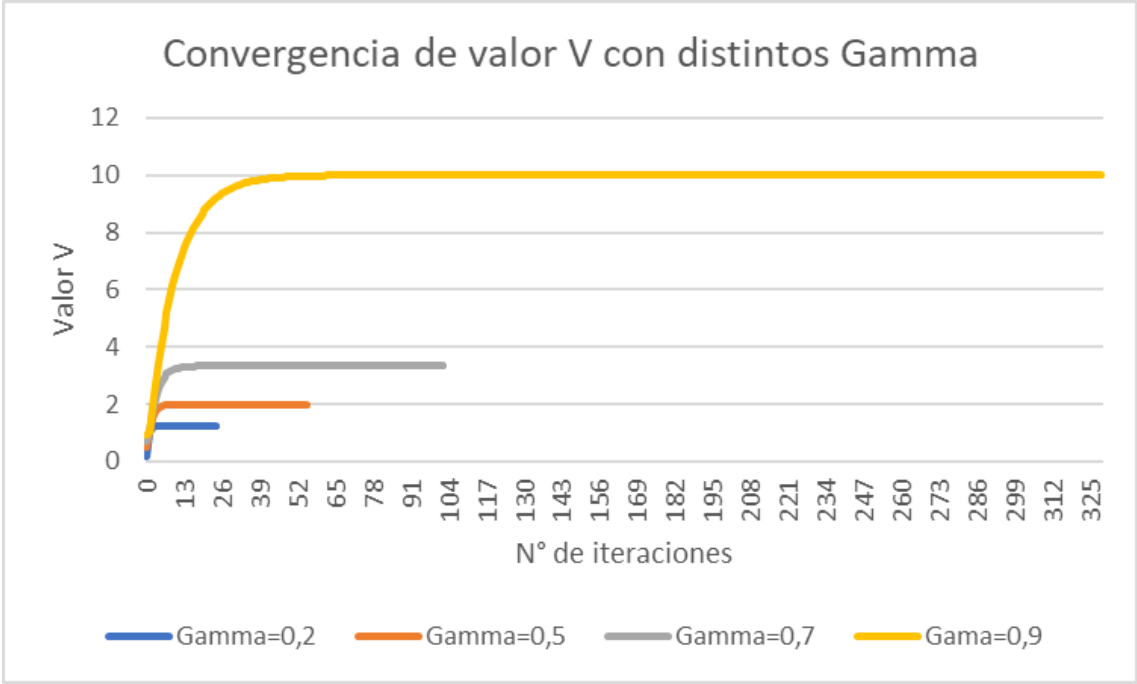


Figura 3.10: Convergencia de un estados para distintos valores de Gamma

Al definirse los parámetros respectivos se procede a las pruebas del sistema con Value Iteration, para ello se realizaron pruebas de la semana del 16 al 20 de enero del año 2023 donde se determinó aspectos relevantes como los estados más visitados y las temperaturas de esos estados óptimos. En la figura 3.11 se puede observar los estados y las veces que el sistema eligió dichos estados durante un día completo, es importante mencionar que los estados ausentes no fueron incluidos en el gráfico ya que tienen un valor de cero visitas a dichos estados. De igual forma para complementar el gráfico 3.11 se pone en evidencia la figura 3.12 la cual nos muestra la interacción de los estados ya mencionados durante el día completo mediante las estampas de tiempo donde se puede apreciar que el estado predominante es el número 13 desde aproximadamente las 9 de la mañana hasta el medio día y luego una variación casi continua desde el medio día hasta las 4 de la tarde. Esta última parte de la figura 3.12 es muy importante ya que se puede apreciar una interacción constante del usuario con el sistema. En ese caso el resultado

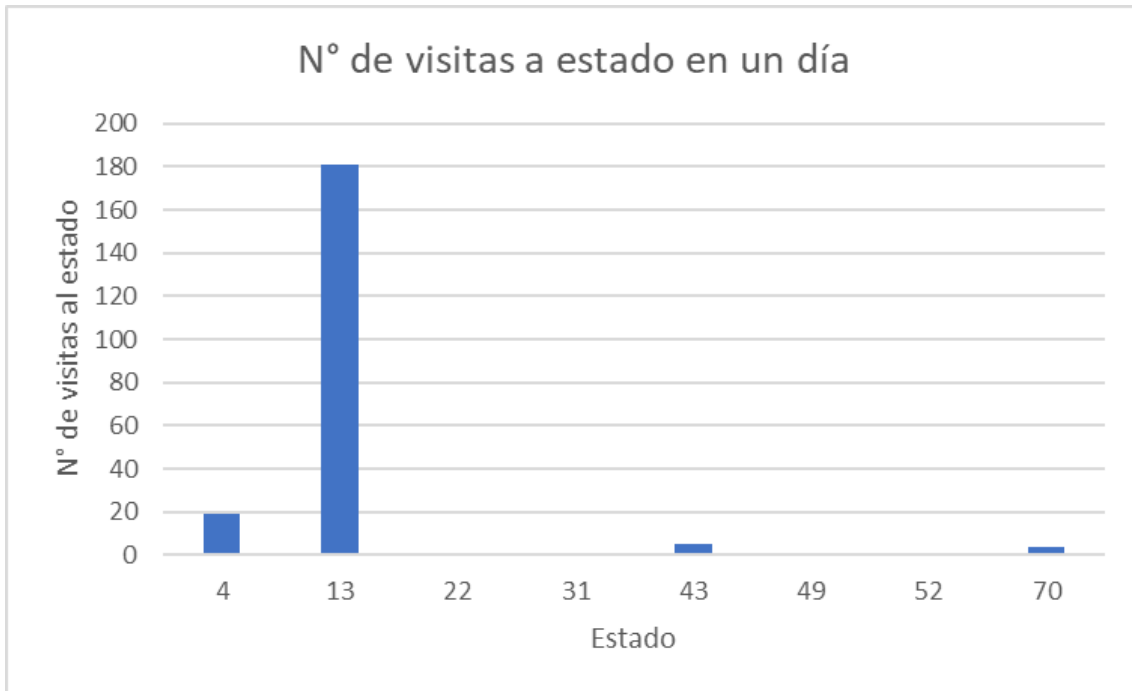


Figura 3.11: Número de visitas a los distintos estados durante un día completo



Figura 3.12: Comportamiento de estados en un día determinado

es el esperado debido a que es un horario donde se dictan clases continuamente y cada grupo que ingresa a la siguiente clase ajusta su nivel de confort (votación del nivel de confort).

3.3.2 Análisis de resultados del control térmico basado en aprendizaje por refuerzo (value iteration)

De la misma manera que en el control térmico basado en reglas, también se debe validar la eficiencia del sistema si resultó mejor o no en contraste con el sistema basado en reglas. Cabe recalcar que este modo de uso mejora día con día con las transiciones de los estados que alimenta la matriz de probabilidades que a su vez mejora el rendimiento y la precisión del sistema al determinar el mejor estado posible y su respectiva acción. Luego de realizar varias pruebas durante esa semana se decidió realizar finalmente la encuesta a un grupo de estudiantes que dieron uso al sistema con la finalidad de validar la eficiencia del mismo.

Las preguntas del formulario de Google Forms son las mismas que se hicieron en el control térmico basado en reglas y se encuentran en [apendice tal]. Globalmente los resultados del control térmico basado en aprendizaje por refuerzo tienen promedios que superan al control térmico basado en reglas. Por ejemplo la figura 3.7 los usuarios respondieron que estaban cómodos con el ambiente en un 68.8% mientras que cuando usaban value iteration ese valor se incrementó a un 81.3% lo que significa una eficiencia mayor y gran aceptación por parte de los usuarios que hicieron uso de este modo de control térmico.

3.4 Costos del proyecto

Nuestra solución final consiste en el sistema descrito en la figura 2.1 consiste de una capa de presentación e interacción con el cliente (aplicación web), capa de lógica e AI y capa de almacenamiento. Nuestro análisis se enfocará primero en el costo de crear el primer MVP, es decir el costo unitario. A partir de este precio escalaremos y nos proyectaremos a unos 2 años de inversión, desarrollo, producción y salida al mercado. A continuación, se muestra el desglose de los costos dividido en hardware y software:

A nivel de hardware, el precio de los componentes fueron cotizados tanto local como internacionalmente por medio de distribuidores que ofrecen productos de alta calidad, para garantizar el óptimo funcionamiento del sistema.

Con respecto a software, se hace la respectiva cuantificación económica por mano de

Descripción	Valor
Broadlink RM 4C mini	\$20.00
Kit Raspberry Pi4	\$125.00
Total:	\$145.00

Tabla 3.1: **Costos de Hardware**

obra de los distintos profesionales que intervienen en esta etapa de la solución y por los servicios en la nube que estamos utilizando.

Descripción	Costo por hora	Valor
Desarrollador Frontend (320 horas)	\$5.00/h	\$1600.00
Administrador de Backend (160 horas)	\$6.00/h	\$960.00
Desarrollador de AI (160 horas)	\$6.00/h	\$960.00
Firebase Plan Blaze	Depende del uso	
	Total:	\$3520.00

Tabla 3.2: **Costos de desarrollo del software**

Ya que el precio de Firebase depende del uso vamos a desarrollarlo un poco en la siguiente sección:

3.4.1 Costos de Firebase

Los dos servicios principales de Firebase que usamos son la Realtime Database y Firestore. Cada una tiene una tiene un coste diferente de acuerdo sus funcionalidades.

Para Realtime Database se cobra por el espacio que ocupa la base de datos y por las peticiones de lectura y escritura. Para el almacenamiento son 5 USD por cada gigabyte, con 1 GB gratis y para las peticiones son 1 USD por cada 5 GB, con lo primeros 10 GB gratis. Ya que nuestra base de datos es bastante pequeña realmente no tendríamos ningún importe por el primer rubro. Para el segundo se hará una pequeña estimación utilizando valores de ejemplos.

Cómo se puede notar en la tabla 3.3 para alcanzar recién el techo gratis que otorga el servicio, necesitaríamos tener al menos 400000 usuarios que hagan 1250 peticiones al mensuales. esto para una app que recién sale es bastante improbable por lo que en los

Descripción	Peso
Lectura	20 bytes
Escritura	20 bytes
Usuario hace 1250 peticiones	25 KB
1000 usuarios	25 MB
400,000 usuarios	10 GB
1,000,000 usuarios (peor caso)	25 GB

Tabla 3.3: **Valores de lectura/escritura en Realtime Database**

1

primeros años es probable que no tengamos que pagar por este servicio. Sin embargo, tomaremos el peor escenario posible diciendo que de salida nuestra aplicación se harán 15 GB de operaciones de lectura y escritura pagadas.

Por el lado de firestore, los precios son un poco más específicos así que los desarrollaremos en una tabla y a partir de eso vamos a desarrollar un pequeño ejemplo para determinar el mejor de los casos y el peor de los casos.

Operación	Precio por operación/unidad	Máximo gratuito
Lectura	\$0.06 / 100,000 documentos	50.000
Escritura	\$0.18 / 100,000 documentos	20.000
Eliminación	\$0.02 / 100,000 documentos	20.000
Almacenamiento	\$0.18 / GB	1 GB

Tabla 3.4: **Precios de Firestore**

Ahora, ciertamente depende de lo que se guarde en Firestone para determinar la cantidad de documentos a los a los que tengamos que acceder, sus pesos y la cantidad de almacenamiento que vayamos a necesitar. No es lo mismo guardar imágenes de alta resolución a pequeñas cadenas de caracteres y fechas. En el caso de nuestro proyecto los documentos que guardamos no son pesados y tienen pocos campos. Sin embargo, las lecturas y escrituras se hacen por parte de nuestro sistema basado en reglas y el de inteligencia artificial cada 15 minutos. Además, hay que recordar que el sistema está siendo usado en un solo ambiente. Estos 2 factores significan que mientras más más réplicas haya del producto en el mercado, la cantidad de operaciones que sean cada 15 minutos aumentarán de forma colosal, por lo que seguramente lleguemos

y sobrepasemos el techo de operaciones máximas gratuitas. Y esto se agrava más sabiendo que en las pruebas que hemos hecho hemos alcanzado un pico máximo de 5000 escrituras diarias. Entonces plantearemos un precio de Firestore para las siguientes condiciones: tenemos 50 clientes que compraron el producto y lo usan activamente, alcanzando las 5000 operaciones al día. Se harían entonces 250,000 operaciones diarias. En lectura, por ejemplo, son 0.18 USD por cada 100,000 operaciones. Esto daría un valor de 0.45 USD diarios, lo que mensualmente se traduciría a 13.5 USD mensuales.

Teniendo en cuenta ambos valores podemos sacar un precio estimado de lo que nos costaría tener el servicio de Google Firebase mensualmente.[26] realmente es un precio bastante asequible por lo que este servicio es un es un exelente elección para nuestro proyecto. El precio final sería de 28.5 USD como se puede ver en la tabla 3.5.

Servicio	Valor Mensual
Realtime Database	\$15.00
Firestore	\$13.50
Total:	\$28.50

Tabla 3.5: **Costo en el peor caso de Firebase**

3.4.2 Valor unitario del producto

Después de haber analizado los costos vamos a determinar el valor unitario de nuestro proyecto. Realmente lo que vamos a vender es el hardware o el aparato que se va a conectar a internet y también a los aires acondicionados para asegurar el confort de los usuarios. La plataforma web va a contar más como una inversión inicial que como un costo de producción. Por esto solo tomaremos en cuenta el hardware. Los gastos de Firebase los cubriremos con otro medio de ingresos, el cual describiremos en la sección de modelo de negocio.

Como se describió, que el precio de los 2 componentes la Raspberry Pi y el Broadlink es de 140 dolares en conjunto, es decir que esto nos costaría construir una unidad. Teniendo en consideración que no planeamos vender un gran volumen de unidades en los primeros años creemos que la mejor decisión es buscar un margen de ganancia del 30%, por lo que nuestro producto finalmente costaría \$280.

3.5 Análisis Económico

Habiendo desarrollado el precio unitario de nuestro producto, podemos realizar la proyección correspondiente. Describiremos nuestro modelo de negocio y haremos un análisis financiero de costos fijos, costos variables, punto de equilibrio y rentabilidad.

Precio unitario	\$280.00
------------------------	----------

Tabla 3.6: **Precio de venta unitario**

Cabe recalcar que ese precio de venta fue determinado tomando en cuenta hoy todas las variables discutidas hasta ahora y los costos variables y fijos que discutiremos en las siguientes 2 secciones. Cómo se me verá más adelante en diferentes gráficos y tablas este valor nos permite recuperar la inversión y generar ganancias en un tiempo aceptable, manteniendo un precio accesible para nuestro mercado objetivo.

3.5.1 Costos fijos

Los costos fijos son aquellos que no varían con la producción o venta del producto. Estos costos son constantes y deben ser cubiertos independientemente de la cantidad de unidades producidas o vendidas.

Servicio	Valor Mensual
Administrador el Backend	\$800,00
Ingieniero de Operaciones	\$800,00
Firebase	\$30.00
Total:	\$1630.00

Tabla 3.7: **Costos fijos mensuales**

Nuestros costos fijos constan de 3 rubros, hoy 2 empleados que se encargarán de la administración de nuestro sistema. Además, uno de ellos también se encargará de hacer las instalaciones. y por último los costos de Firebase mensuales en el peor de los casos. esto da un total de costos fijos de \$1630.

Es importante recalcar que los costos fijos también influyen en la determinación del precio de venta del producto. Si los costos fijos son altos, entonces el precio de venta

del producto deberá ser más alto para cubrir los gastos. Además, los costos fijos no son inmutables, ya que pueden ser reducidos a través de una variedad de medidas como la optimización de procesos, la automatización, la negociación con proveedores y la eliminación de gastos innecesarios.[27]

3.5.2 Costos Variables

Los costos variables son aquellos que varían en función de la producción o venta del producto. Estos costos incluyen gastos directamente relacionados con la producción o venta del producto, como materias primas, mano de obra directa y gastos de envío.

Servicio	Valor Mensual
Microcontrolador	\$125,00
Broadlink	\$20,00
Total:	\$145.00

Tabla 3.8: **Costos fijos mensuales**

Como ya fue mencionado en la sección anterior cuando determinamos el precio de venta unitario, nuestros únicos costos variables son por los componentes que necesitamos para construir nuestra solución, hoy esto da en total \$140. Es probable que más adelante surjan otros rubros en los que haya que gastar, pero por ahora esto es lo esencial.

Es importante identificar y controlar los costos variables para tener una comprensión precisa de los gastos totales y para determinar el punto de equilibrio, que es el nivel de producción o ventas donde los ingresos y los costos se igualan. Este indicador que desarrollaremos en la siguiente sección.

3.6 Punto de equilibrio

Para conocer cuantas unidades debemos vender para poder cubrir los costos de producción utilizaremos el indicador económico conocido como punto de equilibrio. Este nos proporcionara puntos de referencia importantes para la planificación a largo plazo de nuestro negocio.[28]

Es importante destacar que este análisis financiero se enfoca en los costos de inversión y determina el número de productos que deben venderse mensualmente para alcanzar el punto de equilibrio. La fórmula utilizada para este cálculo es

$$Q_e = \frac{C_{FijoMensual}}{P_{ValorUnitario} - C_{ValorUnitario}} \quad (3.1)$$

La solución de esta fórmula indica que es necesario vender aproximadamente 12 unidades del producto para comenzar a ser rentables. En la figura 3.10 se muestra que efectivamente este es el número, además, haciendo el cálculo, también podemos determinar el ingreso de equilibrio. Este es de \$3.380,75

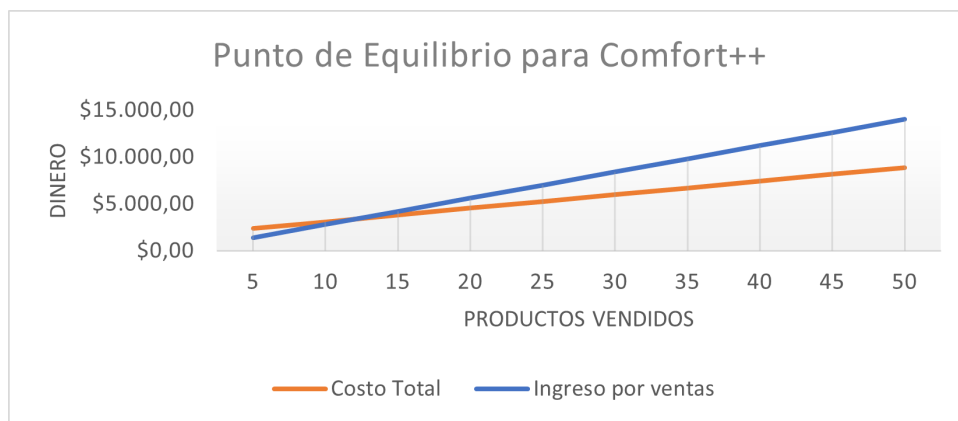


Figura 3.13: **Punto de Equilibrio**

Conociendo este dato, nos proponemos el alcanzar al menos 15 ventas mensuales, de forma que podamos generar ganancias desde el primer mes. Con esto podremos alcanzar más rápido las cifras que queremos para ser rentables, y hacer de nuestro producto exitoso. En la siguiente sección exploraremos la rentabilidad del producto y veremos el flujo de caja tomando en cuenta que se cumplen la meta de ventas mensuales.

3.7 Modelo de Negocio

Nuestra idea de modelo de negocio era simplemente el cómo se hace dinero. Sin embargo, según Aramis Rodríguez, profesor en IESA (Instituto de Estudios Superiores de Administración) en Venezuela, este concepto va más allá: “El modelo de negocio es la lógica empleada por una organización para crear, entregar, y capturar valor” [29]. Sentimos que ambos conceptos son importantes. Sobre el primero, nuestra principal

fuerza de ingresos será la venta del producto, con una clave para acceder a la plataforma web. La cantidad máxima de dispositivos IoT que se podrán controlar serán 2. Si se desea agregar más al sistema, se lo podrá hacer pagando una suscripción mensual en la página. Esta será nuestra segunda fuerza de ingresos, que servirá para cubrir los costos de los servicios de Firebase. Por otro lado, nuestro producto entrega a los usuarios el confort que necesitan para realizar de la mejor forma sus actividades, de esta forma estamos creando y entregando valor.

Ahora bien, también es importante definir quiénes van a ser nuestros clientes. hoy en la primera etapa de nuestra salida al mercado creemos que el objetivo ideal serían oficinas de empresas. Según el Directorio de Empresas 2021[30], una estadística hecha por el INEC, en Ecuador existen 4,234 grandes empresas. Estas compañías mueven miles de millones de dólares al año. En el peor de los casos, si cada una de estas empresas tiene únicamente 1 oficina, entonces tendremos 4,234 clientes potenciales. Cada uno con un alto poder adquisitivo. Esto es sin contar a otras empresas que también pueden tener oficinas pero que no sean consideradas empresas grandes, es decir, este mercado tiene muchísimo potencial.[31]

Dicho todo esto, tenemos un modelo de negocio sólido, un mercado objetivo muy prometedor y dado el análisis económico que hemos hecho, se puede asegurar con altos niveles de confianza que nuestro producto tendrá éxito. En la siguiente sección veremos que tan rentable será.

3.8 Rentabilidad

Para cerrar la sección financiera, realizaremos el análisis de rentabilidad mediante el valor actual neto (VAN) y la tasa interna de retorno (TIR) [32][33].

Para esto necesitamos primero calcular el flujo de caja. El flujo de caja es un registro de todos los ingresos y gastos de una empresa o proyecto durante un período determinado, por lo general un año o un mes.[34] En nuestro caso lo haremos de una forma, poco común. proyectaremos a 2 años y dividiremos cada año en trimestres. Esto por un tema más de preferencia en cómo mostrar los datos que por otra razón específica. El anexo G muestra nuestro flujo de caja desarrollado de la forma descrita.

Ahora para ver la rentabilidad del modelo de negocio se usarán la ecuación 3.2 para

el cálculo del VAN.[35]

$$VAN = (CF_1/(1+r)^1) + (CF_2/(1+r)^2) + \dots + (CF_n/(1+r)^n) - I \quad (3.2)$$

donde: CF = flujo de caja

r = tasa de descuento

n = número de períodos

I = inversión inicial

VAN	TIR
\$18.269,55	15%

Tabla 3.9: **Valor de VAN y TIR**

Después de hacer los cálculos, se obtuvo como resultado que el VAN es de \$18.269,55, generando ganancias desde el primer año y una TIR[36] positiva del 15%. Finalmente, como demuestran los valores de la tabla, el proyecto es rentable y viable en términos económicos y no hay pérdidas sino ganancias que se generan a lo largo del tiempo.[37]

CAPÍTULO 4

4. CONCLUSIONES, RECOMENDACIONES Y LÍNEAS FUTURAS

4.1 Conclusiones

- Se diseñó una interfaz gráfica para el monitoreo del sistema de control térmico donde se puede visualizar en tiempo real la temperatura actual del aire que se está controlando. La UI permite a los usuarios dar su input, en forma de un voto, a nuestro sistema de cómo se sienten en términos de confort térmico. Este voto, en conjunto con el periodo del día, el estado actual del aire acondicionado y la temperatura externa del sitio, son las variables que se utilizan para brindar al usuario el mejor estado de confort posible dentro de las instalaciones donde se aloja el proyecto.
- Se utilizó principalmente el servicio de almacenamiento de Firebase como base de datos del sistema de control térmico. Al ser un servicio en la nube brinda grandes beneficios al reducir potencialmente el uso de recursos del microcomputador ofreciendo tiempos de respuestas muy cortos en la determinación del mejor estado posible mediante el análisis de los datos alojados en la base de datos. Adicionalmente permite almacenar una masiva cantidad de datos ofreciendo gran escalabilidad al ser una base de datos NoSQL.
- Utilizando todas estas herramientas, construimos nuestros modos de confort. El primero que levantamos fue el modo basado en reglas, elegidas por nosotros, con el objetivo de asegurar el confort. Utilizando solamente los votos de los usuarios, dependiendo de que estado gane (calor, neutral, frío), se sube, se baja o se mantiene la temperatura cada 15 minutos. Con este modo alcanzamos en nuestras encuestas un 62.5% de aceptación.

- Para el segundo modo, se implementó un algoritmo de aprendizaje por refuerzo conocido como value iteration donde mejora por completo el confort del usuario tomando en cuenta los parámetros descritos en secciones anteriores y con la actualización continua de la matriz de probabilidad con la cual se obtiene resultados más eficientes día a día (76.6% de aceptación), en contraste con el modo basado en reglas (62.5% de aceptación) donde se pudo apreciar en las encuestas realizadas a los usuarios que utilizaron ambos modos de control térmico una mejora considerable al utilizar el modo de aprendizaje por refuerzo.

4.2 Recomendaciones

- Aumentar el número de días de pruebas para el algoritmo de aprendizaje por refuerzo, puesto que al realizar un mayor número de pruebas los resultados acerca del confort de los usuarios serían más precisos y adicionalmente permitiría analizar el comportamiento del sistema en diferentes épocas del año.
- El dispositivo infrarrojo y la microcomputadora (Raspberry Pi 3) deben estar recubiertos en una caja protectora con la respectiva ventilación y en un punto de visión ideal con el aire acondicionado para que los tiempos de respuesta con el cambio de temperatura sean precisos al ejecutar la acción desde el Broadlink.
- De igual forma se debe considerar la contratación del plan de pago para el servicio de Firebase debido a la gran cantidad de almacenamiento de datos que necesita el algoritmo para su precisión a la hora de determinar el mejor estado que conlleva al mejor confort para el usuario, a su vez es imprescindible mejorar el servicio de comunicación entre el dispositivo, la plataforma web y el aire acondicionado que actualmente trabaja con MQTT de forma local, requiriendo utilizar un servicio en la nube para la utilización de forma remota del sistema.

4.3 Líneas Futuras

Como futuras mejoras para el presente proyecto, se puede considerar aumentar el número de variables como el número de personas dentro del sitio (ocupancia),

implementar sensores para la obtención de la temperatura real interna, humedad, entre otras. Con el propósito de evitar el uso del sistema en los periodos de tiempo donde no se encuentre nadie dentro del lugar para evitar la repercusión de gastos innecesarios de energía y brindar un mejor estado de confort considerando las variables como las descritas.

BIBLIOGRAFÍA

- [1] F. H. Mohamed Salleh, M. b. Saripuddin, and R. bin Omar, "Predicting thermal comfort of hvac building using 6 thermal factors," in *2020 8th International Conference on Information Technology and Multimedia (ICIMU)*, pp. 170–176, 2020.
- [2] N. Yildirim and H. Sekerci, "Performance assessment of university buildings based on provided thermal comfort," in *2019 2nd Asia Conference on Energy and Environment Engineering (ACEEE)*, pp. 7–11, 2019.
- [3] G. Gao, J. Li, and Y. Wen, "Deepcomfort: Energy-efficient thermal comfort control in buildings via reinforcement learning," *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 8472–8484, 2020.
- [4] G. Çalış, M. Kuru, and B. Alt, "Analysis of thermal comfort conditions in an educational building: A field study in izmir," in *Journal of Engineering Faculty*, vol. 22, pp. 93–106, Uludağ University, 2017.
- [5] Z. S. Zomorodian, M. Tahsildoost, and M. Hafezi, "Thermal comfort in educational buildings: A review article," *Renewable and Sustainable Energy Reviews*, vol. 59, pp. 895–906, 2016.
- [6] K. W. Tham and H. C. Willem, "Room air temperature affects occupants' physiology, perceptions and mental alertness," *Building and Environment*, vol. 45, no. 1, pp. 40–44, 2010. International Symposium on the Interaction between Human and Building Environment Special Issue Section.
- [7] L. Pérez-Lombard, J. Ortiz, and C. Pout, "A review on buildings energy consumption information," *Energy and buildings*, vol. 40, no. 3, pp. 394–398, 2008.
- [8] J. van Hoof, "Forty years of fanger's model of thermal comfort: comfort for all?," *Indoor air*, vol. 18, p. 182–201, June 2008.

- [9] O. Uviase and G. Kotonya, "IoT architectural framework: Connection and integration framework for IoT systems," *Electronic Proceedings in Theoretical Computer Science*, vol. 264, pp. 1–17, feb 2018.
- [10] OECD, *OECD Employment Outlook 2022. Hours worked (indicator)*. 2022.
- [11] OECD, *OECD Compendium of Productivity Indicators.GDP per hour worked (indicator)*. 2021.
- [12] S. S. Lang, "Warm offices linked to higher productivity," *Human Ecology*, vol. 32, p. 23, 03 2005. Copyright - Copyright Cornell University Mar 2005; Document feature - Photographs; Last updated - 2017-11-09.
- [13] M. Maksimovic, V. Vujovic, N. Davidović, V. Milosevic, and B. Perisic, "Raspberry pi as internet of things hardware: Performances and constraints," 06 2014.
- [14] H. Ghael, "A review paper on raspberry pi and its applications," 01 2020.
- [15] J. Y. Park, T. Dougherty, H. Fritz, and Z. Nagy, "Lightlearn: An adaptive and occupant centered controller for lighting based on reinforcement learning," *Building and Environment*, vol. 147, pp. 397–414, 2019.
- [16] S. G. Pérez Ibarra, J. R. Quispe, F. F. Mullicundo, and D. A. Lamas, *Herramientas y tecnologías para el desarrollo web desde el FrontEnd al BackEnd*. 2021.
- [17] N. Chatterjee, S. Chakraborty, A. Decosta, and A. Nath, "Real-time communication application based on android using google firebase," *Int. J. Adv. Res. Comput. Sci. Manag. Stud*, vol. 6, no. 4, 2018.
- [18] S. Gilbert and N. Lynch, "Perspectives on the cap theorem," *Computer*, vol. 45, no. 2, pp. 30–36, 2012.
- [19] C. Khawas and P. Shah, "Application of firebase in android app development-a study," *International Journal of Computer Applications*, vol. 179, no. 46, pp. 49–53, 2018.
- [20] L. Moroney, *The Firebase Realtime Database*, pp. 51–71. Berkeley, CA: Apress, 2017.

- [21] F. Luo, G. Ranzi, W. Kong, G. Liang, and Z. Y. Dong, "Personalized residential energy usage recommendation system based on load monitoring and collaborative filtering," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 2, pp. 1253–1262, 2021.
- [22] Z. Cui, X. Xu, F. XUE, X. Cai, Y. Cao, W. Zhang, and J. Chen, "Personalized recommendation system based on collaborative filtering for iot scenarios," *IEEE Transactions on Services Computing*, vol. 13, no. 4, pp. 685–695, 2020.
- [23] A. Oak and R. Daruwala, "Assessment of message queue telemetry and transport (mqtt) protocol with symmetric encryption," in *2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC)*, pp. 5–8, 2018.
- [24] L. Yu, Y. Sun, Z. Xu, C. Shen, D. Yue, T. Jiang, and X. Guan, "Multi-agent deep reinforcement learning for hvac control in commercial buildings," *IEEE Transactions on Smart Grid*, vol. 12, no. 1, pp. 407–419, 2021.
- [25] B. Luo, Y. Yang, H.-N. Wu, and T. Huang, "Balancing value iteration and policy iteration for discrete-time control," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 11, pp. 3948–3958, 2020.
- [26] Google. *Firebase Documentation 2021* [Online].
- [27] S. A. Torres *et al.*, "Contabilidad de costos," 1996.
- [28] L. Mazón Arevalo, D. Villao Burgos, W. Núñez, and M. Serrano-Luyó, "Análisis de punto de equilibrio en la toma de decisiones de un negocio: caso grand bazar riobamba–ecuador," *Revista de estrategias del desarrollo empresarial*, vol. 3, no. 8, pp. 14–24, 2017.
- [29] M. Rodríguez, "Modelo de negocio," *Business Model*. Bogotá: Universidad de la Salle, Colombia, 2012.
- [30] Instituto Nacional de Estadista y Censos INEC, "Directorio de empresas 2021." Accessed Feb 02, 2023.
- [31] C. I. Maldonado, "Determinación del mercado objetivo y la demanda insatisfecha, cuando no se dispone de estadísticas," *Retos*, vol. 1, no. 1, pp. 12–12, 2011.

- [32] W. A. Valencia, "Indicador de rentabilidad de proyectos: el valor actual neto (van) o el valor económico agregado (eva)," *Industrial data*, vol. 14, no. 1, pp. 15–18, 2011.
- [33] G. S. Milanesi, "La tasa interna de retorno promedio borrosa: desarrollos y aplicaciones," *Journal of Economics, Finance and Administrative Science*, vol. 21, no. 40, pp. 39–47, 2016.
- [34] H. O. Anaya, D. A. O. Niño, *et al.*, "Flujo de caja y proyecciones financieras con análisis de riesgo (incluye cd)," *Books*, vol. 1, 2009.
- [35] M. R. Mete, "Valor actual neto y tasa de retorno: su utilidad como herramientas para el análisis y evaluación de proyectos de inversión," *Fides et Ratio-Revista de Difusión cultural y científica de la Universidad La Salle en Bolivia*, vol. 7, no. 7, pp. 67–85, 2014.
- [36] A. Chaves, H. Y. A. Gómez, Á. H. C. Castro, *et al.*, "Cálculo de la tasa interna de retornos educativos en Colombia," tech. rep., Universidad Externado de Colombia, 2021.
- [37] J. G. Altuve, "El uso del valor actual neto y la tasa interna de retorno para la valoración de las decisiones de inversión," *Actualidad contable FACES*, vol. 7, no. 9, pp. 7–17, 2004.

APÉNDICES

A Código de control térmico basado en reglas

En esta sección se puede visualizar el código en python del control térmico basado en reglas. El cual unicamente recibe los votos del confort de los usuarios y luego retorna un valor de temperatura de acuerdo a las reglas definidas por el programador.

Código 1: Código de control térmico basado en reglas

```
1 import time
2 from datetime import datetime
3 import firebase_admin
4 from firebase_admin import credentials
5 from firebase_admin import firestore
6 from firebase_admin import db
7 import numpy as np
8 import paho.mqtt.client as mqtt #import the client1
9 import random
10
11 # creacion de la app para firebase
12 cred = credentials.Certificate(
13     'energy-coach-f5270-firebase-adminsdk-o4rki-881d6550e6.json')
14 app = firebase_admin.initialize_app(cred, {
15     'databaseURL': 'https://energy-coach-f5270-default-rtdb.firebaseio.com/'
16 })
17 print(app.name)
18 dbF = firestore.client()
19
20 broker_address="127.0.0.1"
21 port = 1883
22 topic = "integradora/confort"
23 client_id = f'python-mqtt-{random.randint(0, 1000)}'
24 client = mqtt.Client(client_id)
25 client.username_pw_set("labtelematica", "l4bt3l3m4tic@")
26 client.connect(broker_address,port)
27
28 def sendToFirestoreRecomendations(optionChange, dtm, recomendacion):
29     data = {
30         u'date': dtm,
31         u'option': optionChange,
```

```

32     u'recomendation': recomendacion
33 }
34 dbF.collection(u'recomendations2').add(data)
35 return None
36
37
38 def sendToFirestoreComfort(comfortCould, comfortNeutral, comfortHold, dtm):
39     data = {
40         u'cold': comfortCould,
41         u'neutral': comfortNeutral,
42         u'warm': comfortHold,
43         u'date': dtm
44     }
45     dbF.collection(u'Comfort').add(data)
46     return None
47
48
49 # nueva Funcion
50
51
52 def accionRBC(option, confort):
53     option_change = 0
54     if option == 1: # ON
55         if confort == "cold":
56             option_change = 9
57         elif confort == "neutral":
58             option_change = 9
59         elif confort == "warm":
60             option_change = 7
61     elif option == 2: # OFF
62         if confort == "cold":
63             option_change = option
64         elif confort == "neutral":
65             option_change = option
66         elif confort == "warm":
67             option_change = 1
68     elif option == 3: # 16°
69         if confort == "cold":
70             option_change = option+2

```

```

71     elif confort == "neutral":
72         option_change = option
73     elif confort == "warm":
74         option_change = option
75 elif option == 4: # 17°
76     if confort == "cold":
77         option_change = option+2
78     elif confort == "neutral":
79         option_change = option
80     elif confort == "warm":
81         option_change = option-1
82 elif option == 5: # 18°
83     if confort == "cold":
84         option_change = option+2
85     elif confort == "neutral":
86         option_change = option
87     elif confort == "warm":
88         option_change = option-2
89 elif option == 6: # 19°
90     if confort == "cold":
91         option_change = option+2
92     elif confort == "neutral":
93         option_change = option
94     elif confort == "warm":
95         option_change = option-2
96 elif option == 7: # 20°
97     if confort == "cold":
98         option_change = option+2
99     elif confort == "neutral":
100        option_change = option
101    elif confort == "warm":
102        option_change = option-2
103 elif option == 8: # 21°
104    if confort == "cold":
105        option_change = option+2
106    elif confort == "neutral":
107        option_change = option
108    elif confort == "warm":
109        option_change = option-2

```

```

110 elif option == 9: # 22°
111     if confort == "cold":
112         option_change = option+2
113     elif confort == "neutral":
114         option_change = option
115     elif confort == "warm":
116         option_change = option-2
117 elif option == 10: # 23°
118     if confort == "cold":
119         option_change = option+1
120     elif confort == "neutral":
121         option_change = option
122     elif confort == "warm":
123         option_change = option-2
124 elif option == 11: # 24°
125     if confort == "cold":
126         option_change = option
127     elif confort == "neutral":
128         option_change = option
129     elif confort == "warm":
130         option_change = option-2
131 print("la opcion dada por las reglas es: ", option_change)
132 return option_change
133
134
135 while (1):
136     try:
137         #variables para encerrar la real time
138         confort_cold = 0
139         confort_neutral = 0
140         confort_warm = 0
141
142         # lectura de fecha actual
143         dtm = firestore.SERVER_TIMESTAMP
144
145         # temperatura actual del aire acondicionado
146         opcion = db.reference('recomendation').child('rec').get()
147         print("La opcion actual de temperatura es: ", opcion)
148

```



```

149     # valores de comfort (votaciones)
150     comfort = db.reference('comfort')
151     comfort_get = comfort.get()
152     comfort_cold_rt = comfort_get.get('cold')
153     comfort_neutral_rt = comfort_get.get('neutral')
154     comfort_warm_rt = comfort_get.get('warm')
155     print("Cold: %d, Neutral: %d, Calor: %d"
156           (comfort_cold_rt,comfort_neutral_rt,comfort_warm_rt))
157
158     if (comfort_cold_rt == 0 and comfort_neutral_rt == 0 and comfort_warm_rt == 0):
159         print("Sin datos, se mantiene el valor actual de
160               comfort y temperatura en el AC")
161         sendToFirestoreComfort(comfort_cold_rt,
162                               comfort_neutral_rt, comfort_warm_rt, dtm)
163
164     else:
165         print("si hay datos, se procede a validar la votacion.")
166         arr_comfort = np.array(
167             [comfort_cold_rt, comfort_neutral_rt, comfort_warm_rt])
168         lista_comfort = ["cold", "neutral", "warm"]
169         #Obtencion del mayor en la votacion, opcion correspondiente y envio por MQTT
170         valorComfortMayor = lista_comfort[np.argmax(arr_comfort)]
171         value_change = accionRBC(opcion, valorComfortMayor)
172         client.publish(topic,str(value_change))
173         print("Se envio por mqtt el valor: %d"%value_change)
174         sendToFirestoreRecommendations(
175             value_change, dtm, valorComfortMayor)
176         sendToFirestoreComfort(
177             comfort_cold_rt, comfort_neutral_rt, comfort_warm_rt, dtm)
178
179     # se encera las votaciones
180     comfort.set(
181         {'cold': comfort_cold,
182          'neutral': comfort_neutral,
183          'warm': comfort_warm
184         }
185     )
186     print("Sistemas de votacion en cero nuevamente")
187     time.sleep(900)

```

```
188 except Exception as e:
189     print("Ocurrio un error, pero se restablecio la conexion",e)
```

B Código de control basado en aprendizaje por refuerzo

En esta sección se muestra el código de value iteration y todas las variables adicionalmente que necesita el código para determinar el mejor estado como son: temperatura externa, estado del HVAC, periodo del día y nivel de confort.

Código 2: Código de control térmico basado en value iteration

```
1 import numpy as np
2 from probability import probs as P
3 from rewards import R
4 import random
5 import time
6 from datetime import datetime
7 import firebase_admin
8 from firebase_admin import credentials
9 from firebase_admin import firestore
10 from firebase_admin import db
11 import paho.mqtt.client as mqtt #import the client1
12 import json
13 import requests
14 from common_functions import get_state, get_next_state
15
16 #API para temperatura ext
17 url = "https://api.openweathermap.org/data/2.5/weather"
18
19 querystring = {"lat": "-2.061163", "lon": "-79.905671",
20               "appid": "89fc41f21b8e2b4a7561e2acafa082de",
21               "units": "metric", "lang": "38"}
22
23 headers = {
24     'Cache-Control': 'no-cache'
25 }
26
27
```

```

28 # parametros MQTT
29 broker = '127.0.0.1'
30 port = 1883
31 topic = "integradora/confort"
32 username= "lxxxxxxxxxa"
33 password= "lxxxxxxxxxxx"
34 client_id = f'python-mqtt-{random.randint(0, 1000)}'
35 #client = mqtt.Client(client_id)
36 #client.username_pw_set(username, password)
37 #client.connect(broker,port)
38
39
40 # creacion de la app para firebase
41 cred = credentials.Certificate(
42     'energy-coach-f5270-firebase-adminsdk-o4rki-881d6550e6.json')
43 app = firebase_admin.initialize_app(cred, {
44     'databaseURL': 'https://energy-coach-f5270-default-rtdb.firebaseio.com/'
45 })
46 print(app.name)
47 dbF = firestore.client()
48
49 N = 107
50 S = [*range(0,N+1)]
51 A = [0,1,2]
52 gamma=0.9
53
54
55 def value_iteration(S,A,P,R, gamma):
56     V = {s: 0 for s in S}
57     while True:
58         oldV = V.copy()
59         optimal_policy = {s: 0 for s in S}
60         Qs = []
61         for s in S:
62
63             Q={}
64             for a in A:
65                 Q[a] = R[s][a] + gamma*sum (P[s][a][s_next]* oldV[s_next]
66                 for s_next in S)

```

```

67     Qs.append(Q)
68
69     V[s] = max(Q.values())
70     optimal_policy[s] = max(Q, key=Q.get)
71     print(oldV==V)
72     if all(oldV[s]==V[s] for s in S):
73         break
74     print("matriz Q: ",Q)
75     print("matriz Q Complete", Qs)
76     print("Funcion de valor",V)
77     print("politica optima: ",optimal_policy)
78     return V,optimal_policy
79
80 V,optimal_policy = value_iteration(S,A,P,R,gamma)
81 ref = db.reference("recomendation")
82
83 def sendToFirestoreRecomendationsNState(status_ac,temp_ext,
84     time, comfort, state, new_state):
85     data = {
86         u'date': time,
87         u'status_ac': status_ac,
88         u'temp_ext': temp_ext,
89         u'comfort': comfort,
90         u'state': state,
91         u'new_state': new_state,
92     }
93     dbF.collection(u'historicalCollection').add(data)
94     return None
95
96 def sendToDB(state, action):
97     print("estado y accion:",state,action)
98     code = 0
99     if action == 0:#apagar
100         code = 2
101     if action == 1: #subir
102         if (0 <= state <=8) or (36 <= state <= 44) or (72<= state <=80):
103             print("hola")
104             code = random.randint(9,11) #seteamos a T3
105     elif 9 <= state <=17 or 45 <= state <= 53 or 81<= state <=89:

```

```

106     code = random.randint(6,8) #seteamos a T2
107     elif 18 <= state <=26 or 54 <= state <= 62 or 90<= state <=98:
108         code = random.randint(9,11) #seteamos a T3
109     elif 27 <= state <=35 or 63 <= state <= 71 or 99<= state <=107:
110         code = random.randint(9,11) #seteamos a T3
111     else:
112         print("no entra a los otros")
113         code=2 #apagar
114     if action == 2: #bajar
115         if 0 <= state <=8 or 36 <= state <= 44 or 72<= state <=80:
116             code = 2 #apagar
117         elif 9 <= state <=17 or 45 <= state <= 53 or 81<= state <=89:
118             code = random.randint(3,5) #seteamos a T1
119         elif 18 <= state <=26 or 54 <= state <= 62 or 90<= state <=98:
120             code = random.randint(3,5) #seteamos a T1
121         elif 27 <= state <=35 or 63 <= state <= 71 or 99<= state <=107:
122             code = random.randint(6,8) #seteamos a T2
123         else:
124             code=2 #apagar
125     print("codigo a enviar:",code)
126     ref.update({"rec":code})
127     # enviar al broker con publish
128     client = mqtt.Client(client_id)
129     client.username_pw_set(username, password)
130     client.connect(broker,port)
131     client.publish(topic,str(code))
132     time.sleep(1800)
133     return None
134
135
136 def main():
137     # Menu principal
138     previous_state = 0
139     while 1:
140         try:
141             date_time = firestore.SERVER_TIMESTAMP
142             current_time = datetime.now().hour
143             print("Hora del sistema: ",current_time)
144

```

```

145     comfort_ref = dbF.collection(u'recomendations2')
146     query = comfort_ref.order_by(
147         u'date', direction=firestore.Query.DESCENDING).limit(1)
148     results = query.get()[0].to_dict()['recomendation']
149     comfort=0
150     if(results=="cold"):
151         comfort =0
152     elif(results=="neutral"):
153         comfort=1
154     else:
155         comfort=2
156     print("El codigo comfort es: ",comfort)
157
158     ac_status = db.reference('recomendation/rec').get()
159     print("El estado del aire es:",ac_status)
160
161     response = requests.request(
162         "GET", url, headers=headers, params=querystring)
163     data = json.loads(response.content)
164     temp_ext = data['main']['temp']
165     print("La temperatura externa es:",temp_ext)
166
167     state = get_state(ac_status,comfort,current_time,temp_ext)
168     action= optimal_policy[state]
169     new_state = get_next_state(state,action)
170     print("estado, accion, nexstate", state,action,new_state)
171     sendToFirestoreRecomendationsNState(ac_status,
172         temp_ext, date_time, comfort, state, new_state)
173
174
175     if state == previous_state:
176         print("si es igual")
177         time.sleep(60)
178     elif state == new_state:
179         print("no es ")
180         time.sleep(60)
181     else:
182         sendToDB(state,action)
183     previous_state = state

```

```

184     except Exception as e:
185         print("Ocurrio un error, pero se restablecio la conexion")
186
187
188 if __name__ == "__main__":
189     main()

```

C Código de control de dispositivo Broadlink

Código 3: Código para realizar control del sensor Broadlink

```

1 import broadlink
2 import time
3 import firebase_admin
4 from firebase_admin import credentials
5 from firebase_admin import db
6 from paho.mqtt import client as mqtt_client
7 #import the client1
8 import random
9
10 TURN_OFF= ""
11 TURN_ON = ""
12 TEMP_16 = ""
13 TEMP_17 = ""
14 TEMP_18 = ""
15 TEMP_19 = ""
16 TEMP_20 = ""
17 TEMP_21 = ""
18 TEMP_22 = ""
19 TEMP_23 = ""
20 TEMP_24 = ""
21 #configuraciones del broadlink RM4C mini
22
23 devices= broadlink.discover()
24 while(len(devices)==0):
25     print("No existen dispositivos disponibles")
26     devices = broadlink.discover()

```

```

27 print("Existe dispositivo broadlink")
28 device=devices[0]
29 device.auth()
30
31 # configuraciones del broker mqtt
32
33 broker = '127.0.0.1'
34 port = 1883
35 topic = "integradora/confort"
36 username= "lxxxxxxxa"
37 password= "lxxxxxxxxx"
38 client_id = f"python-mqtt-{random.randint(0,100)}"
39
40 def connect_mqtt():
41     def on_connect(client, userdata, flags, rc):
42         if rc == 0:
43             print("Connected to MQTT Broker!")
44         else:
45             print("Failed to connect, return code %d\n", rc)
46     # Set Connecting Client ID
47     client = mqtt_client.Client(client_id)
48     client.username_pw_set(username, password)
49     client.on_connect = on_connect
50     client.connect(broker, port)
51     return client
52
53 def subscribe(client:mqtt_client):
54     def on_message(client, userdata, msg):
55         option = int (msg.payload.decode())
56         control_AC(option)
57
58
59     client.subscribe(topic)
60     client.on_message = on_message
61
62
63
64
65

```



```
66 def turn_on_AC(device):
67
68     device.send_data(TURN_ON)
69     print("Se ha encendido el aire acondicionado!")
70
71 def turn_off_AC(device):
72
73     device.send_data(TURN_OFF)
74     print("Se ha apagado el aire acondicionado!")
75
76 def temp_16_AC(device):
77
78     device.send_data(TEMP_16)
79     print("Se ha cambiado la temperatura a 16°C")
80
81 def temp_17_AC(device):
82
83     device.send_data(TEMP_17)
84     print("Se ha cambiado la temperatura a 17°C")
85
86 def temp_18_AC(device):
87
88     device.send_data(TEMP_18)
89     print("Se ha cambiado la temperatura a 18°C")
90
91 def temp_19_AC(device):
92
93     device.send_data(TEMP_19)
94     print("Se ha cambiado la temperatura a 19°C")
95
96 def temp_20_AC(device):
97
98     device.send_data(TEMP_20)
99     print("Se ha cambiado la temperatura a 20°C")
100
101 def temp_21_AC(device):
102
103     device.send_data(TEMP_21)
104     print("Se ha cambiado la temperatura a 21°C")
```

```

105
106 def temp_22_AC(device):
107
108     device.send_data(TEMP_22)
109     print("Se ha cambiado la temperatura a 22°C")
110
111 def temp_23_AC(device):
112
113     device.send_data(TEMP_23)
114     print("Se ha cambiado la temperatura a 23°C")
115
116 def temp_24_AC(device):
117
118     device.send_data(TEMP_24)
119     print("Se ha cambiado la temperatura a 24°C")
120
121
122 def control_AC(code: int):
123     ans = True
124
125     if code == 1:
126         #encender aire
127         turn_on_AC(device)
128     if code == 2:
129         #apagar aire
130         turn_off_AC(device)
131     if code == 3:
132         #setear a 16 la temperatura
133         temp_16_AC(device)
134     if code == 4:
135         #setear a 17 la temperatura
136         temp_17_AC(device)
137     if code == 5:
138         #setear a 18 la temperatura
139         temp_18_AC(device)
140     if code == 6:
141         #setear a 19 la temperatura
142         temp_19_AC(device)
143     if code == 7:

```

```
144     #setear a 20 la temperatura
145     temp_20_AC(device)
146     if code == 8:
147         #setear a 21 la temperatura
148         temp_21_AC(device)
149     if code == 9:
150         #setear a 22 temperatura
151         temp_22_AC(device)
152     if code == 10:
153         #setear a 23 la temperatura
154         temp_23_AC(device)
155     if code == 11:
156         #setear a 24 la temperatura
157         temp_24_AC(device)
158     if code == 12:
159         print("Standby")
160
161
162 def run():
163     client = connect_mqtt()
164     subscribe(client)
165     client.loop_forever()
166
167
168
169 if __name__ == '__main__':
170     run()
```

D Encuestas acerca del funcionamiento del control térmico basado en value iteration

¿Cómo se siente en este preciso momento?

16 respuestas

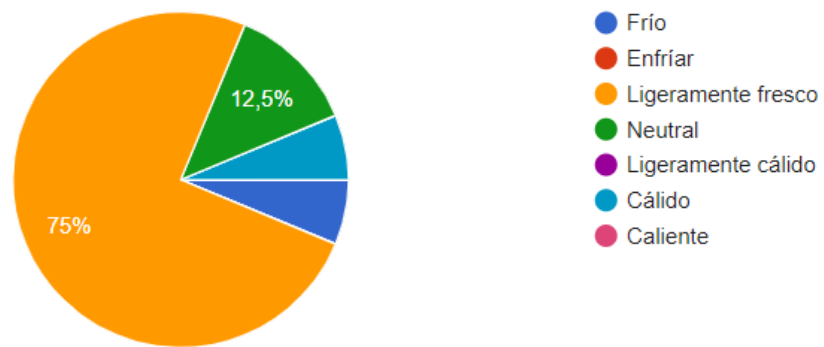


Figura 1: Pregunta 1 de la encuesta

¿Qué tan cómodo con el ambiente se encuentra en este momento?

16 respuestas

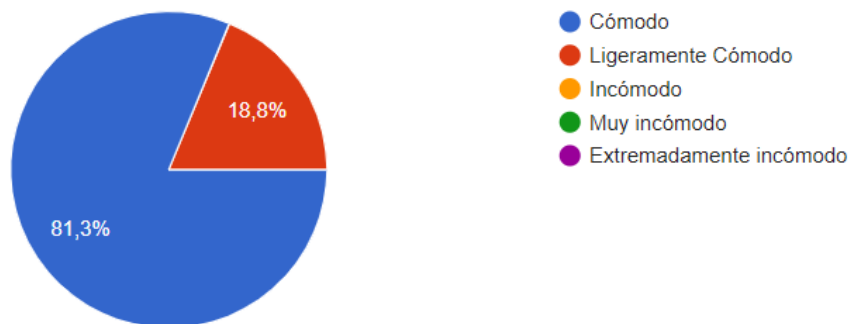


Figura 2: Pregunta 2 de la encuesta

¿Cómo preferiría sentirse usted en este instante?

16 respuestas

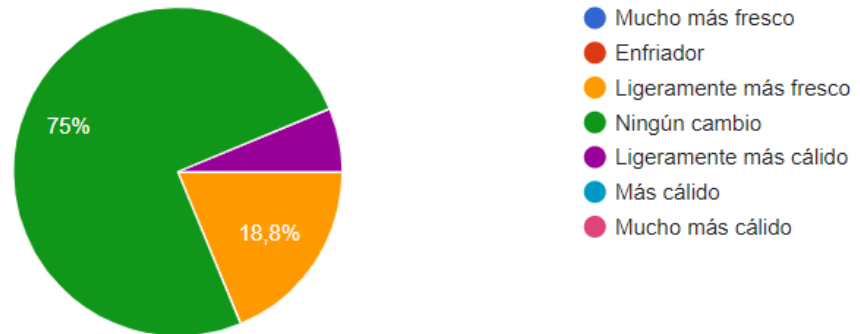


Figura 3: Pregunta 3 de la encuesta

¿Qué tan aceptable es para usted el ambiente térmico en estos momentos?

16 respuestas

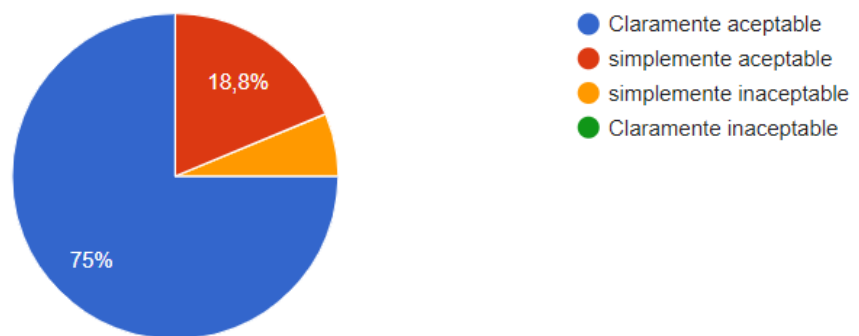


Figura 4: Pregunta 4 de la encuesta

E Matriz de recomendaciones basada en reglas

condiciones	ON	OFF	T n+1	T n+2	T n-1	T n-2	Stand by
temp == 16 & comfort == cold				x			
temp == 16 & comfort == neutral							x
temp == 16 & comfort == hold		x					
temp == 17 & comfort == cold				x			
temp == 17 & comfort == neutral							x
temp == 17 & comfort == hold					x		
temp == 18 & comfort == cold				x			
temp == 18 & comfort == neutral							x
temp == 18 & comfort == hold						x	
temp == 19 & comfort == cold				x			
temp == 19 & comfort == neutral							x
temp == 19 & comfort == hold						x	
temp == 20 & comfort == cold				x			
temp == 20 & comfort == neutral							x
temp == 20 & comfort == hold						x	
temp == 21 & comfort == cold				x			
temp == 21 & comfort == neutral							x
temp == 21 & comfort == hold						x	
temp == 22 & comfort == cold				x			
temp == 22 & comfort == neutral							x
temp == 22 & comfort == hold						x	
temp == 23 & comfort == cold			x				
temp == 23 & comfort == neutral							x
temp == 23 & comfort == hold						x	
temp == 24 & comfort == cold		x					
temp == 24 & comfort == neutral							x
temp == 24 & comfort == hold	x					x	

Figura 5: Matriz de recomendaciones

F Matriz de recomendaciones basada en Aprendizaje por Refuerzo

T_SET	T_OUT	COMFORT	PERIOD OF DAY			REWARDS
			P1	P2	P3	
OFF	T1	Frio	s0	s36	s72	-1
		neutral	s1	s37	s73	1
		calor	s2	s38	s74	0
	T2	frio	s3	s39	s75	1
		neutral	s4	s40	s76	
		calor	s5	s41	s77	-1
	T3	Frio	s6	s42	s78	1
		neutral	s7	s43	s79	
		calor	s8	s44	s80	-1
T1	T1	Frio	s9	s45	s81	-1
		neutral	s10	s46	s82	1
		calor	s11	s47	s83	0
	T2	Frio	s12	s48	s84	-1
		neutral	s13	s49	s85	1
		calor	s14	s50	s86	0
	T3	Frio	s15	s51	s87	0
		neutral	s16	s52	s88	1
		calor	s17	s53	s89	-1
T2	T1	Frio	s18	s54	s90	0
		neutral	s19	s55	s91	1
		calor	s20	s56	s92	0
	T2	Frio	s21	s57	s93	0
		neutral	s22	s58	s94	1
		calor	s23	s59	s95	-1
	T3	Frio	s24	s60	s96	0
		neutral	s25	s61	s97	1
		calor	s26	s62	s98	-1
T3	T1	Frio	s27	s63	s99	0
		neutral	s28	s64	s100	1
		calor	s29	s65	s101	0
	T2	Frio	s30	s66	s102	0
		neutral	s31	s67	s103	1
		calor	s32	s68	s104	-1
	T3	Frio	s33	s69	s105	0
		neutral	s34	s70	s106	1
		calor	s35	s71	s107	-1

Figura 6: Matriz de recomendaciones basada en aprendizaje por refuerzo

G Proyección trimestral del flujo de caja para 2 años

Flujo de caja									
Mes	Saldo Inicial	Unidades vendidas	Ingresos	Total Ingresos	Gastos			Total Egresos	Flujo de Efectivo Neto
					Costos fijos	Costos Variables			
3	-18520	15	4200	4200					-14320
6	-14320	30	8400	8400			2100	6990	-12910
9	-12910	45	12600	12600			4200	9090	-9400
12 (Año 1)	-9400	60	16800	16800			6300	11190	-3790
15	-3790	75	21000	21000	Flujo de Ingresos		8400	13290	3920
18	3920	90	25200	25200			10500	15390	13730
21	13730	105	29400	29400			12600	17490	25640
24 (Año 2)	25640	120	33600	33600			14700	19590	39650
Datos adicionales									
Inversion Inicial			Finacimientto para producir primers 100 unidades						
3520			15000						

Figura 7: Flujo de caja