

# INSTITUTO TECNOLÓGICO DE QUERÉTARO

Desarrollo e instalación de un sistema de control  
automatizado y sustentable para la habitación 354 del  
Hotel Misión Juriquilla

TESIS

Que para obtener el Grado de  
MAESTRO EN INGENIERÍA

presenta

ISAAC GUTIÉRREZ VALLADARES

Dirigido por:

Dr. Raúl Ramírez López

M.C. Agustín Barrera Navarro.

M.C. Omar A. Cervantes Gloria.

# Autorización de impresión de tesis



**EDUCACIÓN**  
SECRETARÍA DE EDUCACIÓN PÚBLICA



Instituto Tecnológico de Querétaro  
División de Posgrado e Investigación

"2019, Año del Caudillo del Sur, Emiliano Zapata"

Querétaro, Qro., **31 de Octubre del 2019**  
DIV. DE EST. POSG. E INV.  
DEPI-241/2019

**GUTIÉRREZ VALLADARES ISAAC**  
**ESTUDIANTE**  
**MAESTRÍA EN INGENIERÍA**  
**PRESENTE**

De acuerdo con el Reglamento para Exámenes Profesionales de la Dirección General de Educación Superior Tecnológica, se le autoriza la impresión de la Tesis, para obtener el Grado de MAESTRIA EN INGENIERÍA, titulada:

**"DESARROLLO E INSTALACIÓN DE UN SISTEMA DE CONTROL AUTOMATIZADO Y SUSTENTABLE PARA LA HABITACION 354 DEL HOTEL MISIÓN JURQUILLA"**

Para el correspondiente Examen de Grado.

**ATENTAMENTE**

*Excelencia en Educación Tecnológica.  
"la tierra será como sean los hombres"*

**MA. DEL CONSUELO ALCÁNTARA TÉLLEZ**  
**JEFA DE LA DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN**



**SEP**

TECNOLÓGICO NACIONAL DE MÉXICO  
INSTITUTO TECNOLÓGICO DE QUERÉTARO  
DIVISIÓN DE ESTUDIOS DE  
POSGRADO E INVESTIGACIÓN

C.C.P., Archivo  
MCAT/mrr\*



Av. Tecnológico s/n esq. Mariano Escobedo, Col. Centro, C.P. 76000, Querétaro, Qro., México

Plantel Centro tel. 01(442) 2274400, Ext. 4421 e-mail: [depin@mail.itq.edu.mx](mailto:depin@mail.itq.edu.mx)

Plantel Norte tel. 01(442) 2435554

[www.tecnm.mx](http://www.tecnm.mx) / [www.itq.edu.mx](http://www.itq.edu.mx)



SNAI  
Reg. No. 3616

## Consentimiento para biblioteca

Santiago de Querétaro, Qro. 22 de noviembre de 2019

El que suscribe, egresado de la MAESTRIA en INGENIERIA; de manera libre y voluntaria autorizo al Centro de información del Instituto Tecnológico de Querétaro difundir la obra de mi autoría con el Título del trabajo: "Desarrollo e instalación de un sistema de control automatizado y sustentable para la habitación 354 del Hotel Misión Juriquilla". Para fines académicos, científicos y tecnológicos, mediante formato CD\_ROM o digital, desde internet, Intranet y en general cualquier formato conocido o por conocer.

Dicha obra estará disponible al estudiantado de esta institución a partir del día 22 de noviembre del 2019, fecha en la cual se puede difundir la obra.

Postulante: ISAAC GUTIÉRREZ VALLADARES

No. de Control: M16141547

Correo electrónico: m16141547@itq.edu.mx

Título de la obra: Desarrollo e instalación de un sistema de control automatizado y sustentable para la habitación 354 del Hotel Misión Juriquilla.

Área del conocimiento: Ingeniería

Palabras clave de la obra: Consumo energético, índice de confort, sustentable, flexible, interfaz.

  
ISAAC GUTIÉRREZ VALLADARES

# Constancia de aprobación de tesis



**SEP**  
SECRETARÍA DE  
EDUCACIÓN PÚBLICA



TECNOLÓGICO NACIONAL DE MÉXICO

Instituto Tecnológico de Querétaro

## DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN COORDINACIÓN DE POSGRADO

### CONSTANCIA DE APROBACIÓN DE TESIS

Santiago de Querétaro, Qro. a 6 de mayo de 2019

**María del Consuelo Alcántara Tellez**  
Encargada de la Jefatura de la División de Estudios de Posgrado e Investigación  
Instituto Tecnológico de Querétaro  
**PRESENTE**

Nos permitimos hacer de su conocimiento, que después de haber procedido a la revisión y evaluación rigurosa y detallada de la Tesis del (de la) C.

Ing. Isaac Gutiérrez Valladares

Cuyo título es: Desarrollo e Instalación de un Sistema de Control Automatizado y Sustentable para la habitación 354 del Hotel Misión Juriquilla

Este jurado considera **concluida** dicha Tesis y se le notifica que el testista puede continuar con los trámites correspondientes para obtener el Grado de Maestría.

Sin más por el momento, nos despedimos de usted.

Atentamente,  
Comité Tutorial

Raúl Ramírez López  
DIRECTOR

Omar Alejandro Cervantes Gloria  
CODIRECTOR(A)

Agustín Barrera Navarro  
ASESOR(A)

c. c. p. Presidente comité tutorial  
Expediente del alumno  
alumno

Recibí  
Coord. Posgrado  
7 mayo - 2019



Av. Tecnológico s/n esq. Mariano Escobedo, Col. Centro, C.P. 76000, Querétaro, Qro., México  
Campus Centro Tel 01(442) 2274400 Fax 01(442)2169931  
Campus Norte 01(442) 2435554  
[www.itq.edu.mx](http://www.itq.edu.mx)



## Resumen

Este proyecto busca desarrollar un sistema domótico sustentable enfocado al Hotel Misión Juruquilla, el sistema pretende reducir el consumo energético y aumentar el índice de confort. Existen sistemas domóticos en el mercado, pero estos no cubren en su totalidad las necesidades del Hotel.

El principal requerimiento para que esta aplicación pueda llegar al mercado es, una interfaz simple para que el huésped pueda controlar los servicios de la habitación, sin hacer grandes modificaciones en la habitación.

El sistema propuesto debe ser económico, fácil de usar y flexible.

**Palabras clave**— Consumo energético, índice de confort, sustentable, flexible, interfaz.

## **Agradecimientos**

A mí familia, docentes, compañeros y amigos GRACIAS...

# Contenido

## Índice

<i>I. INTRODUCCIÓN</i>	12
1.1 Hotel Misión Juriquilla.	12
1.2 Planteamiento del problema	12
1.3 Problemática.	13
1.4 Objetivo General	13
1.5 Objetivos específicos:	13
1.6 Metas:	13
1.7 Justificación	13
1.8 Alcances y limitaciones	15
<i>II. MARCO TEÓRICO</i>	16
2.1 Domótica e inmótica	16
2.2 Medios de transmisión	16
2.2.1 WiFi	16
2.2.2 WPAN	17
2.2.3 IrDA	17
2.2.4 Bluetooth	18
2.2.5 Zigbee	18
2.3 Hardware	18
2.4 PLC	18
2.5 PCB	19
2.6 Sistemas Embebidos	19
2.6.1 Lemaker	19
2.6.2 Raspberry Pi	20
2.6.3 Cubietech	20
2.6.4 Arduino	20
2.6.5 UDOO X86	21
2.7 Interfaz de E/S de Propósito General GPIO	21
2.8 Sistema operativo	21
2.9 Software libre	21
2.10 Linux	22
2.10.1 Debian GNU/Linux	22

2.10.2	RedHat Linux	22
2.10.3	Fedora	22
2.10.4	Raspbian	23
2.10.5	Archlinux	23
2.10.6	Kali Linux	23
2.10.7	Pidora	23
2.10.8	Ubuntu Core	23
2.10.9	RISC OS	23
2.10.10	SARPi	24
2.10.11	FreeBSD	24
2.10.12	RetroPie	24
2.10.13	Android	24
2.11	WINDOWS 10 IoT	24
2.12	FreeRTOS	24
2.13	Lenguajes de control	25
2.13.1	Lenguaje C	25
2.13.2	Delphi	25
2.13.3	ANSI	25
2.13.4	BASIC	25
2.13.5	Python	26
2.13.6	JAVA	26
2.14	Sistemas de Control	26
2.14.1	ON-OFF	26
2.14.2	Proporcional	27
2.14.3	Proporcional + Integral	27
2.14.4	Proporcional + Integral + Derivativo	27
2.14.5	Inteligencia artificial	27
2.14.6	Redes neuronales	27
2.14.7	Red Neuronal Artificial tipo Perceptrón	27
2.14.8	Lógica difusa	27
2.14.9	Algoritmos Genéticos	28
2.15	Internet de las Cosas	28
2.16	Página web responsiva	28



2.17	Gestión energética	28
<i>III. SELECCIÓN</i>		29
3.1.	Selección de comunicación.	29
3.1.1	Calificación para WiFi	29
3.1.2	Calificación para Bluetooth:	29
3.1.3	Calificación para IRDA:	30
3.1.4	Calificación para Zigbee:	30
3.1.5	Resultado	30
3.2	Selección de Hardware	30
3.2.1	Calificación para PLC	31
3.2.2	Calificación para Sistema embebido	31
3.2.3	Calificación para Diseño propio (PCB)	31
3.2.4	Resultado	31
3.3	Comparación de sistemas embebidos	31
3.3.1	Selección del sistema	33
3.3.2	Calificación para Banana Pi M3	33
3.3.3	Calificación para Raspberry Pi 3	33
3.3.4	Calificación para Cubieboard5	34
3.3.5	Calificación para UDOO X86 Advanced Plus	34
3.3.6	Calificación para Arduino	34
3.3.7	Resultado	34
3.4	Selección de sistema operativo.	34
3.4.1	Calificación para Raspbian	37
3.4.2	Calificación para Arch Linux	37
3.4.3	Calificación para Kali Linux	37
3.4.4	Calificación para Pidora	37
3.4.5	Calificación para Windows 10 IoT	38
3.4.6	Calificación para Ubuntu Core	38
3.4.7	Calificación para RISC OS	38
3.4.8	Calificación para SARPi	38
3.4.9	Calificación para FreeBSD	38
3.4.10	Calificación para RetroPie	38
3.4.11	Resultados	38

3.5	Selección de lenguajes de control.	39
3.6	Selección de metodología de control.	39
3.7	Selección sistemas de control para de confort y consumo energético	39
<i>IV. DESARROLLO</i>		41
4.1	Primera etapa. - Desarrollo del prototipo	42
4.2	Segunda etapa. - Mejora del Prototipo	42
4.3	Tercera etapa. - Diseño final	43
4.4	Interfaz	44
4.4.1	Inicio	44
4.4.2	Control de iluminación:	45
4.4.3	Control de televisión	46
4.4.4	Control de aire acondicionado	47
4.4.5	Encuesta	48
4.5	Módulos:	49
4.5.1	Gateway	49
4.5.2	Maestro	50
4.5.3	Esclavo	51
<i>V. RESULTADOS</i>		51
<i>VI. RECOMENDACIONES Y CONCLUSIONES</i>		53
6.1	Recomendaciones	53
6.1.1	Hardware de la unida maestra	53
6.1.2	Software de la unida maestra	54
6.1.3	Hardware de las unidades modulares	54
6.1.4	Software de las unidades modulares	54
6.2	Conclusiones	54
<i>VII. Bibliografía:</i>		54
7.1	Libros	54
7.2	Publicación	55
7.3	WEB	55

## *Índice de Ilustraciones*

Ilustración 1 Modelo SERVQUAL de Calidad de Servicio .....	14
Ilustración 2 Porcentaje y consumo de energía para su uso final en las viviendas,2014 .....	15
Ilustración 3 Primera etapa .....	42
Ilustración 4 Segunda etapa.....	43
Ilustración 5 Tercera etapa .....	44
Ilustración 6 Inicio, pantalla mayor a 720px .....	45
Ilustración 7 Inicio, pantalla menor de 400px.....	45
Ilustración 8 Control de iluminación en pantalla mayor a 720px .....	46
Ilustración 9 Control de iluminación, en pantalla menor de 400px. ....	46
Ilustración 10 Control de televisión en pantalla mayor a 720px.....	47
Ilustración 11 Consola de control de televisión en pantalla menor a 400px.....	47
Ilustración 12 Control de clima en pantalla mayor a 720px. ....	48
Ilustración 13 Control de televisión en pantalla menor a 400px. ....	48
Ilustración 14 Encuesta de confort.....	49

## *Índice de tablas*

Tabla 1 Protocolos IEEE 802.11 .....	17
Tabla 2 Selección de sistemas de comunicación inalámbricas. ....	30
Tabla 3 Selección de hardware. ....	31
Tabla 4 Comparativa de sistemas embebidos.....	33
Tabla 5 Selección de sistemas embebidos.....	34
Tabla 6 Ventajas y desventajas de Raspbian.....	35
Tabla 7 Ventajas y desventajas de Arch Linux. ....	35
Tabla 8 Ventajas y desventajas de Kali Linux. ....	35
Tabla 9 Ventajas y desventajas de Pidora.....	35
Tabla 10 Ventajas y desventajas de Windows 10 IoT.....	36
Tabla 11 Ventajas y desventajas de Ubuntu Core.....	36
Tabla 12 Ventajas y desventajas de RISC OS.....	36
Tabla 13 Ventajas y desventajas de SARPI.....	36
Tabla 14 Ventajas y desventajas de Free BSD.....	36
Tabla 15 Ventajas y desventajas de RetroPie. ....	37
Tabla 16 Selección de sistema operativo.....	38
Tabla 17 - Consumo energético Vs Encuesta de calidad. ....	39
Tabla 18Pseudocodigogateway.....	50
Tabla 19Seudocódigo maestro.....	51
Tabla 20 Seudocódigo esclavo .....	51

# I. INTRODUCCIÓN

## **1.1 Hotel Misión Juriquilla.**

Actualmente, busca actualizar su certificado de calidad al registro ISO9001-2015 el cual tiene como objetivo, el turismo sustentable. Para obtener el registro el hotel tiene que luchar en tres categorías: Sustentabilidad por el medio ambiente, compromiso social y ser de los mejores hoteles en el estado de Querétaro.

Para lograrlo, se intenta educar al huésped sutilmente con hábitos sustentables, colocando letreros con rótulos indicando (favor de apagar las luces antes de salir), entre otras. Estas medidas ahorran energía. Datos obtenidos por la Lic. Ana Laura López Gerente de Calidad

## **1.2 Planteamiento del problema**

El turismo sustentable es entendido como aquel que satisface las necesidades presentes de los turistas, al mismo tiempo que preserva los destinos e incrementa nuevas oportunidades para el futuro. Debe ser concebido de modo que conduzca la gestión de todos los recursos existentes, tanto desde el punto de vista de la satisfacción de las necesidades económicas, sociales y estéticas como del mantenimiento de la integridad cultural, de los procesos ecológicos esenciales, de la diversidad biológica y de los sistemas de soporte de la vida (OMT 1998:21). Por esta razón, el Hotel Misión Juriquilla está buscando en sus habitaciones: Mejorar de forma factible la relación entre el confort, economía, calidad e impacto positivo al medio ambiente.

Las medidas que requiere el hotel se pueden dividir en confort, control de mejora continua, economía y calidad.

En el área de confort, se requiere que el huésped pueda controlar los diferentes servicios de la habitación los cuales son:

Aire acondicionado.

Ventilador de techo.

Televisión.

Iluminación.

Para un control de mejora continua, se requiere que el sistema posea:

- Medición del confort.
- Identificar si la habitación está ocupada o desocupada.
- Una interfaz gráfica compatible con cualquier dispositivo.

Desde el punto económico, se requiere que el sistema:

- Se pueda instalar en la habitación sin hacer grandes modificaciones.
- Los dispositivos instalados no deben ser costosos
- El sistema tiene que ser flexible.

En el área de calidad:

- Los dispositivos instalados deberán contar con certificados de calidad.
- La interfaz tiene que demostrar que es intuitiva y segura.

### 1.3 **Problemática.**

- La habitación 354, destinada para la automatización, no cuenta con una instalación pensada para este nuevo requisito. Los hoteles nuevos o en construcción, planean su automatización desde un inicio bajando los costos de los sistemas, ya que estos pueden ser alámbricos. En la habitación se pueden hacer ranuras e instalar sistemas alámbricos, pero el tiempo y el costo que toma retirar los muebles, ranurar, cablear, pintar, no es algo que el hotel esté dispuesto a permitir. En el proyecto se tendrá que **diseñar e implementar** un sistema que se adapte a estos requisitos.
- No hay claridad sobre la capacidad económica del Hotel para financiar el proyecto. Existe un gran interés por parte de los directivos para su desarrollo. Pero el proyecto necesita ser **viable** y el **costo - beneficio** necesita ser mayor.
- La sustentabilidad del proyecto puede quedar en riesgo al añadir un mayor número de sistemas electrónicos a la habitación, pues el consumo eléctrico tenderá al alza. Por lo tanto, se deberá buscar la **optimización del consumo energético**, sin reducir el **confort** para los huéspedes.

### 1.4 **Objetivo General**

- Diseñar un sistema domótico que mejore la relación entre el **confort** y **consumo energético** en la habitación 354.

### 1.5 **Objetivos específicos:**

- Definir los términos de mejora entre el confort y consumo energético.
- Definir y seleccionar el hardware, software y metodología de instrumentación y control para la automatización de la habitación.
- Construir e instalar el sistema de automatización.

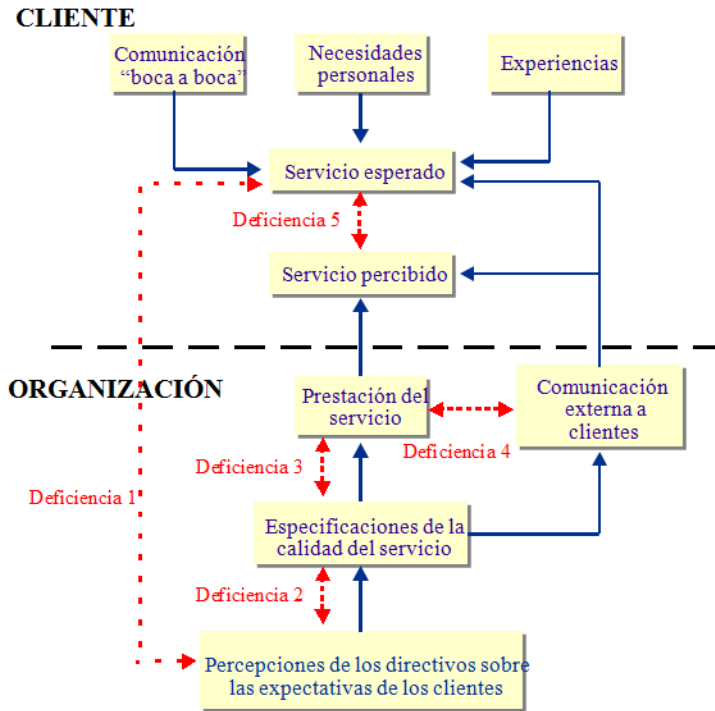
### 1.6 **Metas:**

- Establecer los términos y medidas cualitativas y cuantitativas en Tablas de selección para la mejora de la relación confort - consumo energético.
- Una habitación con el sistema construido e instalado y funcionando para abril del 2018.
- Obtener carta de satisfacción de la gerencia del hotel sobre la mejora de la relación confort - consumo energético.

### 1.7 **Justificación**

- **Al superar las expectativas del huésped en la habitación, se aumentará su satisfacción.**

La Satisfacción del cliente *"es una estrategia de negocios que se enfoca en generar valor para los clientes y en gestionar sus expectativas"*. (Dominici & Guzzo, 2010). Una herramienta de común aceptación en estudios de calidad hotelera es SERVQUAL. Esta permite comprender como los diversos factores afectan la calidad en el servicio de hoteles.



**Ilustración 1 Modelo SERVQUAL de Calidad de Servicio**

Esta define un servicio de calidad como la diferencia entre las expectativas y percepciones de los clientes. De este modo, si las percepciones superaran las expectativas, implicaría una alta satisfacción del huésped ("El Modelo SERVQUAL de Calidad de Servicio - Aiteco Consultores", 2018).

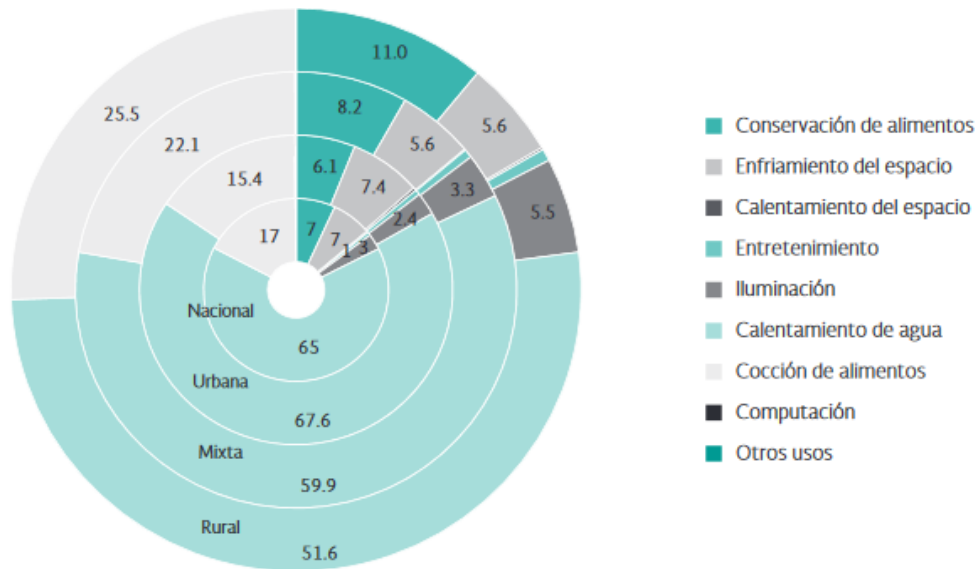
- **El hotel aumentará su competitividad en el mercado y generará una mayor ganancia a los socios.**

Uno de los objetivos de la Secretaría de Turismo (SECTUR) es contribuir en la consolidación y fortalecimiento de los destinos turísticos del país en la búsqueda por elevar la competitividad de la oferta turística nacional orientado al incremento en los flujos de turistas, su estadía y gasto promedio.

Para alcanzar este objetivo se propuso el Modelo de Competitividad de la Empresa Turística Mexicana (MCETM), la cual define a la competitividad como "la capacidad que tiene un hotel para mantener una posición relevante en su contexto de operación y en el mercado, distinguiéndose de sus competidores por la presencia de ventajas competitivas, que surge de una estrategia orientada principalmente a la diferenciación de sus productos turísticos, generando una experiencia única en el turista, de tal forma que es reconocida por sus clientes y sus competidores, quienes llegan a adoptar sus prácticas, y genera utilidades para distribuir entre sus colaboradores y un crecimiento sostenido y sustentable en el largo plazo" (Sector y Universidad Anáhuac, 2014: 28).

- **Al reducir el consumo energético se beneficiará al medio ambiente.**

En los hogares mexicanos, las viviendas ubicadas en localidades urbanas destinaron el 67.6 por ciento de la energía al calentamiento de agua, seguido de la cocción de alimentos (15.4%), enfriamiento del espacio (7.4%) y conservación de alimentos (6.1%). (véase gráfica 1).



**Ilustración 2 Porcentaje y consumo de energía para su uso final en las viviendas,2014 (Estimaciones del CONAPO con base en el INEGI, Encuesta Nacional de Ingresos y Gastos de los Hogares 2014)**

La producción y el consumo de energía son estratégicos para el desarrollo nacional. No obstante, han generado enormes costos ambientales que han detonado y acelerado procesos como el cambio climático, ya que la necesidad de confort de la sociedad, así como el crecimiento y desarrollo económico de las naciones, derivan en mayores niveles de consumo energético (gcec, 2014).

## 1.8 Alcances y limitaciones

### Alcances del Hotel Misión Juriquilla.

- ✓ Hotel con reconocimiento y prestigio en la zona.
- ✓ Cuenta con 192 habitaciones Estándar y 4 suites con servicios y amenidades de primera clase.
- ✓ Aceptabilidad para implementar nuevas tecnologías dentro de sus instalaciones y ampliar su oferta.

### Limitaciones.

- ✓ Aceptación del presupuesto
- ✓ Instalación. No se permite el ranurado de las paredes

### Alcances del dispositivo.

- ✓ Durabilidad.
- ✓ Repetitividad
- ✓ Eficiencia en automatización.

### Limitaciones

- ✓ Calidad
- ✓ Costo

### Alcances en el sector comercial.

- ✓ Contar con un registro o patente del software de automatización.
- ✓ Creación de una EBT para la comercialización del software.
- ✓ Contar con un local comercial.
- ✓ Posicionamiento del producto en el sector hotelero.

### **limitaciones.**

- ✓ Que el producto no cumpla con los requerimientos para ser patentado.
- ✓ Dificultades legales para la creación de una EBT.
- ✓ No contar con permisos legales para la apertura del local comercial.
- ✓ La competencia.

## **II. MARCO TEÓRICO**

Este capítulo es un compendio de datos necesarios para poder entender los conceptos requeridos en la realización del proyecto.

En el capítulo se analizan las tecnologías existentes con las cuales se puede automatizar una habitación de hotel, se analizan, siendo estas: Los medios de transmisión de datos, sistemas electrónicos, sus sistemas operativos, lenguajes de control y sistemas de control.

### **2.1 Domótica e inmótica**

La tecnología encuentra en el hogar un campo de posibilidades apasionantes, y la Domótica es una de ellas. Este término, que se introdujo en España a través de los Pirineos en la década de los noventa, procede del latín domus (casa y hogar) y del griego automática (que actúa autónomamente) (MartínDomínguez&Sáez Vacas, 2006).

Toda esta tipificación en el empleo de la palabra Domótica ha encaminado la Gestión Técnica de Instalaciones hacia las viviendas, pero ha dejado de lado lo referente a edificios de uso no residencial. Por ello ha tomado más fuerza la utilización del término Inmótica, que de forma muy sencilla podemos asociar con "la domótica de los edificios".

El término edificio inteligente se puede entender como la interacción del consumidor o usuario, entonces un edificio inteligente presenta una característica que se interprete como inteligente ya sea anticipar las necesidades del usuario o el manejo eficiente de información (Archila, 2012).

### **2.2 Medios de transmisión**

Las comunicaciones alámbricas transcurren en un medio estable ya sea en cable o fibra óptica. como ventaja principal es que las propiedades del medio están bien definidas, además, no varían en el tiempo. Por otra parte, las tecnologías inalámbricas se utilizan en informática para conectar nodos, equipos o dispositivos sin necesidad de una conexión física. Dicha conexión se da mediante ondas electromagnéticas y la recepción y transmisión a través de puertos.

Las tecnologías inalámbricas brindan grandes ventajas en cuanto a costos y movilidad, ya que se elimina todo el cable y la conexión física entre los dispositivos, además, los usuarios pueden permanecer en movimiento.

Sin embargo, hay que tener en cuenta que para este tipo de tecnologías es necesario establecer una mayor seguridad, si se quiere evitar los intrusos.

#### **2.2.1 WiFi**

El wifi (sustantivo común en español, incluido en el Diccionario de las Academias, proveniente de la marca Wi-Fi) es un mecanismo de conexión de dispositivos electrónicos de forma inalámbrica. Los dispositivos habilitados con wifi (como una computadora personal, un televisor inteligente, una videoconsola, un teléfono inteligente o un reproductor de música) pueden conectarse a internet a través de un punto de acceso de red inalámbrica. Dicho punto de acceso tiene un alcance de unos veinte metros en interiores, distancia que es mayor al aire libre.

Existen diversos tipos de wifi, basado cada uno de ellos en una estándar IEEE 802.11 aprobado. Son los siguientes:



- Los estándares IEEE 802.11b, IEEE 802.11g e IEEE 802.11n disfrutan de una aceptación internacional debido a que la banda de 2,4 GHz está disponible casi universalmente, con una velocidad de hasta 11 Mbit/s, 54 Mbit/s y 300 Mbit/s, respectivamente (Wi-fi.org, 2016).
- La siguiente tabla presenta un resumen conceptual de la familia de estándares IEEE 802.11:

Estándar	Descripción
802.11	Estándar WLAN original. Soporta de 1 a 2 Mbps.
802.11a	Estándar WLAN de alta velocidad en la banda de los 5 GHz. Soporta hasta 54 Mbps.
802.11b	Estándar WLAN para la banda de 2.4 GHz. Soporta 11Mbps.
802.11e	Está dirigido a los requerimientos de calidad de servicio para todas las interfaces IEEE WLAN de radio.
802.11f	Define la comunicación entre puntos de acceso para facilitar redes WLAN de diferentes proveedores.
802.11g	Establece una técnica de modulación adicional para la banda de los 2.4 GHz. Dirigido a proporcionar velocidades de hasta 54 Mbps.
802.11h	Define la administración del espectro de la banda de los 5 GHz para su uso en Europa y en Asia-Pacífico
802.11i	Está dirigido a abatir la vulnerabilidad actual en la seguridad para protocolos de autenticación y de codificación.

**Tabla 1 Protocolos IEEE 802.11**

- Las redes se pueden dividir en LANs, MANs, WANs e interredes, con sus propias características, tecnologías, velocidades y nichos. Las LANs inalámbricas se están haciendo cada vez más comunes en casas y oficinas pequeñas, donde instalar Ethernet se considera muy problemático, así como en oficinas ubicadas en edificios antiguos, cafeterías de empresas, salas de conferencias y otros lugares. Existe un estándar para las LANs inalámbricas, llamado IEEE 802.11, que la mayoría de los sistemas implementa y que se ha extendido ampliamente. Las WANs se extienden por un país o un continente (Tanenbaum et al., 2003).

## 2.2.2 WPAN

Es el nombre de la especificación de un conjunto de protocolos de alto nivel de comunicación inalámbrica para su utilización con radiodifusión digital de bajo consumo, basada en el estándar IEEE 802.15 de redes inalámbricas de área personal (*wireless personal areanetwork*, WPAN). Su objetivo son las aplicaciones que requieren comunicaciones seguras con baja tasa de envío de datos y maximización de la vida útil de sus baterías (Tanenbaum & Wetherall, n.d.).

## 2.2.3 IrDA

Esta tecnología fue lanzada al mercado en el año de 1993. Fue conocida como *Infrared Data Association* (IrDA), “Asociación de Datos Infra-rojos”, define un estándar físico en la forma de transmisión y recepción de datos por rayos infrarrojos. Aunque se ha quedado estancada en los últimos años, puede llegar hasta velocidades de transmisión de 4 Mbps.

Existen varias clases de dispositivos que utilizan esta tecnología y sus principales diferencias radican en la velocidad y la separación que puede haber entre el receptor y transmisor. La distancia típica para las comunicaciones IrDA es de 5 a 60 centímetros de separación entre los transceptores, en el medio del cono.

La comunicación IrDA funciona solamente en modo *half-duplex* debido a que el receptor es cegado por la luz de su transmisor. (Knutson, 2004).

## **2.2.4 Bluetooth**

Bluetooth es una interfaz universal que nos permite la conexión de forma inalámbrica de una serie de dispositivos electrónicos y realizar la transmisión de datos y voz entre ellos, todo esto es factible para transmisiones que requieran poco ancho de banda.

Las principales características de esta forma de comunicación son:

- Opera en la banda libre de los 2,4 GHz por lo que no necesitamos adquirir ninguna licencia de emisión.
- Tiene una capacidad máxima de transmisión de hasta 3 Mbps.
- Implementa diversos mecanismos de ahorro energético de forma que el dispositivo no siempre va a consumir la misma potencia con el consiguiente ahorro energético en la batería del dispositivo.
- Posee un precio económico que permite implementarlo en casi cualquier dispositivo sin encarecerlo desmesuradamente.
- Alcance de hasta 100 metros en función de la potencia de emisión que posea el transmisor Bluetooth ("IEEE 802.15.1", 2018).

## **2.2.5 Zigbee**

Es el nombre de la especificación de un conjunto de protocolos de alto nivel de comunicación inalámbrica para su utilización con radiodifusión digital de bajo consumo, basada en el estándar (IEEE 802.15.4). Tiene una baja velocidad de datos (de hasta 250 kbps).

La Principal función de ZigBeees la de crear una topología de red jerárquica para que un número de dispositivos se comuniquen entre ellos, además de establecer características adicionales de comunicación tales como la autenticación, encriptación, de asociación y en los servicios de aplicación de la capa superior (Merz, Hansemann &Hübner, 2009).

## **2.3 Hardware**

El término hardware o soporte físico es el conjunto de los componentes que integran su parte material; es decir, el conjunto de circuitos electrónicos, cables, armarios, dispositivos electromecánicos y otros elementos que conforman el sistema informático (Torres Cantero, Ruiz and Espinosa, 2013).

## **2.4 PLC**

Según lo define la Asociación Nacional de Fabricantes Eléctricos de los Estados Unidos un PLC – Programable LogicController (Controlador Lógico Programable) es un dispositivo digital electrónico con una memoria programable para el almacenamiento de instrucciones, permitiendo la implementación de funciones específicas como ser: lógicas, secuenciales, temporizadas, de conteo y aritméticas; con el objeto de controlar máquinas y procesos.

Un PLC es modular cuando se puede componer o armar en un bastidor o base demontaje,sobre el cual se instalan la CPU, los módulos de entradas/salidas y losmódulos de comunicaciones si fueran necesarios, entre otros.

Sus principales ventajas son: Su robustez y facilidad de interconexión al proceso y su tendencia no es la de acercarse a los ordenadores, si no de dotarlo de funciones específicas de control y de canales de comunicación para que puedan conectarse entre sí y a los ordenadores, ampliando enormemente las presentaciones del conjunto (Balcells & Romeral, 2003).

## 2.5 PCB

Cuando hablamos de “placa hardware” nos estamos refiriendo en concreto a una PCB (del inglés “printedcircuitboard”, o sea, placa de circuito impreso).

Las PCBs son superficies fabricadas de un material no conductor (normalmente resinas de fibra de vidrio reforzada, cerámica o plástico) sobre las cuales aparecen laminadas (“pegadas”) pistas de material conductor (normalmente cobre). Las PCBs se utilizan para conectar eléctricamente, a través de los caminos conductores, diferentes componentes electrónicos soldados a ella.

Una PCB es la forma más compacta y estable de construir un circuito electrónico (en contraposición a una breadboard, perfboard o similar) pero, al contrario que estas, una vez fabricada, su diseño es bastante difícil de modificar. (Torrente Artero, 2013)

## 2.6 Sistemas Embebidos

El término «sistema embebido» hace referencia a todo circuito electrónico digital capaz de realizar operaciones de computación, generalmente en tiempo real, que sirven para cumplir una tarea específica. Los sistemas embebidos no son equivalentes a los sistemas de cómputo usados en las laptops o en computadoras de escritorio que se venden en las tiendas tecnológicas, ya que los sistemas embebidos suelen tener recursos ilimitados y aplicaciones en múltiples ambientes, como en el campo automotriz (sistemas de inyección de gasolina, alarmas contra robos, control de climatización, sistema de frenado ABS) o en teléfonos móviles, ipad, reproductores Blu-ray, refrigeradoras, alarmas de casas, lavadoras, cámaras fotográficas, instrumentación industrial, equipos médicos, Set Top Boxes, entre otros.

La arquitectura de un sistema embebido contiene un microprocesador dedicado capaz de ejecutar instrucciones a una determinada velocidad, controlada por una señal de reloj. De acuerdo con la arquitectura del microprocesador del sistema embebido, los recursos internos (periféricos) y la máxima frecuencia de operaciones, se define la potencia de procesamiento. Normalmente este parámetro se mide en unidades de MIPS (millones de instrucciones por segundo).

El programa que ejecuta un sistema embebido es por lo general elaborado en lenguajes como el ensamblador, ANSI C, C++ o Basic. Una de las características actuales de todo sistema embebido, y muy importante, es el bajo consumo de energía. Como muchas de sus aplicaciones involucran dispositivos que utilizan baterías, los modos de bajo consumo de corriente (modo sleep) son sumamente importantes para que puedan ser utilizados en este tipo de aplicaciones.

Un microcontrolador es un sistema embebido, y actualmente podemos encontrar microcontroladores en un solo chip, los cuales son sistemas computarizados completos, ya que contienen en su interior un microprocesador, unidades de memoria (de programa y datos), unidades de entrada-salida y periféricos. Hoy en día existen en el mercado microcontroladores de diversas arquitecturas (8 bits, 16 bits y 32 bits, diferentes marcas y modelos dedicados para aplicaciones específicas. (Salas Arriarán, 2015).

### 2.6.1 Lemaker

LeMaker es un equipo entusiasta que se centra en la idea de "opensource, innovación, educación y compartir".

Lemaker hizo una elección de diseño inteligente al lanzar su placa Banana Pi. Utiliza el mismo tamaño de placa y colocación de conectores Como Raspberry Pi, de ahí el nombre Banana Pi (Schinagl, 2014).

Los sistemas operativos disponibles son:

- Bananian
- Raspbian para Banana Pi.
- Scratch para Banana Pi.
- Lubuntu para Banana Pi.
- Android 4.2.2 para Banana Pi.
- openSUSE para Banana Pi.
- Kali Linux para Banana Pi.
- nOS para Banana Pi.
- Kano para Banana Pi.

- Fedora.  
(Banana-pi.org, 2018).

### **2.6.2 Raspberry Pi**

Raspberry Pi es un computador de placa reducida, computador de placa única de bajo coste desarrollado en Reino Unido por la Fundación Raspberry Pi, con el objetivo de estimular la enseñanza de ciencias de la computación en las escuelas

El software es open source, siendo su sistema operativo oficial una versión adaptada de Debian, denominada RaspBian, aunque permite otros sistemas operativos:

- Kali Linux
- Pidora
- Windows 10 IoT Core
- Ubuntu Core
- RISC OS
- SARPi (Slackware ARM for Raspberry Pi)
- Arch Linux ARM
- FreeBSD
- RetroPie

El diseño no incluye un disco duro ni unidad de estado sólido, ya que usa una tarjeta SD para el almacenamiento permanente; tampoco incluye fuente de alimentación, ni carcasa. (Raspberry Pi, 2016)

### **2.6.3 Cubietech**

Cubietech fue formado por los exempleados de Allwinner y fue uno de los primeros Desarrollo disponibles con el SoCAllwinner.

Aunque no es de código abierto el Hardware, que ofrece los esquemas para su descarga. Cubietech lanzó cinco Tableros: el Cubieboard1, el Cubieboard2, Cubieboard3, Cubieboard4 y Cubieboard5

Los sistemas operativos que soporta son:

- Android 4
- Fedora
- Armbian
- Cubian,
- OpenBSD

(Schinagl, 2014).

### **2.6.4 Arduino**

Quando hablamos de “placa Arduino”, deberíamos especificar el modelo concreto, ya que existen varias placas Arduino oficiales, cada una con diferentes características (como el tamaño físico, el número de pines-hembra ofrecidos, el modelo de microcontrolador incorporado –y como consecuencia, entre otras cosas, la cantidad de memoria utilizable, etc.).

Los microcontroladores incorporados en las diferentes placas Arduino pertenecen todos a la misma “familia tecnológica”, por lo que su funcionamiento en realidad es bastante parecido entre sí. En concreto, todos los microcontroladores son de tipo AVR, una arquitectura de microcontroladores desarrollada y fabricada por la marca Atmel (<http://www.atmel.com>).

Los proyectos Arduino pueden ser autónomos o no. En el primer caso, una vez programado el microcontrolador, la placa no necesita estar conectada a ningún computador y puede funcionar autónomamente si dispone de alguna fuente de alimentación. En el segundo caso, la placa debe estar conectada de alguna forma permanente (por cable USB, por cable de red Ethernet, etc.) a un computador ejecutando algún software específico que permita la comunicación entre este y la placa y el intercambio de datos entre ambos dispositivos. Estos comandos se escriben mediante el entorno de desarrollo Arduino (Torrente Artero, 2013).

## 2.6.5 UDOO X86

UDOO X86 es la nueva PC: la placa del fabricante x86 más poderosa y una plataforma compatible con Arduino™ 101, todas integradas en la misma placa.

UDOO X86 puede ejecutar software disponible para PC, desde juegos hasta transmisión de video, desde editores gráficos hasta plataformas de desarrollo profesional, además es compatible con Arduino.

Se basa en procesadores x86 de nueva generación Quad Core de 64 bits fabricados por Intel®, diseñados para el dominio de PC. Procesadores concentrados en 14 nm, con una cantidad de consumo de energía de 5 o 6 watts (UDOO, 2018).

## 2.7 Interfaz de E/S de Propósito General GPIO

Se conoce como *General Purpose Input/Output* (GPIO) o Entrada/Salida de Propósito General. Esta interfaz contiene puertos de señales digitales de entrada y/o salida. Generalmente se trata de un puerto paralelo el cual se puede configurar como entrada o salida. Cuando es configurado como salida, se puede escribir en un determinado registro para que controle la salida de cada pin del puerto. Cuando es configurado como entrada, se puede detectar el estado del pin de entrada leyendo el estado de un registro interno asociado al puerto. (Guerrero, 2007)

## 2.8 Sistema operativo

Es un programa compuesto por una serie de módulos o subprogramas que controlan el funcionamiento global del computador de forma que este sea utilizado por los usuarios de forma eficiente y cómoda. Los subprogramas del sistema operativo se utilizan con un lenguaje específico denominado lenguaje de control. A las instrucciones del lenguaje de control se les suele denominar ordenes o comandos (Torres Cantero, Ruiz and Espinosa, 2013).

## 2.9 Software libre

Aunque en muchas ocasiones se confunde el concepto de software libre con el de software gratuito (en inglés, free tiene los dos significados). Debemos entender como software libre programas de los cuales podemos conseguir su código fuente, estudiarlo, modificarlo y redistribuirlo sin que nos obliguen a pagar por ello. Lo que debemos tener claro es que sí que podemos pedir el dinero que queramos por los programas y su código fuente, el soporte que podemos ofrecer a los usuarios, los libros que vendamos o el material que proporcionemos, tal y como muchas compañías que distribuyen GNU/Linux hacen. Sin embargo, en ningún momento, podemos obligar a que los usuarios no distribuyan el software que les hemos vendido. Éste debe poder ser distribuido de forma libre. Es una forma diferente de entender el software a la que estamos acostumbrados. En muchos de los textos de la FSF (Free Software Foundation) se habla más de filosofía que de ingeniería. Debemos entender todo este movimiento más como una forma de pensar o hacer las cosas que como una compañía más de software. (BaigViñas and Aulí Llinàs, 2003)

Según la Free Software Foundation (<http://www.fsf.org>), organización encargada de fomentar el uso y desarrollo del software libre a nivel mundial, un software para ser considerado libre ha de ofrecer a cualquier persona u organización cuatro libertades básicas e imprescindibles:

- Libertad 0: la libertad de usar el programa con cualquier propósito y en cualquier sistema informático.
- Libertad 1: la libertad de estudiar cómo funciona internamente el programa, y adaptarlo a las necesidades particulares. El acceso al código fuente es un requisito previo para esto.
- Libertad 2: la libertad de distribuir copias.

- Libertad 3: la libertad de mejorar el programa y hacer públicas las mejoras a los demás, de modo que toda la comunidad se beneficie. El acceso al código fuente es un requisito previo para esto.

Un programa es software libre si los usuarios tienen todas estas libertades. Así pues, el software libre es aquel software que da a los usuarios la libertad de poder ejecutarlo, copiarlo y distribuirlo (a cualquiera y a cualquier lugar), estudiarlo, cambiarlo y mejorarlo, sin tener que pedir ni pagar permisos al desarrollador original ni a ninguna otra entidad específica. La distribución de las copias puede ser con o sin modificaciones propias, y atención, puede ser gratis ¡o no!: el "software libre" es un asunto de libertad, no de precio.

(Torrente Artero, 2013)

## **2.10 Linux**

Un profesor de la Universidad de Holanda, Andrew Tanenbaum, decidió escribir un sistema operativo para que sus estudiantes pudieran estudiarlo. Igual que Stallman, hasta el momento había podido utilizar el código fuente del UNIX de AT&T para que sus alumnos aprendieran a diseñar sistemas operativos. Su idea era escribir un sistema operativo que pudiera ser estudiado y modificado por cualquiera que quisiera. En 1987 se puso manos a la obra y llamó a su proyecto mini UNIX, dando lugar a MINIX. Al no utilizar ni una sola línea de código del UNIX de AT&T, no hay ninguna restricción en coger el código, utilizarlo y modificarlo libremente.

Tanenbaum quiso crear un sistema orientado a fines docentes, por lo que lo diseñó utilizando una arquitectura micro-kernel, ideal para una fácil comprensión y aportando una tecnología muy novedosa para la época que le permitía versatilidad, multiplataforma, etc.

La tecnología micro-kernel se basa en dividir las diferentes funcionalidades del núcleo de un sistema operativo en programas totalmente separados y que se comunican entre sí. Esto lo hace muy modular, facilitando muchísimo el test, detección y corrección de errores, mantenimiento, etc.

Existen muchas distribuciones diferentes basadas en GNU/Linux. Las hay para toda clase de ordenadores y dispositivos electrónicos: ordenadores portátiles o de sobremesa, pocketPC o PDA, puntos de acceso de redes wireless, etc. A continuación, haremos una breve descripción de algunas de las distribuciones de GNU/Linux:

### **2.10.1 Debian GNU/Linux**

Una de las primeras distribuciones de GNU/ Linux que aparecieron y aún siguen existiendo y evolucionado. El sistema de paquetes nos permite diferenciar claramente el software libre del que no lo es, permitiéndonos disponer de todo el sistema solamente con programas de licencia Free Software. Está desarrollada por un grupo de colaboradores distribuidos por todo el mundo y no cuenta con el respaldo de ninguna empresa. Aunque es de las más estables y seguras que existen, su sistema de instalación y configuración necesita de conocimientos previos (BaigViñas and Aulí Llinàs, 2003).

### **2.10.2 RedHat Linux**

Junto con SuSE, es una de las distribuciones de mayor popularidad. Está creada por una empresa de EUA, aportando software de gran calidad. Tiene un entorno muy intuitivo que facilita mucho su instalación y configuración (BaigViñas and Aulí Llinàs, 2003)

### **2.10.3 Fedora**

Es una distribución Linux para propósitos generales basada en RPM, que se caracteriza por ser un sistema estable, la cual es mantenida gracias a una comunidad internacional de ingenieros, diseñadores gráficos y usuarios que informan de fallos y prueban nuevas tecnologías. Cuenta con el respaldo y la promoción de Red Hat. (BaigViñas and Aulí Llinàs, 2003)

#### **2.10.4 Raspbian**

Es un sistema operativo libre basado en Debian optimizado para el hardware de Raspberry Pi. Un sistema operativo es el conjunto de programas básicos y utilidades que hacen funcionar tu Raspberry Pi. Sin embargo, Raspbian proporciona más que un sistema operativo puro: viene con más de 35.000 paquetes, software precompilado incluido en un formato agradable para una fácil instalación. (Raspbian.org, 2016)

#### **2.10.5 Archlinux**

Arch Linux ARM es una distribución de Linux para computadoras ARM. Proporciona soporte de software y kernel específico para ARMv5te de flotación suave, ARMv6 y ARMv7 de flotación dura, y conjuntos de instrucciones ARMv8 AArch64 en una variedad de dispositivos de consumo y plataformas de desarrollo. La colaboración con Arch Linux brinda a los usuarios la mejor plataforma, los paquetes más nuevos y el soporte de instalación.

Arch Linux ARM lleva adelante la filosofía Arch Linux de simplicidad y centrado en el usuario, enfocando y acomodando a usuarios competentes de Linux al darles un control completo y responsabilidad sobre el sistema. Se proporcionan instrucciones para ayudar a navegar los matices de la instalación en las distintas plataformas ARM; sin embargo, el sistema en sí ofrecerá poca asistencia al usuario (Archlinuxarm.org, 2018).

#### **2.10.6 Kali Linux**

Kali Linux es una distribución de Linux basada en Debian dirigida a pruebas avanzadas de penetración y auditoría de seguridad. Kali contiene varios cientos de herramientas orientadas a diversas tareas de seguridad de la información, como Pruebas de penetración, Investigación de seguridad, Informática forense e Ingeniería inversa.

La amplia comunidad toma parte en el desarrollo y mantenimiento de la distribución de Linux a través de diversos foros (Docs.kali.org, 2018).

#### **2.10.7 Pidora**

Pidora es una distribución de software de Linux para la computadora Raspberry Pi. Contiene paquetes de programas del proyecto Fedora compilados para la arquitectura ARMv6 usada en Raspberry Pi, lo cuales han sido específicamente reescritos o modificados para esta computadora.

Una de las funciones principales de Pidora es el llamado modo headless (sin monitor), que ofrece la posibilidad de acceder al miniordenador sin necesidad de un monitor o una pantalla.

En caso de problemas con la distribución de Raspberry Pi, existen diferentes puntos de contacto: en los foros de la comunidad y en las wikis de Raspberry Pi y Fedora no solo se ofrece asistencia, sino que también se puede interactuar con otros usuarios (Pidora.ca, 2018).

#### **2.10.8 Ubuntu Core**

El software, basado en Debian y desarrollado desde 2004 por Canonical, destaca en primer lugar por su elevada adaptabilidad y usabilidad. Bajo el nombre Ubuntu Core, los desarrolladores publicaron una versión en 2014 que representa una escisión minimalista de la edición de servidor y puede utilizarse como sistema operativo para Raspberry Pi.

Las principales diferencias de Ubuntu Core con otros sistemas operativos para Raspberry Pi son que cada paquete de software representa una unidad individual ("snap"), incluso el núcleo de Linux. De esta forma, las brechas de seguridad, que se subsanan gracias a las actualizaciones automáticas, solo suelen amenazar a componentes individuales y rara vez a todo el sistema (Ubuntu.com, 2018).

#### **2.10.9 RISC OS**

RISC OS es un sistema operativo de computadora diseñado en Cambridge, Inglaterra por Acorn. Lanzado por primera vez en 1987, sus orígenes se remontan al equipo original que desarrolló el microprocesador Arm. RISC OS es propiedad de CastleTechnology Ltd.

Para aquellos usuarios que no tienen experiencia con los antiguos ordenadores Acorn, el software puede parecer algo inusual a primera vista, pues RISC OS apuesta muy fuerte por la técnica del drag and drop (arrastrar y soltar). Así, los archivos no se abren directamente en un programa, sino solo cuando se arrastran desde el directorio correspondiente a la ventana del programa, sin importar si se trata de un programa de dibujo o de un procesador de texto (Riscosopen.org, 2018).

### **2.10.10 SARPi**

Desarrollado en 1993, Slackware es la distribución de Linux más antigua activa en la actualidad. La portabilidad de ARM, que está disponible bajo el nombre de ARMEdslack y que hoy es conocida como Slackware ARM, es apta para Raspberry Pi como sistema operativo, entre otros.

Para tales fines, el equipo de SARPi ofrece instrucciones paso a paso, así como paquetes de instalación e imágenes para todas las versiones de Raspberry Pi. Los componentes consolidados conceden a la distribución, una estabilidad y seguridad notables. Por lo tanto, el sistema de gestión de paquetes (pkgtool) otorga al administrador un amplio margen de maniobra. Así, las bibliotecas y otras aplicaciones necesarias para la funcionalidad de los programas, no se instalan automáticamente, sino de forma manual (Sarpi.fatdog.eu, 2018).

### **2.10.11 FreeBSD**

Destaca, entre otros factores, por sus funciones de seguridad y almacenamiento y por sus excelentes características de red. Gracias a la compatibilidad con las arquitecturas ARMv6 y ARMv7, BSD puede utilizarse también como sistema operativo para Raspberry Pi (1 y 2). Más adelante, la tercera versión del miniordenador también será compatible con la versión actual de FreeBSD.

Debido a su fortaleza en aspectos como la funcionalidad de red y la estabilidad, FreeBSD se usa principalmente en el ámbito de los servidores. En este sentido, los usuarios se benefician de la rapidez del sistema operativo, que se atribuye principalmente al subsistema de almacenamiento que fue revisado con la versión 10.0. Gracias a su API bien documentada, este sistema operativo para Raspberry Pi se adapta de manera notable a las propias necesidades o puede ampliarse con componentes de software propios (Davis, 2018).

### **2.10.12 RetroPie**

El sistema operativo RetroPie, desarrollado por defecto sobre Raspbian y otros componentes de software, transforma el miniordenador Raspberry Pi en una consola con la que jugar a tus clásicos favoritos de consola o PC. La EmulationStation, escrita en C++, funciona como frontend y su diseño o apariencia puede adaptarse con ayuda de temas predefinidos.

### **2.10.13 Android**

Android es un sistema operativo y una plataforma software, basado en Linux para teléfonos móviles. Además, también usan este sistema algunas marcas de tablets, reproductores de música e incluso PC's. Android permite programar en un entorno de trabajodeJava, aplicaciones sobre una máquina virtual Dalvik (una variación de la máquina de Java con compilación en tiempo de ejecución). Además, lo que le diferencia de otros sistemas operativos, es que cualquier persona que sepa programar puede crear nuevas aplicaciones, o incluso, modificar el propio sistema operativo (Báez and Sanz, 2013).

## **2.11 WINDOWS 10 IoT**

Microsoft lanzó al mercado en 2015 su propio sistema operativo para dispositivos de la Internet de las cosas (Internet ofThings, IoT) como el Raspberry Pi (2 o 3). La aplicación se orienta principalmente a desarrolladores y aficionados a la programación que quieren conectar los aparatos cotidianos con Internet o crear elementos interconectados.

Microsoft se ocupa del desarrollo del software y ofrece tanto varios vídeos educativos como asistencia propia. Además, en el foro de la comunidad puede interactuarse con otros desarrolladores y en él también hay un apartado para Raspberry Pi. Tanto la descarga como la utilización del software para Raspberry Pi son gratuitas, pero no pueden realizarse cambios en el núcleo del sistema (Developer.microsoft.com, 2018).

## **2.12 FreeRTOS**

Se distribuye bajo licencia GPL con una excepción: permite que el código propietario de los usuarios siga siendo código cerrado, manteniendo el núcleo en sí como código abierto, lo que facilita el uso de FreeRTOS en aplicaciones propietarias brindando:

- Poco uso de memoria.
- Ejecución rápida para las tareas.
- Es de código abierto: está bajo continuo desarrollo, y no hay costo de implementación.



(Freertos.org, 2018).

## **2.13                   Lenguajes de control**

Un lenguaje de programación es un conjunto de símbolos y de reglas para combinarlos, que se usan para expresar algoritmos. Los lenguajes de programación, al igual que los lenguajes que usamos para comunicarnos, poseen un léxico (vocabulario o conjunto de símbolos permitidos), una sintaxis, que indica cómo realizar instrucciones de lenguaje, y una semántica, que determina el significado de cada construcción correcta (Torres Cantero, Ruiz and Espinosa, 2013).

### **2.13.1                   Lenguaje C**

Es un lenguaje de programación originalmente desarrollado por Dennis Ritchie entre 1969 y 1972 en los Laboratorios Bell,2 como evolución del anterior lenguaje B, a su vez basado en BCPL.

Al igual que B, es un lenguaje orientado a la implementación de Sistemas Operativos, concretamente Unix. C es apreciado por la eficiencia del código que produce y es el lenguaje de programación más popular para crear software de sistemas, aunque también se utiliza para crear aplicaciones.

Se trata de un lenguaje de tipos de datos estáticos, débilmente tipificado, de medio nivel, ya que dispone de las estructuras típicas de los lenguajes de alto nivel, pero, a su vez, dispone de construcciones del lenguaje que permiten un control a muy bajo nivel. Los compiladores suelen ofrecer extensiones al lenguaje que posibilitan mezclar código en ensamblador con código C o acceder directamente a memoria o dispositivos periféricos. (Kernighan, Ritchie and Padilla, 1985)

### **2.13.2                   Delphi**

Es un lenguaje de programación y kit de desarrollo de software (SDK) para aplicaciones de escritorio, móviles, web y consola. Los compiladores de Delphi utilizan su propio dialogo Pascal de Pascal y generan código nativo para varias plataformas: Windows (x86 y x64), OS X (32 bits solamente), iOS (32 y 64 bits) y Android.

Delphi fue desarrollado originalmente por Borland como una herramienta de desarrollo rápido de aplicaciones para Windows como el sucesor de Turbo Pascal. Delphi agregó orientación completa al lenguaje existente y, desde entonces, el lenguaje ha crecido y soporta muchas otras características del lenguaje moderno, incluyendo genéricos y métodos anónimos, así como características inusuales como tipos de cadenas incorporadas y compatibilidad COM nativa. Delphi y su homólogo de C ++, C ++ Builder, comparten muchos componentes principales, especialmente el IDE, la Biblioteca de componentes visuales (VCL) y gran parte de RTL, y son compatibles entre sí: C ++ Builder 6 y adelante pueden consumir Los archivos de lenguaje Delphi y C ++ en un proyecto y los paquetes compilados con C ++ Builder escritos en C ++ se pueden utilizar desde Delphi. En 2007, los productos fueron lanzados conjuntamente como RAD Studio. RAD Studio es un host compartido para Delphi y C ++ Builder, y se puede comprar con uno o ambos. (Embarcadero.com, 2016)

### **2.13.3                   ANSI**

Una de las ventajas más importantes de los lenguajes de alto nivel es la portabilidad de los ficheros fuente resultantes. Quiere esto decir que un programa desarrollado en un PC podrá ser ejecutado en un Macintosh o en una máquina UNIX, con mínimas modificaciones y una simple recompilación. El lenguaje C, originalmente desarrollado por D. Ritchie en los laboratorios Bell de la AT&T, fue posteriormente estandarizado por un comité del ANSI (American National Standard Institute) con objeto de garantizar su portabilidad entre distintos computadores, dando lugar al ANSI C, que es la variante que actualmente se utiliza casi universalmente (Balagurusamy, 2002).

### **2.13.4                   BASIC**

El Dartmouth BASIC, fue diseñado en 1964 por John George Kemeny y Thomas Eugene Kurtz en el Dartmouth College en Nuevo Hampshire, Estados Unidos, como un medio para facilitar la programación en ordenadores a estudiantes (y profesores) que no fueran de ciencias. De aquella, casi todo el uso de los ordenadores requería codificar software hecho a medida, con lo cual quedaba restringida a personas con formación como científicos y matemáticos. BASIC

originalmente fue desarrollado como una herramienta de enseñanza. El lenguaje y sus variantes llegaron a estar ampliamente disponibles en los microcomputadores a finales de los años 1970 y en los años 1980. BASIC sigue siendo popular a día de hoy en un puñado de dialectos altamente modificados, y en nuevos lenguajes, influenciados por BASIC tales como Microsoft Visual Basic o Gambas en GNU/Linux. (Gottfried and Caro Murillo, 1983)

### **2.13.5 Python**

Python es similar en estilo a pseudocódigo, el lenguaje de programación utilizado para el diseño de software. Es fácil de leer, divertido de aprender y no requiere compilación, es decir se puede ejecutar el script tan pronto como lo haya guardado sin tener que realizar ningún paso intermedio. Python también es un lenguaje útil, ya que puede usarse para tratar una amplia gama de problemas con relativa facilidad, y tiene una buena selección de módulos preconstruidos para hacer frente a muchas de las tareas más comunes de la computadora. Es similar a jugar con bloques de construcción conceptuales: los elementos del lenguaje son bastante simples de entender, pero pueden combinarse en casi cualquier forma que se pueda imaginar para crear estructuras aparentemente complejas. (Hall and Stacey, 2009).

### **2.13.6 JAVA**

A principios de los años noventa, poner inteligencia en los aparatos domésticos se consideraba como el siguiente paso de la tecnología. Algunos ejemplos de enseres domésticos inteligentes incluyen las cafeteras controladas por computadoras, o televisores controlados por un dispositivo programable interactivo.

Anticipándose a un fuerte mercado para tales dispositivos, en 1991, Sun Microsystems fundó un equipo de investigación para trabajar en un "Proyecto verde" secreto, cuya misión era desarrollar software para enseres domésticos inteligentes.

La inteligencia de un aparato doméstico proviene de los chips del procesador que contiene y del software que se ejecuta en los chips de ese procesador. Los chips del procesador del aparato cambian constantemente porque los ingenieros continuamente encuentran formas de hacerlos más pequeños, a menor costo y más poderosos. Para ajustar el nuevo tamaño de los chips, el software que se ejecuta en ellos debe ser extremadamente flexible. Originalmente, Sun planeó la utilización de C para su software de enseres domésticos, pero pronto se dio cuenta de que C no era suficientemente portátil. En lugar de escribir software en C y lidiar con los problemas de portabilidad inherentes a C, Sun decidió desarrollar un nuevo lenguaje de programación para su software de dispositivos del hogar.

El nuevo lenguaje de Sun fue originalmente nombrado Oak (Roble, debido al árbol que se veía desde la ventana del líder de proyecto, James Gosling), pero sucedió que el nombre de Oak ya existía y que era el nombre de otro lenguaje de programación. Lo que sigue en esta historia, es que mientras un grupo de empleados de Sun tomaban un descanso en una tienda de café local, se les ocurrió el nombre de "Java".

Les gustó el nombre de "Java" porque la cafeína tiene un papel significativo en las vidas de los desarrolladores de software. (Dean, n.d.).

## **2.14 Sistemas de Control**

Una forma general de entender a un sistema de control es como un sistema que ante unos objetivos determinados responde con una serie de actuaciones para alcanzarlos (ÑecoGarcía, 2003). Las formas de llegar a estos objetivos se describen a continuación.

### **2.14.1 ON-OFF**

Es un sistema de control de dos posiciones, ya que el elemento de actuación solo tiene dos posiciones fijas que, en muchos casos, son simplemente encendido y apagado. El control de dos posiciones o de

encendido y apagado es relativamente simple y barato, razón por la cual su uso es extendido en sistemas de control tanto industriales como domésticos (Gutiérrez Hinestroza & Iturralde Kure, 2017).

### **2.14.2 Proporcional**

Un controlador con acción de control proporcional mejora la relación entre la salida del controlador y la salida esperada. La diferencia entre el punto de control y el valor de estado estacionario de la variable que se controla, se le llama desviación (Gutiérrez Hinestroza & Iturralde Kure, 2017).

### **2.14.3 Proporcional + Integral**

La mayoría de los procesos no se pueden controlar con una desviación, es decir, se deben controlar en el punto de control, y en estos casos se debe añadir inteligencia al controlador proporcional, para eliminar la desviación. Esta nueva inteligencia o nuevo modo de control es la acción integral o de reajuste y, en consecuencia, el controlador se convierte en un controlador proporcional-integral (pi) (Gutiérrez Hinestroza & Iturralde Kure, 2017).

### **2.14.4 Proporcional + Integral + Derivativo**

Algunas veces se añade otro modo de control al controlador pi, este nuevo modo de control es la acción derivativa, que también se conoce como rapidez de derivación o preactuación; tiene como propósito anticipar hacia dónde va el proceso, mediante la observación de la rapidez para el cambio del error, su derivada (Gutiérrez Hinestroza & Iturralde Kure, 2017).

### **2.14.5 Inteligencia artificial**

Los métodos de inteligencia artificial (IA) son una respuesta al deseo de aproximar el comportamiento y el pensamiento humano a diversos sistemas para la solución de determinadas problemáticas. Por ello, no es de sorprender que actualmente se tiene sistema+s muy avanzados que pueden emular ciertas características humanas, sin embargo, aún nos encontramos muy lejos de poder recrear algunas otras.

En la actualidad los métodos de la inteligencia artificial (IA) tienen un gran auge y muchos investigadores se encuentran estudiando nuevas alternativas en el área. Hoy en día es común el empleo de sistemas que utilizan la IA para su funcionamiento cotidiano, entre ellos los equipos electrodomésticos como lavadoras, hornos de microondas, cámaras de video, e inclusive sistemas de transporte (Ponce-Cruz, 2010).

### **2.14.6 Redes neuronales**

La Redes Neuronales Artificiales son herramientas informáticas que se asemejan a redes neuronales biológicas. Las RNA se representan por la suma de entradas (externas  $z_j$ ), seguida de una función de activación  $f(z_j)$ , esta función utiliza la suma de los estímulos o entradas para determinar la actividad de la salida de la neurona.

### **2.14.7 Red Neuronal Artificial tipo Perceptrón**

La neurona tipo perceptrón suma las entradas y se multiplican por los pesos, la fase en la que aprende y hace el reconocimiento compara el patrón de entrada con un patrón preestablecido por nosotros (Ponce Cruz and Herrera, 2010).

### **2.14.8 Lógica difusa**

La lógica difusa hoy en día es muy común y se halla en diferentes sectores de la tecnología, tanto en la electrónica como el control, las matemáticas, la robótica, etc. El objetivo principal de la lógica difusa es crear un sistema basado en el comportamiento y pensamiento humanos. Esto se logra gracias al planteamiento de un modelo en cualquier contexto y traducirlo a reglas gramaticales o lenguaje humano. (Ponce Cruz and Herrera, 2010)

La lógica difusa es un conjunto de principios matemáticos basados en grados de membrecía o pertenencia, cuya función es modelar información. Este modelado se hace con base en reglas lingüísticas que aproximan una función mediante la relación de entradas y salidas del sistema

(composición). Esta lógica presenta rangos de membresía dentro de un intervalo entre 0 y 1, a diferencia de la lógica convencional, en la que el rango se limita a dos valores: el cero o el uno.

La lógica difusa utiliza enunciados que no son ni totalmente ciertos ni completamente falsos y se aplica a expresiones que pueden tomar un valor de veracidad de todo un conjunto de valores comprendido entre dos extremos, la verdad absoluta y la falsedad total. De esta manera los conjuntos difusos se constituyen en la generalización de los conjuntos clásicos, que consideran únicamente la pertenencia o no pertenencia de un elemento a un determinado conjunto, a diferencia de los difusos, en los cuales, un elemento del conjunto presenta cierto grado de pertenencia y toma un determinado valor entre 0 y 1. (Ponce-Cruz, 2010)

### **2.14.9 Algoritmos Genéticos**

Los Algoritmos Genéticos (AGs) son métodos adaptativos que pueden usarse para resolver problemas de búsqueda y optimización. Están basados en el proceso genético de los organismos vivos. A lo largo de las generaciones, las poblaciones evolucionan en la naturaleza de acorde con los principios de la selección natural y la supervivencia de los más fuertes, postulados por Darwin (1859). Por imitación de este proceso, los Algoritmos Genéticos son capaces de ir creando soluciones para problemas del mundo real. La evolución de dichas soluciones hacia valores óptimos del problema depende en buena medida de una adecuada codificación de estas.

Los Algoritmos Genéticos usan una análoga directa con el comportamiento natural. Trabajan con una población de individuos, cada uno de los cuales representa una solución factible a un problema dado. A cada individuo se le asigna un valor o puntuación, relacionado con la bondad de dicha solución. En la naturaleza esto equivaldría al grado de efectividad de un organismo para competir por unos determinados recursos. Cuanto mayor sea la adaptación de un individuo al problema, mayor será la probabilidad de que el mismo sea seleccionado para reproducirse, cruzando su material genético con otro individuo seleccionado de igual forma. Este cruce producirá nuevos individuos – descendientes de los anteriores – los cuales comparten algunas de las características de sus padres. Cuanto menor sea la adaptación de un individuo, menor será la probabilidad de que dicho individuo sea seleccionado para la reproducción, y por tanto de que su material genético se propague en sucesivas generaciones. (Ponce-Cruz, 2010).

### **2.15 Internet de las Cosas**

El concepto "Internet de las Cosas" (IoT) apareció por primera vez en 1999, cuando el empresario inglés Kevin Ashton lo usó para describirlo como la red de objetos físicos que están embebidos con la tecnología electrónica, software, sensores y conexiones. Estas conexiones proporcionan un nivel avanzado de conectividad entre dispositivos y sistemas que consiguen un mayor valor intercambiando datos con otros dispositivos, fabricantes u operadores.

Una gran variedad de dispositivos compone la capa física de IoT, desde los más simples hasta los más complejos. Los implantes de monitorización o los transpondedores de biochip para granjas animales son ejemplos de dispositivos integrados compuestos, que trabajan recopilando datos y enviándolos para procesarlo posteriormente.

Este tipo de dispositivos llevan a las computadoras de uso general, modificando el hardware y software para adaptar su funcionalidad a un propósito particular (Waher, n.d.).

### **2.16 Pagina web responsiva**

El sitio web debe tener la tecnología para responder automáticamente al contexto del usuario. Una razón convincente para el uso de página web responsiva, es que se crea un sitio web que no sólo se ve bien y funciona correctamente en los dispositivos que están en el mercado, sino que probablemente se verá bien y funcionará en los nuevos dispositivos que estarán disponibles en el futuro. (Peterson, 2014).

### **2.17 Gestión energética**

La gestión de la energía es uno de los principales propósitos de la automatización del hogar, junto con la seguridad y la vigilancia. La gestión de la energía significa ahorrar y utilizar la energía eficientemente en las zonas residenciales para que pueda reducir el consumo de electricidad y crear lo que actualmente se conoce como una casa verde (Dundar, 2015).

## III. SELECCIÓN

En este capítulo se presenta un método que permite seleccionar de forma analítica la solución propuesta, a partir de una serie de opciones factibles. Dicha metodología pertenece al área de la ingeniería en toma de decisiones. Se auxilia de tablas cuantitativas que evalúan el desempeño de las diferentes propuestas, bajo criterios definidos en cada caso, y asignando un peso a cada variable relevante para el sistema, siendo -2 la más baja, 2 la más alta, y usando números enteros para dicha evaluación.

### **3.1. Selección de comunicación.**

Existen dos tipos de comunicación que se pueden utilizar: alámbricas e inalámbricas. Como el hotel ya está construido y operando, no es posible utilizar como medio de comunicación un medio de transmisión alámbrico.

Para seleccionar el mejor medio de comunicación inalámbrico los puntos a considerar son:

- **Velocidad:** Como la velocidad para la aplicación no es tan relevante para el funcionamiento del sistema se le asigna un peso del 10%.
- **Seguridad:** La seguridad es un factor clave para este sistema por lo que se aplica un peso del 35%.
- **Nodos:** El número de nodos que debe soportar la red es una característica importante por lo que tiene un peso del 30%, ya que si el hotel desea aumentar el número de habitaciones o aplicaciones la red será más robusta.
- **Consumo energético:** Es importante disminuir el consumo energético por lo que se le asignó un peso del 5%.
- **Instalación:** La instalación es uno de los mayores problemas al sugerir un tipo de red. Esto ocasiona que tenga un peso del 20%.

#### **3.1.1 Calificación para WiFi**

- **Velocidad = 2.** Tiene una tasa de transmisión de datos que llega hasta los 300Mb/s.
- **Seguridad = 2.** Cuenta con el protocolo de seguridad WPA2 que permite la encriptación de datos.
- **Nodos = 2.** Permite diferentes topologías de red y el número de nodos es ilimitado.
- **Consumo = -2.** La energía que consume el sistema de comunicación wifi varía dependiendo de los sistemas (Access Point, Router, Extensores) y al requerir de tantos sistemas para su funcionamiento, su consumo es el más alto.
- **Instalación = -1.** Como se mencionó en el punto anterior un sistema de comunicación Wifi requiere varios sistemas.

#### **3.1.2 Calificación para Bluetooth:**

- **Velocidad = 0.** Tiene una tasa de transmisión de datos que llega hasta los 24Mb/s.
- **Seguridad = 1.** Gracias al emparejamiento se evita que intermediarios modifiquen la comunicación. Pero no que evita que puedan ver lo enviado.
- **Nodos = -1.** El número de nodos es ilimitado y para la creación de una red se requieren sistemas avanzados.

- **Consumo = -1.** Aunque el consumo energético en los sistemas bluetooth ha mejorado el consumo tiene que ser alto ya que las Antenas RN45 son punto a punto y no punto multipunto por lo que se tienen que tener varias antenas.
- **Instalación = 0.** Como se mencionó en el punto anterior el sistema de comunicación Bluetooth requiere varias antenas.

### 3.1.3 Calificación para IRDA:

- **Velocidad = -2.** Tiene una tasa de transmisión de datos llega hasta los 4Mb/s.
- **Seguridad = -2.** No existe seguridad en la transmisión de datos.
- **Nodos = -2.** Solo funciona en Half- Duplex.
- **Consumo = 2.** El consumo energético es 20 y por IRled y solo lo consume cada que envía y recibe datos.
- **Instalación = -2.** La comunicación tiene un alcance de 5 a 60cm haciendo muy difícil su instalación.

### 3.1.4 Calificación para Zigbee:

- **Velocidad = -1.** Tiene una tasa de transmisión de datos de 250 kb/s
- **Seguridad = 0.** Permite la encriptación de datos, pero tiene el canal es constante.
- **Nodos = 2.** Permite la comunicación entre nodos por multi-punto.
- **Consumo = 2.** Su consumo es muy pequeño permitiendo a baterías operar durante años.
- **Instalación = 1.** La comunicación tiene un alcance de 10m por lo que para fortalecer la red es necesario colocar un mayor número de dispositivos.

### 3.1.5 Resultado

Factor	Peso	WiFi	Bluetooth	IRDA	Zigbee
Velocidad	0,1	2	0	-2	-1
Seguridad	0,35	2	1	-2	0
Número de Nodos	0,3	2	-1	-2	2
Consumo	0,05	-2	1	2	2
Instalación	0,2	-1	0	-2	1
	1	1,2	0,1	-1,8	0,8

Tabla 2 Selección de sistemas de comunicación inalámbricas.

Como se puede notar de la tabla anterior se obtuvo que en primer lugar las comunicaciones WiFi son lo mejor para el sistema en segundo lugar las redes Zigbee.

## 3.2 Selección de Hardware

En cuanto al hardware se pueden utilizar las siguientes opciones: PLC, sistemas embebidos o fabricar un diseño propio. Los puntos para determinar la mejor opción son:

- **Robustez.** Se refiere a la capacidad del hardware para soportar ruidos electromagnéticos, su peso es del 35%.
- **Instalación.** Debido a las limitaciones este aspecto se considera de gran importancia otorgándole un 45% en el peso.

- **Calidad.** Este factor es importante ya que todo el sistema estropearse de no tenerlo en cuenta su factor de peso es del 20%.

### 3.2.1 Calificación para PLC

- **Robustez = 2.** Los PLC están diseñados para la industria por lo que desde su diseño se cumple con esta especificación.
- **Instalación = -2.** Los PLC compactos tienen la desventaja de sólo poder ampliarse con muy pocos módulos.
- **Calidad = 2.** Los fabricantes de PLC realizan pruebas de calidad a sus productos.

### 3.2.2 Calificación para Sistema embebido

- **Robustez = 0.** La robustez de un sistema embebido depende de las protecciones que se le instalen, ya sea en: Gabinete, protecciones eléctricas y filtros.
- **Instalación = 2.** Debido a su compacto tamaño, variedad de puertos de entrada y salida y facilidad para integrarlo a otros sistemas.
- **Calidad = 2.** Los sistemas embebidos cuentan con certificados de calidad.

### 3.2.3 Calificación para Diseño propio (PCB)

- **Robustez = 0.** La robustez de un diseño propio depende de las protecciones que el mismo diseño determine necesarias para la aplicación.
- **Instalación = 2.** Al ser un sistema diseñado para una aplicación específica su instalación deberá de cumplir el requisito que la aplicación requiera.
- **Calidad = -2.** Obtener las certificaciones de calidad para el proyecto no resulta costeable.

### 3.2.4 Resultado

Factor	Peso	PLC	Embebido	Diseño propio
Robustez	0,35	2	0	0
Instalación	0,45	-2	2	2
Calidad	0,2	2	2	-2
	1	0,2	1,3	0,05

Tabla 3 Selección de hardware.

De la tabla anterior podemos observar que los sistemas que convienen para este proyecto son los embebidos, ya que tienen calidad, y su instalación es sencilla. Sin embargo, se tiene que considerar que la robustez del hardware dependerá del diseño final.

### 3.3 Comparación de sistemas embebidos

A continuación, se realiza una comparativa de los sistemas embebidos más relevantes, los cuales son: Banana Pi M3, Raspberry Pi 3 ModelB, Cubieboard5 y UD00 X86 Advanced Plus.

	Banana Pi M3	Raspberry Pi 3 Model B	Cubieboard 5	UDOO X86 Advanced Plus
<b>SoC</b>	AllWinner A83T	Broadcom BCM2837	AllWinner H8	Intel Celeron N3160
CPU	ARM Cortex-A7 (32-bit)	ARM Cortex-A53 (64-bit)	ARM Cortex-A7 (32-bit)	Braswell (1.60GHz with 2.24GHz burst) (64-bit)
	2GHz octacore	1.2GHz quadcore	2GHz octacore	2.24GHz quadcore
<b>Procesador de Gráficos</b>	PowerVR SGX544MP1	VideoCore IV	PowerVR SGX544	640MHz Intel HD Graphics Gen 8-LP
Espacio en RAM	2GB	1GB	2GB	4GB
<b>Espacio interno</b>	8GB	No	8GB	32GB
SD card	MicroSD card slot	MicroSD card slot	MicroSD card slot	MicroSD card slot
	Max. size: 64GB	Max. size: 32GB	Max. size: 32GB	Max. size: 32GB
<b>PuertosUSB</b>	2	4	1	3
USB versión	2.0	2.0	2.0	3.0
<b>Ethernet</b>	1 port	1 port	1 port	1 port
	<b>Type: Gigabit</b>	<b>Type: 10/100</b>	<b>Type: Gigabit</b>	<b>Type: Gigabit</b>
HDMI	Si	Si	Si	Si
<b>VGA</b>	No	No	No	No
Interfaz de Video	MIPI DSI	MIPI DSI	No	No
<b>Audio output</b>	Si	Si	Si	Si
Conexión de Audio	3.5mm Jack	3.5mm jack	3.5mm Jack	3.5mm Jack
<b>HDMI audio</b>	Si	Si	Si	Si
I2S	Si	Si	Si	No
<b>Entrada de Micrófono</b>	No	No	Si	Si
Micrófono integrado	Si	No	Si	No
<b>Puerto SATA</b>	Si	No	Si	Si
Sensor IR	Si	No	Si	Si
<b>Wi-Fi</b>	802.11 b/g/n (AP6212)	802.11b/g/n	802.11 b/g/n (dual band)	OptionalWi-Fi 802.11ac
Bluetooth	Bluetooth 4.0	Bluetooth 4.1	Bluetooth 4.0	Bluetooth BLE
<b>GPIO Pines</b>	40	40	70	14
PWM pines	1	1	1	4
<b>ADC pines</b>	No	No	Si	Si
I2C	1	1	1	2



SPI	1	1	1	1
UART	2	1	1	2
Arduino pinout	No	No	No	Si
Voltaje	5V	4.8V - 5.2V	5V	3.3V
Consumo energético	1A - 2 <sup>a</sup>	600mA - 1.6A	2A - 3 <sup>a</sup>	1A - 3 <sup>a</sup>
Soporta Windows	No	Windows 10 IoT	No	Windows 7; Windows 8.1; Windows 10
Soporta Android	Android 5.1	No	Android 4.4.4	Si
Tamaño (mm)	92 x 60 x 17	85 x 56 x 20	110 x 80 x 14	120 x 85 x 25
Precio USD	\$74	\$35	\$99	\$165

Tabla 4 Comparativa de sistemas embebidos.

### 3.3.1 Selección del sistema

Como se puede observar en la tabla anterior, todos los sistemas embebidos comparados son muy similares, ya que no tienen microcontroladores como los sistemas Arduino, siendo estos mucho más simples, pero se van a considerar en la tabla final puesto que su rango de aplicación es muy extenso. Los puntos destacados son los siguientes:

- **Información:** La cantidad de información disponible es muy importante ya que se requiere cumplir con este proyecto en un plazo de tiempo establecido, por lo que al tener una mayor cantidad de información permite terminar en el plazo estipulado. Peso 40%.
- **Sistema operativo:** La cantidad de sistemas operativos que se pueden instalar al sistema, son importantes ya que da una mayor flexibilidad al sistema final. Peso 35%.
- **Precio:** Uno de los objetivos de este proyecto es que el sistema sea económico. Peso 15%.
- **Consumo energético:** Con la finalidad de obtener un producto final sustentable, es necesario considerar este punto con un peso del 10%. No se otorga un peso mayor, pues éste consumo representa unos cuantos watts en el día, lo que es, para fines prácticos, irrelevante.

### 3.3.2 Calificación para Banana Pi M3

- **Información = -1.** Al Realizar una búsqueda de libros se encontraron dos: Learning Banana Pi de [Dan Blair](#) y Schnelleinstieg Banana Pi de [Mattias Schlenker](#).
- **Sistema Operativo = 1.** Soporta nueve sistemas operativos, véase marco teórico.
- **Consumo energético = 1.** Consume de 1A - 2A dependiendo de la tarea ejecutada.
- **Precio = 1.** Tiene un costo de \$74 USD.

### 3.3.3 Calificación para Raspberry Pi 3

- **Información = 2.** Se encontraron más de 50 libros.

- **Sistema Operativo = 1.** Soporta más de 10 sistemas operativos.
- **Consumo energético = 2.** Consumo de 600mA - 1.6A dependiendo de la tarea ejecutada.
- **Precio = 2.** Tiene un costo de \$35 USD.

### 3.3.4 Calificación para Cubieboard5

- **Información = -1.** Se encontró solo un libro titulado, GettingStartedwithCubieboardde [Olliver M. Schinagl](#).
- **Sistema Operativo = 1.** Soporta cinco sistemas operativos, pero con Cubian es posible realizar el proyecto.
- **Consumo energético = -2.** Tiene un consumo alto de 2A - 3A dependiendo la ejecución.
- **Precio = 0.** Tiene un costo de \$99 USD.

### 3.3.5 Calificación para UDOO X86 Advanced Plus

- **Información = 2.** Se encontró un libro titulado, GettingStartedwith UDOO de EmanuelePalazzetti, pero al soportar Windows la cantidad de información es muy grande.
- **Sistema Operativo = 2.** Al tener un procesador tan robusto soporta varias versiones de Windows.
- **Consumo energético = -2.** Tiene un consumo alto de 1A - 3A dependiendo la ejecución.
- **Precio = -2.** Tiene un costo de \$165 USD.

### 3.3.6 Calificación para Arduino

- **Información = 2.** Existe mucha información, proyectos y más de 50 libros.
- **Sistema Operativo = -1.** Soporta FreeRTOS.
- **Consumo energético = 2.** Tiene un consumo promedio de 500 mA.
- **Precio = 2.** Tiene un costo mínimo de \$5 USD.

### 3.3.7 Resultado

Factor	Peso	Banana Pi	RaspberryPi 3	Cubieboard5	UDOO X86	Arduino
Información	0,4	-1	2	-1	2	2
Sistema operativo	0,35	1	1	1	1	-1
Consumo	0,1	1	2	-2	-2	2
Precio	0,15	1	2	0	-2	2
	1	0,2	1,65	-0,25	0,65	0,95

Tabla 5 Selección de sistemas embebidos.

De la tabla anterior se observa que el mejor dispositivo a seleccionar es la Raspberry Pi 3 y en segundo lugar para las aplicaciones Arduino, ya que es económico, versátil y existe una cantidad muy grande de información.

### 3.4 Selección de sistema operativo.

Los sistemas operativos que Raspberry pi 3, permite instalar son:

- *Raspbian*
- *Arch Linux*

- *Kali Linux*
- *Pidora*
- *Windows 10 IoT Core*
- *Ubuntu Core*
- *RISC OS*
- *SARPi*
- *FreeBSD*
- *RetroPie*

A continuación, se analizarán las ventajas y desventajas de cada uno de ellos:

Raspbian	
Ventajas	Inconvenientes
Fácil de usar (adecuado para inexpertos)	Actualizaciones de software con algo de retraso
Excelente soporte al usuario	

**Tabla 6 Ventajas y desventajas de Raspbian.**

Arch Linux	
Ventajas	Inconvenientes
Actualizaciones seguras y rápidas del paquete de software	Configuración compleja
Control total del sistema	Riesgo de paquetes inestables debido a períodos de prueba muy breves

**Tabla 7 Ventajas y desventajas de Arch Linux.**

Kali Linux	
Ventajas	Inconvenientes
Diversas herramientas de seguridad integradas	No apto para inexpertos en Linux
Permite realizar tests de seguridad en ordenadores y redes	Requisitos de RAM muy elevados

**Tabla 8 Ventajas y desventajas de Kali Linux.**

Pidora	
Ventajas	Inconvenientes
El modo headless permite configurar los dispositivos sin pantalla o monitor	No apto para inexpertos
Especialmente concebido para Raspberry Pi	Necesita muchos recursos

**Tabla 9 Ventajas y desventajas de Pidora.**

Windows 10 IoT Core	
Ventajas	Inconvenientes
Conexión fluida de dispositivos IoT	Propietario
Asistencia especial sobre Raspberry Pi	Se necesita la versión actual de Windows 10

**Tabla 10 Ventajas y desventajas de Windows 10 IoT.**

Ubuntu Core kad	
Ventajas	Inconvenientes
Actualizaciones de software regulares	Los programas requieren más espacio que los paquetes de software clásicos, ya que deben guardarse en varias bibliotecas.
Función de regreso en todas las actualizaciones	

**Tabla 11 Ventajas y desventajas de Ubuntu Core.**

RISC OS	
Ventajas	Inconvenientes
Núcleo de sistema muy ligero	Número limitado de desarrolladores activos
Se amplía fácilmente gracias a su estructura modular	

**Tabla 12 Ventajas y desventajas de RISC OS.**

SARPI	
Ventajas	Inconvenientes
Actualizaciones seguras y rápidas del paquete de software	Configuración compleja
Control total del sistema	Riesgo de paquetes inestables debido a períodos de prueba muy breves

**Tabla 13 Ventajas y desventajas de SARPI.**

FreeBSD	
Ventajas	Inconvenientes
Muy rápido, estable y con gran cantidad de recursos	La versión actual no está disponible para Raspberry Pi 3
Comunidad amplia y activa	

**Tabla 14 Ventajas y desventajas de Free BSD.**

RetroPie	
Ventajas	Inconvenientes

Rápida reacción a nuevas ediciones de Raspberry Pi	Documentación insuficiente sobre cada uno de los componentes de software
Centro multimedia (Kodi) integrado	El uso del controlador se puede ampliar

**Tabla 15 Ventajas y desventajas de RetroPie.**

Una vez analizados los diferentes sistemas operativos, tanto las ventajas e inconvenientes de cada uno, se les asignan a los factores más importantes un peso dependiendo su relevancia para el proyecto.

- **Documentación:** El sistema operativo debe contar con un buen respaldo para la conclusión del proyecto. Peso 45%.
- **Flexibilidad:** Las distribuciones Linux deben ser compatibles con diferentes arquitecturas de procesadores para poder cambiar el sistema sin dificultad, por ejemplo: i386, x86\_64, ia64, s390, s390x, ppc, sparc, entre otras. Peso 35%.
- **Actualización:** Debido a las constantes actualizaciones y nuevos sistemas embebidos es importante que sea compatible con estos. Peso 20%.

Las calificaciones otorgadas a cada sistema operativo son las siguientes:

#### 3.4.1 Calificación para Raspbian

- **Documentación = 2.** Existen muchos libros de Debian y hay un libro titulado, Learning Raspbian por William Harrington.
- **Compatibilidad = 0.** Es una versión de Debian, diseñada para soportar la arquitectura ARM.
- **Actualización = 2.** Las actualizaciones son constantes, estables y seguras provocando que los periodos de lanzamiento tarden.

#### 3.4.2 Calificación para Arch Linux

- **Documentación = 1.** Existe un libro titulado, Arch Linux environment set-up how-to por [Devolder, ike](#).
- **Compatibilidad = 1.** Soporta ARM, i686, x86\_64.
- **Actualización = -2.** Las actualizaciones son poco estables ya que los periodos de prueba son muy breves.

#### 3.4.3 Calificación para Kali Linux

- **Documentación = 1.** Existe mucha información y libros, pero es sistema operativo más complejo ya que esta entrado en seguridad.
- **Compatibilidad = 2.** Soporta [x86](#), [x86-64](#) y [ARM](#),
- **Actualización = 2.** Las actualizaciones son constantes, estables y seguras provocando que los periodos de lanzamiento tarden.

#### 3.4.4 Calificación para Pidora

- **Documentación = -2:** Existen muchos libros de Fedora, pero no se encontró ninguno de Pidora.
- **Compatibilidad = -1.** Es una versión de Fedora diseñada para soportar arquitectura ARM, por lo que para migrar a otros sistemas solo es necesario cambiar a Fedora.
- **Actualización = -2.** No hay actualizaciones constantes.

### 3.4.5 Calificación para Windows 10 IoT

- **Documentación = -1.** Existe poca información, aun así se encuentra un libro titulado Programmingforthe Internet ofThings: Using Windows 10 IoT Core and Azure IoT Suite por [DawidBorycki](#).
- **Compatibilidad = 1.** Soporta ARM, x86-64
- **Actualización = 2.** Las actualizaciones las desarrolla Windows.

### 3.4.6 Calificación para Ubuntu Core

- **Documentación = -2.** Existe poca información.
- **Compatibilidad = 1.** Soporta ARM y x86
- **Actualización = 2.** Las actualizaciones son regulares.

### 3.4.7 Calificación para RISC OS

- **Documentación = -2.** Existe poca información.
- **Compatibilidad = 1.** Soporta ARM y x86
- **Actualización = -2.** Debido a que hay muy pocos desarrolladores casi no hay actualizaciones.

### 3.4.8 Calificación para SARPİ

- **Documentación = -2.** Existe poca información.
- **Compatibilidad = -1.** Soporta ARM.
- **Actualización = 2.** Las actualizaciones son seguras y rápidas.

### 3.4.9 Calificación para FreeBSD

- **Documentación = 2:** Existen varios libros que hablan del sistema operativo.
- **Compatibilidad = -1.** Soporta ARM
- **Actualización = 1.** Las actualizaciones son seguras.

### 3.4.10 Calificación para RetroPie

- **Documentación = 1:** Existen un libro que habla de este sistema operativo titulado Retro Gamingonthe Raspberry Pi: TheEssentialGuideUpdatedforRetroPie 4.1 por Matt Smith
- **Compatibilidad = -2.** No es una opción para la aplicación del proyecto ya que este sistema operativo está enfocado a video juegos.,
- **Actualización = 1.** Las actualizaciones son seguras.

### 3.4.11 Resultados

Factor	Peso	Raspbian	Arch Linux	Kali Linux	Pidora	Windows 10 IoT	Ubuntu Core	RISC OS	SARPİ	FreeBSD	RetroPie
Documentación	0,55	2	1	1	-2	1	-2	1	-2	2	1
Compatibilidad	0,25	0	1	2	2	1	2	1	-1	-1	-2
Actualización	0,2	2	-2	2	-2	2	2	-2	2	2	2
	1	1,5	0,4	1,45	-1	1,2	-0,2	0,4	-0,95	1,25	0,45

Tabla 16 Selección de sistema operativo.

De la tabla anterior obtenemos que el sistema operativo a instalar en la Raspberry Pi es Raspbian.

### 3.5 Selección de lenguajes de control.

Debido a que se utilizará una Raspberry y el sistema operativo a utilizar es Raspbian, el único lenguaje de control disponible para comunicarse con los puertos de entrada y salida GPIO es Python. Siendo versión actual Python3.

### 3.6 Selección de metodología de control.

Para reducir el consumo energético se requiere un sistema de control, debido a que el tiempo de acción es irrelevante para la aplicación y que los actuadores de control de luces solo tienen dos estados, encendido y apagado. El mejor sistema de control a utilizar, en este caso, es ON - OFF.

### 3.7 Selección sistemas de control para de confort y consumo energético

A continuación, se muestra el desarrollo de un experimento para conocer cuáles son las variables más importantes para reducir el consumo energético y aumentar el confort.

Como se desea mejorar la relación consumo energético vs confort, se realizó un experimento en la habitación del hotel, variando 3 factores, la interfaz gráfica (A), control de iluminación(B) y sensores para detección de personas(C). Al mismo tiempo se tomaron calificaciones del confort mediante una encuesta. Los factores (A) y (B) son presentados a partir de mediciones del consumo energético de la habitación.

Para realizar el experimento se utilizaron dos niveles aleatorios encada uno (1, -1). Siendo (1), que la habitación cuenta con el sistema de control y (-1) establece que la habitación no cuenta con un sistema de control.

El experimento se realiza con una sola réplica, esto con la finalidad de saber cómo se afectan los valores de confort y consumo con las interacciones de los factores.

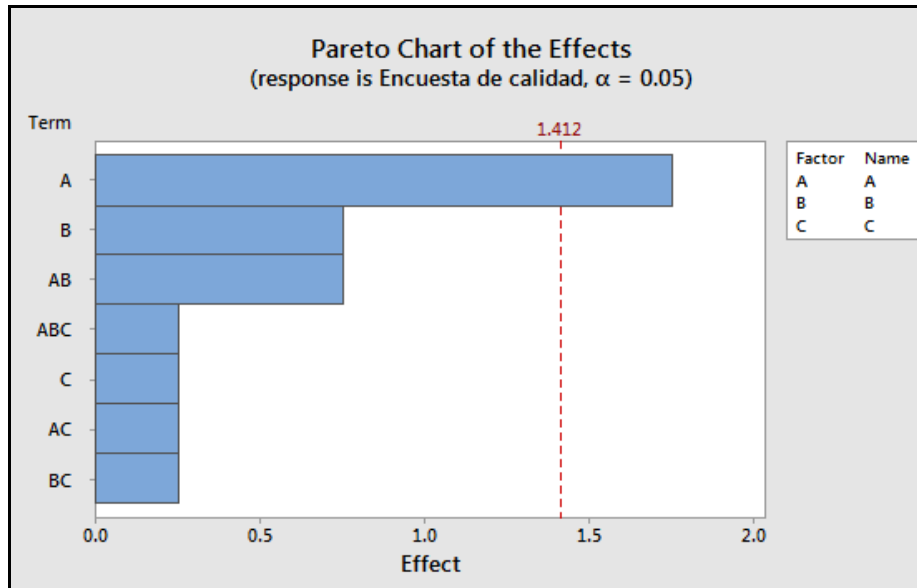
La encuesta de confort consideró como calificación más baja (-2) y la calificación más alta posible era (2), esto para evitar el error que una calificación negativa pueda ser considerada positiva.

Los datos obtenidos se muestran en la tabla siguiente:

Tabla 17 - Consumo energético Vs Encuesta de calidad.

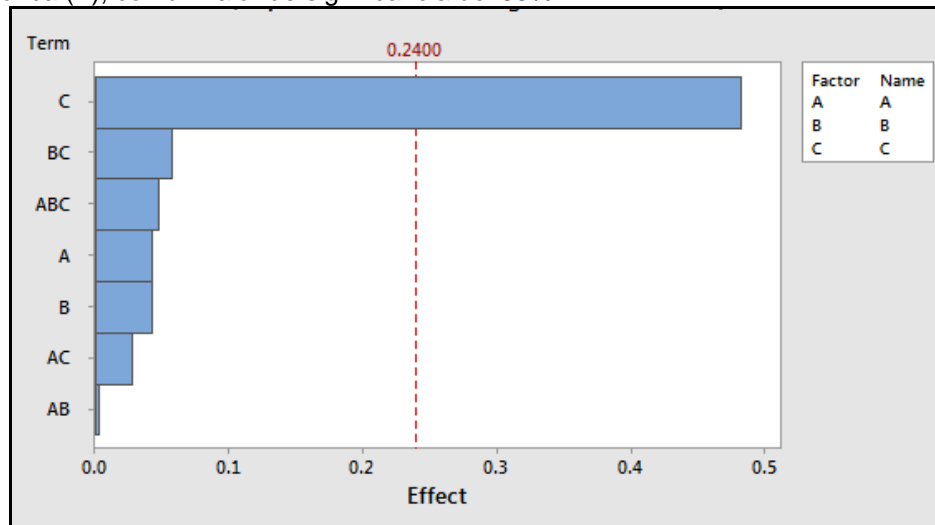
A	B	C	Consumo energético KW	Encuesta de calidad
1	-1	-1	1.27	2
1	-1	1	0.92	2
-1	-1	1	0.89	0
-1	1	-1	1.44	1
-1	1	1	0.92	1
1	1	1	0.86	2
1	1	-1	1.42	2
-1	-1	-1	1.39	-1

Haciendo el análisis de la tabla anterior, se obtienen las gráficas de Pareto, en las que se observan los factores más significativos.



**Ilustración 3 - Pareto en encuesta de calidad**

De la gráfica anterior, se observa que el valor más significativo para aumentar el confort en el huésped es la interfaz gráfica (A), con un valor de significancia del 95%.



**Ilustración 4 - Pareto en consumo energético.**

De la gráfica anterior, se observa que el valor más significativo para disminuir el consumo energético es la instalación de sensores para detección de personas (C), asegurando esta afirmación con un 95%. Otro dato importante que se puede observar en las gráficas anteriores es el factor (B), ya que no es significativo en los modelos. Para comprobarlo de forma visual, se realizan las gráficas de normalidad:



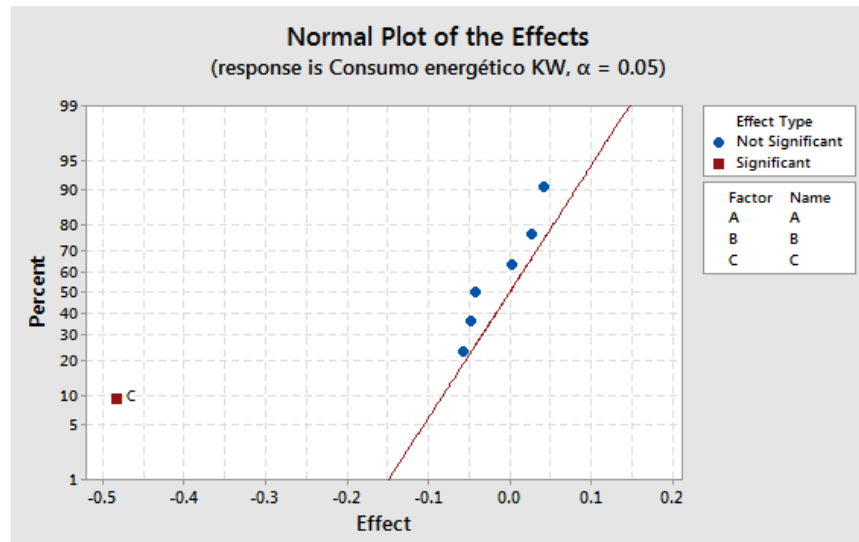


Ilustración 5 - Grafica de normalidad en consumo energético.

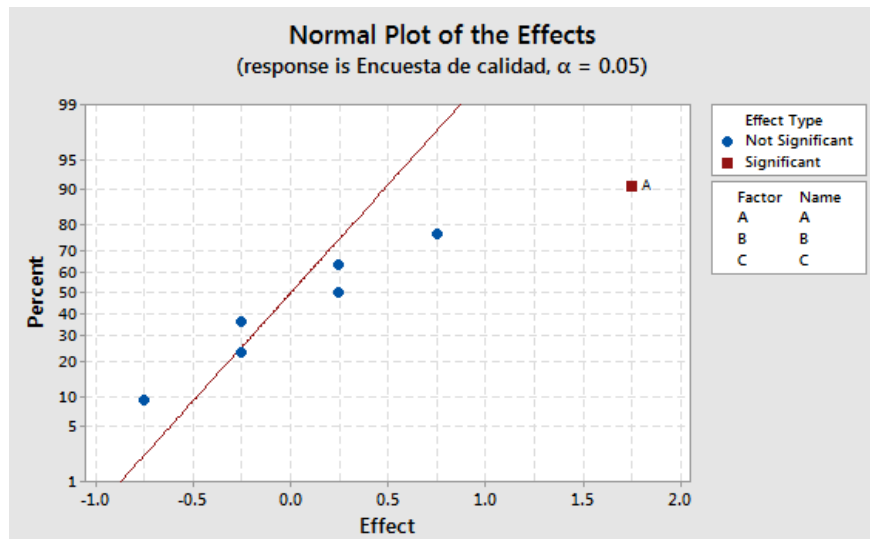


Ilustración 6 - Grafica de normalidad en encuesta de calidad

Las gráficas de normalidad permiten observar las diferencias entre las medias siendo estas estadísticamente significativas tanto en el factor (A) como en el factor (C). Por lo que el factor (B) en el modelo no es significativo para la mejora del consumo energético, ni para la mejora del confort.

### 3.7.1 Resultado

Si se desea mejorar la relación consumo energético vs confort, no es significativo hacer un control de temperatura. Pero se puede afirmar que, al tener una interfaz gráfica y un control por detección de personas, el consumo energético baja y el confort del huésped aumenta.

## IV.DESARROLLO

Para cumplir con el objetivo mencionado, el proyecto se realizó en tres etapas, cada una mejorando a la anterior para llegar al resultado.

## 4.1 Primera etapa. - Desarrollo del prototipo

Esta etapa consistió en hacer el desarrollo en la habitación con actuadores, sensores e interfaz gráfica que funcionaba mediante una aplicación en el sistema operativo Android. Probando así el funcionamiento de este sistema y su durabilidad.

Para solucionar el problema de la instalación, se optó por utilizar módulos flexibles que se pudieran comunicar por medio de tecnologías, tanto alámbricas como inalámbricas, quedando el prototipo de la siguiente forma, Ilustración 3.

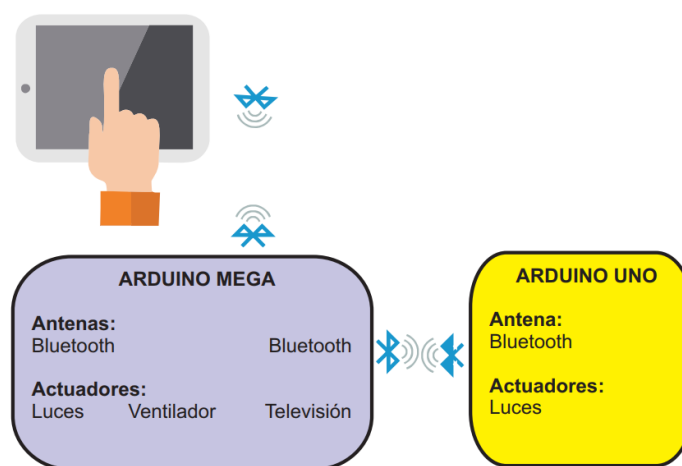


Ilustración 7 Primera etapa

Una vez que el prototipo fue terminado se notaron sus **deficiencias**, como son:

- Desarrollar una gran variedad de aplicaciones que estén pensadas para las distintas versiones de Android, IOs y Windows Mobile.
- Enseñarle al personal a instalar las aplicaciones y en caso de que no se pueda instalar, el huésped no podrá tener acceso al sistema domótico.

Por esta razón, se procedió a la siguiente opción:

## 4.2 Segunda etapa. - Mejora del Prototipo

Para resolver la problemática expuesta en las deficiencias anteriores, es necesario eliminar las aplicaciones. Porque las aplicaciones no son fáciles de instalar y obligan al hotel a tener un encargado para instalarlas.

Las aplicaciones tendrían que ser realizadas para distintos sistemas operativos: iOS, Android y Windows Mobile, donde cada una tendría que tener una lista de aplicaciones compatibles con las diferentes versiones que hay en el mercado y se tendrían que estar actualizando constantemente. Originando un gasto por cada desarrollo.

Para solucionar esta problemática se investigó y desarrollo un método distinto, que no requiere ninguna instalación y es compatible con cualquier dispositivo, sin importar su tamaño o sistema operativo, siendo este una página web responsiva.

Para poder desarrollar la interfaz, es necesario integrarle al prototipo un sistema de control conocido como Gateway, utilizando para éste una Raspberry Pi 3 (Tabla 5) y como es necesario realizar una mayor cantidad de pruebas del funcionamiento, se instaló un segundo modulo inalámbrico en el control

de luces, sin cambiar la tecnología bluetooth para evitar error de fiabilidad en las pruebas. La ilustración siguiente, muestra el esquema utilizado en la segunda versión.

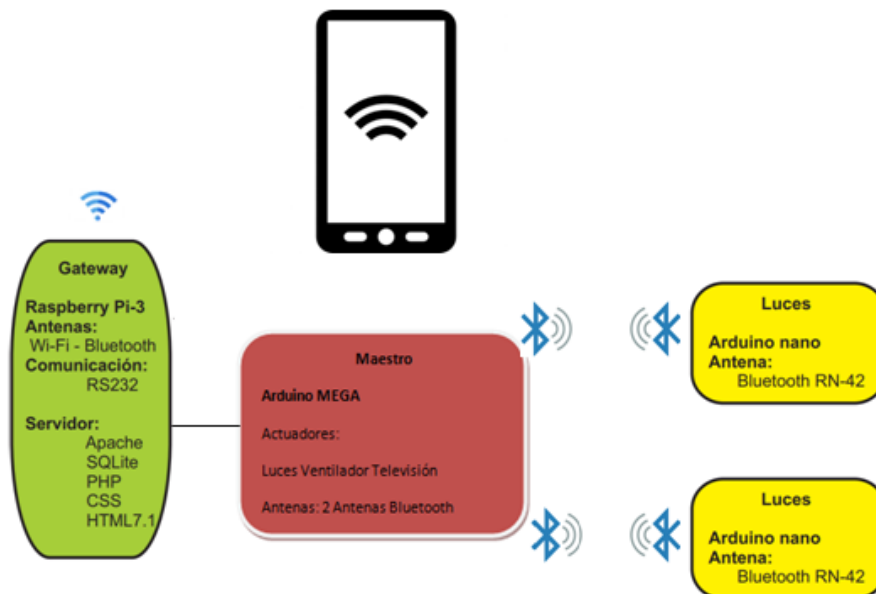


Ilustración 8 Segunda etapa

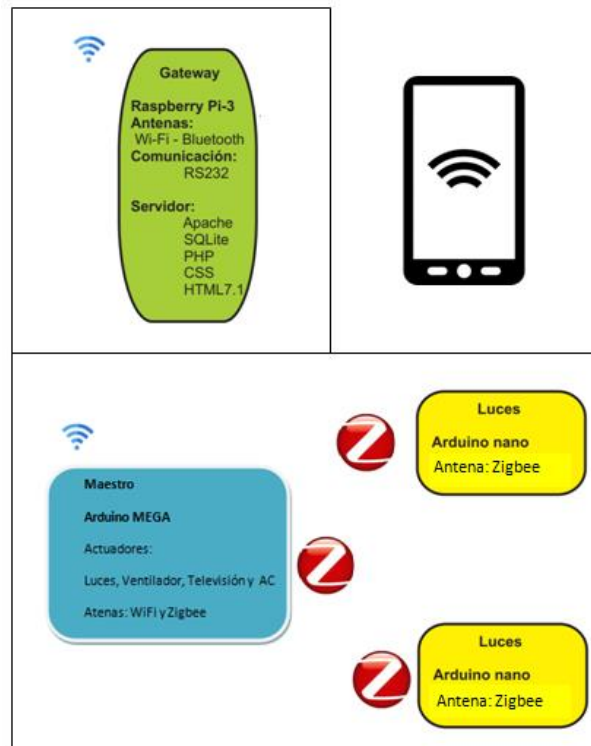
Esta etapa al estar instalada y probada en la habitación, se pudo notar un problema de conexión del gateway a la red inalámbrica WiFi del hotel, ya que al dejar funcionando correctamente el sistema, después de cierto tiempo se desconectaba de la red impidiendo al huésped utilizar el sistema.

### 4.3 Tercera etapa. - Diseño final

La problemática de la etapa anterior sucede cuando el hotel reinicia el sistema las conexiones WiFi, lo hace periódicamente para evitar conflictos de red, provocando así una desconexión del Gateway hasta que este vuelva a ser dado de alta. Este problema se soluciona al conectar la Raspberry a la red alámbrica ya que esta no se reinicia y mantiene una conexión fija.

Para lograr esta separación del gateway con el módulo maestro, se instaló una antena WiFi y al gateway, se configuró la comunicación mediante una IP (Dirección). Funcionando el proyecto mediante **internet de las cosas**. En la ilustración siguiente se muestra el esquema de esta versión, los cuadros simbolizan la separación que existe entre los elementos.

Además, se cambiaron las antenas Bluetooth, por Zigbee (Tabla 2). Como los módulos zigbee no tienen limitante con el número de nodos, la red puede ser más extensa, brindando al diseño final flexibilidad y costos bajos.



**Ilustración 9 Tercera etapa**

Esta etapa funciona, por cuestiones de su política interna se requiere permisos de los directivos para su correcta instalación ya que deben permitir un cambio en los permisos de seguridad de la antena *gateway* que envía la señal a la habitación.

La forma en la que el huésped puede utilizar la interfaz gráfica se muestra a continuación.

#### **4.4 Interfaz**

Como se menciona anteriormente el sistema utiliza como método de interfaz una página web responsiva o flexible, siendo que ésta se adapta automáticamente al tamaño de la pantalla. En las imágenes siguientes se muestran las visualizaciones en pantallas mayores a 700px (PC o Laptop) y menores de 400px la cual es la utilizada en los teléfonos inteligentes (*Smartphone*) se consideraron estos dos tamaños para las ilustraciones, ya que son los más comunes en estos dispositivos.

##### **4.4.1 Inicio**

La página tiene como objetivo mostrar de una forma clara cuales son los servicios accesibles para el huésped y pueda moverse de una forma intuitiva entre las siguientes opciones: iluminación, televisión, aire acondicionado y encuesta.

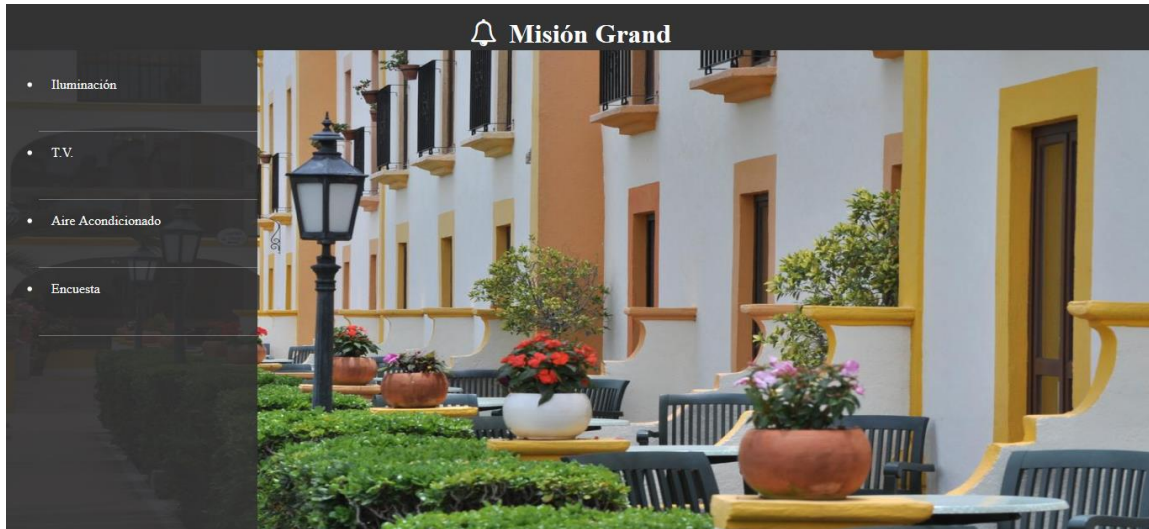


Ilustración 10 Inicio, pantalla mayor a 720px

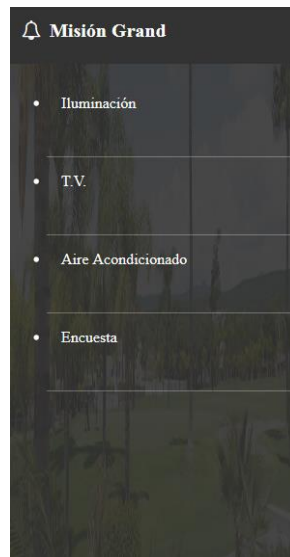


Ilustración 11 Inicio, pantalla menor de 400px

#### 4.4.2 Control de iluminación:

Esta página, permite controlar el encendido y/o apagado de las lámparas. La interfaz cuenta con un bosquejo de la habitación para facilitarle huésped su uso. El diseño se aprecia en la Ilustración 7 y 8.

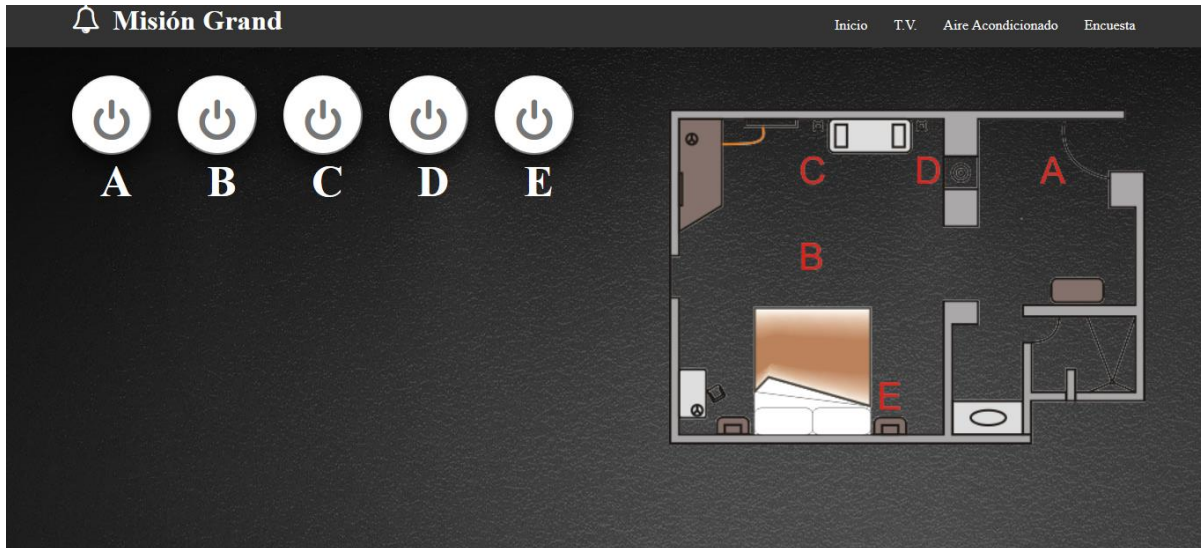


Ilustración 12 Control de iluminación en pantalla mayor a 720px



Ilustración 13 Control de iluminación, en pantalla menor de 400px.

#### 4.4.3 Control de televisión

La interfaz de control para televisión permite: Apagar/encender la TV, Cambiar el canal, cambiar el volumen y cuenta con canales preprogramados para facilitarle al huésped la búsqueda, promocionando al mismo tiempo el canal propio de la cadena misión.



Ilustración 14 Control de televisión en pantalla mayor a 720px.

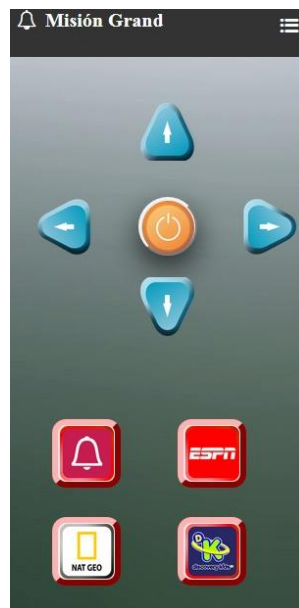


Ilustración 15 Consola de control de televisión en pantalla menor a 400px.

#### 4.4.4 Control de aire acondicionado

La interfaz permite el control del: Encendido y/o Apagado del aire acondicionado, regular su temperatura, la oscilación del viento y controlar el ventilador de techo. Muestra la temperatura en (grados) Centígrados y Fahrenheit.

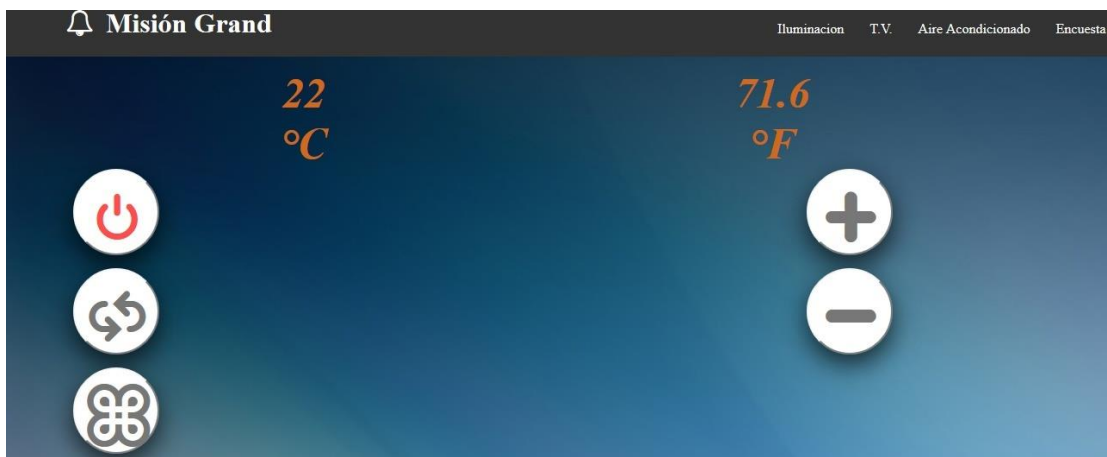


Ilustración 16 Control de clima en pantalla mayor a 720px.

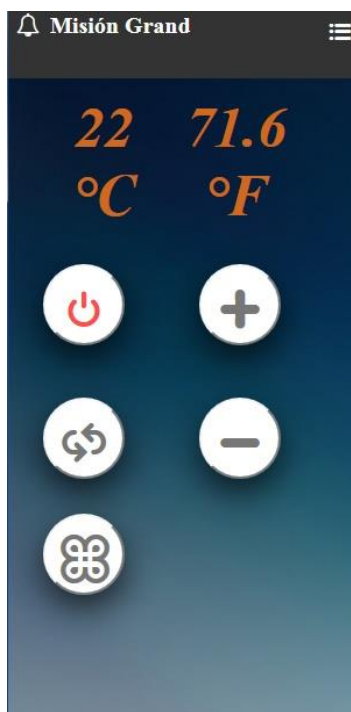


Ilustración 17 Control de televisión en pantalla menor a 400px.

#### 4.4.5 Encuesta

La encuesta sirve para conocer el índice de confort que tienen los huéspedes en la habitación. Esta al ser contestada envía los resultados de manera automática a un correo electrónico, y permite conocer el resultado de una forma más rápida y simple

Las calificaciones a las preguntas son: 2 la más alta y -2 la más baja, esto permite restar las calificaciones bajas al resultado final y así obtener un índice de confort correcto. Una vez contestada la encuesta, las calificaciones, como el resultado, se envían al correo electrónico al presionar el botón de enviar, la encuesta se muestra en la ilustración siguiente.



MISIÓN GRAND

1. ¿Cómo percibe la temperatura en la habitación?

A) Claramente aceptable

B) Aceptable

C) Me es indiferente

D) Inaceptable

E) Claramente inaceptable

2. ¿Cómo se siente con el apagado automático de focos en la habitación?

A) Claramente aceptable

B) Aceptable

C) Me es indiferente

D) Inaceptable

E) Claramente inaceptable

3. ¿Cómo siente al estar en una habitación automatizada?

A) Claramente aceptable

B) Aceptable

C) Me es indiferente

D) Inaceptable

E) Claramente inaceptable

4. Cuando vuelva a visitarnos prefiere:

A) Solicitar una habitación automatizada.

B) Me es indiferente.

C) Solicitar una habitación sin automatizar.

5. Usted está ahora mismo

A) Feliz

B) Satisfecho

C) Apático

D) Insatisfecho

E) Triste

Enviar

Ilustración 18 Encuesta de confort

## 4.5 Módulos:

Cada módulo varía dependiendo del sistema de la comunicación requerida (Wifi o Zigbee) y su aplicación. Los módulos son los siguientes:

### 4.5.1 Gateway

El Gateway, o pasarela, está constituido por una tarjeta Raspberry Pi3. Esta entidad es el intermediario entre el software de interfaz gráfica y el módulo maestro, Vía WiFi (Ilustración 5).

Su función principal radica en establecer un servidor que contenga las páginas web anteriormente mostradas para que cualquier dispositivo móvil pueda acceder a ellas siempre y cuando introduzcan su usuario y contraseña.

De esta manera cuando el usuario accede al servidor y ejecuta los comandos previamente programados en cada botón, inicializa la comunicación con el módulo maestro para él envío de la información. La siguiente tabla ejemplifica su funcionamiento.

```
Procedimiento interfaz
    Crea un servidor donde almacene las páginas web responsivas;
    Solicita usuario y contraseña para acceso al servidor;
    Si Botón- Se ejecuta;
        Entonces Inicializa comunicación con maestro
    Envía contraseña almacenada en Botón
    Termina comunicación con maestro;
Fin del procedimiento
```

Tabla 18 Pseudocódigo gateway

#### 4.5.2 Maestro

Contiene dos antenas, una WiFi y otra Zigbee, permitiéndole recibir señales enviadas por el gateway y esclavos. si recibe la señal por WiFi, la compara con una contraseña previamente almacenada, al ser la comparación positiva se realiza la acción para cual la contraseña está programada (control de luces, aire acondicionado, televisión y ventilador). Si no coincide la contraseña, envía la señal por el módulo Zigbee.

La antena Zigbee es la encargada del envío de señales de control a los esclavos y de reducir el consumo energético, mediante una constante comunicación con los esclavos, se puede conocer si hay o no, personas ocupando la habitación. Cada esclavo le envía una palabra clave que indica de que esclavo proviene la señal cada que se detecta movimiento.

A continuación, se ejemplifica el programa utilizado para el control de esclavos y reducción de consumo energético en pseudocódigo

```
Procedimiento comunicación
    K=contraseña;

    Serial 1- WiFi; Serial 2- Zigbee

    Si K == x
        Entonces
            Realiza acción en GPIO
        Si no, Envía dato a Serial 2

    Solicita información a esclavos
    Si (Serial 2 == "No hay movimiento");
        Entonces
            Inicializa contador de 15 minutos
            Si Serial 2 == ("si hay huésped, esclavo", E);
                Entonces
                    Desactiva contador;
                Si no,
                    Cambia estado GPIO en esclavos y maestro de ON a OFF
            Si no,
                Espera a que se active esclavo en puerta para volver a solicitar información.
```

Fin del procedimiento

Tabla 19 Seudocódigo maestro.

### 4.5.3 Esclavo

Los esclavos contienen:

1. **Sensor de presencia PIR.** -Para detección de huéspedes.
2. **Antena Zigbee.** - Para enviar y recibir información.

#### 1.- Identificación del huésped en la habitación:

Al utilizar sensores de presencia en cada esclavo se puede conocer si está o no ocupada la habitación. Para esto se ocupa como mínimo de dos esclavos uno en la entrada de la habitación y otro a la mitad. Su funcionamiento es sencillo si detecta movimiento en la puerta (esclavo, E=2) pero no en el sensor que está a mitad de la habitación.

#### 2. Antena Zigbee:

Con esta antena se reciben los datos provenientes del módulo maestro. Al recibirla se compara la información con las contraseñas almacenadas y el esclavo que contenga el mismo dato guardado que la señal es el que va a realizar la acción de encender o apagar algún foco.

El programa en pseudocódigo de los esclavos se muestra a continuación:

```
Procedimiento esclavo
K=password
E= número de esclavo

Serial- Zigbee
GPIO- Entrada PIR
x = Dato de entrada;
Si K == x; Entonces
Realiza acción en Gpio;
Si GPIO == alto;
Envía señal: "si hay huésped"; + E;
Fin del procedimiento
```

Tabla 20 Seudocódigo esclavo

## V. RESULTADOS

En este capítulo se analizan los resultados obtenidos en la disminución de consumo energético, utilizando el sistema de control mediante sensores de detección de presencia.

Los datos se recopilaron en un día, sin utilizar el sistema domótico de ahorro, pero detectando las entrada y salidas del huésped, para así comparar los resultados y obtener el porcentaje de ahorro que se hubiese obtenido utilizando el sistema domótico.

Las mediciones se tomaron con un intervalo de tiempo, cada 10 minutos el día 26 de marzo del 2019, siendo CSD (acrónimo de *Con Sistema Domótico*) en color naranja y SSD (acrónimo de *Sin Sistema Domótico*) en color azul, la gráfica comparativa es la siguiente:

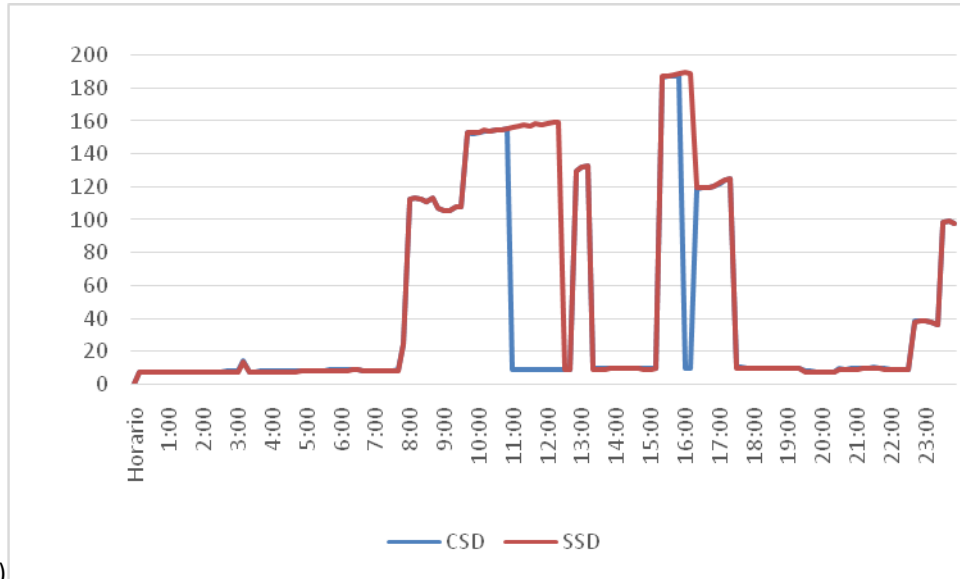


Ilustración 19–Medición del consumo energético factible de ser evitado en azul.

Para apreciar de una mejor forma el área bajo la curva y obtener el consumo energético final, se utilizó una aproximación tipo Monte Carlo, obteniendo así las siguientes gráficas y resultados:

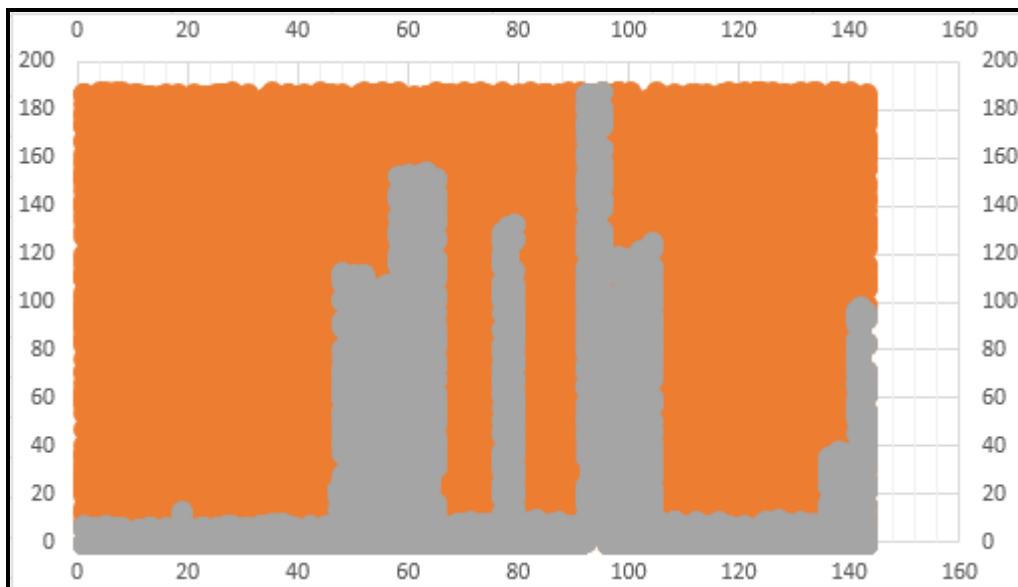


Ilustración 20 - Consumo energético con sistema domótico.

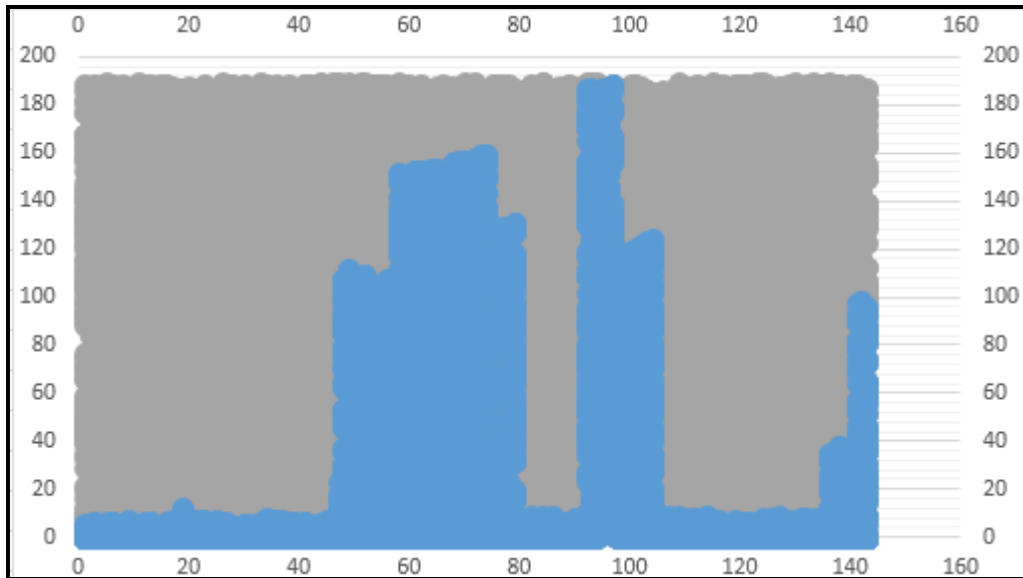


Ilustración 21 - Consumo energético sin sistema domótico.

De las mediciones se obtienen los siguientes resultados:

- El consumo energético del sistema domótico: 3.7 W
- Consumo energético en el día con sistema domótico [CSD]: 0.95 KW.
- Consumo energético en el día sin sistema domótico [SSD]: 1.24KW

Al comparar los resultados obtenidos, se obtiene que el ahorro con el sistema domótico generó un ahorro del 23%.

## **VI. RECOMENDACIONES Y CONCLUSIONES**

### **6.1 Recomendaciones**

El proyecto se encuentra en fase de prototipo dentro del hotel. En esta sección se mencionan ciertas recomendaciones las cuales pueden ayudar si el hotel quiera aumentar el número de habitaciones automatizadas.

#### **6.1.1 Hardware de la unida maestra**

Debido a que es un sistema centralizado se recomienda su instalación en una zona controlada por el personal del hotel; además, es necesario realizar actualizaciones cada año para que el sistema no quede obsoleto con nuevas tecnologías entrantes.

### 6.1.2 Software de la unida maestra

Un paso de mejora será añadir tecnologías de control inteligente, que aprendan el comportamiento de consumo en los huéspedes para optimizar aún más los recursos.

### 6.1.3 Hardware de las unidades modulares

Los diferentes dispositivos se instalaron aumentando el número de chalupas, pero para una instalación más sencilla será necesario calcular los tamaños de cada módulo y construir los gabinetes que cumplan con esas medidas.

### 6.1.4 Software de las unidades modulares

Al aumentar el tamaño de la red Zigbee será posible enviar comandos para el monitoreo de cada estación con lo cual, se podrá hacer el mantenimiento o reparación de alguna unidad de forma más rápida.

## 6.2 Conclusiones

El sistema desarrollado, se encuentra instalado en la habitación 354 y cumple con las expectativas del hotel. Tras las pruebas realizadas, los directivos confían en que el sistema va a mejorar la relación entre confort y consumo energético.

Además, desde la instalación del prototipo hasta la tercera etapa no ha reportado fallos, por lo que se afirma que el sistema es confiable, durable y funcional.

Para validar el sistema, es necesario esperar el permiso de los principales socios accionistas puesto que involucra un cambio en sus políticas de seguridad. De momento se constata que puede ser una buena herramienta para mejorar el confort y disminuir el consumo energético, pudiendo convertirse en un proyecto tecnológico a futuro.

## VII. Bibliografía:

### 7.1 Libros

Rey Martínez, F., Velasco Gómez, E., León Medina, S., & Varela Díez, F. (2010). *Eficiencia energética en edificios* (p. 8). Madrid: Thomson-Paraninfo.

Archila, J. (2012). *Estudio técnico para implementación de inmótica*. 1st ed. Editorial AcadémicaEspa.

Knutson, C. (2004). *IrDA principles and protocols*. Salem, UT: MCL Press

Merz, H., Hansemann, T., & Hübner, C. (2009). *Building automation*. Berlin: Springer.

Torres Cantero, J., Ruiz, A. and Espinosa, A. (2013). *Introducción a la Informática*. España: McGraw-Hill España, p.20.

Salas Arriarán, S. (2015). *Todo sobre sistemas embebidos*. 1st ed. Lima: Editorial UPC.

Torrente Artero, O. (2013). *Arduino*. 1st ed. México D.F.: Alfaomega.41.

Balcells, J., & Romeral, J. (2003). *Autómatas programables*. Barcelona: Marcombo-Boixareu.

Torres Cantero, J., Ruiz, A. and Espinosa, A. (2013). *Introducción a la Informática*. España: McGraw-Hill España, p.561.

Kernighan, B., Ritchie, D. and Padilla, J. (1985). *El lenguaje de programación C*. 1st ed. México: Prentice-Hall Hispanoamericana.

Gottfried, B. and Caro Murillo, G. (1983). *Teoría y problemas de programación BASIC*. México: McGraw-Hill.

Hall, T. And Stacey, J. (2009). *Python 3 for absolute beginners*. [New York, NY]: Apress, p.10.

Dean, J. (n.d.). *Introducción a la programación con Java*. p.13.

Balagurusamy, E. (2002). *Programming in Ansi C*. 1st ed. Tata McGraw-Hill.

Torres Cantero, J., Ruiz, A. and Espinosa, A. (2013). *Introducción a la Informática*. España: McGraw-Hill España, p.14.

BaigViñas, R. And Aulí Llinàs, F. (2003). *Sistema operativo GNU/Linux básico*. 1st ed. Barcelona: UOC.

Báez, M. and Sanz, D. (2013). *Introducción a Android*. Madrid: E.M.E. Editorial, p.1.

Tanenbaum, A., Núñez Ramos, E., Trujillo Fernández, F. and García Espinosa, A. (2003). *Redes de computadoras*. México [etc.]: Pearson Educación.

Tanenbaum, A., & Wetherall, D. (2010). *Computer Networks, Fifth Edition*. Prentice Hall.

Åström, K., Dormido Bencomo, S., Guzmán Sánchez, J., & Hagglund, T. (2009). *Control PID avanzado* (p. 81). Madrid: Pearson Educación.

Gutiérrez Hinestroza, M., & Iturralde Kure, S. (2017). *Los Fundamentos Básicos de Instrumentación y Control* (1st ed., pp. 120-130). Ecuador: UPSE.

Ponce Cruz, P. And Herrera, A. (2010). *Inteligencia artificial con aplicaciones a la ingeniería*. México: Alfaomega.

## 7.2 Publicación

Dominici, G. and Guzzo, R. (2010) Customer Satisfaction in the Hotel Industry: A Case Study from Sicily. *International Journal of Marketing Studies*, 2, 3-12.

Secretaría de Turismo y Universidad Anáhuac del Sur A. C. (2014). *Metodología para la medición de la competitividad de los prestadores de servicios turísticos inscritos en las certificaciones y los sellos de calidad de la Secretaría de Turismo (SECTUR)*. México: Lid Editorial Mexicana.

González Muñoz, R. (2015). *Improving the beaglebone board with embedded Ubuntu, enhanced GPMC driver and Python for communication and graphical prototypes*. Final Master. Universidad Politécnica de Catalunya.

Schinagl, O. (2014). *Getting Started with Cubieboard*. Packt Publishing, p17.

Waher, P. (n.d.). *Learning Internet of Things*, p17.

Dundar, O. (2015). *Home Automation with Intel Galileo*. MUMBAI: Birmingham, p

## 7.3 WEB

gcec. (2014). *Global Commission on the Economy and Climate*. [online] Available at: <http://newclimateeconomy.report/2014/> [Accessed 18 Oct. 2018].

El Modelo SERVQUAL de Calidad de Servicio - Aiteco Consultores. (2018). Retrieved from <https://www.aiteco.com/modelo-servqual-de-calidad-de-servicio/>

IEEE 802.15.1. (2018). Retrieved from <http://www.ieee802.org/15/pub/TG1.html>

Cubieboard, (2016). *Cubieboard*. [image] Available at: <http://cubieboard.org> [Accessed 21 Sep. 2016]

Arduino.cc. (2016). *Arduino - Home*. [online] Available at: <https://www.arduino.cc> [Accessed 22 Sep. 2016].

UDOO. (2018). *UDOO X86 - UDOO*. [online] Available at: <https://www.udoo.org/udoo-x86/> [Accessed 28 Oct. 2018].

Banana Pi - BPI Single Board Computers Official Website. (2018). Retrieved from <http://www.banana-pi.org/>

Raspberry Pi. (2016). *Raspberry Pi - Teach, Learn, and Make with Raspberry Pi*. [online] Available at: <http://www.raspberrypi.org> [Accessed 18 Nov. 2016].

Raspbian.org. (2016). *Frontpage - Raspbian*. [online] Available at: <https://www.raspbian.org/> [Accessed 12 Nov. 2016].

Archlinuxarm.org. (2018). *Arch Linux ARM*. [online] Available at: <https://archlinuxarm.org/> [Accessed 30 Oct. 2018].

Docs.kali.org. (2018). [online] Available at: <https://docs.kali.org/introduction/what-is-kali-linux> [Accessed 30 Oct. 2018].

Pidora.ca. (2018). *Pidora - Raspberry Pi Fedora Remix*. [online] Available at: <http://pidora.ca/> [Accessed 30 Oct. 2018].

Ubuntu.com. (2018). *Ubuntu Core | Ubuntu*. [online] Available at: <https://www.ubuntu.com/core> [Accessed 31 Oct. 2018].

Developer.microsoft.com. (2018). *Windows 10 IoT*. [online] Available at: <https://developer.microsoft.com/en-us/windows/iot> [Accessed 8 Feb. 2018].

Riscosopen.org. (2018). *RISC OS Open: Welcome*. [online] Available at: <https://www.riscosopen.org/content/> [Accessed 21 Oct. 2018].

Sarpi.fatdog.eu. (2018). *SARPi Project - Slackware ARM on a Raspberry Pi*. [online] Available at: <http://sarpi.fatdog.eu/> [Accessed 1 Nov. 2018].

Davis, B. (2018). *Welcome to RaspBSD*. [online] Raspbsd.org. Available at: <https://www.raspbsd.org/> [Accessed 6 Nov. 2018].

Freertos.org. (2018). *FreeRTOS - Market leading RTOS (Real Time Operating System) for embedded systems with Internet of Things extensions*. [online] Available at: <https://www.freertos.org/> [Accessed 5 Nov. 2018].

Python - Raspberry Pi Documentation. (2017). Retrieved from <https://www.raspberrypi.org/documentation/usage/python/>



# Congreso Internacional de Investigación Academia Journals



CENTRO  
PANAMERICANO  
DE ESTUDIOS  
SUPERIORES  
0079

CONGRESO INTERNACIONAL DE INVESTIGACIÓN  
DE ACADEMIA JOURNALS MORELIA 2018  
ACADEMIAJOURNALS.COM

## CERTIFICADO

OTORGADO A

ING. ISAAC GUTIÉRREZ VALLADARES  
DR. RAÚL RAMÍREZ LÓPEZ

M.C. AGUSTÍN BARRERA NAVARRO

### POR SU ARTÍCULO INTITULADO

DESARROLLO E INSTALACIÓN DE UN SISTEMA DE CONTROL  
AUTOMATIZADO Y SUSTENTABLE PARA LA HABITACIÓN  
356 DEL HOTEL MISIÓN JURIQUILLA

EL CUAL FUE PRESENTADO EN EL CONGRESO DESARROLLADO DEL 16 AL 18 DE MAYO DE 2018  
EN MORELIA, MICHOACÁN, MÉXICO Y PUBLICADO EN EL PORTAL DE INTERNET  
ACADEMIAJOURNALS.COM, CON ISSN 1946-5351 ONLINE, VOL. 10#3, 2018 Y EN EL LIBRO  
ELECTRÓNICO ONLINE TITULADO *COMPENDIO DE INVESTIGACIÓN MORELIA 2018*  
CON ISBN 978-1-939982-36-0

Handwritten signature of Rafael Moras.

**RAFAEL MORAS, PH.D. P.E**  
EDITOR, ACADEMIA JOURNALS

Handwritten signature of Rector Aldo Emilio Tello Carrillo.

**RECTOR ALDO EMILIO TELLO CARRILLO**  
CENTRO PANAMERICANO DE ESTUDIOS SUPERIORES

Artículo More100



