

TRABAJO ESPECIAL DE GRADO

SISTEMA DE SUPERVISIÓN BASADO EN UNA PLATAFORMA DE SOFTWARE GRATUITO PARA INMÓTICA

Presentado ante la Ilustre
Universidad Central de Venezuela
por el Br. Freitas R., Cecilio A.
para optar al Título de
Ingeniero Electricista

Caracas, 2017

TRABAJO ESPECIAL DE GRADO

SISTEMA DE SUPERVISIÓN BASADO EN UNA PLATAFORMA DE SOFTWARE GRATUITO PARA INMÓTICA

Tutor Académico: Ing. Tamara Pérez

Presentado ante la Ilustre
Universidad Central de Venezuela
por el Br. Freitas R., Cecilio A.
para optar al Título de
Ingeniero Electricista

Caracas, 2017

CONSTANCIA DE APROBACIÓN

Caracas, 03 de noviembre de 2017

Los abajo firmantes, miembros del Jurado designado por el Consejo de Escuela de Ingeniería Eléctrica, para evaluar el Trabajo Especial de Grado presentado por el Br. Cecilio A Freitas R., titulado:

“SISTEMA DE SUPERVISIÓN BASADO EN UNA PLATAFORMA DE SOFTWARE GRATUITO PARA INMÓTICA”


Consideran que el mismo cumple con los requisitos exigidos por el plan de estudios conducente al Título de Ingeniero Electricista en la opción de Electrónica y Control, y sin que ello signifique que se hacen solidarios con las ideas expuestas por el autor, lo declaran APROBADO.



Prof. Mercedes Arocha
Jurado



Prof. Rafael Rivero
Jurado



Prof. Tamara Pérez
Tutor Académico

DEDICATORIA

Todo el esfuerzo durante estos largos y difíciles años llenos de pruebas, obstáculos y momentos de dicha se los dedico a mi mamá, Amanda Ramones, por ella y para ella son todos mis logros y a mi papá, Antonio Freitas, aunque no estas físicamente sé que siempre estarás cuidando de mí y yo orgulloso de ser tu hijo.

A mis abuelas, Ramona y María, que guían mi camino desde el cielo.

A todos mis amigos y hermanos de vida, Los Fp's, El Equipo Eficiencia y Tamarindo VIP, además de aquellos que no están dentro de un grupo de WhatsApp pero que de igual manera tuve la dicha de contar con su amistad y apoyo, para mencionar tan solo algunos: Edgarith, Mariana P, Paola, Crespollo, Centeno, Yermael, Tabare, Sasa, Ana, Ismael, Aníbal, Jessica, Andrés, Lu, Benito, Male, Mariu, Rosa, Shakira, Leandro, El Enano, Santos, Kenny, Fabiany...

Una dedicatoria especial y disculpas para Jesús Delgado, Daniel Pérez, Bárbara Hernández, Carla Zurbaran, Luisana Cisnero, Mariana Moncada, Geruzza Ramones, Juanito y Gerardo Bethencourt, por no haber logrado culminar esta meta antes de que emprendieran un nuevo camino fuera de Venezuela, nos veremos nuevamente en alguna parte del mundo, el abrazo de felicitaciones lo quiero en persona.

RECONOCIMIENTOS Y AGRADECIMIENTOS

En primer lugar a mi vieja, Amanda Ramones, por todo lo que me ha dado sin esperar nada a cambio, te amo. A mí no muy pequeña familia, mis hermanas, tíos, tías y primos, en especial a la negra más bella, Zoraida, mi segunda madre.

A la honorable Universidad Central de Venezuela por haberme formado, permitido crecer y enseñarme a vencer la sombra. A los docentes que durante este largo camino lleno de obstáculos y pruebas, tuve la oportunidad y el placer de aprender de ellos, ciertas condiciones aplican.

A Mercedes Arocha, por ser el ejemplo de fuerza, lucha y perseverancia que deseo seguir; por ser la mejor consejera en momentos de desesperación y cuando todo parece irse a pique; por ser “LA PROFE” que irradia alegría en la escuela y por ser esa amiga que está en momentos buenos y malos.

A mi tutora Tamara Pérez, por su apoyo, paciencia y sobre todo por su exigencia en la elaboración de este humilde Trabajo Especial de Grado, gracias y disculpe las canas verdes.

Por último y no por eso menos importante a Miriam de Hernández, Diana Kadeer, Belén Delgado, Lissette Flores, Nora Gómez, Patricia Leal, Thais Rodríguez, Verónica y Ricardo Bethencourt por su apoyo, animo, ayuda y casi adopción en algún momento de este camino.

A Morgan Freeman.

Cecilio A. Freitas R.

SISTEMA DE SUPERVISIÓN BASADO EN UNA PLATAFORMA DE SOFTWARE GRATUITO PARA INMÓTICA

Tutor Académico: Ing. Tamara Pérez. Tesis. Caracas. U.C.V. Facultad de Ingeniería. Escuela de Ingeniería Eléctrica. Ingeniero Electricista. Opción: Electrónica. Institución: Universidad Central de Venezuela. 2017. 69h. + Anexos.

Palabras claves: Sistema de Supervisión; Software Gratuito; OpenHAB; Inmótica.

Resumen. La inmótica integra diversos sistemas en uno para controlar y automatizar edificios no destinados a la vivienda, dando como resultado una gestión eficiente del uso de la energía, aportar seguridad, confort y comunicación entre el usuario y el sistema. La Universidad Central de Venezuela (UCV) se caracteriza por ser precursora de conocimiento, ideas, desarrollo y líderes, por lo tanto una propuesta de un sistema inmótico para la Sala de Computación A “J.J. Martini” ubicada en la Escuela de Ingeniería Eléctrica, se convierte en un excelente punto de partida para futuras investigaciones. En este trabajo se establecieron los parámetros gestionados por el sistema de supervisión inmótico, la selección del software gratuito, la plataforma tecnológica y los componentes que dotan al sistema con la capacidad de supervisar y controlar los parámetros establecidos, el diseño del sistema de supervisión y la validación del mismo. Después de diseñar el sistema de supervisión inmótico, fue puesto a prueba tomando en cuenta los siguientes aspectos, la estrategia de control, la comunicación y el manejo de entradas y salidas, obteniendo como resultado un buen desempeño en cada uno de ellos. El software seleccionado se adapta a los parámetros, tiene la versatilidad de ser ejecutado en varias plataformas tecnológicas, presenta una gran variedad de complementos y se ajusta perfectamente a las necesidades de la Sala de Computación A, concluyendo que el sistema de supervisión inmótico diseñado cumple las necesidades establecidas, cuenta con un diseño sencillo y estrategia de control que satisface los parámetros determinados a gestionar, además puede ser expandido a otros ambientes dentro de la Escuela de Ingeniería Eléctrica de la UCV.

ÍNDICE GENERAL

<i>CONSTANCIA DE APROBACIÓN</i>	<i>iii</i>
<i>DEDICATORIA</i>	<i>iv</i>
<i>RECONOCIMIENTOS Y AGRADECIMIENTOS</i>	<i>v</i>
<i>RESUMEN</i>	<i>vi</i>
<i>ÍNDICE GENERAL</i>	<i>vii</i>
<i>LISTA DE TABLAS</i>	<i>x</i>
<i>LISTA DE FIGURAS</i>	<i>xi</i>
<i>LISTA DE SIGLAS Y ACRÓNIMOS</i>	<i>xiii</i>
<i>INTRODUCCIÓN</i>	<i>1</i>
<i>CAPÍTULO I</i>	<i>3</i>
<i>MARCO REFERENCIAL</i>	<i>3</i>
PLANTEAMIENTO DEL PROBLEMA	4
OBJETIVOS	4
Objetivo general	5
Objetivos específicos	5
ALCANCE	5
LIMITACIONES	6
JUSTIFICACIÓN DEL PROYECTO	6
ANTECEDENTES DE LA INVESTIGACIÓN	7
<i>CAPÍTULO II</i>	<i>9</i>
<i>MARCO TEÓRICO</i>	<i>9</i>
EDIFICIOS INTELIGENTES	9

INMÓTICA	10
DOMÓTICA	10
Características de la Domótica:	11
Aplicaciones de la Domótica:	12
Elementos básicos de un sistema domótico	12
Topologías de red de un sistema domótico	14
Arquitectura de un sistema domótico	15
PROTOCOLO DE COMUNICACIÓN	17
ESTÁNDARES USADOS EN DOMÓTICA	18
NORMATIVAS PARA LA DOMÓTICA	19
SOFTWARE ABIERTO PARA DOMÓTICA	20
ENTORNO DE PROGRAMACIÓN	23
<i>CAPÍTULO III</i>	25
<i>DISEÑO DEL SISTEMA INMÓTICO</i>	25
DIAGNÓSTICO DEL ÁREA DE TRABAJO	27
PARÁMETROS DE LA PROPUESTA DE SISTEMA INMÓTICO	28
CRITERIOS DE DISEÑO PARA EL SISTEMA INMÓTICO	28
SELECCIÓN DEL SOFTWARE ABIERTO PARA EL SISTEMA INMÓTICO	29
PLATAFORMA TECNOLÓGICA	31
COMPONENTES PARA EL SISTEMA INMÓTICO	31
ESTRUCTURA DEL SISTEMA INMÓTICO	32
Arquitectura	32
Topología	33

ESTRATEGIA DE CONTROL DEL SISTEMA INMÓTICO PROPUESTO	33
Gestión de energía	33
Gestión de seguridad	34
Gestión de confort	34
PROGRAMACIÓN DEL SISTEMA INMÓTICO	34
<i>CAPÍTULO IV</i>	37
<i>VALIDACIÓN DE LA PROPUESTA</i>	37
VALIDACIÓN DE LA ESTRATEGIA DE CONTROL	37
Apagado automático según horario establecido	37
Alerta de cuando se enciende el aire acondicionado y se encuentra la puerta o una ventana abierta	39
Alerta de incendios	40
Alerta de presencia fuera del horario establecido	41
Escenarios preestablecidos	42
Automatización selectiva	45
VALIDACIÓN DE LA COMUNICACIÓN Y MANEJO DE ENTRADAS Y SALIDAS	47
Comunicación Modbus	47
Comunicación por puerto Serial	57
Comunicación vía Ethernet	61
<i>CONCLUSIONES</i>	63
<i>RECOMENDACIONES</i>	64
<i>BIBLIOGRAFÍA</i>	68
<i>GLOSARIO</i>	69
<i>ANEXOS</i>	70
ANEXO A	70

LISTA DE TABLAS

Tabla 1: Actividades por objetivo específico _____	25
Tabla 2: Criterios para selección de Software _____	29
Tabla 3: Componentes Seleccionados _____	32
Tabla 4: Parámetros de los escenarios del SCA _____	43

LISTA DE FIGURAS

Figura 1: Topologías de red. [2] _____	14
Figura 2: Arquitectura Centralizada _____	15
Figura 3: Arquitectura Descentralizada _____	16
Figura 4: Arquitectura Distribuida _____	16
Figura 5: Secciones del Eclipse Smarthome _____	24
Figura 6: Esquema del sistema inmótico propuesto _____	26
Figura 7: Interfaz de usuario obtenida _____	35
Figura 8: Lista de eventos ocurridos a las 9pm _____	38
Figura 9: Alerta de aire acondicionado encendido con puerta o ventana abierta. ___	39
Figura 10: Alerta de incendios _____	40
Figura 11: Alarma de presencia después de las 9pm _____	41
Figura 12: Estado inicial en OFF _____	43
Figura 13: Cambio al escenario AHORRO _____	44
Figura 14: Cambio de escenario a CLASES _____	44
Figura 15: Registro de eventos entre los cambios de escenarios _____	45
Figura 16: Automatización selectiva (Miércoles 8:00am) _____	46
Figura 17: Apagado automático (Martes 9:00pm) _____	46
Figura 18: Registro del evento automatizado selectivamente _____	47
Figura 19: Montaje de prueba vista frontal _____	48
Figura 20: Montaje de prueba visto desde arriba _____	49
Figura 21: Micro820 y conexiones _____	49
Figura 22: Diagrama de escalera Micro820 _____	50
Figura 23: Asignación de variables Modbus del Micro820 _____	51
Figura 24: Configuración del complemento Modbus en OpenHAB _____	52
Figura 25: Error en el complemento de OpenHAB _____	52
Figura 26: Configuración inicial de variables Modbus en TOP Server 6 _____	53
Figura 27: Valores de las variables del Micro820 vistas desde TOP Server 6 _____	53
Figura 28: Prueba de escritura en los registros del Micro820 desde TOP Server 6 _	54

Figura 29: Variables tipo coil y holding de Modbus Poll _____	55
Figura 30: Configuración de los parámetros de comunicación de Modbus Poll ____	55
Figura 31: Configuración de los elementos y complemento Modbus de OpenHAB	56
Figura 32: Comportamiento entre OpenHAB y Modbus Poll _____	56
Figura 33: Estación #1 encendida y respuesta en el Arduino Uno _____	59
Figura 34: Estación #1 apagada y respuesta en el Arduino Uno _____	59
Figura 35: Encendido de LED según porcentaje de cierre de las persianas _____	60
Figura 36: Comportamiento de la aplicación Android _____	62
Figura 37: Conexión entre aplicación Android e interfaz Basic UI _____	62

LISTA DE SIGLAS Y ACRÓNIMOS

- API:** *Application Program Interface* (Interfaz de Programas de Aplicación)
- CEDOM:** Asociación Española de Domótica e Inmótica
- CEN:** Comité Europeo de Normalización
- CENELEC:** Comité Européen de Normalisation Electrotechnique (Comité Europeo de Normalización Electrotécnica)
- EIB:** *European Installation Bus* (Bus de Instalación Europeo)
- EHS:** *European Home System* (Sistema Europeo del Hogar)
- IDE:** *Integrated Drive Electronics* (Ambiente de Desarrollo Integrado)
- IEC:** *International Electrotechnical Commission* (Comisión Electrotécnica Internacional)
- IEEE:** *Institute of Electrical and Electronics Engineers* (Instituto de Ingeniería Eléctrica y Electrónica)
- IFTTT:** *If This, Then That* (Si ocurre Esto, Haz esto Otro)
- IoT:** *Internet of Things* (Internet de las Cosas)
- ISO:** *International Organization for Standardization* (Organización Internacional de Normalización)
- MQTT:** *Message Queueing Telemetry Transport* (Transporte de Telemetría de Mensaje de Cola)
- OSI:** *Open System Interconnection* (Modelo de Interconexión de Sistemas Abiertos)
- PC:** *Personal Computer* (Computador Personal)
- SCA:** Sala de Computación A
- SMS:** *Short Message Service* (Servicio de Mensajes Cortos)
- URL:** *Uniform Resource Locator* (Localizador Uniforme de Recursos)
- USB:** *Universal Serial Bus* (Bus Universal Serial)

INTRODUCCIÓN

La computación, la telefonía celular, la iluminación artificial entre un sinnúmero de aplicaciones basadas en el aprovechamiento del fenómeno de la electricidad, están desarrolladas con el fin de facilitar las actividades diarias de la vida del ser humano y satisfacer sus necesidades. La tecnología con elementos tan comunes como el teléfono celular, el televisor, los aires acondicionados, las cocinas eléctricas o los conjuntos de dispositivos que conforman un sistema de seguridad, han tenido un impacto ambiental negativo.

Para dar una solución a la problemática del deterioro ambiental que permita el uso inteligente y eficiente de los recursos mediante el uso de los avances tecnológicos en diferentes áreas se origina la inmótica, que ofrece la integración de diversos sistemas en uno solo para controlar y automatizar edificios no destinados a la vivienda, como hoteles, centros comerciales, escuelas, universidades, hospitales y todos los edificios terciarios, dando como resultado una gestión eficiente del uso de la energía, además de aportar seguridad, confort y comunicación entre el usuario y el sistema. [1]

La idea de tomar una serie de elementos de relativamente bajo costo, como una computadora de escritorio, laptop o inclusive un sistema embebido en conjunto de un software gratuito enfocado a inmótica, puede ser una solución más que accesible para supervisar o controlar muchos de estos espacios no destinados a la vivienda. Es importante destacar, que cualquier mejora que busque el ahorro energético, tendría como consecuencia un mejor aprovechamiento de los recursos naturales causando así un impacto ambiental positivo.

La Universidad Central de Venezuela desde su fundación el 22 de diciembre de 1721 se ha caracterizado por ser precursora de conocimiento, ideas, desarrollo y líderes, y en la actualidad no es muy diferente, porque se mantiene intacto sus ideales de innovar, fomentar y poner en práctica nuevas tendencias y tecnologías en busca del

desarrollo y bienestar del país. Por lo tanto una propuesta de un sistema inmótico para la Sala de Computación A “J.J. Martini” ubicada en la Escuela de Ingeniería Eléctrica, se convierte en un excelente punto de partida para futuras investigaciones y un ejemplo más de vanguardia de la casa que vence la sombra.

El presente trabajo especial de grado se estructura en cuatro capítulos, el Capítulo I presenta el marco referencial que incluye el planteamiento del problema, el objetivo general, los objetivos específicos, los alcances y limitaciones. En el Capítulo II se profundiza en los conceptos necesarios para comprender el trabajo y se presenta una breve descripción sobre el entorno de programación del sistema inmótico diseñado. Mientras tanto en el Capítulo III se especifican los detalles de las diferentes etapas establecidas para el diseño del sistema inmótico propuesto en este trabajo especial de grado. Por último el Capítulo IV muestra los resultados obtenidos de cada una de las pruebas realizadas para verificar el funcionamiento del sistema inmótico diseñado. Finalmente se muestran las conclusiones y recomendaciones en base a los resultados obtenidos.

CAPÍTULO I

MARCO REFERENCIAL

Las edificaciones cuyo propósito no sea habitacional presentan una serie de problemáticas distintas a las de un hogar, en primer lugar la seguridad es diferente, ya que existe una alta posibilidad de fuga de gases y líquidos tóxicos o nocivos para la salud, además está lo relacionado con el control de acceso a personas y vehículos no autorizados, debido a que se debe resguardar las áreas críticas; en cuanto al control del consumo de energía eléctrica, este se dificulta, porque los usuarios de estas edificaciones no son los responsables directos del pago de estos servicios y existe la tendencia de un consumo no eficiente de este recurso, lo que genera un gasto importante para las organizaciones que hacen vida en estos edificios. [2]

Lo anterior ha conllevado a que la inmótica, esté ganando terreno y relevancia en los últimos años, ya que mediante la integración de varios elementos de automatización se atacan las problemáticas existentes, entre las más comunes se pueden mencionar: [2]

- Control de acceso
- Alarmas de incendio, fugas de gas, inundación, entre otros
- Climatización interna
- Control de tomacorrientes
- Control de iluminación
- Consumo de energía

Otra característica importante, es que cada sistema inmótico es único, ya que este se diseña especialmente según el uso de cada infraestructura y las actividades que en esta se realiza. Debido a esto, una gran cantidad de empresas han creado aplicaciones

tanto en software como en hardware, creando soluciones casi para cualquier necesidad que se presente en un hotel, centro comercial o inclusive un ambiente académico como la Universidad Central de Venezuela.

La Sala de Computación A “J.J. Martini” ubicada en la Escuela de Ingeniería Eléctrica de la Universidad Central de Venezuela, está destinada como sala de computación y además es utilizada como salón de clases. Las actividades en esta sala se desarrollan durante casi todo el día, permaneciendo las luces, aire acondicionado y estaciones de computación encendidas la mayor parte del tiempo debido a su necesidad, sin embargo es común que por descuido de los ocupantes al momento de salir del aula, algunos de estos elementos permanezcan encendidos, es por esta razón que es necesario fomentar el uso consciente de la energía eléctrica.

PLANTEAMIENTO DEL PROBLEMA

- ¿Hay en la actualidad software gratuito para el desarrollo de inmótica?
- ¿Qué necesidades se pueden cubrir con la implementación de inmótica dentro de un ambiente académico?
- ¿Se puede mejorar la experiencia del docente al dar clases y la de los estudiantes al incorporar la inmótica en un salón de clases?
- ¿Qué componentes son adecuados para ser implementados dentro de un aula de clases de la Universidad Central de Venezuela?

OBJETIVOS

Se establecen los siguientes objetivos a cumplir con el presente Trabajo Especial de Grado.

Objetivo general

Proponer un sistema de supervisión basado en una plataforma de software gratuito para la inmótica de un aula.

Objetivos específicos

1. Establecer los parámetros a controlar por el sistema de supervisión inmótico.
2. Seleccionar el software gratuito a utilizar
3. Seleccionar la plataforma tecnológica donde se ejecutará el software.
4. Seleccionar los componentes adecuados para dotar al sistema con la capacidad de supervisar y/o controlar los parámetros establecidos.
5. Diseñar el sistema de supervisión inmótico en la plataforma seleccionada.
6. Validar el sistema diseñado.

ALCANCE

Este Trabajo Especial de Grado incluye el diseño de un sistema de supervisión basado en un software gratuito para inmótica el cual será aplicado en la Sala de Computación A “J.J. Martini” ubicada en la Escuela de Ingeniería Eléctrica de la Universidad Central de Venezuela con los conocimientos adquiridos en la investigación teórica sobre el tema. Además cuenta con una versión del sistema con una interfaz que permite observar su funcionamiento y control de los parámetros establecidos y la validación del sistema donde se demuestra la capacidad del sistema para supervisar elementos y manejar una estrategia de control.

LIMITACIONES

- Este Trabajo Especial de Grado no incluye la implementación parcial ni total del sistema en las instalaciones de la Sala de Computación A “J.J. Martini” ubicada en la Escuela de Ingeniería Eléctrica de la Universidad Central de Venezuela.
- No se contempló una validación del sistema de supervisión inmótico de manera física.
- No se contempló el desarrollo de una aplicación Android o IOS como parte del sistema de supervisión inmótico.

JUSTIFICACIÓN DEL PROYECTO

El presente Trabajo Especial de Grado en primer lugar permite estudiar la domótica y la inmótica, temas con mucho potencial para desarrollar a nivel académico bien sea en trabajos de grado, líneas de investigación, cátedras y proyectos y que han sido abordados en escasas ocasiones dentro de la Universidad Central de Venezuela. Durante la carrera de Ingeniería Eléctrica se adquieren una serie de conocimientos teóricos y prácticos en diferentes áreas como la electrónica, programación, redes, comunicaciones, sistemas de control, entre otras que de manera pertinente guardan íntima relación con el tema a desarrollar.

El consumo inconsciente de energía eléctrica dentro de la Universidad Central de Venezuela y en general en la edificaciones no destinadas a la vivienda, que trae consigo graves consecuencias para el ambiente, además del impacto económico y sin dejar a un lado el creciente deterioro del sistema eléctrico nacional dan lugar a este esfuerzo académico y surge esta propuesta para dar solución a este problema real. De igual manera, el diseño para esta sala de computación, donde se imparte la formación académica de futuros ingenieros, permitirá desarrollar en estos la competencia de responsabilidad sobre el entorno y sobre todo inculcar la idea de que la tecnología y el

cuidado del ambiente deben ir de la mano para la preservación de los recursos del planeta.

Finalmente se estudia este tema como solución a varias necesidades que se tienen dentro de la Universidad Central de Venezuela, como la seguridad y la gestión energética, tomando como muestra la Sala de Computación A “J.J. Martini” ubicada en la Escuela de Ingeniería Eléctrica y pudiendo extrapolar este sistema a un nivel macro, que permita a la universidad un ahorro de electricidad que repercutiría en una disminución del gasto de este servicio.

ANTECEDENTES DE LA INVESTIGACIÓN

Güette (2015), realizó el Diseño de un Sistema Inmótico Mediante el Uso del Sistema Embebido Intel Galileo, en este trabajo se controlaron las luminarias, el encendido y apagado del aire acondicionado, se monitorea la temperatura y la humedad y se tiene un detector de humo del Laboratorio de Control, Instrumentación y Digitales (LCID) de la Escuela de Ingeniería Eléctrica, Universidad Central de Venezuela. De los resultados obtenidos el autor concluye: que los sistemas embebidos junto con las plataformas IoT son una solución alternativa y viable frente a los sistemas inmóticos comerciales y podrían convertirse en el futuro de los mismos, con la gran diferencia, de que estos sistemas serían abiertos y mucho más flexibles, lo que permitiría disminuir los costos. [2]

El trabajo descrito anteriormente se relaciona con este trabajo especial de grado debido a que se busca desarrollar un sistema inmótico en una plataforma de software abierto, además de buscar una alternativa viable y de menor costo en relación a los sistemas inmóticos comerciales.

Porras (2012), realizó el Estudio y Diseño de un Sistema Inmótico para su Aplicación en el Edificio de Laboratorios de la Universidad Católica Andrés Bello,

donde parte de la descripción de las instalaciones, sus características y de las actividades que se desarrollan en él, para determinar los parámetros a gestionar y la mejor configuración del sistema. Posterior a esto, desarrolló la propuesta del hardware y software para cubrir los requerimientos técnicos como el control de iluminación o detección de movimiento, y de los cuales creó un prototipo para observar su funcionamiento. Concluyendo, que es de suma importancia realizar pruebas a pequeña escala, de cualquier sistema diseñado para depurar errores y corroborar el cumplimiento de los objetivos, aumentando gradualmente los ensayos y agregando las etapas y funciones más complejas. [3]

El trabajo descrito anteriormente se relaciona con este trabajo especial de grado ya que se partió de la descripción y uso del aula SCA “J.J. Martini” de la Escuela de Ingeniería Eléctrica, para generar una propuesta de sistema Inmótico. De igual manera se buscó validar el correcto funcionamiento del sistema a pequeña escala a través de maquetas o simulaciones y que a su vez verifiquen el cumplimiento de los objetivos.

Colon y Fritz (2005), realizaron un Diseño e Implementación de un Sistema Domótico de Manejo Remoto Utilizando Internet y Tecnología Celular, donde utilizaron microcontroladores para construir interfaces que les permitieran adquirir datos de sensores de humo y de apertura de puertas y ventanas, capturar imágenes, y la conexión a internet y la línea telefónica para enviar instrucciones al sistema. Recomendaron una serie de ampliaciones del sistema en el que se observa como sus funciones pueden ampliarse para incluir integrados en la misma plataforma, diferentes servicios. El trabajo descrito anteriormente se relaciona con este trabajo especial de grado ya que en el también existe la conexión vía internet entre el servidor y los diferentes usuarios y de igual manera las funciones aquí establecidas pueden ampliarse al resto de la Escuela de Ingeniería Eléctrica e inclusive al resto de la Universidad Central de Venezuela. [4]

CAPÍTULO II

MARCO TEÓRICO

En este capítulo, se encuentra el resultado de la investigación realizada y en la que se apoya la realización de la propuesta. El marco teórico que se presenta a continuación, permite conocer los conceptos básicos, los complementarios y los específicos, necesarios para la comprensión de desarrollo de este trabajo especial de grado.

EDIFICIOS INTELIGENTES

De acuerdo a la aplicación de tecnología a los edificios se ha dado lugar a varios conceptos que pueden resultar confusos debido a su similitud, incluso a ser utilizados de forma indistinta. Por esto se definen algunos de los más utilizados:

- **Edificio automatizado:** Término que comenzó a utilizarse en el siglo XIX con el desarrollo industrial cuando se automatizaban los procesos productivos y define a un edificio que tenga algún tipo de automatización. Siendo las primeras aplicaciones de la automatización los controles de temperatura y de consumo energético. Un ejemplo característico de estos edificios automatizados son los centros comerciales con sus sistemas de aire acondicionado central, escaleras y puertas mecánicas, control de iluminación, entre otros, controlados desde un cuarto central. [5]

- **Edificio domótico:** Edificio destinado a la vivienda, que tiene un sistema con la capacidad de controlar variables referentes a la gestión energética, seguridad, confort y accesibilidad entre otras aplicaciones domésticas. [5]

- **Edificio inmótico:** Es un edificio al que se le ha proporcionado también la capacidad de controlar varias tareas, como un edificio domótico, pero el sistema no solo está orientado a la calidad de vida si no a la calidad de trabajo. Por ende para

colocar un sistema inmótico en un edificio es importante determinar variables específicas, por ejemplo si se tratase de una biblioteca parámetros como el nivel de humedad son sumamente importantes, pero no para una casa de familia convencional. [5]

• **Edificio inteligente:** Es un concepto en el que a la casa o al edificio se le otorga un grado de inteligencia, por ejemplo, la inteligencia artificial por medio del aprendizaje de la rutina de sus habitantes, con el fin de anticiparse a sus necesidades. [5]

INMÓTICA

“La inmótica es la domótica aplicada a edificios que no están destinados como viviendas” [5]. Consiste en aplicar la tecnología utilizada en domótica a cualquier edificio en general, orientándola a las necesidades propias de los usuarios de ese edificio.

La Asociación Española de Domótica, CEDOM, ofrece la siguiente definición: “La domótica aplicada a edificios no destinados a vivienda, es decir oficinas, hoteles, centros comerciales, de formación, hospitales y terciario, se denomina, inmótica”. [6]

DOMÓTICA

La Real Academia Española, RAE, la define en su diccionario como: “Conjunto de sistemas que automatizan las diferentes instalaciones de una vivienda”. [7]

La CEDOM, define a la domótica como “el conjunto de tecnologías aplicadas al control y la automatización inteligente de la vivienda, que permite una gestión eficiente del uso de la energía, además de aportar seguridad, confort, y comunicación entre el usuario y el sistema”. [1]

Entonces la domótica comprende el uso de tecnología para automatizar diferentes tareas del hogar y mejorando así el estilo de vida de sus habitantes, contribuyendo al mismo tiempo con el ahorro de energía y reduciendo en consecuencia el impacto ambiental ya que los recursos energéticos se aprovechan de mejor manera.

Características de la Domótica:

Las características con las que debe cumplir un sistema de automatización en un edificio inteligente se resumen en las siguientes [2]:

- **Simple y fácil de utilizar.** La interfaz con el usuario debe ser muy simple e interactiva para permitir un aumento de confort. Por ejemplo, control remoto de la iluminación por medio de un dispositivo móvil.

- **Flexible.** “El sistema debe permitir ampliaciones y modificaciones a un bajo costo”.

- **Modular.** “Para evitar que el sistema sea afectado al fallar solo un elemento, éste debe ser modular”.

- **Integral.** “El sistema debe permitir el intercambio de información y la comunicación entre las diferentes áreas de gestión del edificio”.

Estas características son las que resaltan a estos sistemas, ya que están pensados para satisfacer las necesidades de los usuarios al mismo tiempo que cumple con sus objetivos como la gestión de energía. Sin duda alguna, el usuario querrá que el sistema sea sencillo de manejar, casi intuitivo, y que se le puedan agregar más funcionalidades a medida que se acostumbre al mismo.

Aplicaciones de la Domótica:

Su aplicación como se menciona en su definición, consiste en gestionar el consumo energético y aplicaciones de confort, comunicación, seguridad y accesibilidad, entre otras, pudiéndose incluir combinación de los siguientes [3]:

- **Gestión de energía.** Esta rama de la domótica se orienta al consumo eficiente de energía en los elementos controlados por el sistema racionalizando su uso a través de temporizadores, termostatos y otros mecanismos.

- **Gestión del confort.** El desarrollo de esta gestión le permite al usuario contar con ciertas comodidades como control automático de las luces, calefacción, accesos, persianas y muchos otros.

- **Gestión de la seguridad.** Se incluyen tres áreas que por lo general requieren de sistemas distintos, como lo es la seguridad en los bienes, a través de alarmas, detectores o simuladores de presencia, entre otros. La seguridad a las personas, con tele-asistencia o mensajes de seguridad por ejemplo y control de incidentes y averías, pudiendo detectar fallos en los sistemas del hogar como una fuga de gas o agua, incendios y más.

- **Gestión de las comunicaciones.** Facilita el intercambio de información entre el usuario y el sistema, por ejemplo enviar notificaciones al celular de cuando una alarma se active, monitorear y acceder al sistema desde internet y mucho más.

Elementos básicos de un sistema domótico

Un sistema se compone básicamente de los siguientes elementos [5] - [8]:

- **Dispositivo de entrada:** Sensor, control remoto, teclado o cualquier dispositivo que envíe información al nodo.

- **Sensores:** componentes capaces de detectar una magnitud que puede ser física o química y son los que capacitan al sistema para detectar todo tipo de información como presencia, humedad, gases, entre otros.

- **Actuadores:** Son aquellos dispositivos que al recibir una señal eléctrica pueden producir una acción, por ejemplo motores abran o cierran la puerta del garaje.

- **Interfaces:** Algunas veces la señal entre los elementos de control y los actuadores no son compatibles, por lo que requieren de una interfaz entre ellas. Como ejemplo, las etapas de conmutación de los transistores utilizados como interruptores en los circuitos electrónicos.

- **Infraestructura:** Es la manera en que se lleva la información de los sensores al sistema de control y de este a los actuadores.

- **Unidad de control:** Recibe la información de los sensores a través de la infraestructura y envía la señal de activación a los actuadores. Además, es la comunicación del usuario con el sistema; algunas unidades de control comunes son tableros con pantallas táctiles para acceder y controlar todo el sistema. La unidad de control puede ser una sola, puede ser una sola, que concentre toda la información de sensores y se comunique directamente con los actuadores, o puede que existan varios elementos de control con sus sensores y actuadores, y se comuniquen entre ellos formando al sistema.

- **Software de gobierno:** Software que controla todo el hardware del sistema y se comunica con él, anteriormente se utilizaba el lenguaje de programación C para desarrollar este software, pero en la actualidad se utilizan lenguajes como Java.

- **Nodo:** Las unidades del sistema capaces de recibir información y procesarla, y en caso de aplicarse, comunicarla a otras unidades dentro del sistema. Se pueden considerar a los nodos como una caja que incluirá a las unidades de control, que implican a su vez un software de gobierno, y las interfaces para que los actuadores puedan acoplarse a ellas.

Topologías de red de un sistema domótico

La topología de red consiste en la distribución física de los elementos y su conexión al medio de comunicación. Tal como se puede observar en la Figura 1 las topologías de red empleadas para los sistemas domótico son [2] - [5]:

- **Topología en estrella.** Los elementos se unen entre sí por el elemento principal del controlador. Teniendo como principal ventaja que se pueden agregar nuevos elementos de manera sencilla y si falla alguno que no sea el principal no afecta el sistema, pero su desventaja es que si falla el elemento central, fallan todos.

- **Topología en bus.** Todos los elementos comparten una única línea de comunicación, utilizando generalmente una identificación. Permite añadir fácilmente nuevos elementos, las fallas de un elemento no afecta al resto y el cableado es muy poco. La desventaja es que en los nodos requieren inteligencia de procesamiento.

- **Topología en anillo.** Los elementos se interconectan formando un anillo cerrado. Esta topología reduce los requerimientos de cableado e implica que la información pasa por todos los elementos, pero tiene como desventaja que una falla en un único punto hará fallar el sistema entero y es complicado agregar nuevos elementos.

- **Topología en árbol.** Es una variante de la topología en estrella. Los nodos están conectados a un controlador central que se encarga del tráfico de la red. La mayoría de los elementos se conectan a un controlador secundario el cual se conecta a su vez con el controlador principal.

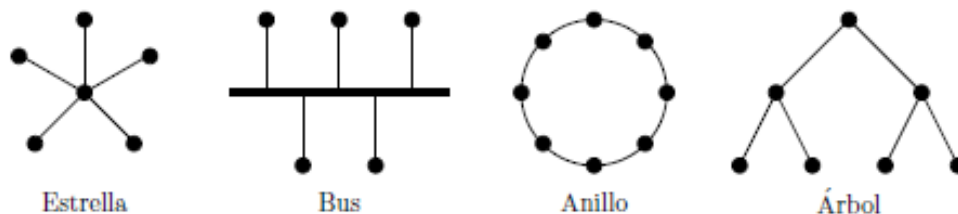


Figura 1: Topologías de red. [2]

Arquitectura de un sistema domótico

La arquitectura de los sistemas se clasifica de acuerdo a la manera en que están organizados sus componentes, unas de las principales son [9]:

- **Arquitectura centralizada:** Este sistema está organizado de tal forma que el controlador sea el eje central del sistema, recibiendo la información de los sensores, analizándola, y enviando una orden a los actuadores, según la configuración, o la información que reciba por parte del usuario, un esquema de este tipo de arquitectura se puede observar en la Figura 2.

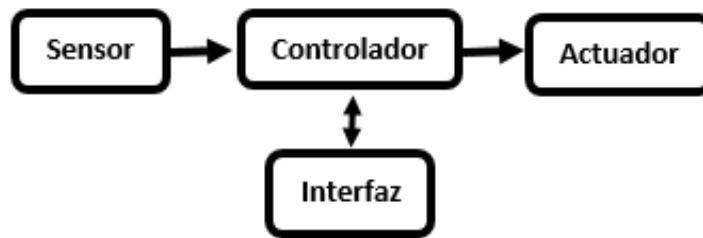


Figura 2: Arquitectura Centralizada

- **Arquitectura Descentralizada:** Como se puede notar en la Figura 3 en un sistema descentralizado existen varios controladores, conectados a sensores y actuadores, quienes a su vez están interconectados por medio de un “Bus”. Este modelo nació de la necesidad de tener mejor acceso a ciertos dispositivos y a causa de la existencia de diferencia en los protocolos y características de los distintos fabricantes.

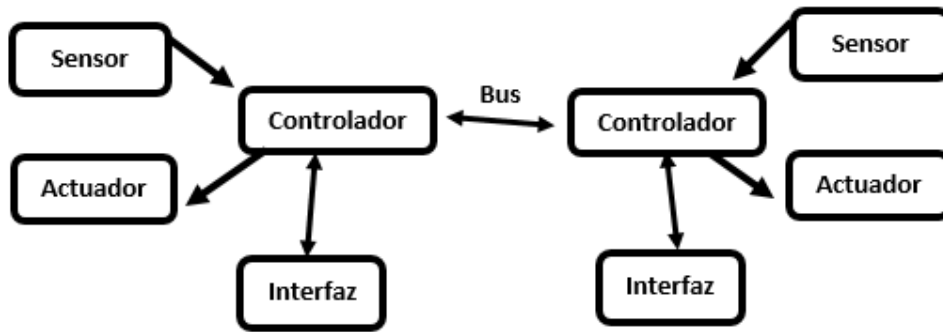


Figura 3: Arquitectura Descentralizada

• **Arquitectura distribuida:** Este tipo de arquitectura se diferencia por tener sensores y actuadores que son a su vez controladores, es decir son capaces de analizar la información, y están conectados a través de un “Bus” central como se muestra en la Figura 4.

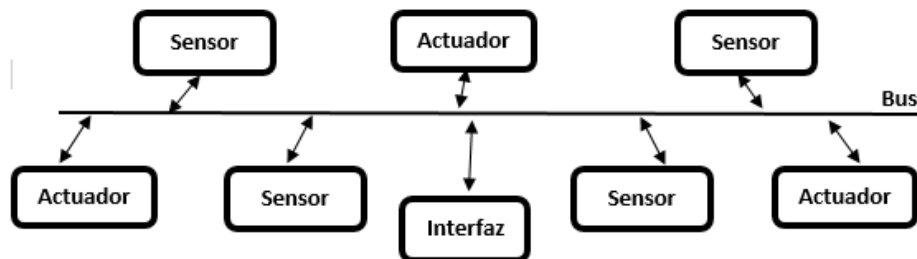


Figura 4: Arquitectura Distribuida

• **Arquitectura Híbrida / Mixta:** En un sistema basado en este tipo de arquitectura se combinan las arquitecturas de los sistemas distribuidos, centralizados o descentralizados. Por lo que puede disponer de un controlador central o varios controladores descentralizados, los dispositivos de interfaces, sensores y actuadores pueden también ser controladores y procesar la información (que captan ellos mismos u otro sensor), según el programa, o la configuración, y pueden actuar de acuerdo a ella, como por ejemplo, enviándola a otros dispositivos de la red, sin que necesariamente pase por un controlador.

PROTOCOLO DE COMUNICACIÓN

A continuación se listan algunos de los protocolos de comunicación usados en la implementación de domótica a nivel mundial:

- **BACnet:** Protocolo norteamericano para la automatización de viviendas, este fue desarrollado con el patrocinio de una asociación de fabricantes de equipos de acondicionadores de aire y calefacción. Esta iniciativa se originó con el objetivo de crear un protocolo abierto para interconectar estos equipos y gestionar la energía de manera eficiente. Define un conjunto de reglas de hardware y software que permiten la comunicación entre dos dispositivos independientemente del protocolo que utilice cada uno. [10]

- **LonWorks:** “En 1992 fue presentado por la Corporación Echelon, diseñado para implementar cualquier sistema de control. Está pensado para emular la red neuronal donde los cientos o miles de nodos estén conectados a una red. El protocolo de esta tecnología, llamado LonTalk, se desenvuelve en las siete capas del modelo OSI, con las interconexión física (capa 1: física), acceso al medio (capa 2: enlace), entrega de mensajes (capa 3: red), fiabilidad punto a punto (capa 4: transporte), control (capa 5: sesión), interpretación de datos (capa 6: presentación) y compatibilidad de aplicación (capa 7: aplicación).” [11]

- **EHS:** “European Home System es un protocolo estándar creado en Europa (1984) por la comisión Europea para implementar la tecnología domótica en el mercado residencial. De resultado se obtuvo la especificación del EHS en 1992, la cual está basada en la topología de niveles OSI. Estos niveles son: físico, enlace de datos y aplicación. Al inicio, estuvieron involucrados los fabricantes de electrodomésticos más importantes, las empresas eléctricas, las de telecomunicaciones y los fabricantes de equipamiento eléctrico. La idea fue crear un protocolo abierto que dejara cubrir las necesidades de interconexión de los productos de los fabricantes y proveedores de los servicios. El objetivo de la EHS es de cubrir las necesidades de automatización de la mayoría de los hogares europeos. Los propietarios no se podían dar el lujo de usar

sistemas más potentes o caros; por ejemplo, “Lonworks”, “EIB” o “Batibus”, por la mano de obra especial que se requiere para su instalación.” [12]

- **X-10:** “Diseñado en Escocia entre los años 1976 y 1978 con el objetivo de transmitir datos por las líneas de baja tensión a muy baja velocidad (60 bps en EEUU y 50 bps en Europa) y costes muy bajos. Al usar las líneas de eléctricas de la vivienda, no es necesario tender nuevos cables para conectar dispositivos. El protocolo X-10 usa una modulación muy sencilla, comparado con las que usan otros protocolos de control por ondas portadoras. El transceiver X-10 monitorea los pasos por cero de la onda senoidal de 50 Hz típica de la alimentación eléctrica (60 Hz en EEUU) para insertar un instante después una ráfaga muy corta de señal en una frecuencia fija.” [12]

ESTÁNDARES USADOS EN DOMÓTICA

De manera representativa a continuación se describe una pequeña selección de modelos de sistemas de automatización que existen en la actualidad:

- **KNX:** Único estándar internacional aprobado para el control de viviendas y edificios. Su origen se da gracias a la unión de tres estándares europeos: BatiBus, estándar que mediante un bus abierto permite a cualquier fabricante desarrollar el acceso de sus equipos a él; EHS, sistema completo de funciones domóticas, modular, expansible y configurable, basado en que cada unidad se conecta obteniendo una dirección red y comienza a buscar otras unidades a las que pueda interesarles o le interesen para sus funciones; y EIB. [13]

- **ZigBee:** Diseñado por la Alianza Zigbee en la que participan empresas como HP y Motorola, es un conjunto de protocolos definidos por el estándar 802.15.4 de la IEEE, en el cual se define acceso al medio e intercambio de mensajes a través de comunicaciones inalámbricas, en la capa de red, entre aplicaciones. Entre sus principales características se destaca el bajo consumo, bajas tasas de transferencia con una velocidad de 20 – 250kB/s, alcances de 10 – 75m. Permite el control de hasta 2^{16}

dispositivos por lo que se usó en redes inalámbricas de sensores, que ameritan el despliegue de grandes cantidades de sensores para monitorear parámetros de interés. [14]

• **Z-Wave:** Estándar abierto para la comunicación inalámbrica de elementos inteligentes diseñado por la alianza Z-Wave en el año 2013, este describe el transporte de mensajes de Z-Wave a través de redes IP. Esto incluye una implementación de referencia denominada Z/IP Gateway, y la especificación API completa describe como construir una aplicación on-top. En este estándar todo el tráfico IP viaja a través de un túnel seguro TLS 1.1, lo que elimina la vulnerabilidad hacia y desde la nube. Actualmente el estándar Z-Wave es utilizado por más de 600 empresas y cuenta con más de 21000 productos disponibles en el mercado. [15]

NORMATIVAS PARA LA DOMÓTICA

Para un sistema de automatización se aplican las especificaciones, normas y regulaciones para cada tecnología utilizada y tipo de instalación, esto debido a que en la actualidad no existe ningún acuerdo universal sobre proyectos de domótica e inmótica. Además de esto la escogencia de una u otra alternativa dependerá de las necesidades a cubrir, las dimensiones, infraestructura, entre otras. [3]

Hay países que de acuerdo a sus normativas de regulación interna, sugieren una serie de normas para la instalación de sistemas de automatización, como ejemplo España, que recomienda varias pautas para un buen funcionamiento, además de sugerir las normas con las especificaciones propias de cada tipo de tecnología utilizada, todo esto en la Guía Técnica de Aplicación BT-51. [8]

Existen algunos estándares abiertos internacionales aprobados para todas las aplicaciones de viviendas y edificios, como por ejemplo KNX, el cual fue aprobado como estándar internacional bajo el ISO/IEC 14543-3 luego de ser aprobado en primera instancia como un estándar regional en los estándares europeos CENELEC EN500090

y CEN EN 13321-1. Este estándar cuenta con una única herramienta puesta en marcha que permite el funcionamiento de distintos equipos, independientemente del fabricante. [16]

También existen estandarizaciones que incluyen las especificaciones y documentación suministrada por el fabricante de cada tecnología, como ejemplo Zigbee, que esta descrita en el IEEE 802.15.4. [17]

SOFTWARE ABIERTO PARA DOMÓTICA

El IoT pasó de ser un término de moda a una realidad que presenta una expansión muy rápida, debido a esto se cuenta cada día con un mayor número de dispositivos para la automatización, protección o vigilancia. Al mismo tiempo, surge en los usuarios la preocupación debido a la seguridad y privacidad que implica tener estos elementos en su hogar. Los usuarios desean controlar quien tiene acceso a los sistemas vitales que controlan los aparatos y que registran cada momento de su vida cotidiana, ya que nos encontramos en una época donde la nevera puede llamar por teléfono a la lavadora es importante restringir el acceso de estos dispositivos para comunicarse con el exterior.

Por estas preocupaciones han surgido una serie de softwares gratuitos capaces de comunicarse con una gran variedad de dispositivos de automatización, a pesar de que la mayoría de estos presentan componentes propietarios, estos programas o plataformas permiten crear un centro de control y configuración de estos dispositivos de automatización, lo que da respuesta a la interrogante ¿Hay en la actualidad software gratuitos para el desarrollo de inmótica?, planteada en el Capítulo I. Entre una gran lista destacan:

- **Home Asistant:** Plataforma de automatización del hogar de código abierto, y está diseñado para ser implementado fácilmente en casi cualquier máquina que pueda ejecutar Python 3, a partir de una Raspberry Pi. Se integra con un número de código

abierto, así como ofertas comerciales, lo que le permite enlazar, por ejemplo, IFTTT, la información del tiempo, o su dispositivo de Amazon Eco, a los controles de las cerraduras a las luces a incluso un notificador de línea de comandos. [18]

- **Domoticz:** Sistema domótico con una muy amplia biblioteca de dispositivos compatibles, que van desde estaciones meteorológicas a los detectores de humo a los mandos a distancia, con un gran número de terceros adicionales integraciones documentados en la página web del proyecto. Está diseñado con una interfaz HTML 5, por lo que es accesible desde navegadores de escritorio, así como la mayoría de los teléfonos inteligentes modernos, y es ligero, que se ejecuta en muchos dispositivos de baja potencia, como el Raspberry Pi. [19]

- **PiDome:** Plataforma de automatización del hogar desarrollado exclusivamente para Raspberry Pi. Proporciona soluciones para los usuarios finales, así como para los desarrolladores y aficionados. Este cuenta actualmente con el protocolo de transporte de datos MQTT, servicio de mensajería PushBullet y SMS, además del soporte de algunas marcas comerciales como: Philips Hue, My Sensors y Unipi. [20]

- **OpenHAB 2.0** [21]: Herramienta de automatización del hogar, que cuenta con una gran comunidad de usuarios debido al número de dispositivos e integraciones compatibles. Escrito en Java, es perfectamente compatible para trabajar bajo cualquier sistema operativo, bien sea Windows, Linux o Mac OS. OpenHAB también cuenta con aplicaciones nativas Android y IOS, para el control de dispositivos, así como una herramienta de diseño para crear su propia interfaz de usuario. Diseñado para ser independiente del dispositivo, cuenta con el soporte para cientos de estos, debido a esto a los desarrolladores se le facilita el añadir sus propios dispositivos o plugins para el sistema. Entre la lista de marcas, estándares y protocolos con los que cuenta OpenHAB se puede mencionar: KNX, MQTT, Amazon, Oceanic, Siemens, LG, Pioneer, Panasonic, Samsung, Tesla, Logitech, Yamaha, ZWave, Belkin, Modbus, HTTP, Bticino, Bluetooth, entre muchos otros.

OpenHAB es un software de código abierto cuyo fin es integrar diferentes sistemas y tecnologías de automatización del hogar en una única solución que permite

la configuración de reglas de automatización y ofrece interfaces de usuario uniformes. Además es un software altamente modular, lo que significa que la instalación base (el "tiempo de ejecución") se puede extender a través de diferentes tipos de "complementos", ya sea para comunicarse con la nueva solución de domótica o para ofrecer un nuevo tipo de interfaz de usuario.

Es una solución escrita completamente en Java y se basa en el marco de Eclipse SmartHome, el cual permite establecer una diferencia entre la vista física y la vista funcional en el sistema. Si bien la vista física es necesaria para la configuración, la resolución de problemas, etc., el lado funcional cubre la información que es importante para las aplicaciones, como las interfaces de usuario y la lógica de automatización.

OpenHAB ofrece una visión en conjunto a través de las diferentes interfaces disponibles, de cuatro elementos fundamentales, como lo son las Cosas (Things), Canales (Channels), Artículos (Items) y Enlaces (Links). En primer lugar las cosas son las entidades que se pueden agregar físicamente a un sistema y que pueden proporcionar muchas funcionalidades a la vez. Es importante tener en cuenta que las cosas no tienen por qué ser dispositivos, sino que también pueden representar un servicio web o cualquier otra fuente manejable de información y funcionalidad. Estas proporcionan su funcionalidad a través de un conjunto de canales, los cuales pueden considerarse como una declaración de lo que puede ofrecer. Depende de la configuración individual, cuál de los canales se usa activamente a través de los artículos.

Los artículos representan la funcionalidad que utilizan las aplicaciones, como interfaces de usuario o lógica de automatización. Estos tienen un estado y pueden recibir comandos. La unión entre Cosas y Artículos son Enlaces, los cuales son asociaciones entre un canal de cosa y un artículo. Si un canal está vinculado a un elemento, está "habilitado", lo que significa que la funcionalidad que representa el

elemento se maneja a través del canal determinado. Los canales se pueden vincular a varios artículos y los artículos se pueden vincular a múltiples canales.

Cabe destacar que OpenHAB cuenta con tres interfaces de usuarios diferentes, la principal es *Paper UI*, es la interfaz descargada por defecto al momento de instalar OpenHAB y tiene la principal característica que es en ella donde se realiza la descarga, instalación y desinstalación de las demás interfaces de usuario y principalmente de todos los complementos disponibles. Es en ella donde se puede obtener las interfaces *Basic UI* configurada mediante Eclipse SmartHome y la segunda *HabPanel UI* cuya configuración es realizada con HTML AngularJS.

ENTORNO DE PROGRAMACIÓN

Durante el desarrollo de este trabajo especial de grado se seleccionó como lenguaje de programación Java así como un ambiente de desarrollo integrado como lo es el software Eclipse Smarthome, haciendo necesaria la inclusión de información referente a ellos a continuación.

- **Lenguaje de programación Java:** Creado por Sun Microsystems y preferido por gran parte de los desarrolladores de software. Java se considera como la evolución directa de del lenguaje de programación C/C++, siendo este último el lenguaje de programación más utilizado, difundido y en el cual se encuentran programadas un enorme porcentaje de software a nivel mundial. Entre las principales características de este lenguaje se encuentra que se puede desarrollar en una plataforma (sistema operativo) y utilizarlo en otra además de esto gran parte de las aplicaciones, especialmente las web, y hasta para los equipos de telefonía celular o dispositivos en general. [22]

- **Eclipse Smarthome:** Es un IDE gratuito, desarrollado por The Eclipse Foundation en la que trabajan en conjunto más de veinte empresas relacionadas con la tecnología, automatización y programación, como por ejemplo IBM o Siemens.

Considerado como un proyecto en desarrollo, Eclipse Smarthome se basa en los servicios y APIs para la manipulación de datos, motores de reglas, interfaces de usuario declarativos y la gestión de persistencia, todo esto con el fin de simplificar el desarrollo de sistemas domótico con OpenHAB mediante del uso de secciones mostradas en la Figura 5 que facilitan la programación de los mismos. En este se puede escribir en el lenguaje Java, que permite escribir, compilar y depurar programas. Además de esto cuenta con una serie de extensiones para los sistemas pertinentes, protocolos o estándares, lo que permite la creación de soluciones inteligentes para el hogar que permiten la integración de diferentes protocolos o normas, que lo hace ideal para la incorporación a un servicio en la nube o IoT. [23]

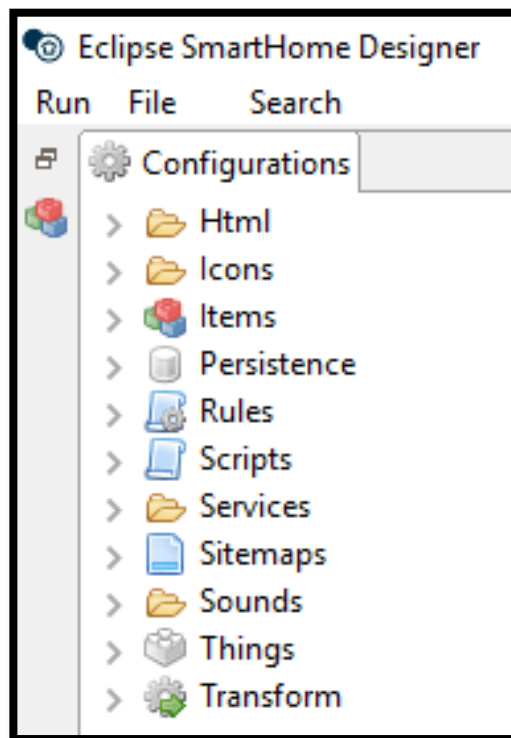


Figura 5: Secciones del Eclipse Smarthome

CAPÍTULO III

DISEÑO DEL SISTEMA INMÓTICO

De acuerdo con los objetivos específicos planteados para el proyecto y presentados en el Capítulo I, la Tabla 1 refleja la metodología empleada para cumplir con las actividades correspondientes a cada uno de ellos. Posterior a esto se fueron cumpliendo con ellas de manera progresiva en las diferentes etapas que se muestran en este capítulo hasta cumplir con cada objetivo específico.

Tabla 1: Actividades por objetivo específico

OBJETIVOS ESPECÍFICOS	ACTIVIDADES
Establecer los parámetros a controlar por el sistema de supervisión inmótico.	<ul style="list-style-type: none">• Descripción del aula y distribución.• Descripción de las actividades realizadas en el aula.• Determinación de los parámetros a ser controlados.
Seleccionar el software gratuito a utilizar.	<ul style="list-style-type: none">• Recopilación de información sobre diferentes softwares gratuitos que pueden ser destinados a inmótica.• Revisión de la información de los softwares gratuitos recolectada.• Determinación del software gratuito a utilizar.
Seleccionar la plataforma tecnológica donde se ejecutará el software.	<ul style="list-style-type: none">• Recopilación de información sobre las distintas plataformas donde se puede ejecutar el software seleccionado.• Revisión de la información recolectada sobre las plataformas donde se puede ejecutar el software seleccionado.• Determinación de la plataforma donde se ejecutará el software seleccionado.
Seleccionar los componentes adecuados para dotar al sistema con la capacidad de supervisar y/o controlar los parámetros establecidos.	<ul style="list-style-type: none">• Recopilación de información sobre elementos soportados por el software seleccionado que dotan al sistema la capacidad de supervisar y/o controlar los parámetros establecidos.• Revisión de información sobre elementos soportados por el software seleccionado que dotan al sistema la capacidad de supervisar y/o controlar los parámetros establecidos.• Selección de los elementos soportados por el software seleccionado que dotan al sistema la capacidad de supervisar y/o controlar los parámetros establecidos.

... Continuación de la Tabla 1

OBJETIVOS ESPECÍFICOS	ACTIVIDADES
Diseñar el sistema de supervisión inmótico en la plataforma seleccionada.	<ul style="list-style-type: none"> • Determinar el lenguaje de programación según el software seleccionado. • Determinar el IDE según el software seleccionado. • Escritura del código para el sistema de supervisión.
Validar el sistema diseñado.	<ul style="list-style-type: none"> • Determinar el método de validación. • Determinar elementos actúen de manera similar a los previamente seleccionados para la propuesta y logren verificar el correcto funcionamiento del sistema desarrollado. • Realizar la prueba de validación del sistema.

Con las diferentes etapas el sistema de supervisión inmótico que se desarrollan en el presente capítulo, tal como se muestra en la Figura 6 se obtuvo una propuesta con un software de control OpenHAB, que cuenta con varias interfaces de usuario como la Basic UI y Paper UI, además de un conjunto de canales de comunicación capaces de enviar y/o recibir información de los sensores (temperatura, presencia, estado puertas, estado de ventanas y humo), actuadores (interruptores, motores de persianas automáticas y pantalla de proyección), equipos industriales (PLC), servicios en el IoT (aplicaciones Android y IOS), placas de desarrollo (Raspberry Pi, Arduino e Intel Galileo), teléfonos celulares y equipos electrónicos (proyector de video).

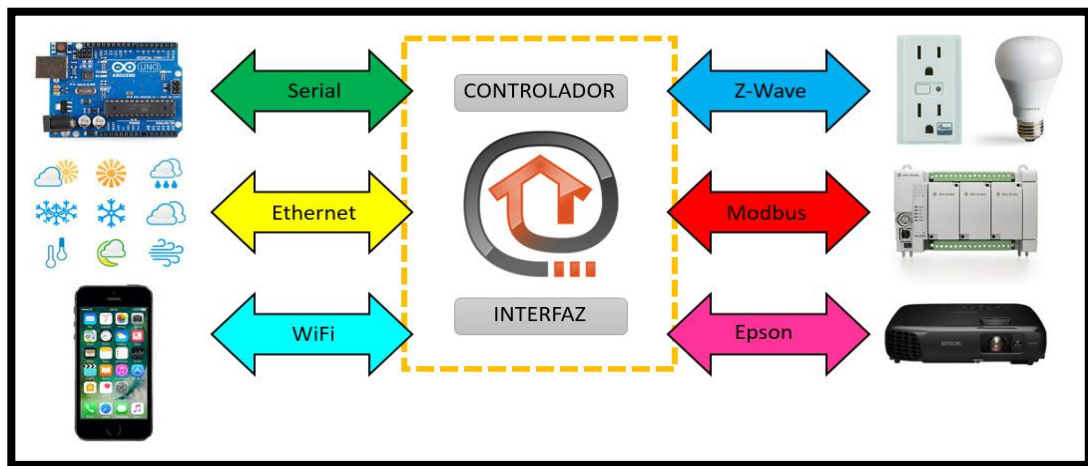


Figura 6: Esquema del sistema inmótico propuesto

DIAGNÓSTICO DEL ÁREA DE TRABAJO

El primer paso para diseñar un sistema inmótico es el diagnóstico de las principales características del área donde se ejecutará el proyecto. Cabe resaltar que si el área a automatizar no se ha construido se facilita la instalación del sistema, debido a que su implementación se considera desde un inicio. Para el caso contrario, se debe estudiar el tipo de tecnología a utilizar que mejor se adapte a los requerimientos de los usuarios y a las características del inmueble.

Para el punto central de esta propuesta, que es el aula 311 o Sala de Computación A (SCA) de la Escuela de Ingeniería Eléctrica de la UCV, cuenta con un área de 32 m² y una altura de 3 m. Como su nombre lo indica está orientada a ser una sala de computación, aunque en la actualidad y debido a diversos factores no cuenta con los equipos de computación necesarios para cumplir a cabalidad esta función, pero se espera que a corto plazo el aula pueda ser acondicionada con los equipos necesarios para los cuales fue diseñada por lo que la propuesta del sistema inmótico no va a excluir estos elementos del diseño.

La SCA tiene una capacidad instalada para albergar hasta veintiuna computadoras, dado que la misma tiene una cantidad de 21 puntos de red, 20 para las estaciones de computación y una adicional para el docente, y 20 tomacorrientes dobles, además de otros 4 tomacorrientes dobles a disposición de los usuarios del aula. Adicional a esto la SCA dispone de un aire acondicionado marca Haier de 18.000 BTU, pantalla de proyección (manual), persianas verticales (manuales) en sus ventanas y 6 luminarias, cada una con 4 lámparas fluorescentes, distribuidas en 3 circuitos eléctricos, adicionalmente cuenta con un par de luminarias tipo domo dirigidas hacia el pizarrón con un circuito independiente.

PARÁMETROS DE LA PROPUESTA DE SISTEMA INMÓTICO

Con base a la información obtenida sobre la SCA, las actividades que se realizan en esta y las mejoras que un sistema inmótico ofrece, se determinaron qué parámetros son los más convenientes para incluir en un sistema inmótico. Los resultados de la evaluación realizada arrojaron los siguientes parámetros con los que cuenta la propuesta:

- ✓ Luces (2)
- ✓ Tomacorrientes (2)
- ✓ Aire acondicionado (1)
- ✓ Puerta (1)
- ✓ Ventanas (1)
- ✓ Proyector de video (1)
- ✓ Pantalla de proyección automática (1)
- ✓ Persianas automáticas (1)
- ✓ Control de temperatura (1)
- ✓ Detector de presencia (1)
- ✓ Detector de humo (1)

CRITERIOS DE DISEÑO PARA EL SISTEMA INMÓTICO

El diseño del sistema propone una solución práctica con base a los conocimientos y destrezas adquiridos en el proceso de aprendizaje durante la carrera y en los complementarios adquiridos durante la investigación realizada. En la fase de implementación del sistema es donde interviene el personal especialista que determinará ciertos parámetros, por consiguiente, algunas características que caen en esta etapa serán explicadas y quedarán como recomendaciones.

Dada la falta de planos eléctricos, se asumirá que el cableado eléctrico para los sistemas de iluminación y alimentación eléctrica es de 120V AC, mientras que para la unidad de aire acondicionado se consideró de 208V AC. Además el sistema diseñado se presenta de una manera conceptual y será desarrollada una versión sencilla que permita observar su funcionalidad.

SELECCIÓN DEL SOFTWARE ABIERTO PARA EL SISTEMA INMÓTICO

El software seleccionado para el desarrollo de este trabajo especial de grado fue OpenHAB 2.0, esta decisión fue tomada con base a la Tabla 2. Los recuadros en verde [▲] indican disponibilidad, los de color rojo [▼] indica que no se encuentra disponible y lo de amarillo [±] que se encuentra en desarrollo [18]- [19]- [21]- [20]:

Tabla 2: Criterios para selección de Software.

Criterios		Home Assistant	Domoticz	OpenHAB 2.0	PiDome
Servidor	Windows	▼	▲	▲	±
	Linux	▼	▲	▲	±
	MacOS	▼	▲	▲	▼
	Rapsberry Pi	▲	▲	▲	▲
	Arduino	▲	▼	▼	▼
Clientes	Windows	▼	▲	▲	±
	Linux	▼	▲	▲	±
	Rapsberry Pi	▼	▲	▲	▲
	MacOS	▼	▲	▲	±
	Android	▼	▼	▲	±
	Windows Phone	▼	▼	▲	±
	IOS	▲	▼	▲	▼
Comunicaciones	Bluetooth	▼	▼	▲	▼
	eBUS	▼	▼	▲	▼
	HTTP	▼	▼	▲	▲
	KNX	▼	▼	▲	▼
	Modbus	▼	▼	±	▼
	Serial	▼	▼	▲	▼
	TCP/UDP	▼	▼	▲	▼

...Continuación de la Tabla 2.

Criterios		Home Assistant	Domoticz	OpenHAB 2.0	PiDome
Compatibilidad con marcas/servicios	All Play	▼	▼	▲	▼
	Amazon	▲	▼	▲	▼
	Apple TV	▲	▼	▼	▼
	Astro	▼	▼	▲	▼
	Bosch	▼	▼	±	▼
	Bticino	▼	▼	±	▼
	Chromecast	▼	▼	▲	▼
	D-Link	▼	▼	±	▼
	DigitalSTROM	▼	▼	▲	▼
	Eco	▼	▼	▲	▼
	Ecobee	▲	▼	±	▼
	Email	▼	▼	±	±
	EnOcean	▼	▲	±	▼
	Epson	▼	▼	±	▼
	Fatek PLC	▼	▼	±	▼
	Homematic	▼	▼	▲	▼
	Hue	▲	▼	▲	▲
	IFTTT	▲	▼	▲	▼
	KODI	▲	▼	▲	▲
	Koubachi	▼	▼	±	▼
	LG TV	▼	▼	▲	▼
	Max!	▼	▼	▲	▼
	MQTT	▲	▼	±	▲
	My Sensors	▲	▼	▼	▲
	Nest	▲	▼	±	▼
	Panasonic	▼	▼	±	▼
	Pioneer	▼	▼	▲	▼
	Pushbullet	▲	▼	▼	▲
	Siemens Logo	▼	▼	±	▼
	Samsung	▼	▼	▲	▼
Tesla	▼	▼	▲	▼	
UniPi	▼	▼	▼	▲	
Weather	▼	▲	▲	▲	
Wemo	▲	▼	▲	▲	
Z-Wave	▼	▲	▲	±	
Información y soporte disponible	±	▲	▲	±	

PLATAFORMA TECNOLÓGICA

Para los efectos de esta propuesta, se seleccionó trabajar con un computador personal bajo el sistema operativo Windows 10 de 64 bits, esto debido a que OpenHAB cuenta con la posibilidad de ejecutarse en él sin mayor dificultad. Posterior a esto, se procedió a generar el Manual de Configuración y Uso que reúne información más detallada tal como su nombre lo indica sobre la configuración y uso en la plataforma seleccionada, este manual se encuentra en el Anexo A.

COMPONENTES PARA EL SISTEMA INMÓTICO

Debido a que la Ciudad Universitaria de Caracas, sede de la Universidad Central Venezuela, que fue declarada Patrimonio de la Humanidad por la UNESCO en el año 2001 [24] no se pueden hacer modificaciones a la infraestructura y para dar respuesta a la interrogante ¿Qué componentes son adecuados para ser implementados dentro de un aula de clases de la Universidad Central de Venezuela? presentada en el Capítulo I, los componentes que se comunican de forma inalámbrica al servidor los adecuados para ser implementados dentro de la UCV. Es por esto que para la selección de los componentes se buscó los canales actualizados por OpenHAB que cumpliera con esta característica.

También se consideró los catálogos de productos, buscando el que tuviera diversidad de elementos disponibles en diversas áreas como iluminación, energía, control y seguridad; con el fin de que con la utilización de un canal se pudiera abarcar la mayoría del hardware necesario cuando llegase el momento de implementar el sistema diseñado. El resultado de esta selección es resumido a continuación en la Tabla 3: [25] - [26]

Tabla 3: Componentes Seleccionados

Categoría	Canal	Marca	Componente(s)
Iluminación	Z-Wave	GE	Z-Wave In-wall smart switch 12722
Tomacorriente	Z-Wave	GE	Z-Wave In-wall smart duplex outlet 12721
Persianas / Pantalla de Proyección	Z-Wave	<u>Motor:</u> Somfy <u>Controlador:</u> Somfy	<u>Motor:</u> Sonesse 30 RTS #1001524 <u>Controlador:</u> Z-Wave digital motor interface
Detector de movimiento	Z-Wave	GoControl	Z-Wave Wireless IR motion detector wapirz-1
Detector de incendio	Z-Wave	First Alert	Smoke/Fire and carbon monoxide alarm, white zcombo-g
Puerta/Ventana	Z-Wave	Sensitive	Strips-MaZw Magnetic Door\Window Sensor Strips-MaZw
Aire Acondicionado / Termostato / Sensor de Temperatura	Z-Wave	Stelpro	KI Z-Wave Plus Thermostat for Electric Baseboards and Convectors STZW402WB +
Proyector	Epson	Epson	Proyector PowerLite S27 SVGA 3LCD
Interfaz Pc - Red Z-Wave	USB 2.0	Nortek	Lineal QUICKSTICK USB Combo HUSBZB-1

ESTRUCTURA DEL SISTEMA INMÓTICO

Arquitectura

La arquitectura seleccionada para este diseño es la centralizada, ya que los sensores, actuadores e interfaz estarán directamente conectados directamente al nodo controlador esto debido a que OpenHAB se encuentra en él, siendo el software de gobierno del sistema inmótico.

Topología

La topología elegida corresponde con la de tipo estrella, ya que todos los elementos anteriormente seleccionados facilitan la configuración de los mismos, debido a que se comunican directamente con el nodo, por lo cual, según la definición de *Topología* dada en el Capítulo I, cuenta con la unidad de control, que implica a su vez un software de gobierno y las interfaces para que los actuadores puedan acoplarse a ellas. Además de que esta topología da la ventaja de que si falla un elemento, que no sea el controlador, no afectaría al resto del sistema.

ESTRATEGIA DE CONTROL DEL SISTEMA INMÓTICO PROPUESTO

La estrategia de control del sistema da respuesta a la interrogante ¿Qué necesidades se pueden cubrir con la implementación de inmótica dentro de un ambiente académico?, planteada en el Capítulo I, fijando una serie de lineamientos que atienden necesidades en la gestión de energía, la gestión de seguridad y la gestión de confort. Estos lineamientos son los siguientes:

Gestión de energía

- Monitoreo y control de las luces (Encendido/Apagado).
- Control de la habilitación de los tomacorrientes de las estaciones de computación.
- Control de la unidad de aire acondicionado (Encendido/Apagado).
- Fijar una hora en que se apague automáticamente las luces y la unidad de aire acondicionado. Esta hora se configuró a las 9:00 pm, hora en la que ya no debe haber actividades en el aula.
- Dar una alerta al administrador si se enciende la unidad de aire acondicionado y la puerta está abierta o alguna de las ventanas se encuentra abierta.

Gestión de seguridad

- Monitoreo de presencia.
- Monitoreo del estado de la puerta de entrada (Abierto/Cerrado).
- Monitoreo del estado de las ventanas (Abierto/Cerrado).
- Alarma de incendio.
- Encender las luces y enviar un mensaje de alerta al administrador si se activa la alarma de incendios.
- Dar un mensaje de alerta si se detecta presencia fuera de los horarios establecidos.

Gestión de confort

- Monitoreo de la temperatura del aula.
- Control de la temperatura de la unidad de aire acondicionado.
- Control de apertura de las persianas.
- Control del proyector de video (Encendido/Apagado).
- Control de la pantalla de proyección (Desplegada/Guardada).
- Encender unidad de aire acondicionado diez minutos antes de algún evento programado.
- Escenarios predefinidos para las distintas actividades que se pueden realizar en la SCA.

PROGRAMACIÓN DEL SISTEMA INMÓTICO

Para el desarrollo del sistema se utilizó como IDE el Eclipse Smarthome, esto debido al soporte que cuenta con la recomendación de OpenHAB, una vez en el entorno de desarrollo y considerando los parámetros seleccionados para esta propuesta, se crearon los artículos correspondientes a cada uno de ellos en el archivo *Items*, siendo en esta sección donde se configura el tipo, el nombre interno con el cual se hace

referencia dentro del software, la etiqueta y el ícono a mostrar en la interfaz. De igual manera en el archivo *Items* es donde se unieron los diferentes canales con los artículos.

Una vez creados los artículos necesarios, se procedió a dar forma a la interfaz *Basic UI* de OpenHAB en el archivo *Sitemap*, es aquí donde se desarrollaron, agruparon y organizaron los distintos artículos con el fin de generar una interfaz intuitiva para el usuario tal como se observa en la Figura 7. Luego de obtener una interfaz amigable y basado en la estrategia de control previamente establecida, se realizaron en el archivo *Rules* todas las reglas que permiten que el sistema de supervisión inmótico propuesto tenga el comportamiento definido.



Figura 7: Interfaz de usuario obtenida

En la sección *Service* de Eclipse Smarthome se encuentran los diferentes enlaces descargados, los cuales permiten a los artículos creados enviar y recibir información, cambios de estados y comandos de acción de los periféricos correspondientes a los componentes seleccionados de esta propuesta. Para el sistema propuesto se realizó la descarga y configuración del complemento *Mail* que permite enviar alertas vía correo electrónico, *Http* que ofrece la detección de dispositivos

conectados a la red local, *Epson Projector* el cual dota al sistema con la capacidad de comunicarse con los diferentes proyectores de video de la marca Epson.

De igual manera se descargó y configuró el complemento *Z-Wave* que da la posibilidad de establecer un canal de comunicación efectiva entre el nodo de control y los diferentes componentes seleccionados que conforman la propuesta. Por último, se incluyó el complemento *OpenHAB Connection* el cual ofrece la posibilidad de conectar el sistema de supervisión propuesto con IoT. Estos archivos de configuración descritos se encuentran contenidos en la carpeta “conf” en el CD anexo.

CAPÍTULO IV

VALIDACIÓN DE LA PROPUESTA

Los resultados obtenidos con la realización de esta trabajo especial de grado se presentan a continuación de acuerdo a las pruebas realizadas para validar dos aspectos fundamentales del sistema inmótico diseñado, tales como la *Estrategia de Control* establecido en el Capítulo III y la *Comunicación y Manejo de Entradas y Salidas*, características fundamentales que todo sistema inmótico debe cumplir, ya que estos deben tener la capacidad de monitorear y controlar entradas y salidas, discretas y analógicas, además de realizar la comunicación con diferentes sensores y actuadores que permiten el correcto funcionamiento del mismo.

VALIDACIÓN DE LA ESTRATEGIA DE CONTROL

Se realizó mediante la emulación de un comportamiento aleatorio de los elementos que conforman el sistema. Para esto se modificó el código original con una serie de algoritmos que emulan los cambios generados por los componentes seleccionados anteriormente. A continuación se muestran de manera detallada las diferentes pruebas realizadas, así como los resultados de las mismas para corroborar el cumplimiento de la estrategia de control.

Apagado automático según horario establecido

El segmento de código que se muestra a continuación es el encargado de apagar todas las luces y aire acondicionado de la SCA a las 9 pm:

```

// Gestión de Energía

rule "Establecer Hora de Apagado (9pm)"
when
Time cron "0 0 21 ? * * *"
then
sendCommand (Lights, OFF)
sendCommand (Climate, OFF)
end

```

Es debido a este que se puede observar en la Figura 8 como en la columna de la izquierda (*When*) a las 9:00 pm, la unidad de aire acondicionado así como una de las luces que se encontraban encendidas a esa hora se apagaron automáticamente mostrando su nuevo estado en la columna de la derecha (*What*).

	When	Where	What
All	Yesterday at 9:20 PM	Light_SF_SCA_Traseras	ON
Today	Yesterday at 9:15 PM	Temperature_SF_SCA	21.4
Last week	Yesterday at 9:00 PM	Climate_SF_SCA	OFF
	Yesterday at 9:00 PM	Light_SF_SCA_Traseras	OFF
	Yesterday at 9:00 PM	Light_SF_SCA_Delanteras	OFF
	Yesterday at 8:22 PM	Temperature_SF_SCA_Setpoint	19
	Yesterday at 8:22 PM	Temperature_SF_SCA_Setpoint	20
	Yesterday at 8:22 PM	Temperature_SF_SCA_Setpoint	21

Figura 8: Lista de eventos ocurridos a las 9pm

Alerta de cuando se enciende el aire acondicionado y se encuentra la puerta o una ventana abierta

Para realizar esta prueba se desarrolló una regla, que se muestra en el código a continuación, la cual al momento de abrir una ventana o la puerta de la SCA y esta permanezca en el estado previo durante dos minutos con el aire acondicionado encendido y se envíe un correo electrónico al administrador con el fin de notificar lo ocurrido. En la Figura 9 se observa una muestra del mensaje enviado.

```
// Gestión de Energía

rule "Alerta de ACC=ON y Puerta abierta"
when
Item Door_SF_SCA received update open or
Item Windows_SF_SCA received update open
then
if (Climate_SF_SCA==ON) {
timer3 = createTimer(now.plusSeconds(120)) [
sendMail("El ACC se encuentra encendido, por favor mantenga la puerta
y ventanas cerradas. Ahorremos Energía!")]
}
else {
if(timer3!=null) {
timer3.cancel
timer3 = null}}
end
```

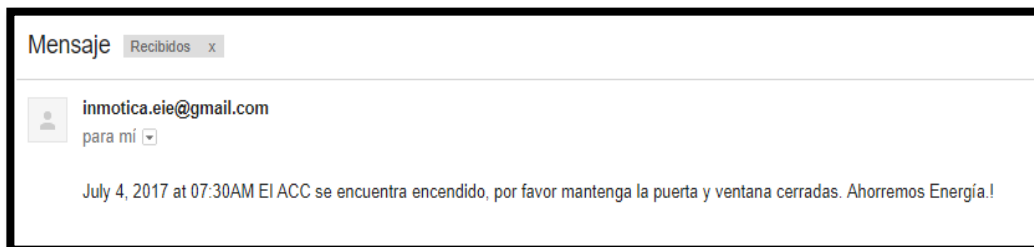


Figura 9: Alerta de aire acondicionado encendido con puerta o ventana abierta.

Alerta de incendios

Al momento de que la alarma de incendios se active y según lo estipulado en la estrategia de control, se notifica al administrador a través del complemento Open Cloud Connection de OpenHAB que se ha detectado un incendio y se debe llamar a los bomberos como se muestra en la Figura 10, además se encienden las luces para facilitar la evacuación del aula. Esto se logró mediante al segmento de código que se muestra a continuación:

```
//Gestión de Seguridad

rule "Encender Luces si se activa la alarma de incendios"
when
Item Fire_SF_SCA changed from OPEN to CLOSED
then
sendCommand (Lights, ON)
sendLogNotification("ALARMA, Se ha detectado INCENDIO, LLAME A LOS BOMBEROS!!!")
end
```



Figura 10: Alerta de incendios

Alerta de presencia fuera del horario establecido

En un aula como la SCA después de las 9 pm no debe haber movimiento dentro de la misma, debido a que la jornada académica ya ha finalizado. Es por esto que incluyó en la estrategia de control una alerta cuando se detecta movimiento después de esta hora, esto se logró con la creación de la regla que se muestra en el siguiente segmento de código:

```
//Gestión de Seguridad

rule "Alerta de movimiento después de (9pm) "
when
Time cron " 0 * 21-5 ? * MON,TUE,WED,THU,FRI *"
then
if (Move_SF_SCA==ON) {
sendMail("ALARMA, Se ha detectado movimiento fuera
del horario previsto!!!")
}
end
```

Debido a que esta alerta se presenta a una hora donde el administrador no se encuentra en horario laboral, se configuró para que la misma fuera recibida por el administrador o personal de seguridad vía correo electrónico como se observa en la Figura 11.

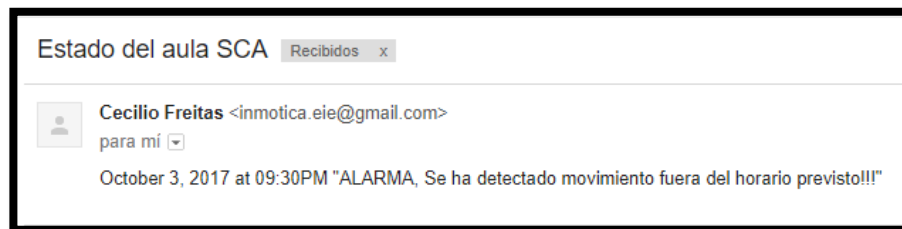


Figura 11: Alarma de presencia después de las 9pm

Escenarios preestablecidos

Con la creación de la regla que se puede observar en el siguiente segmento de código, se establecieron 4 escenarios (*OFF*, *Clases*, *Ahorro* y *Expo*) respondiendo de manera afirmativa la interrogante ¿Se puede mejorar la experiencia del docente al dar clases y la de los estudiantes al incorporar inmótica en un salón de clases?, planteada en el Capítulo I.

```
                                //Gestión de Confort
rule "Actuación de los escenarios SCA"
when Item Scene_SF_SCA received command
then
    if (Scene_SF_SCA==0) { //OFF
        sendCommand (Beam_SF_SCA, OFF)
        sendCommand (Cinema_SF_SCA, UP)
        sendCommand (Light_SF_SCA_Delanteras, OFF)
        sendCommand (Light_SF_SCA_Traseras, OFF)
        sendCommand (Television_SF_SCA_1, OFF)
        sendCommand (Television_SF_SCA_2, OFF)
        sendCommand (Climate_SF_SCA, OFF) }
    if (Scene_SF_SCA==1) { //CLASES
        sendCommand (Beam_SF_SCA, OFF)
        sendCommand (Cinema_SF_SCA, UP)
        sendCommand (Light_SF_SCA_Delanteras, ON)
        sendCommand (Light_SF_SCA_Traseras, ON)
        sendCommand (Climate_SF_SCA, ON) }
    if (Scene_SF_SCA==2) { //AHORRO
        sendCommand (Beam_SF_SCA, OFF)
        sendCommand (Cinema_SF_SCA, UP)
        sendCommand (Light_SF_SCA_Delanteras, ON)
        sendCommand (Light_SF_SCA_Traseras, OFF)
        sendCommand (Temperature_SF_SCA_Setpoint, 23)
        sendCommand (Shutter_SF_SCA, 0) }
    if (Scene_SF_SCA==3) { // EXPO
        sendCommand (Beam_SF_SCA, ON)
        sendCommand (Shutter_SF_SCA, 100)
        sendCommand (Cinema_SF_SCA, DOWN)
        sendCommand (Light_SF_SCA_Delanteras, OFF)
        sendCommand (Light_SF_SCA_Traseras, OFF) }
    end
```

En la Tabla 4 se muestra el estado de los diferentes parámetros de la SCA para cada escenario creado. De igual manera en la Figura 12, Figura 13 y Figura 14 se observa la interfaz de usuario con los diferentes estados de los parámetros para los escenarios *OFF*, *AHORRO* y *CLASES* respectivamente.

Tabla 4: Parámetros de los escenarios del SCA

ELEMENTO	OFF	CLASES	AHORRO	EXPO
Luces Delanteras	Off	On	On	Off
Luces Traseras	Off	On	Off	Off
Proyector	Off	Off	Off	On
Pantalla de Proyección	Arriba	Arriba	Arriba	Abajo
ACC	Off	On	On	On
Temperatura del ACC	-	-	23 °C	-
Persianas	-	-	Abierto	Cerrado

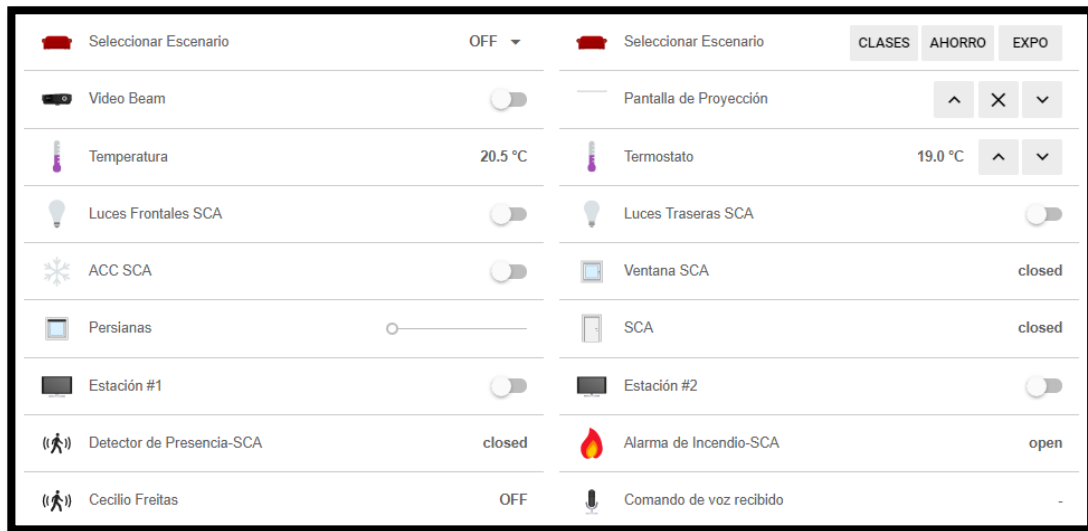


Figura 12: Estado inicial en OFF



Figura 13: Cambio al escenario AHORRO



Figura 14: Cambio de escenario a CLASES

En la Figura 15 se observa la lista de eventos, aquí se puede detallar los estados en los diferentes parámetros que conforman el sistema inmótico y que varían al seleccionar un nuevo escenario, como por ejemplo la temperatura del termostato que

varía hasta llegar a los 19 °C y al momento de cambiar al escenario Ahorro, este se estableció automáticamente a 23 °C.

	When	Where	What
Clases	Yesterday at 10:12 PM	Light_SF_SCA_Traseras	ON
	Yesterday at 10:12 PM	Climate_SF_SCA	ON
	Yesterday at 10:12 PM	Scene_SF_SCA	1
Ahorro	Yesterday at 10:06 PM	Light_SF_SCA_Delanteras	ON
	Yesterday at 10:06 PM	Temperature_SF_SCA_Setpoint	23
	Yesterday at 10:06 PM	Scene_SF_SCA	2
	Yesterday at 10:05 PM	Temperature_SF_SCA_Setpoint	19
	Yesterday at 10:05 PM	Temperature_SF_SCA_Setpoint	20
Off	Yesterday at 10:04 PM	Light_SF_SCA_Delanteras	OFF
	Yesterday at 10:04 PM	Scene_SF_SCA	0

Figura 15: Registro de eventos entre los cambios de escenarios

Automatización selectiva

El funcionamiento de esta característica de la estrategia de control se basó en la creación de 65 reglas para cubrir 13 bloques de una hora por cada día de la semana, excluyendo los sábados y domingos. Cada una de estas reglas son similares al siguiente segmento de código:

```

//MIERCOLES
rule "Encender ACC 10min antes de un evento programado SCA MIERCOLES-8am"
when Time cron "0 50 7 ? * WED *"
then if (SCA_MIERCOLES_8==ON) {
if (Climate_SF_SCA==OFF) {sendCommand (Climate_SF_SCA, ON)} } end

```

Para verificar el funcionamiento de la automatización selectiva y como se puede observar en la Figura 16 se activó el bloque correspondiente al miércoles de 8 a 9 am, una vez activado se comprobó nuevamente en la Figura 17 el apagado automático establecido a las 9:00 pm, para el día en cuestión a las 7:50 am se encendió el aire acondicionado del aula SCA tal como se puede detallar en la Figura 18 el nuevo estado.

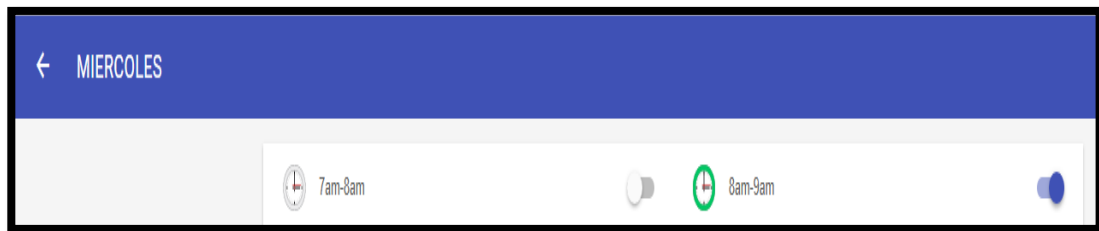
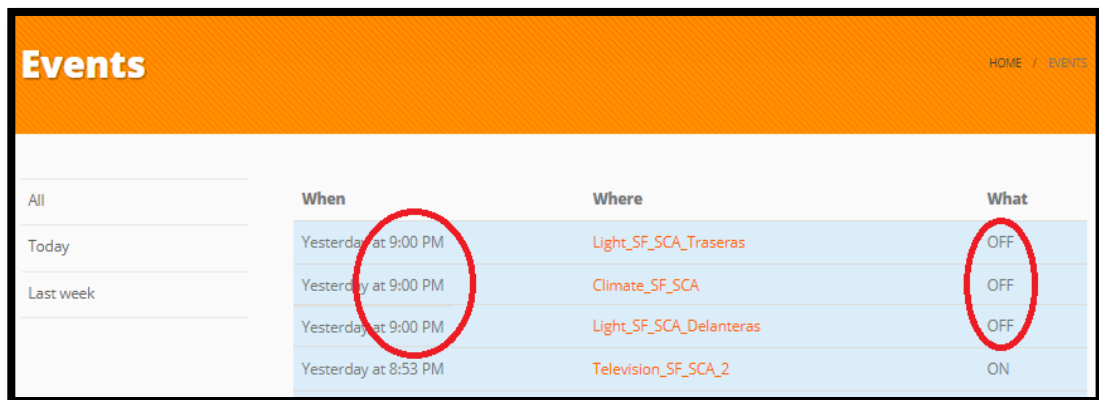


Figura 16: Automatización selectiva (Miércoles 8:00am)



	When	Where	What
All	Yesterday at 9:00 PM	Light_SF_SCA_Traseras	OFF
Today	Yesterday at 9:00 PM	Climate_SF_SCA	OFF
Last week	Yesterday at 9:00 PM	Light_SF_SCA_Delanteras	OFF
	Yesterday at 8:53 PM	Television_SF_SCA_2	ON

Figura 17: Apagado automático (Martes 9:00pm)

	When	Where	What
All			
Today	Today at 7:50 AM	Climate_SF_SCA	ON
Last week	Yesterday at 9:00 PM	Climate_SF_SCA	OFF

Figura 18: Registro del evento automatizado selectivamente

VALIDACIÓN DE LA COMUNICACIÓN Y MANEJO DE ENTRADAS Y SALIDAS

Luego de verificar la estrategia de control establecida para el sistema inmótico diseñado se procedió a realizar una serie de pruebas de comunicación, las cuales fueron agrupadas de acuerdo al medio seleccionado para las mismas. Además de la comunicación, en estas pruebas se comprobó la capacidad que tiene el sistema diseñado de manejar entradas y salidas discretas y analógicas.

Comunicación Modbus

Mediante una prueba a pequeña escala y trabajando de acuerdo a los recursos disponibles con un controlador lógico programable, usando dos entradas del mismo que emulan dos entradas al sistema una digital que puede representar un sensor de movimiento o alarma de incendio y una analógica que representa el sensor de temperatura. Adicional a estas señales hay una segunda señal digital que representa cualquiera de las señales de salida digitales de un sistema inmótico. Con estas tres señales se abarcan todos los tipos de señales que se encuentran presentes en el sistema diseñado y que este debe recibir, interpretar y enviar.

Para la realización del montaje que se puede observar en la Figura 19, la Figura 20 y la Figura 21 se necesitó de los siguientes elementos adicionales:

- Controlador programable Micro 820. Allen-Bradley
- Potenciómetro 10 k Ω
- Interruptor
- Base de aluminio
- Placa de aluminio

Mientras tanto para la configuración para establecer la comunicación Modbus del controlador programable Micro 820 realizada se hizo uso del software Connected Components Workbench.

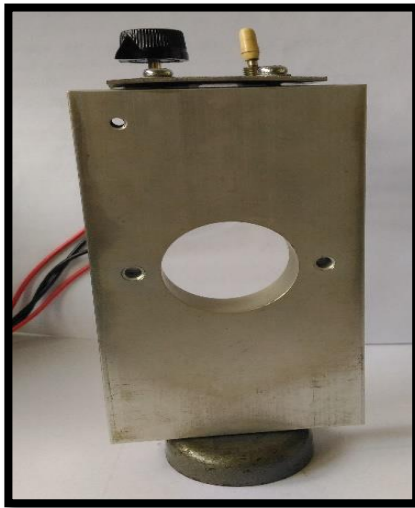


Figura 19: Montaje de prueba vista frontal

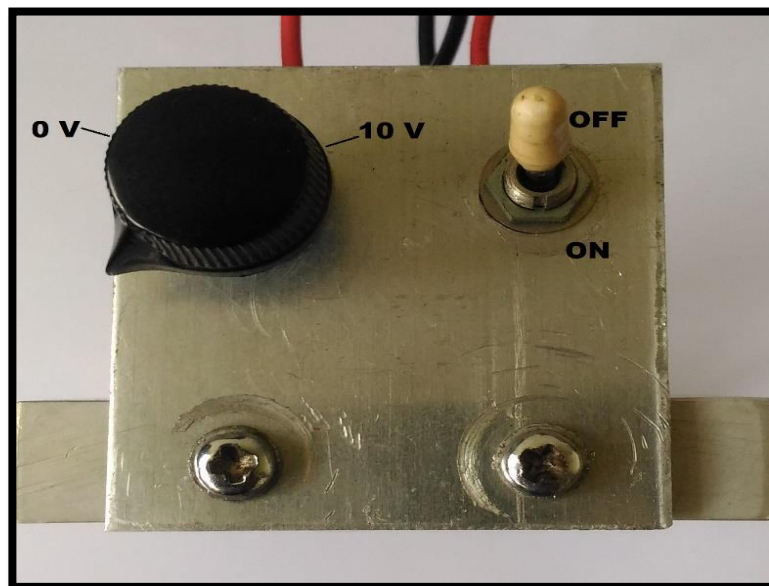


Figura 20: Montaje de prueba visto desde arriba

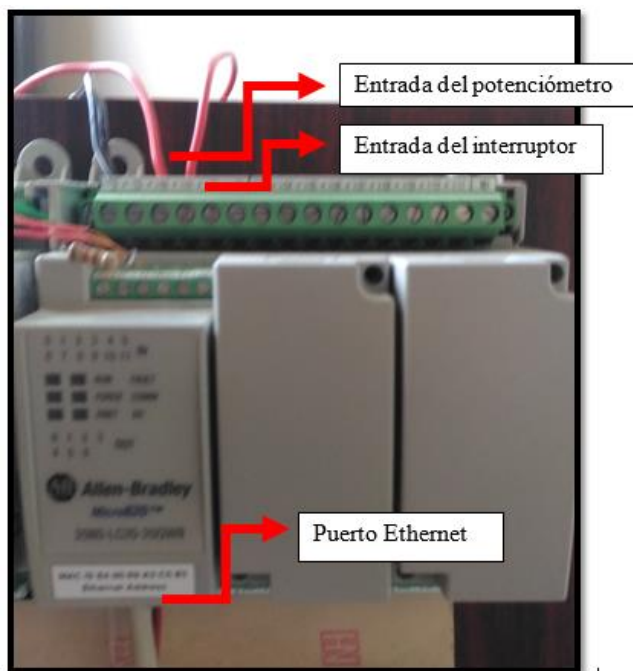


Figura 21: Micro820 y conexiones

En la Figura 22 se puede observar claramente el diagrama de escalera realizado para cumplir con la prueba a escala del sistema; en la primera línea del diagrama se detalla como la entrada digital 1 actúa directamente en la salida digital 1. En la segunda línea del diagrama la entrada analógica 1 es convertida a una variable de tipo Real, esto con el fin de poder ser escalada con la ayuda del bloque *SCALER* el cual le da un rango entre 10 y 40, emulando así un sensor de temperatura. En la línea siguiente esta variable Real escalada es transformada en una tipo Word. Por último se creó una variable global llamada “Encender” la cual actúa directamente en la salida digital 2. Mientras que en la Figura 23 muestra como las variables de interés son asignadas a espacios de memoria o registros dentro del controlador lógico programable con el fin de poder tener acceso a ellas a través del maestro Modbus.

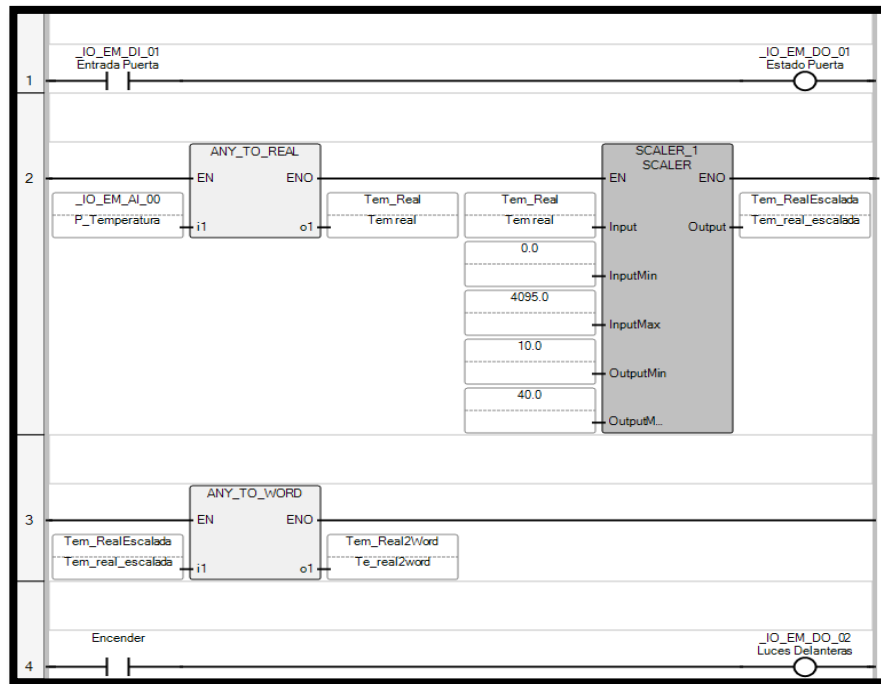


Figura 22: Diagrama de escalera Micro820

Nombre de la variable	Tipo de datos	Dirección	Direcciones usadas
_IO_EM_DI_01	BOOL	<input type="text" value="000001"/>	000001 X
_IO_EM_DO_02	BOOL	000002	000002
Encender	BOOL	000003	000003
Tem_RealEscalada	REAL	400001	400001 - 400002
Tem_Real2Word	WORD	400003	400003

Figura 23: Asignación de variables Modbus del Micro820

Posterior a la configuración del PLC, se procedió a configurar el complemento Modbus TCP/Serial de OpenHAB, esta configuración se realizó de acuerdo a la documentación del mismo, dando como resultado lo mostrado en la Figura 24. Luego de configurar el complemento se procedió a enlazar los elementos del sistema con los esclavos configurados anteriormente, este enlace es lo que se observa en el segmento de código mostrado a continuación:

```
//MODBUS

Number Temperature SF_SCA "Temperatura [%1f °C] "
<temperature> (Temperature, SF_SCA, Histogramal)
{modbus="<[slave1:0]" }

Switch Television SF_SCA_1 "Estación #1" <television>
(SF_SCA, Pcs) {modbus=">[slave3:0]" }

String Door_SF_SCA "Puerta SCA [%d]" <door> (SF_SCA)
{modbus="<[slave2:0]" }
```



```
modbus.cfg
#####
#Configuracion

tcp.slave1.connection=192.168.0.3:502:60:0:0:3:100
tcp.slave1.type=holding
tcp.slave1.start=0
tcp.slave1.length=2
tcp.slave1.valuetype=float32

tcp.slave2.connection=192.168.0.3:502:60:0:0:3:100
tcp.slave2.type=coil
tcp.slave2.start=0
tcp.slave2.valuetype=bit

tcp.slave3.connection=192.168.0.3:502:60:0:0:3:100
tcp.slave3.type=coil
tcp.slave3.start=2
tcp.slave3.valuetype=bit
```

Figura 24: Configuración del complemento Modbus en OpenHAB

Luego de realizar las pruebas y verificar las configuraciones en el controlador programable y en OpenHAB no se pudo establecer una comunicación efectiva entre ambos, al verificar la bitácora para determinar la falla y tal como se puede observar en la Figura 25 como el complemento Modbus de OpenHAB considera el puerto de comunicaciones, configurado previamente como 502, como si fuera una taza de baudios, originando así un error de argumento invalido en la configuración de dicho complemento.



```
openhab: Bloc de notas
Archivo Edición Formato Ver Ayuda
java.lang.IllegalArgumentException invalid baud rate: 502
2017-10-02 20:13:40.622 [ERROR] [org.apache.felix.configadmin ] -
[org.osgi.service.cm.ManagedService, org.osgi.service.event.EventHandler, id=357,
bundle=248/mvn:org.openhab.binding/org.openhab.binding.modbus/1.9.0]: Updating property
tcp.slave1.connection of configuration org.openhab.modbus caused a problem: Exception when parsing
configuration parameter tcp.slave1.connection = 192.168.0.3:502:60:0:0:3:100 --
java.lang.IllegalArgumentException invalid baud rate: 502
org.osgi.service.cm.ConfigurationException: tcp.slave1.connection : Exception when parsing configuration
parameter tcp.slave1.connection = 192.168.0.3:502:60:0:0:3:100 -- java.lang.IllegalArgumentException
invalid baud rate: 502
```

Figura 25: Error en el complemento de OpenHAB

Adicional a esta prueba y para corroborar que el PLC se configuró como esclavo de manera correcta, se realizó una prueba de comunicación Modbus con la ayuda de TOP Server 6, software que actuó como maestro durante la siguiente prueba, con el fin de obtener acceso a las variables Modbus asignadas previamente en el Micro820, la configuración en Top Server 6 se puede observar en la Figura 26.

El desempeño que se muestra en la Figura 27 y la Figura 28 entre el Micro820 y TOP Server 6 fue satisfactorio ya que a través de Top Server 6 se obtuvo acceso a las diferentes variables Modbus configuradas previamente en el Micro 820, lo cual certifica una configuración correcta del Micro820.

The screenshot shows the 'TOP Server 6 Configuration' window with a tree view on the left and a table of tag configurations on the right. The tree view includes 'Project', 'Connectivity', 'Test', 'Device1', 'Aliases', 'Advanced Tags', and 'Alarms & Events'. The table lists the following tags:

Tag Name	Address	Data Type	Scan Rate	Scaling	Description
Encender	000003	Boolean	100	None	
Entrada digital 1	000001	Boolean	100	None	
Real2Word	400003	Word	100	None	
RealEscalada	400001	Float	100	None	
Salida Digital 2	000002	Boolean	100	None	

Figura 26: Configuración inicial de variables Modbus en TOP Server 6

The screenshot shows a table of real-time values for the configured Modbus tags. The table includes columns for Item ID, Data Type, Value, Timestamp, Quality, and Update Count.

Item ID	Data Type	Value	Timestamp	Quality	Update Count
Test.Device1.Encender	Boolean	0	15:11:27.170	Good	1
Test.Device1.Entrada digital 1	Boolean	1	15:11:27.170	Good	1
Test.Device1.Real2Word	Word	33	15:11:27.201	Good	1
Test.Device1.RealEscalada	Float	33.5531	15:18:46.265	Good	394
Test.Device1.Salida Digital 2	Boolean	0	15:11:27.170	Good	1

Figura 27: Valores de las variables del Micro820 vistas desde TOP Server 6

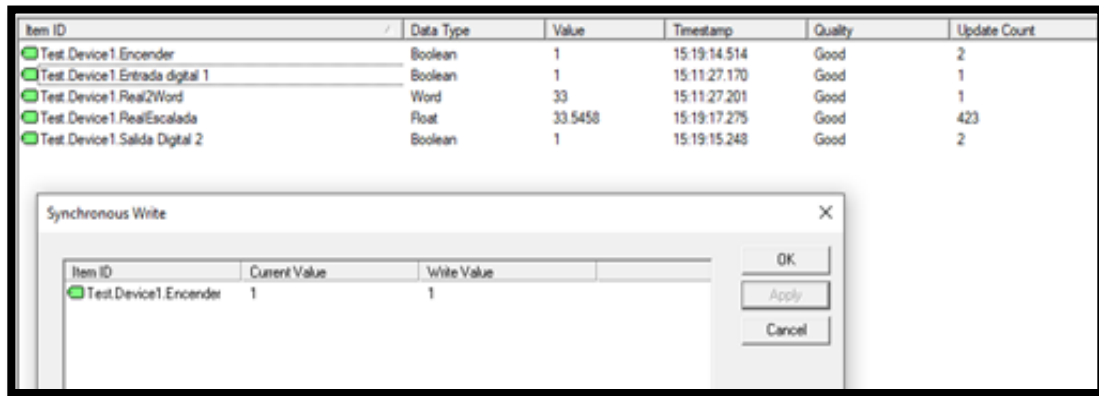


Figura 28: Prueba de escritura en los registros del Micro820 desde TOP Server 6

Para certificar el funcionamiento de OpenHAB como maestro durante el proceso de comunicación Modbus, se utilizó el software Modbus Poll en su versión de esclavo con las configuraciones mostradas en la Figura 29 donde se inician las variables tipo coil y holding, la Figura 30 donde se configura las características de la comunicación Modbus y en la Figura 31 se configura el complemento Modbus en OpenHAB de acuerdo a lo configurado en Modbus Poll.

Dando como resultado lo mostrado en la Figura 32 donde se observa como la "Estación #1" se encuentra encendida y el registro coil asociado en la configuración de OpenHAB tiene un valor de 0, además al momento de cambiar el valor en el coil de Modbus Poll enlazado a la "Puerta SCA" de OpenHAB no ocurrió cambio alguno en su estado inicial de abierto.

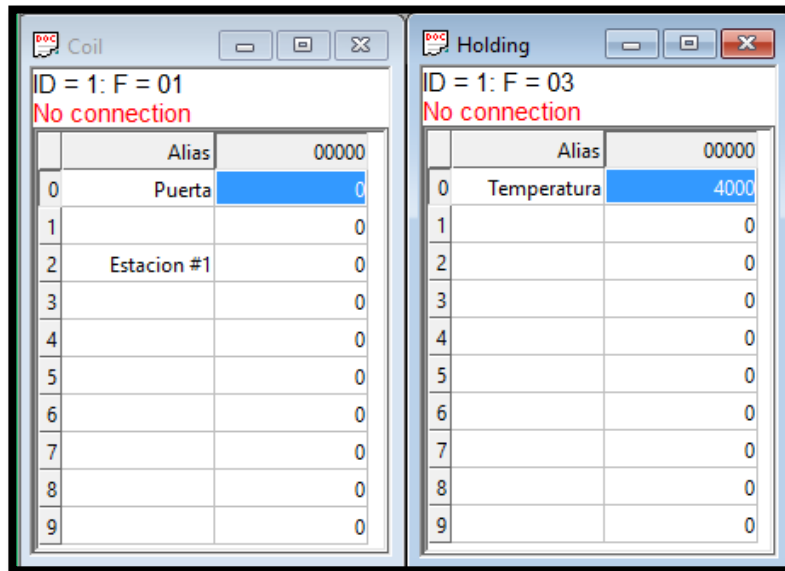


Figura 29: Variables tipo coil y holding de Modbus Poll

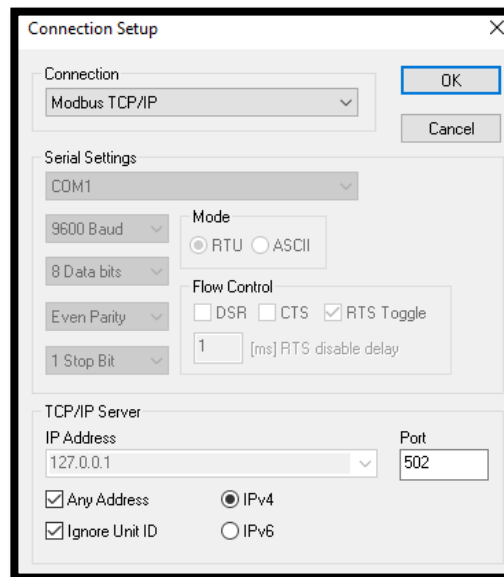


Figura 30: Configuración de los parámetros de comunicación de Modbus Poll

```

demo.items //MODBUS

Number Temperature_SF_SCA "Temperatura [%1.f °C]" <temperature> (Temperature, SF_SCA, Histogramal) {modbus="<[slave1:0]"}
Switch Television_SF_SCA_1 "Estación #1" <television> (SF_SCA, Pcs) {modbus=">[slave3:0]"}
Contact Door_SF_SCA "SCA [MAP(en.map):%s]" <door> (SF_SCA, Doors) {modbus="<[slave2:0]"}

modbus.cfg
#####
#Configuracion

tcp.slave1.connection=127.0.0.1:502:60:0:0:3:100
tcp.slave1.type=holding
tcp.slave1.start=0
tcp.slave1.length=2
tcp.slave1.valuetype=float32

tcp.slave2.connection=127.0.0.1:502:60:0:0:3:100
tcp.slave2.type=coil
tcp.slave2.start=0
tcp.slave2.valuetype=bit

tcp.slave3.connection=127.0.0.1:502:60:0:0:3:100
tcp.slave3.type=coil
tcp.slave3.start=2
tcp.slave3.valuetype=bit

```

Figura 31: Configuración de los elementos y complemento Modbus de OpenHAB

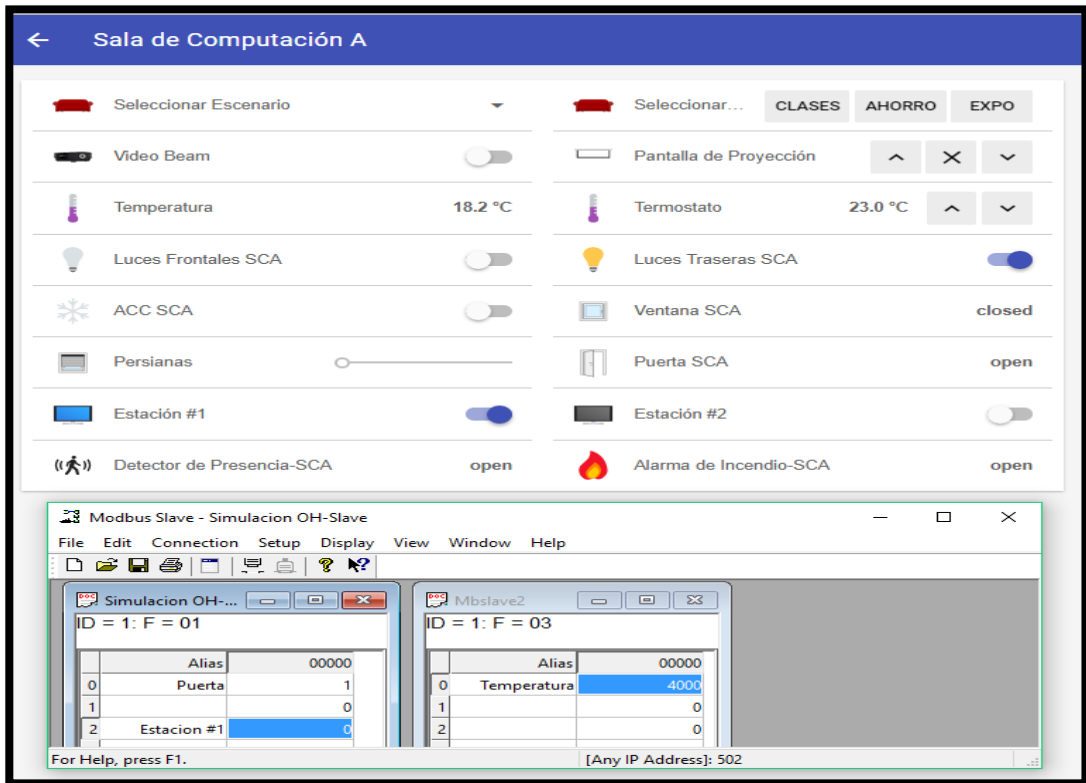


Figura 32: Comportamiento entre OpenHAB y Modbus Poll

Luego de tres pruebas y con base a la información recolectada en cada una de ellas, la falla detectada y que se mostró en la Figura 25, confirma que dicho error en la comunicación recae en el complemento modbus de OpenHAB, lo que impide la validación del sistema con este protocolo de comunicación.

Comunicación por puerto Serial

Como segunda opción para validar la capacidad de conexión del sistema diseñado en OpenHAB, se procedió a realizar una prueba de comunicación via puerto serial con una de placa de desarrollo como lo es el Arduino Uno, cabe destacar que la comunicación serial de Arduino es en código ASCII, característica fundamental ya que esta es la manera en la cual según la documentación oficial de OpenHAB está configurado el complemento de comunicación serial del mismo.

Para esto se desarrolló una rutina en el Arduino Uno además de la respectiva configuración, con el fin de corroborar la comunicación, para esto se establecieron dos pruebas, en la primera el Arduino Uno tomó el comportamiento de un actuador, controlando el encendido y apagado de un LED que emula el estado de la “*Estación #1*” controlada a través de la interfaz. En la segunda prueba el Arduino Uno muestra en que rango se encuentra una variable analógica como la apertura de las persianas de la SCA, estos rangos se establecieron entre 0-25%, 26-50%, 51-75% y 76-100%, donde según sea el caso se enciende un LED en particular, con la excepción del primer rango entre 0-25% donde todos los estarán apagados. La rutina del Arduino Uno que permite la realización de estas pruebas se presenta a continuación:

```
void loop() {  
char dato = Serial.read();  
if (dato ==100 ){  
digitalWrite(13, !digitalRead(13));  
digitalWrite(7, !digitalRead(7));}  
  
if (dato ==101 ){  
digitalWrite(3, LOW);  
digitalWrite(4, LOW);  
digitalWrite(5, LOW);}  
  
if (dato ==102 ){  
digitalWrite(3, HIGH);  
digitalWrite(4, LOW);  
digitalWrite(5, LOW);}  
  
if (dato ==103 ){  
digitalWrite(3, LOW);  
digitalWrite(4, HIGH);  
digitalWrite(5, LOW);}  
  
if (dato ==104 ){  
digitalWrite(3, LOW);  
digitalWrite(4, LOW);  
digitalWrite(5, HIGH);}  
}
```

Tal como se observa en la Figura 33 y la Figura 34 cuando el usuario enciende o apaga la “Estación #1” el LED que posee el Arduino Uno en el pin 13 ejecuta la acción deseada, de igual manera en la Figura 35 se observa cómo según el porcentaje de cierre de las persianas del SCA se enciende un LED para cada caso.



Figura 33: Estación #1 encendida y respuesta en el Arduino Uno



Figura 34: Estación #1 apagada y respuesta en el Arduino Uno

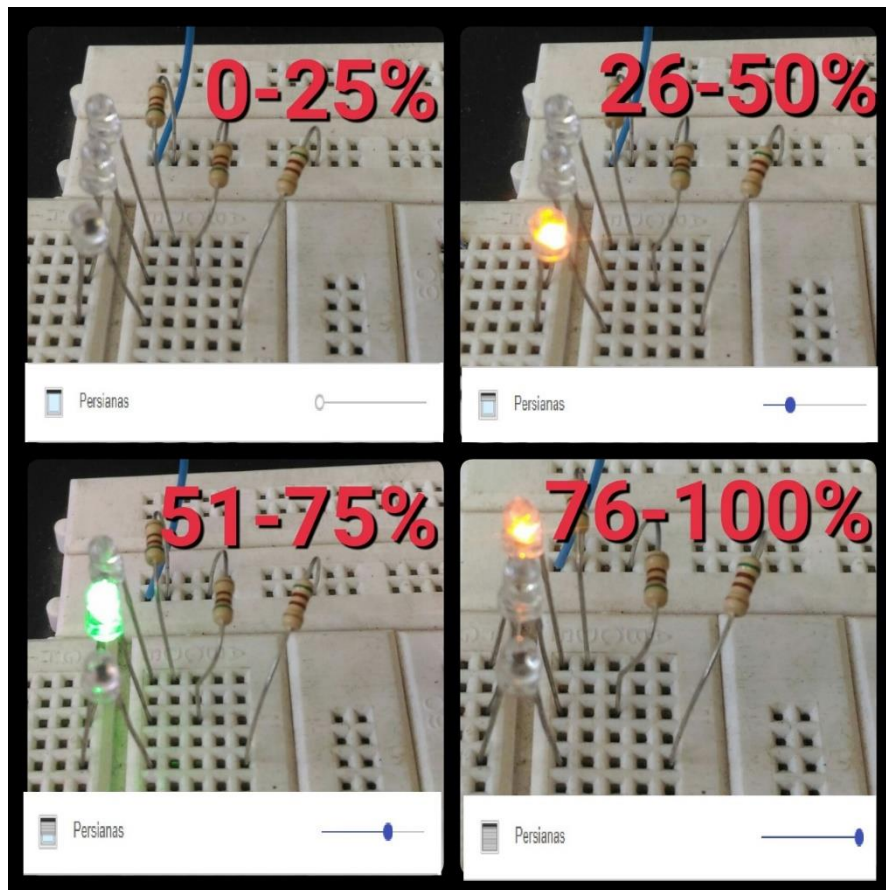


Figura 35: Encendido de LED según porcentaje de cierre de las persianas

Con estas dos pruebas se validó la capacidad que tiene el sistema inmótico diseñado de manejar salidas discretas (*Estación #1*) y analógicas (*Persianas*), además de establecer una comunicación efectiva haciendo uso del complemento Serial, esto quiere decir, que OpenHAB por ser una herramienta de código abierto tiene complementos desactualizados o no compatibles con las versiones de OpenHAB para cada sistema operativo, como resultó ser el caso de Modbus, pero de igual manera, hay complementos como el Serial que se encuentran actualizados y presentan un desempeño estable para la versión 2.0 de OpenHAB para Windows de 64 bits.

Comunicación vía Ethernet

Con el fin de demostrar la capacidad de OpenHAB de comunicarse con distintos dispositivos y de distintas maneras, se realizaron tres pruebas adicionales, la primera consistió en configurar un teléfono celular como detector de presencia de una persona en particular, el cual indicará un estado “ON” cuando el mismo se encuentre conectado a la misma red local que el equipo donde se ejecuta el servidor local; caso contrario el elemento tendrá un estado “OFF”.

La segunda prueba válida el manejo de una entrada analógica como lo son los comando de voz, lo cuales son captados mediante la aplicación Android y mostrados en la interfaz de usuario. Esta prueba también permitió verificar una conexión a través de una URL remota cuando un usuario tal como se especifica en el Anexo B, previamente creado y autorizado por el administrador del sistema inmótico propuesto, se encuentra conectado mediante el uso de la aplicación Android de OpenHAB.

Como se puede observar en la Figura 36 con la ayuda de la aplicación Android se capturó el comando “off”, posterior a esto el comando captado sustituyó al último que había sido recibido tal como se ve en la parte inferior de la Figura 36. De igual manera también se puede notar como el estado del detector de presencia de “Cecilio Freitas” cambio su estado de “OFF” a “ON” lo que indica que el comando de voz fue enviado desde una URL remota y no a través de una conexión local, validando de esta manera una entrada discreta (*Detector de presencia*) y una entrada analógica (*Comandos de voz*).

Posterior a estas y con el fin de certificar una conexión efectiva entre la interfaz *Basic UI* y la aplicación Android, en la cual se captó el comando de voz “perro” a través de la aplicación Android acción que se evidencia en la parte superior de la Figura 37, este comando no está configurado para hacer alguna acción, pero de igual manera

se muestra en la interfaz del servidor local tal como se observa en la parte inferior de la Figura 37.



Figura 36: Comportamiento de la aplicación Android

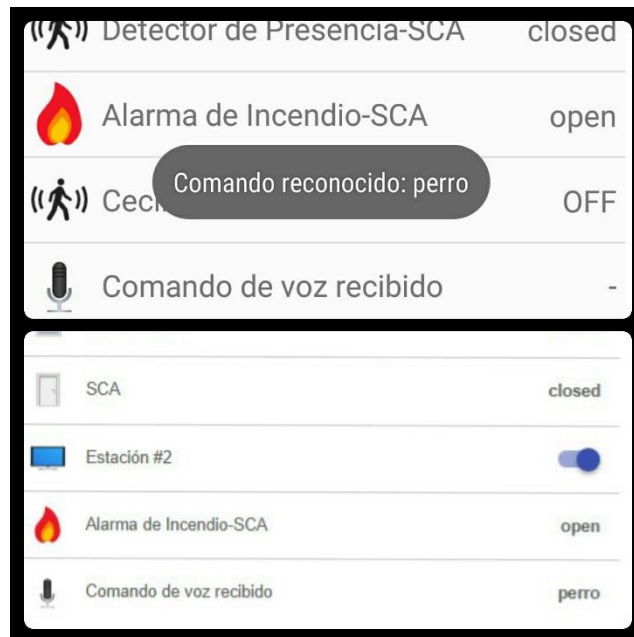


Figura 37: Conexión entre aplicación Android e interfaz Basic UI

CONCLUSIONES

De este trabajo especial de grado se obtuvieron las siguientes conclusiones:

- ✓ Los parámetros de control seleccionados permitieron la adaptación del sistema inmótico a las necesidades requeridas del ambiente escogido, en este caso la sala de computación A.
- ✓ El software seleccionado se adaptó a los parámetros de control establecidos, tiene la versatilidad de poder ser ejecutado en varias plataformas tecnológicas y presenta una gran variedad de complementos lo que permite actualizaciones futuras.
- ✓ La plataforma tecnológica donde se ejecutó el software seleccionado se ajusta perfectamente a las necesidades de la SCA.
- ✓ Los componentes propuestos para supervisar y controlar los parámetros seleccionados no requieren de modificaciones a la infraestructura de la SCA ya que estos se comunican con el servidor de manera inalámbrica.
- ✓ El sistema de supervisión inmótico diseñado se adapta a las necesidades que se espera atender con su implementación, cuenta con un diseño sencillo y estrategia de control que satisface los parámetros determinados a gestionar, además se encuentra orientado a la expansión dentro de la Escuela de Ingeniería Eléctrica de la UCV.
- ✓ Se logró validar satisfactoriamente la estrategia de control propuesto, el manejo de entradas discretas, entradas analógicas, salidas discretas y salidas analógicas; además de la capacidad que tiene el sistema inmótico diseñado de comunicarse con elementos periféricos a través de diferentes canales de comunicación.

RECOMENDACIONES

- ✓ Realizar una evaluación al resto de las áreas dentro de la Escuela de Ingeniería Eléctrica para listar los requerimientos específicos de cada una, para conocer las necesidades y problemáticas de las instalaciones.

- ✓ Para optimizar la propuesta se pueden incluir nuevas etapas, por ejemplo una de alimentación de respaldo.

- ✓ Se recomienda la implementación del sistema en un área a modo de prueba piloto, con lo que se podrán realizar las pruebas necesarias antes de implementar por completo el sistema propuesto.

- ✓ Poner en prueba la comunicación vía Modbus con una propuesta que utilice un software de código abierto y plataforma tecnológica diferente ya que en este trabajo especial de grado no se pudo llevar a cabo la misma.

REFERENCIAS BIBLIOGRÁFICAS

- [1] « Asociación Española de Domótica e Inmótica,» [En línea]. Disponible: <http://www.cedom.es/sobre-domotica/que-es-inmotica>. [Último acceso: 24 1 2017].
- [2] Guette. L., Diseño de un sistema inmótico mediante el uso del sistema embebido intel galileo [Tesis], Caracas: Universidad Central de Venezuela, 2015.
- [3] Porras. C., Estudio y diseño de un sistema inmótico para su aplicación en el Edificio de Laboratorios de la Universidad Católica Andrés Bello [Tesis], Caracas: Universidad Católica Andrés Bello, 2012.
- [4] Fritz. A. y Colon. L., Diseño e implementación de un sistema domótico de manejo remoto utilizando internet y tecnología celular [Tesis], Caracas: Universida Católica Andrés Bello, 2005.
- [5] Romero. C., Vazquez. F. y Castro. C., Domótica en Inmótica. Vivienda y Edificios Inteligentes, Madrid: RA-Ma, 2006.
- [6] «Asociación Española de Domótica e Inmótica,» [En línea]. Disponible: <http://www.cedom.es/sobre-domotica/que-es-domotica>. [Último acceso: 02 02 2017].
- [7] «Real Academia Española,» [En línea]. Disponible: <http://dle.rae.es/?id=E7W0v9b>. [Último acceso: 7 5 2017].
- [8] T. y. C. d. E. Ministerio de Industria, «Fundación para el Fomento de la Innovación Industrial,» [En línea]. Disponible: http://www.f2i2.net/documentos/lsi/rbt/guias/guia_bt_51_feb07R1.pdf. [Último acceso: 27 04 2017].
- [9] «Domótica, El futuro de las viviendas,» [En línea]. Disponible: <https://domoticaudem.wordpress.com/arquitectura-de-los-sistemas/>. [Último acceso: 22 05 2017].

- [10] «BACnet,» [En línea]. Disponible: <http://www.bacnet.org/Contact/BACnet-L.htm>. [Último acceso: 05 06 2017].
- [11] «Lonworks.es,» [En línea]. Disponible: <http://www.lonworks.es/index.php/component/content/article/35-webs-oficiales/46-una-introduccion-a-lonworks.html>. [Último acceso: 05 06 2017].
- [12] Gmo, «arqcompus-domotica,» [En línea]. Disponible: <http://arqcompus-domotica.blogspot.com/2009/06/protocolos-de-comunicacion.html> . [Último acceso: 20 10 2017].
- [13] «KNX Association,» [En línea]. Disponible: <https://www.knx.org/knx-es/knx/associacion/que-es-knx/>. [Último acceso: 05 06 2017].
- [14] «Zigbee Alliance,» [En línea]. Disponible: <http://www.zigbee.org/zigbeealliance/>. [Último acceso: 05 06 2017].
- [15] I. Sigma Designs. [En línea]. Disponible: <http://z-wave.sigmadesigns.com/design-z-wave/z-wave-public-specification>. [Último acceso: 05 06 2017].
- [16] «KNX Association,» [En línea]. Disponible: <https://www.knx.org/mx/knx/tecnologia/estandarizacion/index.php>. [Último acceso: 18 05 2017].
- [17] «Wikipedia,» [En línea]. Disponible: <https://es.wikipedia.org/wiki/ZigBee>. [Último acceso: 18 05 2017].
- [18] «Home Assistant,» [En línea]. Disponible: <https://home-assistant.io/>. [Último acceso: 07 06 2017].
- [19] «Domoticz,» [En línea]. Disponible: <https://domoticz.com/>. [Último acceso: 07 06 2017].
- [20] «PiDome,» [En línea]. Disponible: <https://pidome.org/>. [Último acceso: 07 06 2017].
- [21] «OpenHAB Community and OpenHAB Foundation,» [En línea]. Disponible: <http://www.openhab.org/introduction.html>. [Último acceso: 07 06 2017].

- [22] «Oracle,» [En línea]. Disponible: <https://www.java.com/es/about/>. [Último acceso: 29 05 2017].
- [23] «Eclipse SmartHome,» [En línea]. Disponible: <http://www.eclipse.org/smarthome/index.html>. [Último acceso: 29 05 2017].
- [24] «Universidad Central de Venezuela,» [En línea]. Disponible: <http://www.ucv.ve/index.php?id=7771>. [Último acceso: 17 06 2017].
- [25] «Sigma Designs,» [En línea]. Disponible: <http://www.z-wave.com/shop-z-wave-smart-home-brands>. [Último acceso: 28 06 2017].
- [26] «Epson America,» [En línea]. Disponible: <https://epson.com/For-Work/Projectors/Classroom/PowerLite-S27-SVGA-3LCD-Projector/p/V11H694020>. [Último acceso: 28 06 2017].
- [27] Amazon, «Amazon,» [En línea]. Disponible: <https://www.amazon.com/Amazon-Echo-Bluetooth-Speaker-with-WiFi-Alexa/dp/B00X4WHP5E> . [Último acceso: 21 10 2017].
- [28] Arduino, «Arduino,» [En línea]. Disponible: <https://store.arduino.cc/usa/arduino-uno-rev3> . [Último acceso: 21 10 2017].
- [29] IFTTT, «IFTTT,» [En línea]. Disponible: <https://ifttt.com/discover>. [Último acceso: 21 10 2017].
- [30] MQTT, «MQTT,» [En línea]. Disponible: <http://mqtt.org/>. [Último acceso: 21 10 2017].
- [31] Raspberry, «Raspberry,» [En línea]. Disponible: <https://www.raspberrypi.org/>. [Último acceso: 21 10 2017].
- [32] Universidad Pedagógica Experimental Libertador, Manual de Trabajos de Grado de Especialización y Maestría y Tesis Doctorales, Caracas: FEDUPEL, 2005.

BIBLIOGRAFÍA

OpenHAB Community and OpenHAB Foundation, [En línea]
<http://www.openhab.org>. [Consulta: 2017].

GLOSARIO

• **Amazon Eco:** altavoz manos libres controlado por voz. Eco se conecta al Servicio de voz de Alexa para reproducir música, hacer llamadas, enviar y recibir mensajes, proporcionar información, noticias, puntajes deportivos, clima y mucho más, al instante. [27]

• **Arduino Uno:** placa de microcontrolador basada en ATmega328P. Tiene 14 pines digitales de entrada / salida (de los cuales 6 se pueden usar como salidas PWM), 6 entradas analógicas, un cristal de cuarzo de 16 MHz, una conexión USB, un conector de alimentación, un encabezado ICSP y un botón de reinicio. [28]

• **IFTTT:** es un servicio gratuito basado en web para crear cadenas de enunciados condicionales simples, llamados applets. Un applet se desencadena por los cambios que se producen en otros servicios web, como Gmail, Facebook, Instagram o Pinterest. [29]

• **MQTT:** es un protocolo de mensajería "ligero" basado en la publicación-subscripción estándar ISO (ISO / IEC PRF 20922) para usar sobre el protocolo TCP / IP. [30]

• **Raspberry Pi:** computadoras de una sola placa desarrolladas en el Reino Unido por la Fundación Raspberry Pi para promover la enseñanza de la informática básica en las escuelas y en los países en desarrollo. [31]

ANEXOS

ANEXO A

MANUAL DE CONFIGURACIÓN Y USO

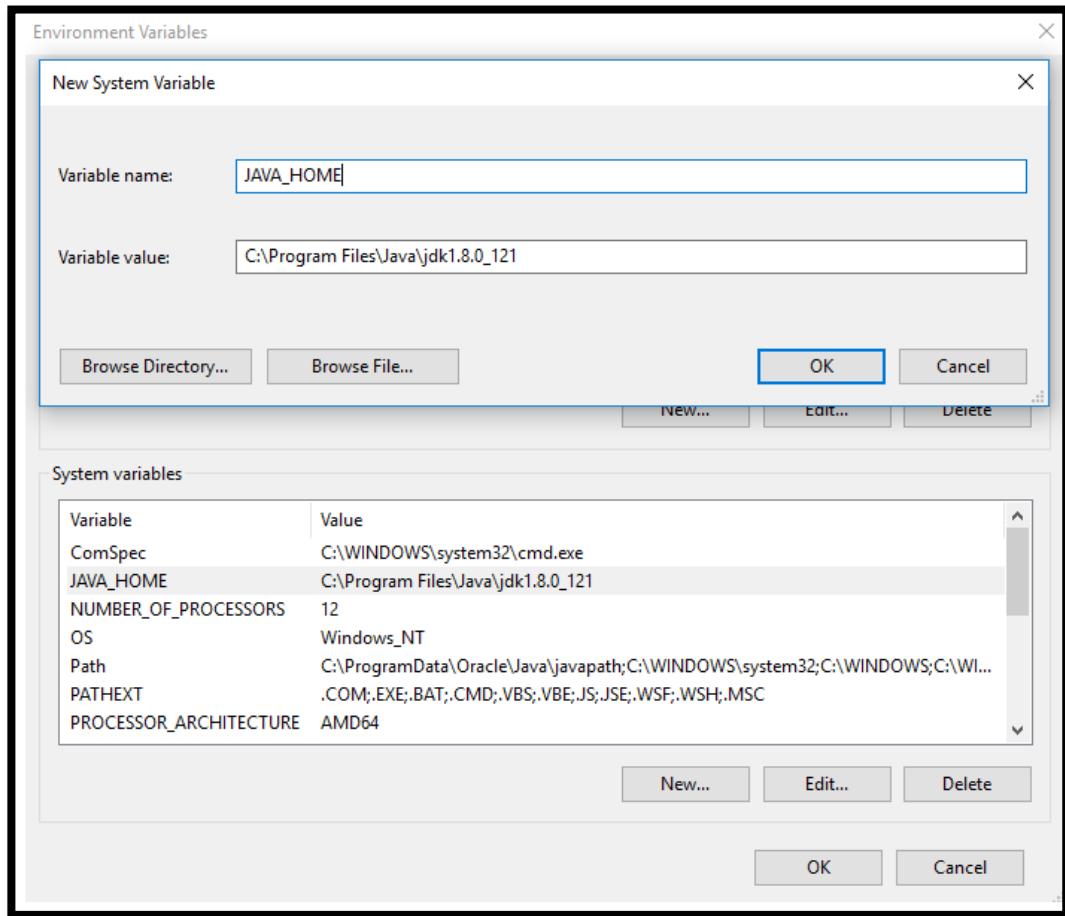
Requisitos previos

En primer lugar, se debe contar con un equipo con sistema operativo Windows de 64 bits, el cual debe tener instalada una plataforma Java actualizada pero esta debe ser una versión de 32 bits, debido a la existencia de una serie de conexiones que no funcionan con la versión de 64 bits, como por ejemplo Z-Wave. No se recomienda la instalación de Java 9, debido a que no es compatible con OpenHAB 2.0.

Para asegurar una mejor compatibilidad en las conexiones, la versión recomendada de Java 8 es la 101, es decir al momento de solicitar o verificar la versión de Java instalada en su equipo la información debe ser “*java versión 1.8.0_121*”. Una vez se verifique la versión instalada de Java 8, se debe proceder a crear “*JAVA_HOME*” como una nueva variable de entorno del sistema, para esto se debe ir a:

Panel de control ► Sistema y seguridad ► Sistema ► Configuración avanzada del sistema ► Variables de entorno ► Variables del sistema

Luego de acceder a las *Variables del sistema* se debe crear una nueva variable de sistema basada en el directorio de instalación de Java como se muestra a continuación:

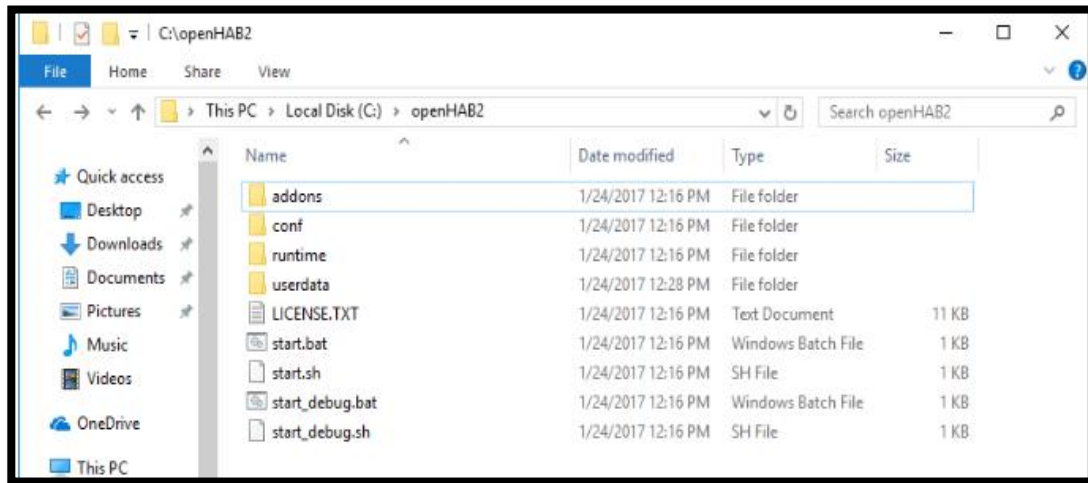


Adicional a esto se debe instalar *Eclipse SmartHome Designer* para administrar los archivos de configuración, este software cuenta con soporte incorporado para la sintaxis OpenHAB y elementos.

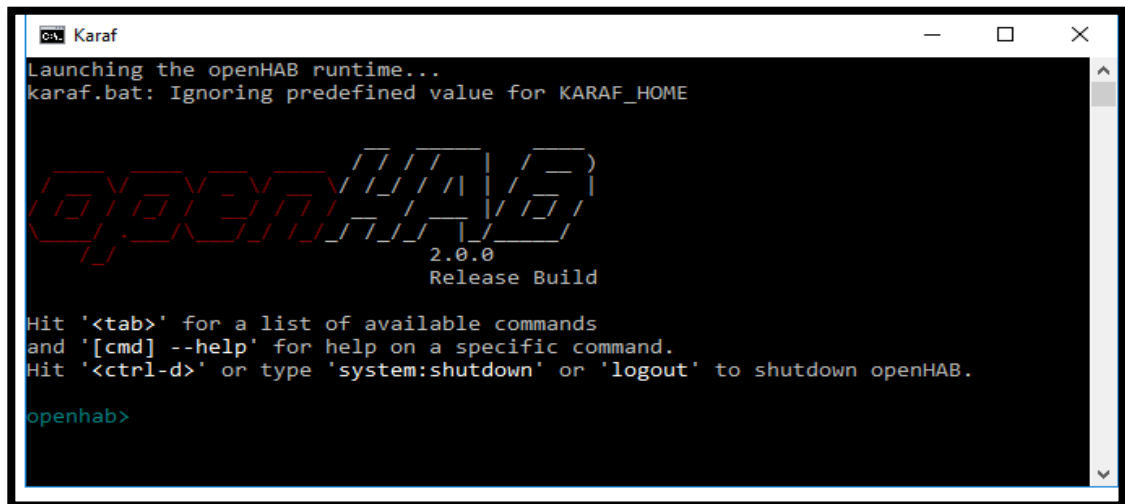
Instalación

Los pasos para instalar OpenHAB son los siguientes:

1. Copie y pegue en su computadora el archivo “*Sistema Inmotico Propuesto.rar*” que se encuentra en el CD anexo.
2. Descomprimir el archivo en el directorio elegido, por ejemplo “C:\openHAB2”

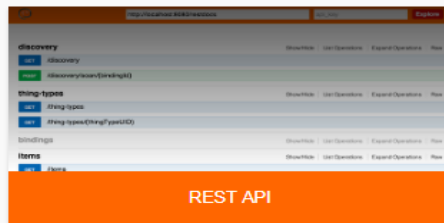
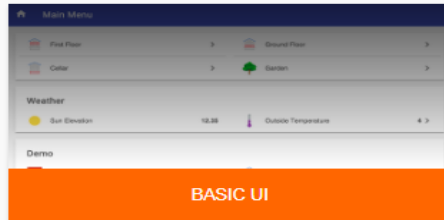


3. Inicie el servidor ejecutando el archivo “C:\openHAB2\start.bat” y espere un momento para que se inicie.



4. Diríjase a la siguiente dirección <http://localhost:8080> en el navegador de su preferencia y seleccione la interfaz *BASIC UI* este contendrá la interfaz de usuario diseñada en esta propuesta.

Welcome to openHAB 2



Getting started? Please refer to the [online documentation](#).

Instalación como servicio de Windows

1. Complete los requisitos previos y los pasos de la instalación regular.
2. Emita los siguientes comandos en la consola de OpenHAB

```
feature:install service-wrapper
wrapper:install --name "openHAB2" --display "openHAB2" --description "openHAB 2 Service"

Karaf
Launching the openHAB runtime...
karaf.bat: Ignoring predefined value for KARAF_HOME

openHAB2
2.8.0
Release Build

Hit '<tab>' for a list of available commands
and '[cmd] --help' for help on a specific command.
Hit '<ctrl-d>' or type 'system:shutdown' or 'logout' to shutdown openHAB.

openhab> feature:install service-wrapper
openhab> wrapper:install --name "openHAB2" --display "openHAB2" --description "openHAB 2 Service"
Creating missing directory: C:\openHAB2\userdata\bin
Creating file: C:\openHAB2\userdata\bin\openHAB2-wrapper.exe
Creating file: C:\openHAB2\userdata\etc\openHAB2-wrapper.conf
Creating file: C:\openHAB2\userdata\bin\openHAB2-service.bat
Creating missing directory: C:\openHAB2\userdata\lib\wrapper
Creating file: C:\openHAB2\userdata\lib\wrapper\wrapper.dll
Creating file: C:\openHAB2\userdata\lib\wrapper\karaf-wrapper.jar
Creating file: C:\openHAB2\userdata\lib\wrapper\karaf-wrapper-main.jar

Setup complete. You may wish to tweak the JVM properties in the wrapper configuration file:
C:\openHAB2\userdata\etc\openHAB2-wrapper.conf
before installing and starting the service.

MS Windows system detected:
To install the service, run:
C:> C:\openHAB2\userdata\bin\openHAB2-service.bat install

Once installed, to start the service run:
C:> net start "openHAB2"

Once running, to stop the service run:
C:> net stop "openHAB2"

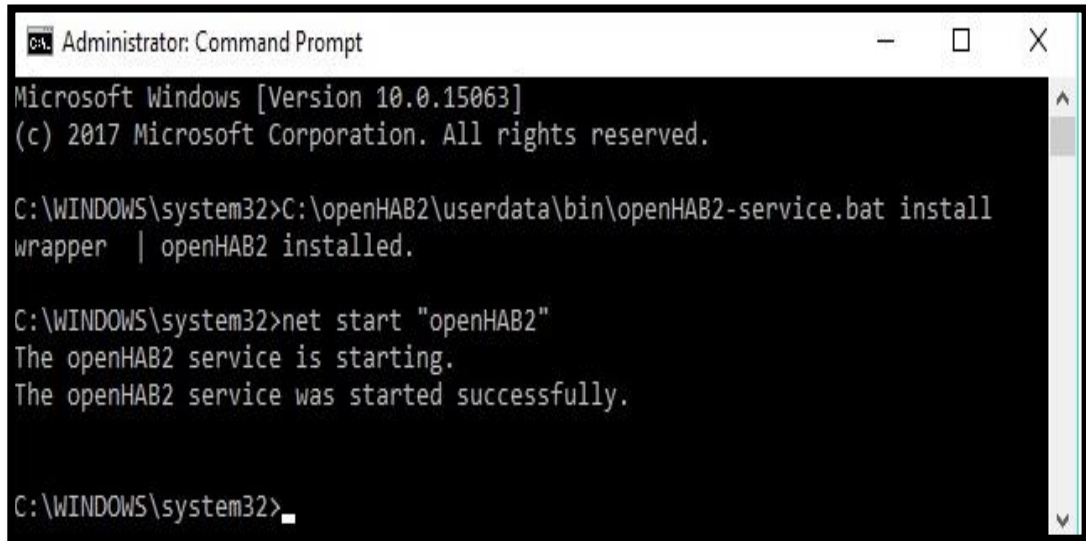
Once stopped, to remove the installed service run:
C:> C:\openHAB2\userdata\bin\openHAB2-service.bat remove

openhab>
```

3. Apague la instancia OpenHAB escribiendo *logout* en la consola de ejecución de OpenHAB.

4. Actualice la nueva creación `C:\openHAB2\userdata\etc\openHAB2-wrapper.conf` para incluir todos los parámetros necesarios, copiando todo el texto de configuración de la sección que se encuentra la sección de instalación de “Pasos de instalación del servicio de Windows” en el siguiente link <http://docs.openhab.org/installation/windows.html#file-locations> y péguelo en su `openHAB2-wrapper.conf`, reemplazando todo el contenido existente.

5. Adapte la primera entrada (*OPENHAB_HOME*) para que coincida con su directorio de instalación de OpenHAB.
6. Abra un símbolo del sistema elevado y escriba los siguientes comandos *C:\openHAB2\userdata\bin\openHAB2-service.bat install* y *net start "openHAB2"*



```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.15063]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>C:\openHAB2\userdata\bin\openHAB2-service.bat install
wrapper | openHAB2 installed.

C:\WINDOWS\system32>net start "openHAB2"
The openHAB2 service is starting.
The openHAB2 service was started successfully.

C:\WINDOWS\system32>
```

7. Su servicio OpenHAB Windows ya está instalado y en funcionamiento. Se puede validar las operaciones adecuadas a través de la siguiente dirección <http://localhost:8080> y comprobando de que el servicio de Windows está en ejecución y configurado en tipo de inicio automático.

OpenHAB Cloud Connector

Este servicio ya se encuentra instalado en la propuesta y permite el acceso remoto al servidor local de OpenHAB permitiendo que a través de una cuenta en myopenHAB.org la cual se creó y configuró específicamente para el sistema inmótico diseñado, siendo el usuario “*inmótica.eie@gmail.com*” y la clave de acceso “*eieUCV3233*”

Creación de usuarios

Una vez se haya accedido a la cuenta de myopenHAB.org en la sección de “Users” existe la posibilidad de crear usuarios, los cuales tendrán acceso al servidor local. Es importante resaltar que la creación de esta cuenta es obligatoria para poder configurar la aplicación móvil que permite gestionar y tener acceso como usuario del sistema inmótico propuesto.

The screenshot shows the 'My Users' page in the openHAB web interface. At the top left is the openHAB logo with the tagline 'empowering the smart home'. To the right is a 'Navigate...' dropdown menu. Below this is an orange banner with the text 'My Users' and a breadcrumb 'HOME / USERS'. The main content area is divided into two columns. The left column has a '+ Add new user' button and a list of existing users: 'inmotica.eie@gmail.com' and 'cecilio_fr-5000@hotmail.com'. The right column contains a form with the following fields: 'Email address', 'Password', 'Verify password', and a dropdown menu set to 'Regular user'. Below the form is an 'Add user' button. At the bottom of the page, there is a footer with the text: 'Copyright © 2016 by the openHAB Community and the openHAB Foundation'.

Configuración de la aplicación móvil

Antes que nada se debe realizar la descarga e instalación de la aplicación móvil “openHAB” según el sistema operativo, la misma se encuentra disponible para dispositivos Android, IOS y Windows Phone. Lo primero que se debe hacer en la aplicación es entrar a las configuraciones y deshabilitar el modo demo, el cual se ejecuta por defecto, una vez hecho esto se procede a completar los campos “Nombre

de usuario” y “Contraseña” correspondientes al “Email address” y “Password” configurados en la cuenta master de myopenHAB.org

