



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE ESTUDIOS SUPERIORES CUAUTITLÁN

INTERFAZ INTELIGENTE PARA EL CONTROL DE UN
SISTEMA DOMÓTICO GESTIONADO POR *PROLOG*.

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

INGENIERO MECÁNICO ELÉCTRICO

P R E S E N T A :

VICTOR ADRIAN ESPINOSA DE LOS MONTEROS PARRA

TUTOR

DR. DAVID TINOCO VARELA

CUAUTITLÁN IZCALLI, ESTADO DE MÉXICO, 2022



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

**FACULTAD DE ESTUDIOS SUPERIORES CUAUTITLÁN
SECRETARÍA GENERAL
DEPARTAMENTO DE TITULACIÓN**

U.N.A.M.
FACULTAD DE ESTUDIOS
SUPERIORES CUAUTITLÁN
ASUNTO: VOTO APROBATORIO



**Jefa del Departamento de Titulación
de la FES Cuautitlán**
DEPARTAMENTO DE
EXÁMENES PROFESIONALES

**M. en C. JORGE ALFREDO CUÉLLAR ORDAZ
DIRECTOR DE LA FES CUAUTITLÁN
PRESENTE**

**ATN: I.A. LAURA MARGARITA CORTAZAR FIGUEROA
Jefa del Departamento de Titulación**

Con base en el Reglamento General de Exámenes, y la Dirección de la Facultad, nos permitimos comunicar a usted que revisamos el: **Trabajo de tesis y examen profesional**

Interfaz Inteligente para el Control de un Sistema Domótico Gestionado por PROLOG

Que presenta el pasante: **Victor Adrian Espinosa de los Monteros Parra**
Con número de cuenta: **309093011** para obtener el título de: **Ingeniero Mecánico Eléctrico**

Considerando que dicho trabajo reúne los requisitos necesarios para ser discutido en el **EXAMEN PROFESIONAL** correspondiente, otorgamos nuestro **VOTO APROBATORIO**.

ATENTAMENTE
"POR MI RAZA HABLARÁ EL ESPÍRITU"
Cuautitlán Izcalli, Méx. a 04 de octubre de 2021.

PROFESORES QUE INTEGRAN EL JURADO

	NOMBRE	FIRMA
PRESIDENTE	Ing. Julio Cesar Vázquez Fuentes	
VOCAL	Ing. Noemí Hernández Domínguez	
SECRETARIO	Dr. David Tinoco Varela	
1er. SUPLENTE	M. en C. Leopoldo Martín Del Campo Ramírez	
2do. SUPLENTE	Dr. Fernando Gudiño Peñaloza	

NOTA: los sinodales suplentes están obligados a presentarse el día y hora del Examen Profesional

LMCF/cga*



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

**FACULTAD DE ESTUDIOS SUPERIORES CUAUTITLÁN
SECRETARÍA GENERAL
DEPARTAMENTO DE TITULACIÓN**

ASUNTO: VOTO **APROBATORIO**

UNAM
FACULTAD DE ESTUDIOS
SUPERIORES CUAUTITLÁN



DEPARTAMENTO DE
TITULACIÓN
DE LA FES CUAUTITLÁN
PROFESIONALES

**M. en C. JORGE ALFREDO CUÉLLAR ORDAZ
DIRECTOR DE LA FES CUAUTITLÁN
PRESENTE**

**ATN: I.A. LAURA MARGARITA CORTAZAR FIGUEROA
Jefa del Departamento de Titulación**

Con base en el Reglamento General de Exámenes, y la Dirección de la Facultad, nos permitimos comunicar a usted que revisamos el: **Trabajo de tesis y examen profesional**

Interfaz Inteligente para el Control de un Sistema Domótico Gestionado por PROLOG

Que presenta el pasante: **Victor Adrian Espinosa de los Monteros Parra**
Con número de cuenta: **309093011** para obtener el título de: **Ingeniero Mecánico Eléctrico**

Considerando que dicho trabajo reúne los requisitos necesarios para ser discutido en el **EXAMEN PROFESIONAL** correspondiente, otorgamos nuestro **VOTO APROBATORIO**.

ATENTAMENTE
"POR MI RAZA HABLARÁ EL ESPÍRITU"
Cuautitlán Izcalli, Méx. a 04 de octubre de 2021.

PROFESORES QUE INTEGRAN EL JURADO

	NOMBRE	FIRMA
PRESIDENTE	Ing. Julio Cesar Vázquez Fuentes	_____
VOCAL	Ing. Noemí Hernández Domínguez	
SECRETARIO	Dr. David Tinoco Varela	_____
1er. SUPLENTE	M. en C. Leopoldo Martín Del Campo Ramírez	_____
2do. SUPLENTE	Dr. Fernando Gudiño Peñaloza	_____

NOTA: los sinodales suplentes están obligados a presentarse el día y hora del Examen Profesional

LMCF/cga*



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

**FACULTAD DE ESTUDIOS SUPERIORES CUAUTITLÁN
SECRETARÍA GENERAL
DEPARTAMENTO DE TITULACIÓN**

U.N.A.M.
FACULTAD DE ESTUDIOS
SUPERIORES CUAUTITLÁN
ASUNTO: VOTO APROBATORIO



DEPARTAMENTO DE
TITULACIÓN DE
LA FES CUAUTITLÁN
EXÁMENES PROFESIONALES

**M. en C. JORGE ALFREDO CUÉLLAR ORDAZ
DIRECTOR DE LA FES CUAUTITLÁN
PRESENTE**

**ATN: I.A. LAURA MARGARITA CORTAZAR FIGUEROA
Jefa del Departamento de Titulación**

Con base en el Reglamento General de Exámenes, y la Dirección de la Facultad, nos permitimos comunicar a usted que revisamos el: **Trabajo de tesis y examen profesional**

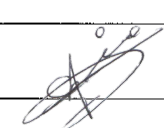
Interfaz Inteligente para el Control de un Sistema Domótico Gestionado por PROLOG

Que presenta el pasante: **Victor Adrian Espinosa de los Monteros Parra**
Con número de cuenta: 309093011 para obtener el título de: Ingeniero Mecánico Eléctrico

Considerando que dicho trabajo reúne los requisitos necesarios para ser discutido en el **EXAMEN PROFESIONAL** correspondiente, otorgamos nuestro **VOTO APROBATORIO**.

ATENTAMENTE
"POR MI RAZA HABLARÁ EL ESPÍRITU"
Cuautitlán Izcalli, Méx. a 04 de octubre de 2021.

PROFESORES QUE INTEGRAN EL JURADO

	NOMBRE	FIRMA
PRESIDENTE	Ing. Julio Cesar Vázquez Fuentes	_____
VOCAL	Ing. Noemí Hernández Domínguez	_____
SECRETARIO	Dr. David Tinoco Varela	
1er. SUPLENTE	M. en C. Leopoldo Martín Del Campo Ramírez	_____
2do. SUPLENTE	Dr. Fernando Gudiño Peñaloza	_____

NOTA: los sinodales suplentes están obligados a presentarse el día y hora del Examen Profesional

LMCF/cga*



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

**FACULTAD DE ESTUDIOS SUPERIORES CUAUTITLÁN
SECRETARÍA GENERAL
DEPARTAMENTO DE TITULACIÓN**

U.N.A.M.
FACULTAD DE ESTUDIOS
SUPERIORES CUAUTITLÁN
ASUNTO: VOTO APROBATORIO



DEPARTAMENTO DE
TITULACIÓN DE
LA FES CUAUTITLÁN
EXÁMENES PROFESIONALES

**M. en C. JORGE ALFREDO CUÉLLAR ORDAZ
DIRECTOR DE LA FES CUAUTITLÁN
PRESENTE**

**ATN: I.A. LAURA MARGARITA CORTAZAR FIGUEROA
Jefa del Departamento de Titulación**

Con base en el Reglamento General de Exámenes, y la Dirección de la Facultad, nos permitimos comunicar a usted que revisamos el: **Trabajo de tesis y examen profesional**

Interfaz Inteligente para el Control de un Sistema Domótico Gestionado por PROLOG

Que presenta el pasante: **Victor Adrian Espinosa de los Monteros Parra**
Con número de cuenta: 309093011 para obtener el título de: Ingeniero Mecánico Eléctrico

Considerando que dicho trabajo reúne los requisitos necesarios para ser discutido en el **EXAMEN PROFESIONAL** correspondiente, otorgamos nuestro **VOTO APROBATORIO**.

ATENTAMENTE
"POR MI RAZA HABLARÁ EL ESPÍRITU"
Cuautitlán Izcalli, Méx. a 04 de octubre de 2021.

PROFESORES QUE INTEGRAN EL JURADO

	NOMBRE	FIRMA
PRESIDENTE	Ing. Julio Cesar Vázquez Fuentes	_____
VOCAL	Ing. Noemí Hernández Domínguez	_____
SECRETARIO	Dr. David Tinoco Varela	_____
1er. SUPLENTE	M. en C. Leopoldo Martín Del Campo Ramírez	
2do. SUPLENTE	Dr. Fernando Gudiño Peñaloza	_____

NOTA: los sinodales suplentes están obligados a presentarse el día y hora del Examen Profesional

LMCF/cga*



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

**FACULTAD DE ESTUDIOS SUPERIORES CUAUTITLÁN
SECRETARÍA GENERAL
DEPARTAMENTO DE TITULACIÓN**

U.N.A.M.
FACULTAD DE ESTUDIOS
SUPERIORES CUAUTITLÁN
ASUNTO: VOTO APROBATORIO

**M. en C. JORGE ALFREDO CUÉLLAR ORDAZ
DIRECTOR DE LA FES CUAUTITLÁN
PRESENTE**

ATN: I.A. LAURA MARGARITA CORTAZAR ECHEGURRO
Jefa del Departamento de Titulación
de la FES Cuautitlán
DEPARTAMENTO DE
EXÁMENES PROFESIONALES

Con base en el Reglamento General de Exámenes, y la Dirección de la Facultad, nos permitimos comunicar a usted que revisamos el: **Trabajo de tesis y examen profesional**

Interfaz Inteligente para el Control de un Sistema Domótico Gestionado por PROLOG

Que presenta el pasante: **Victor Adrian Espinosa de los Monteros Parra**
Con número de cuenta: **309093011** para obtener el título de: **Ingeniero Mecánico Eléctrico**

Considerando que dicho trabajo reúne los requisitos necesarios para ser discutido en el **EXAMEN PROFESIONAL** correspondiente, otorgamos nuestro **VOTO APROBATORIO**.

ATENTAMENTE
"POR MI RAZA HABLARÁ EL ESPÍRITU"
Cuautitlán Izcalli, Méx. a 04 de octubre de 2021.

PROFESORES QUE INTEGRAN EL JURADO

	NOMBRE	FIRMA
PRESIDENTE	Ing. Julio Cesar Vázquez Fuentes	
VOCAL	Ing. Noemí Hernández Domínguez	
SECRETARIO	Dr. David Tinoco Varela	
1er. SUPLENTE	M. en C. Leopoldo Martín Del Campo Ramírez	
2do. SUPLENTE	Dr. Fernando Gudiño Peñaloza	

NOTA: los sinodales suplentes están obligados a presentarse el día y hora del Examen Profesional

LMCF/cga*

A toda mi familia y amigos que siempre han sido un apoyo incondicional, a todos los que me han permitido aprender algo nuevo cada día y me motivan a dar siempre ese extra para cumplir mis sueños y metas...

Agradecimientos

Gracias a mis padres Eduardo y Esther que siempre han sido mi más grande apoyo en toda circunstancia y pese a cualquier adversidad que se presenta en la vida; que han sido mi motor e inspiración para dar siempre el siguiente paso y llegar más allá hasta lograr todas y cada una de mis metas, por creer en mi e inculcarme los valores que me hacen ser quien hoy soy. A mis hermanos Adriana y Eduardo, quienes me han demostrado que no existen límites cuando tenemos un objetivo y, más importante aún, tenemos el coraje y las ganas de cumplirlos, por ser un ejemplo a seguir y aconsejarme cuando lo he necesitado. A Yesenia por ser mi pilar y apoyarme en todo, motivarme a cumplir mis sueños y confiar en que puedo lograr todo lo que me proponga.

Al doctor David Tinoco por creer en mí y en mis capacidades, por brindarme todo su apoyo para desarrollar este proyecto y motivarme a seguir creciendo en el área de la investigación. A mis compañeros de la carrera que siempre me brindaron su apoyo y con quienes aprendí y me formé como un profesionista.

A la Universidad Nacional Autónoma de México por ser mi centro de estudios que me permitió ser parte de la experiencia de estudiar en el extranjero en dos ocasiones diferentes y a todos sus profesores quienes aportaron algo para mi desarrollo académico e impulsaron mi preparación para enfrentarme a la vida laboral.

Agradecimiento especial a los proyectos PAPIME PE100221 y PIAPI 2053 de la UNAM y la FES-C.

Índice general

Agradecimientos	VII
1. Introducción	1
1.1. Planteamiento del problema	2
1.2. Justificación	3
1.3. Antecedentes	4
1.4. Objetivos	5
1.4.1. Objetivo general	5
1.4.2. Objetivo específico	5
2. Preliminares	6
2.1. Inteligencia artificial	6
2.2. Domótica	7
2.3. PROLOG	8
2.3.1. Elementos básicos de PROLOG	9
2.4. Interfaces humano máquina	12
2.5. LabView	12
2.6. Protocolos	13
3. Desarrollo del Sistema	14
3.1. Sensores	14
3.1.1. Sensor de temperatura (DS18B20)	15
3.1.2. Sensor de luz (BH1750)	16
3.2. Actuadores	18
3.2.1. Motor a pasos	18
3.2.2. Motor 28BYJ-48 con módulo controlador ULN2003	19
3.3. Conexiones electrónicas	20
3.4. Lógica de programación (Programación inteligente)	21
3.5. Recolección de datos	28
3.6. Gestión de datos	30
3.7. Interacción total de software y hardware	30
4. Análisis de resultados	33
4.1. Tiempos de respuesta	33
4.2. Ventajas contra otros sistemas	34

Capítulo 1

Introducción

PROLOG es un lenguaje de programación enfocado principalmente a aplicaciones de inteligencia artificial (IA), este lenguaje fue desarrollado en la *Universidad de Aix-Marseille I* (Marsella, Francia) por los estudiantes *Alain Colmerauer* y *Philippe Roussel*.

Dentro de las principales características de este lenguaje se pueden mencionar que está basado en nociones matemáticas de relaciones de inferencia, su esquema de funcionamiento se basa en los fundamentos lógicos, además de que un programa de *PROLOG* se enfoca más en el conocimiento (bases de datos existentes que se ingresan al programa para su análisis) que en los algoritmos. Una característica interesante, es que a diferencia de la mayoría de lenguajes de programación, *PROLOG* busca llegar a un resultado, en lugar de centrarse en cómo llegar a ese resultado, se usa para representar conocimientos sobre un determinado dominio y las relaciones entre objetos de ese dominio. Para la generación de un programa en este lenguaje, es necesario tener los datos conocidos (hechos) y como se relacionan estos datos (reglas), y a partir de tales datos, poder llegar a una resolución lógica del problema planteado.

Debido a las características mencionadas, *PROLOG* es ampliamente utilizado en sistemas expertos (SE), los cuales son programas que buscan reproducir el proceso intelectual de un humano experto en un campo particular, en la literatura existen una gran cantidad de SE que han sido aplicados a diferentes ramas del conocimiento tales como sistemas médicos, sistemas de detección de fallas en sistemas de distribución, sistemas de diagnóstico psiquiátrico, entre otros.

A pesar de la facilidad para realizar IA con *PROLOG*, este lenguaje no ha sido ampliamente explotado cuando se trata de aplicaciones de hardware, mayoritariamente se ha enfocado en aplicaciones de software y con datos que vienen directamente del usuario o de sistemas informáticos. Han existido proyectos que tratan de interactuar con elementos reales, sin embargo, esto se ha reducido a la simulación de las variables físicas[18].

Además de los SE, este lenguaje también se ha utilizado para otro tipo de apli-

caciones, tal como la generación de un paradigma computacional en el que se busca representar y controlar conversaciones computadora-humano.

A pesar de las capacidades de procesamiento y de respuesta inteligente que tiene este lenguaje (fue el primer lenguaje de programación diseñado específicamente para la generación de inteligencia artificial), generalmente se ha dejado de lado la interacción con elementos físicos reales, gestionando principalmente información ingresada directamente por software. En este punto radica la importancia de este proyecto, ya que se pretende gestionar información proveniente desde un entorno real, y controlar un sistema electrónico inteligente.

1.1. Planteamiento del problema

La automatización es un fenómeno a gran escala que ha ido incursionando principalmente a nivel industrial, lo que reduce costos de mano de obra y garantiza una mejor calidad en el desarrollo de procesos repetitivos; de igual manera aumenta las horas de producción respecto a operadores humanos. A partir del desarrollo de estos nuevos procesos, la automatización ha incursionado en el desarrollo de diferentes áreas, no solo a nivel industrial, entre las que se encuentran el área de la domótica, la cual se involucra en la automatización a nivel residencial para el desarrollo de casas inteligentes que permitan el control y monitoreo de distintos fenómenos que ocurren en el día a día en el entorno del inmueble. La domótica es una especialidad en desarrollo que actualmente está siendo explotada para el mejoramiento de calidad de vida y seguridad en edificios residenciales y casas en particular. Con ello la demanda de usuarios incrementa y distintas empresas entran en el desarrollo de dispositivos para automatizar desde el encendido de una lámpara, hasta el control de clima e iluminación para habitaciones. Además de los dispositivos que interactúan, el software, con el cual son controlados y que permite monitorearlos, es un factor muy importante. En el diseño comercial de estos sistemas, cada empresa que diseña un dispositivo electrónico, genera su propio software para realizar el control y monitoreo del mismo. De acuerdo a los fines técnicos y comerciales de la empresa, es el tipo de lenguaje que implementa para programar los dispositivos. Es interesante mencionar, que aunque la finalidad de la domótica es generar casas inteligentes, los lenguajes de programación utilizados para tal fin son lenguajes de uso general, tal como C++ o Java.

PROLOG es un lenguaje de programación que fue diseñado principalmente para el desarrollo de inteligencia artificial. A pesar de las características con las que cuenta este lenguaje, actualmente no existen aplicaciones en las que *PROLOG* gestione información proveniente de un entorno real, centrándose principalmente en la obtención de datos por medio de software; y a diferencia de C++ o Java, este lenguaje permitirá una facilidad de trabajo para el desarrollo de estos sistemas ya que no hay que diseñar los esquemas inteligentes, estos ya están inmersos en el lenguaje. Es en este punto donde el presente proyecto de tesis tiene gran relevancia, ya que *PROLOG* es un lenguaje poco explorado para el diseño de sistemas domóticos inteligentes.

Dado el alcance y facilidad de implementación de este lenguaje de programación y el bajo costo de recursos a nivel máquina para implementarlo, se generará una interfaz que pueda aprovechar todas sus características para que interactúe en tiempo real con un sistema físico por medio de sensores, con lo cual se puede tener un control inteligente del sistema propuesto.

1.2. Justificación

Actualmente, con el desarrollo de nuevos esquemas tecnológicos tales como el Internet de las Cosas (IoT) o la industria 4.0, es necesario la generación y diseño de edificaciones, en este caso casas, que puedan interactuar de manera autónoma con el entorno, y con diferentes elementos tecnológicos, para tal fin es necesario que dichas edificaciones puedan responder de forma inteligente, para así poder reaccionar de la mejor manera a las necesidades del usuario, conocidas o desconocidas, así como también a las condiciones del entorno. Este tipo de sistemas actúan recibiendo una señal por medio de sensores y producen una respuesta mediante actuadores, tales como una alarma, un mensaje de texto al celular del usuario, un corte energía en el sistema eléctrico de la vivienda, control de un ventilador, etc.

El potencial de la domótica consiste en centralizar en un único sistema de control todas las aplicaciones posibles de automatizar.

Estas nuevas tecnologías contribuyen a mejorar la comodidad, la seguridad y el bienestar dentro de los hogares, de ahí que cada vez tengan más presencia. Una consecuencia directa de esta situación es que la domótica se haya convertido en una de las tecnologías con más futuro.

Gracias a la domótica se puede fácilmente automatizar un hogar, permitiendo por ejemplo automatizar la gestión de la climatización de una casa, mejorando el confort y el consumo, la domótica también permite mejorar considerablemente la seguridad del hogar y mejorar la red de comunicaciones.

Por las situaciones mencionadas, se propone el uso del lenguaje PROLOG para la generación de un sistema domótico inteligente, ya que cuenta con características inherentes que le permiten realizar un control inteligente de forma natural y sencilla. La base de conocimientos que está relacionada a este proyecto, permitira poder gestionar de forma eficiente las magnitudes físicas de entrada y el control lógico de los actuadores de salida.

PROLOG es un lenguaje sencillo de implementar, pero potente, y presenta una característica adicional, es un lenguaje económico en su utilización, ya que es de uso libre.

1.3. Antecedentes

Como ya se ha mencionado, *PROLOG* es un lenguaje que a pesar de sus características inherentes, este ha sido enfocado en la generación de bases de conocimiento o sistemas expertos, basados únicamente en la obtención de datos a través de software, o de forma manual, dejando de lado la adquisición de datos desde un entorno físico.

Los SE, como se menciona anteriormente, son programas que buscan reproducir el proceso intelectual de un humano experto en un campo particular, utilizado todo el conocimiento adquirido por un individuo o un grupo de expertos de un área particular de conocimiento. Con ellos se busca generar sistemas que puedan llegar a conclusiones a partir de hechos de entrada, por ejemplo, nosotros podemos ingresar diferentes síntomas a un SE, y él nos definirá de que enfermedad se trata.

En la actualidad los SE, son considerados dentro del campo de la inteligencia artificial. Estos sistemas permiten la creación de máquinas que razonan como el hombre, restringiéndose a un espacio de conocimientos limitado. En teoría pueden razonar siguiendo los pasos que seguiría un humano experto (médico, analista, empresario, etc.) para resolver un problema concreto. Este tipo de modelos de conocimiento por ordenador ofrece un extenso campo de posibilidades en resolución de problemas y en aprendizaje.

De acuerdo a lo antes mencionado, se puede observar que un SE puede encontrarse inmerso en cualquier área del conocimiento. *PROLOG* funciona muy bien en el diseño y desarrollo de diferentes SE, este paradigma de programación ha servido para la generación de SE que se encargan de diferentes áreas del conocimiento tales como el diagnóstico de fallas de subestaciones eléctricas [1], y más interesante aún, para electromiografía (EMG) que ayuda a un médico a diagnosticar y planificar la prueba EMG dentro de la gama completa de trastornos neuromusculares [2].

Los sistemas expertos médicos, son ampliamente utilizados para ayudar a realizar un mejor diagnóstico y tratamiento para cada paciente y *PROLOG* ha mostrado ser sumamente funcional en este tipo de sistemas, por ejemplo, se tiene el SE que realiza el diagnóstico de enfermedades renales, tomando en cuenta la observación médica y los resultados de laboratorio mostrado en [3]. Por otro lado, en [4] se presentó un sistema médico experto, también basado en *PROLOG*, para diagnosticar las principales enfermedades pulmonares entre los pacientes. Según los autores, el diagnóstico se realiza teniendo en cuenta los síntomas que se pueden ver o sentir.

En el caso de la psiquiatría también se tienen antecedentes de SE desarrollados en la plataforma *PROLOG*, tales como el sistema experto para monitorear el tratamiento farmacológico en un entorno psiquiátrico, presentado en [5]. Por otro lado, en [6] se presentó *Psyxpert*, un prototipo de SE diseñado para ayudar a los psiquiatras en el diagnóstico de trastornos mentales, según los autores, utilizado en los casos en que las características psicóticas son la parte prominente del cuadro clínico.

En [7] se diseñó un sistema experto que puede diagnosticar algún tipo de enfermedad mental que este relacionada con el consumo de alcohol u otras sustancias enervantes.

Tomando en cuenta las referencias antes mencionadas, se observa que *PROLOG* tiene un amplio alcance y funcionalidad en el área de los SE, ya que ha sido utilizado en distintas campos científicos. De acuerdo a investigaciones previas realizadas para el desarrollo de este proyecto, no existen (a conocimiento del autor) aplicaciones de *PROLOG* que gestionen la interacción de un medio físico real a través de sensores con una interfaz visual en la que se controle el sistema mediante *PROLOG*. Siendo [8] el único afín a lo que se ha desarrollado en este proyecto; cabe mencionar que dicho proyecto fue realizado por compañeros de generación de la misma facultad, pero controlando un único sensor.

Para el desarrollo de este proyecto, se generará un sistema experto que estará dentro del motor de inferencia del control inteligente propuesto, lo interesante es que este SE, se llenará con la información que se obtenga directamente de los datos obtenidos por los sensores. Haciendo de este un SE autónomo.

1.4. Objetivos

1.4.1. Objetivo general

Ampliar el panorama de aplicación del lenguaje de programación *PROLOG* que existe actualmente, utilizándolo como un gestor de información orientado al control de sistemas domóticos utilizando las características del procesamiento inteligente con los que cuenta el lenguaje, en este caso, mediante la gestión de bases de datos para el análisis de fenómenos sentidos en el entorno.

1.4.2. Objetivo específico

Implementar un control y monitoreo para un sistema domótico por medio de *PROLOG*, utilizando sensores que interactúen de manera continua en un entorno con necesidades reales para que sea eficiente el aprovechamiento de recursos en un edificio. Plantear una interfaz que en un futuro pueda controlar inmuebles grandes tal como un edificio domótico y la interfaz electrónica para el mismo fin.

Capítulo 2

Preliminares

Para poder explicar el desarrollo del proyecto, primero que nada, es necesario determinar los conceptos que lo comprenden, desde la parte técnica que representa el área en la que se desarrolla hasta el concepto de los programas o software que están involucrados en el proceso del mismo.

En esta sección se dará una breve descripción de algunos de los conceptos más fundamentales.

2.1. Inteligencia artificial

La Inteligencia artificial es el campo científico y tecnológico que busca generar entes virtuales o físicos que se puedan comportar como seres inteligentes, que puedan pensar, razonar, relacionar, interpretar, entre otras acciones, como si de un ser inteligente se tratara. En otras palabras, la IA es el concepto según el cual “las máquinas pueden pensar como seres humanos”. Este es un concepto difícil de explicar, debido a que la inteligencia en sí misma es un concepto abstracto difícil de poder definir con certeza. La figura 2.1 muestra una representación abstracta de la IA.

Normalmente, un sistema de IA es capaz de analizar datos en grandes cantidades, identificar patrones y tendencias y, por lo tanto, formular predicciones de forma automática, con rapidez y precisión. En otras palabras, analizan los datos y, en base a ellos, toman decisiones de forma autónoma.

Por lo tanto, hablar de IA es referirse a un sistema digital el cual puede reconocer y aprender de su entorno, guardándolo para sí mismo (como un ser viviente al tener una experiencia) y tomar decisiones basadas en las condiciones actuales y lo previamente experimentado. Todos los parámetros deben ser considerados o por lo menos en su mayoría, lo cual permite al sistema cubrir la mayor cantidad de las posibilidades, lo cual podría emular a cualquier ser humano tomando una decisión.



Figura 2.1: Representación abstracta de la Inteligencia Artificial: imagen recuperada de <https://pixabay.com/es/illustrations/inteligencia-artificial-cerebro-3382507/>

2.2. Domótica

La domótica (Figura 2.2) es el conjunto de tecnologías aplicadas al control y la automatización inteligente de una vivienda, que permite una gestión eficiente del uso de la energía, que aporta seguridad y confort, además de proporcionar la comunicación entre el usuario y el sistema.

El principio de funcionamiento de la domótica (o en general de la automatización), es la gestión y obtención de datos de un entorno físico (generalmente una base de datos adquirida a partir de sensores) que funcionan como las entradas del sistema; el procesamiento de dichos datos (esto dependerá de las condiciones y características deseadas del usuario las cuales se convertirán en los parámetros de la programación, o bien, los criterios de comportamiento de nuestro sistema); y por último, la emisión de ordenes (son las funciones que realizarán los actuadores que estén contemplados e involucrados en el sistema, los cuales pueden ser de diferentes tipos de acuerdo a las necesidades del sistema) que serán las salidas del sistema.

Al igual que cualquier proceso de automatización, es necesario implementar un monitoreo y control del proceso, en este caso es el monitoreo de la información obtenida de los sensores que determinarán el estado físico del entorno, la programación permitirá al sistema tomar decisiones de acuerdo a lo que se desea; de igual manera, el usuario puede controlar las respuestas del sistema en tiempo real, por ello es indispensable una interfaz que sea fácil de leer o interpretar por el usuario y que genere una respuesta prácticamente instantánea al momento de ser solicitada.

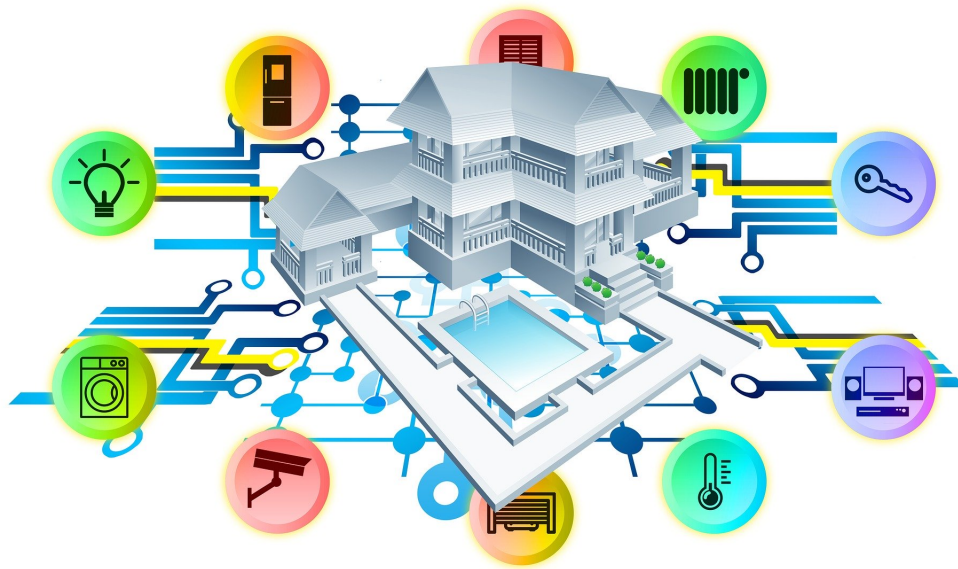


Figura 2.2: Sistema referencial de la Domótica: imagen recuperada de <https://pixabay.com/es/illustrations/casa-inteligente-casa-tecnología-2769239/>

2.3. PROLOG

PROLOG es un lenguaje de programación semi interpretado (sus instrucciones o código fuente escrito por el programador en un lenguaje de alto nivel, es traducido por el intérprete a un lenguaje entendible para la máquina paso a paso, instrucción por instrucción). Su funcionamiento es muy similar a Java. El código fuente se compila a un código de byte el cuál se interpreta en una máquina virtual denominada Warren Abstract Machine (comúnmente denominada WAM).

Por eso, un entorno de desarrollo *PROLOG* se compone de:

- **Compilador:** Transforma el código fuente en código de byte. A diferencia de Java, no existe un estándar al respecto. Por eso, el código de byte generado por un entorno de desarrollo no tiene por qué funcionar en el intérprete de otro entorno.
- **Intérprete:** Ejecuta el código de byte. Un *shell* o *top-level*. Se trata de una utilidad que permite probar los programas, depurarlos, etc. Su funcionamiento es similar a las interfaces de línea de comando de los sistemas operativos.
- **Biblioteca de utilidades:** Estas bibliotecas son, en general, muy amplias y en ellas se incluyen comandos y "palabras reservadas" (palabras preestablecidas) que realiza una instrucción predefinida por los desarrolladores. Muchos entornos incluyen unas bibliotecas *standard-ISO* que permiten funcionalidades básicas.

cas como manipular cadenas, entrada/salida, etc.

Generalmente, los entornos de desarrollo ofrecen extensiones al lenguaje como pueden ser la programación con restricciones, concurrente, orientada a objetos, etc.

El realizar un programa en *PROLOG* se tiene dos cuerpos principales: la especificación de los hechos y las preguntas sobre esos objetos o relaciones. Cuando se crea una base de conocimientos con ciertas especificaciones de hechos y reglas, el usuario podrá preguntar relaciones existentes entre tales especificaciones dando como resultado verdadero, si se logra concretar una cadena lógica de hechos; y falsa si no se logra concretar una cadena lógica de hechos.

PROLOG determina la respuesta explorando cada uno de los hechos introducidos en la base de datos hasta encontrar uno que coincida, que será el caso en el que la respuesta será afirmativa, o hasta que termine toda la base de datos, cuyo caso dará una respuesta negativa. Las preguntas que el usuario puede hacer sobre la base de conocimientos pueden ser más complejas usando operadores lógicos como *AND*, *OR* y *NOT*. En este caso, *PROLOG* busca que la primera parte de la pregunta sea respondida de forma verdadera y si esto no es así, busca en la segunda parte de la pregunta, es decir después del operador lógico.

2.3.1. Elementos básicos de PROLOG

Como todo lenguaje, *PROLOG* esta compuesto de una cantidad elevada de palabras reservadas, comandos y estructuras, hay que comprender que no es posible profundizar en el lenguaje dentro de este texto, ya que esto llevaría hojas y hojas de un solo tema, dejando de lado el tema principal. Por tal motivo, solo se presentarán algunos detalles generales del lenguaje.

1. Variables. Las variables en *PROLOG* deben de iniciar con letra mayúscula para poder identificarlas, es decir, es correcto usar la variable “Variable”, pero incorrecto definirla como “variable”.
2. Constantes. Los datos constantes deben de iniciar su definición con minúscula, para que puedan ser identificadas como tales. Es correcto definir una constante como “constante”, pero incorrecto definirla como “Constante”.
3. Afirmación. Cada afirmación debe ser terminada por medio de un punto, esto indicará al lenguaje la finalización de cada sentencia, y obviamente, la posibilidad de comenzar una nueva sentencia. NO se pueden nombrar constantes, variables, predicados, reglas, etc., con nombres compuestos separados por espacios, es incorrecto definir un nombre llamado “somos ingenieros”, pero es correcto definirlo como “somos_ingenieros”.

4. Comentarios. Se puede comentar el software por medio del símbolo reservado "%", cada que se coloque dicho símbolo, PROLOG entenderá que los enunciados posteriores a él, serán solo comentarios y no los tomará en cuenta para la lectura y ejecución del programa realizado.

El lenguaje de programación PROLOG, está basado principalmente en la definición de hechos, relaciones y reglas. Esto realizado por medio de lógica de predicados, de lo que se puede definir:

1. Hechos monoádico. Son la representación más simple dentro del lenguaje de programación. Por medio de estos, se busca darle una característica a un objeto (sujeto). Son representados en la forma: felino(gato). En este caso, se puede ver que "felino" es el predicado del sujeto "gato", el predicado definido se puede leer como "gato es un felino".
2. Hecho poliádico. En este tipo de predicados, se busca enmarcar las relaciones que existen entre los distintos objetos, por ejemplo: es_un_animal(tiburón, acuático). En este caso, el predicado se puede leer como "tiburón es un animal acuático", y de esa manera se genera la relación que existe entre los objetos definidos.
3. Reglas. Están definidas como la interconexión de predicados verdaderos, para confirmar un hecho que debe ser también verdadero. La definición de reglas se da por medio de la estructura:

```
regla(objeto1, objeto2):- hecho1(objeto1, objeto_n), hecho2(objeto_n, objeto2).
```

Esta sentencia puede leerse en la forma la "regla" es verdadera si y sólo si el "hecho1" y el "hecho2" son verdaderos. En esta sentencia es importante hacer mención a la situación en la cual el "hecho1" y el "hecho2" están relacionadas entre sí por medio de "objeto_n".

Los predicados, o hechos, pueden tener una cantidad indefinida de objetos o argumentos, con lo cual podemos tener una representación general en la forma:

```
hecho(objeto_1, objeto_2, . . . . ., objeto_n).
```

Definiendo cada predicado mediante con objetos relacionados lógicamente entre sí.

Las reglas también pueden tener una cantidad de hechos, y de objetos indefinidos, la forma general de las reglas puede ser mostrada en la siguiente forma:

Regla(objeto_1, objeto_2, ..., objeto_i):- hecho_1(objeto_1, objeto_2, ..., objeto_n), hecho_2(objeto_1, objeto_2, ..., objeto_n), ..., hecho_m(objeto_1, objeto_2, ..., objeto_n).

Algunas características importantes de mencionar al momento de generar las reglas y los hechos son:

1. Ambos pueden ser declarados con una cantidad nula de objetos.
2. Si existen relaciones entre los hechos, estas deben de ser cumplidas exitosamente. No puede haber una relación entre hechos u objetos que no sea cumplida.
3. Para que la regla sea verdadera, todos los hechos deben ser verdaderos.
4. Cuando se tienen predicados simples, se considera que solamente se está llenando una base de conocimientos, base que puede ser utilizada posteriormente por programas más complejos.

Basado en lo descrito, se puede definir una base de conocimientos en la forma:

```
ave(loro).
ave(pato).
ave(ganzo).
ave(pingüino).
ave(gallo).
ave(perico).
```

Donde se ve que existen datos que pertenecen al campo ave.

También se puede definir una base de conocimiento donde se defina la relación “que tipo de animal es” de dos objetos, tal como :

```
es_tipo(perro, terrestre).
es_tipo(ganso, aéreo).
es_tipo(tiburón, acuático).
es_tipo(tigre, terrestre).
es_tipo(murciélago, aéreo).
es_tipo(delfín, acuático).
```

De la base, podemos leer que un perro es del tipo terrestre.

2.4. Interfaces humano máquina

Son interfaces que permiten la interacción entre un humano y una máquina, también son conocidas como interfaces hombre-máquina (IHM). Anteriormente estas interfaces podían ser poco prácticas debido a su tamaño, no naturales y difíciles de entender, sin embargo, en la actualidad se busca que estas interfaces sean naturales, de fácil uso, inteligentes y portables. En la norma ISO 9241-110, el término interfaz de usuario se define como "todas las partes de un sistema interactivo (software o hardware) que proporcionan la información y el control necesarios para que el usuario lleve a cabo una tarea con el sistema interactivo".

La interfaz del usuario, permite que el operador, en ciertas circunstancias, vaya más allá del manejo de la máquina o sistema, y observe el estado del equipo e intervenga en el proceso.

Dentro de los beneficios que se pueden obtener gracias a las interfaces, destaca lo siguiente:

- Permiten simplificar y optimizar los procesos.
- Proporcionan acceso instantáneo a los datos, pudiendo así mejorar la toma de decisiones.
- Contribuye a tener un mejor registro de los datos proporcionados.
- Mejora la productividad y eficiencia de los procesos.
- Mejoras en la gestión de los procesos.
- Se pueden adaptar a las necesidades de cada proceso.

2.5. LabView

De acuerdo a la página web de *National Instruments*, *LabView* es un software de ingeniería diseñado para aplicaciones que requieren pruebas, medidas y control con acceso rápido a información de datos y hardware. Esta herramienta ha ido tomando gran importancia en el mundo de la ingeniería, debido a su gran capacidad de integración en diferentes tipos de proyectos, y principalmente a la facilidad con la que se puede programar. Básicamente *LabView*, ha introducido el concepto de programación gráfica, ya que en lugar de utilizar comandos en un lenguaje de programación estándar, utiliza funciones representadas por iconos y bloques gráficos.

LabView ha sido parte importante de diferentes proyectos de investigación, desde sistemas de monitoreo ambiental [11], medición de sistemas eléctricos de potencia [12], monitoreo remoto de diferentes parámetros ambientales [13], hasta el análisis de señales utilizando *wavelets* [14].

2.6. Protocolos

Dentro del área de la informática, podemos definir un protocolo como un sistema de reglas el cual permite que dos o más entidades (componentes o sistemas) puedan comunicarse entre sí para transmitir información.

Existe una amplia variedad de sistemas o protocolos domóticos para realizar un hogar digital o instalación domótica. Cada sistema domótico es distinto, hay varios tipos y protocolos: cableados, inalámbricos, PLC, Wifi, Zigbee, Bluetooth, entre otros, aunque en esencia, todos ellos son parecidos, ya que buscan la misma finalidad, cambiando tipo de paquetes de envío o forma de arquitectura, como centralizada [10] o distribuida [9]. En este sentido, sin importar el protocolo, se busca que cada sistema se comunique con los distintos dispositivos conectados y les provea de instrucciones para realizar una función.

Capítulo 3

Desarrollo del Sistema

El sistema desarrollado es un prototipo de un sistema domótico el cual consta de dos sensores de iluminación (uno en el interior y otro en el exterior de la casa) y un sensor de temperatura sumergible. Para la parte de la iluminación, se monitoreo el nivel de luminosidad en el exterior como en el interior de la casa; auxiliados de unas persianas se compensa la luz interna con la externa al modificar la apertura y cierre de la persiana; cuando se llega al punto en el que en el exterior la iluminación es escasa, se activa el apagador del interior para que esa compensación de iluminación se realice con la luz artificial de un foco. En cuanto al sensor de temperatura sumergible, se instaló en la tubería de la regadera de un baño para que el usuario configure la temperatura deseada para bañarse y el tiempo que desea destinar para su aseo; esta parte del sistema esta orientado a un calentador de agua de paso, el cual regula su temperatura de acuerdo a la necesidad del usuario con el apoyo de un actuador que la mantiene constante durante el periodo de tiempo definido por el mismo. Posterior a este tiempo, el calentador regresa a su posición de “piloto” para el ahorro en el consumo de gas.

En este capítulo se especifica todo el equipo, material y procedimientos necesarios para el desarrollo del proyecto, sus especificaciones técnicas y la utilidad por la que fueron seleccionados, así como las conexiones utilizadas, la lógica de programación, la recolección y gestión de la información obtenida. En general, se explica cómo funciona y como se desarrolla el proyecto tanto a nivel técnico, como a nivel funcional o de producto final.

3.1. Sensores

Los sensores son todos aquellos dispositivos que tienen la capacidad de convertir magnitudes físicas y químicas en magnitudes eléctricas, esto es convertir un fenómeno del entorno que puede ser “medido” en una señal eléctrica que puede ser asociada a una cantidad escalar de dicho fenómeno, el cual puede ser reconocido e interpretado por una computadora o microcontrolador dándole un significado y sentido a esa señal eléctrica.

Un sensor tiene tres parámetros fundamentales, el rango (valores entre los que puede medir), la resolución (la variación mínima que puede detectar) y la sensibilidad (lo que varía la magnitud de salida en relación con la variación de la magnitud medida). Los sensores se clasifican atendiendo a varios criterios, como:

- Por la señal que emiten, existen analógicos y digitales (o discretos)
- Por la influencia que tienen sobre el proceso, se tienen sensores pasivos cuando no influyen, y activos cuando absorben energía
- Por los parámetros que pueden variar en el sensor pueden ser mecánicos, eléctricos, electromagnéticos, ópticos, etc.

3.1.1. Sensor de temperatura (DS18B20)

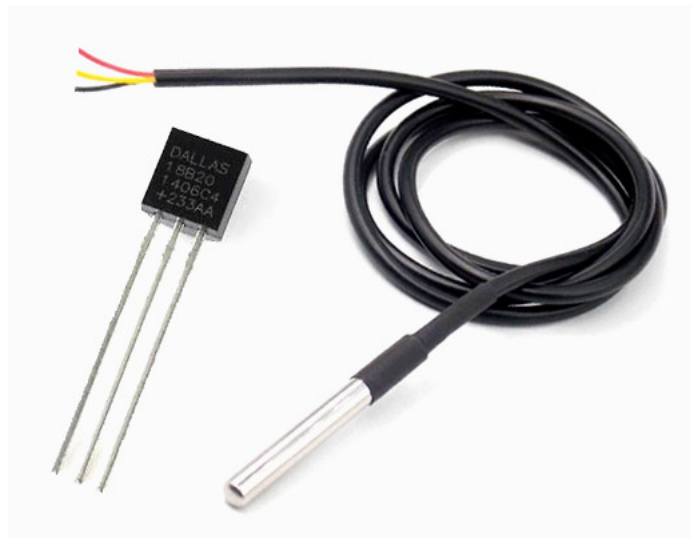


Figura 3.1: Sensor de temperatura sumergible DS18B20

El sensor DS18B20 (Figura 3.1) es un termómetro digital resistente al agua (puede ser sumergido) que proporciona lecturas de temperatura de 9 a 12 bits (configurables) que indican la temperatura del dispositivo.

La información se envía a/desde el DS18B20 a través de una interfaz llamada *1-Wire*, de modo que solo un cable (y la tierra) se debe conectar desde un microprocesador central a un DS18B20. La capacidad para leer, escribir y realizar conversiones de temperatura puede derivarse de la propia línea de datos sin necesidad de una fuente de alimentación externa.

Debido a que cada DS18B20 contiene un número de serie único de silicón, pueden existir múltiples DS18B20 en el mismo bus *1-Wire*. Esto permite colocar sensores de temperatura en muchos lugares diferentes. Las aplicaciones en las que esta función es

útil incluyen controles ambientales de HVAC [15] (sistema de ventilación, calefacción y aire acondicionado por sus siglas en inglés *Heating, Ventilating and Air Conditioning*), sensado de temperaturas dentro de edificios, equipos o maquinaria, y monitoreo y control de procesos.

Características

- La interfaz única *1-Wire* requiere solo un pin de puerto para la comunicación.
- La capacidad multipunto simplifica las aplicaciones de detección de temperatura distribuida.
- No requiere componentes externos.
- Puede ser alimentado desde la línea de datos. El rango de alimentación es de 3.0V a 5.5V.
- Mide temperaturas desde - 55 °C hasta + 125 °C.
- Precisión de $\pm 0.5^\circ\text{C}$ desde - 10 °C hasta + 85 °C.
- La resolución del termómetro es programable de 9 a 12 bits.
- Convierte la temperatura de 12 bits en palabra digital en 750 ms (máx.).

Bus de comunicación *1-Wire*

1-Wire es un protocolo de comunicaciones en serie diseñado por *Dallas Semiconductor*. Está basado en un bus, un maestro y varios esclavos de una sola línea de datos en la que se alimentan. Por supuesto, necesita una referencia a tierra común a todos los dispositivos.

La línea de datos/alimentación requiere una resistencia en *pull-up* conectada a la alimentación y que le proporciona ésta.

El DS18B20 se comporta como un esclavo. La discusión de este sistema de bus se divide en tres temas: configuración de hardware, secuencia de transacciones y señalización *1-Wire* (tipos de señales y temporización).

3.1.2. Sensor de luz (BH1750)

El módulo BH1750 (Figura 3.2) es un sensor de iluminación digital para medición de flujo luminoso (iluminancia) de la empresa *Rohm Semiconductor*. Posee un conversor interno de 16-bit, por lo que entrega una salida digital en formato I2C. Su desempeño es mejor al de un Foto-Resistor (LDR), pues no es necesario realizar

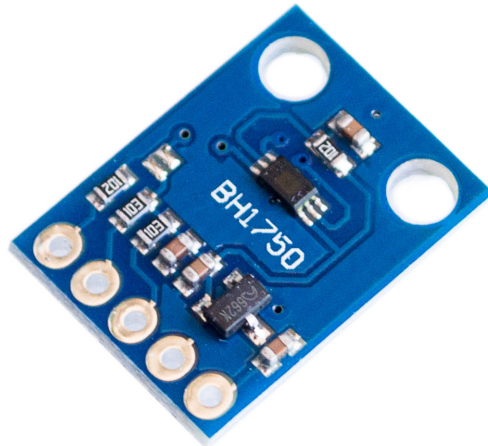


Figura 3.2: Sensor de luz BH1750

conversiones de voltaje para obtener datos interpretables.

El BH1750 entrega la intensidad luminosa directamente en unidades Lux (Lx). El Lux es la unidad derivada del S.I. de Unidades para la iluminancia o nivel de iluminación. Equivale a un lumen/m². Se usa en la fotometría como medida de la luminancia, tomando en cuenta las diferentes longitudes de onda según la función de luminosidad, un modelo estándar de la sensibilidad a la luz del ojo humano.

Características

- Voltaje de Operación: 3V – 5V.
- Interfaz digital a través del bus I2C con capacidad de seleccionar entre 2 direcciones.
- Respuesta espectral similar a la del ojo humano.
- Realiza mediciones de iluminancia y convierte el resultado a una palabra digital.
- Amplio rango de medición (1-65535 lux).
- Modo de bajo consumo de energía.
- Rechazo de ruido a 50/60 Hz.
- Baja dependencia de la medición contra la fuente de luz: halógeno, led, incandescente, luz de día, etc.

Bus de comunicación I2C

I2C es un puerto y protocolo de comunicación serial, define la trama de datos y las conexiones físicas para transferir bits entre 2 dispositivos digitales. El puerto incluye dos cables de comunicación, SDA y SCL. Además el protocolo permite conectar hasta 127 dispositivos esclavos con esas dos líneas, con velocidades de hasta 100, 400 y 1000 kbits/s. También es conocido como IIC o TWI (*Two Wire Interface*).

La conexión de tantos dispositivos al mismo bus, es una de las principales ventajas. Además, si se compara a I2C con otro protocolo serial, este incluye más bits en su trama de comunicación lo que permite enviar mensajes más completos y detallados.

Los mensajes que se envían mediante un puerto I2C, incluye además del byte de información, una dirección tanto del registro como del sensor. Para la información que se envía siempre existe una confirmación de recepción por parte del dispositivo. Por esta razón es bueno diferenciar a los distintos elementos involucrados en este tipo de comunicación. En el caso de I2C se diferencian dos elementos básicos, un MAESTRO y un ESCLAVO.

El MAESTRO se encarga de controlar al cable de reloj, por sus siglas en inglés llamada SCL (*Serial CLock*). Además el MAESTRO se encarga de iniciar y parar la comunicación. La información binaria serial se envía sólo por la línea o cable de datos seriales, en inglés se llama SDA (*Serial DAta*). Dos MAESTROS no pueden hacer uso de un mismo puerto I2C. Puede funcionar de dos maneras, como maestro-transmisor o maestro-receptor.

El ESCLAVO, generalmente suele ser un sensor. Este elemento suministra la información de interés al MAESTRO. Puede actuar de dos formas: esclavo-transmisor o esclavo-receptor. Un dispositivo I2C esclavo, no puede generar a la señal SCL.

3.2. Actuadores

Dentro de los sistemas de automatización, los actuadores se definen como todos aquellos componentes capaces de transformar energía hidráulica, neumática o eléctrica en la activación de un proceso con la finalidad de generar un efecto en el sistema o, dicho de otra manera, ejecutan un efecto o acción basados en una señal o energía de entrada.

3.2.1. Motor a pasos

Un motor a pasos es un motor de corriente directa (CD) sin escobillas que puede ser de imán permanente o reluctancia variable, el cual tiene como características de desempeño rotar en ambas direcciones, moverse con incrementos angulares precisos,

sostener un torque de retención a velocidad cero y controlarse mediante circuitos digitales.

El número y tasa de los pulsos controla la posición y velocidad del eje del motor. Por lo general, los motores de pasos se fabrican con pasos por revolución de 12, 24, 72, 144, 180 y 200, lo que resulta en incrementos de eje de 30° , 15° , 2.5° , 2° y 1.8° por paso. A medida que se activan los bobinados del motor en un orden en particular, fluye una corriente a través de ellos que magnetiza el estator provocando polos electromagnéticos que causan la propulsión del motor.

3.2.2. Motor 28BYJ-48 con módulo controlador ULN2003

El motor 28BYJ-48 (Figura 3.3) tiene un ángulo de paso de 5.625° con un reductor interno que tiene una relación de $1/64$. Combinados, la precisión total es $64 \times 64 = 4096$ pasos por vuelta, lo cual equivale a un paso de 0.088° , que resulta en una precisión muy elevada. Esto quiere decir que por cada paso que da el motor, debido al reductor, el eje gira 0.088° .



Figura 3.3: Motor 28BYJ-48 con módulo controlador ULN2003.

El módulo ULN2003 consta de una matriz de transistores de alta tensión y corriente que cuenta con salidas de alto voltaje para la conmutación de cargas inductivas, lo que permite controlar de manera exacta el giro del motor por medio de impulsos eléctricos que, a su vez, son controlados a partir de una tarjeta lógica como puede ser Arduino.

El motor cuenta con 4 bobinados, los cuales trabajan a una tensión de 5V, que dependiendo de la secuencia específica de activación hacen girar el eje en una configuración de paso completo o medio paso (Figura 3.4), asimismo, depende del sentido en que se activen dichos bobinados cual será la dirección de giro del eje (al sentido de las manecillas del reloj o en contra del sentido de las manecillas del reloj).

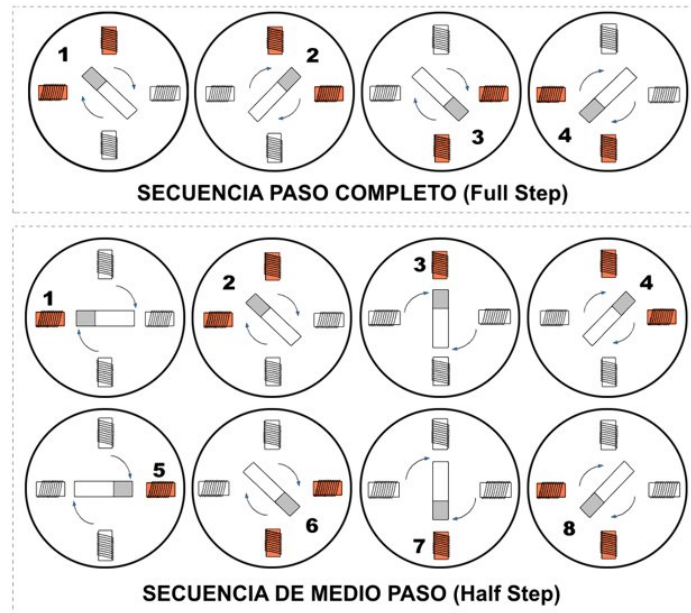


Figura 3.4: Esquema de activación de bobinas.

3.3. Conexiones electrónicas

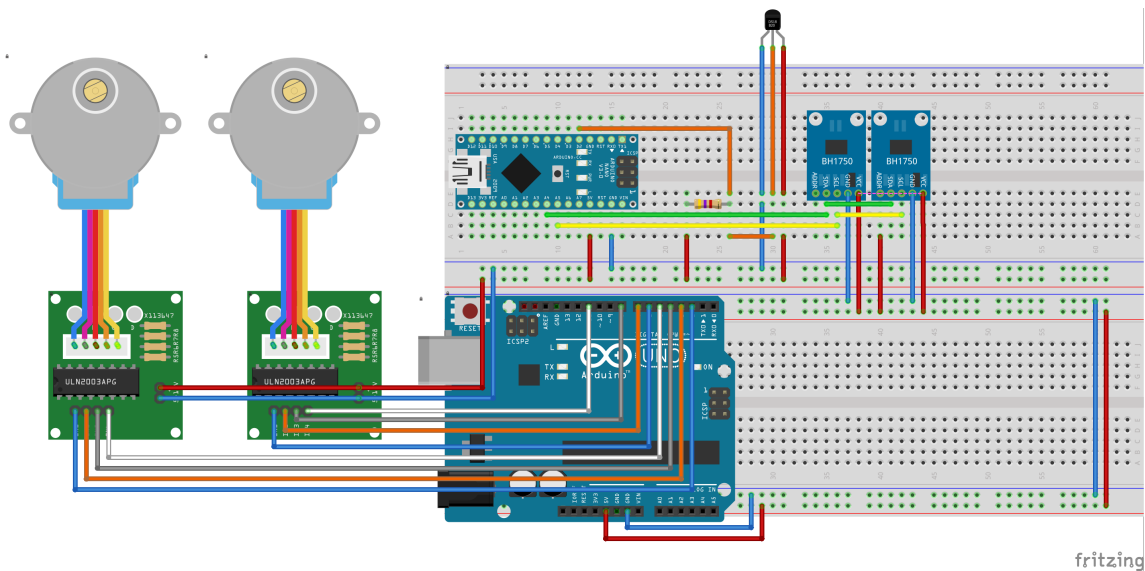


Figura 3.5: Esquemático de conexiones electrónicas.

Para el desarrollo de este proyecto y con ayuda de los sensores antes mencionados, es requerido el uso de placas de desarrollo Arduino para gestionar la toma de datos del medio que fue controlado. Dadas las condiciones y características requeridas en este proyecto, se utilizaron dos placas de desarrollo (Arduino UNO y Arduino NANO), esto debido a la limitante de pines de conexión disponibles en una sola de las placas, además del bajo costo que éstas representan. Todas las conexiones fueron realizadas sobre protoboards (Figura 3.5), para verificar la funcionalidad de cada uno de los elementos integrados.

Las conexiones correspondientes a los sensores se gestionan por medio de la placa de Arduino NANO. La conexión de los sensores de iluminación (BH1750) se lleva a cabo conectado a los pines analógicos 4 y 5 (A4 y A5) los cables de conexión correspondientes a la comunicación I2C (SCL y SDA respectivamente), siendo éstos alimentados por la misma placa con un voltaje de cinco volts (5V). Para el caso del sensor de temperatura (DS18B20) es necesario alimentarlo de igual manera con un potencial de 5V hacia el pin digital de entrada 2 (D2); haciendo esto en una configuración de *pull up*, lo que consiste en conectar una resistencia fija (4700 Ohms para este sensor en particular) entre la alimentación (En este caso 5V) y el pin de entrada del sensor (D2); esto para asegurar un valor de lectura lógico alto (*HIGH*) eliminando cualquier posibilidad de indeterminación en su estado eléctrico.

Para las conexiones que corresponden a los actuadores, que para este proyecto son dos motores a pasos de cuatro polos (28BYJ-48), se conectaron a la placa de Arduino UNO, con la ayuda para su control de módulos ULN2003APG los cuales permiten controlar de manera directa los motores ya que proporciona una corriente mayor de salida de la que nos pueden suministrar los pines digitales de salida; siendo éstos alimentados por la misma placa con un voltaje de 5V, y controlados por 4 pines digitales de salida (D2, D3, D4 y D5 para el primer motor y D6, D7, D8 y D11 para el segundo motor) para cada uno de los motores.

Para el caso de ambas placas de desarrollo Arduino (NANO y UNO), se mantienen conectadas a la computadora en la cual se registran todos los datos obtenidos y donde se gestiona toda la interacción del sistema, por medio de los respectivos cables USB de manera independiente cada uno de ellos.

3.4. Lógica de programación (Programación inteligente)

Para comenzar con este tema es necesario definir que la lógica es la base de todo el conocimiento en programación, ya que desarrollando ésta y conociendo las palabras reservadas es posible aprender a escribir un código para que la computadora lo interprete correctamente. Es decir que se puede aprender a comunicarse con la máquina a partir de un lenguaje. La lógica es la técnica utilizada para desarrollar instrucciones

en una secuencia para lograr determinado objetivo o, dicho en otras palabras, es la organización y planificación de instrucciones en forma de un algoritmo, con el objetivo de plasmar de una manera visible o tangible la implementación de un programa o software.

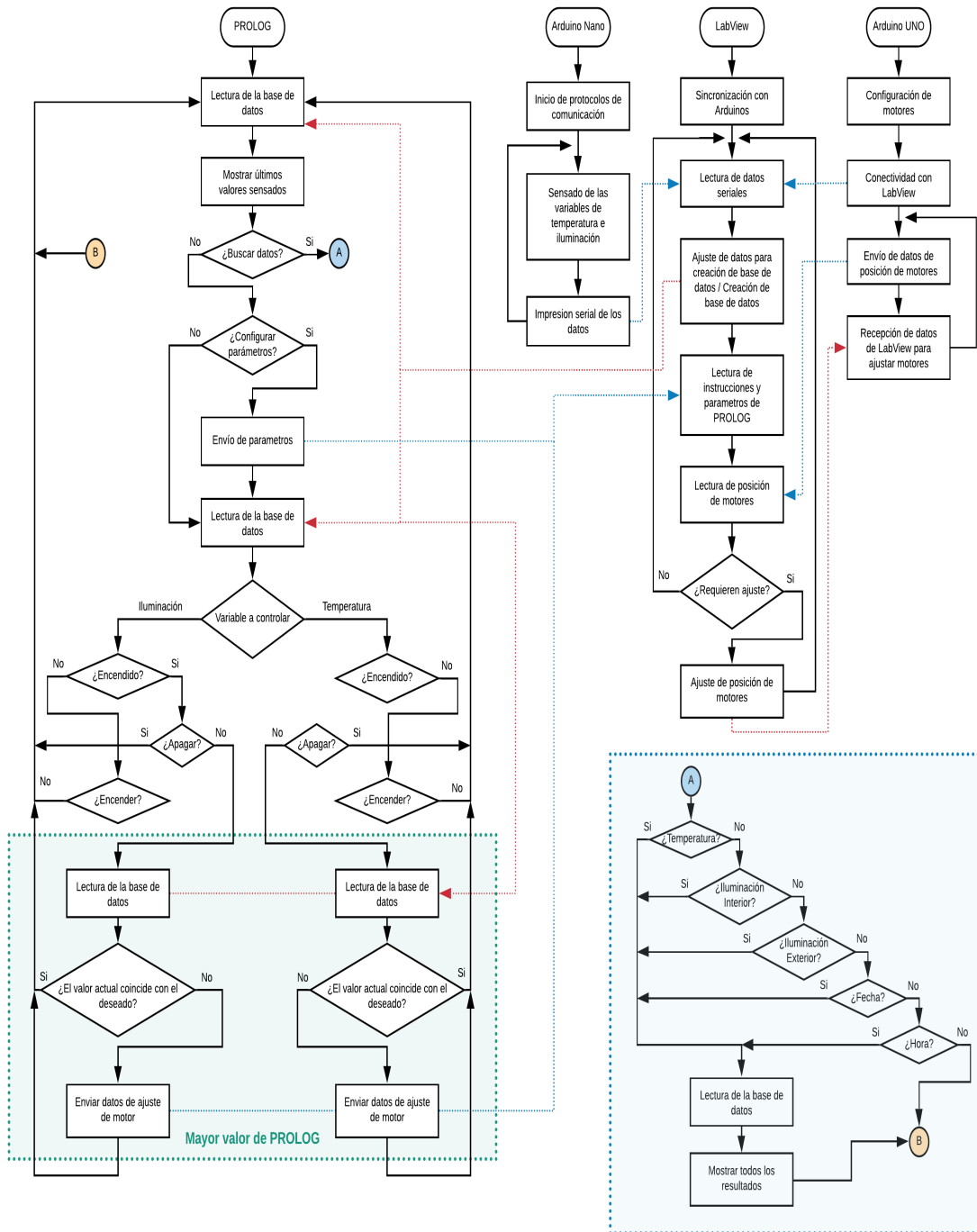


Figura 3.6: Diagrama de Flujo del sistema

Una vez que se ha definido cuál es el propósito del proyecto y cuáles son las variables a considerar tanto de entrada (sensores, datos, requerimientos del usuario, etc.) como de salida (actuadores, pantallas de información, etc.), es necesario identificar el flujo de los procesos e interacción a desarrollar en un posterior código de programación. Para fines de este proyecto, un diagrama de flujo es utilizado para visualizar de manera general la interacción del software con el hardware como se muestra en la Figura 3.6.

Tal como se comentó anteriormente, una de las partes fundamentales en la programación de este proyecto corresponde a la implementación de los sensores como método de lectura del entorno que se estudió. Esto se logró implementando los respectivos protocolos de comunicación que corresponden a cada sensor (*1-Wire* e *I2C* respectivamente) lo cual permite leer e interpretar la información que se está tomando del medio (temperatura e iluminación), siendo estos leídos por la placa de desarrollo Arduino NANO, para su posterior interpretación dentro del programa generado en *LabView*. Por otra parte, únicamente se configura la placa de desarrollo Arduino UNO para que interactúe con *LabView* recibiendo instrucciones para que de esta manera se pueda controlar desde el mismo programa el movimiento de los motores.

Dentro de *LabView* se genera un programa (Figura 3.7) en el cual se realiza la interacción de todos los sistemas; desde la generación de la base de datos (Figura 3.8), creando un archivo de texto (.txt) con el formato correspondiente para que pueda ser leído directamente desde *PROLOG*, incluyendo en él las mediciones de cada uno de los sensores, así como la asignación de una fecha y una hora de la obtención de cada dato para tener un mejor registro de lo que se está sensando; hasta las instrucciones de movimiento de los motores de manera independiente. Esto basado en un sistema de aproximaciones en el caso de la temperatura y llegar al punto exacto en el caso de la iluminación. Para esto; y debido a que la temperatura es una variable un poco complicada de manejar (ya que no se puede regular de manera exacta, comparada con otras variables como puede ser la distancia o intensidad de voltaje, por ejemplo) se hace una aproximación en la cual mientras más próximo se este a la temperatura programada, se modifica gradualmente la calefacción del agua, esto para que no se rebase excesivamente la temperatura, o bien, se llegue exactamente a lo deseado. En el caso de la iluminación es un proceso similar, solo que, en este caso, es una variable más fácil de controlar, por lo que, si falta iluminación, se gira en una dirección; si es demasiada iluminación, se gira en sentido opuesto.

Cabe mencionar que una parte fundamental de esta programación recae en conocer las características del sistema, en este caso se habla de la perilla del calentador de agua y el mecanismo de apertura y cierre de las persianas. Esto queda plasmado en grados para el sistema de giro de los motores, quedando éstos como los parámetros límites de giro de cada uno de los motores. Gracias a esta información y para complementar y verificar el correcto funcionamiento del sistema, con el apoyo de una interfaz visual de *LabView* (Figura 3.9) se corroboró que los datos se adquieran de manera adecuada, además de mostrar la posición en grados de cada uno de los motores y

si cada uno de los sistemas (Calefacción e iluminación) se encuentran encendidos o apagados.

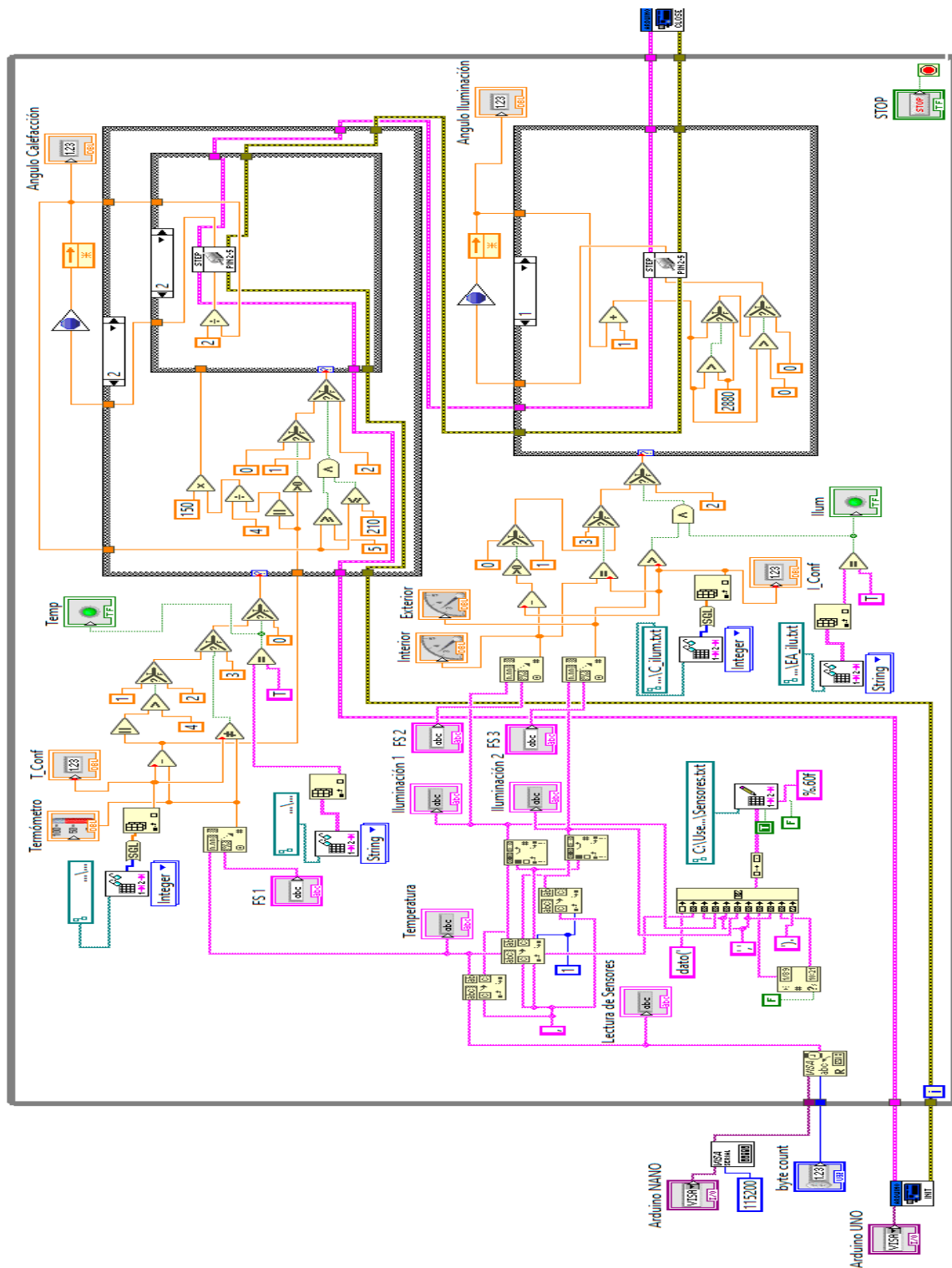


Figura 3.7: Diagrama de bloques del programa en *LabView*

```

Archivo Edición Formato Ver Ayuda
dato('24.50','160','0','25/03/2020','03:52 p. m.').
dato('24.50','160','0','25/03/2020','03:52 p. m.').
dato('24.50','160','0','25/03/2020','03:52 p. m.').
dato('24.50','160','0','25/03/2020','03:52 p. m.').
dato('24.50','160','0','25/03/2020','03:52 p. m.').
dato('24.50','160','0','25/03/2020','03:52 p. m.').
dato('24.50','160','0','25/03/2020','03:52 p. m.').
dato('24.50','156','0','25/03/2020','03:52 p. m.').
dato('24.50','156','0','25/03/2020','03:52 p. m.').
dato('24.50','160','0','25/03/2020','03:52 p. m.').
dato('24.50','170','0','25/03/2020','03:52 p. m.').
dato('24.50','199','163','25/03/2020','03:52 p. m.').
dato('24.50','209','216','25/03/2020','03:52 p. m.').
dato('24.50','213','223','25/03/2020','03:52 p. m.').
dato('24.50','216','223','25/03/2020','03:52 p. m.').
dato('24.50','216','223','25/03/2020','03:52 p. m.').
dato('24.50','219','223','25/03/2020','03:52 p. m.').
dato('24.50','216','223','25/03/2020','03:52 p. m.').
dato('24.50','223','226','25/03/2020','03:52 p. m.').
dato('24.50','223','229','25/03/2020','03:52 p. m.').
dato('24.50','229','229','25/03/2020','03:52 p. m.').
dato('24.50','229','229','25/03/2020','03:52 p. m.').
dato('24.50','236','229','25/03/2020','03:52 p. m.').

```

Figura 3.8: Base de datos con formato para *PROLOG*Figura 3.9: Interfaz visual de *LabView*

Para la parte de la programación que corresponde a *PROLOG*, se desarrolló un sistema de ventanas emergentes que interactúan con el usuario, mostrando un menú principal (Figura 3.10) en el cual el usuario puede ver cuáles son los valores de los últimos datos registrados justo en el momento en el que se ingresó al menú. De igual manera permitiéndole al usuario seleccionar diferentes opciones para realizar (como son 'Configurar parámetro', 'Buscar información' y 'Encender/Apagar') y así interactuar con el sistema de manera inmediata.

Para la configuración de parámetros (Figura 3.11) se abre otra ventana en la que proporciona la posibilidad de configurar cada uno de los parámetros de manera independiente (Figura 3.12) para así dar la instrucción directamente al sistema de cuál es la temperatura que se desea para el agua y la iluminación que se desea para la habitación.

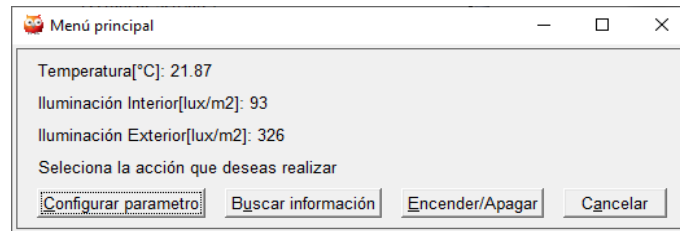


Figura 3.10: Menú principal

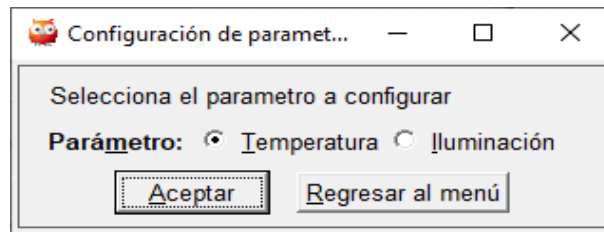
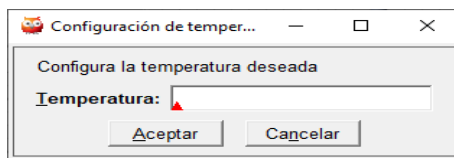
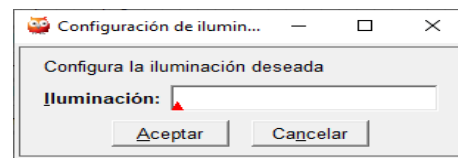


Figura 3.11: Configuración de parámetros



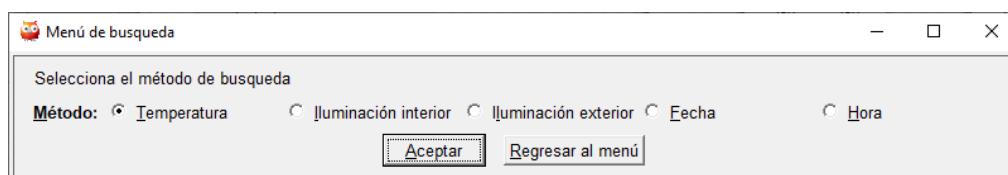
(a) Temperatura



(b) Iluminación

Figura 3.12: Configuración de parámetros

Dentro del menú de búsqueda de información (Figura 3.13) se muestran los diferentes métodos de búsqueda disponibles para encontrar la información registrada dentro de la base de datos generada por los sensores, lo que permite tener un panorama del historial de datos registrados a lo largo del uso del sistema, así como verificar coincidencias entre cada uno de los distintos parámetros de búsqueda.

Figura 3.13: 'Menú de búsqueda' en *PROLOG*

En cada una de las ventanas que corresponden a la búsqueda de cada uno de los parámetros (Figura 3.14) se cuenta con un ejemplo del formato indicado para realizar la búsqueda, esto es para que la búsqueda se realice de manera efectiva y evitar

errores de sintaxis que se puedan dar en el programa.

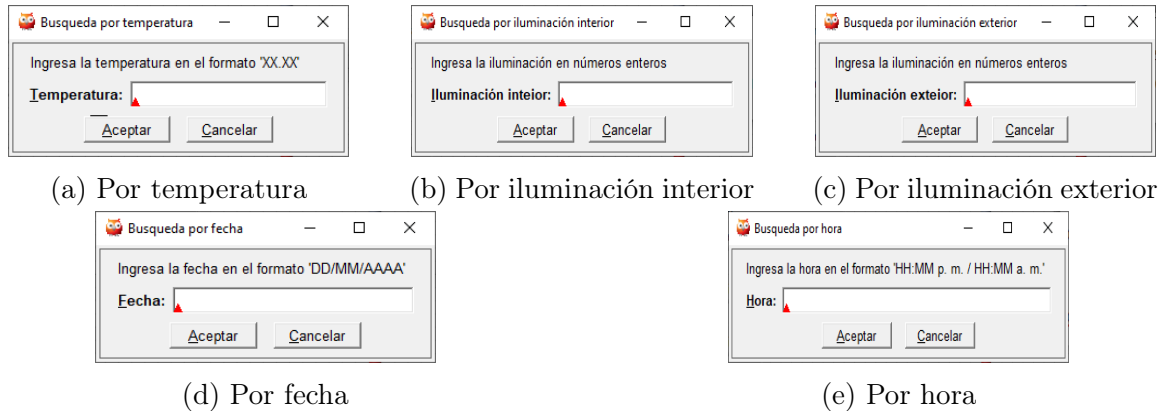


Figura 3.14: Búsqueda de información

Los resultados mostrados de acuerdo al tipo de búsqueda (mostrará todos los demás valores excepto el del parámetro ingresado) son desplegados en una ventana emergente, mostrando en ella cada uno de los parámetros adicionales y los valores que coinciden con la búsqueda realizada, por ejemplo, se busca una temperatura específica en la base de datos (Figura 3.15) y los resultados mostrados son todos los valores de iluminación interior, iluminación exterior, fecha y hora que correspondan con la temperatura que se busca.

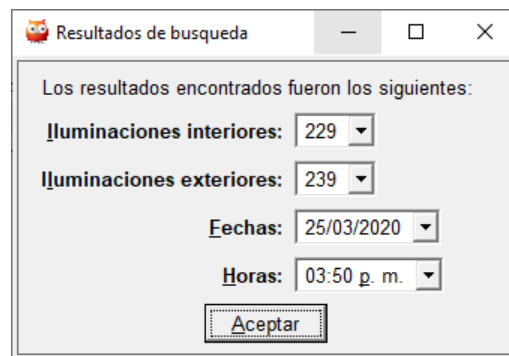
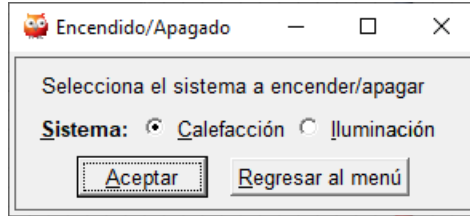
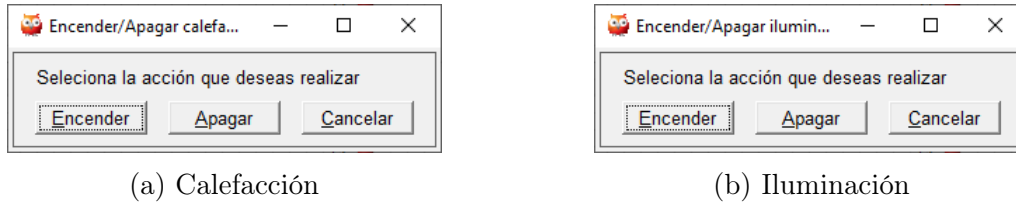


Figura 3.15: 'Resultados de búsqueda' en *PROLOG*

Por último, el menú de 'Encender/Apagar' (Figura 3.16) permite activar o desactivar cada uno de los sistemas (calefacción e iluminación), permitiendo que estos interactúen, ya sea de manera independiente o simultáneamente. Esto se logra ya que cada una de las opciones permite encender o apagar cada sistema individualmente (Figura 3.17) y se mantiene en ese estado hasta que el usuario seleccione la opción contraria.

Figura 3.16: Menú 'Encendido/Apagado' en *PROLOG*

(a) Calefacción

(b) Iluminación

Figura 3.17: Sistema de Encendido/Apagado

Cada una de las ventanas que se muestran en *PROLOG* es programada de manera individual, considerando cada una de ellas como una función dentro del código, interactuando siempre entre ellas enviando y recibiendo información para poder funcionar como un solo programa. Es importante destacar que dentro de la programación, cada una de las ventanas subsiguientes al menú principal, tienen la opción de regresar al inicio que es el mismo menú, esto, permite hacer una 'actualización' de los valores mostrados en ese menú, ya que como se comentó anteriormente, dicho menú muestra los valores sensados en el momento exacto en el que se está ingresando a esa ventana, lo cual permite tener un monitoreo en tiempo real de los valores y parámetros que se encuentran interactuando en el medio.

3.5. Recolección de datos

La interacción de los sensores con el entorno, como se muestra en los diagramas anteriores, se lleva a cabo mediante el uso de placas Arduino. Como se mencionó al inicio de este capítulo, cada uno de los sensores maneja un protocolo de comunicación diferente (*1-Wire* e *I2C*). Para generar un entorno de desarrollo y gestión de información más simple se realizó la conexión entre *LabView* y Arduino, lo cual facilitó el desarrollo y la programación del sistema. Debido a estos protocolos de comunicación que manejan los sensores, se configuró la plataforma de *LabView* para poder obtener de manera directa los datos adquiridos de los sensores.

Actualmente existe una librería llamada LIFA (*LabView Interface for Arduino*) que se carga directamente a la placa Arduino por medio de su IDE (entorno de desarrollo integrado o entorno de desarrollo interactivo, en inglés *Integrated Development Environment*) la cual permite la conexión directa entre Arduino y *LabView*. Esta misma fue desarrollada por los desarrolladores de *National Instruments* (encargados del

desarrollo del software *LabView*), en el cual se integraron funciones de programación y conexión de algunos periféricos de mayor uso en el desarrollo de Arduino, así mismo el protocolo de comunicación I2C. El problema surge a partir del protocolo *1-Wire*, el cual no se encuentra considerado en esta librería lo que no permite una conexión directa para obtener los datos generados a partir de sensores que trabajan bajo ese protocolo.

Investigando en distintas fuentes, y dado que el protocolo *1-Wire* es relativamente nuevo (y la librería para Arduino que conecta con *LabView* tiene mas tiempo que se desarrolló), no existe un método el cual permita hacer directamente esta configuración e incluir el protocolo para uso directo; por lo mismo se opto por utilizar una segunda placa Arduino para obtener desde su IDE la información de dicho sensor y conectarlo así con la interfaz de *LabView*, esto mediante una comunicación serial dada por VISA (arquitectura de software de instrumentos virtuales, en inglés *Virtual Instrument Software Architecture*) si esta definida en *LabView* y por la cual se puede ingresar cualquier tipo de entrada al sistema desde un entorno virtual.

¿Por qué no conectar desde un inicio todos los sensores a una sola placa de Arduino y comunicarlo mediante VISA? La respuesta a esto es sencilla, al realizar una conexión mediante VISA en *LabView* únicamente se esta definiendo esa información como una entrada, lo cual representa que el flujo de información ocurre en un único sentido, lo cual se vuelve como solo la lectura de información sin una interacción de regreso; al realizar la conexión mediante la librería de *National Instruments*, se genera un flujo de información en dos sentidos, se puede leer la información que entra desde Arduino y a su vez ingresar información al mismo para así poder reaccionar de acuerdo a lo programado. Esto es el principio fundamental y esencial para la parte de la automatización, ya que de esta manera se interactúa directamente con el entorno y se genera una reacción o respuesta que es lo que se busca para poder automatizar el sistema.

De acuerdo a lo anterior, se configuran los protocolos en la interfaz de *LabView* para así adquirir los datos necesarios, de la placa principal (Arduino UNO), se recolectaron los datos de los sensores de iluminación, los cuales, de acuerdo a nuestra ficha técnica del sensor, se diferenciaron por la dirección de cada uno en el bus de comunicación I2C, diferenciando así el sensor que se encuentra en el exterior y el sensor que se encuentra en el interior de la casa. Por otro lado, de la placa auxiliar (Arduino Nano), se obtuvieron los datos referentes al sensor de temperatura.

Es importante mencionar que al realizar la configuración y ajuste de los datos de los sensores en la interfaz de *LabView*, se realizó una conexión directa con el calendario y reloj interno de la computadora, esto es para darle un mejor formato y diferenciación a la recolección de datos, ya que de esta manera a cada dato sensado se le asigno una fecha de recolección, así como también una hora especifica en la cual fue adquirido; esto es importante para generar una base de datos más detallada y que pueda ser más fácil de entender y gestionar.

3.6. Gestión de datos

La gestión de los datos que se generan a partir de los sensores es la base del proyecto, ya que es lo que gestiona *PROLOG* y a partir de ello es que se hace la interacción con el entorno.

Es importante partir del hecho que se requiere un formato específico en la información generada para que *PROLOG* pueda interpretarla, y a su vez, pueda diferenciarla del sensor y fenómeno físico que se está tratando. Esto se hace mediante *LabView*, el cual permite generar un archivo de texto con la información registrada de los sensores en la cual a cada dato que es generado le va a atribuir una etiqueta de “sensor” (lo cual ayuda a determinar de qué sensor se está adquiriendo la información); de igual manera se sincroniza con el reloj y calendario de la computadora para que cada dato tenga la hora en la que se registró y por último el valor numérico de la magnitud sensada.

Esto permite tener un control sobre cada valor que se está obteniendo y distinguir de todas las entradas que se le puedan agregar al sistema (en este caso solo 3 sensores de entrada).

3.7. Interacción total de software y hardware

Como se mencionó anteriormente, para generar una interacción entre los sensores, actuadores y el software se generó una base de conocimiento (Figura 3.18), la cual es monitoreada en tiempo real por medio de *PROLOG*, quien se encarga de interpretar todos los datos adquiridos de acuerdo a su programación.

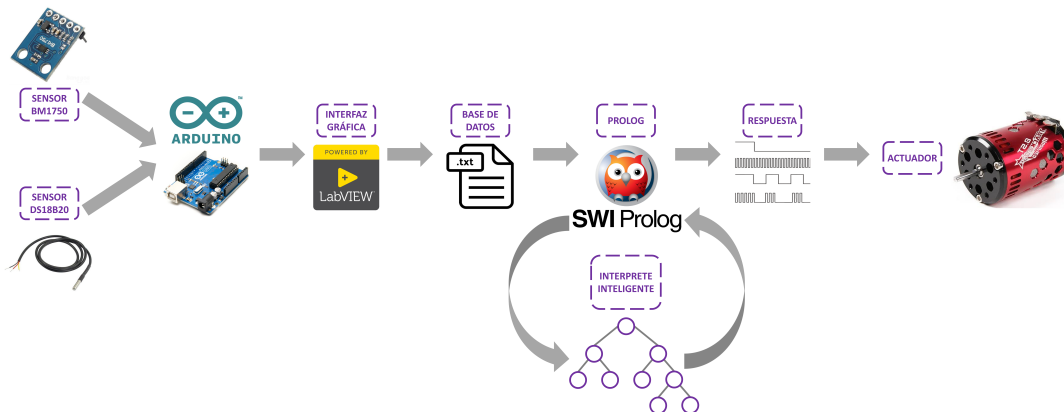


Figura 3.18: Interacción total del sistema.

Los sensores se encuentran conectados a la placa de desarrollo Arduino, la cual se encarga de interpretar las señales y generar un valor numérico que representa la magnitud medida. Para interpretar en la computadora estos valores, se conecta Arduino con *LabView*, el cual se encarga de darle formato a esta información para así distinguir cada dato registrado correspondiente a cada uno de los sensores.

Posteriormente, *LabView* genera un archivo tipo “.txt” en el cual se quedan registrados los datos y va acumulando los datos que se siguen generando mientras el dispositivo se encuentre conectado y encendido. Una vez generado el archivo de texto (la base de conocimiento), *PROLOG* se encarga de revisar continuamente el archivo, por lo que se puede ir leyendo cada dato que se genere en tiempo real, justo en el momento en el que se generó.

PROLOG permite hacer una selección y configuración de los datos que está interpretando, para que, de esta manera, se pueda realizar la interacción con los actuadores y asimismo configurar los parámetros ideales del sistema; en el caso del sensor de temperatura, permite programar la temperatura adecuada a la que se requiere calentar el agua, y para la iluminación, se programa la cantidad luminosa buscada para mantenerla dentro de la habitación. Como se mencionó anteriormente, esto se logra a partir de un sistema de ventanas emergentes que va indicando, dependiendo de la acción que se quiera realizar, como configurar, buscar, encender o apagar los sistemas, ya sea que éstos trabajen simultáneamente o independientemente.

Después de realizar la programación de los parámetros, se puede realizar una búsqueda de datos registrados en toda la base de conocimiento y, a su vez, “encender” o “apagar” el sistema automático que reacciona de acuerdo a los datos que son sentidos. Esto se logra generando otro archivo “.txt” a partir de *PROLOG* a manera de señal, la cual regresa a *LabView* y es interpretada a su vez por Arduino y genera la interacción con nuestros actuadores (motores a pasos).

Para entender el proceso inteligente que realiza *PROLOG*, es necesario mencionar que *PROLOG* va relacionando los datos de forma lógica (hay que distinguir entre lógica de programación, que cualquier lenguaje de programación utiliza; y lógica clásica de interpretación), utilizando la llamada lógica clásica o lógica de predicados, es decir, relaciona los datos que se encuentran dentro de su base de conocimientos para encontrar una solución verdadera a un problema; un ejemplo muy simple de esto es si los perros ladran y firulais es un perro, entonces firulais ladra. Este predicado indica que para que el resultado “firulais ladra” los dos predicados “los perros ladran” y “firulais es un perro” deben de ser verdaderos. Este ejemplo tan sencillo, muestra la relación que existe entre los predicados para llegar a un resultado lógico.

Este proceso es precisamente el que *PROLOG* ejecuta internamente para relacionar los datos que se encuentran dentro de su base de conocimiento y determinar los datos correctos para que la consulta sea verdadera.

Capítulo 4

Análisis de resultados

Los resultados que se obtuvieron en base al desarrollo de este proyecto, han cumplido con todos los objetivos planteados desde el inicio. Se cumplió adecuadamente y se logró desarrollar el sistema propuesto por lo que esto nos sugiere una apertura en el panorama de aplicación de este lenguaje de programación (*PROLOG*).

4.1. Tiempos de respuesta

Para poder determinar los tiempos de respuesta del sistema, es necesario implementar algunas instrucciones adicionales dentro de la programación del mismo, esto para determinar tiempos exactos que le toma al sistema pasar de una instrucción a otra, por lo que se parte de los diferentes pasos transcurridos a lo largo del desarrollo de este sistema, siendo estos: la lectura de los sensores, la escritura de la base de datos, el número de pasos de motor en un tiempo determinado, la búsqueda de información en *PROLOG* y en general la interacción total del sistema.

Para la toma de datos de los sensores, se programo un tiempo de respuesta de 20 milisegundos entre cada dato desde la IDE de Arduinio, esto le permite al sistema no sobrecargarse y asimismo realizar un muestreo continuo en el transcurso del funcionamiento del sistema.

Durante la escritura o generación de la base de datos, se genera un tiempo de respuesta similar al de la lectura de datos, corroborando que se está realizando en tiempo real, siendo este tiempo de aproximadamente 30 milisegundos, quedando así registrados en la base de datos aproximadamente 2,045 datos por cada minuto.

El giro de los motores esta a su vez sincronizado con la velocidad de escritura de la base de datos, esto debido a que cada que se genera una escritura en la base de datos se da un paso en cada motor, lo cual se determinó en base a las condiciones de programación y características de los motores, quedando estos en una configuración de paso completo lo que se traduce en 1 paso de motor cada 30 milisegundo, que a su vez, esto representa un tiempo de 0.34 segundos aproximadamente por cada grado de giro.

La parte principal del proyecto, que es la interacción con el entorno de *PROLOG*, cuenta con tiempos de respuesta similares, la apertura del programa, el desplazamiento entre las ventanas de los diferentes menús, la configuración de parámetros (escritura de archivo '.txt') y el encendido y apagado de los sistemas (igualmente escritura de archivo '.txt') ronda el rango de los 20 a 30 milisegundos de respuesta entre cada uno de ellos. La parte más lenta durante la interacción es la búsqueda de información, esto es debido al tamaño de la base de datos y es subjetivo ya que depende de la capacidad del equipo de cómputo (velocidad del procesador, memoria RAM, etc.); ya que esto tomó aproximadamente 22 segundos para mostrar todos los resultados de la búsqueda.

4.2. Ventajas contra otros sistemas

Para poder determinar cuáles son las ventajas o desventajas que representa este proyecto respecto a otros sistemas, se realizó una tabla comparativa (Tabla 4.1) respecto a otros esquemas similares. Esto tomando en cuenta que los sistemas comparados son proyectos realizados con la finalidad de reducir costos, implementar o explotar algún protocolo de comunicación, innovar o aplicar nuevos conocimientos desarrollando los proyectos desde nada hasta obtener un resultado; tal como se hizo con este proyecto.

Cabe mencionar que los parámetros seleccionados para comparar tienen el propósito de resaltar las cualidades antes mencionadas, siendo estos: tipo de interfaz, tipo de programación, tipo de comunicación y tipo de sistema.

El tipo de interfaz se refiere al tipo de *software* implementado o implementados para el desarrollo del sistema, desde la parte de programación de dispositivos hasta la interfaz con el usuario. El tipo de programación se basa en la lógica de programación implementada para el funcionamiento de los dispositivos y las respuestas que estos generan para interactuar con el medio. En cuanto al tipo de comunicación se refiere al protocolo o sistema de comunicación entre los dispositivos, ya sea de manera alámbrica o inalámbrica y que tipo de tecnología se está usando para llevar a cabo dicha comunicación. Por último, el tipo de sistema se refiere a la organización del sistema, ya sea modular o centralizado; esto quiere decir que si el sistema es modular, cada uno de los componentes o subsistemas (calefacción, iluminación, seguridad, etc.) se desenvuelven de manera independiente aunque existe una relación entre todos ellos, esto beneficia el hecho de que si algún modulo llega a fallar, no se ve afectado todo el sistema general; por otro lado, hablar de un sistema centralizado representa un componente (microchip o microcontrolador) central el cual se encarga de controlar y gestionar cada uno de los subsistemas, con la desventaja que en este caso si algún subsistema falla, se ve afectado todo el sistema general, además, los tiempos de respuesta son un poco más lentos debido a la cantidad de información que es manejada (esto depende a su vez de la cantidad de subsistemas que sean adecuados).

Proyecto	Tipo de interface	Tipo de programación	Tipo de comunicación	Tipo de sistema
Control Domótico [19]	Arduino - Seeeduino	Programación secuencial con " <i>switch case conditions</i> "	WiFi / Puerto Serie (USB)	Centralizado
Prototipo de sistema domótico [20]	Matlab	Programación orientada a redes neuronales	<i>ZigBee Wireless Technology</i>	Modular
<i>Wireless home automation</i> [21]	MPLab / Proteus	Programación secuencial con " <i>switch case conditions</i> "	<i>ZigBee Wireless Technology</i>	Modular
Sistema domótico mediante control PDA [22]	PDA (<i>Personal Digital Assistant</i>) / <i>LabView</i>	Programación secuencial con " <i>switch case conditions</i> "	<i>Bluetooth</i>	Centralizado
Sistema domótico con controlador difuso [23]	Arduino / <i>LabView</i>	Programación difusa	WiFi	Modular
Interfaz de control inteligente de un sistema domótico mediante <i>PROLOG</i>	Arduino / <i>LabView</i> / <i>PROLOG</i>	Programación basada en relaciones de inferencia (<i>PROLOG</i>)	Puerto Serie (USB) / Cableado	Centralizado

Tabla 4.1: Comparación respecto a otros sistemas

De acuerdo a la tabla 4.1, se pueden observar algunas desventajas de esta propuesta haciendo referencia al tipo de comunicación y tipo de sistema, ya que este proyecto sigue siendo un sistema centralizado, lo cual repercute directamente en la velocidad de respuesta con respecto a los demás, además que se comunica mediante cableado (lo cual puede ser modificado en versiones posteriores). Por otra parte en cuanto al tipo de interfaz se observa que la mayoría de ellos se encuentra en condiciones similares, todos y cada uno de ellos explotando las características de interfaces visuales, en específico *LabView*; Sin embargo, la ventaja que presenta esta propuesta es el uso del lenguaje *PROLOG* que es lo que sustenta la importancia de este proyecto, ya que, por medio de este lenguaje se explota la capacidad inherente al lenguaje de generar sistemas inteligentes, sistemas que puedan relacionar la información obtenida por los

sensores y a partir de una relación lógica de información, definir la mejor respuesta de salida, situación que no comparten los otros esquemas mostrados, ya que tales esquemas deben de identificar exactamente las características de salida del sistema para que este funcione de forma autónoma. Al pertenecer a un modelo imperativo, en PROLOG no es necesario describir la solución del problema, basta con describir la información de entrada y PROLOG se encargara de devolver el resultado de salida. Bajo este esquema, es posible ver que la propuesta amplía el panorama de aplicación de un lenguaje de programación (*PROLOG*) y de esta manera genera la capacidad de competir contra otros sistemas de mayor nivel (en cuanto a complejidad de protocolos y programación) con un lenguaje básico y no tan explotado.

Capítulo 5

Conclusiones

PROLOG fue el primer lenguaje de programación orientado al desarrollo de inteligencia artificial, el cual funciona con bases de datos. Dicha característica ha limitado la aplicación de este lenguaje a un entorno en el que sea utilizado con bases de datos que se ingresan manualmente o a través de otro software. Gracias al prototipo desarrollado, se ha podido realizar una interacción directa con un entorno físico lo que abre un panorama de aplicaciones para explotar el potencial, capacidades y características de este lenguaje.

Durante el desarrollo de este proyecto y a lo largo de las etapas de prototipos y fases de desarrollo del mismo, existió la posibilidad de poder presentarlo en diferentes congresos especializados en innovación y desarrollo tecnológico ingenieril, así como la aplicación de inteligencia artificial en el desarrollo de nuevas tecnologías, lo cual ha avalado la importancia y trascendencia de este proyecto como una innovación respecto a la aplicación y uso de esta herramienta. De igual manera, se obtuvieron algunas publicaciones de los artículos generados en los congresos, como lo fue en el Congreso Científico Tecnológico de las carreras de Ingeniería Mecánica Eléctrica, Industrial y Telecomunicaciones, sistemas y electrónica [16], en el que se obtuvo la publicación en las memorias del mismo congreso. Asimismo, la publicación de un artículo en el Congreso Internacional sobre Innovación y Desarrollo Tecnológico (CIINDET)[17], el cual fue organizado por el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE) de la Sección Morelos, en el cual se obtuvo la publicación de un capítulo del libro del mismo congreso.

De acuerdo con los objetivos planteados al inicio de este proyecto, se puede determinar que se cumplieron todos en su totalidad, ya que se logró realizar la interacción de un sistema físico de sensores que interactúan en tiempo real con un sistema que es monitoreado y controlado a través de *PROLOG*. Como se ha mencionado a lo largo del proyecto, las limitantes de este lenguaje han sido superadas exitosamente al poder interactuar de manera continua y en tiempo real con variables físicas creando un entorno visual que nos permite optimizar los recursos con los que contamos (actuadores y sensores).

En el desarrollo de este proyecto se valió únicamente de dos sensores diferentes y dos actuadores que acompañan uno a uno a los sensores para realizar la interacción; lo que se demuestra en base a este proyecto es que de acuerdo al alcance del sistema de sensado o medio de adquisición de datos del entorno físico, se puede controlar y gestionar de una manera óptima mediante *PROLOG*, gracias a su enfoque y optimización de recursos orientados a inteligencia artificial.

El proyecto se llevó a un punto de desarrollo óptimo para cumplir con los objetivos planteados, aunque el desarrollo se puede llevar más allá adecuando un sistema de monitoreo visual con más diseño e imágenes que puedan ser más agradables para el usuario y demás detalles que beneficien la parte visual del mismo lo que puede generar un entorno más amigable, comprensible y manejable para una posible distribución e implementación a nivel comercial.

Bibliografía

- [1] PROTOPAPAS, C. A., PSALTIRAS, K. P., & MACHIAS, A. V. (1991). *An expert system for substation fault diagnosis and alarm processing. IEEE Transactions on Power Delivery*, 6(2), 648-655.
- [2] FUGLSANG-FREDERIKSEN, A., RØNAGER, J., & VINGTOFT, S. (1989). *PC-KANDID: an expert system for electromyography. Artificial Intelligence in Medicine*, 1(3), 117-124.
- [3] ROVENTA, E., & ROSU, G. (2009, JULY). *The diagnosis of some kidney diseases in a small prolog Expert System. In Soft Computing Applications, 2009. SOFA'09. 3rd International Workshop on* (pp. 219-224). IEEE.
- [4] SINGLA, J. (2013). *The diagnosis of some lung diseases in a prolog expert system. International Journal of Computer Applications*, 78(15).
- [5] GOETHE, J. W., & BRONZINO, J. D. (1995). *An expert system for monitoring psychiatric treatment. IEEE Engineering in Medicine and Biology Magazine*, 14(6), 776-780.
- [6] OVERBY, M. A. (1987). *Psyxpert: an expert system prototype for aiding psychiatrists in the diagnosis of psychotic disorders. Computers in biology and medicine*, 17(6), 383-393.
- [7] JUAN IVÁN DÍAZ REYES, ALBERTO MARTÍNEZ CONTRERAS y DAVID TINOCO VARELA. *Interfaz de diagnóstico de trastornos mentales relacionados al consumo de alcohol realizada por medio de programación lógico-recursiva en PROLOG*. Memorias del Congreso Científico Tecnológico de las carreras de Ingeniería Mecánica Eléctrica, Industrial y Telecomunicaciones, sistemas y electrónica AÑO 3. No. 3. ISSN-2448-7236. SEPTIEMBRE 2018 – AGOSTO 2019, IC-05.
- [8] PINEDO ARRIAGA O. TS'ZUL, PINEDO ARRAIGA CARLOS D., HERRERA ALVARADO EDUARDO, & TINOCO VARELA DAVID. *Identification of wavelengths from the visible spectrum by means of Arduino for the generation of a knowledge base managed by PROLOG*. MATEC Web of Conferences. Vol. 210. EDP Sciences, 2018.
- [9] HSU, G.. *U.S. Patent No. 7,526,539. Washington, DC: U.S. Patent and Trademark Office*. 2009.

- [10] JIVANI, F. D., MALVANKAR, M., SHANKARMANI, R.. *A Voice Controlled Smart Home Solution with a Centralized Management Framework Implemented Using AI and NLP*. In 2018 International Conference on Current Trends towards Converging Technologies (ICCTCT) (pp. 1-5). IEEE. 2018.
- [11] KOUTROULIS, EFTICHIOS, AND KOSTAS KALAITZAKIS. *Development of an integrated data-acquisition system for renewable energy sources systems monitoring*. Renewable Energy 28.1 (2003): 139-152.
- [12] FRANSISKA, R. W., ET AL. *Electrical power measurement using arduino uno microcontroller and labview*. Instrumentation, Communications, Information Technology, and Biomedical Engineering (ICICI-BME), 2013 3rd International Conference on. IEEE, 2013.
- [13] CĂLINOIU, D., ET AL. *Arduino and LabVIEW in educational remote monitoring applications*. Frontiers in Education Conference (FIE), 2014 IEEE. IEEE, 2014.
- [14] JABER, ALAA A., AND ROBERT BICKER. *Real-time wavelet analysis of a vibration signal based on arduino-uno and labview*. International Journal of Materials Science and Engineering 3 (2015): 66-70.
- [15] AFRAM, A., JANABI-SHARIFI, F.. *Theory and applications of HVAC control systems—A review of model predictive control (MPC)*. Building and Environment, 72, 343-355. 2014.
- [16] VICTOR ADRIAN ESPINOSA DE LOS MONTEROS PARRA y DAVID TINOCO VARELA. *Interfaz de adquisición de datos lumínicos gestionada por PROLOG*. Memorias del Congreso Científico Tecnológico de las carreras de Ingeniería Mecánica Eléctrica, Industrial y Telecomunicaciones, sistemas y electrónica AÑO 4. No. 4. ISSN-2448-7236. SEPTIEMBRE 2019 – AGOSTO 2020, IF-01.
- [17] ESPINOSA DE LOS MONTEROS PARRA, V.A., TINOCO VARELA, D., & GUDIÑO PEÑALOZA, F.(2019) *Interfaz de obtención de datos a través de Arduino y LabView, gestionados por PROLOG*. En V. A. Espinosa de los Monteros Parra, D. Tinoco Varela, & F. Gudiño Peñaloza (Eds.). Tecnologías Útiles para la Sustentabilidad Energética para el Beneficio de la Sociedad (pp. 67-73). Cuernavaca, Morelos, México: IEEE.
- [18] WELZER, T AND ROZMAN, I AND GYORKOS, J. (1989) *Using PROLOG for digital circuits simulation*, en Proceedings. Electrotechnical Conference Integrating Research, Industry and Education in Energy and Communication Engineering' (pp. 746–749),IEEE.
- [19] LLEDÓ SÁNCHEZ, E., SÁEZ BARONA, S., & ATIENZA VANACLOIG, V. (2012) *Diseño de un sistema de control domótico basado en la plataforma Arduino* (tesis de pregrado). Universitat Politècnica de València, Valencia, España.

- [20] BARRERA DURANGO, MELISA, & LONDOÑO OSPINA, NELSON, & CARVAJAL, JORGE, & FONSECA, ALEJANDRO (2012). *Análisis y diseño de un prototipo de sistema domótico de bajo costo*. en Revista Facultad de Ingeniería Universidad de Antioquia, (63),117-128.[fecha de Consulta 12 de Mayo de 2020]. ISSN: 0120-6230. Disponible en: <https://www.redalyc.org/articulo.oa?id=430/43025100010>
- [21] V. SATHYA NARAYANAN, & S. GAYATHRI (2013). *Design of Wireless Home automation and security system using PIC Microcontroller*. en International Journal of Computer Applications in Engineering Sciences, Volume III, Special Issue, 135-140. ISSN: 2231-4946.
- [22] QUINTANAR OYOS, E., & CORREA FERNÁNDEZ, A., & BLANCO ORTIZ, A. (2010). *Diseño y simulación de un sistema domótico mediante un control PDA para una pasarela domótica utilizando software LabView* (tesis de pregrado). Universidad de Pinar del Rio, Pinar del Rio, Cuba.
- [23] VÁZQUEZ FERNÁNDEZ, JOSÉ ALBERTO, & TINOCO VARELA, DAVID (2018). *Desarrollo de un sistema domótico con controlador difuso y controlador manual, implementado en LabView y Arduino IDE*. en Research in Computing Science, 147(6), 249-264. ISSN 1870-4069.