

BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA



**FACULTAD DE CIENCIAS DE LA
ELECTRÓNICA**



Tesis

**“DISEÑO Y CONSTRUCCIÓN DE UN SENSOR
INTELIGENTE DISCRETO PARA EL USO DE DIFERENTES
APLICACIONES”**

TESIS PRESENTADA PARA OBTENER EL TITULO DE:

LICENCIADO EN ELECTRÓNICA

Presenta:

Carlos Eduardo Rugerio Mendieta

Asesor: M.C. Nicolás Quiroz Hernández

Puebla, Pue. Julio de 2016

Agradecimientos

Agradezco a Dios por darme la oportunidad de terminar una etapa más de mi vida.

Agradezco de manera especial a mi mamá Vero, pues ella fue el principal cimiento para la construcción de mi vida profesional, inculco en mí, las bases de superación y responsabilidad, en ella tengo el mejor ejemplo de perseverancia, cariño y comprensión, contando con un gran amor que me motivaron a culminar mi licenciatura, sin sus consejos y regaños nunca podría haber terminado.

Agradezco a mi papá Carlos por la paciencia que me tuvo en momentos difíciles, por enseñarme que la vida está llena de retos y ante todo darme su amor y cariño como principales herramientas para poder superarme y ver mis sueños hechos realidad.

Agradezco a mi asesor por darme la oportunidad de consolidarme como profesionista, por apoyarme con sus consejos y sobre todo por compartir conmigo sus conocimientos durante la elaboración de la tesis.

Agradezco a mi novia, por haberme apoyado de una manera incondicional para que todo esto fuera posible, por su paciencia y lealtad, pero sobre todo por su amor y cariño que sirvió de inspiración para superarme con este logro.

Agradezco a mis amigos por compartir tantos momentos agradables y sobre todo por su verdadera amistad en las buenas y en las malas.

Agradezco a mi compañero de carrera Adán por sus consejos y sobre todo su paciencia en momentos difíciles.

CAPÍTULO 1: INTRODUCCIÓN.....	6
1.1 ANTECEDENTES.....	7
1.2 PLANTEAMIENTO DEL PROBLEMA.....	7
1.3 JUSTIFICACIÓN.....	8
1.4 OBJETIVO GENERAL.....	8
1.5 OBJETIVOS ESPECÍFICOS.....	8
1.6 ALCANCES.....	9
1.7 LIMITACIONES.....	9
1.8 CONTENIDO DE LOS CAPÍTULOS.....	9
CAPÍTULO 2: SENSORES	10
2.1 DEFINICIÓN DE UN SENSOR.....	11
2.2 DEFINICIÓN DE UN SENSOR INTELIGENTE	11
2.3 ESTADO DE ARTE DE LOS SENSORES	11
2.3.1 <i>Sensor de temperatura LM35</i>	11
2.3.2 <i>Sensor de movimiento PIR eléctrico HC-SR501</i>	13
2.3.3 <i>Sensor de gas MQ2</i>	14
2.3.4 <i>Sensor de iluminación TSL2561</i>	15
2.4 ACONDICIONAMIENTO Y DIGITALIZACIÓN DE SEÑALES	16
2.4.1 <i>Muestreo</i>	17
2.4.2 <i>Cuantificación</i>	17
2.4.3 <i>Codificación</i>	18
CAPÍTULO 3: MICROCONTROLADORES Y COMUNICACIÓN DEL SENSOR INTELIGENTE.....	19
3.1 DEFINICIÓN DE UN MICROCONTROLADOR.....	20
3.2 IMPORTANCIA DE LOS MICROCONTROLADORES PIC	20
3.3 COMUNICACIÓN DEL SENSOR INTELIGENTE	20
3.4 DEFINICIÓN DE COMUNICACIÓN	20
3.5 TIPOS DE COMUNICACIÓN.....	21
3.5.1 <i>Comunicaciones BUS I2C</i>	21
3.5.2 <i>Comunicación inalámbrica Bluetooth</i>	24
3.6 RELOJ DE TIEMPO REAL (RTC) DS1307	26
3.7 CARACTERÍSTICAS PARA LA ELECCIÓN DEL MICROCONTROLADOR.....	27
CAPÍTULO 4: DISEÑO DE SOFTWARE DEL SENSOR INTELIGENTE	29
4.1 SELECCIÓN DE DISPOSITIVOS.....	30
4.2 DIAGRAMA DE FLUJO GENERAL DEL SENSOR INTELIGENTE.....	30
4.3 DIAGRAMA DE FLUJO GENERAL DE CONFIGURACIONES DEL SENSOR INTELIGENTE	32

4.3.1	Configuración fecha y hora.....	34
4.3.2	Algoritmo de configuración de reloj-calendario del sensor inteligente.....	34
4.3.3	Configuración temporizador	35
4.3.4	Algoritmo configuración de temporizador	35
4.3.5	Configuración ahorro de energía.....	36
4.3.6	Algoritmo de configuración ahorro de energía.....	37
4.3.7	Configuración prevención de incendios.....	38
4.3.8	Algoritmo configuración prevención de incendios.....	38
4.3.9	Configuración temperatura.....	39
4.3.10	Algoritmo configuración de temperatura	40
4.3.11	Configuración sistema de seguridad	41
4.3.12	Algoritmo de configuración de seguridad	42
4.3.13	Configuración de iluminación	43
4.3.14	Algoritmo de configuración de iluminación	43
4.3.15	Configuración alarma despertador	44
4.3.16	Algoritmo de alarma despertador.....	45
4.4	DIAGRAMA DE FLUJO GENERAL DE LA MONITORIZACIÓN DEL SENSOR INTELIGENTE	46
4.4.1	Algoritmo del reloj-calendario sensor inteligente.....	48
4.4.2	Algoritmo de temporizador.....	50
4.4.3	Algoritmo de ahorro de energía	52
4.4.4	Algoritmo de prevención de incendios	54
4.4.5	Algoritmo de temperatura	56
4.4.6	Algoritmo de sistema de seguridad	58
4.4.7	Algoritmo de regulación de iluminación.....	60
4.4.8	Algoritmo de alarma.....	61
4.5	PROGRAMACIÓN DEL SENSOR INTELIGENTE	64
4.6	SOFTWARE PARA LA APLICACIÓN MÓVIL.....	66
4.6.1	Diseño de la aplicación	68

CAPÍTULO 5: DISEÑO DE HARDWARE Y CONSTRUCCIÓN DEL SENSOR INTELIGENTE..... 73

5.1	SIMULACIÓN DEL SISTEMA DEL SENSOR INTELIGENTE.....	74
5.2	PROGRAMA EAGLE.....	75
5.3	DISEÑO DEL PCB EN EAGLE VERSIÓN 7.3	76
5.4	CONSTRUCCIÓN DEL SENSOR INTELIGENTE	79

CAPÍTULO 6: RESULTADOS..... 82

6.1	RESULTADOS EXPERIMENTALES DEL SENSOR INTELIGENTE.....	83
6.2	IMPLEMENTACIÓN DEL SENSOR INTELIGENTE.....	88

CAPÍTULO 7: CONCLUSIONES	97
7.1 CONCLUSIÓN	98
7.2 TRABAJOS A FUTURO.....	99
BIBLIOGRAFÍA.....	100

Capítulo I

Introducción

Uno de los avances en los sistemas de automatización ha sido el desarrollo de aplicaciones con “sensores inteligentes”. Teniendo como tarea principal la monitorización de procesos determinados, dando como resultado la obtención de información en tiempo real para que posteriormente sea enviada a los centros de control donde se realiza la automatización y supervisión global del sistema.

1.1 Antecedentes

Los procesos automatizados en la actualidad tienen gran importancia a nivel mundial donde la automatización ha pasado de ser una herramienta de trabajo deseable a una herramienta indispensable para competir en el mercado globalizado, dando como resultado el aumento de la calidad de sus productos, reducir tiempos de producción, realizar tareas complejas y especialmente aumentar la rentabilidad.

Es importante mencionar que los sensores inteligentes se han fabricado en 2 dominios: analógico y digital. En el caso analógico es debido a que solo este tipo de señal está normalizada a nivel industrial, tanto en corriente como en tensión, la tendencia a ir a un bus digital se debe a que permitiría que todos los sensores y actuadores puedan entregar o recibir información del mismo bus, así el sistema de control solo tendría que identificar la fuente de cada de señal y generar las adecuadas respuestas del sistema [12].

Es importante mencionar que los sensores inteligentes representan una tecnología y un concepto nuevo de monitorización adecuado que supera a los sensores tradicionales que por separado tienen la parte del sensor, acondicionamiento de señal y sistema de comunicación [2].

1.2 Planteamiento del problema.

Un problema general de los sensores es que cada uno tiene una señal de salida diferente, propia, dependiente del fabricante y del medio de transmisión que se utilice. Una de las ventajas de los sensores inteligentes es que permiten uniformizar el tipo de salida de la señal ya sea entregando una señal analógica o una digital [2].

Los sensores inteligentes representan una nueva tecnología y un nuevo concepto de plantear el acondicionamiento y adquisición de señales. Estos elementos superan a los sensores tradicionales que integran por separado, el sensor, un acondicionamiento de la señal y comunicación, de esta manera un sensor inteligente puede integrar en una sola pastilla todos los elementos mencionados permitiendo una mejor y más rápida transferencia de información.

1.3 Justificación.

El desarrollo de la automatización con sensores inteligentes en aplicaciones como domótica, industria automotriz, invernaderos, etc. Es una rama que se pone en práctica cada día más en la sociedad actual, fundamentalmente se consolida en mejorar los niveles de vida de las personas, dando servicios como por ejemplo: sistemas automatizados, monitorización de variables, control de temperatura, control de luminosidad en invernaderos, sistema automático para estacionarse, detector de fallas, etc.

Con los avances tecnológicos se ha modificado la forma de trabajar, relacionarse socialmente, permitiendo elevar la calidad de vida de la sociedad, con las diferentes tecnologías se tienen alternativas que mejoran y proporcionan comodidades, tal es el caso de los edificios inteligentes, el internet de las cosas (IoT, Internet of Things) por mencionar solo algunas de las aplicaciones de los sensores inteligentes, por esto la necesidad de plantear un proyecto que dé a conocer una forma de vivir diferente con la utilización de nuevas tecnologías de la información y comunicación, como lo es la implementación de sensores inteligentes.

1.4 Objetivo General.

Diseñar un sensor inteligente capaz de comunicarse en una red de sensores de forma inalámbrica que aventaje a los sensores tradicionales, con una interface amigable para su uso en diversas aplicaciones.

1.5 Objetivos específicos.

- Investigar y clasificar los sensores para la aplicación del sensor inteligente.
- Determinar las señales para el sensor inteligente.
- Acondicionar las señales de los sensores y digitalizar sus señales.
- Diseñar un sensor inteligente con las variables determinadas.
- Implementar una etapa de radio frecuencia para interconectar el sensor inteligente.
- Diseñar una etapa de configuración para obtener flexibilidad de los sensores.
- Implementar los algoritmos para el sensor inteligente.
- Integrar las etapas en una tarjeta PCB.
- Probar el sistema completo.

1.6 Alcances

El sensor inteligente tiene 5 variables las cuales son las siguientes: temperatura, movimiento, iluminación, gas y humo, teniendo como tarea principal la monitorización de un sistema en el área de seguridad, dando como resultado la obtención de información para obtener una base de datos para que posteriormente sea enviada de forma inalámbrica a los centros de control donde se realiza la supervisión global del sistema. El sensor inteligente cuenta con una fase de configuración que otorga flexibilidad a la monitorización, contando con 7 opciones de configuración y teniendo una interfaz amigable para las diferentes alternativas que otorga el sensor inteligente.

1.7 Limitaciones

El sensor inteligente no cuenta con comunicación a internet, en futuras generaciones se podrá implementar para que el sistema entre en internet de las cosas (IoT) creciendo exponencialmente su labor de monitorización.

El sensor inteligente no se va a integrar en un circuito integrado dando como resultado un mayor costo de fabricación, menor eficiencia eléctrica y un tamaño considerable, para trabajos posteriores se podría realizar la implementación en un CI.

1.8 Contenido de los capítulos

En el capítulo 2 se define un sensor inteligente, se hace el análisis de la investigación de los sensores a utilizar en el desarrollo del sistema, así como sus principales características de cada uno de ellos. También se hace una reseña del procesamiento de señales. En el capítulo 3 se define un microcontrolador, así como sus principales características y la importancia en la actualidad. Además se define la comunicación, los tipos que existen y cuál es la más idónea para el sensor inteligente. En el capítulo 4 se hace un análisis de diseño de software del sensor inteligente, así como el desarrollo de los diferentes programas a utilizar, además se menciona las opciones de configuración que ofrece el sistema. En el capítulo 5 se ve el diseño de hardware y la construcción del sensor inteligente describiendo los diferentes programas utilizados para el diseño del hardware, además se describe todos los elementos que componen la construcción del sensor inteligente. En el capítulo 6 se menciona la implementación del sensor inteligente en un sistema domótico en el área de seguridad, además incluye los resultados obtenidos de la monitorización del sistema. En el capítulo 7 se ven las conclusiones del presente trabajo, mencionando los beneficios de utilizar un sensor inteligente, así como los trabajos a futuro que se pueden implementar.

Capítulo II

Sensores

El objetivo de este capítulo es analizar los principios elementales de los sensores así como sus características principales y su extenso uso actual en diferentes aplicaciones tales como: invernaderos, sistemas domóticos, automatización de procesos industriales, sistemas automotrices, etc. En términos generales, un sistema de monitorización consta de cuatro dispositivos: un elemento sensor, un acondicionador de señales, un microcontrolador y comunicaciones. El elemento sensor es el primero del sistema de medición, está en contacto con la monitorización del sistema que se mide.

2.1 Definición de un sensor

Un sensor es un elemento capaz de detectar magnitudes físicas o químicas, llamadas variables y transformarlas en señales eléctricas. Estas variables pueden ser: intensidad, temperatura, distancia, aceleración, desplazamiento, presión, fuerza, humedad, movimiento, etc. Una magnitud eléctrica puede ser una resistencia eléctrica (RTD), una capacidad eléctrica (sensor de humedad), tensión eléctrica (termopar), etc.

2.2 Definición de un sensor inteligente

Un sensor inteligente se trata de un conjunto de uno o varios elementos de sensores y algún instrumento de acondicionamiento de señal en una misma unidad física, es decir, la combinación de un sensor analógico o digital, acondicionamiento de señal, memoria, comunicación en un mismo circuito integrado [2].

2.3 Estado de arte de los sensores

La obtención de información en la actualidad tiene un papel muy importante a nivel mundial, en el cual se ven involucrados los sensores que son dispositivos encargados de tomar dicha información del medio ambiente y transformarlas en magnitudes eléctricas. En seguida se menciona una lista de sensores:

- Sensor de temperatura LM35.
- Sensor de movimiento PIR eléctrico HC-SR501.
- Sensor de gas MQ-2
- Sensor de iluminación TSL2561

A continuación se describe cada uno de los sensores mencionados con sus características principales.

2.3.1 Sensor de temperatura LM35

El Sensor de temperatura analógico LM35 de la figura 2.1, es un circuito integrado de temperatura de precisión y de fácil calibración, funciona como un Zener de 2 terminales. El LM35 tiene una tensión de ruptura directamente proporcional a la temperatura absoluta a $10 \text{ mV}/^\circ\text{K}$. Cuando se calibra a $25 \text{ }^\circ\text{C}$ el LM35 tiene por lo general menos de $1 \text{ }^\circ\text{C}$ de error en un rango de $100 \text{ }^\circ\text{C}$ de temperatura. A diferencia de otros sensores el LM35 tiene una salida lineal siendo su rango

de operación de $-40\text{ }^{\circ}\text{C}$ a $+100\text{ }^{\circ}\text{C}$. En la tabla 2.1 se observa las características principales del sensor de temperatura LM35.

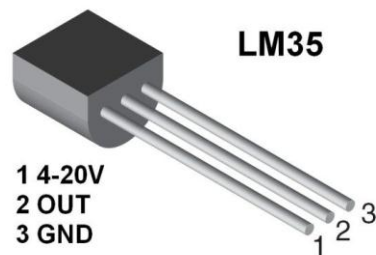


Figura 2.1. Sensor de temperatura LM35

Tabla 2.1. Características del sensor de temperatura LM35

Fabricante	Texas Instruments National
Categoría de producto	Sensores de temperatura
RoHs	RoHs
Tipo de salida	Analógica
Configuración	Local
Exactitud	$\pm 2\text{ }^{\circ}\text{C}$
Temperatura de trabajo máxima	$+ 150\text{ }^{\circ}\text{C}$
Temperatura de trabajo mínima	$0\text{ }^{\circ}\text{C}$
Voltaje de alimentación máxima	30 V
Voltaje de alimentación mínima	4 V
Apagado	No Shutdown
Estilo de montaje	Through Hole
Paquete/Cubierta	TO-92-3
Función del dispositivo	Sensor
Ganancia	$10\text{ mV}/^{\circ}\text{C}$
Corriente de salida	10 mA
Empaquetado	Bulk
Corriente de suministro	$91.5\text{ }\mu\text{A}$

2.3.2 Sensor de movimiento PIR eléctrico HC-SR501

Sensor de movimiento PIR eléctrico IR HC-SR501 de la figura 2.2, mide cambios en los niveles de radiación infrarroja emitida por objetos a su alrededor a una distancia máxima de 7 metros, como respuesta lógica cambia el nivel lógico de un pin, lo que facilita su integración a cualquier sistema. Una de sus principales aplicaciones es la detección de presencia de objetos o personas, en la tabla 2.2 se observa las características principales del sensor de movimiento PIR HC-SR501.

El módulo HC-SR501 tiene 3 pines de conexión +5 V, OUT (3.3 V) y GND (Tierra), y dos resistencias variables de calibración (CH1 y RL2).

- CH1: Con esta resistencia podemos establecer el tiempo que se va a mantener activa la salida del sensor. Una de las principales limitaciones de este módulo es que el tiempo mínimo que se puede establecer es de más o menos 3 segundos. Si la cambiamos a la resistencia por una de 100 K Ω , se podrá bajar el tiempo mínimo a 0.5 segundos.
- RL2: Esta resistencia variable permite establecer la distancia de detección que puede variar entre 3 a 7 metros.

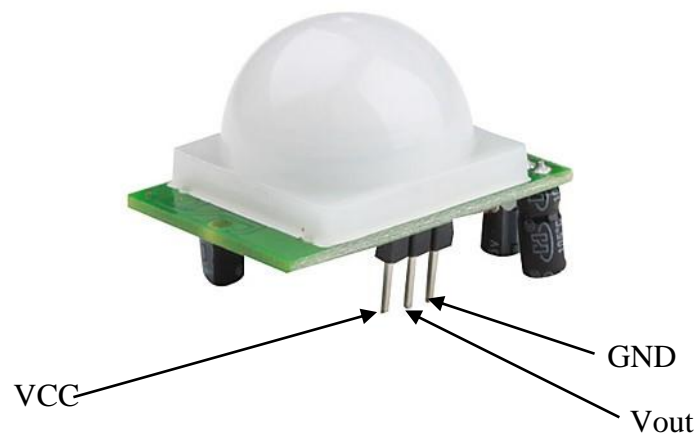


Figura 2.2 Sensor de movimiento

Tabla 2.2. Características del sensor de movimiento PIR eléctrico IR HC-SR501

Color	Blanco/Verde
Dimensiones	32*24*18 mm
Sensor infrarrojo	Tarjeta de control
Sensibilidad	Ajustable
Voltaje de trabajo	4.5 V a 20 V DC
Drenado de corriente	<60 μ A
Salida de voltaje:	Nivel de señal High/Low: Salida TTL
Distancia de detección	3 m -7 m (Ajustable)
Intervalo de detección	<140°
Retardo de tiempo	5-200 s (Ajustable, default 5 s +-3 %)
Tiempo de bloqueo	2.5 s (Default)
Disparo	L: no repetible, H repetible (Default)
Temperatura de trabajo	-20 °C a +80 °C

2.3.3 Sensor de gas MQ2

El funcionamiento de detectores de humo fotoeléctricos tiene el principio de dispersión de luz, es el interior de la cámara del detector, se encuentra un led emisor de luz, y un foto sensor, la luz emitida por el led incide en un área de la cámara donde no puede ser captada por el foto sensor, cuando se genera un incendio también se genera humo que entra en la cámara del detector y oscurece el medio en el que se propaga la luz emitida, que al recibir la luz genera la condición de alarma.

Sensor de gas MQ-2 de la figura 2.3a es un dispositivo encargado para el censado de gases en el hogar, industria, invernaderos, etc. Siendo adecuado para detectar gases como butano, metano LPG, propano, alcohol, hidrogeno y humo. En la figura 2.3b se observar el diagrama de conexiones y en la tabla 2.3 se observa las características principales del sensor de gas MQ-2.



Figura 2.3. a) Sensor de Gas MQ-2, b) Diagrama de conexiones

Tabla 2.3 Características principales del sensor de gas MQ-2

Símbolo	Nombre parámetro	Condiciones	Observaciones
V_C	Circuito de voltaje	5 V \pm 0.1	AC o DC
V_H	Voltaje calefacción	5 V \pm 0.1	AC o DC
R_L	Resistencia carga	ajustable	
R_H	Calentador resistencia	33 Ω \pm 5 %	
P_H	Consumo calefacción	800 mW	
T_{ao}	Usando	-20 °C a 50 °C	
T_{as}	Almacenamiento	-20 °C a 70 °C	
R_H	Humedad relativa	95 % Rh	
O_2	Concentración de oxígeno	21 % (condición estándar) concentración de oxígeno puede afectar sensibilidad	Mínimo valor es 2%

2.3.4 Sensor de iluminación TSL2561

Sensor de iluminación TSL2561 de la figura 2.4 fabricado por Texas Advanced Optoelectronic Solution (TAOS) es un dispositivo de medición de intensidad luminosa, ideal para usarlo en lugares donde se requieran un rango amplio de medición, este sensor es uno de los más precisos por el costo del mismo y puede ser configurado en ganancia y/o tiempo de integración para detectar diferentes rangos de luz. Este sensor transforma el nivel de luminosidad en una señal eléctrica que

podrá ser utilizada con distintos fines a través de un microcontrolador externo. En la tabla 2.4 se puede observar las características principales del sensor de iluminosidad TSL2561.



Figura 2.4. Sensor de iluminación TSL2561

Tabla 2.4. Sensor de iluminación TSL2561. Información obtenida del data sheet de la empresa Texas Advanced Optoelectronic

Descripción	Min	Norm	Max	unidad
Suministro de voltaje (V_{DD})	2.7	3	3.6	V
Operación de temperatura libre-aire T_A	-30		70	°C
SCL, SDA Entrada baja de voltaje V_{IL}	-0.5		0.8	V
SCL, SDA Entrada alta de voltaje V_{IH}	2.1		3.6	V
Parámetro	Min	Típico	Max	unidad
I_{DD} Suministro de corriente		0.24	0.6	mA
V_{OL} INT, SDA Salida bajo voltaje	0		0.6	V

2.4 Acondicionamiento y digitalización de señales

Los elementos acondicionadores de señales convierten la salida de los elementos sensores en una forma adecuada para el procesamiento posterior. Esta forma por lo general es un voltaje o una corriente proporcional al fenómeno que mide el sensor; el acondicionamiento puede constar de una etapa de amplificación, filtraje y acoplamiento de impedancias [6].

La digitalización de una señal analógica consta de tres procesos: muestreo, cuantificación y codificación para un convertidor de aproximaciones sucesivas. En algunos casos es necesario conectar un filtro pasa bajas para evitar traslape espectral (aliasing) [6].

2.4.1 Muestreo

Una señal continua $y(t)$ (figura 2.5a) podría representarse por un conjunto de muestras y_i , $i=1, \dots, N$, tomadas en intervalos discretos de tiempo ΔT (intervalos de muestreo). La operación se muestra (figura 2.5b), la frecuencia de muestreo es $f_s=1/\Delta T$. Para que la señal de muestreo $y_s(t)$ sea una representación adecuada de $y(t)$, f_s debe de cumplir con las condiciones del teorema de muestreo de Nyquist: para digitalizar una señal analógica se debe realizar a un frecuencia de muestreo mayor a dos veces la frecuencia máxima de la señal, $f_s \geq 2f_{MAX}$.

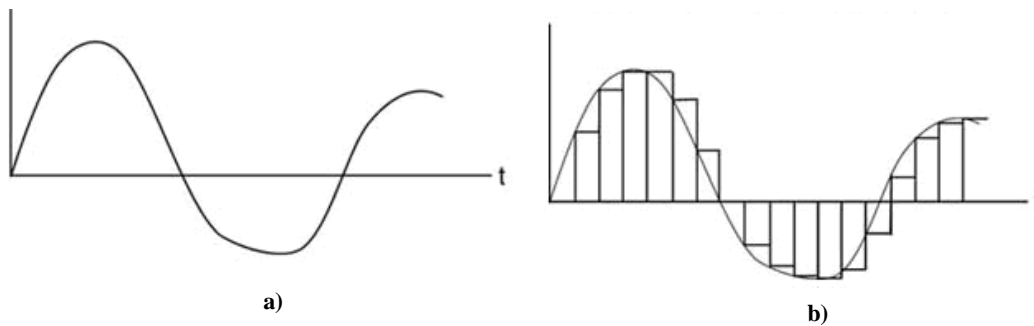


Figura 2.5. a) Señal original b) Señal muestreada

2.4.2 Cuantificación

La cuantificación de la figura 2.6 es el proceso de convertir valores continuos (voltajes) en serie de valores discretos de la señal analógica. Para esta parte del proceso los valores continuos de la señal original se convierten en series de valores numéricos decimales discretos correspondientes a los diferentes niveles o variaciones de voltajes que contiene la señal analógica original [7].

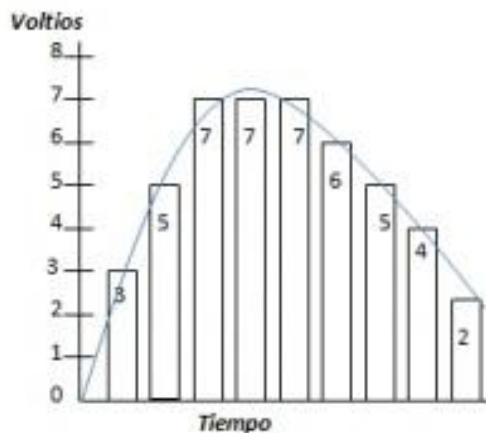


Figura 2.6 Cuantificación de señales

2.4.3 Codificación

La codificación (figura 2.7b) consiste en traducir los valores obtenidos durante la cuantificación (figura 2.7a) al código binario. El codificador transforma las muestras discretas obtenidas en el proceso de muestreo, en un conjunto finito de posibles amplitudes que se pueden codificar mediante una expresión binaria [7].

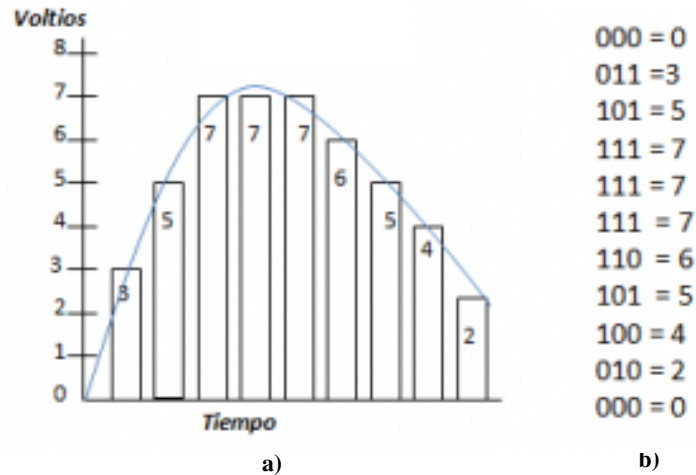


Figura 2.7 a) Cuantificación b) Codificación

El material revisado es necesario para el desarrollo del sensor inteligente, siendo los sensores el dispositivo que proporciona toda la información al dispositivo de control, por lo tanto es importante conocer todas sus características para que cumplan con los requerimientos del sistema. Por otro lado la digitalización de señales es necesaria para el desarrollo del sensor inteligente, por ello la importancia que representa la conversión de una señal analógica en formato digital. En el capítulo 3 se define a un microcontrolador y su importancia para el desarrollo de este proyecto.

Capítulo III

Microcontroladores y comunicación del sensor inteligente

En este capítulo se describe la importancia de los microcontroladores en la actualidad, algunas de sus características, formas de comunicación, memoria, etc. Debido a todas estas razones los microcontroladores se utilizan en una infinidad de circuitos electrónicos, debido a que permiten reducir el tamaño y el precio en varios sistemas. También se realiza el análisis de los tipos de comunicación que se puede integrar al sensor inteligente, así como las ventajas de cada una de estas.

3.1 Definición de un microcontrolador

Un microcontrolador es un circuito integrado programable que contiene todos los componentes necesarios para controlar el funcionamiento de una tarea determinada, como el control de una lavadora, un teclado de un ordenador, una impresora, un sistema de alarma, etc.[10]

3.2 Importancia de los microcontroladores PIC

Los microcontroladores están conquistando el mundo, están presentes en nuestro trabajo, en nuestra casa y en nuestra vida diaria. Se pueden encontrar controlando el funcionamiento de los ratones y teclados de las computadoras, en teléfonos, en hornos de microondas, televisores, etc. Seremos testigos de la conquista masiva de estos diminutos sistemas de control, que gobernarán la mayor parte de aparatos que fabriquen y usemos los seres humanos.

3.3 Comunicación del sensor inteligente

El creciente interés en sensores inteligentes es un resultado directo de la necesidad de comunicar información de los sensores a los sistemas de control. Desafortunadamente numerosos protocolos se han definido para la comunicación de datos, estándares de la industrial han sido y están siendo desarrollados en varias aplicaciones dentro de un segmento de mercado determinado, varias propuestas están compitiendo por ser aceptadas. Por eso la importancia de plantear un buen tipo de transmisión de datos que cumpla con todas las normas establecidas [8].

3.4 Definición de comunicación

La comunicación de datos se define como el intercambio de información entre uno más dispositivos. Los elementos básicos de la comunicación son la intención de comunicar, la composición del mensaje, la codificación, la transmisión de la señal, la recepción de la señal, la decodificación del mensaje y finalmente, la interpretación de la información por parte de un receptor. En la figura 3.1 se observa un diagrama a bloques de un sistema de comunicación.

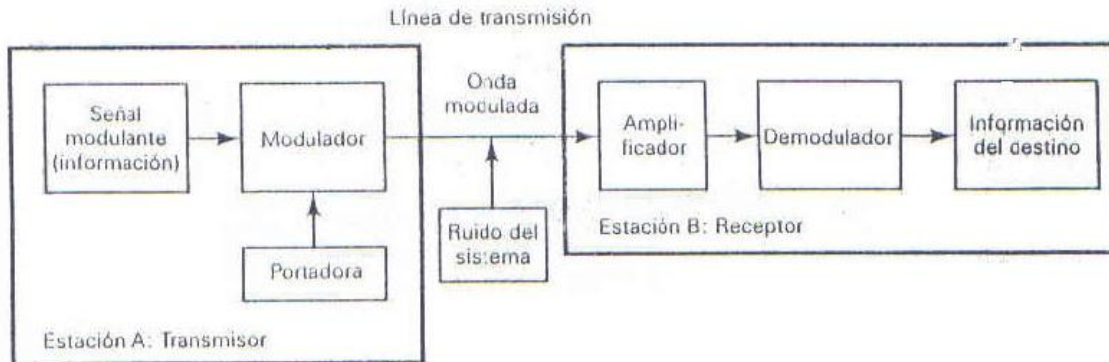


Figura 3.1. Sistema de comunicación. Figura obtenida del libro "Comunicaciones electrónicas de Wayne Tomasi"

3.5 Tipos de comunicación

Existen dos tipos básicos de sistemas de comunicación electrónica, analógica y digital. Un sistema de comunicación analógico es un sistema en el cual la energía electromagnética se transmite y recibe en forma analógica (una señal variando continuamente tal como una onda senoidal). Los sistemas de radio comerciales emiten señales analógicas. Un sistema de comunicación digital es un sistema en el cual la energía electromagnética se transmite y recibe en forma digital (niveles discretos de 5 V y tierra) [9].

Los sistemas de comunicación analógicas fueron los primeros en desarrollarse; sin embargo, en los últimos años la comunicación digital se ha hecho más común teniendo como plataforma la forma de transmisión de datos guiada y no guiada.

3.5.1 Comunicaciones BUS I2C

Actualmente hay en el mercado multitud de dispositivos que son gobernados por un bus serie desarrollado por la empresa Philips que es conocido como bus I2C, entre estos dispositivos podemos encontrar desde un simple circuito integrado hasta sistema complejos, como puedes ser un sintonizador.

Numerosos proyectos que utilizan microcontroladores de 8 bits suele regirse por criterios de diseños muy reiterados:

- El sistema consta de, al menos, un microcontrolador y varios dispositivos periféricos como memorias, LCD, convertidores ADC, etc.
- La conexión entre los distintos dispositivos que componen el sistema debe ser fácil de realizar y su costo mínimo.
- Estos sistemas suelen realizar funciones que no requieren una alta tasa de transferencia de datos, generalmente no superior a 100 Kbits por segundo.
- El sistema no debe depender de los dispositivos conectados a él. De otro modo no sería posible realizar modificaciones o mejoras.

Para implementar un sistema que satisfaga estos criterios se necesita una estructura de bus serie, ya que aunque no tienen ni la capacidad ni la velocidad de los buses paralelos, requieren poco hardware y un mínimo de cableado, este bus serie no debe ser simplemente un hilo de conexión, debe de incorporar una serie de procedimientos o protocolos para la correcta comunicación entre los componentes del sistema [10].

Todas estas bases son la base sobre la que se fundamentan las especificaciones del bus serie I2C para la interconexión de circuitos integrados o bus I²C desarrollado por Philips Semiconductors y que es ampliamente utilizado en la industria electrónica [10].

El I2C es un bus serie, formado por dos hilos. Que puede conectar varios dispositivos mediante en hardware muy simple, tal como se ilustra en la figura 3.2. Por esos hilos se produce una comunicación serie, bit a bit. Se transmiten dos señales, una por cada línea:

- SCL (Serial Clock). Es la señal de reloj que se utiliza para la sincronización de los datos.
- SDA (Serial Data). Es la línea para la transferencia serie de los datos.

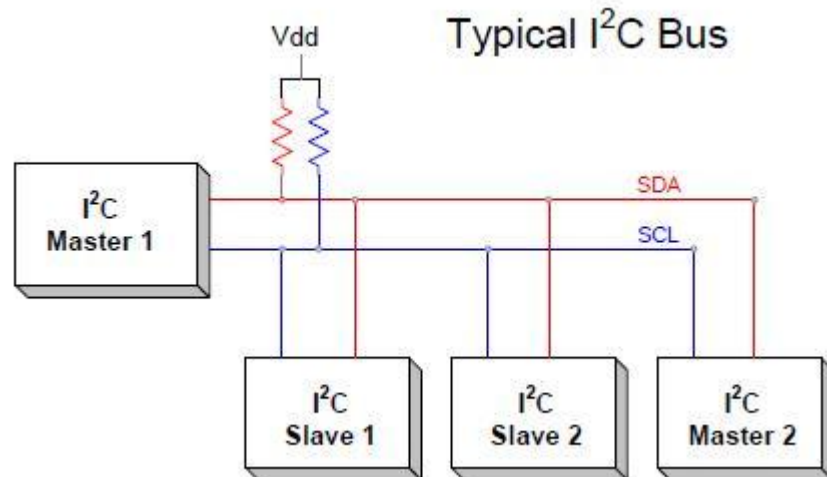


Figura 3.2. Comunicación típica I2C

Los dispositivos conectados al bus I2C mantienen un protocolo de comunicaciones del tipo maestro/esclavo. Las funciones del maestro y del esclavo se diferencian en:

- El circuito maestro inicia y termina la transferencia de información, además de controlar la señal de reloj. Normalmente es un microcontrolador.
- El esclavo es el circuito direccionado por el maestro.

La línea SDA es bidireccional, es decir, tanto el maestro como los esclavos pueden actuar como transmisores o receptores de datos, dependiendo de la función del dispositivo. Así por ejemplo, un display es solo un receptor de datos mientras que una memoria recibe y transmite datos. La generación de señales de reloj (SCL) es siempre responsabilidad del maestro.

Cada dispositivo conectado al bus I2C es reconocido por una única dirección que la diferencia del resto de los circuitos conectados. Los dispositivos compatibles con bus I2C suele llevar 2 o 3 pines para poder modificar esta dirección de modo que el diseñador puede evitar que un mismo diseño haya 2 o más esclavos con la misma dirección.

El bus I2C puede ser multi-master, esto significa que puede soportar más de un dispositivo capaz de controlar el bus. Los sistemas más comunes están constituidos por un solo microcontrolador maestro.

3.5.2 Comunicación inalámbrica Bluetooth

Los medios no guiado o sin cable han tenido gran acogida al ser un buen medio para cubrir grandes distancias en cualquier dirección, su mayor logro se dio desde la conquista espacial a través de los satélites y su tecnológica no para de cambiar. De manera general podemos definir las siguientes características de este tipo de medios tales como la transmisión y recepción se realiza por medio de antenas, las cuales deben estar alineada cuando él envió de información es direccional, o si es omnidireccional la señal se propaga en todas las direcciones.

Bluetooth es el nombre común de la especificación industrial IEEE 802.15.1, que define un estándar global de comunicaciones inalámbricas que posibilita la trasmisión de voz y datos entre diferentes dispositivos mediante un enlace de radiofrecuencia segura, (2.4 GHz).

El funcionamiento de dispositivos con tecnología Bluetooth pueden comunicarse de forma inalámbrica mediante redes de corto alcance nombradas piconets, cada dispositivo puede comunicarse simultáneamente con otros 7 dentro de una misma piconet, trabajando de 2.4 a 2.4835 GHz con amplio espectro y con la posibilidad de transmitir en Full Duplex con un máximo de 1600 saltos/s. los saltos de frecuencia se dan entre un total de 79 frecuencias con intervalos de 1 MHz; esto permite dar seguridad. Las principales características de este tipo de comunicación son:

- Facilitar las comunicaciones entre equipos móviles y fijos.
- Eliminar cables y conectores entre estos.
- Ofrecer la posibilidad de crear pequeñas redes inalámbricas.
- Bluetooth utiliza un método jerárquico con 2 niveles: maestro y esclavo.

En la figura 3.3 se puede observar un scatternet que es una interconexión de piconets, es este tipo de redes puede existir más de un maestro y además un esclavo puede tener más de una conexión asociada, siendo una desventaja el desempeño de este tipo de redes se va degradando cuando más elementos van añadiéndose a la scatternet.

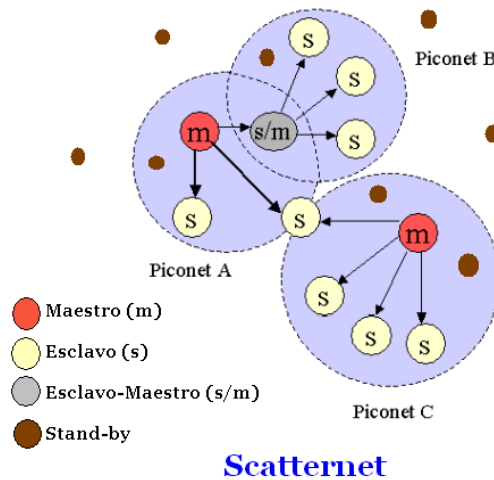


Figura 3.3. Conexión Scatternet

En la figura 3.4 se observa el dispositivo Bluetooth HC-05, con las siguientes características:

- Bluetooth versión v2.0.
- Tensión de alimentación de 3.3 V
- Frecuencia de 2.4 GHz
- Temperatura de trabajo: -20 °C a +75 °C.
- Seguridad de autenticación y encriptación.
- Soporta comando AT para configuración a través de un puerto serie.
- Distancia de conexión de 10 metros.



Figura 3.4. Módulo Bluetooth HC-05

3.6 Reloj de tiempo real (RTC) DS1307

El DS1307 es un reloj en tiempo real RTC (Real Time Clock) con líneas de conexión a un bus I2C. Este circuito integrado de la figura 3.5 es un poderoso reloj y calendario de tiempo real, que cumple perfectamente con muchas de las necesidades normales en la adquisición y registro del tiempo [10].

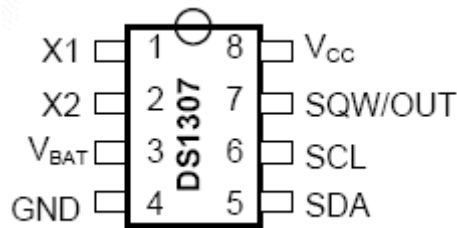


Figura 3.5 Patillaje del reloj-calendario DS1307

La tabla 3.1 describe los pines del reloj-calendario DS1307.

Tabla 3.1. Características del dispositivo DS1307

Vcc	Fuente de alimentación.
GND	Masa.
SDA	Línea de datos de bus I2C.
SCL	Línea de reloj del bus I2C.
X1,X2	Cristal de cuarzo de 32768 Hz.
SQW/OUT	Señal cuadrada
V _{BAT}	Batería de respaldo

A continuación se describe las características principales del módulo del reloj-calendario:

- Fabricado por Dallas Semiconductors (www.dalsemi.com), en un encapsulado de 8 pines (figura 3.5).
- El DS1307 es un reloj y calendario de tiempo real que cuenta los segundos, los minutos, las horas, los días de la semana (lunes, martes, etc.) los días del mes, los meses y los años, válido hasta el año 2100.
- Almacena los datos de formato BCD para que se pueda trabajar directamente con ellos.
- Tiene 56 bytes de RAM no volátil para almacenamiento de datos.

- En su pin SQW/OUT proporciona una onda cuadrada programable.
- Tiene una circuitería interna de “respaldo” para alimentación en caso de fallo de la alimentación principal, por tanto, es capaz de mantener el tiempo y la fecha actualizados aun cuando el sistema es apagado.
- Se puede alimentar entre 4.5 a 5.5 V, siendo su valor típico 5 V.
- Posee un bajo consumo, menos a 500 nA en el modo respaldo.
- Utiliza un cristal de cuarzo propio de 32.768 Hz para lograr tiempos exactos y no depender del microcontrolador.
- El último día del mes es automáticamente ajustado a 28, 29,30, o 31 días según corresponda, tiene en cuenta los años bisiestos.
- Puede trabajar en formato europeo de 24 horas o el americano de 12 con indicador de AM/PM.

3.7 Características para la elección del microcontrolador

A la hora de decidirse en términos generales sobre qué tipo o familia de microcontrolador a emplear hay que tener en cuenta varios factores, tales como:

- Experiencia previa: Si ya se ha trabajado con algún microcontrolador en particular, lo mejor es ver que nuevas posibilidades ofrecen los diversos fabricantes que trabajan con ese microcontrolador. Por ejemplo, el 8051 fue muy popular hace algún tiempo, y hay muchos microcontroladores actuales que derivan de este, como pueden ser los AT89 de Atmel, MCS251 de Intel, DS8 de Maxim (Dallas), MSC12 de Texas Instruments, lo importante que el conjunto de instrucciones se mantienen, modos de direccionamiento, nombres de registro, etc.
- Documentación existente: Es un factor importante si se quiere conocer bien el tipo de microcontrolador elegido y su entorno de desarrollo. Además, si existe una amplia literatura de aplicaciones podrán utilizarse programas y diseños ya realizados para adaptarlos a nuestras necesidades mediante libros especializados, revista de electrónica y sobre todo Internet, donde se puede encontrar información necesaria sobre cualquier dispositivo.

- Herramientas de desarrollo y su precio: Uno de los factores que más importancia tiene a la hora de seleccionar un microcontrolador entre todos los demás es el soporte software como hardware de que dispone. Un buen conjunto de herramienta de desarrollo puede ser decisivo en la elección, ya que pueden suponer una ayuda inestimable en el desarrollo de proyectos. Algunos fabricantes de microcontroladores ofrecen paquetes IDE de calidad de forma gratuita, como política para inclinarse por el uso de sus microcontroladores, ejemplo de ello son AVR Studio de Atmel, Code Warrior de Freescale (Motorola), MPLAB de Microchip o Eclipse de Texas Instruments. Estos paquetes IDE gratuitos permiten programar en código ensamblador, puesto que los compiladores de lenguaje de alto nivel (BASIC y C) no suelen ser gratis.

Con la selección del microcontrolador PIC18F4550 para la etapa de control del sistema, se puede concluir que tiene el rendimiento de procesamiento de señal adecuados para el sensor inteligente, además cuenta con una gran variedad de buses de comunicación fundamentales para el desarrollo del sistema. Una vez finalizado con la selección de sensores, el microcontrolador y la comunicación se puede visualizar el desarrollo de software requerido para la implementación del sensor inteligente.

Capítulo IV

Diseño de software del sensor inteligente

En este capítulo se presenta el diseño del sensor inteligente, el cual utiliza una gran variedad de herramientas de diseño de software, siendo de gran utilidad en cuestiones de simulación, desarrollo e implementación.

4.1 Selección de dispositivos

La elección de los elementos que conforman el diseño del sensor inteligente, fueron considerados en un sistema domótico en el área de seguridad. La investigación de los sensores se realizó mediante una búsqueda de las variables a utilizar, dando como resultado las siguientes: temperatura, iluminosidad, movimiento, gas y humo, siendo controladas por un microcontrolador PIC18F4550. La tarea principal del sistema es la monitorización, dando como resultado una base de datos del comportamiento del proceso, siendo de gran ayuda para diversas aplicaciones. Las características principales que proporciona el sensor inteligente son las siguientes:

- Configuraciones del sistema.
- Flexibilidad en las variables a utilizar.
- Monitorización en tiempo real.
- Capacidad de almacenamiento de datos en la memoria.
- Crecimiento del sistema a futuras generaciones.
- Comunicación vía inalámbrica.

4.2 Diagrama de flujo general del sensor inteligente

En la figura 4.1 se muestra el algoritmo general del sensor inteligente, donde su primer tarea será la de inicializar la comunicación Bluetooth para que posteriormente iniciar el sistema, cargando la última configuración guardada en memoria, si se requiere un cambio, se deben realizar las modificaciones pertinentes en la opción de configuración del sistema, están se guardan en la memoria para poder ejecutarlas posteriormente, de lo contrario si no existe ninguna modificación, el sistema se ejecutara sin cambio alguno y realizara las tareas encomendadas al sensor inteligente, hasta que se requiera de un cambio en su configuración.

Configuración

La configuración del sensor inteligente es la etapa encargada de realizar los cambios que el sistema requiera, teniendo en memoria varias opciones de configuración siendo de gran ayuda en diferentes aplicaciones. Por mencionar algunas se tienen ahorro de energía, temporizador, temperatura, alarma despertador, prevención de incendios, etc. Una vez seleccionado la configuración que se necesite se guarda en memoria para posteriormente pasar a la etapa

ejecución, de existir un cambio se retorna a esta etapa de “configuración” para la selección una nueva opción.

Monitorización

La monitorización es la etapa encargada de obtener uno o varios parámetros que entrega el sensor inteligente, teniendo como propósito el de enviar los valores obtenidos por la comunicación implementada, para que otros sistemas tomen decisiones.

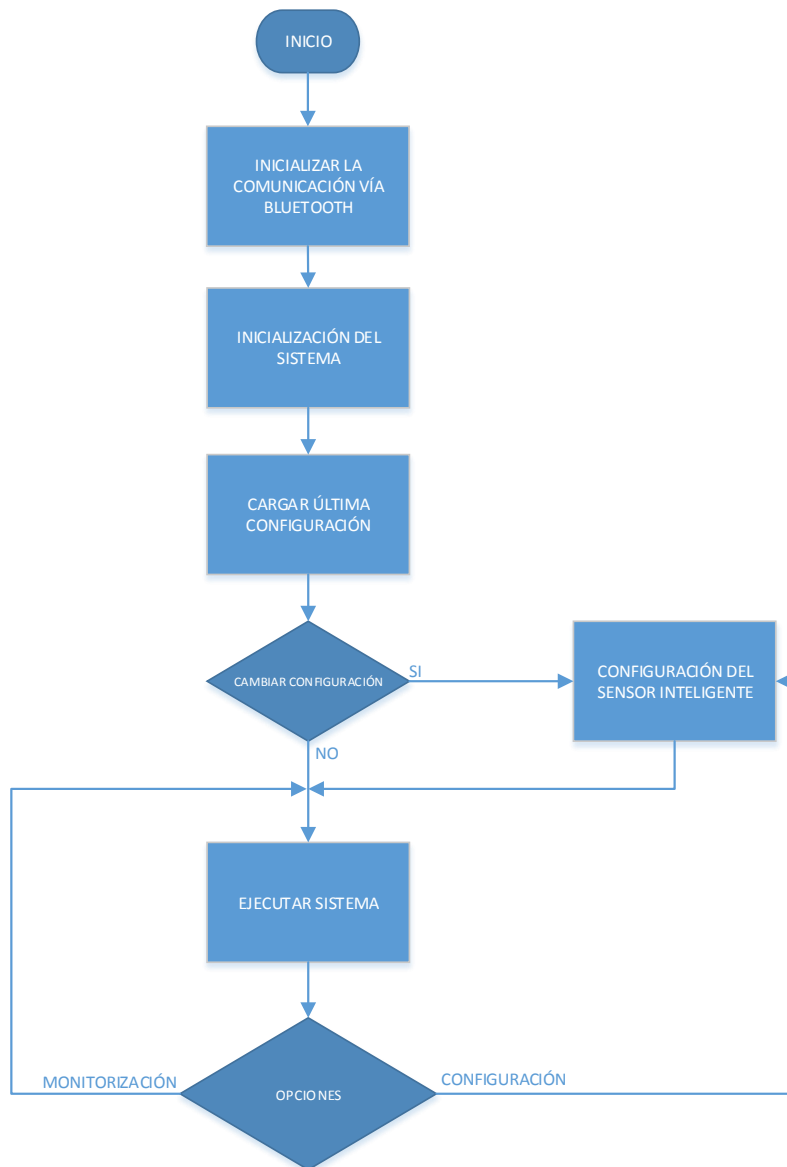


Figura 4.1. Diagrama General del Sensor inteligente

En seguida se describe cada una de las opciones de configuración del sensor inteligente:

4.3 Diagrama de flujo general de configuraciones del sensor inteligente

Las configuraciones del sistema se definen como un conjunto de etapas especiales que tienen como finalidad realizar una tarea específica definida por el usuario, estas opciones generalmente son cargadas en su inicio y en algunos casos se deberá reiniciar el sistema para ver los cambios realizados, ya que no podrá cargarlos mientras se esté ejecutando

La figura 4.2 se puede observar el diagrama de flujo general de las configuraciones del sensor inteligente permitiendo a los usuarios opciones diferentes de configuraciones facilitando el trabajo en varias aplicaciones en sistemas tales como invernaderos, industria automotriz, domótica, etc. Dando como resultado el ahorro de tiempo así como beneficios que pueden generar un impacto positivo a futuras generaciones de sensores, A continuación se nombran las opciones de configuraciones que ofrece el sensor inteligente.

- Configuración fecha y hora.
- Configuración temporizador.
- Configuración ahorro de energía.
- Configuración prevención de incendios.
- Configuración de temperatura.
- Configuración sistema de seguridad.
- Configuración de iluminación.
- Configuración alarma despertador.

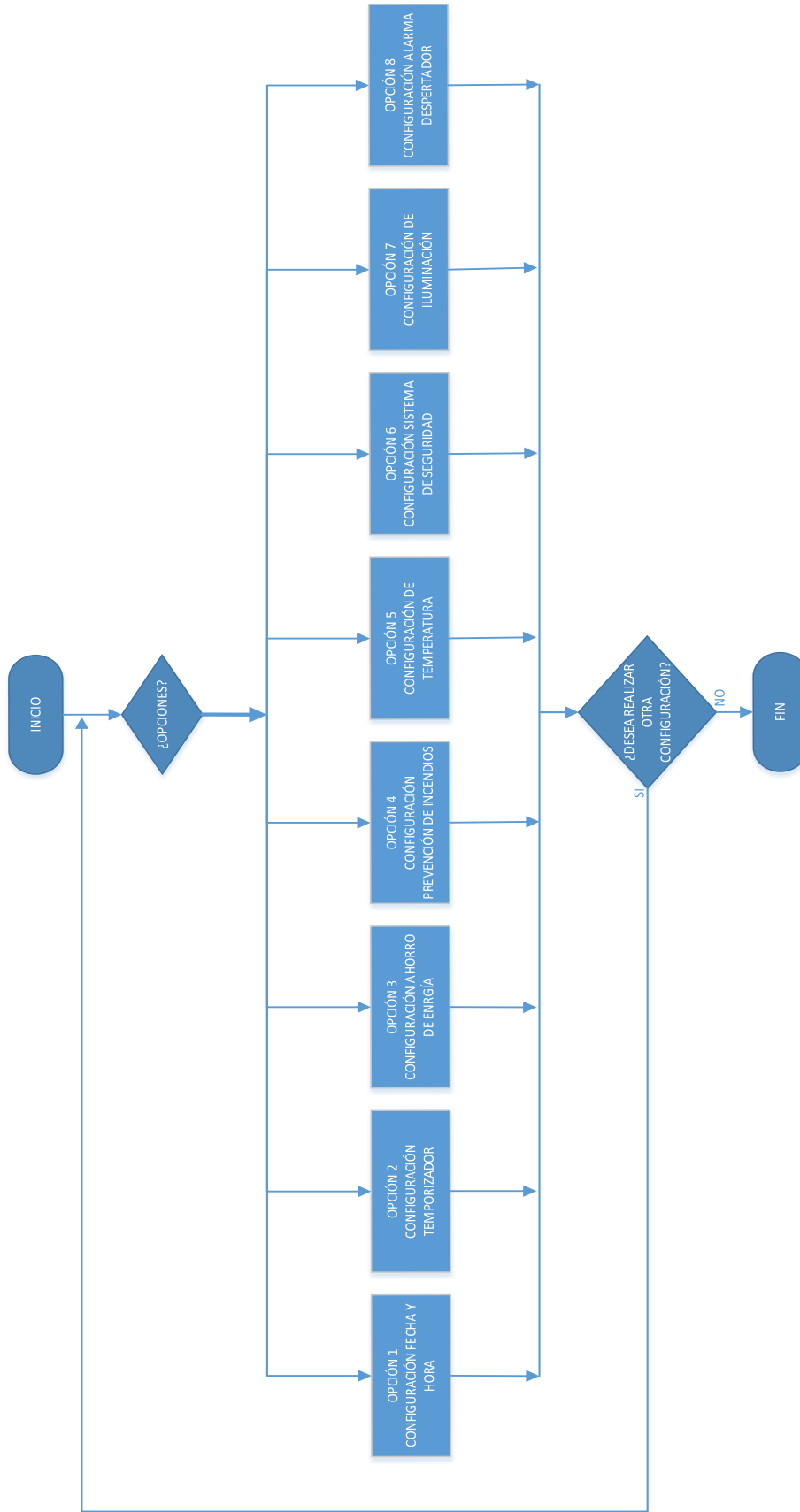


Figura 4.2. Diagrama de flujo general de configuraciones del sensor inteligente

4.3.1 Configuración fecha y hora

Más allá de ser un simple calendario y reloj informativo, la fecha y hora del sensor inteligente es la referencia temporal que tiene nuestro sistema permitiendo al usuario saber el estado e historial de los sensores en todo momento, ayudando a la monitorización en procesos tales como: cultivos, temperaturas, iluminosidad, etc. Siendo una herramienta fundamental para las configuraciones del sistema, dando como resultado un proceso actualizado.

4.3.2 Algoritmo de configuración de reloj-calendario del sensor inteligente.

En la figura 4.3 muestra el algoritmo de la configuración de un reloj digital y calendario basado en el chip DS1307 que representa un RTC (Real Time Clock) con líneas de conexión a un bus I2C y controlado por un microcontrolador PIC18F4550. El algoritmo tiene la opción de “introducir hora y fecha” para la configuración del sistema ayudando a trabajar en tiempo real, cumpliendo con muchas de las necesidades normales en la adquisición y registro de tiempos.

Una vez actualizada la fecha y hora el sistema trabajará sin necesidad de otra actualización hasta que el usuario lo requiera, teniendo una fecha garantizada hasta el año 2100, la información se guarda en la memoria dando como resultado un sistema actualizado.

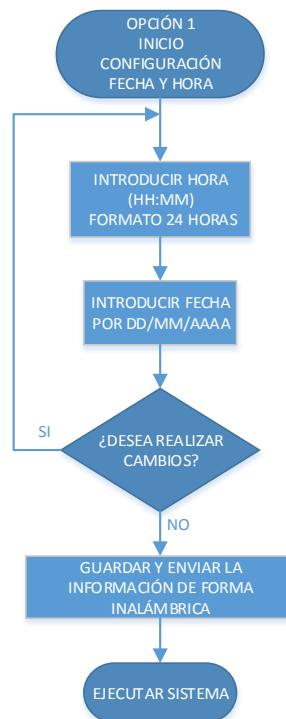


Figura 4.3. Configuración de fecha y hora

4.3.3 Configuración temporizador

Un temporizador es un dispositivo capaz de controlar un lapso de tiempo determinado, siendo de gran ayuda en una infinidad de sistemas. En la actualidad la mayor parte de los aparatos electrónicos cuentan con una función de temporizador tales como: teléfonos móviles, ordenadores personales, tabletas, despertador digital, etc.

Hay diversos tipos de temporizadores desde los que son usados en tareas simples en el hogar, hasta los que son utilizados en la automatización de procesos industriales, teniendo diferentes clases de componentes pero con la misma función, la de contar un lapso de tiempo específico. Su utilidad puede ser desde una simple alarma capaz de avisar un tiempo determinado hasta un sistema de control, permitiendo a dispositivos electrónicos (como una iluminación o una apertura de puertas) permanezca activada o desactivadas durante un tiempo establecido, esto puede ser implementando en una sistema domótico. Por estas razones es de gran importancia contar con un temporizador en nuestro sistema.

4.3.4 Algoritmo configuración de temporizador

El algoritmo de configuración de temporizador que ofrece el sensor inteligente de la figura 4.4 se observa, la etapa de introducir un tiempo en horas (HH) y minutos (MM) para el seguimiento de un proceso siendo de gran ayuda en una gran gama de aplicaciones donde se requiera un temporizador. El algoritmo entrega temporizaciones exactas, teniendo una alarma para indicadora que el tiempo se ha cumplido y también cuenta con la posibilidad de un paro al temporizador, esta será realizada por el usuario.



Figura 4.4. Configuración de temporizador

4.3.5 Configuración ahorro de energía

Actualmente el uso de energía eléctrica es fundamental para realizar gran parte de nuestras actividades, gracias a esto tenemos una mejor calidad de vida. Con el ahorro de energía se aprovecha los recursos energéticos, ahorrando el consumo de combustibles y evitando también la emisión de gases contaminantes en la atmosfera, previniendo el calentamiento global de nuestro planeta.

El ahorro y el cuidado del medio ambiente no son sinónimos de sacrificar o reducir nuestros niveles de bienestar o el grado de satisfacción de nuestras actividades cotidianas, por lo contrario, un cambio de hábitos puede favorecer una mayor eficiencia en el uso de la energía y así ayudar al ahorro de energía.

4.3.6 Algoritmo de configuración ahorro de energía

La figura 4.5 muestra el algoritmo de configuración de ahorro de energía, teniendo una lista de sensores que se pueden seleccionar, dando opciones de activación y desactivación, estos serán guardados en la memoria para posteriormente realizar la ejecución seleccionada (se puede elegir más de uno a la vez) dando como resultado un ahorro de energía en el estado de desactivación.

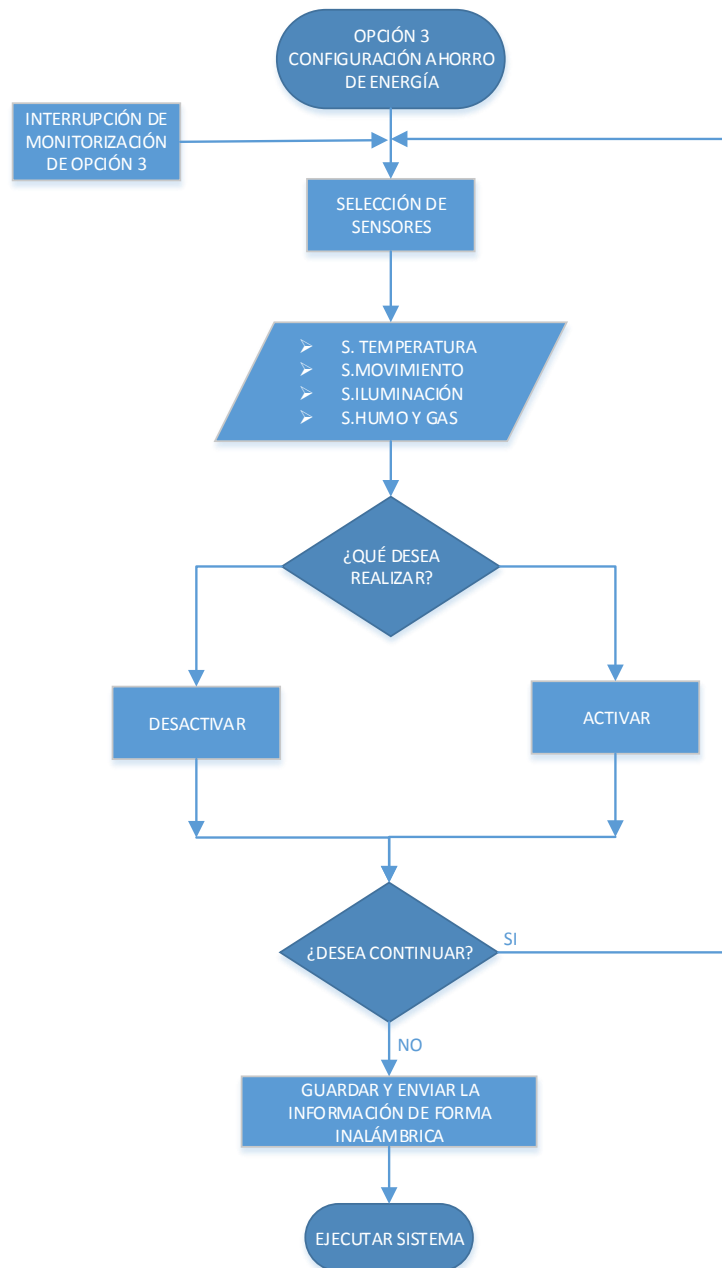


Figura 4.5. Configuración ahorro de energía

4.3.7 Configuración prevención de incendios

El humo, por su gran movilidad favorece a la propagación de incendios, poniendo en peligro la seguridad de los seres vivos como también la estabilidad estructural de edificios, hogares, zonas industriales, etc. Esto representa un riesgo de aislamiento y daño para las personas que se encuentran en estos lugares, una solución a este problema es utilizar un sensor inteligente que representa un sistema de gran ayuda para la detección de este tipo de peligros, teniendo una configuración llamada “prevención de incendios” que es capaz de activar una alarma cuando detecta la presencia de humo en el aire, emitiendo una señal acústica avisando del peligro y ayudando a prevenir una posible contingencia. Esta configuración es muy útil en una infinidad de sistemas, tales como: invernaderos, industria, centros comerciales, hogar, edificios, etc.

4.3.8 Algoritmo configuración prevención de incendios

En la figura 4.6 se observar el algoritmo de configuración prevención de incendios, teniendo como primera etapa la de introducir las contantes de gas mínimo y gas máximo para que posteriormente sean guardadas en memoria, contando con un tiempo de monitorización de cada segundo y así ayudar a llevar una base de datos de cómo se está comportando, dando como resultado el seguimiento del sistema.

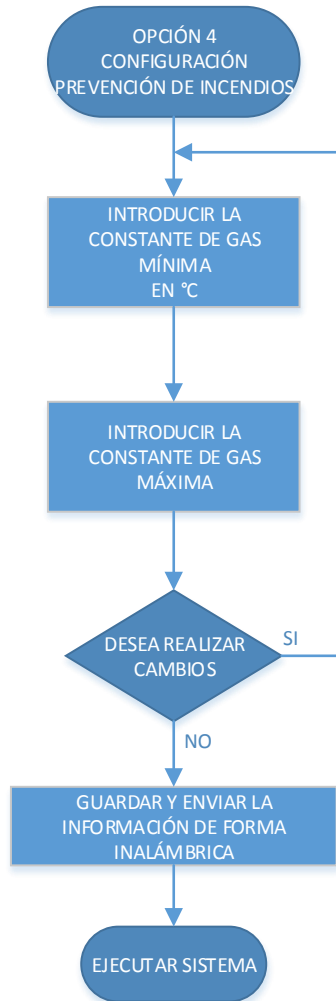


Figura 4.6. Configuración prevención de incendios

4.3.9 Configuración temperatura

La temperatura en la actualidad tiene gran importancia a nivel mundial, donde se ven involucrados el control de procesos, siendo hoy en día una de las variables más importantes en diferentes sistemas tales como: la industrial, investigación y desarrollo, obtención de información, etc.

A medida que la temperatura no era un aspecto crítico en sistemas que demandaban precisión, la tolerancia de pocos grados centígrados a menudo no comprometía el proceso, pero con el uso de métodos más complejos y que utilizan mediciones más exactas, la precisión es fundamental para el óptimo funcionamiento de los sistemas, teniendo en cuenta estos factores, la temperatura

ideal depende de muchos elementos tales como zona geográfica, horario del día, estación del año, etc. Considerando estos aspectos que son determinantes para su monitorización, el sensor inteligente tiene configuraciones que cumplen con todas las especificaciones a considerar, entregando al sistema una monitorización de temperatura precisa, misma que será configurada por el usuario, ayudando a controlar procesos y a generar una base de datos que facilite un registro de cómo se comporta nuestro sistema permitiendo a la toma de decisiones en tiempo real.

4.3.10 Algoritmo configuración de temperatura

La figura 4.7 se observa el algoritmo de la configuración de temperatura, a continuación se describe el funcionamiento:

➤ Intervalo de monitorización de temperatura

El intervalo de temperatura es para una monitorización de temperatura tanto mínima como máxima, teniendo como primera actividad la de introducir los parámetros de configuración de temperatura, el sistema solicitara los siguientes valores:

1. Valor de temperatura mínimo que posteriormente se almacenara en memoria.
2. Valor de temperatura máximo que de la misma manera se almacenara en memoria.

Una vez finalizado la configuración deseada se guarda y se envía al sistema de forma inalámbrica para el inicio de la monitorización, de existir algún cambio se regresa a fase de configuraciones, por ultimo tiene la opción de guardar la temperatura máxima de la monitorización para llevar un registro de cómo se comporta el sistema, (figura 4.7).

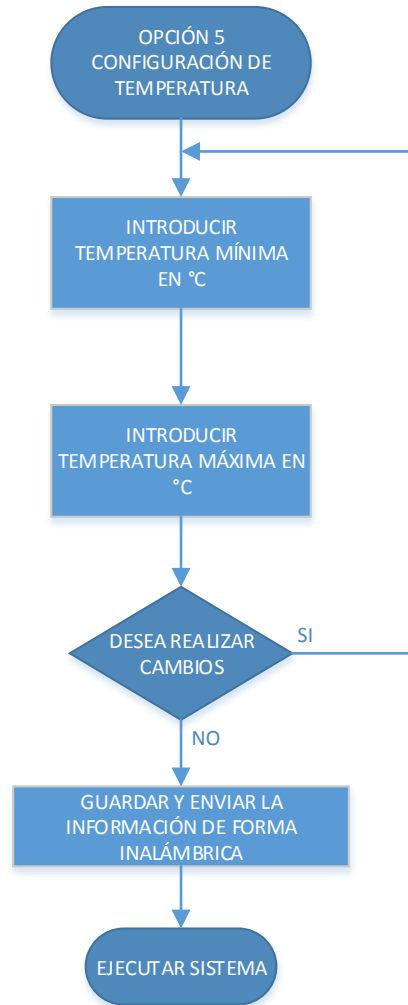


Figura 4.7. Configuración de temperatura

4.3.11 Configuración sistema de seguridad

Contar con un sistema de vigilancia en el hogar, la industria u oficinas no sólo es una cuestión de seguridad si no también la de evitar posibles allanamientos, dando tranquilidad mientras se está realizando actividades, con la configuración sistema de seguridad que ofrece el sensor inteligente se tiene la opción de modo guardián capaz de entregar la monitorización para la detección de la presencia de alguna persona, dando como resultado que el usuario recibe una advertencia en su Smartphone en tiempo real para avisarle y actué rápidamente, siendo de gran ayuda en nuestra vida cotidiana.

4.3.12 Algoritmo de configuración de seguridad

La figura 4.8 muestra el algoritmo de configuración de seguridad, teniendo como primera opción un modo guardián capaz de detectar posibles intrusos, de activarse el sistema queda en estado de monitorización hasta que se registre una presencia y se desactive de forma automática hasta que la detección sea inexistente.

Esta configuración cuenta con la opción de activar la alarma indicadora ayudando a notificar la detección, de no necesitarse esta alternativa, la configuración queda en estado de monitorización para posteriormente ser ejecutada hasta que necesite un cambio en su configuración realizada.

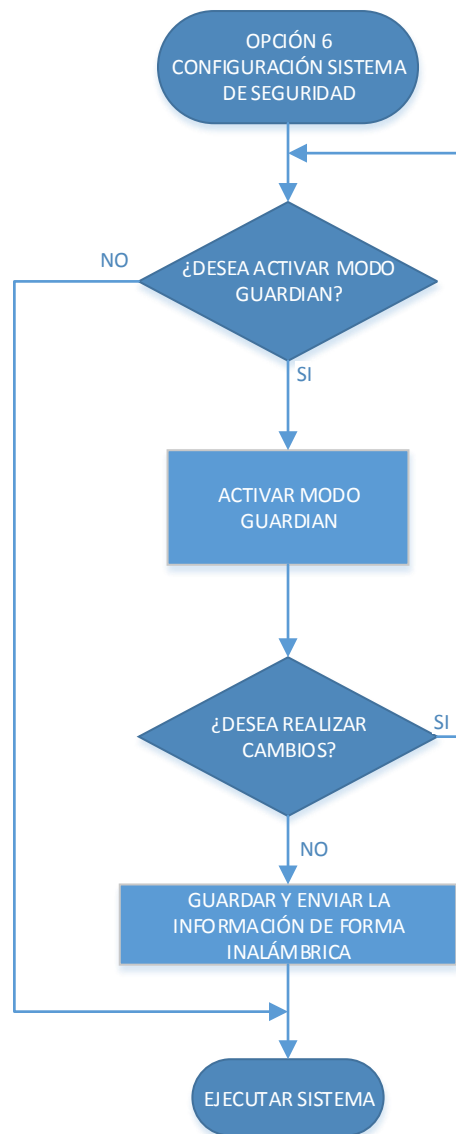


Figura 4.8. Configuración de seguridad

4.3.13 Configuración de iluminación

La iluminación se suele activar o desactivar de forma manual con un mecanismo de interruptor dando como resultado posibles accidentes por no encontrar dicho dispositivo en la oscuridad. Una solución para optimizar la iluminación en edificios, hogares o empresas es mediante el uso de un sensor inteligente, siendo capaz de la monitorización de luz ambiente, ayudando para el aprovechamiento de la luz solar y control de iluminación distribuida, dando como resultado un ahorro energético, también se puede utilizar para la monitorización de la iluminosidad en diferentes sistemas tales como: invernaderos, la industria, el hogar, etc.

4.3.14 Algoritmo de configuración de iluminación

La figura 4.9 se observa el algoritmo de configuración de iluminación, teniendo como primera opción la de introducir las constantes de iluminación mínima e iluminación máxima en formato luxes (lx) que serán las encargadas del monitoreo del sistema, estas son determinadas por el usuario, para posteriormente guardarlas en memoria. Es importante mencionar que el sistema tiene un rango de luxes mínimo de 0 y un máximo de 40000 luxes.

Esta configuración cuenta con la opción de activar la alarma indicadora ayudando a notificar si la iluminación mínima o máxima han sido rebasadas en su totalidad, de no necesitarse esta alternativa, la configuración queda en estado de monitorización para posteriormente ser ejecutada hasta que necesite un cambio en su configuración realizada.

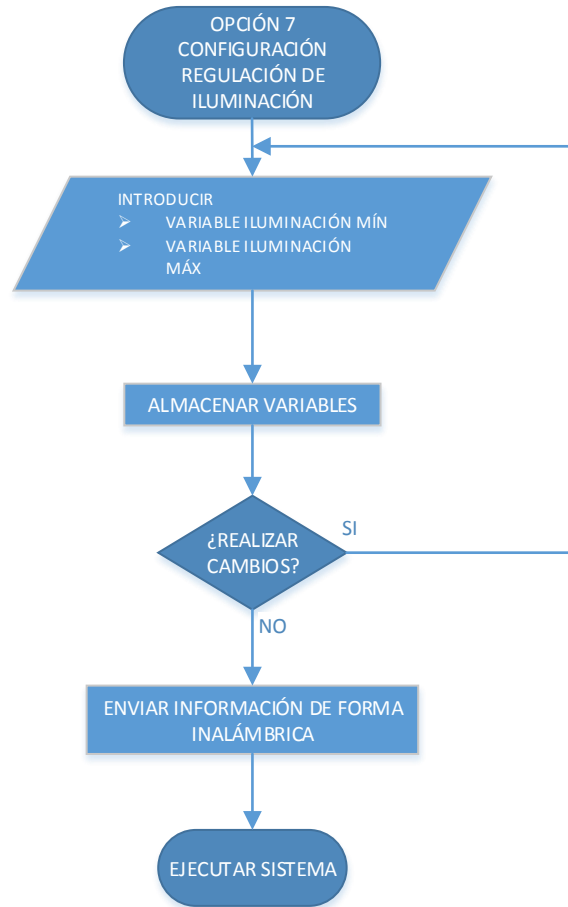


Figura 4.9. Configuración de iluminación

4.3.15 Configuración alarma despertador

En la actualidad una alarma es un dispositivo de gran ayuda, teniendo como tarea principal el monitoreo de un tiempo determinado, indicando mediante un sonido que puede ser un ruido, música, estación de radio o en ocasiones una luz indicadora para avisar que se ha cumplido el lapso de tiempo decretado. La gran mayoría de los dispositivos electrónicos cuentan con esta herramienta, por mencionar alguno tenemos un teléfono móvil que contiene esta función de despertador que incluye la repetición de la alarma cada cierto tiempo, en caso de que el usuario se vuelva a dormir, desechando la posibilidad de llegar tarde a sus labores cotidianas, por estas razones es importante contar con un sistema de alarma despertador en el sensor inteligente.

4.3.16 Algoritmo de alarma despertador

La figura 4.10 se observa el algoritmo de la alarma despertador que ofrece el sensor inteligente, teniendo como primera tarea la de introducir una hora (HH), minutos (MM), una fecha en formato de días (DD), meses (MM) y años (AAAA), para que posteriormente sean almacenados en memoria, cuenta también con la opción de poder repetir la alarma de forma diaria, semanal o mensual, si no se desea esta opción se desecha y se llega a la fase de realizar cambios, de necesitar alguno, se retorna a la introducción de hora y fecha, en caso contrario se ejecuta el sistema.

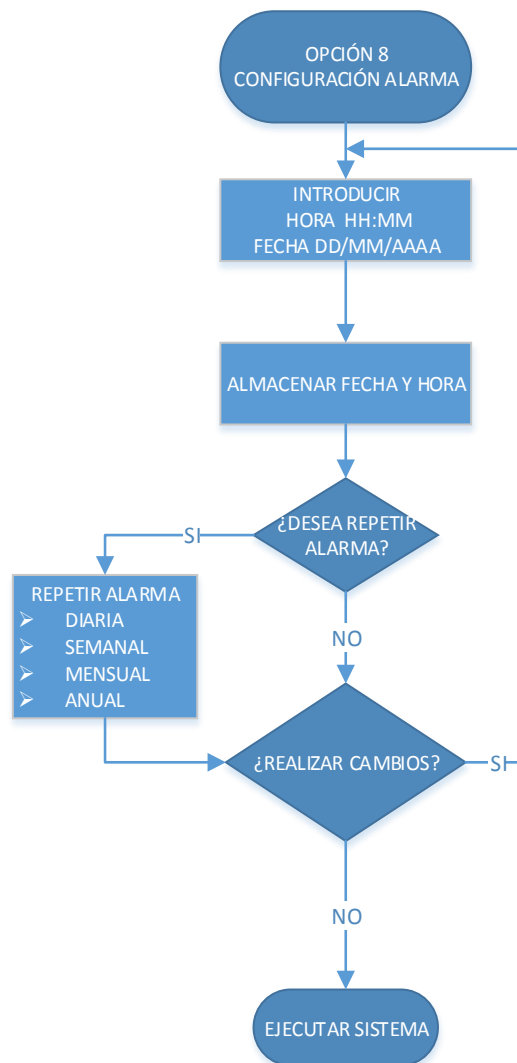


Figura 4.10. Configuración alarma despertador

4.4 Diagrama de flujo general de la monitorización del sensor inteligente

La figura 4.11 se observa el diagrama de flujo general de monitorización del sensor inteligente, en el cual se analiza las opciones de monitorización que entrega el sistema, en seguida se menciona cada una de estas:

- Opción 1 “Fecha y hora”
- Opción 2 “Temporizador”
- Opción 3 “Ahorro de energía”
- Opción 4 “Prevención de incendios”
- Opción 5 “Temperatura”
- Opción 6 “Sistema de seguridad”
- Opción 7 “Control de iluminación”
- Opción 8 “Alarma despertador”
- Sin cambios.

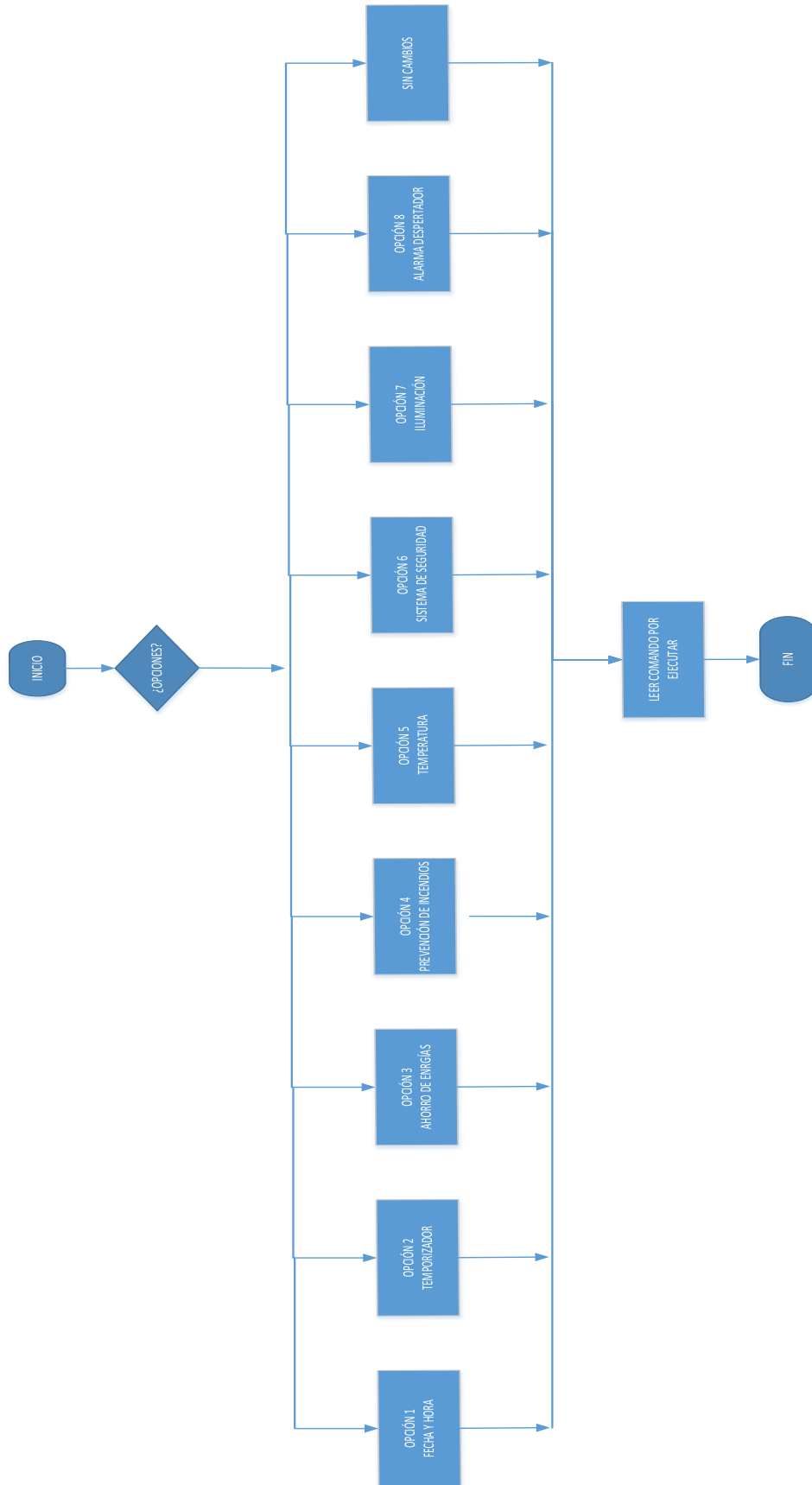


Figura 4.11. Diagrama de flujo general de monitorización

En seguida se describe cada uno de los algoritmos que componen la monitorización del sensor inteligente.

4.4.1 Algoritmo del reloj-calendario sensor inteligente

La figura 4.12 se puede observar el diagrama a flujo del reloj-calendario, teniendo como primera tarea la de iniciar la comunicación entre el dispositivo DS1307 (Reloj de Tiempo Real) y el microcontrolador por medio de la comunicación I2C. Una vez realizada la conexión, si fuera el caso de iniciarse por primera vez el dispositivo DS1307 se cargara una hora y fecha programadas por default que serán las siguientes, hora: 12:00 am y fecha de: 01/01/2016, en caso contrario de tener una ya programada se leerán las variables de tiempo y fecha otorgadas por el DS1307.

De existir alguna modificación se realizara una interrupción al sistema para ir a la etapa de configuración del sensor inteligente en la opción 1 “configuración de fecha y hora”, se efectuará los cambios pertinentes para después ejecutarse dando como resultado la fecha y hora actualizadas, en caso contrario el sistema estará en estado de monitorización esperando una modificación, de no presentarse permanecerá en fase de monitorización.

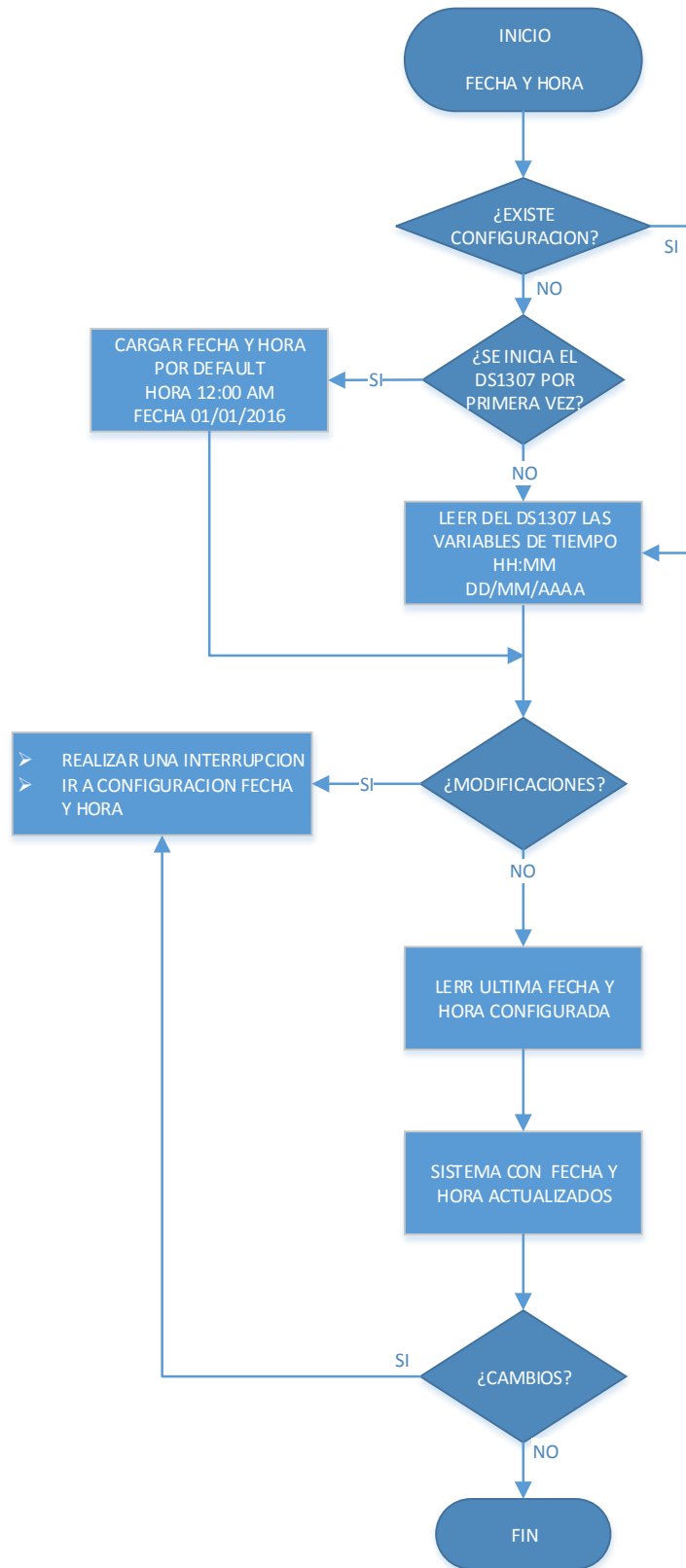


Figura 4.12. Algoritmo del reloj-calendario

4.4.2 Algoritmo de temporizador

El algoritmo del temporizador de la figura 4.13 inicia con la fase de leer las variables de *int tiempo* (variable de tiempo) y la variable *int temporizador* (variable de temporizador) siendo la encargada de indicar el tiempo que durar la temporización.

Si llegara a existir una interrupción al temporizador se realiza una pausa en el tiempo de monitorización y el programa se va a la fase de configuraciones del sensor inteligente opción 2 “Configuración temporizador” permaneciendo en esa opción hasta que el usuario vuelva reiniciar la temporización o realice un cambio al tiempo programado para posteriormente retornar a ejecutarlo, en caso contrario continua con la temporización hasta llegar a la igualdad de, temporización es igual al tiempo configurado ($int\ tiempo == int\ temporizador$), de no realizarlo sigue con el temporizador hasta que se cumpla esa igualdad, una vez confirmada la identidad, el programa tiene la tarea la activación de una alarma que durara 1 minuto y finalizara con el tiempo configurado del temporizador.

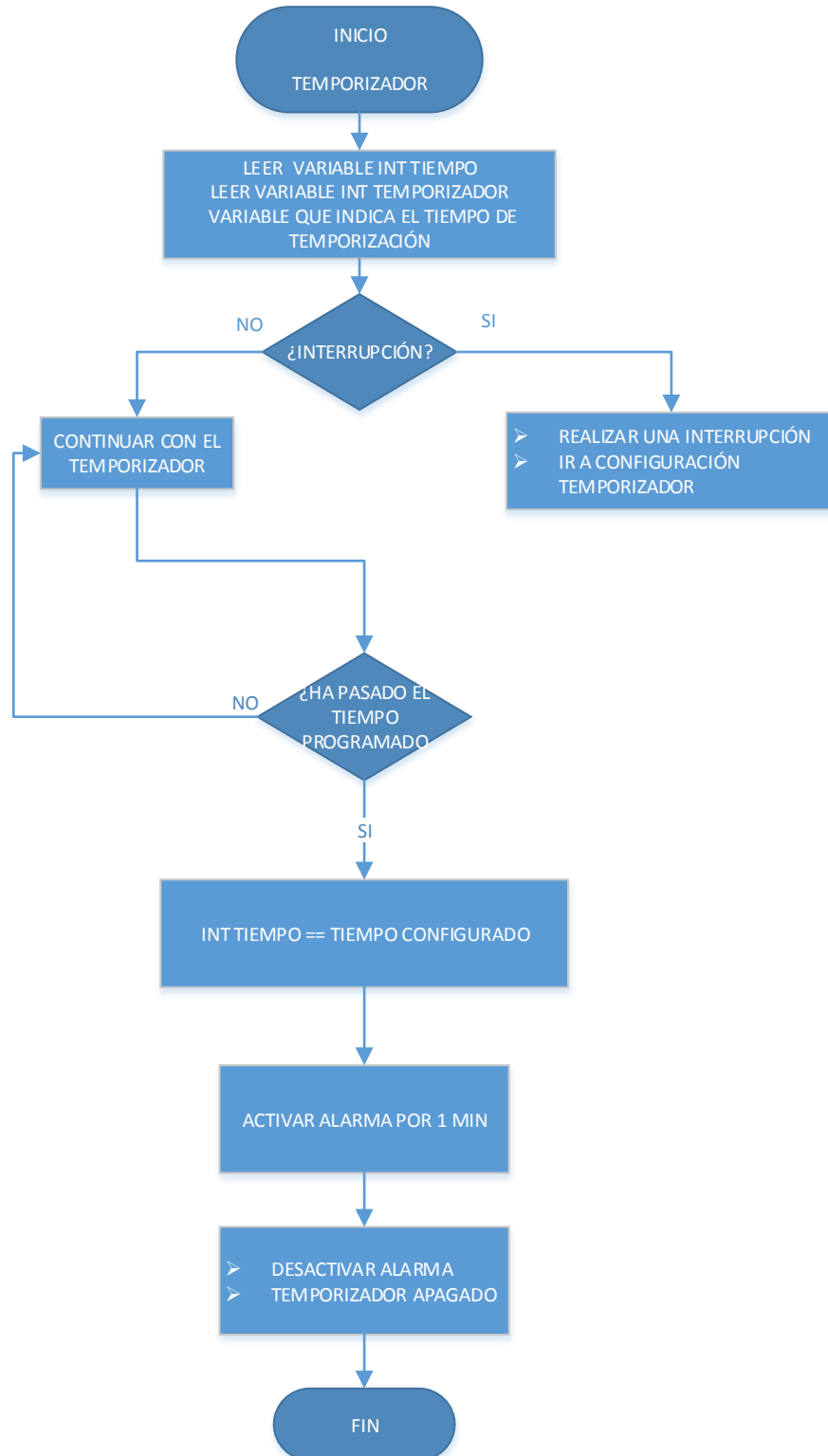


Figura 4.13. Algoritmo de temporizador.

4.4.3 Algoritmo de ahorro de energía

La figura 4.14 se observa el algoritmo de ahorro de energía en la fase ejecución del sistema, el cual tiene como primera tarea leer la información guardada en la memoria EEPROM que encuentra en el microcontrolador. Posteriormente se procede a ejecutar el sistema, en caso contrario de no existir algún dato se pone todos los sensores en estado activo hasta que se inicie una interrupción externa para su respectiva selección de sensores. Esta opción es generada en la opción 3, configuraciones del sensor inteligente “Ahorro de energía”. A continuación se describe cada una de las variables de los sensores:

- *Activar_tem*: Selección del sensor de temperatura.
- *Activar_mov*: Selección del sensor de movimiento.
- *Activar_ilum*: Selección del sensor de iluminación.
- *Activar_humo*: Selección del sensor de humo y gas.

Una vez seleccionada la variable, se tiene las opciones de activación o desactivación del sensor.

1. Si la variable *activar_tem* = 1 el sensor accede al estado de activación, en caso contrario si *activar_tem* = 0 el sensor permanece en estado desactivo.
2. Si la variable *activar_mov* = 1 el sensor accede al estado de activación en caso contrario si *activar_mov* = 0 el sensor permanece en estado desactivo.
3. Si la variable *activar_ilum* = 1 el sensor accede al estado de activación en caso contrario si *activar_ilum* = 0 el sensor permanece en estado desactivo.
4. Si la variable *activar_humo* = 1 el sensor accede al estado de activación en caso contrario si *activar_humo* = 0 el sensor permanece en estado desactivo.

Si se requiere un cambio en la configuración de ahorro de energía, se realizara una interrupción a la ejecución del sistema dando como resultado ir a la parte de configuración del sensor inteligente en la cual se podrá seleccionar otro sensor para su activación o desactivación. En caso contrario el proceso no sufrirá cambio alguno.

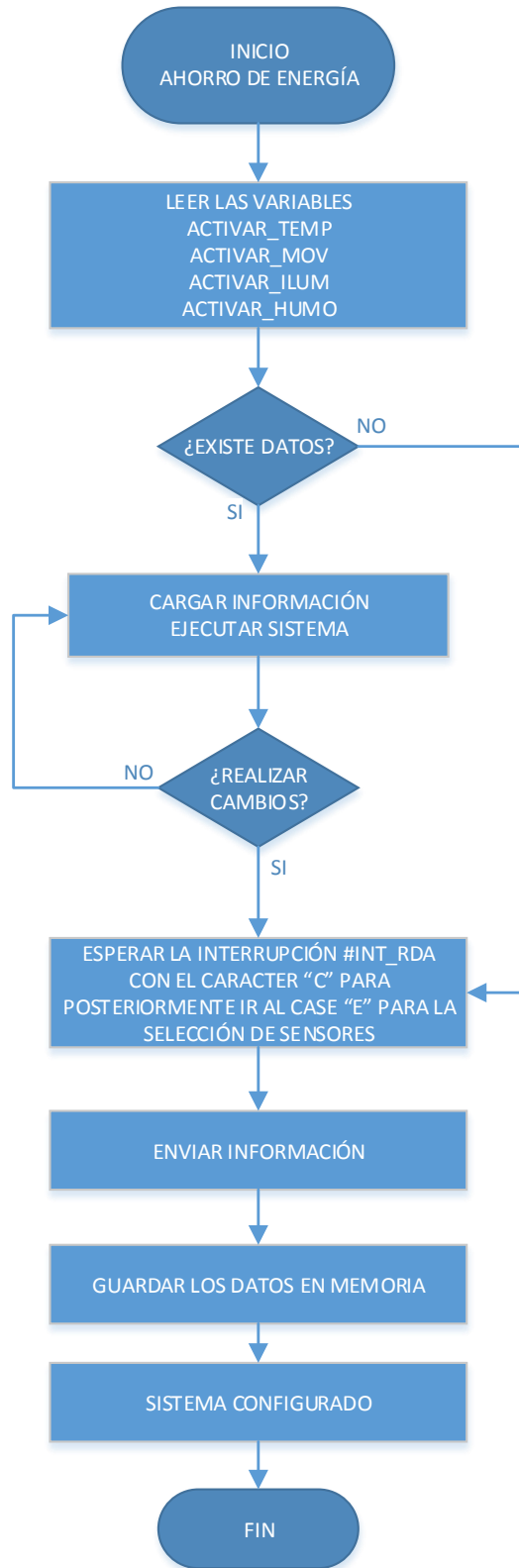


Figura 4.14. Algoritmo ahorro de energía.

4.4.4 Algoritmo de prevención de incendios

La figura 4.15 se observa el algoritmo de prevención de incendios en la fase ejecución del sistema, el cual tiene como primera tarea leer la información guardada en la memoria EEPROM que se encuentra en el microcontrolador. Posteriormente se procede a ejecutar el sistema, en caso contrario de no existir algún dato se espera una interrupción al sistema para su respectiva configuración.

Una vez inicializada la monitorización, el sensor de humo detecta los parámetros que se encuentran en el ambiente, como se trata de un sensor analógico se realiza su conversión analógico digital para que posteriormente sea guardada en una variable con el nombre de *int humo*. Existen dos variables de control llamadas *int hmind* (humo mínimo) e *int hmaxd* (humo máximo) que son los umbrales para el seguimiento del sistema, se realiza un etapa de control con la instrucción *if* que tiene como tarea la de comparar los valores obtenidos del sistema con los valores configurados en la fase de configuración “prevención de incendios”. En el caso que la variable *int humo* sea mayor a las variables *int hmind* e *int hmaxd* se activara una alarma para alertar la presencia de humo o gas. Toda la información obtenida se guarda en memoria para tener una base de datos de cómo se comporta el sistema y se podrá visualizar en el Screen de monitorización que cuenta la app.

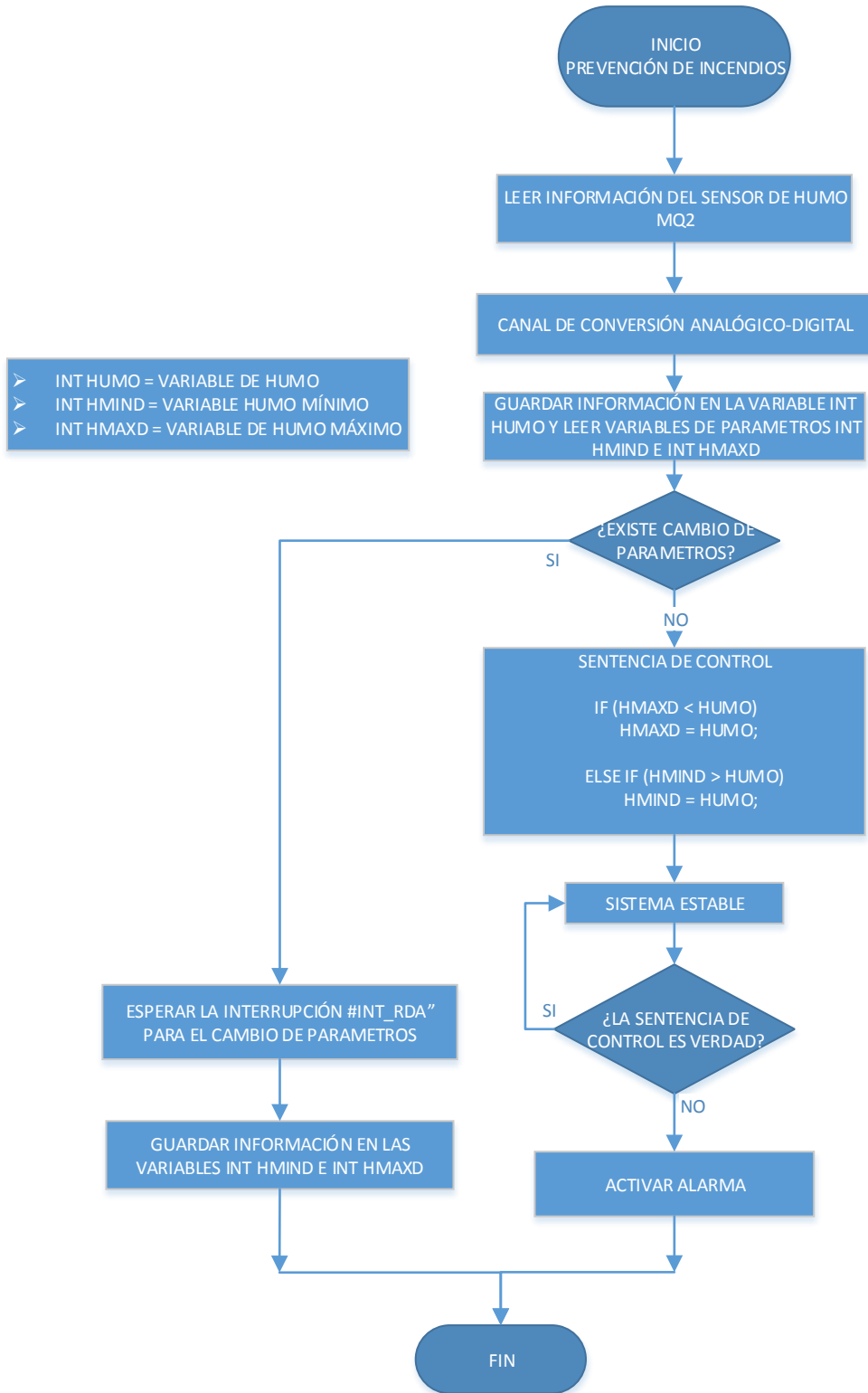


Figura 4.15. Algoritmo prevención de incendios.

4.4.5 Algoritmo de temperatura

La figura 4.16 se observa el algoritmo de temperatura en la fase ejecución del sistema, el cual tiene como primera tarea leer la información guardada en la memoria EEPROM que se encuentra en el microcontrolador. Posteriormente se procede a ejecutar el sistema, en caso contrario de no existir algún dato se espera una interrupción al sistema para su respectiva configuración.

Una vez obtenida la información se comienza con la monitorización que tiene como primera tarea leer los datos obtenidos por el sensor de temperatura LM35 para que posteriormente se empiece la conversión analógica digital (ADC). Una vez digitalizada la información se guarda en la variable *int temp* (variable de temperatura), se cuenta con dos variables de control *int tmind* (temperatura mínima de detección) y *int tmaxd* (temperatura máxima de detección) que son los umbrales para el seguimiento del sistema. Existe una etapa de control con la instrucción *if* que es la encargada de realizar la comparación entre los valores obtenidos del sistema con los valores configurados en la fase de configuración “temperatura”. En el caso que la variable *int temp* sea mayor a las variables *int tmind* e *int tmaxd* se activara una alarma para alertar que los umbrales han sido rebasados en su totalidad. Toda la información obtenida se guarda en la memoria para tener una base de datos de cómo se comporta el sistema y se podrá visualizar en el Screen de monitorización que cuenta la app.

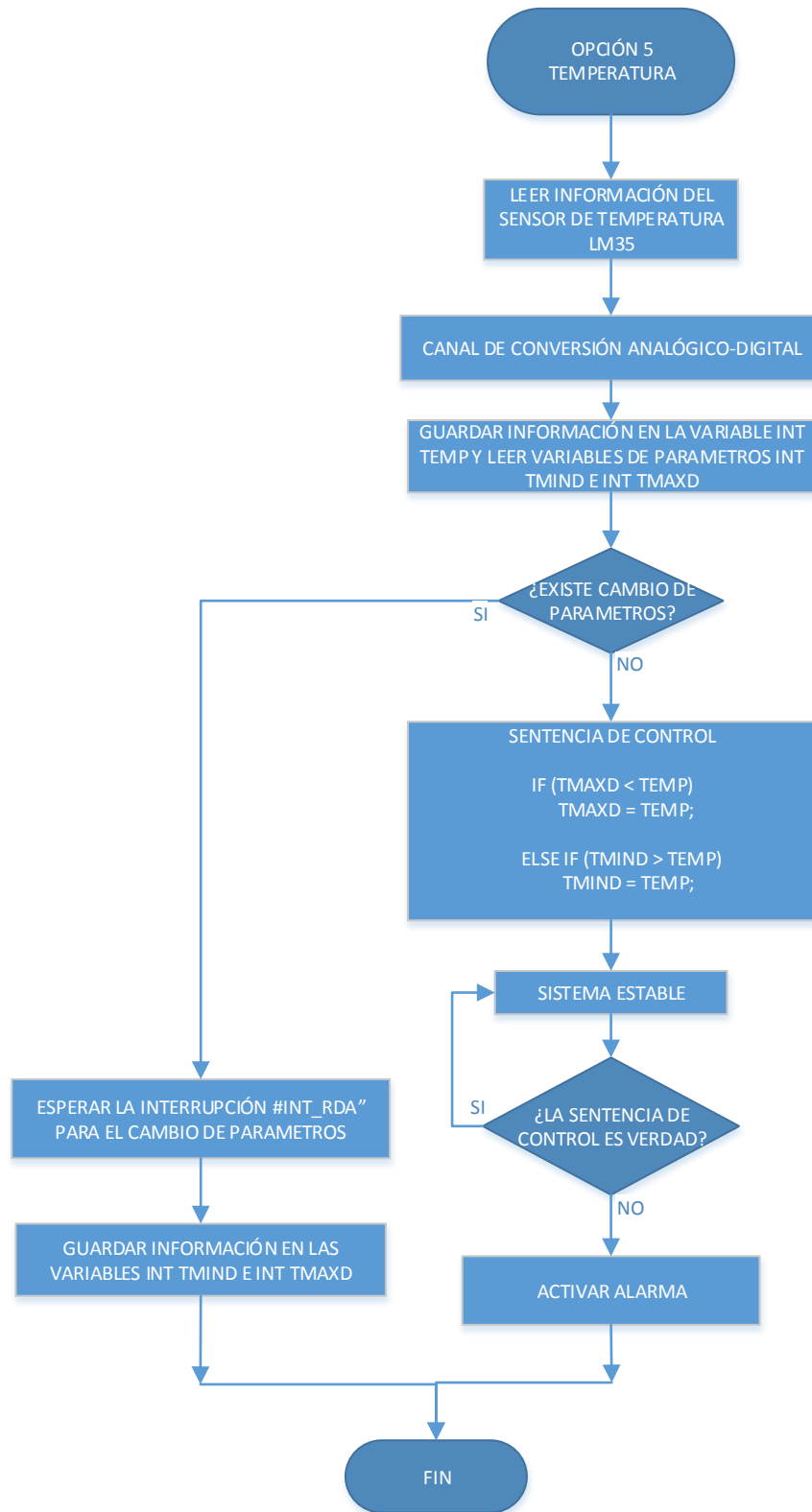


Figura 4.16. Algoritmo de temperatura.

4.4.6 Algoritmo de sistema de seguridad

La figura 4.17 se observa el algoritmo de sistema de seguridad en la fase ejecución, el cual tiene como primera tarea leer la información guardada en la memoria EEPROM que se encuentra en el microcontrolador. Posteriormente se procede a ejecutar el sistema, en caso contrario de no existir algún dato se espera una interrupción al sistema para su respectiva configuración.

Una vez obtenida la información se comienza con la monitorización que tiene como primera tarea leer los datos obtenidos por el sensor de movimiento HC SR501 para que posteriormente se guarde la información en la variable *int mov*. Como se trata de un sensor digital no es necesario el proceso de digitalización y su proceso de monitorización comienza en el estado 0 (desactivo), el algoritmo espera un cambio de estado de la variable *int mov* entrando en el estado 1 (activación) o estado de notificación de intrusos, de no existir dicha cambio el sistema se encuentra en la condición de monitorización hasta generar el cambio de domino de desactivo a activo.

Si la variable *int mov* pasa al estado activo (1) se accionara una alarma sonora para indicar de la una posible amenaza, posteriormente la información se guarda en la memoria EEPROM del microcontrolador para generar una base de datos para llevar un control del sistema.

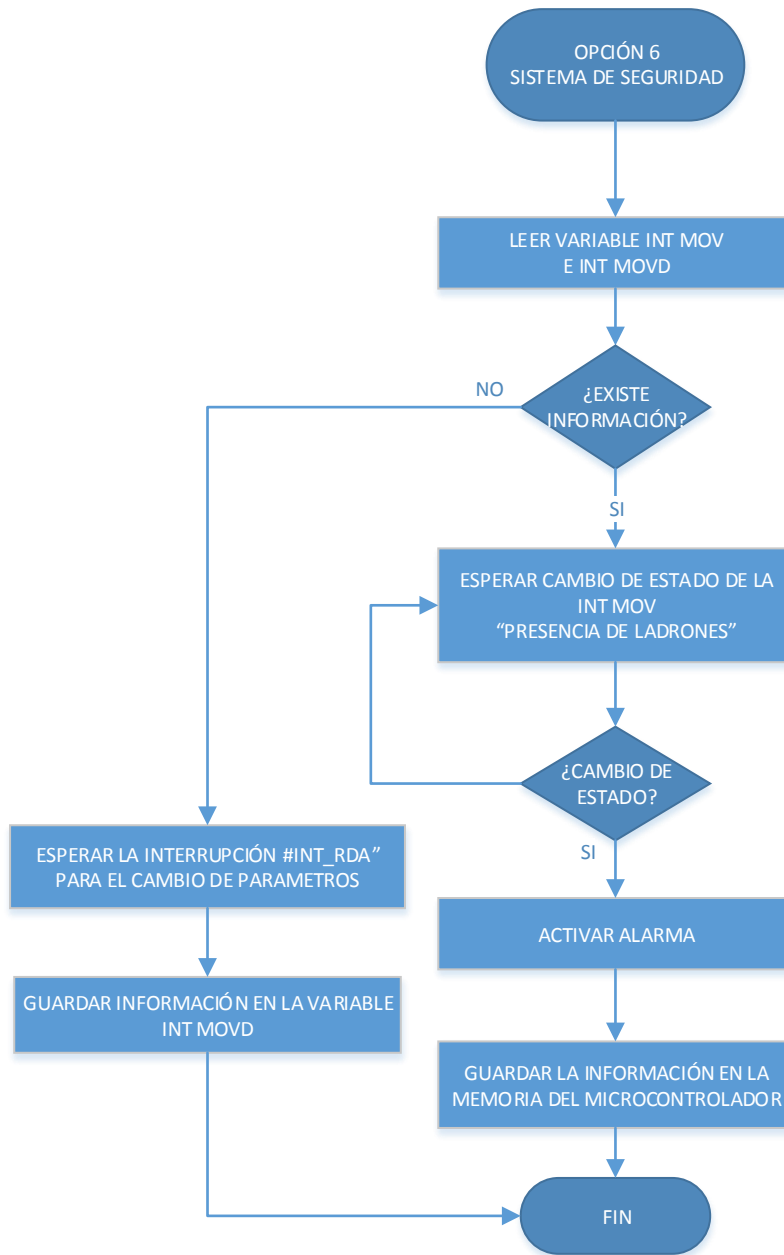


Figura 4.17. Algoritmo sistema de seguridad.

4.4.7 Algoritmo de regulación de iluminación

La figura 4.18 contiene el algoritmo de regulación de iluminación, teniendo como primera tarea la de leer las variables *int ilumon*, *int ilumin* e *ilumax* (variables del sensor de iluminación) generada por el sensor TSL2561.

El sensor de iluminación entra en el estado de monitorización y es controlado por dos variables *int ilumin* (variable de iluminación mínima) e *int ilumax* (variable de iluminación máxima) configuradas en la etapa de configuración de regulación de iluminación y encargadas de vigilar cómo se comporta el proceso.

Las variables *int ilumin* e *int ilumax* tienen como tarea principal la de controlar el sistema, teniendo un dominio en la iluminación a continuación se describe cada uno de estos:

- *Int ilumon <= int ilumin*: La variable de iluminación (*int ilumon*) se encuentra en el estado de monitorización, de existir un decremento en la luminosidad, el sistema realiza la siguiente identidad *int ilumon <= int ilumin* de ser verdadero, el sistema activa una alarma preventiva para indicar que la luminosidad ha decrementado lo suficiente para disminuir la variable de control, de no encontrarse esta identidad el sistema sigue en estado de monitorizaron hasta que determine que exista la identidad de control.
- *Int ilumon => int ilumin*: La variable de iluminación (*int ilumon*) se encuentra en el estado de monitorización, de existir un incremento en la luminosidad, el sistema realiza la siguiente identidad *int ilumon => int ilumax* de ser verdadero, el sistema activa una alarma preventiva para indicar que la luminosidad ha incrementado lo suficiente para exceder la variable de control, de no encontrarse esta identidad el sistema sigue en estado de monitorizaron hasta que determine que exista la identidad de control.

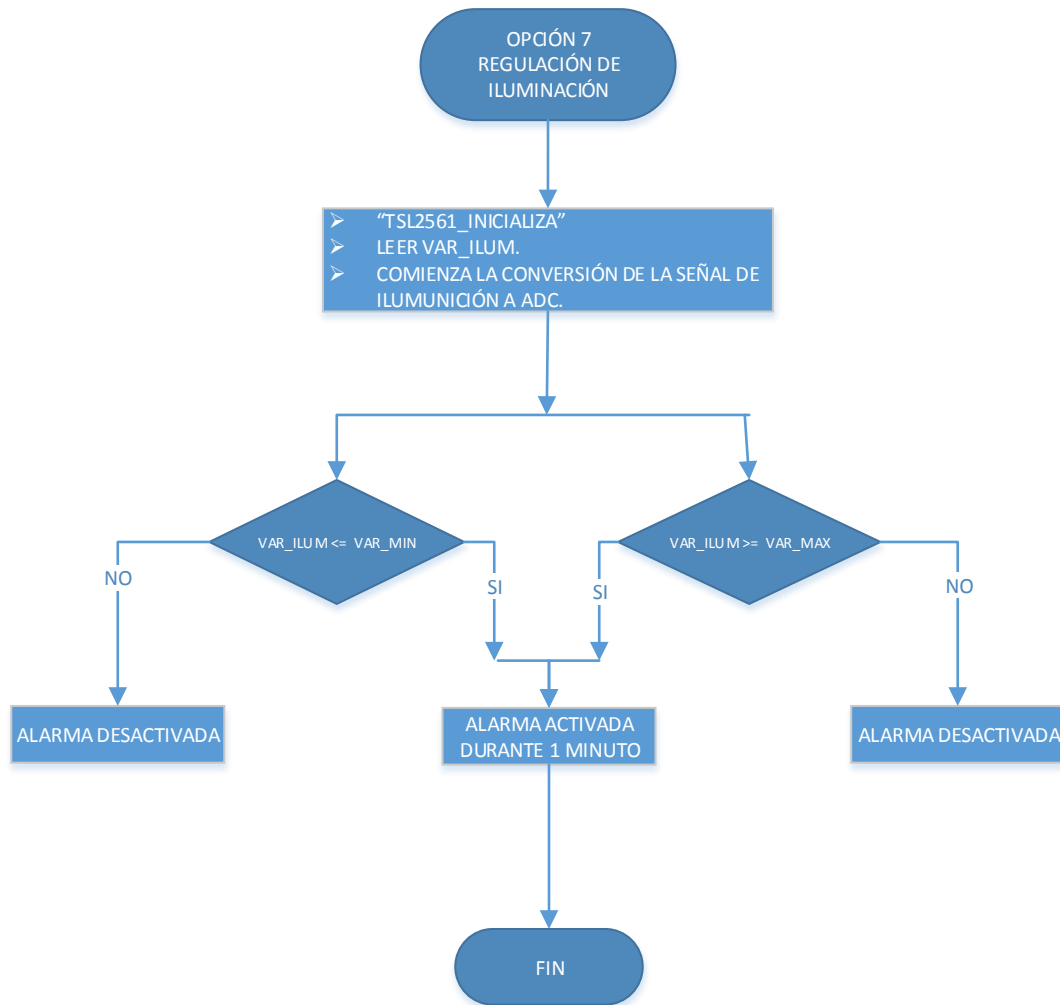


Figura 4.18. Algoritmo regulación de iluminación.

4.4.8 Algoritmo de alarma

La figura 4.19 se observa el algoritmo de la alarma en su fase de ejecución, teniendo como primera tarea leer la información de la memoria EEPROM del microcontrolador, dicha información es la siguiente: variables de tiempo *int hora* (hora), *int min* (minutos) y variables de fecha: *int día* (días), *int mes* (mes) y *int anio* (año), estos parámetros son configurados en la opción 1 fecha y hora. De no existir información se espera a que se realice una interrupción #INT RDA para la configuración del tiempo y fecha.

El algoritmo posee variables de control que son las encargadas de notificar que se cumpla adecuadamente con los tiempos programados por el usuario, a continuación se describe cada una de ellas:

- *Int horad*: Encargada de la configuración de la hora.
- *Int mind*: Encargada de la configuración de los minutos.
- *Int diad*: Encargada de la configuración de los días.
- *Int mesd*: Encargada de la configuración de los meses.
- *Int aniod*: Encargada de la configuración de los años.

La hora y fecha de la alarma despertador son seleccionadas en la opción 8 (configuración de alarma), para que posteriormente se cumplan las siguientes condiciones:

- *Int hora == int horad*
- *Int min == int mind*
- *Int día == int diad*
- *Int mes == int mesd*
- *Int anio == int aniod*

El dispositivo DS1307 comienza su fase de conteo hasta llegar a las identidades mencionadas, una vez ejecutadas todas las igualdades, el microcontrolador tiene la tarea de activar una alarma, encargada de notificar que ha transcurrido el tiempo programado, esta durará 1 min, para que posteriormente se desactive, de existir la configuración de repetición, se retorna con el proceso, en caso contrario se termina con el algoritmo.

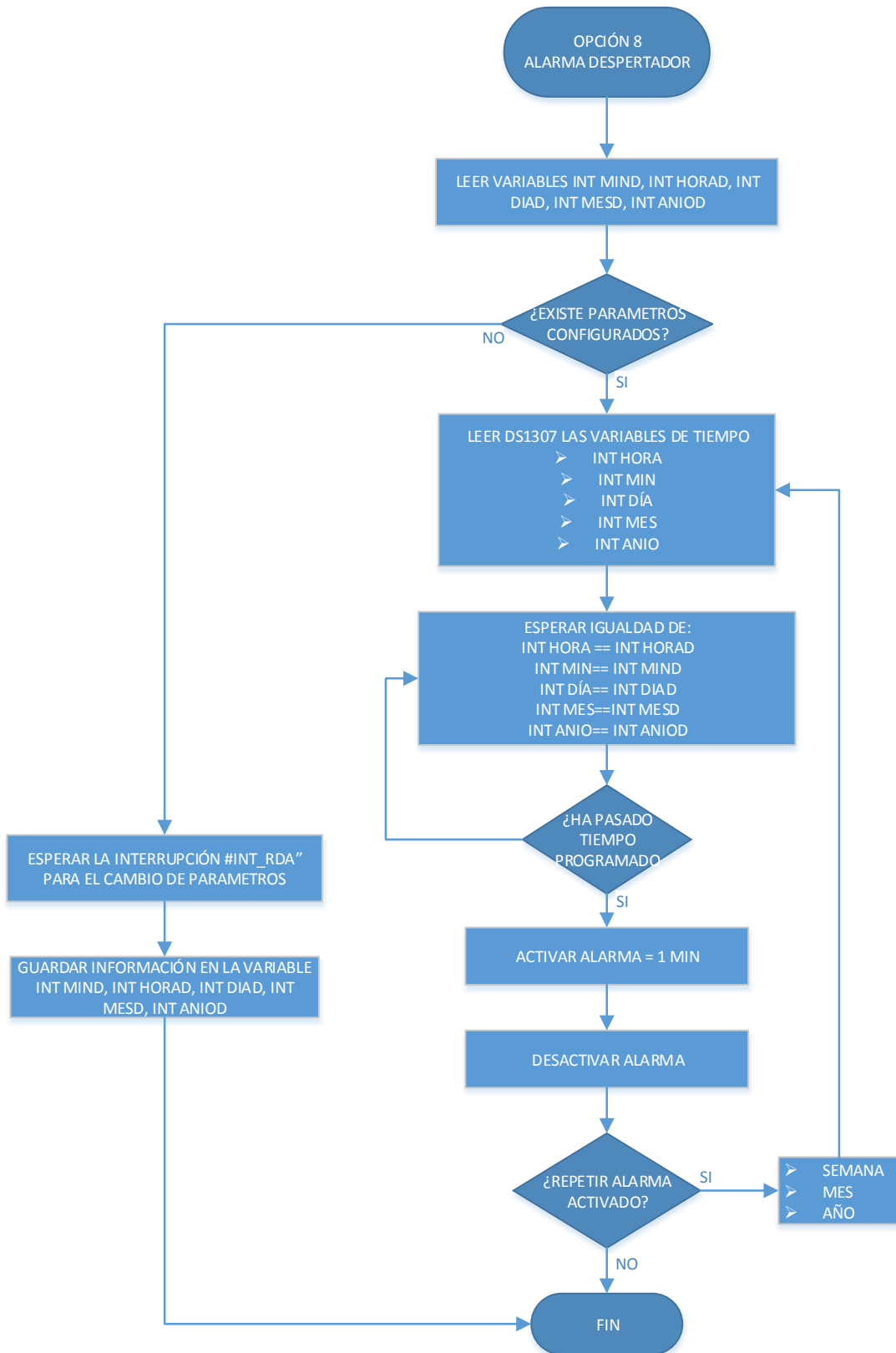


Figura 4.19. Algoritmo alarma

4.5 Programación del sensor inteligente

La programación del sensor inteligente se realizó mediante el compilador C de CCS versión 5 que ha sido desarrollado específicamente para PIC MCU, obteniendo la máxima optimización del compilador con estos dispositivos. Dispone de una amplia biblioteca de funciones predefinidas. Además, suministra los controladores (drivers) para diversos dispositivos como LCD, convertidores ADC, relojes en tiempo real, EEPROM serie, etc.

A grandes rasgos se puede decir que un compilador convierte el lenguaje de alto nivel a instrucciones en código máquina; un cross-compiler es un compilador que funciona en un procesador (normalmente en una PC) diferente al procesador objeto. El compilador CCS C es un cross-compiler. Los programas son editados y compilados a instrucciones máquina en el entorno de trabajo de la PC, el código máquina puede ser cargado de la PC al sistema mediante ICD2 (o mediante cualquier programador) y puede ser depurado (puntos de ruptura, paso a paso, etc.) desde el entorno de trabajo del PC [11].

El CCS C es C estándar y, además de las directivas estándar (*#include*, etc.), suministra unas directivas específicas para PIC (*#device*, etc.); además incluye funciones específicas (*bit_set()*, etc.). Se suministra con un editor que permite controlar la sintaxis del programa.

En la figura 4.20 se observa parte del código del sensor inteligente que se desarrolló en PIC C Compiler y a continuación se describe cada una de sus fases:

1. Tipo de microcontrolador a utilizar, en este caso es el PIC18F4550.
2. Fusibles como watchdog, el pin de reset, el conjunto de instrucciones, etc.
3. Comunicación I2C, pin para enviar datos B0, pin de señal de reloj B1.
4. Directiva del reloj de tiempo real DS1307.
5. Interrupción generada cuando se va a configurar el sensor inteligente.

```

1
2 #include <18F4550.h> ← 1
3 #device adc=10
4 #fuses INTHS,NOWDT,NOPROTECT,NOLVP,NOPUT ← 2
5 #use delay (clock=8MHz)
6 #use RS232 (BAUD=9600, XMIT=PIN_C6, RCV=PIN_C7, parity=N)
7 #use i2c (Master,sda=PIN_B0,scl=PIN_B1) ← 3
8 #use standard_io(B)
9 #include <DS1307.c> ← 4
10 #include <TSL256X.c>
11 #include <Sensor_inteligente.c>
12
13 byte buffer;
14 unsigned INT a_mov,a_temp,a_humo,a_lux,a_zumb;
15 unsigned INT led_mov,led_temp,led_humo,led_lux;
16 unsigned INT temperatura,temperatura_min,temperatura_max,humo,humo_min,humo_max,hr,min,sec,dia,sem,mes,anio;
17 unsigned INT temperatura_min_p,temperatura_max_p,humo_min_p,humo_max_p;
18 unsigned LONG lectura0,lectura1,luxes,movimiento,luxes_min,luxes_max;
19 unsigned LONG luxes_min_p,luxes_max_p,movp;
20
21 #INT_rda ← 5
22 VOID serial_isr ()
23 {
24     buffer = getc ();
25 }
26
27 #INT_TIMER0
28
29 VOID timer0_isr ()
30 {
31     delay_ms (20);
32
33     IF (a_mov == 1)

```

Figura 4.20 Código del sensor inteligente

En la figura 4.21 se observa parte del código del sensor inteligente que se desarrolló en PIC C Compiler y a continuación se describe cada una de sus fases:

1. Rutina de la monitorización del sistema encargada de leer toda la información recabada por los sensores por un lapso de tiempo de cada segundo, siendo las variables de: temperatura, iluminación, humo y movimiento.
2. Línea de código encargada de leer los valores del reloj de tiempo real DS1307.
3. Reinicia la subrutina de la monitorización del sistema en caso que el programa tuviera problemas, para que posteriormente inicie leyendo los datos de iluminación, humo, temperatura y movimiento.

```

}
#INT_EXT2 ← 1
VOID ext2_isr ()
{
  if (a_lux==1){
    READ_TSL256X (lectura0, lectura1) ;
    luxes = CalculateLux (TSL2561_GAIN_1X, TSL2561_INTEGRATIONTIME_402MS, lectura0, lectura1, TSL2561_PACKAGE_T) ;
  }
  else{
    luxes=0;}
  humo = leer_humo (a_humo) ;
  temperatura = leer_temperatura (a_temp) ;
  comparacion_INT (temperatura_max, temperatura_min, temperatura, a_temp) ;
  comparacion_INT (humo_max, humo_min, humo, a_humo) ;
  comparacion_LONG (luxes_max, luxes_min, luxes, a_lux) ; ← 2
  lectura_reloj (hr, min, sec, dia, Sem, mes, anio) ;
  memoria_variables_escritura (DETECCION, luxes_min, luxes_max, temperatura_min, temperatura_max, humo_min, humo_max, movimiento) ;
  edo_foco_long (luxes_max_p, luxes_min_p, luxes, a_lux, led_lux) ;
  edo_foco_int (temperatura_max_p, temperatura_min_p, temperatura, a_temp, led_temp) ;
  edo_foco_int (humo_max_p, humo_min_p, humo, a_humo, led_humo) ;
}
VOID Reinicio_detecciones () ← 3
{
  READ_TSL256X (lectura0, lectura1) ;
  luxes = CalculateLux (TSL2561_GAIN_1X, TSL2561_INTEGRATIONTIME_402MS, lectura0, lectura1, TSL2561_PACKAGE_T) ;
  humo = leer_humo (1) ;
  temperatura = leer_temperatura (1) ;
  luxes_max = luxes;
  luxes_min = luxes;
  temperatura_max = temperatura;
}

```

Figura 4.21. Código del sensor inteligente

4.6 Software para la aplicación móvil

Los dispositivos móviles tales como teléfonos inteligentes y tabletas son de uso común por la sociedad, la mayoría de ellos se programan de forma tradicional teniendo un sistema abierto para el desarrollo de aplicaciones en un sistema operativo llamado Android, que en la actualidad es uno de los más famosos a nivel mundial con una enorme cuota de mercado.

Desarrollar software para esta plataforma permite llegar a una gran cantidad de público. Sin embargo, abordar un desarrollo directo con el lenguaje de programación Java, XML, emulación virtual, puede ser intimidante por esa razón Google ideó una forma sencilla de desarrollar aplicaciones para Android: haciendo uso de programación gráfica y atraer así a los nuevos desarrolladores.

De igual manera App inventor es una herramienta útil de programación, con la ventaja adicional de permitir el desarrollo de aplicaciones para dispositivos móviles que se usen el sistema operativo Android. Una característica interesante es que el desarrollador de las aplicaciones es en la Web; aunque es necesario instalar un módulo de software en la computadora, en el momento del desarrollo se ejecuta la última versión de App Inventor disponible en su sitio web y los proyectos se guardan en línea.

El App Inventor consta de dos segmentos principales: un módulo Web y el editor de bloques de Android. El modulo en Web de la figura 4.22 es la interfaz donde se tendrá acceso a los proyecto en el cual podemos añadir los componentes y configurarlos apropiadamente, si se trata de componentes visuales entonces se define el diseño de la interfaz para la app. En la figura 4.23 se observa la interfaz de editor de bloques donde se ven todos los elementos de programación para la aplicación deseada.



Figura 4.22. Interfaz de App Inventor 2

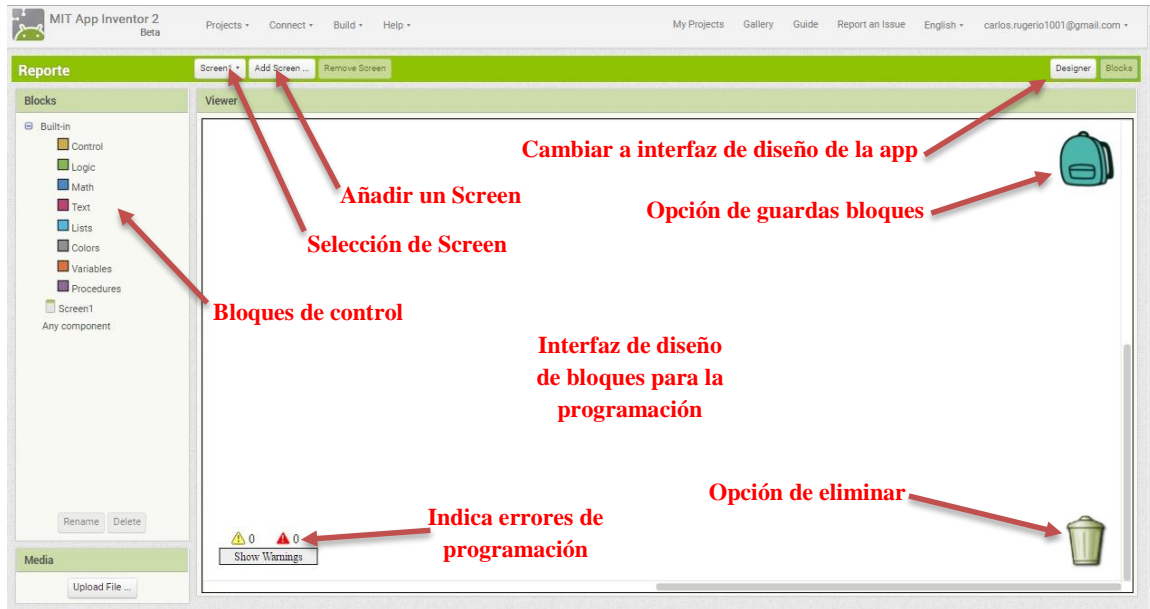


Figura 4.23. Interfaz de bloques de App Inventor 2 para la programación de la app.

4.6.1 Diseño de la aplicación

En la figura 4.24 se muestra el diseño de la aplicación con las configuraciones que ofrece el sensor inteligente y en la figura 4.25 se observa la configuración de hora y fecha, siendo en total 7 opciones para configurar al sistema como se desee. A continuación se describe la interfaz de la aplicación:

1. Botón conectar: Su función es la de conectar la aplicación de forma inalámbrica vía Bluetooth al sensor inteligente encargado de la monitorización del sistema. Es importante mencionar para que exista dicha conexión el modulo Bluetooth HC-05 debe de estar agregado a los dispositivos de conexión Bluetooth del celular.
2. Botón de actualizar datos: Su función es ejecutar la última configuración del sensor inteligente.
3. Casilla de activación y desactivación del sensor de iluminación. Esta casilla es la encargada de activar o desactivar el sensor de iluminación, si se requiere que el sensor este activo se muestra una viñeta en la casilla, de lo contrario estará vacía.
4. Casilla de iluminación mínima: La función de esta casilla es introducir la constante de iluminación mínima que requiera la monitorización del sistema.

5. Casilla de iluminación máxima: La función de esta casilla es introducir la constante de iluminación máxima que requiera la monitorización del sistema.
6. Casilla de activación y desactivación del sensor de temperatura. Esta casilla es la encargada de activar o desactivar el sensor de temperatura, si se requiere que el sensor este activo se muestra una viñeta en la casilla, de lo contrario estará vacía.
7. Casilla de temperatura mínima: La función de esta casilla es introducir la constante de iluminación mínima que requiera la monitorización del sistema.
8. Casilla de temperatura máxima: La función de esta casilla es introducir la constante de temperatura máxima que requiera la monitorización del sistema.
9. Casilla de activación y desactivación del sensor de humo y gas. Esta casilla es la encargada de activar o desactivar el sensor de humo y gas, si se requiere que el sensor este activo se muestra una viñeta en la casilla, de lo contrario estará vacía.
10. Casilla de humo o gas mínimo: La función de esta casilla es introducir la constante de humo o gas mínimo que requiera la monitorización del sistema.
11. Casilla de humo o gas máximo: La función de esta casilla es introducir la constante de humo o gas máximo que requiera la monitorización del sistema.
12. Casilla de activación y desactivación del sensor de movimiento. Esta casilla es la encargada de activar o desactivar el sensor de movimiento, si se requiere que el sensor este activo se muestra una viñeta en la casilla, de lo contrario estará vacía.
13. Casilla de activación y desactivación del zumbador: La función de la casilla es la activación y desactivación del zumbador, si se requiere que este activo se muestra una viñeta, de lo contrario estará vacía.
14. Botón guardar parámetros: Botón encargado de guardas los parámetros configurados.
15. Botón actualizar reloj: Botón encargado de actualizar el reloj para él envío de información.
16. Botón hora: Botón encargado de actualizar la hora del sistema.
17. Botón fecha: Botón encargado de actualizar la fecha del sistema.

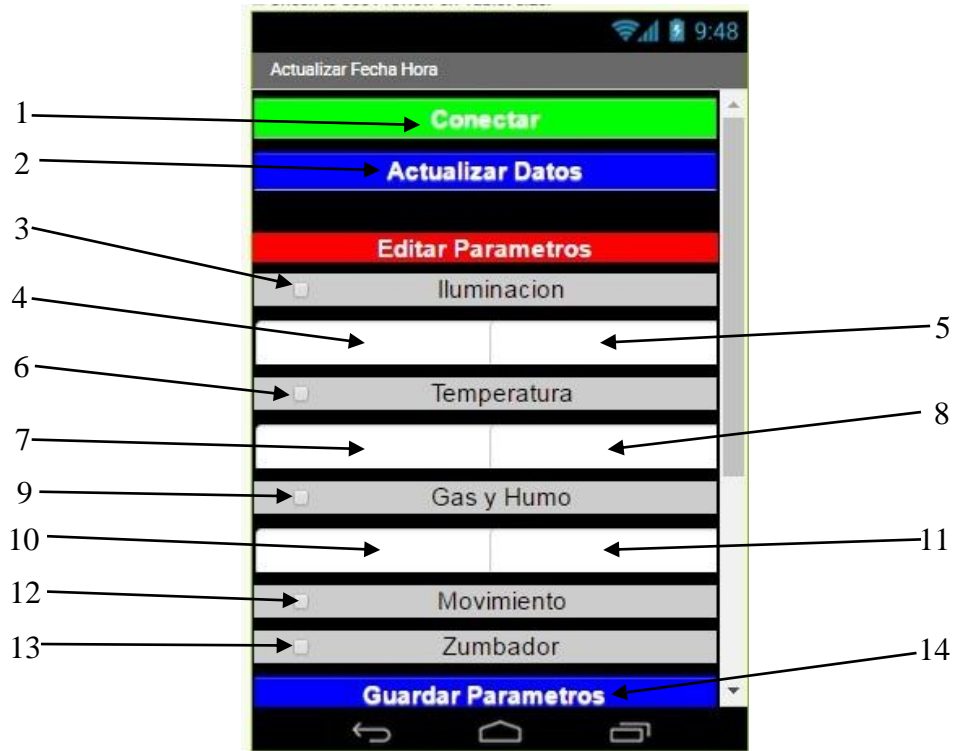


Figura 4.24. Interfaz de la app de configuraciones diseñada en app inventor 2



Figura 4.25. Interfaz de la app de configuraciones diseñada en app inventor 2

Se diseñó un Screen exclusivamente para la monitorización del sistema, el cual otorga toda la información recabada por el sensor inteligente por un lapso de tiempo de 1 segundo. En la figura 4.26 se observa la interfaz de monitorización. A continuación se describe cada una de las casillas encargadas de mostrar los datos recabados por el sistema.

1. Botón de conectar: Botón encargado de la conexión vía Bluetooth con el sensor inteligente.
2. Botón reiniciar: Botón encargado de reiniciar los parámetros de monitorización.
3. Botón iniciar: Botón encargado iniciar la monitorización.
4. Botón para editar parámetros de monitorización: Botón para direccionar al Screen de configuraciones del sensor inteligente.
5. Iluminación máxima: Casilla que indica la iluminación máxima obtenida por el sistema.
6. Iluminación mínima: Casilla que indica la iluminación mínima obtenida por el sistema.
7. Iluminación actual: Casilla que indica la iluminación actual que tiene el sistema.
8. Temperatura máxima: Casilla que indica la temperatura máxima obtenida por el sistema.
9. Temperatura mínima: Casilla que indica la temperatura mínima obtenida por el sistema.
10. Temperatura actual: Casilla que indica temperatura actual monitorizada por el sistema.
11. Gas mínimo: Casilla que indica la concentración de gas mínima obtenida por el sistema.
12. Gas máximo: Casilla que indica la concentración de gas máxima obtenida por el sistema.
13. Gas actual: Casilla que indica la concentración de gas actual obtenida por el sistema.
14. Detección de movimiento: Casilla encargada de indicar la detección de presencias.
15. MM: Casilla que indica los minutos del sistema.
16. HH: Casilla que indica la hora del sistema.
17. DD: Casilla que indica el día del sistema.
18. AAAA: Casilla que indica el año del sistema.
19. MM: Casilla que indica el mes del sistema.
20. Botón desconectar: Botón que desconecta la comunicación Bluetooth.
21. Botón salir de la aplicación: Botón que cierra la aplicación.

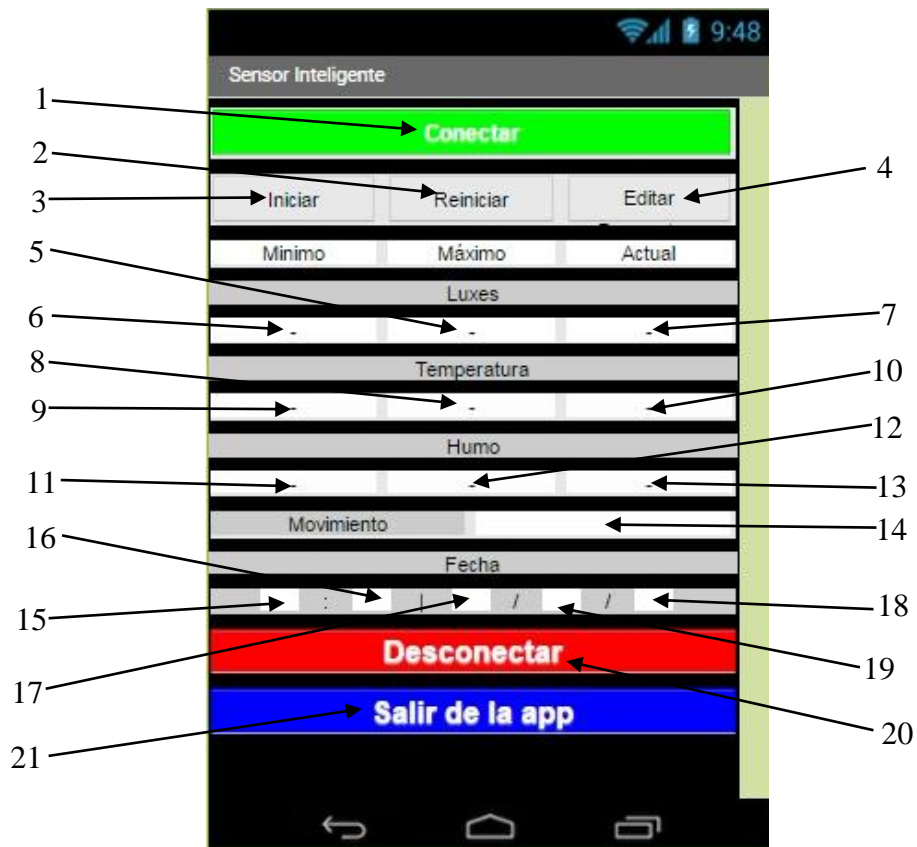


Figura 4.26. Interfaz de monitorización del sensor inteligente

Para el diseño de software se utilizaron varias herramientas de programación, es importante mencionar que al momento de poner en práctica la programación, se debe realizar un análisis previo de todos los elementos que conforma el sensor inteligente, teniendo en cuenta que un algoritmo es la etapa de control encargada del trabajo de cualquier dispositivo electrónico. Así con la culminación del diseño de software se puede comenzar con el diseño de hardware para que posteriormente se empiece la construcción del sensor inteligente.

Capítulo 5

Diseño de hardware y construcción del sensor inteligente

En este capítulo se presenta la construcción de hardware del sensor inteligente, en el cual se utiliza una gran variedad de herramientas de diseño siendo de gran utilidad en cuestiones de simulación, desarrollo e implementación. Dando como resultado la unificación de todos los elementos para crear un “sensor inteligente”.

5.1 Simulación del sistema del sensor inteligente

El programa PROTEUS VSM es una herramienta que permite comprobar la simulación en cualquier diseño, la eficacia del programa desarrollado. Su combinación de simulación de código de programación y simulación mixta SPICE permite la verificación analógico-digital de sistemas basados en microcontroladores y su potencia de trabajo es magnífica [11].

El desarrollo de la simulación se realizó mediante ISIS que es un potente programa de diseño electrónico que permite realizar esquemas que pueden ser simulados en el entorno VSM o convertirlos en circuito impreso en el entorno ARES. Posee una buena colección de bibliotecas de modelos tanto para dibujar, simular o para placas. Además, permite la creación de nuevos componentes.

En la figura 5.1 se observa la interfaz de ISIS que posee un entorno de trabajo formado por distintas barras de herramientas y la ventana de trabajo.

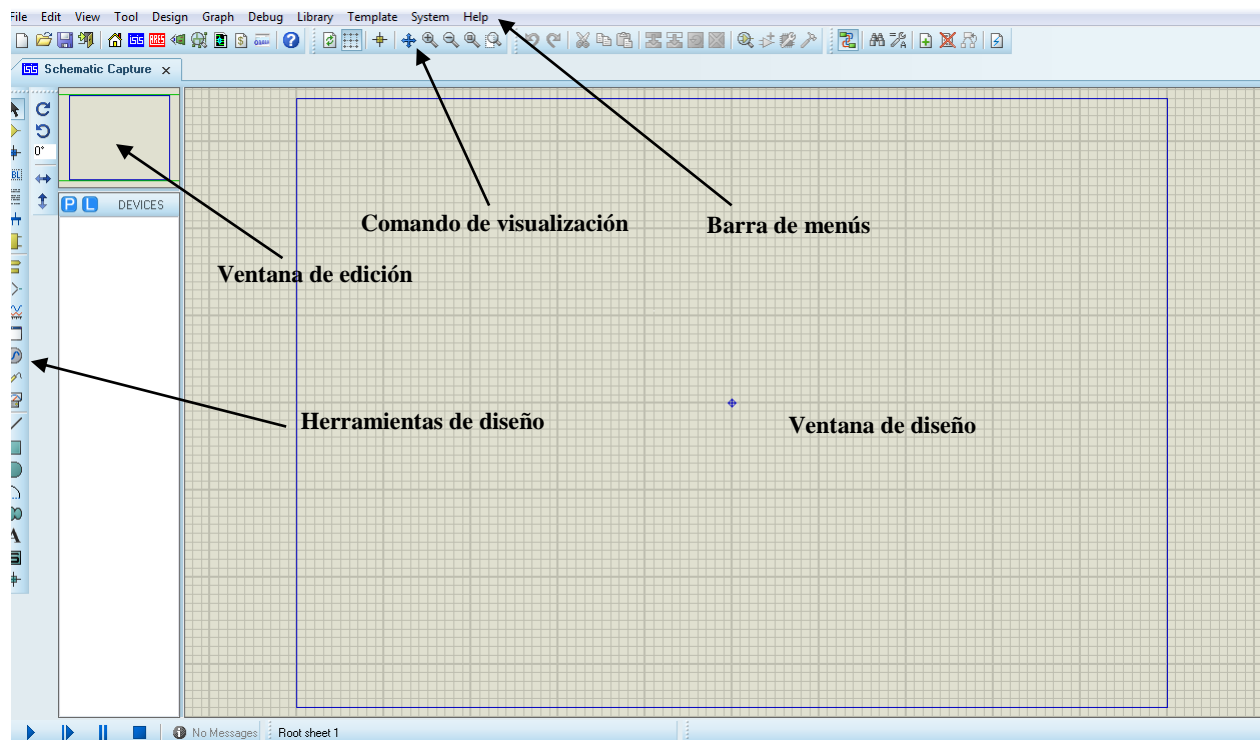


Figura 5.1. El entorno de trabajo del programa ISIS

5.3 Diseño del PCB en Eagle versión 7.3

El diseño del PCB del sensor inteligente fue realizado con el programa de diseño PCB Eagle. La tabla 5.1 se observa todos los elementos ocupados para su realización de la tarjeta PCB, a continuación se describe cada uno de ellos:

Tabla 5.1 Lista de dispositivos para el diseño de la tarjeta PCB del sensor inteligente

Cantidad	Descripción del dispositivo
1	Conector mini USB 32005-201.
1	Push botón 5501.
2	Capacitores de 15 pF.
1	Oscilador de cristal de 20 MHz
1	Sensor de iluminación TSL2561.
1	Sensor de temperatura LM35.
1	Sensor de movimiento HC-SR501
1	Sensor de gas y humo MQ2.
1	Led indicador de pulso de monitorización de 1 segundo.
1	Led indicador de alarma de temperatura.
1	Led indicador de iluminación.
1	Led indicador de alarma de movimiento.
1	Led indicador de alarma de detección de gas y humo.
1	Dispositivo de comunicación inalámbrica vía bluetooth HC-05.
1	Módulo de tiempo real DS1307.
1	Microcontrolador PIC18F4550.
6	Resistencias de 330 Ω .
3	Resistencias de 10 K Ω .
3	Resistencias de 1 K Ω .
1	Transistor 2N2222.
1	Zumbador.
20	Pines tipo macho.
1	Entrada de sensor con comunicación I2C.
2	Resistencias de 2.2 K Ω .

En la figura 5.3 se observa el diagrama eléctrico del sensor inteligente con todos los elementos que lo componen.

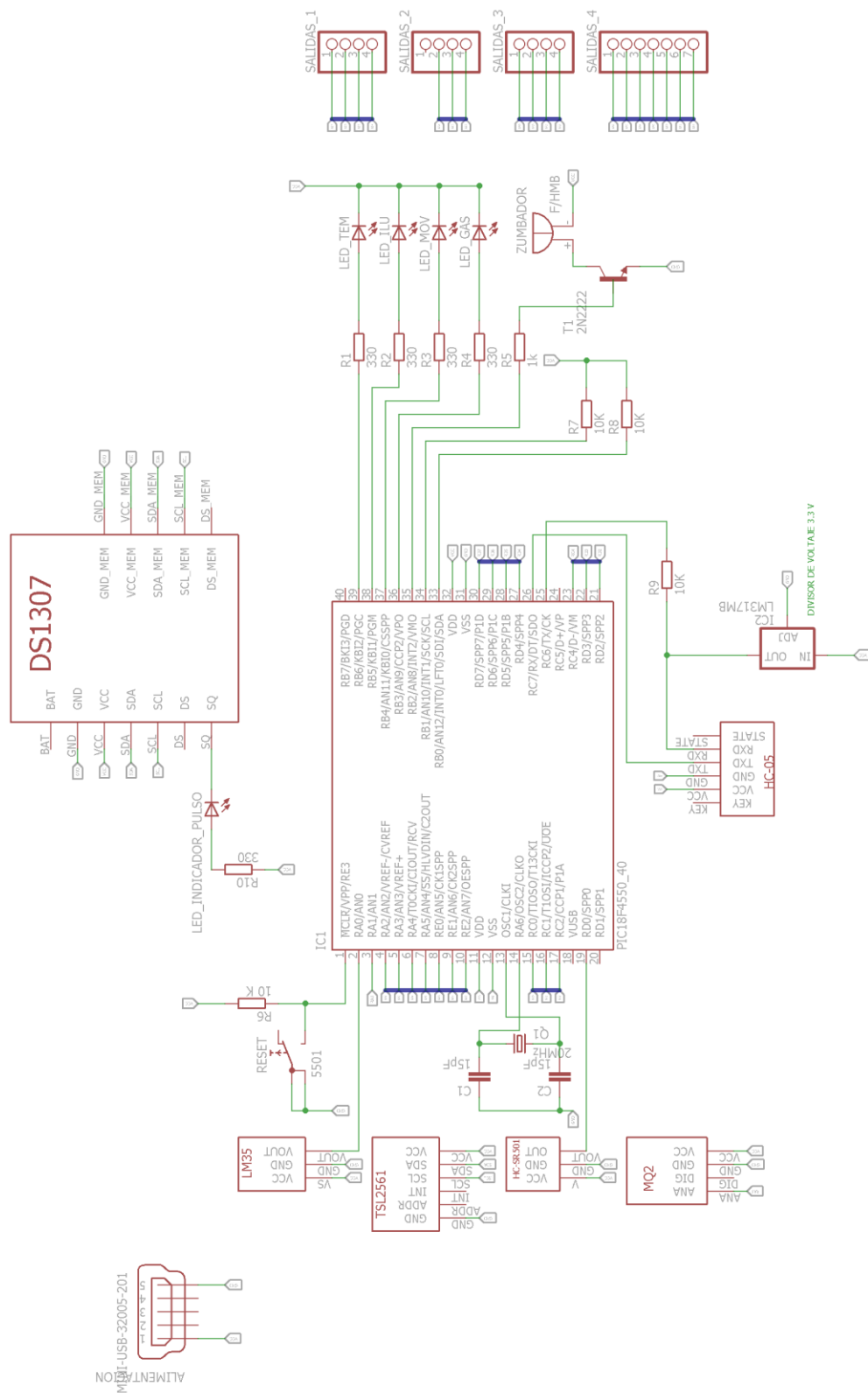


Figura 5.3 Diagrama eléctrico del sensor inteligente

Se realizó una simulación del diseño del sensor inteligente en 3D (figura 5.5.) para que posteriormente se realice su construcción.

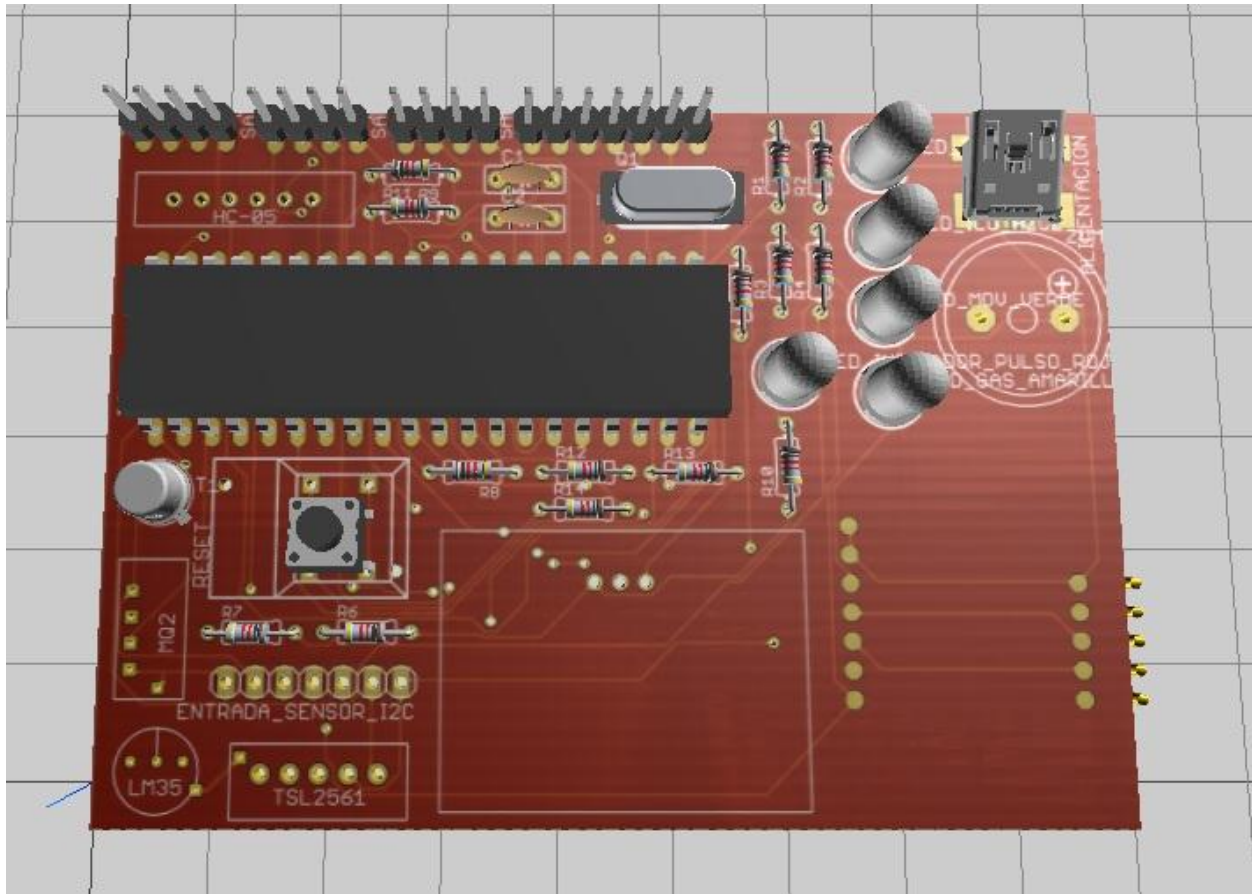


Figura 5.5. Simulación 3D del sensor inteligente

5.4 Construcción del sensor inteligente

Una vez diseñado el PCB, se realizó la construcción comenzando con la aplicación de pintura fotosensible (color azul) en la placa fenólica ya perforada, para que posteriormente se coloque la imagen procesada del negativo y positivo del diseño del sensor inteligente previamente generados en la computadora. Después se coloca en una solución con sales alcalina en agua con el diseño del negativo del PCB, para que se puede imprimir mediante una fuente de luz ultravioleta los trazos que interconectarán los componentes electrónicos.

La última fase es la de montar todos los componentes a la placa para que posteriormente sean soldados. En la figura 5.7 se observa el diseño del sensor inteligente con todos los dispositivos que lo componen.

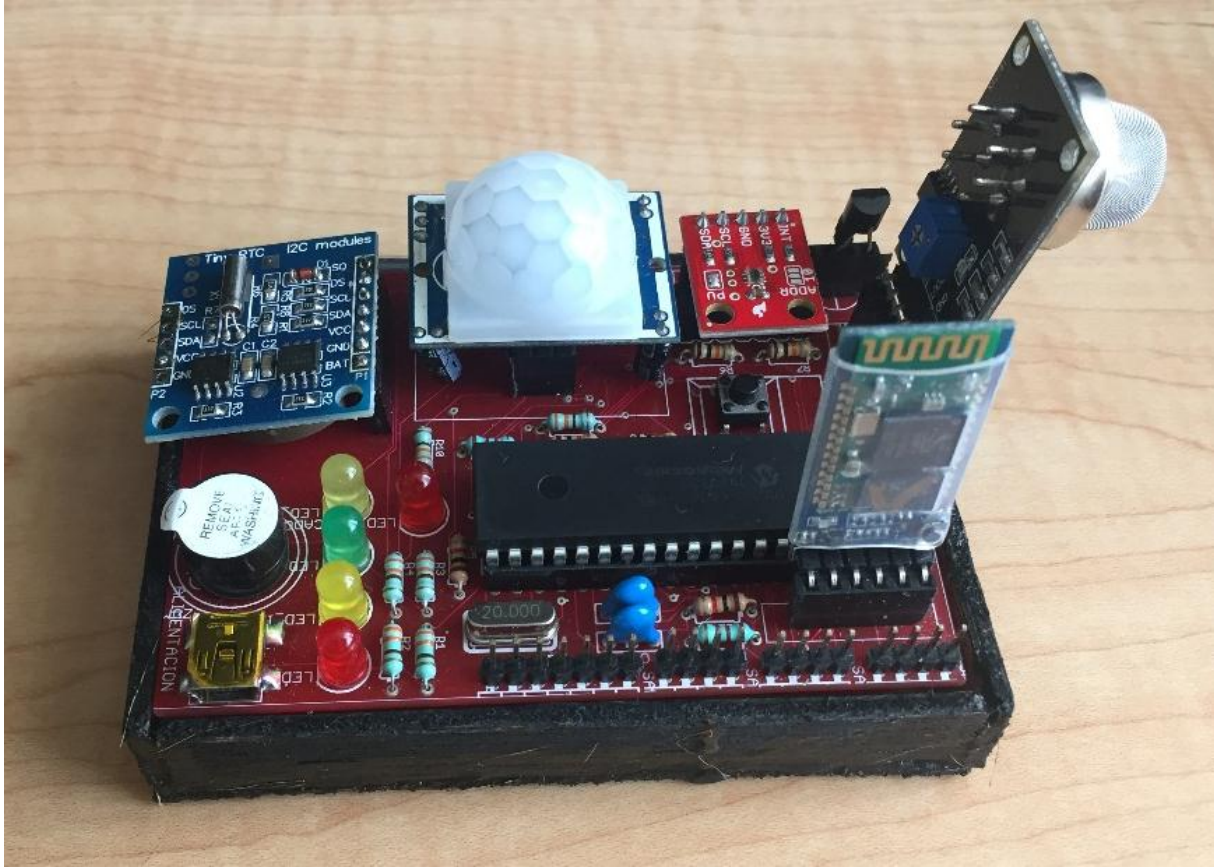


Figura 5.7. Sensor inteligente con todos sus elementos

Para el bosquejo de hardware se utilizaron varias herramientas de diseño, es importante mencionar que al momento de poner en práctica la realización del PCB se debe realizar un análisis previo de todas las características de los dispositivos que conforma el sensor inteligente. El diseño de bibliotecas mediante el programa EAGLE versión 7.3 facilita la construcción de la tarjeta. Una vez finalizada la etapa de construcción del sensor inteligente, se puede implementar para la monitorización de un sistema domótico en el área de seguridad.

Capítulo VI

Resultados

El presente capítulo contiene imágenes y tablas sobre la implementación de diferentes tiempos de monitorización del sensor inteligente, así como los resultados experimentales de las configuraciones que ofrece el sistema dando como resultado la obtención de información en tiempo real.

6.1 Resultados experimentales del sensor inteligente

Con la finalidad de corroborar el funcionamiento del sensor inteligente se realizó la comunicación serial entre la PC y el sistema de monitorización, los resultados obtenidos se visualizan en la figura 6.1, siendo los primeros parámetros obtenidos por del sistema. A continuación se describe cada uno de ellos:

1. Iluminación: 322 luxes
2. Temperatura: 30° C
3. Humo: 9 ppm
4. Movimiento: 10

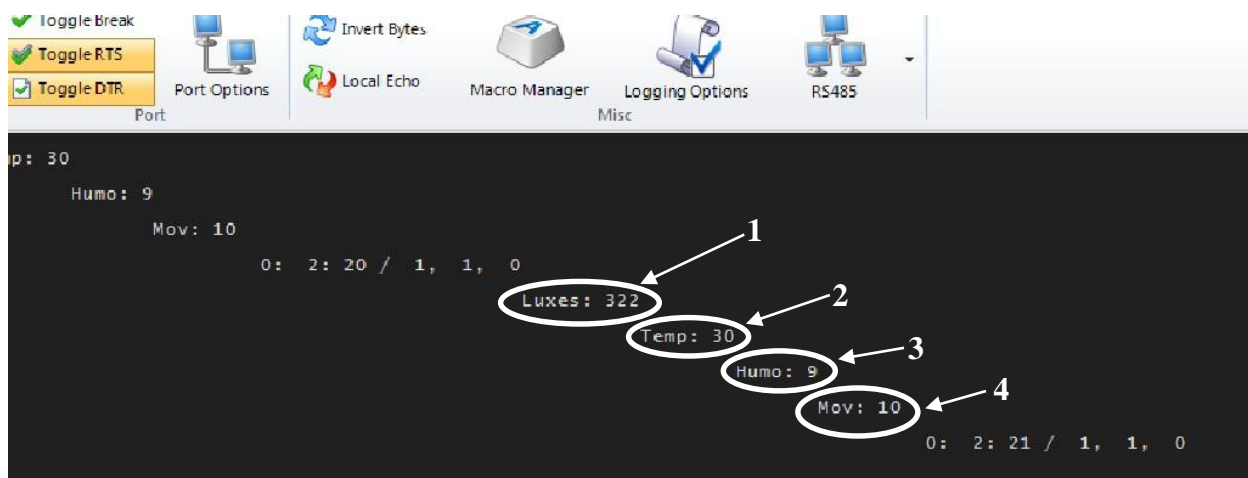


Figura 6.1. Monitorización inicial

Se utilizó un luxómetro modelo CEM DT-1309 para comparar las mediciones obtenidas de iluminación por el sensor inteligente. En la tabla 6.1 se muestran los valores medidos, y se observa que el error de iluminación es de -21.8 luxes con respecto al luxómetro. Es importante mencionar que el sensor inteligente solo muestra valores enteros en la monitorización de iluminación.

Tabla 6.1 Comparación de valores de iluminación

Luxómetro valores en luxes	Sensor inteligente valores en luxes	Diferencia en luxes
240	254	-14
252	274	-22
247	273	-26
248	272	-24
249	274	-25
245	270	-25
250	260	-10
229	258	-29
237	261	-24
248	258	-10
234	260	-26
235	262	-27

En la figura 6.2 se observa el luxómetro que es el encargado de comparar los valores de iluminación con respecto a los valores de monitorización del sensor inteligente. Se visualiza que la variación de la información obtenida por el sensor inteligente con respecto al luxómetro, no existe tanta diferencia y para la monitorización de un sistema domótico no es necesaria la exactitud en los valores conseguidos por el sensor inteligente.



Figura 6.2. Monitorización de la iluminación con respecto a un luxómetro

Se utilizó un multímetro Steren modelo MUL-500 para comparar las mediciones obtenidas de temperatura por el sensor inteligente. En la tabla 6.2 se muestran los valores medidos, y se observa que el error de temperatura es de -2.7°C con respecto al sistema de comparación. Es importante mencionar que el sensor inteligente solo muestra valores enteros en la monitorización de temperatura.

Tabla 6.2 Comparación de valores de temperatura

Multímetro valores en $^{\circ}\text{C}$	Sensor inteligente valores en $^{\circ}\text{C}$	Diferencia en $^{\circ}\text{C}$
23	25	-2
24	26	-2
24	27	-3
25	28	-3
26	28	-2
26	29	-3
26	28	-2
27	29	-2
24	28	-4
24	27	-3
25	28	-3
25	29	-4

En la figura 6.3 se observa el multímetro que es el encargado de comparar los valores de temperatura con respecto a los valores de la monitorización del sensor inteligente. Se visualiza que la variación de la información obtenida por el sensor inteligente con respecto al sistema de comparación, no existe tanta diferencia y para la monitorización de un sistema domótico no es necesaria la exactitud en los valores conseguidos por el sensor inteligente.

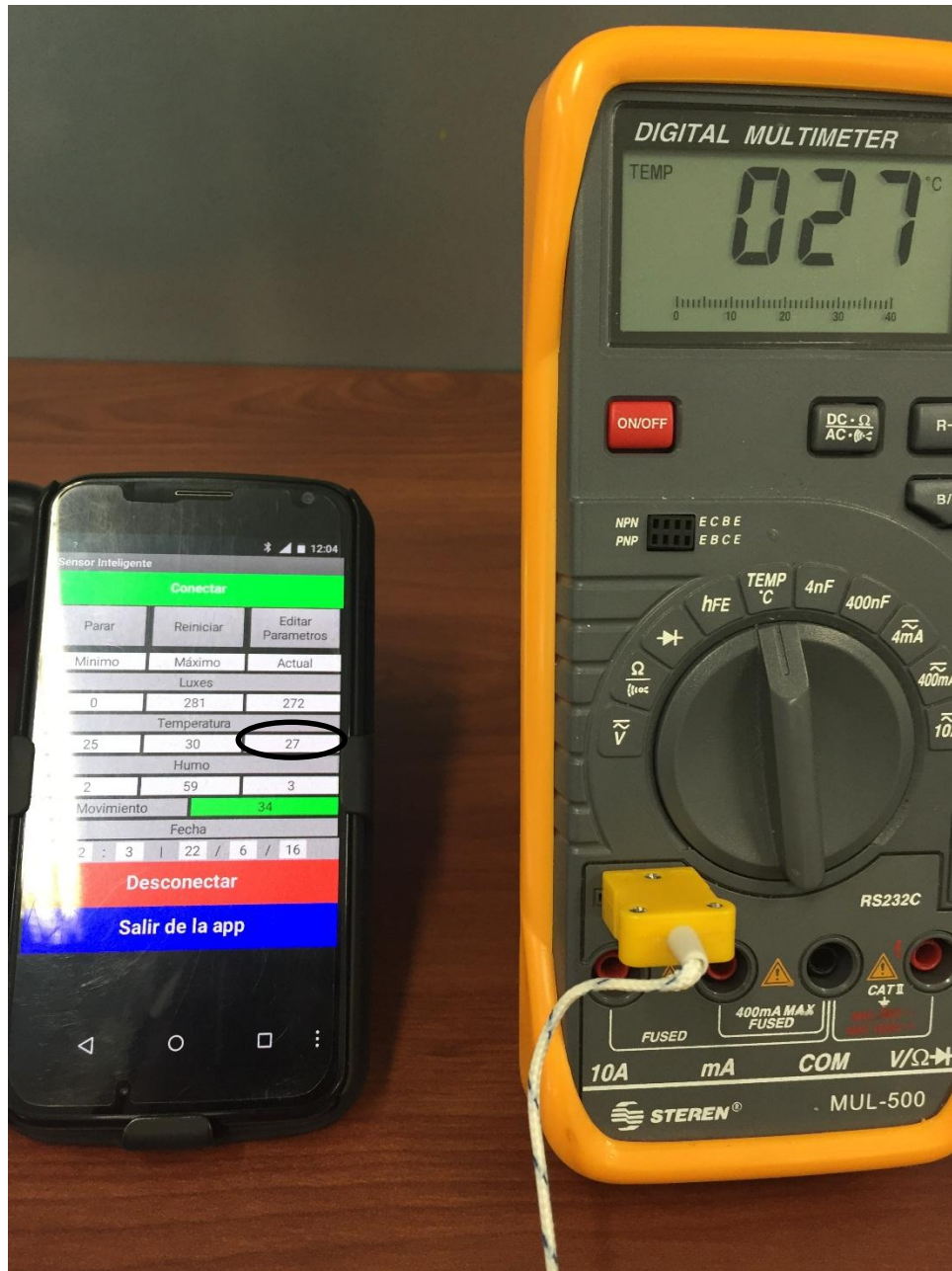


Figura 6.3 Monitorización de temperatura con respecto a un multímetro

6.2 Implementación del sensor inteligente

La implementación del sensor inteligente ha sido empleada en una casa habitacional situada en la ciudad de Puebla, siendo la sala de estar la ubicación exacta de la monitorización del sistema. En la figura 6.4 se observa el montaje del sensor inteligente en la sala de estar.



Figura 6.4. Montaje del sensor inteligente en sala de estar para la monitorización

En la figura 6.5 se observa a detalle el montaje del sensor inteligente con todos los elementos que lo componen.

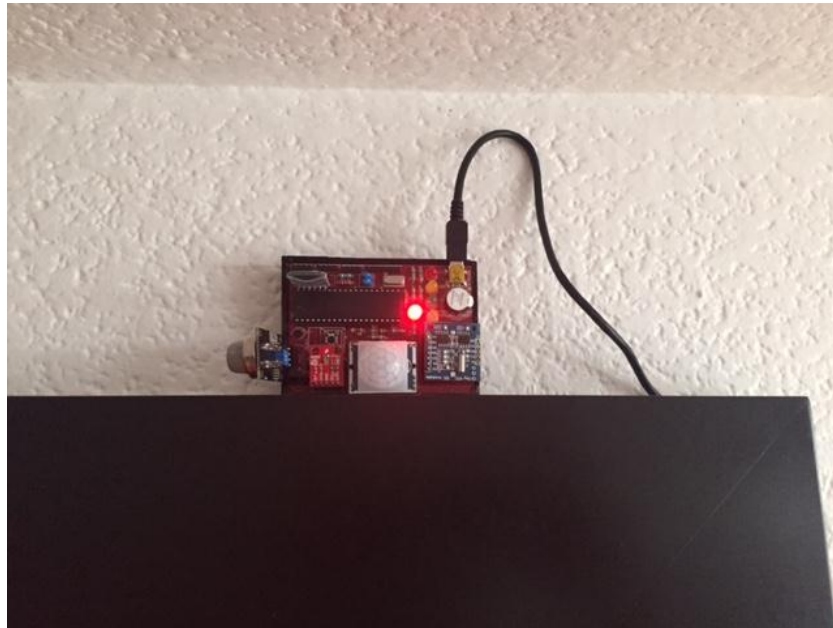


Figura 6.5 Montaje del sensor inteligente a detalle

En la figura 6.6 muestra la aplicación para la monitorización del sistema domótico, en cual se observa la información recabada por el sensor inteligente

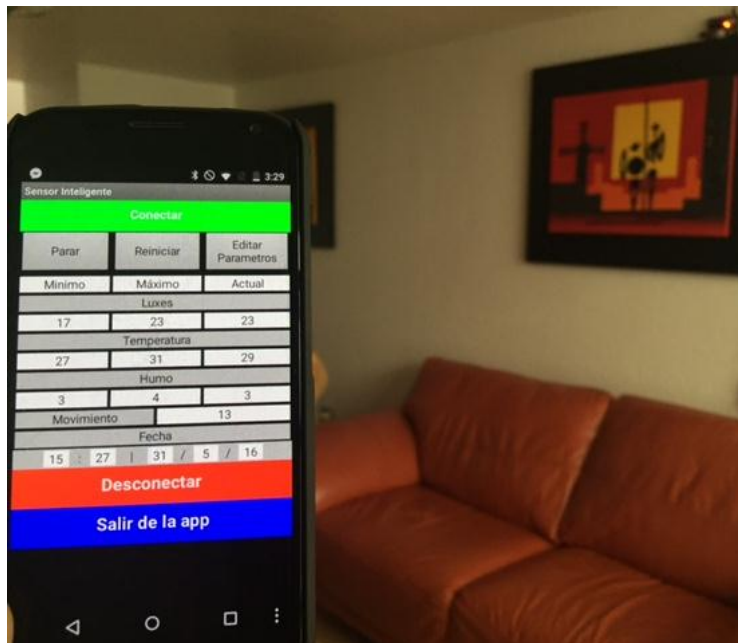


Figura 6.6 Aplicación de monitorización para un sistema domótico

En la figura 6.7 se observa la configuración inicial del sensor inteligente con la que se inicializa la monitorización del sistema, con una hora de inicio de 17:30 horas y fecha: 31 de mayo del 2016. A continuación se describe cada uno de los parámetros configurados.

1. Sensor de iluminación activado.
2. Configuración de iluminación mínima: 0 luxes
3. Configuración de iluminación máxima: 1000 luxes
4. Sensor de temperatura activado.
5. Configuración de la temperatura mínima: 15° C
6. Configuración de la temperatura máxima: 40° C
7. Sensor de humo y gas activado.
8. Configuración de humo o gas mínimo: 1 ppm
9. Configuración de humo o gas máximo: 10 ppm
10. Sensor de movimiento activado.
11. Zumbador activado.

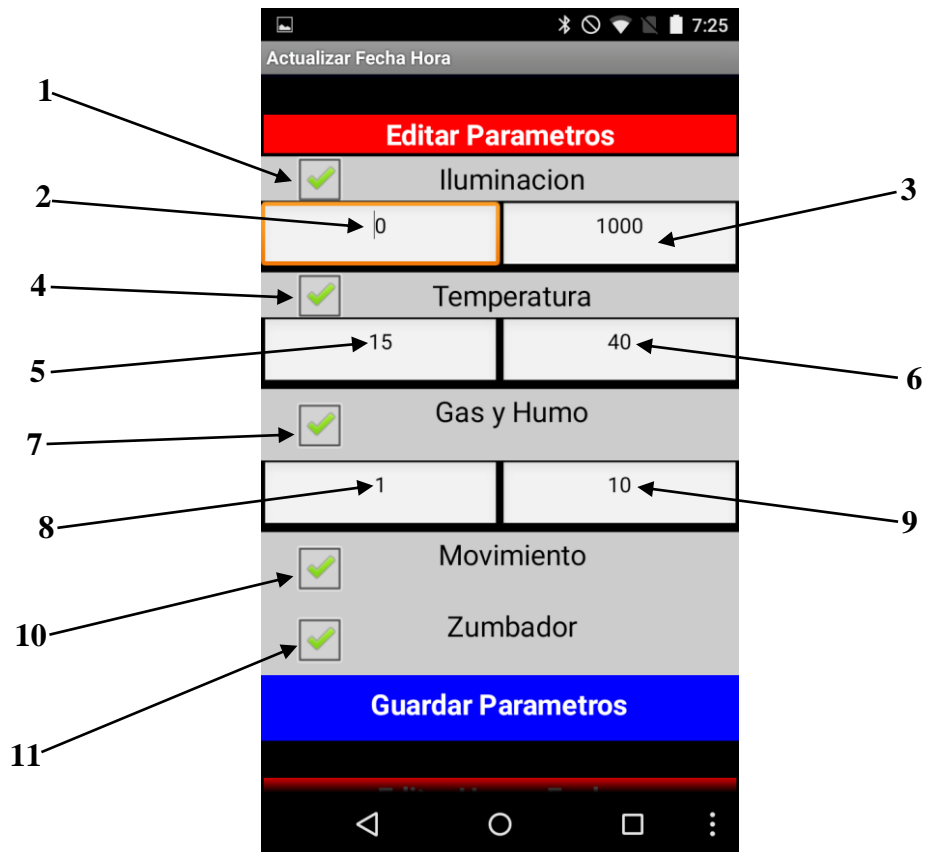


Figura 6.7 Configuración inicial del sensor inteligente

En la figura 6.8 se observa la monitorización del sistema, en estado estable, esto significa que los parámetros configurados no han sido rebasados en su totalidad, a continuación se describe cada una de estos:

1. 10 luxes de iluminación mínima de la monitorización del sistema.
2. 36 luxes de iluminación máxima de la monitorización del sistema.
3. 14 luxes de iluminación actual de la monitorización del sistema.
4. 32° C de temperatura máxima de la monitorización del sistema.
5. 28° C de temperatura mínima de la monitorización del sistema.
6. 30° C de temperatura actual de la monitorización del sistema.
7. 2 ppm gas o humo mínimo de la monitorización del sistema.
8. 3 ppm de gas o humo máximo de la monitorización del sistema.
9. 2 ppm de gas o humo actual de la monitorización del sistema.
10. Movimientos detectados de la monitorización del sistema.

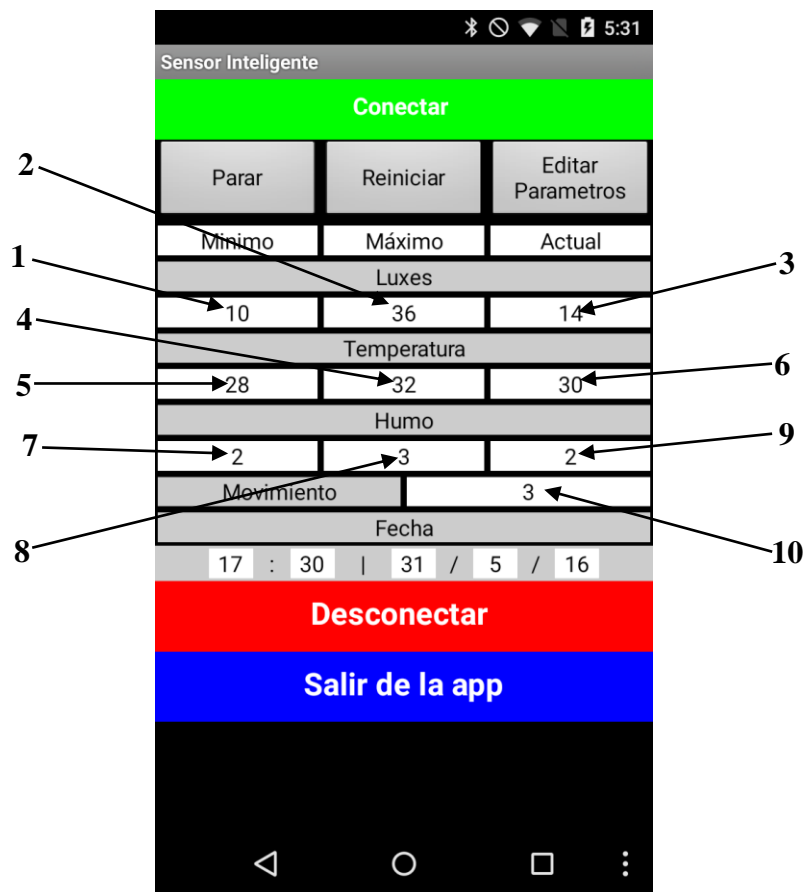


Figura 6.8. Motorización del sensor inteligente hora 17:30 Horas

En la figura 6.9 se tiene la monitorización del sistema con un parámetros superado con respecto a la configuración inicial de la figura 6.7, se puede notar que la casilla de humo o gas se coloca de color rojo para advertir al usuario que los parámetros configurados han sido rebasados en su totalidad. Teniendo como tiempo de detección las 19:20 horas del día 31 de mayo de 2016.

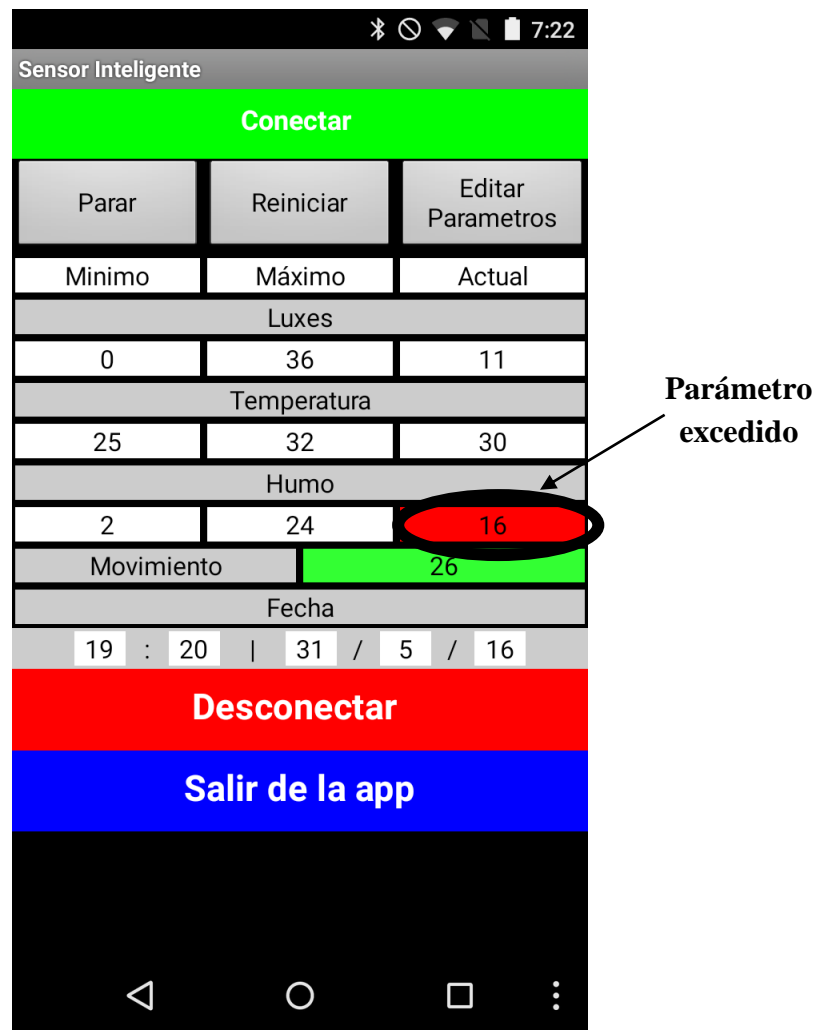


Figura 6.9. Parámetro de configuración excedido

En la tabla 6.3 se tiene la monitorización de iluminación en un sistema domótico, en un lapso de tiempo de 17:00 a 22:00 horas con obtención de valores cada 30 minutos. Las variaciones fueron tomadas con \pm un minuto de diferencia con respecto a la hora indicada.

Tabla 6.3. Monitorización de iluminación

FECHA	HORA	ILUMINACIÓN MÍNIMA EN LUXES	ILUMINACIÓN MÁXIMA EN LUXES	ILUMINACIÓN ACTUAL EN LUXES
31/05/16	17:30	10	36	13
31/05/16	18:00	0	36	1
31/05/16	18:30	0	36	0
31/05/16	19:00	0	36	0
31/05/16	19:30	0	36	1
31/05/16	20:00	0	36	0
31/05/16	20:30	0	36	0
31/05/16	21:00	0	36	0
31/05/16	21:30	0	36	11
31/05/16	22:00	0	36	11

En la tabla 6.4 se tiene la monitorización de temperatura en un sistema domótico, en un lapso de tiempo de 17:00 a 22:00 horas con obtención de valores cada 30 minutos. Las variaciones fueron tomadas con \pm un minuto de diferencia con respecto a la hora indicada.

Tabla 6.4 Monitorización temperatura

FECHA	HORA	TEMPERATURA MÍNIMA EN °C	TEMPERATURA MÁXIMA EN °C	TEMPERATURA ACTUAL °C
31/05/16	17:30	28	32	30
31/05/16	18:00	27	32	31
31/05/16	18:30	27	32	30
31/05/16	19:00	25	32	29
31/05/16	19:30	25	32	29
31/05/16	20:00	24	32	30
31/05/16	20:30	24	32	30
31/05/16	21:00	24	32	29
31/05/16	21:30	24	32	29
31/05/16	22:00	24	32	28

En la tabla 6.5 se tiene la monitorización de humo y gas en un sistema domótico, en un lapso de tiempo de 17:00 a 22:00 horas con obtención de valores cada 30 minutos. Las variaciones fueron tomadas con \pm un minuto de diferencia con respecto a la hora indicada.

Tabla 6.5 Monitorización de humo y gas

FECHA	HORA	HUMO MÍNIMO EN PPM	HUMO MÁXIMO EN PPM
31/05/16	17:30	2	3
31/05/16	18:00	2	4
31/05/16	18:30	2	4
31/05/16	19:00	2	4
31/05/16	19:30	2	24
31/05/16	20:00	2	24
31/05/16	20:30	2	24
31/05/16	21:00	2	24
31/05/16	21:30	2	24
31/05/16	22:00	2	24

En la tabla 6.6 se tiene la monitorización de movimiento en un sistema domótico, en un lapso de tiempo de 17:00 a 22:00 horas con obtención de valores cada 30 minutos. Las variaciones fueron tomadas con \pm un minuto de diferencia con respecto a la hora indicada.

Tabla 6.6 Monitorización de movimiento

FECHA	HORA	DETECCIONES
31/05/16	17:30	3
31/05/16	18:00	3
31/05/16	18:30	7
31/05/16	19:00	10
31/05/16	19:30	28
31/05/16	20:00	28
31/05/16	20:30	34
31/05/16	21:00	34
31/05/16	21:30	44
31/05/16	22:00	45

Se concluye que los resultados obtenidos por el sensor inteligente son los idóneos, teniendo en cuenta que la obtención de información fue en tiempo real, los valores no tienen tanta desventaja entre el sistema de comparación con respecto al sensor inteligente. Por lo tanto se puede concluir que el sensor inteligente funciona como un sistema confiable para la obtención de información.

Capítulo VII

Conclusiones

7.1 Conclusión

A lo largo del desarrollo se realizaron simulaciones con distintas variables para el diseño del sensor inteligente, con los resultados se deduce que los sensores inteligentes representan una tecnología que aventaja a los sensores tradicionales en que integran por separado, un sensor, un acondicionamiento de señal, memoria y comunicación.

Las variables del sensor inteligente dependen de factores relacionados con el proceso al que se va a implementar. Permitiendo al sensor inteligente acoplarse a cualquier sistema y crear una red de información para ser enviada a los centros de control encargados de la monitorización.

Con la implementación del sensor inteligente en el área de seguridad en un sistema domótico se tiene la certeza de la obtención de información en tiempo real para la prevención de algún percance. Si se requiere de tiempos más cortos se puede cambiar el tipo de microcontrolador encargado del proceso de la información, por uno de mayor velocidad.

Las dimensiones del sensor inteligente pueden ser de menor tamaño, se necesita rediseñar el sistema con elementos que cumplan con las características que requiera la implementación de dicho sistema, como elementos de superficie.

Es importante mencionar con respecto a los sensores inteligentes, es que depende de la aplicación donde se requiera implementar la monitorización, son las variables a utilizar, el cual puede ser implementado en una gran variedad de procesos, tales como: sistema automotriz, invernaderos, industrial, etc. Solo se requiere cambiar los sensores con unos que cumplan con las características del requerimiento del sistema.

En el diseño del sensor inteligente se dejaron pines libres tanto de entrada como de salida que posee el microcontrolador para que posteriormente el sistema puede crecer, realizando los cambios necesarios en el software.

El sensor inteligente tiene la capacidad de alertar mediante una alarma tipo led y mediante un zumbador de la presencia de un peligro. Se seleccionó estos dos medios de alerta para facilitar la interacción entre personas discapacitadas.

Gracias a realización de este trabajo, se obtuvo experiencia con la programación de microcontroladores PIC, con el desarrollo de PCB y la comunicación de forma inalámbrica vía Bluetooth. La comunicación vía Bluetooth se diseñó e implemento de manera correcta a una velocidad de 9600 baudios, cumpliendo con el objetivo de transmitir datos con otros sistema.

7.2 Trabajos a futuro

El sensor inteligente puede ser utilizado en una gran variedad de aplicaciones dependiendo de la rama que se vaya a utilizar, esto significa que contiene más opciones de configuración, dichas configuraciones son descritas en el capítulo IV, estas pueden ser implementadas realizando cambios en el algoritmo general e integrarlas en la app diseñada en app inventor 2, solo se menciona para que posteriormente a futuro se puedan incorporar al sistema.

Integrar un sistema de comunicación a internet en el sensor inteligente, representa adentrarse en la categoría de internet de las cosas (IoT), dando como resultado la expansión de la comunicación personal entre sistema y personas.

Se puede realizar un manual de usuario con todas las características del sensor inteligente, en el cual se pueda visualizar todos los parámetros de monitorización que tolere el sistema, esto ayudará a futuros diseñador a implementar procesos de automatización más eficientes y en un menor tiempo de construcción.

Bibliografía

- [1] S. Middelhoek and A. Hoogerwerf, (1985). Smart sensor: When and Where?, *Sensors and actuators*, No. 8, pp.39-48.
- [2] J. Huijsing, F Riedijk and G. Horn, (2004) “Development in integrated smart sensor”, *Sensor and actuators*, No.43, pp.276-288.
- [3] G: Jordan (1985) “Sensor technologies of the future”, *J.Phys. E: Sci. Instrum*, No. 18.
- [5] Angulo Usategui José M. (2006) “Microcontroladores PIC Diseño práctico de aplicaciones” Segunda Edición
- [6] P. Bentley John. (199) “Sistemas de medición principios y aplicaciones” 2ª Edición Pág.229-235.
- [7] J. Tocci Ronal (1993) “Sistemas digitales Principios y aplicaciones” 5ª Edición Pág. 574-580.
- [8] Randy Frank (2000). *Understanding smart sensor* (2a ed.) Artech House, Inc. Norwood. MA, USA.
- [9] Tomasi Wayne () “Comunicación Electronica” Augarten, Stan (1983) “The most widely used computer on a chip” *State of the Art: A Photographic History of the Integrated Circuit*. New Haven y New York.
- [10] Enrique Palacios, Fernando Remiro (2006) “Microcontrolador PIC16F84, Desarrollo de Proyectos” 2ª Edición Pág. 2-10.
- [11] Eduardo García Breijo (2008) “Compilador C CCS y Simulador PROTEUS para Microcontroladores” 1ª Edición Pág. 276.
- [12] Gerard C.M. Meijer (2008) “Smart Sensor Systems” John Wiley & Son Pág.411.
- [13] S. Middelhoek and A. Hoogerwerf (1995). *Smart Sensor: When and Where?*, *Sensors and actuators*, No 8