



**FACULTAD DE INFORMÁTICA Y CIENCIAS APLICADAS
TÉCNICO EN INGENIERÍA DE HARDWARE**



TEMA:

**AUTOMATIZACIÓN DEL CONTROL DE AMBIENTACIÓN Y TIMBRADO
DESDE UNA APLICACIÓN PARA PC DESARROLLADO PARA SISTEMAS
OPERATIVOS WINDOWS, COMO APOYO A LAS LABORES
REPETITIVAS DEL INSTITUTO NACIONAL DE AGUILARES.**

TRABAJO DE GRADUACIÓN PRESENTADO POR:

**ARNOLDO FABRICIO MOLINA LEMUS
HERNÁN ULISES ALONZO ORTIZ
KEVIN DAVID RAUDA SOLIS**

PARA OPTAR AL GRADO DE:

TÉCNICO EN INGENIERÍA DE HARDWARE

SEPTIEMBRE, 2013

SAN SALVADOR, EL SALVADOR, CENTROAMÉRICA.

PAGINA DE AUTORIDADES

**ING. NELSON ZÁRATE SÁNCHEZ
RECTOR**

**LIC. JOSÉ MODESTO VENTURA
VICERRECTOR ACADEMICO**

**ING. FRANCISCO ARMANDO ZEPEDA
DECANO**

JURADO EXAMINADOR

**ING. JULIO CÉSAR ROSALES
PRESIDENTE**

**ING. MARIA EVA CARRANZA
PRIMER VOCAL**

**ING. OMAR OTONIEL FLORES
SEGUNDO VOCAL**

SEPTIEMBRE, 2013

SAN SALVADOR, EL SALVADOR, CENTROAMÉRICA

**Universidad
Tecnológica de El Salvador**
ACTA DE EXAMEN PROFESIONAL



HABIÉNDOSE REUNIDO EL JURADO CALIFICADOR INTEGRADO POR:

Ing. Maria Eva Carranza, Ing. Omar Otoniel Flores Cortez, Ing. Julio César Rosales

a las 12:30p.m. del día Lunes, 24 de junio de 2013

Y LUEGO DE HABER DELIBERADO SOBRE EL EXAMEN PROFESIONAL DE LOS ALUMNOS:

1-Hernan Ulises Alonzo Ortiz Carnet 28-2322-2011

2-Arnoldo Fabricio Molina Lemus Carnet 28-3437-2011

3-Kevin David Rauda Solis Carnet 28-2651-2011

QUIENES PRESENTARON DEFENSA DE SU TRABAJO DE GRADUACION TITULADO:

"Automatización del control de ambientación y timbrado desde una aplicación en PC desarrollado para sistemas operativos Windows 7 como apoyo a las labores repetitivas del Instituto Nacional de Aguilares"

PARA OPTAR AL GRADO DE:

TÉCNICO EN INGENIERIA DE HARDWARE

Y DEL CUAL TAMBIEN EVALUARON LOS CONOCIMIENTOS RELACIONADOS CON EL TEMA DEL MISMO. POR LO QUE ESTE JURADO RESUELVE DECLARAR EL EXAMEN COMO:

APROBADO.

YA QUE CUMPLE CON LOS REQUISITOS ESTABLECIDOS EN EL REGLAMENTO DE GRADUACION DE LA UNIVERSIDAD.

San Salvador, 24 de junio de 2013.

F. 
PRIMER VOCAL
Ing. Maria Eva Carranza

F. 
SEGUNDO VOCAL
Ing. Omar Otoniel Flores Cortez

F. 
PRESIDENTE
Ing. Julio César Rosales

ÍNDICE

introducción	i
Capitulo I: Situación Actual	1
1.1 Situación Problemática	1
1.2 Enunciado del Problema.	1
1.3 Justificación.	2
1.4 Objetivos.....	3
1.4.1 Objetivo General.....	3
1.4.2 Objetivos Específicos.....	3
1.5 Delimitaciones	3
1.5.1 Geográfica	3
1.5.2 Temporal	3
1.5.3 Organizacional	4
1.6 Alcances.....	5
1.7 Estudio de Factibilidad.....	6
1.7.1 Factibilidad Económica	6
1.7.2 Factibilidad Técnica	8
Capitulo II: Documentacion Técnica.....	12
2.1 Marco Teórico de Referencia.....	12
2.1.1 Sistemas Y Tipos de Sistemas.....	12
2.1.2 Definición de Domótica	15
2.1.2 Características de Un Sistema Domótico	16
2.1.3 Topología de La Red	17
2.1.4 Tipos de Arquitectura.....	19
2.1.5 Medios de Transmisión	20
2.1.5.1 Medios de Transmisión Alámbricos.....	21
2.1.6 Áreas a Gestionar	23
2.1.7 Componentes básicos de un Sistema Domótico	24

2.1.7.1 Tipos de Señal.....	25
2.1.7.2 Sensores	25
2.1.7.3 Acondicionamiento de Señal	26
2.1.7.4 Actuadores	26
2.1.7.5 Interfaces.....	29
2.1.7.6 Infraestructura	29
2.1.7.7 Unidad de Control.....	30
2.1.7.8 Software de Control	30
2.1.8 Manuales y Tipos de Manuales.....	31
2.2 Marco Teórico de Solución	33
2.2.1 Luminarias del Auditorio General	34
2.2.2 Control de los Ventiladores	37
2.2.3 Timbre General de la Institución	40
2.2.4 Manual de Usuario.....	44
2.2.5 Código de la Aplicación.....	69
2.3 Marco Teórico Conceptual	82
2.4 Documentación Técnica.....	86
Arduino Mega R3.....	86
Módulo de Relés.....	92
CAPÍTULO III: DESARROLLO DE LA SOLUCIÓN	94
3.1 Propuesta de Solución.....	94
3.1.2 Desarrollo de la Solución.	99
3.1.3 Construcción de la Solución.....	102
3.2 Conclusiones	106
3.3 Recomendaciones	108
Bibliografía.....	109
Anexos	110
Matriz de Congruencia.....	110
Firmware de Control	111
Carta de Aceptacion	139
Carta de Agradecimiento	140

INTRODUCCIÓN

Este documento, presenta de la implementación del proyecto denominado: Automatización del control de ambientación y timbrado desde una aplicación para PC desarrollado para sistemas operativos Windows como apoyo a las labores repetitivas del Instituto Nacional de Aguilares. Este Sistema Domótico tiene como propósito controlar dispositivos eléctricos, gestionando áreas de confort y ahorro energético, administrando así de forma centralizada y de manera sencilla las tareas repetitivas del Instituto Nacional de Aguilares.

Este informe se divide en tres capítulos: el capítulo número uno dará a conocer la situación actual del problema que se pretende solucionar, de la misma manera se justificará el por qué se decidió realizar este proyecto el cual brindara un control de varios dispositivos eléctricos desde un mismo punto, así mismo se conciben algunos objetivos que se pretenden alcanzar, para el desarrollo óptimo de este como la creación de un circuito que sea capaz de realizar las acciones antes mencionadas y la creación de un manual de uso. El proyecto va dirigido a un espacio limitado es por esta razón, que en este documento hay un apartado en el cual se delimita geográficamente y temporalmente el desarrollo del trabajo, así de esta manera se mostrará detalladamente cada uno de los puntos necesarios para poder desarrollar el proyecto.

El capítulo número dos se divide en cuatro apartados principales, en los cuales se desarrollará las investigaciones necesarias y pertinentes para dar un respaldo teórico y técnico al proyecto.

Dichos apartados hacen referencia a:

- **Marco Teórico de referencia**

Este hace referencia al área de investigación e implementación, de las cuales se toman las bases de fundamentación del proyecto.

- **Marco Teórico de Solución**

En este apartado se hace referencia a la teoría de solución proporcionada por el proyecto, mostrando en él la integridad de los componentes del Sistema Domótico, además muestra ciertos avances de cómo quedará el producto final tales como interfaz de usuario del programa de control y sus áreas.

- **Marco Teórico Conceptual**

En éste se muestran conceptos teóricos sobre algunos aspectos inmiscuidos en el desarrollo e implementación del proyecto, generando así una perspectiva conceptual de los aspectos que integran el proyecto.

- **Documentación Técnica**

Este contiene conceptos técnicos de los componentes a utilizar en el proyecto.

El capítulo número tres muestra los procesos necesarios para obtener un producto final en base a las situaciones previamente planteadas en los capítulos anteriores, mostrando en su contenido el firmware de control, los prototipos pruebas del desarrollo del Sistema Domótico, diseños finalizados del PCB, diseño del tablero final donde se almacenará la unidad de control, y las etapas de diseño del manual.

Posee también apartados de recomendaciones y conclusiones anexos en los que se detallan

- Matriz de congruencia

En esta matriz se detallan los objetivos tanto específicos como generales, alcances que se pretendan obtener mediante promesas establecidas y los productos finales los cuales tendrán que ser congruentes entre sí.

CAPITULO I: SITUACIÓN ACTUAL

1.1 Situación Problemática

El Instituto Nacional de Aguilares situado en final calle la ronda, en la Colonia Salinas en el municipio de Aguilares departamento de San Salvador, cuenta con sistemas de ventilación e iluminación en sus aulas y en su auditorio, también cuenta con un timbre el cual se encarga de avisarle a los alumnos que ingresen a sus clases en los horarios establecidos y la salida a los recesos.

En dicha institución actualmente se cuenta con los recursos humanos para activar los dispositivos eléctricos, pero dicho personal es insuficiente para dar el abasto necesario en cuanto al control y el cuidado de los mismos, ya que no se cuenta muchas veces con la coordinación suficiente como para la activación del timbre a las horas establecidas, pues muchas veces el retraso en el manejo de éste, provoca una pérdida de tiempo en horas clases afectando en gran manera el rendimiento académico de los alumnos , así como también para el control de las luminarias y ventiladores cuando los alumnos hacen uso inadecuado de estos, sumando a esto el cuidado de dichos recurso, pues como menciona el personal de mantenimiento de esta institución los alumnos muchas veces dañan las terminales de control, impidiendo así la utilización de dichos dispositivos

1.2 Enunciado del Problema.

¿Será posible crear un Sistema Domótico basado en sistemas digitales programables, desarrollado para ser controlado desde una computadora para la

automatización del Instituto Nacional de Aguilares que controle la ambientación de áreas específicas de dicho recinto?

1.3 Justificación.

Se convierte en una gran necesidad dar una solución factible a los inconvenientes que se presenta el Instituto Nacional de Aguilares y que brinde un beneficio energético, vida útil de los equipos y un ambiente de confort, tanto para los alumnos como para el personal de la institución. Por ello se ve la necesidad de desarrollar como producto un sistema que pueda solventar los diferentes problemas anteriormente mencionados, muchas empresas y hogares que se encuentran a la vanguardia con la tecnología, ahorro energético y confort usan diferentes sistemas que les facilitan ciertas áreas, teniendo de esta manera una vida más tranquila y con tiempo extra para realizar sus labores y optimizar su tiempo.

Pero cuando la realización de actividades repetitivas como verificar si todas las luminarias han sido apagadas o si todos los ventiladores están apagados pero principalmente que el timbre suene exactamente en los horarios establecidos se vuelven prioridades. Es por eso que la necesidad de crear un sistema capaz de enviar órdenes desde un mismo punto a diferentes equipos o componentes eléctricos se convierte en una excelente aplicación.

Generando así un ambiente confortable para el usuario; ya que con este tipo de sistemas, facilitará y coordinará acciones cotidianas que se vuelven repetitivas, todo desde un mando central que cumpla con los requisitos mínimos, como para instalar un programa de interfaces graficas sencillas, que cuente con un puerto serie (USB) y con un sistema operativo Windows.

1.4 Objetivos.

1.4.1 Objetivo General.

Desarrollar un sistema de automatización que controle la ambientación y el timbrado desde una aplicación para PC, desarrollada para sistemas operativos Windows, como apoyo a las labores repetitivas del Instituto Nacional de Aguilares.

1.4.2 Objetivos Específicos.

- Desarrollar un circuito electrónico basado en sistemas digitales programables con aplicación en la Domótica que sea controlado desde una computadora del Instituto Nacional de Aguilares.
- Adecuar e instalar el Sistema Domótico en las instalaciones del Instituto Nacional de Aguilares
- Diseñar un manual digital e impreso para el manejo óptimo tanto del programa como de la instalación.

1.5 Delimitaciones

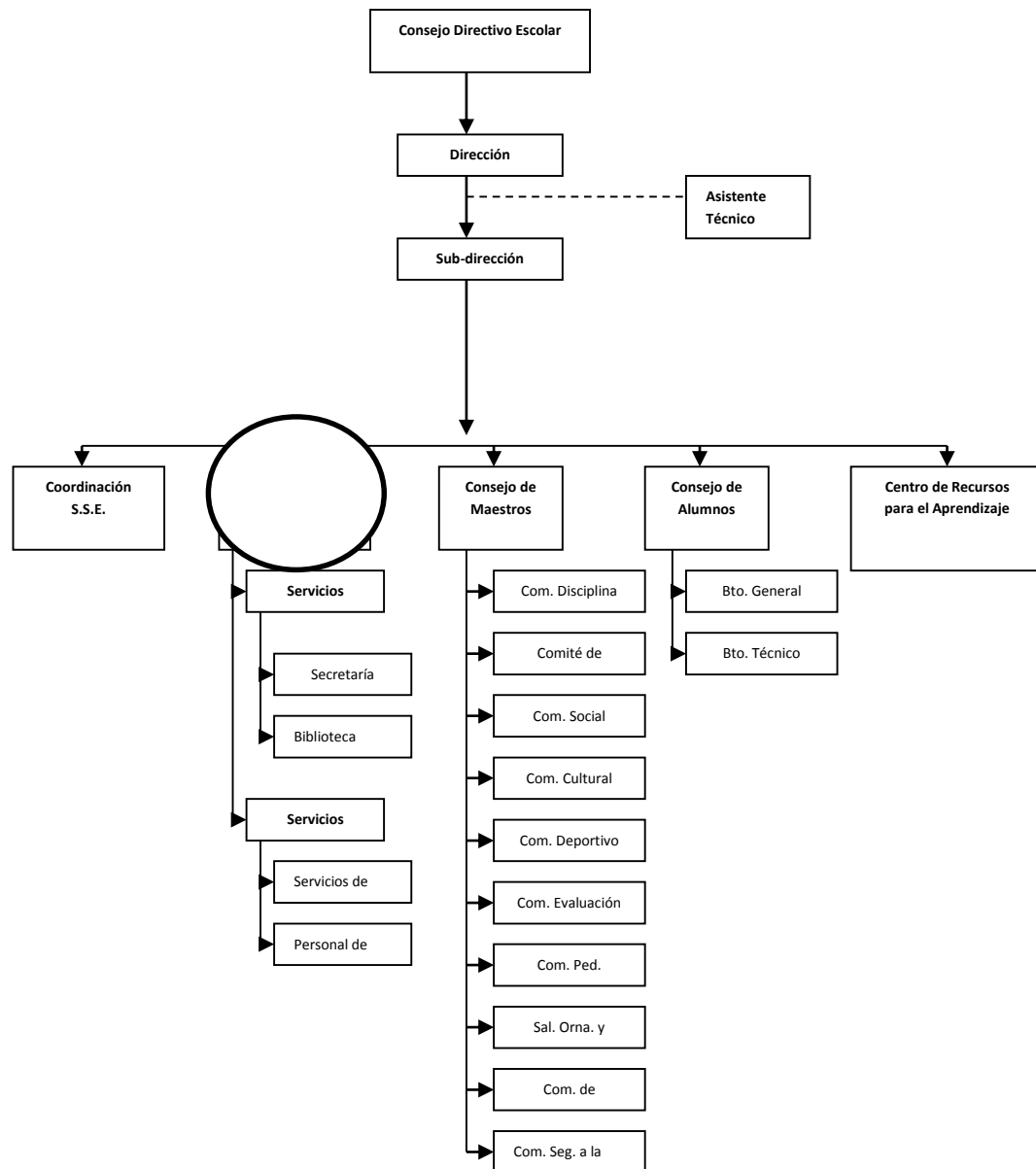
1.5.1 Geográfica

La realización del proyecto se llevará a cabo en el Instituto Nacional de Aguilares, final calle la Ronda colonia Salinas Aguilares, San Salvador.

1.5.2 Temporal

Se realizará en un lapso aproximado de 2 meses, desde mayo de 2013, hasta agosto de 2013.

1.5.3 Organizacional



El proyecto se llevará a cabo en la unidad de administración en el área de servicios de personal ya que esta es la encargada de gestionar las tareas de mantenimiento y es el área que se encarga en primera instancia de velar y controlar la ambientación y el timbrado del recinto.

1.6 Alcances.

En la realización de este proyecto para la implementación del Sistema Domótico se deben considerar ciertos parámetros y lineamientos para la realización de los objetivos como producto final; En el siguiente cuadro se mostrarán promesas y productos que dicho proyecto pretende llevar a acabó.

Promesa	Producto
1 – Desarrollar un circuito electrónico controlado desde una computadora para el manejo de la ambientación del instituto nacional de Aguilares.	1 – Circuito eléctrico basado en ARDUINO con una interface de control de voltajes medios: que reciba órdenes desde una computadora para manejar dispositivos eléctricos de iluminación ventilación y timbrado.
2 – Adecuación e Instalación del Sistema Domótico, demostración de que el circuito creado funciona de forma adecuada	2 –Adecuar e instalar el Sistema Domótico en las instalaciones del Instituto Nacional de Aguilares
3 - Diseñar un manual digital e impreso para el manejo óptimo del Sistema Domótico instalado.	3 – crear un manual digital e impreso: Este manual debe contener la información necesaria para que quien se encargue de administrar el Sistema Domótico pueda acudir a él si se presenta alguna duda.

1.7 Estudio de Factibilidad.

La factibilidad hace referencia a la disponibilidad de recursos con los que se cuenta para la realización de metas y objetivos en un proyecto, y entre ellas están la factibilidad técnica y económica, las cuales en los siguientes apartados serán analizados técnica, económica y funcionalmente cada componente a utilizar.

1.7.1 Factibilidad económica

En este apartado se observa las opciones de adquisición que se tiene en el mercado en cuanto a componentes electrónicos y se estudiarán si estos son factiblemente económicos en su adquisición en una inversión total, para lo cual se mostrarán cuadros comparativos, con tres componentes de la misma categoría o con su misma funcionabilidad, pero de diferentes precios con la finalidad de ver cuál de estos es más factible adquirir.

Dichos cuadros contendrán el precio por unidad de cada uno, y el recargo de envío ya que algunos de estos componentes no se encuentran en nuestro país el costo total y al final de cada cuadro un análisis donde se determina económicamente la elección del componente a utilizar.

MICROCONTROLADOR			
Características	Arduino MEGA R3	PICDuino 18f2550	SANusb
Precio de Cotización unitario	\$20.40	\$65.00	\$79.90
Envío	\$10.00	\$6.28	\$7.72
Total	\$30.40	\$71.28	\$87.62
Lugar de cotización	http://www.ebay.com/itm/Arduino-Mega-R3-ATMega-2560-16AU-Cable-Included-/261176847205?pt=LH_DefaultDomain_0&hash=item3ccf5a6b65	http://produto.mercadolivre.com.br/MLB-458769804-arduino-com-pic-18f2550-kit-desenvolvimento-usb-shield-sbpic-_JM	http://produto.mercadolivre.com.br/MLB-457803879-placa-ferramenta-sanusb-pic-18f2550-e-18f4550-e-cabo-usb-_JM

Elección del microcontrolador: por costos más accesibles se determina que se utilizará el Arduino MEGA R3

MÓDULO RELÉ			
Características	MSP430 de 2 Canales	módulo de Relés 8 canales	MSP430 de 8 Canales
Precio de Cotización Unitario	\$8.00	\$8.99	\$11.80
Envío	ninguno	\$10.00	\$7.50
Total	\$8.00	\$18.99	\$19.30
Lugar de cotización	http://www.ingnac.com/	http://www.ebay.com/itm/8-Channel-5V-Relay-Module-Shield-for-Arduino-uno-1280-2560-ARM-PIC-AVR-DSP-/281068143455?pt=LH_DefaultDomain_0&hash=item4170f7835f	http://www.sunsky-online.com/view/233866/8+Channel+5V+Relay+Module+for+Arduino+AVR+PIC+ARM+MSP430.htm

Elección de módulo de relés: por costos más accesibles se determina utilizar el MSP430 de dos canales y el módulo de relés de 8 canales.

PROGRAMA DE DESAROLLO DE SOFTWARE			
Características	Visual Basic express 2010	Java SE Development Kit 7	Visual Estudio 2012 Ultimate
Precio de Cotización Unitario	Gratis	Gratis	\$180
Lugar de cotización	http://www.microsoft.com/visualstudio/eng/products/visual-studio-2010-express	http://www.oracle.com/technetwork/java/javase/downloads/java-se-jdk-7-download-432154.html	http://articulo.mercadolibre.com.ar/MLA-446259562-visual-studio-2012-ultimate-en-espanol-_JM

Elección del programa de desarrollo de software: por costos más accesibles se cree conveniente utilizar Visual Basic express 2010.

1.7.2 Factibilidad técnica

En este apartado se evalúan según bajo criterios selectivos si los componentes propuestos anteriormente cumplen ciertos requisitos técnicos básicos para el óptimo rendimiento del Sistema Domótico. En esta investigación las características adquisitivas de los componentes quedan bajo ningún criterio de selección ya que la selección será guiada por características técnicas funcionales.

MICROCONTROLADOR (PORCENTAJE MÍNIMO DE ACEPTACION = 85%)							
Características	Ponderación	Arduino MEGA R3		PICDuino 18f2550		SANusb	
		Valor	Porcentaje	Valor	Porcentaje	Valor	Porcentaje
Conexión USB	15%	2	30%	2	30%	2	30%
ID de programación	30%	2	60%	0	0%	0	0%
Tamaño Físico	20%	1	20%	2	40%	2	40%
Salidas Digitales	20%	2	40%	1	20%	1	20%
Alimentación secundaria	15%	2	30%	1	15%	0	0%
Total	100%		180%		105%		90%
			90%		52%		45%

Escala de puntuación: 0 = malo; 1 = regular; 2 =bueno

Elección del microcontrolador: Por evaluación técnica se elige el Arduino MEGA R3 ya que este posee los requisitos mínimos de funcionabilidad necesarios para ser considerado apto para el proyecto, posee una conexión USB para comunicación en serie con la computadora que manipulará el sistema embebido, su ID de programación es un entorno sencillo basado en el lenguaje de programación C++ y el software de desarrollo es de uso libre, este puede ser alimentado con voltajes de 5 a 12 voltios de corriente directa ya que en su circuitería aparte de una interface de programación posee un regulador de voltaje y posee 54 salidas multiplexadas digitales esenciales para la realización del proyecto.

MÓDULO RELÉ (PORCENTAJE MÍNIMO DE ACEPTACION = 85%)							
Características	Ponderación	MSP430 de 2 Canales		módulo de Relés 8 canales		MSP430 de 8 Canales	
		Valor	Porcentaje	Valor	Porcentaje	Valor	Porcentaje
Numero de vías de control	30%	2	60%	2	60%	2	60%
Voltaje que soportan los Contactos	30%	2	60%	2	60%	2	60%
Activación a tiempo real	10%	2	20%	2	20%	1	10%
Dimensiones	10%	2	20%	1	10%	1	10%
Corriente que soportan los Contactos	20%	2	40%	2	40%	2	40%
Total	100%		200%		190%		180%
			100%		95%		90%

Escala de puntuación: 0 = malo; 1 = regular; 2 =bueno

Elección del módulo relé: según el análisis técnico realizado a las propuestas, los que cumplen las expectativas son las tres opciones, ya que cumplen con todas las características necesarias pero se utilizarán dos módulos MSP430 de 2 Canales y el módulo de Relés 8 canales pues en sus descripciones técnicas pues ambos son similares ambos soportan voltajes de 110v hasta 220v en sus contactos de conmutación activación por pulsos de 5v a tiempo real, soporta hasta 10 amperios en sus contactos, pero se es más factible los mencionados anteriormente ya que por sus dimensiones físicas son ideales para sustituir las terminales de conmutación

manual en las aulas en el caso de del MSP430 de 2 canales y en caso del auditorio es más factible técnicamente controlar todas las luces desde un punto en específico.

PROGRAMA DE DESAROLLO DE SOFTWARE (PORCENTAJE MÍNIMO DE ACEPTACION = 85%)							
Características	Pond eraci ón	Visual Basic express 2010		Java SE Development Kit 7		Visual Estudio 2012 Ultimate	
		Valor	Porcenta je	Valor	Porcent aje	Valor	Porcenta je
Leguaje de programación	30%	2	60%	1	30%	1	30%
Interface de programación	10%	2	20%	1	10%	1	10%
Rapidez y sencillez de desarrollo	10%	2	20%	1	10%	1	10%
Herramientas de programación	20%	2	40%	2	40%	2	40%
Opciones graficas	30%	2	60%	2	40%	2	60%
Total	100 %		200%		140%		140%
			100%		70%		70%

Escala de puntuación: 0 = malo; 1 = regular; 2 =bueno

Elección del programa de desarrollo de software: luego de pruebas en los aspectos básicos y en sus características se decide utilizar el Visual Basic Exprees 2010 porque para llevar a cabo el proyecto, este cumple técnicamente con todos los requisitos necesarios en especial, al cumplir uno de los objetivos principales porque los programas generados desde este software de desarrollo suelen ser de interfaces simples con aplicaciones orientadas a objetos.

CAPITULO II: DOCUMENTACION TÉCNICA

2.1 Marco Teórico de Referencia

2.1.1 Sistemas y tipos de Sistemas

Un (Colmenares, 2010)sistema es un conjunto de partes o elementos organizados y relacionados que interactúan entre sí para lograr un objetivo. Los sistemas reciben (entrada) datos, energía o materia del ambiente y proveen (salida) información, energía o materia.

Un sistema puede ser físico o concreto (una computadora, un televisor, un humano) o puede ser abstracto o conceptual (un software).

Cada sistema existe dentro de otro más grande, por lo tanto un sistema puede estar formado por subsistemas y partes, y a la vez puede ser parte de un supersistema. Los sistemas tienen límites o fronteras, que los diferencian del ambiente. Ese límite puede ser físico (el gabinete de una computadora) o conceptual. Si hay algún intercambio entre el sistema y el ambiente a través de ese límite, el sistema es abierto, de lo contrario, el sistema es cerrado.

Tipos de sistemas

Pueden ser físicos o abstractos: Sistemas físicos o concretos: compuestos por equipos, maquinaria, objetos y cosas reales. El hardware.

- Sistemas abstractos

Compuestos por conceptos, planes, hipótesis e ideas. Muchas veces solo existen en el pensamiento de las personas. Es el software.

En cuanto a su naturaleza. Pueden ser cerrados o abiertos:

- Sistemas cerrados:

No presentan intercambio con el medio ambiente que los rodea, son herméticos a cualquier influencia ambiental. No reciben ningún recurso externo y no producen nada que sea enviado hacia fuera. En rigor, no existen sistemas cerrados. Se da el nombre de sistema cerrado a aquellos sistemas cuyo comportamiento es determinista y programado y que opera con muy pequeño intercambio de energía y materia con el ambiente. Se aplica el término a los sistemas completamente estructurados, donde los elementos y relaciones se combinan de una manera peculiar y rígida produciendo una salida invariable, como las máquinas.

- Sistemas abiertos:

Presentan intercambio con el ambiente, a través de entradas y salidas. Intercambian energía y materia con el ambiente. Son adaptativos para sobrevivir. Su estructura es óptima cuando el conjunto de elementos del sistema se organiza, aproximándose a una operación adaptativa.

- Sistemas aislados:

Son aquellos sistemas en los que no se produce intercambio de materia ni energía.

- Características de los Sistemas

Sistema es un todo organizado y complejo; es un conjunto de objetos unidos por alguna forma de interacción o interdependencia. Los límites o fronteras entre el sistema y su ambiente

PROPOSITO U OBJETO: Todo sistema tiene uno o algunos propósitos.

Los elementos (u objetos), como también las relaciones, definen una distribución que trata siempre de alcanzar un objetivo.

- GLOBALISMO O TOTALIDAD: Un cambio en una de las unidades del sistema, con probabilidad producirá cambios en las otras. El efecto total se presenta como un ajuste a todo el sistema. Hay una relación de causa / efecto.

- ENTROPIA: Es la tendencia de los sistemas a desgastarse, a desintegrarse, para el relajamiento de los estándares y un aumento de la aleatoriedad. La entropía aumenta con el correr del tiempo. Si aumenta la información, disminuye la entropía, pues la información es la base de la configuración y del orden. De aquí nace la negentropía, o sea, la información como medio o instrumento de ordenación del sistema.

- HOMEOSTACIA: Es el equilibrio dinámico entre las partes del sistema. Los sistemas tienen una tendencia a adaptarse con el fin de alcanzar un equilibrio interno frente a los cambios externos del entorno. Una organización podrá ser entendida como un sistema o subsistema o un supersistema, dependiendo del enfoque.
(Colmenares, 2010)

2.1.2 Definición de Domótica

Los diccionarios Franceses incorporaron el concepto “domotique” a partir de 1988. Esta palabra, traducida al castellano por Domótica, la cual es originaria de la palabra “domus” (de la que deriva la raíz domo que quiere decir casa) y de la palabra francesa “informatique” (de la que se deriva la palabra informática) o, según otros autores, “robotique” (Robótica). Este término de uso común en España, no ha conseguido imponerse en diversos países de Hispanoamérica; donde aún se implementa el término “inteligente”, sin avanzar a la diferenciación entre Domótica e Inmótica. (Moya, 2010)

Domotique, como lo define la enciclopedia Larousse la cual en 1988 conceptualiza dicho termino como: “El concepto de vivienda que integra todos los automatismos se refiere al conjunto de técnicas utilizadas, en materia de seguridad, del ahorro energético y de las facilidades de comunicación

Por lo que Domótica se refiere al conjunto de técnicas utilizadas para la automatización de las gestiones en una vivienda o recinto.

Dicho esto y para una mejor e integra comprensión del concepto se citará a continuación un concepto establecido por El CEDOM (Asociación Española de Domótica). La cual lo establece como “La incorporación al equipamiento de nuestras viviendas y edificios de una sencilla tecnología que permita gestionar de forma energéticamente eficiente, segura y confortable para el usuario. (Romero, Vázquez, & Castro, 2011)

Como un solo concepto este integra ciertas áreas las cuales se vuelven un requisito indispensable conocerlas con la finalidad de entender la estructuración e implementación del Sistema Domótico a implementar.

Así mismo dentro del concepto de Domótica existe diversidad de aspectos que integran dicho concepto, y que pasan a ser áreas fundamentales de la Domótica tales como sus características, topología de la red, los tipos de arquitectura, medios de transmisión, etc. Todos estos descritos en los siguientes apartados

2.1.2 Características de un Sistema Domótico

A continuación las principales características o rasgos generales que un Sistema Domótico debe tener:

- Simple y fácil de utilizar

El sistema de control debe tener dos características como eje principal, pues tiene que ser simple y fácil de utilizar, para que el usuario a la hora de utilizar encuentre en él una interfaz simple e intuitiva, generando así altos niveles de funcionabilidad y confort.

- Flexible

A la hora de ser desarrollado se tiene que tomar en cuenta que dicho sistema no está libre de modificaciones futuras, no obstante dichas modificaciones implicarían reutilizar los componentes ya instalados y dichas modificaciones deben ser realizadas sin un costo elevado ni un esfuerzo grande.

- Modular

El sistema de control debe ser modular; con la finalidad de evitar fallos en las instalaciones propias del edificio, y en caso de daños al sistema de control debe permitir cambios sin afectar la infra estructura, además debe permitir la adhesión sencilla de nuevos servicios.

- Integral

El sistema debe permitir el intercambio de información y la comunicación entre diferentes áreas de gestión del edificio, de tal manera que si hay en las instalaciones del recinto un sistema más sea este capaz de ser integrado.

2.1.3 Topología de la red

La topología (Romero, Vázquez, & Castro, 2011) de la red, también denominada topología de cableado, se define como la distribución física de los elementos de control, con respecto al medio de comunicación (cable).

Existe diversidad de distribuciones físicas de control o topologías, utilizadas según la necesidad que se desee cubrir entre ellas están.

- Topología en Estrella.
- Topología en Bus.
- Topología en Anillo.
- Topología en Árbol.

Para la implementación de este proyecto se utilizara la topología tipo Estrella, pues en esta todos los elementos están unidos entre sí a través del controlador principal.

Pues este tipo de Topología permite tener como eje central el controlador y por medio de este recibir órdenes desde la computadora de control para luego este procesar los datos enviados y luego tomar decisiones pertinentes y enviar los pulsos necesarios para activar o desactivar los actuadores.

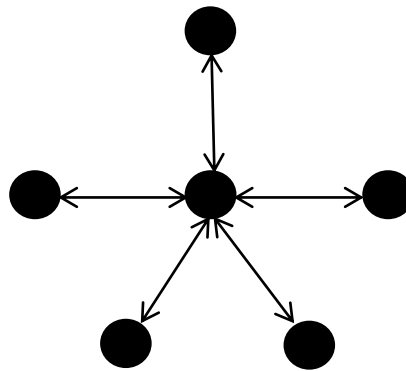


Figura 2.1.3.1 Representación de la Topología tipo estrella (Romero, Vázquez, & Castro, 2011)

Ventajas

- Facilidad de integración de nuevos componentes.
- Al detectar un fallo de un elemento (siempre y cuando este no sea el controlador) no afecta al resto
- Fácil reparar en caso de fallos tanto en controlador o en actuadores
- Fácil y sencilla instalación

Desventajas

- Sistema dependiente del funcionamiento óptimo del controlador central.
- Necesidad de un interfaz de usuario.
- Se produce un cuello de botella en el elemento central por el aglutinamiento de información.

Se decide utilizar dicho tipo de topología pues cumple los bloques funcionales principales los cuales son:

- Censar entorno
- Procesar datos recibidos
- Realizar acciones física mediante datos recibidos

2.1.4 Tipos de Arquitectura

Cuando en la Domótica se utiliza la terminología Arquitectura, esta se entiende como:

“La forma en la que se especifica el modo en que los diferentes elementos de control del sistema se van a ubicar” (Romero, Vázquez, & Castro, 2011).

Dentro de la arquitectura existen algunos tipos establecidos tales como la topología centralizada y distribuida, ambas utilizadas usualmente en la creación de Sistemas Domóticos

Para la implementación de este proyecto se utilizará la arquitectura Centralizada.

- Arquitectura Centralizada

Este tipo de arquitectura tiene la peculiaridad de controlar y supervisar sensores, actuadores, etc. Todo de forma cableada hacia el sistema de control o interface de usuario, todos los sensores capturan la información necesaria para luego ser enviada al controlador, para que este tome las decisiones y se las comunique a los actuadores, siendo así este aparte principal del Sistema Domótico.

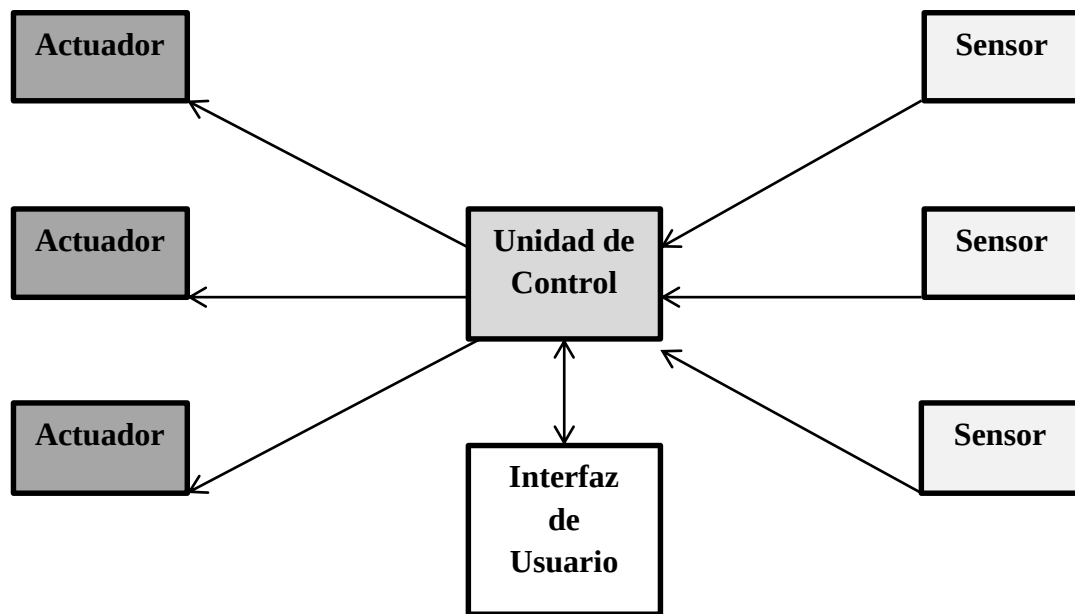


Figura 2.1.4.1 Arquitectura Centralizada (Romero, Vázquez, & Castro, 2011)

2.1.5 Medios de Transmisión

Cuando se habla de los medios de transmisión se quiere dar a entender el tipo de conexión a utilizar ya sea por medio de cable o por conexión inalámbrica, ambas utilizadas para la creación de Sistemas Domóticos.

2.1.5.1 Medios de transmisión Alámbricos

- **Corrientes Portadoras**

Utilizan líneas de distribución ya existentes en la vivienda para la transmisión de datos. Las más utilizadas son las líneas de distribución de energía eléctrica, aunque también se está comenzando a utilizar la línea telefónica tradicional. Si bien no es el medio más adecuado para la transmisión de datos, si es una alternativa a tener en cuenta para las comunicaciones domesticas dado el bajo coste que implica su uso, ya que se trata de una instalación ya existente. Las especiales características de este medio lo hacen idóneo para su uso en las instalaciones domesticas ya existentes. Sus principales ventajas son el coste de la instalación y la facilidad de conexionado. Y sus inconvenientes son la poca fiabilidad de la transmisión de los datos y la baja de transmisión.

- **Soporte metálico**

Son cables metálicos de cobre como soporte de transmisión de las señales eléctricas que procesa. En general se pueden distinguir en dos tipos de cables metálicos:

- **Par metálico**

Los cables formados por varios conductores de cobre pueden dar soporte a un amplio rango de aplicación. Este tipo de cables pueden transportas: datos, voz y alimentación. Los denominados cables de pares formados por cualquier combinación de los tipos de conductores que a continuación se detallan:

Tipos de conductores metálicos

- Cables formados

Por un solo conductor con un aislamiento exterior (por ejemplo los utilizados para la transmisión de las señales telefónicas).

- Par de cables

Cada uno de los cables está formado por un arrollamiento helicoidal de varios hilos de cobre (por ejemplo los utilizados para la distribución de señales de audio).

- Par apantallado

Formado por dos hilos recubiertos por un trenzado conductor en forma de malla cuya misión consiste en aislar las señales que circulan por los cables de las interferencias electromagnéticas exteriores (por ejemplo los utilizados para la distribución de sonido de alta fidelidad o datos).

- Par trenzado

Está formado por dos hilos de cobre recubiertos cada uno por un trenzado en forma de malla (por ejemplo los utilizados para interconexión de ordenadores).

- Coaxial

Un par coaxial es un circuito físico asimétrico, constituido por un conductor filiforme que ocupa el eje longitudinal del otro conductor en forma de tubo, se mantiene el carácter coaxial de ambos mediante un dieléctrico apropiado.

- Fibra Óptica

La fibra óptica está constituida por un material dieléctrico transparente, conductor de luz, compuesto por un núcleo con un índice de refracción menor que el del revestimiento que envuelve a dicho núcleo. Estos dos elementos forman una guía para que la luz se desplace por la fibra. La luz transportada es generalmente infrarroja, y por lo tanto no es visible por el ojo humano. (Romero, Vázquez, & Castro, 2011)

La utilización de cualquiera de los tipos de conducción metálicos variara en gran manera por sus costos monetarios y por su características técnicas pero funcionalmente, dependerá de la implementación que se le dará a este.

2.1.6 Áreas a Gestionar

La creación de un Sistema Domótico implica verificar las áreas de aplicación dentro de la casa o recinto, manejando ciertas áreas que pueden ser agrupadas para identificar y atender de la mejor manera los inconvenientes en ella.

Dentro de las áreas a gestionar están:

- Gestión de la energía

Se encarga del ahorro energético y el uso óptimo de los suministros eléctricos mediante la utilización de sistemas de control automatizado o programable entre ellos interfaces de control programables por el usuario como temporizadores, relojes programables, etc. Con la finalidad de priorizar funciones tanto adentro como afuera

del hogar; encargándose de apagar electrodomésticos cuando están fuera de uso con el propósito de ahorrar energía eléctrica.

- Gestión de confort

Cuando se hace referencia sobre las gestiones de confort, se debe de entender que esta implica la comodidad del usuario como la cantidad de esfuerzo físico a la hora de utilizar el sistema.

Las aplicaciones más comunes de este área son: el encendido y apagado parcial, total y sectorizado de luminarias desde un punto o automáticamente, manejo de electrodomésticos aperturas de puertas y persianas o portones, etc. Todo lo que facilite al usuario la realización de las actividades cotidianas dentro del hogar.

2.1.7 Componentes básicos de un Sistema Domótico

Dentro de los Sistemas Domóticos existen componentes, los cuales se encargan de integrar el sistema, ya que estos se utilizan para la recepción, transmisión, procesamiento de datos, para que luego estos sean acciones físicas como movimiento de válvulas hidráulicas, o el simple hecho de conmutar una señal.

Los componentes que integran un Sistema Domótico se clasifican de la siguiente manera:

- Tipo de señales
- Sensores
- Acondicionamiento de señal
- Actuadores
- Interfaces

- Infraestructura
- Unidad de control
- Software de Gobierno

2.1.7.1 Tipos de Señal

En un Sistema Domótico para denominarle *“inteligente”* debe comunicarse con el mundo exterior y para ello utiliza señales físicas las cuales por medio de los sensores son transformadas a señales eléctricas de las cuales pueden llegar a ser Digitales o Análogas. Dichas señales indican al controlador las acciones a tomar si sus lecturas están dentro del firmware de control.

- Señales Digitales

Las señales Digitales o conocidas también como rectangulares son muy utilizadas para realizar determinadas mediciones, e implementar controles en sistemas de conmutación. Se caracteriza por tener solamente dos valores posibles.

- Señales Análogas

Este tipo de señal son variables eléctricas que evolucionan en el tiempo en forma análoga a alguna variable física. Estas señales pueden presentarse en la forma de una corriente, una tensión o una carga eléctrica. Varían en forma continua entre un límite inferior y un límite superior.

2.1.7.2 Sensores

Son componentes electrónicos los cuales se encargan de transformar magnitudes físicas y químicas a señales de tipo eléctrica, con la finalidad de proporcionar

“sentidos”. Estos pueden ser utilizados tanto en la Domótica como en otras áreas de implementación.

2.1.7.3 Acondicionamiento de señal

Está compuesto por circuitos eléctricos, cuya misión es adaptar el tipo y nivel de señal eléctrica de salida de un sensor, al nivel y tipo óptimo para la entrada del controlador.

Este tipo de circuitería son los encargados de decodificar señales con cadenas de datos o caracteres a niveles lógicos o lenguajes de alto nivel según sea requerido por el control y son también los encargados de cambiar o reducir según sea el tipo de sensor utilizado.

2.1.7.4 Actuadores

Son los dispositivos electromecánicos que actúan sobre el medio exterior y afectan físicamente al edificio. Convierten una magnitud eléctrica en otra de otro tipo (mecánica, terminal, etc.), realizando, de alguna manera, un proceso inverso al de los sensores. Los actuadores pueden mantener niveles de salida continuos o discretos. Ejemplo de actuadores pueden ser el motor de una persiana, los contadores de un circuito de iluminación, lámparas, radiadores, sirenas, etc.

- **RELES.**

Son interruptores que permiten conmutar circuitos de potencia más elevada mediante una señal de baja potencia. Generalmente es un dispositivo electromecánico que basa su funcionamiento en la actuación de un solenoide recorrido por una corriente continua. Al pasar la corriente por la bobina se magnetiza el núcleo de hierro y atrae a la armadura, lo que provoca la apertura y el cierre de contactos electrónicos.

Mucho de los estándares existentes disponen de módulos especiales que incorporan relés para conmutar carga de diversos tipos. Es recomendable leer las hojas de catálogo para conocer la carga máxima conmutable y el tipo de la misma.

- **CONTADORES**

Son relés de potencia. Una bobina se excita con la tensión de alimentación y cierra unas platinas de cobre, cuya anchura y disposición permite el paso de más o menos corriente.

- **REGULADORES O DIMMERS**

Son dispositivos basados en semiconductores, como los diacs o los triacs, que permiten regular la potencia que llega a una carga. En instalaciones domésticas se suelen utilizar para regular la intensidad de bobinas, luminarias, etc. Para su uso hay que prestar especial atención a las especificaciones de los fabricantes, no solo para conocer la carga máxima que puede conectárseles sino también el tipo de la misma: resistiva, inductiva, tubos fluorescentes, halógenos, etc.

- **ELECTROVALVULAS**

Son válvulas cuya apertura es controlada mediante una señal eléctrica externa. Se utiliza principalmente para controlar caudales de líquidos o gases; en edificios se usan para control de gas o agua, así como en sistemas de aire acondicionado. Pueden ser analógicas o digitales binarias, es decir, de paso variable o proporcionales (en cuyo caso también se denominan servo válvulas) o todo/nada.

Está formada por dos piezas, el cuerpo, que se ajusta a la tubería, y el cabezal, encargado de mover el dispositivo de apertura o cierre.

- **MOTORES ELECTRICOS**

Convierten energía eléctrica en mecánica para generar, de esta forma, un movimiento. Como ejemplo de aplicaciones en Inmótica se pueden citar los ventiladores, bombas, posicionadores, etc. Los tipos más comunes de motores son los de corriente continua, en las que la variación de tensión controla la velocidad del mismo. Son precisos y de gran rapidez, pero de baja potencia. También se suelen utilizar los de corriente alterna, muy útiles en el ámbito Domótico debido a que no necesita de fuentes de alimentación adicionales diferentes de la propia red. En ellos, la velocidad depende de la frecuencia de la tensión de alimentación. Los motores paso a paso suelen utilizarse como posicionadores de precisión, en algunos casos acompañando a servo válvulas (giran un determinado ángulo a cada secuencia de impulsos).

- **RESISTENCIA ELECTRICA**

Se utilizan para elevar la temperatura del medio donde se encuentran. Su fundamento es hacer pasar a través de conductor una corriente eléctrica que produce el calentamiento de conductor. Algunos ejemplos son los radiadores, calefactores, secadores, etc. (Romero, Vázquez, & Castro, 2011)

2.1.7.5 Interfaces

Son circuitos determinados a adaptar los niveles de voltaje de salida del controlador ya que por lo general estos no vienen con la adecuación necesaria, como para soportar niveles altos de voltaje si el caso fuese de conmutar o la intensidad necesaria como para poner en marcha un motor, así que la finalidad de este tipo de circuitos es acondicionar o amplificar las terminales de control.

2.1.7.6 Infraestructura

La infraestructura de un Sistema Domótico es la encargada de llevar la información que producen los sensores hasta el sistema de control, y de alimentarlos con una tensión eléctrica adecuada, es decir, el cableado de datos y el cableado de alimentación. (Romero, Vázquez, & Castro, 2011)

A la hora de implementar la infraestructura en un Sistema Domótico se debe tomar en cuenta que existen tres tipos de esta las cuales son:

- **Bus**

Un medio de transmisión común recorre todos los dispositivos

- **Centralizada**

Todos los dispositivos conectados a la unidad central.

- **Mixta**

Híbrida entre las dos anteriores (Romero, Vázquez, & Castro, 2011)

Para la implementación del proyecto se utilizara una infraestructura estilo Centralizada.

2.1.7.7 Unidad de Control

La unidad de control en un Sistema Domótico es aquella que se encarga de procesar los datos recibidos y de cumplir rutinas previamente establecidas. Este cuenta con una memoria reprogramable, en la que almacena las secuencias a realizar según sea los datos que recolecten los sensores para luego enviar señales de accionamiento a los actuadores.

Generalmente se utilizan microprocesadores, microcontroladores, o Autómatas programables (PLC) como controladores principales de los Sistemas Domóticos. Y algunos poseen interfaces de usuarios más fáciles de utilizar pues poseen en ellos pantallas o comunicación vía PC con el usuario.

2.1.7.8 Software de Control

El Software de control o también conocido como Firmware es el que almacena el programa de gobierno, es este al final quien el controlador obedece pues en este están escritos todos los parámetros a seguir indiferentemente o no si se cumplen algunas sentencias dentro de él. Se encarga de manipular el Hardware. (Romero, Vázquez, & Castro, 2011)

2.1.8 Manuales y tipos de manuales

- **Definición de Manual**

Los (Andres, 2010) manuales son textos utilizados como medio para coordinar, registrar datos e información en forma sistémica y organizada. También es el conjunto de orientaciones o instrucciones con el fin de guiar o mejorar la eficacia de las tareas a realizar.

- **Tipos de Manuales**

Pueden distinguirse los manuales de:

Organización: este tipo de manual resume el manejo de una empresa en forma general. Indican la estructura, las funciones y roles que se cumplen en cada área.

Departamental: dichos manuales, en cierta forma, legislan el modo en que deben ser llevadas a cabo las actividades realizadas por el personal. Las normas están dirigidas al personal en forma diferencial según el departamento al que se pertenece y el rol que cumple

Política: sin ser formalmente reglas en este manual se determinan y regulan la actuación y dirección de una empresa en particular.

Procedimientos: este manual determina cada uno de los pasos que deben realizarse para emprender alguna actividad de manera correcta.

Técnicas: estos manuales explican minuciosamente como deben realizarse tareas particulares, tal como lo indica su nombre, da cuenta de las técnicas.

Bienvenida: su función es introducir brevemente la historia de la empresa, desde su origen, hasta la actualidad. Incluyen sus objetivos y la visión particular de la empresa. Es costumbre adjuntar en estos manuales un duplicado del reglamento interno para poder acceder a los derechos y obligaciones en el ámbito laboral.

Puesto: determinan específicamente cuales son las características y responsabilidades a las que se acceden en un puesto preciso.

Múltiple: estos manuales están diseñados para exponer distintas cuestiones, como por ejemplo normas de la empresa, más bien generales o explicar la organización de la empresa, siempre expresándose en forma clara.

Finanzas: tiene como finalidad verificar la administración de todos los bienes que pertenecen a la empresa. Esta responsabilidad está a cargo del tesorero y el controlador.

Sistema: debe ser producido en el momento que se va desarrollando el sistema. Está conformado por otro grupo de manuales.

Calidad: es entendido como una clase de manual que presenta las políticas de la empresa en cuanto a la calidad del sistema. Puede estar ligado a las actividades en forma sectorial o total de la organización. (Andres, 2010)

Para la implementación del proyecto se hará uso de un manual del tipo usuario ya que ira dirigido, al uso del programa para el control del sistema Domótico.

2.2 Marco Teórico de Solución

La instalación del Sistema Domótico en el Instituto Nacional de Aguilares será sectorizada en tres áreas las cuales son

- Luminarias del auditorio general.
- Control de los ventiladores de las aulas alrededor del auditorio.
- Timbre general de la institución.

Para el manejo de dichos sectores se desarrolló una aplicación en Visual Basic 2010 capaz de realizar acciones tales como:

- Encendido y Apagado de las luminarias de forma sectorizada del auditorio general
- Encendido y apagado de los ventiladores de las aulas alrededor del auditorio quitando así el control manual de las aulas evitando que los alumnos lo dañen.
- Control del timbrado para que este suene a las horas establecidas y no hayan retrasos ni pérdidas de tiempo en las horas clases

Dicho programa cuenta con una interface de usuario sencilla que consta con:

- Un Reloj a tiempo real
- Un botón al lado derecho que abre una ventana donde se encuentran los controles de conmutación las luminarias de los sectores del auditorio.
- Un botón al lado izquierdo que abre una ventana donde se encuentran los controles de conmutación de los Ventiladores de los grados.

- Un botón que al centro que abre una ventana donde se encuentran los controles de programación del timbre.



Imagen 2.2.1 Interface principal del programa

2.2.1 Luminarias del auditorio general

El auditorio del Instituto Nacional de Aguilares, cuenta en su instalaciones eléctricas con un sistema manual de conmutación para el manejo de la luminarias, posee 4 sectores principales de iluminación.



Imagen2.2.1.1 Sector 1 del auditorio general



Imagen2.2.1.2 Sector 2 del auditorio general



Imagen2.2.1.3 Sector 3 del auditorio general



Imagen2.2.1.4 Sector 4 del auditorio general



Imagen 2.2.1.6 ventana del programa que controlará las luminarias

Se pretende centralizar el control de las luminarias desde la PC, para que el manejo de éstas sea desde un punto exacto sin necesidad de trasladarse de un pasillo al otro para apagarlas.

2.2.2 Control de los ventiladores

En el instituto Nacional de Aguilares hay 12 salones alrededor del auditorio, de los cuales cada uno tiene un control manual del encendido de la ventilación.



Imagen 2.2.2.1 Estado de las instalaciones actuales de los conmutadores



Imagen 2.2.2.2 Estado de las instalaciones actuales de los conmutadores



Imagen 2.2.2.3 Estado de las instalaciones actuales de los conmutadores



Imagen 2.2.2.4 Estado de las instalaciones actuales de los conmutadores

Como se puede apreciar en las imágenes anteriores los controles de conmutación de los ventiladores activados de forma manual, algunos dañados por los alumnos y muchas veces por descuido del personal de ordenanza de la institución y por el mal uso de los alumnos, estos permanecen encendido en horas ex-aula desperdiciando energía eléctrica.



Imagen 2.2.2.5 Ventana del programa que controlara los ventiladores de las aulas y el auditorio.

Por medio de esta área del programa se pretende centralizar en control de la ventilación en las aulas para que no se malgaste la energía y no se dañe las instalaciones eléctricas de la institución.

2.2.3 Timbre general de la Institución

El instituto Nacional de Aguilares cuenta con timbre el cual se activa manualmente por un pulsador este dispositivo tiene que ser activado desde un punto específico en

el cual no siempre hay alguien que lo active, y muchas veces se pierden horas clases por no contar con la suficiente coordinación afectando directamente a la población estudiantil de la institución.



Imagen 2.2.3.2 Control Manual de timbre

Como solución, El programa lleva dentro de sí un timer o temporizador, el cual se encargara de llevar la hora a tiempo real con el de la BIOS de la computadora que tenga el programa, el cual se sincronizará con un reloj en el controlador del sistema Domótico.



Imagen 2.2.3.3 Ventana que se encarga de programar el timbre de la institución

La funcionalidad de esta área dependerá del botón “Sincronizar Hora”, ya que por medio de éste el reloj a tiempo real de la computadora podrá ser sincronizado con el reloj del controlador para luego programar los horarios del cambio de clases y recesos de la institución, parece una opción repetitiva pues se puede llegar a pensar que con cada inicialización del sistema se tendría que programar pero al contrario, solo se necesita programar una tan sola vez, ya que los horarios programados junto con la hora del reloj a tiempo real quedaran guardados en la memoria EPROM del controlador y serán comparados aunque la interface de usuario sea cerrada o apagada.

Para la instalación del sistema en la institución se tomaron en cuenta algunos aspectos y se delimitó que la institución proporcionaría una pequeña bodega que es donde el ordenanza en turno pasa el tiempo que no está pendiente de las actividades propia de su trabajo, dicha bodega está ubicada a un costado el auditorio.

En los siguientes planos se muestran las instalaciones del Instituto Nacional de Aguilares, donde quedara instalado el sistema y como se pretende implementar la topología.

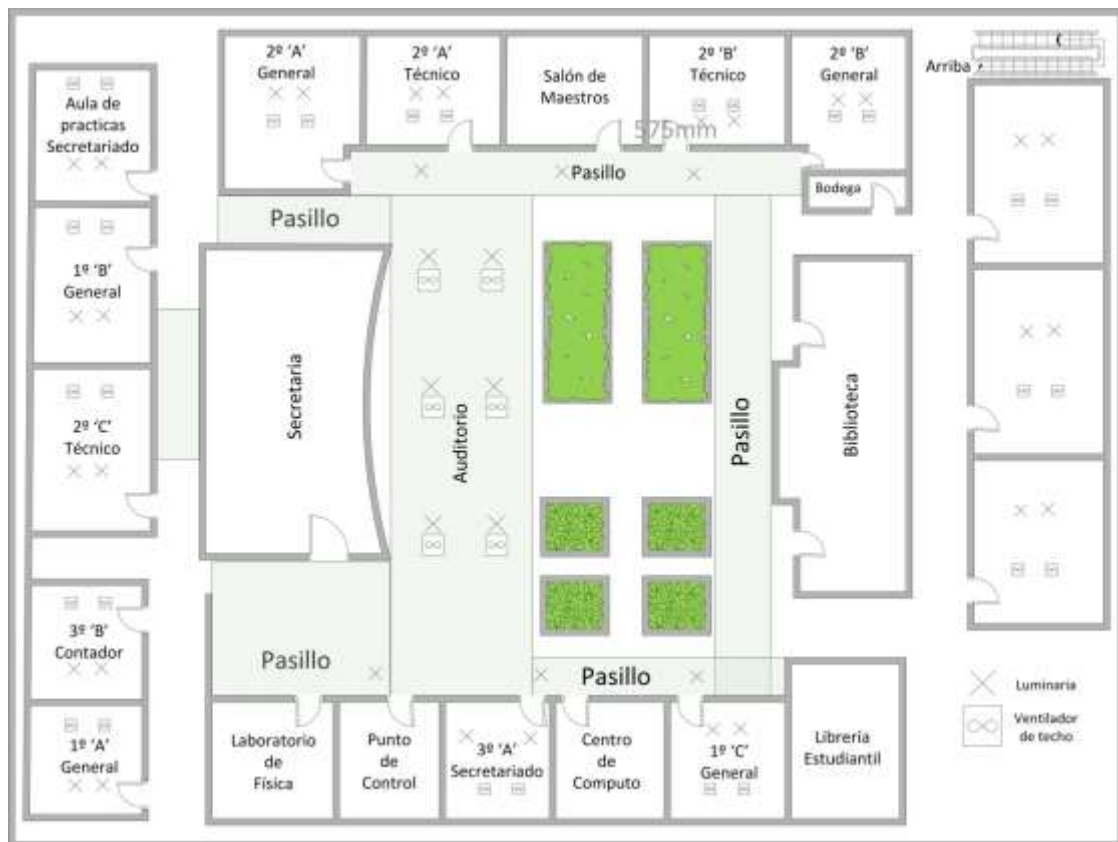


Imagen 2.2.3.4 plano de la infraestructura del Instituto Nacional de Aguilares

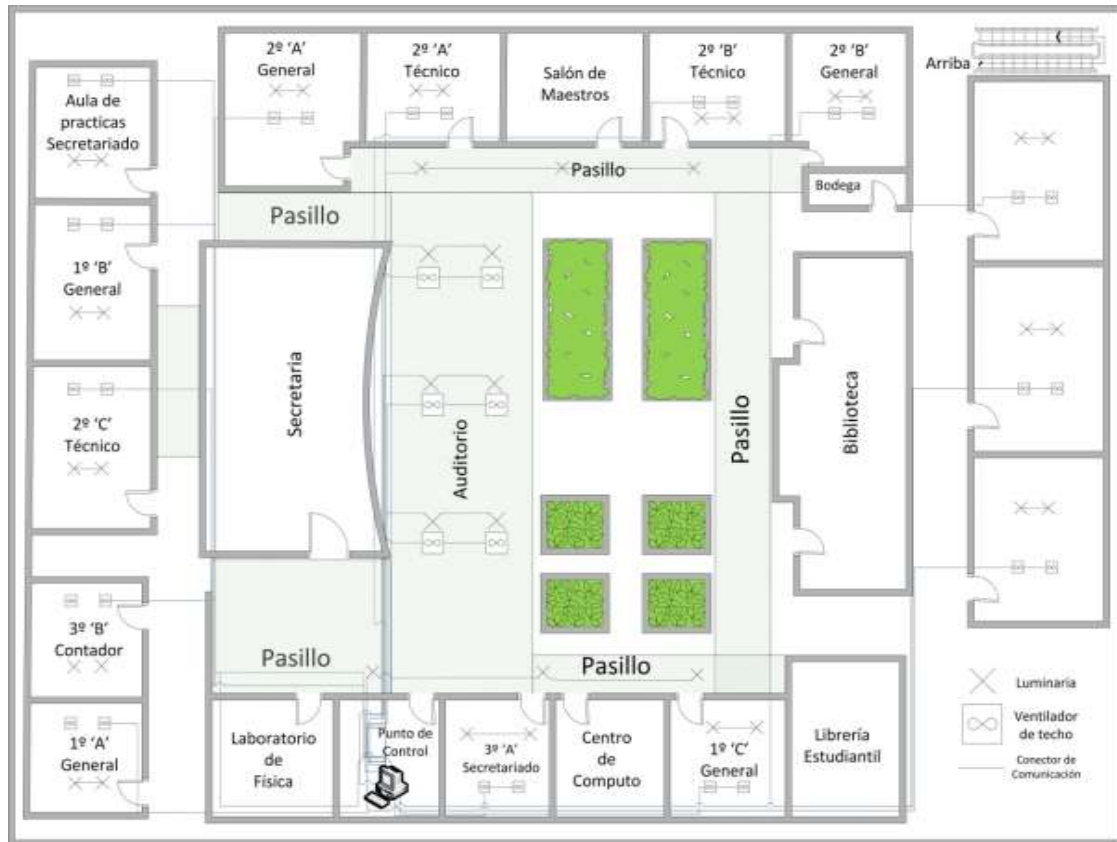


Imagen 2.2.3.5 plano de la infraestructura del Instituto Nacional de Aguilares, con la implementación de la topología.

2.2.4 Manual de usuario

El siguiente manual está dirigido para los encargados del área de administración de servicios de personal, quienes son los encargados de velar por buen uso los bienes de la institución y quienes en primera instancia serán los encargados de usar el Sistema Domótico. Dentro de este manual se encuentran: instrucciones a seguir para instar la interface de usuario en la computadora que controlará el Sistema Domótico, las instrucciones para el uso y configuración adecuada de la interface de usuario el

plano de las instalaciones de la institución y como está estructurada la topología de la instalación, el circuito electrónico del sistema.

Capítulo I: Instalación de la Unidad de Control.

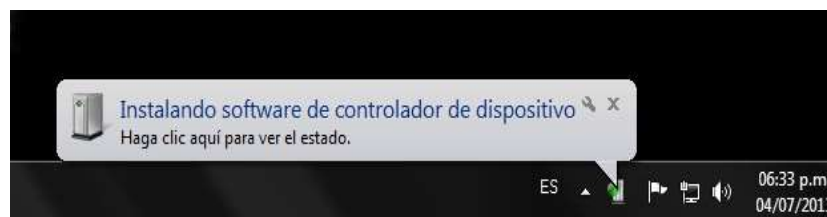
Al instalarse la unidad de control en la interface de usuario existen pasos a seguir para que este funcione adecuadamente, esto ayuda al usuario a identificar el puerto COM a utilizar.

Instrucciones para instalar el controlador en la interface de usuario

NOTA: Los siguientes pasos a realizar, necesitan que la computadora este encendida.

1- Conectar Cable USB de la unidad de control en cualquiera de los puertos USB disponible en la computadora.

Al conectar la unidad de control a la computadora aparecerá al costado inferior derecho del monitor un recuadro indicando que el controlador ha sido reconocido.



La Instalación de la unidad de control se lleva de forma manual, por lo que seguido de este aviso aparecerá otro en forma de advertencia diciendo que el software de control del dispositivo no se pudo instalar.



Este paso es solamente para identificar que la unidad de control si fue detectada aunque esta no se haya instalado.

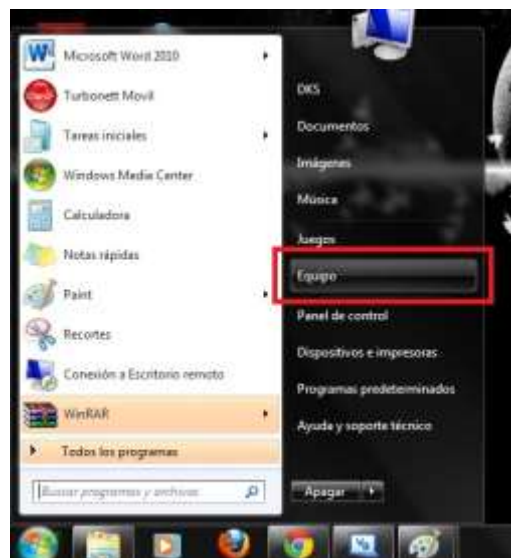
Por lo que se deben seguir los pasos detallados a continuación para la instalación del software de controlador.

2- Dar clic en el icono de “Inicio”



Se desplegará la barra de inicio

3- ubicar el puntero sobre el icono “Equipo”



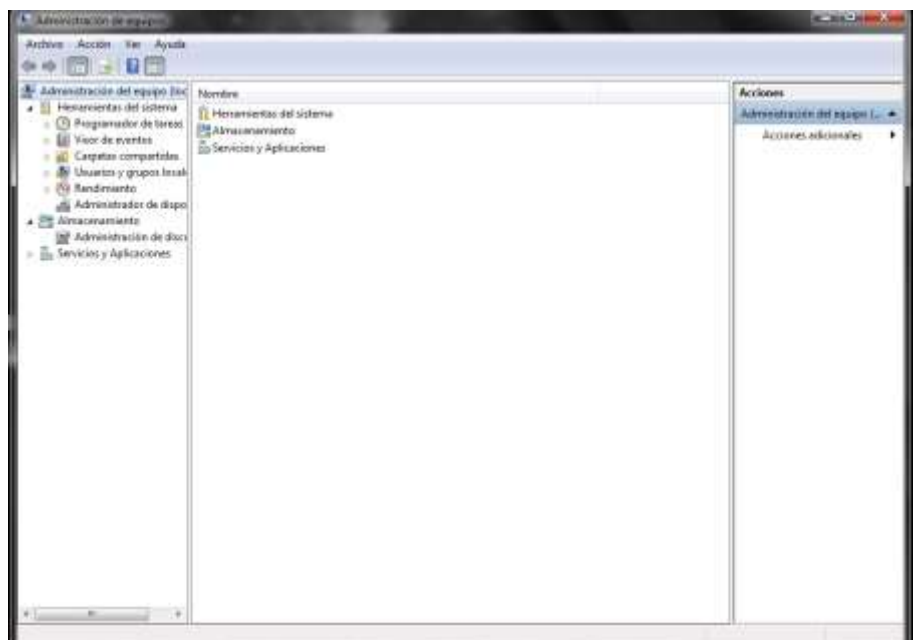
4- Clic derecho en “Equipo”

Al realizar dicha acción saldrá de forma emergente un recuadro de opciones seleccionar la opción “Administrar” y dar clic izquierdo sobre él.



NOTA: Esperara a que se habrá la ventana de “Administración de Equipos” esto puede tardar un par de segundos

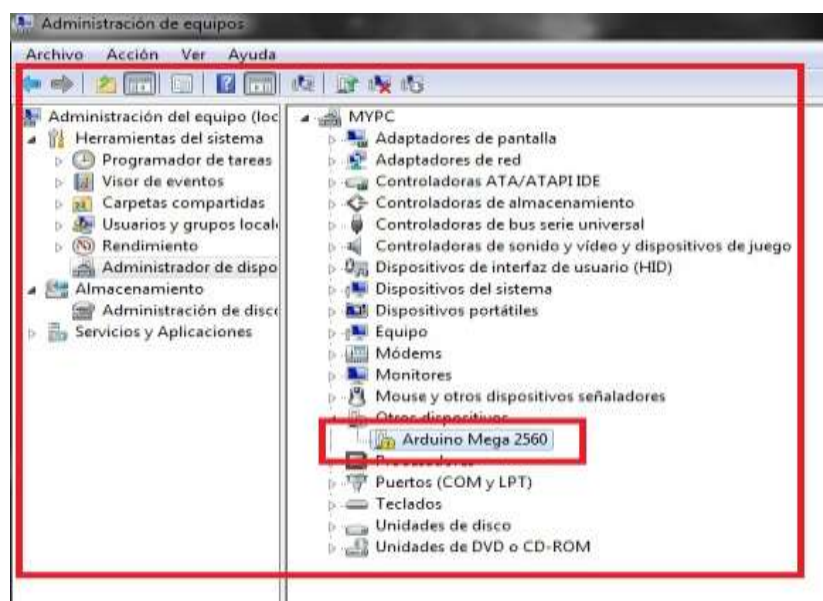
5- ubicarse en el “Administración de Equipos” mostrado en la siguiente imagen.



6- Ubicarse y dar clic sobre la raíz de “Administrar Dispositivos”

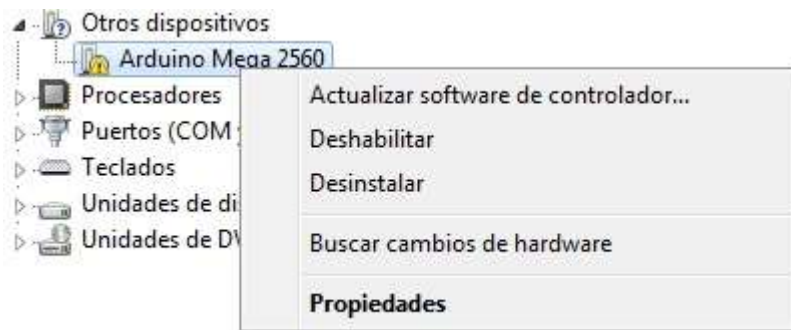


Al dar clic en esta opción se desplegará un menú de raíz donde están una lista de todos los dispositivos instalados en la computadora y es desde aquí donde se puede instalar y desinstalar los driver de estos.



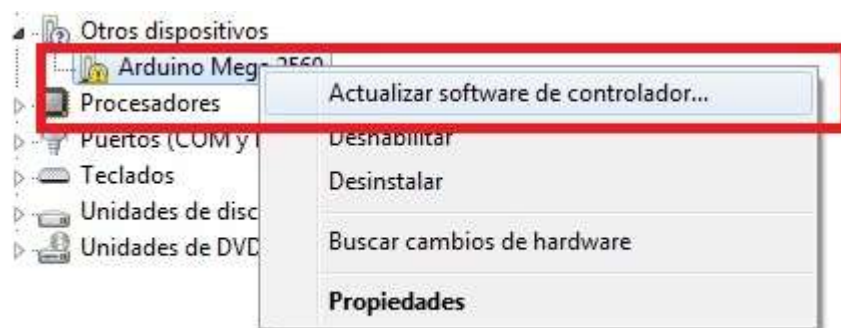
Luego de observar que el menú se ha desplegado, observara el nombre del controlador junto a un icono de color amarillo, ya que por defecto Windows muestra los depósitos sin software de control o driver con ese icono

7- Ubicarse en el controlador y dar clic derecho sobre él.

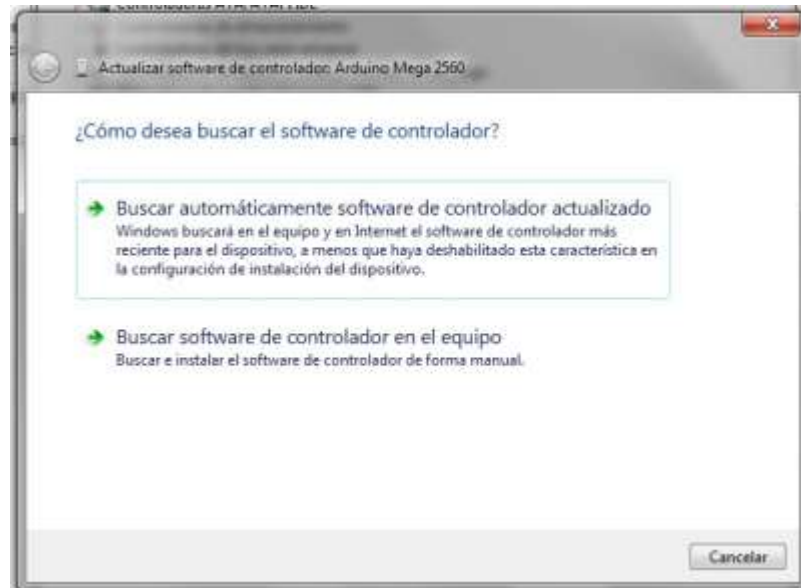


Al dar clic derecho sobre el controlador aparecerá un recuadro en el que se encuentran las opciones de instalación del software de control.

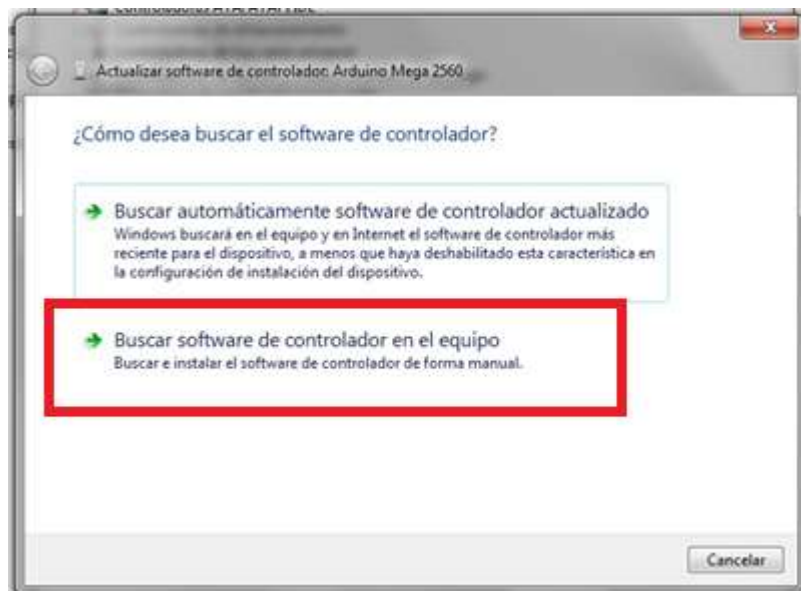
8- Dar clic en la opción “Actualizar software de controlador...”



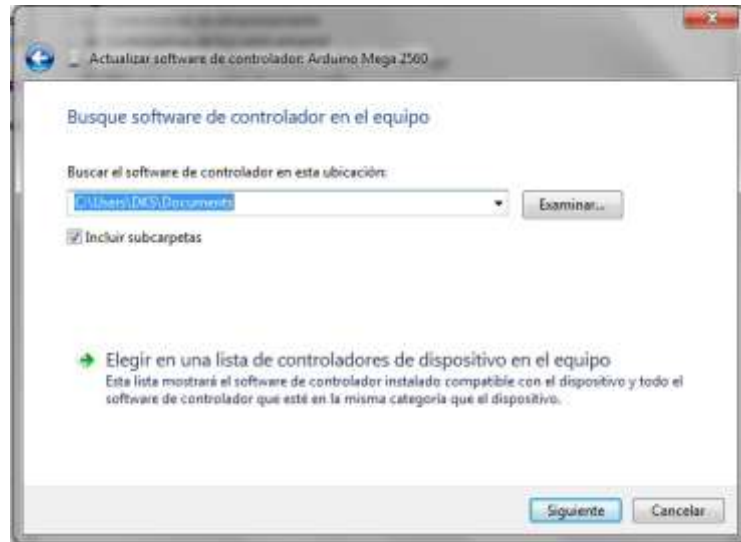
Se abrirá una ventana que pregunta de qué forma se desea actualizar el software de control.



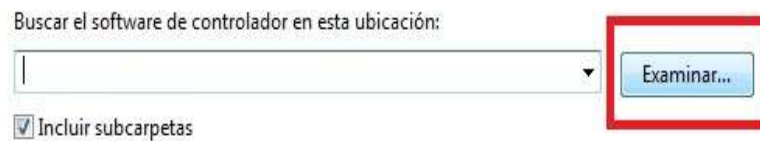
9- dar clic en “Buscar software de controlador en el equipo”



Aparecerá la ventana con las opciones de búsqueda en las carpetas de la computadora



10- Ubicarse en la opción para “Buscar el software de controlador en esta ubicación” y dar clic en el botón “Examinar...”



Se abrirá una ventana mostrando algunas carpetas llamada “Buscar carpeta”



Dentro de esta ventana se encuentran las ubicaciones de los discos duros instalados en la computadora

11- Ubicar el disco local "C"



12- Desplegar el contenido en el disco local "C"



13- Ubicarse, seleccionar la carpeta llamada “arduino-1.0.2”, en esta carpeta están almacenados los drivers del controlador de la unidad de control



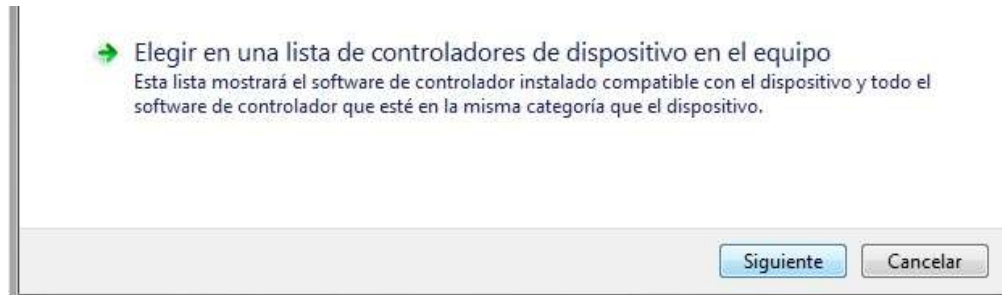
14- Dar clic en aceptar

Luego de dar clic en aceptar se cierra la ventana “Buscar carpeta” y regresa a la pantalla del paso 9 con la diferencia que en el espacio en blanco de “Buscar el software de controlador en esta ubicación” y dar clic en el botón “Examinar...”, aparece la ruta específica de donde se encuentran almacenados los driver del controlador.

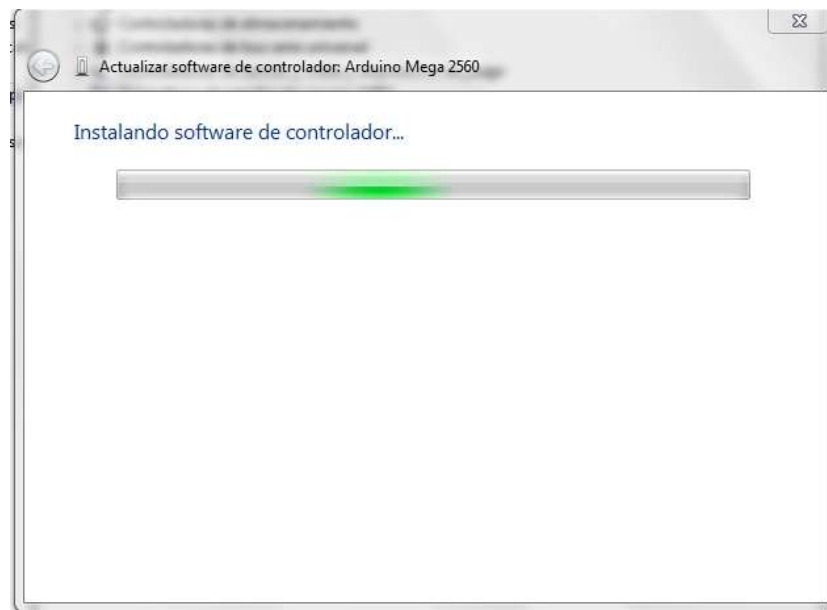
Buscar el software de controlador en esta ubicación:

☒ Incluir subcarpetas

15- Dar clic en siguiente

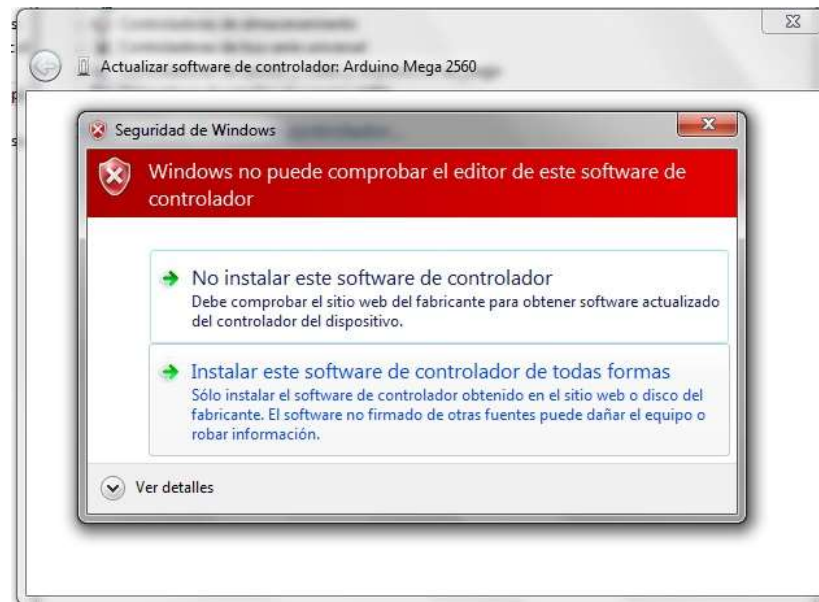


Luego de dar clic en siguiente aparecerá una ventana donde muestra que está cargando el software de control

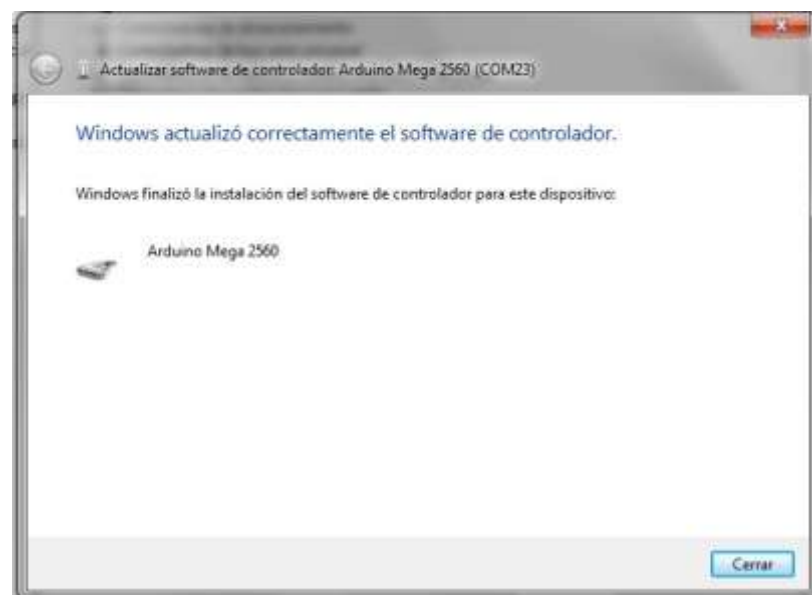


Después de una pequeña espera parece un recuadro con una advertencia debido a que el controlador Arduino es de Open Hardware y Software y no tiene una licencia de fabricante como otros dispositivos.

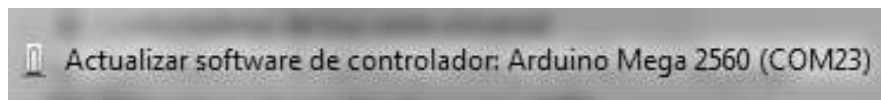
A este se debe hacer caso omiso y dar clic en la opción de “Instalar este software de controlador de todas formas”



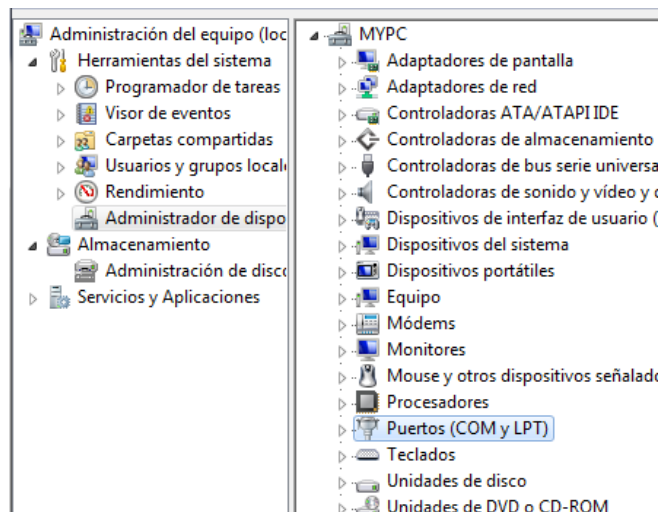
16- Esperar hasta que aparezca la ventana que indica que el controlador ya ha sido instalado.



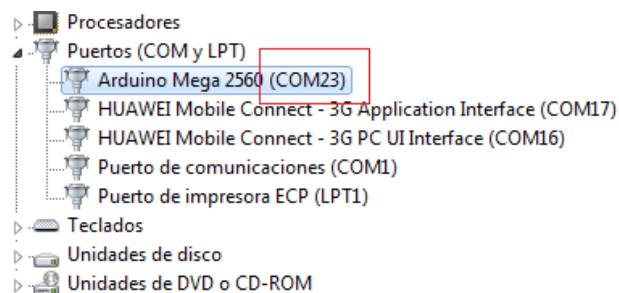
17- Es importante verificar el puerto COM en el que este se instaló pues mediante a ese número se conectaran la unidad de control y la interface de usuario el número de puerto COM se ubica en la parte superior de la venta que avisa que la instalación ha finalizado.



En caso de no haber guardado el número de puerto COM repetir los paso desde 2 hasta el 6 y dar clic en la opción puertos COM y LPT ubicada en el “Administrador de dispositivos”



Y aparecerá una lista de los puertos COM instalados.



NOTA: No siempre será el mismo puerto COM pues este varía dependiendo la computadora donde se instale

- Identificación de ingreso

La aplicación posee una ventana de identificación de usuario como medida de seguridad para que nadie que sea ajeno a la administración manipule dicho sistema automatización.

Capitulo II: Manejo del programa “control de ambientación y timbrado”

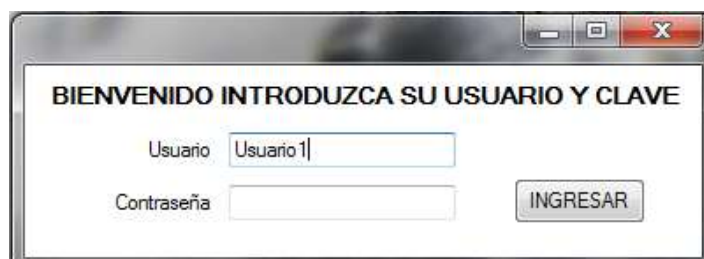
Instrucciones para identificarse

Al abrir la aplicación este abre una venta la cual pide al usuario que se identifique para uso de demostración se utilizara “**Usuario1**” y pide una contraseña la cual es “**utecina2013**”.

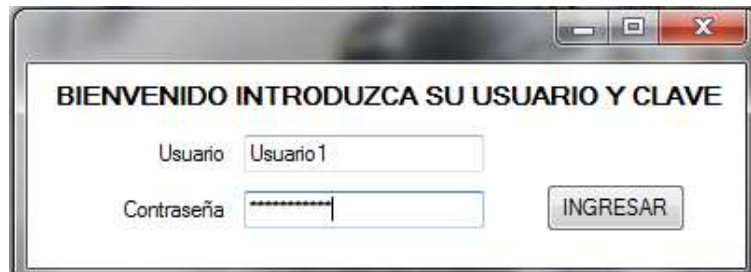
Pantalla principal.

A screenshot of a Windows-style application window. The title bar is at the top with standard minimize, maximize, and close buttons. The main content area has a title "BIENVENIDO INTRODUZCA SU USUARIO Y CLAVE". Below the title, there are two text input fields. The first is labeled "Usuario" and the second is labeled "Contraseña". To the right of the "Contraseña" field is a button with the text "INGRESAR".

1- Introducir usuario en el espacio de “Usuario”

A screenshot of the same application window as before. The "Usuario" input field now contains the text "Usuario1". The "Contraseña" field is still empty, and the "INGRESAR" button remains visible.

2- Introducir contraseña en el espacio de contraseña



3- Clic en el botón “INGRESAR”



Luego de llenar los espacios y presionar el botón ingresar se abrirá la pantalla principal.

NOTA: si el usuario u contraseña son incorrectas aparecerá un mensaje diciendo “contraseña incorrecta” cerrar e intentar nuevamente

- Pantalla Principal

En esta encuentra el nombre de la aplicación, el nombre de la institución y logo de la universidad, como también encontraremos el logo y el nombre de la institución a la que se dona el proyecto, luego a lado derecho, abajo del logo del instituto se encuentran el reloj a tiempo real, el programa está compuesto básicamente por tres botones los cuales poseen dentro de ellos las opciones de control.



Pantalla principal del programa

- 1- El botón para abrir el control de las luces
 - 2- El botón para abrir el control de programación del timbre
 - 3- El botón para abrir el control de los ventiladores
- Conexión de interface de usuario con unidad de control

Esta área se encarga de conectar la interface de usuario con el controlador del Sistema Domótico. Todas las ventanas poseen su propia opción de conexión pues si la institución decide en un futuro utilizar más de un controlador para gestionar las diferentes áreas del instituto pueden sencillamente conectar un controlador diferente desde el mismo programa.

A continuación se muestra la forma de conectar utilizada en las tres ventanas de control de dispositivos, todas estas poseen dentro de ellas los mismos botones para conectar la interface de usuario con el controlador.

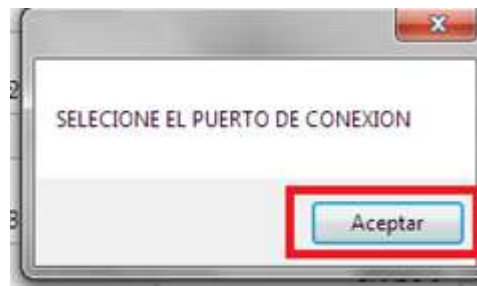
Instrucciones de conexión

1- clic en botón “DETERMINAR CONEXIÓN”.



2- Aparecerá un cuadro de dialogo el cual pide “SELECCIONE EL PUERTO DE CONEXIÓN”.

3- Clic en botón “Aceptar”.

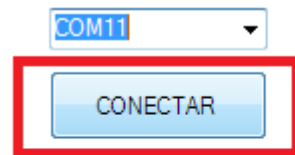


4- Seleccionar el puerto COM con el que se trabajara.



NOTA: Verificar en que puerto COM se instaló el controlador, para mejor referencia revisar capítulo I numeral 17.

5- Clic en botón “CONECTAR”.



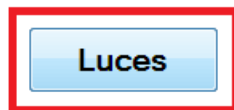
6- Los botones de las demás funciones se activaran seleccionar según su necesidad.

- Manejo de luces de los sectores del Auditorio

Luego de haber seguido los pasos para conectar la interface de usuario con la unidad de control se activaran, de esta manera se puede tener acceso al control de los sectores de las luces los cuales funcionan de la siguiente manera:

Instrucciones para el manejo de las luces del Auditorio

1- Clic en botón luces

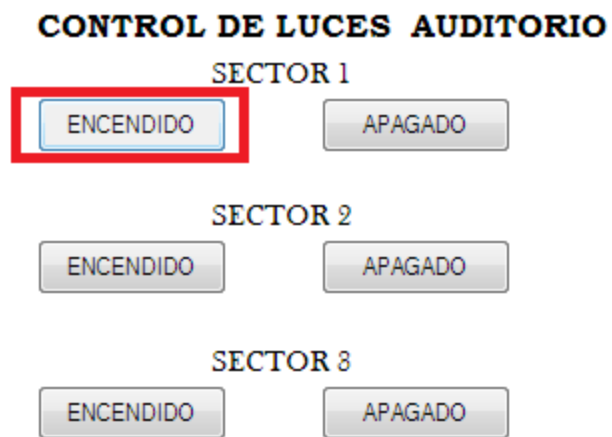


Se abrirá ventana "Control de Luces"



2- Realizar todos los pasos de **instrucciones de conexión**

3- clic en el botón "ENCENDIDO" para activar las luces del sector



4-Clic en botón “APAGADO” para desactivar las luces del sector

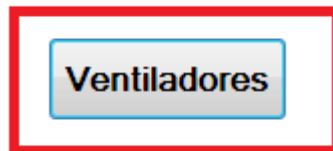


- Manejo de Ventiladores de las Aulas

Luego de conectar todos los dispositivos pueden ser controlados a la misma vez tanto las luces como los ventiladores

Instrucciones para el manejo de los ventiladores de las aulas

1- Clic en botón Ventiladores

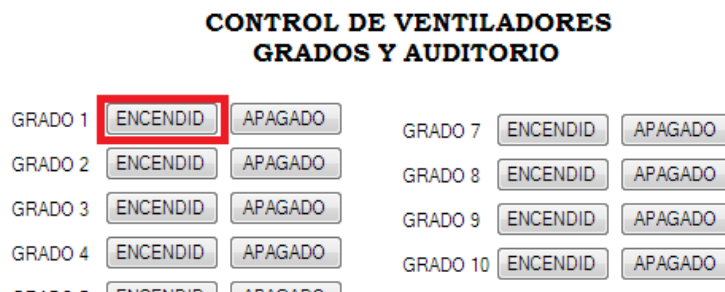


Se abrirá la ventana “Control de ventiladores”



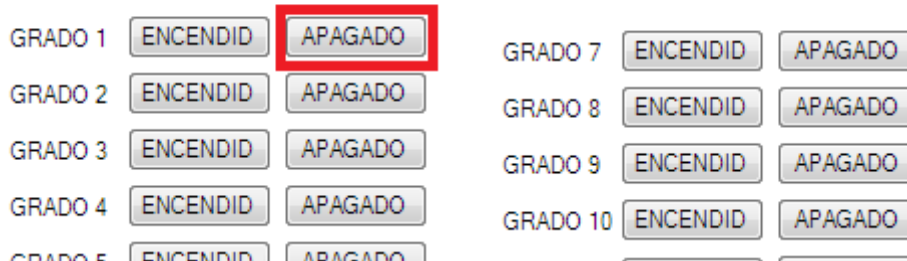
2- Realizar todos los pasos de **instrucciones de conexión**

3- Clic en el botón “ENCENDIDO” para activar el ventilador en la aula de seleccionada.



3- Clic en botón “APAGADO” para desactivar el ventilador en la aula seleccionada.

CONTROL DE VENTILADORES GRADOS Y AUDITORIO



- Manejo de los controles de programación del timbre.

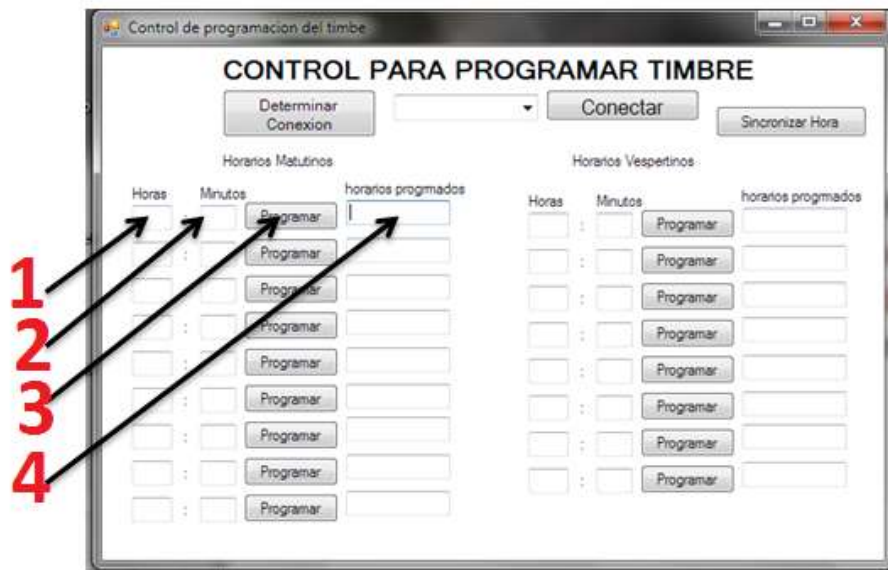
Esta área se encarga de capturar los horarios ingresados en la interface de usuario y enviar estos hacia el controlador principal, quien los guarda en la memoria EPROM para compararlos y tomar las medidas necesarias para activar y desactivar el timbre de la institución, en los horarios pre-establecidos una tan sola vez en la interface del usuario.

Instrucciones para la programación del timbre

1- Clic en botón “Timbre”.



Se abrirá una ventana con los controles de programación del timbre.



Esta ventana consta de 4 áreas (omitiendo el área de conexión al puerto serie) las cuales son:

1. Espacio para ingresar horas.
2. Espacio para ingresar minutos.
3. un botón de programar.
4. Un espacio donde se observa la hora programada.

Para ingresar la hora a programar se deben seguir los siguientes pasos

- 1- Situar el puntero en el espacio 1 e ingresar una hora, para definir la hora en la que se programara el timbre.

Horarios Matutinos

Horas	Minutos	horarios progados
04		
Programar		

2- Situar el puntero en el espacio 2 e ingresar los minutos para definir en qué minuto específico de la hora anteriormente ingresado sonara el timbre.

Horarios Matutinos

Horas	Minutos	horarios progados
04	40	
Programar		

3- clic en el botón “programar”.

Minutos

40	Programar
---------------------------------	--

Al dar clic en dicho botón se encarga de cumplir 2 funciones imprime en el espacio de “horarios programados” los minutos y las horas, y envía por el puerto serie la hora programada, para que el controlador la almacene. Se abrirá una ventana de mensajes que dirá que se programó el horario.

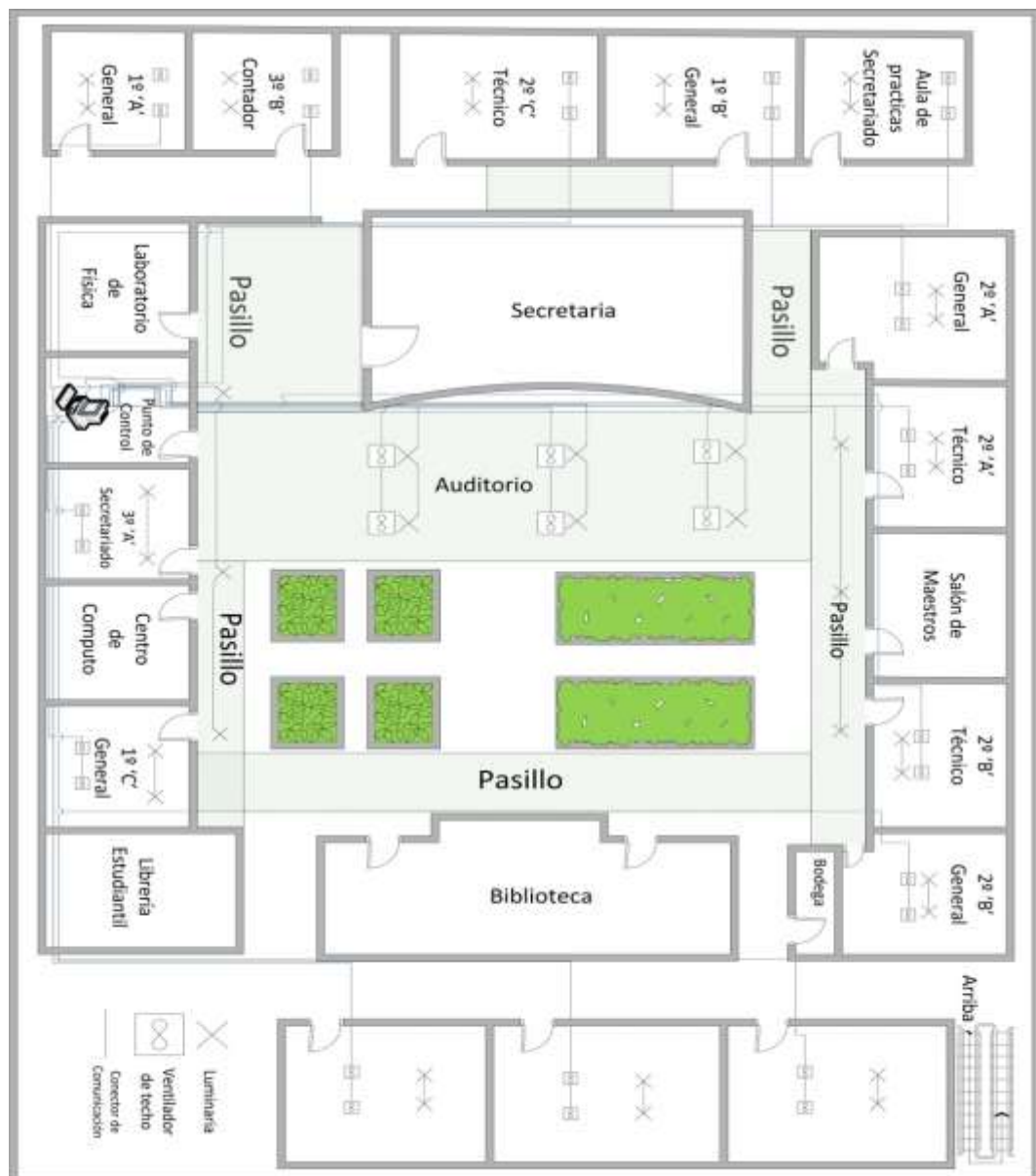


4- Clic en botón aceptar

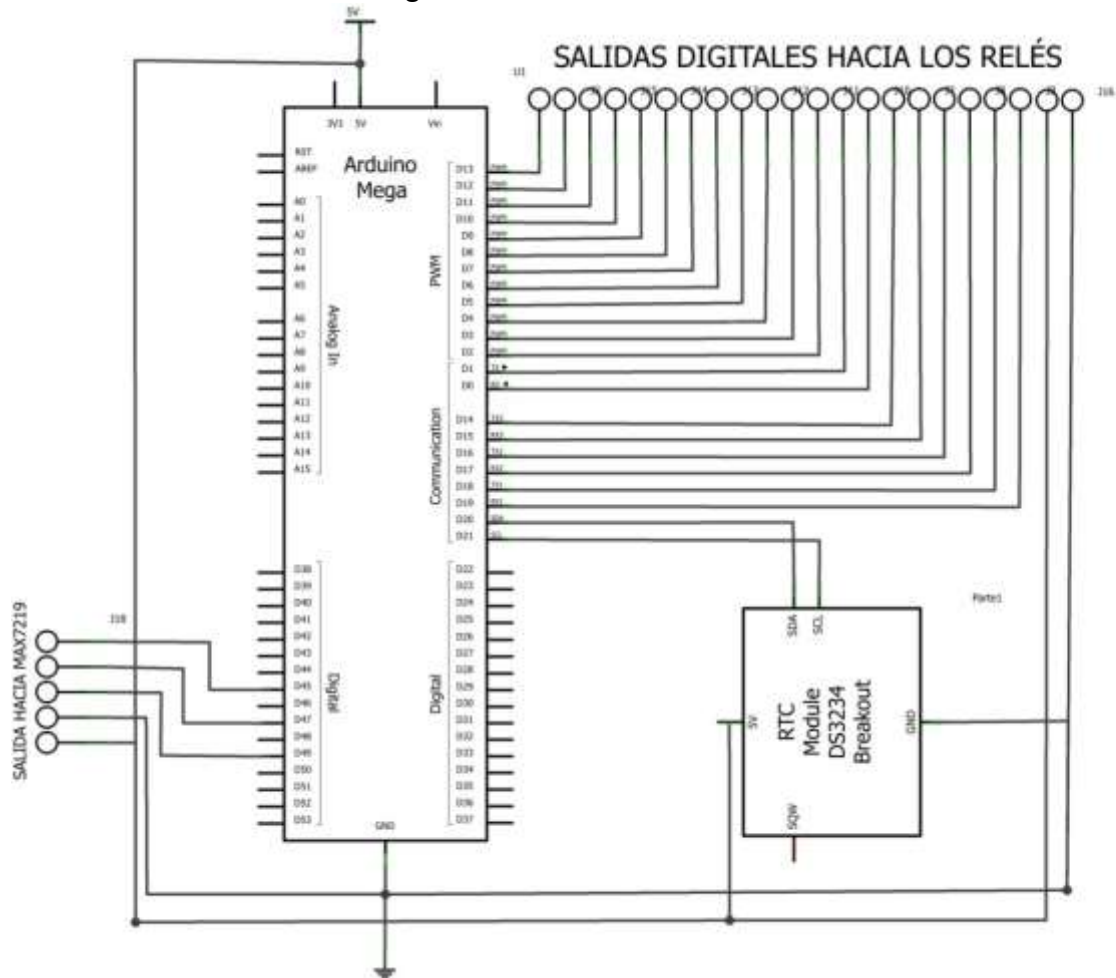
5- Programar las alarmas siguientes.

Capitulo III diagrama de instalzion y diagrama eléctrico

- Plano de la infraestructura del Instituto Nacional de Aguilares, con la implementación de la topología



- Circuito eléctrico general del Sistema Domótico



2.2.5 Código de la aplicación

```
Public Class Form1
```

```

    Private Sub btningre_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btningre.Click
        If TextBox2.Text = "utecina2013" Then
            FormP.Show()
            Me.Visible = False
        Else
            MsgBox("CONTRASEÑA INCORECTA")
        End If
    End Sub

```

```

        End If
    End Sub

End Class

Public Class FormP

    Private Sub btnTimbre_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnTimbre.Click
        Form2.Show()

    End Sub

    Private Sub reloj_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles reloj.Tick
        clk.Text = Now.ToLongTimeString
    End Sub

    Private Sub btnVentiladores_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnVentiladores.Click
        Form4.Show()
    End Sub

    Private Sub btnLuces_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnLuces.Click
        Form5.Show()

    End Sub

    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

    End Sub
End Class

Public Class Form2
    Public Class Form1
        Inherits Form2

    End Class

    Private Sub Form2_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

    End Sub

    Private Sub btnseccion_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnseccion.Click
        spPuertos.Write("1")

```

```

End Sub

Private Sub btnDeterminarconexion_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnDeterminarconexion.Click
    cbpuertos.Items.Clear()
    For Each PuertoDisponibles As String In
My.Computer.Ports.SerialPortNames
        cbpuertos.Items.Add(PuertoDisponibles)
    Next
    If cbpuertos.Items.Count > 0 Then
        cbpuertos.Text = cbpuertos.Items(0)
        MessageBox.Show("SELECCIONE EL PUERTO DE CONEXION")
        btnconectar.Enabled = True

    Else
        MessageBox.Show("NO SE A SELECCIONADO NINGUN PUERTO")
        btnconectar.Enabled = False

    End If
End Sub

```

```

Private Sub btnconectar_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnconectar.Click
    If btnconectar.Text = "Conectar" Then
        spPuertos.PortName = cbpuertos.Text
        btnconectar.Text = "Desconectar"
        btnconectar.Enabled = True
        btnsec1on.Enabled = True
        btnsec1off.Enabled = True
        btnsec2on.Enabled = True
        btnsec2off.Enabled = True
        btnsec3on.Enabled = True
        btnsec3off.Enabled = True
        btnsec4on.Enabled = True
        btnsec4off.Enabled = True
        onall.Enabled = True
        offall.Enabled = True
        spPuertos.Open()
    ElseIf btnconectar.Text = "Desconectar" Then
        btnconectar.Text = "Conectar"
        btnconectar.Enabled = False
        btnsec1on.Enabled = False
        btnsec1off.Enabled = False
        btnsec2on.Enabled = False
        btnsec2off.Enabled = False
        btnsec3on.Enabled = False
        btnsec3off.Enabled = False
        btnsec4on.Enabled = False
        btnsec4off.Enabled = False
        onall.Enabled = False
    End If
End Sub

```

```

        offall.Enabled = False
        spPuertos.Close()
    End If
End Sub

Private Sub btnsec1off_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnsec1off.Click
    spPuertos.WriteLine("2")
End Sub

Private Sub btnsec2on_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnsec2on.Click
    spPuertos.WriteLine("3")
End Sub

Private Sub btnsec2off_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnsec2off.Click
    spPuertos.WriteLine("4")
End Sub

Private Sub btnsec3on_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnsec3on.Click
    spPuertos.WriteLine("5")
End Sub

Private Sub btnsec3off_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnsec3off.Click
    spPuertos.WriteLine("6")
End Sub

Private Sub btnsec4on_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnsec4on.Click
    spPuertos.WriteLine("7")
End Sub

Private Sub btnsec4off_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnsec4off.Click
    spPuertos.WriteLine("8")
End Sub

Private Sub onall_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles onall.Click
    spPuertos.WriteLine("9")
End Sub

Private Sub offall_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles offall.Click
    spPuertos.WriteLine("0")
End Sub
End Class

Public Class Form4

```

```

    Private Sub conventi_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles conventi.Click

        End Sub

    Private Sub Form4_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

        End Sub

    Private Sub btnDeterminarconexion_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles btnDeterminarconexion.Click
        cbpuertos.Items.Clear()
        For Each PuertoDisponibles As String In
My.Computer.Ports.SerialPortNames
            cbpuertos.Items.Add(PuertoDisponibles)
        Next
        If cbpuertos.Items.Count > 0 Then
            cbpuertos.Text = cbpuertos.Items(0)
            MessageBox.Show("SELECCIONE EL PUERTO DE CONEXION")
            btnconectar.Enabled = True

        Else
            MessageBox.Show("NO SE A SELECCIONADO NINGUN PUERTO")
            btnconectar.Enabled = False
        End If
    End Sub

    Private Sub btnconectar_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnconectar.Click
        If btnconectar.Text = "Conectar" Then
            spPuertos.PortName = cbpuertos.Text
            btnconectar.Text = "Desconectar"
            btnconectar.Enabled = True

            spPuertos.Open()
        ElseIf btnconectar.Text = "Desconectar" Then
            btnconectar.Text = "Conectar"
            btnconectar.Enabled = False
            spPuertos.Close()

        End If
    End Sub

    Private Sub btnvenon1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnvenon1.Click
        spPuertos.WriteLine("a")
    End Sub

    Private Sub btnvenoff1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnvenoff1.Click
        spPuertos.WriteLine("b")
    End Sub

```

```

    Private Sub btnvenon2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnvenon2.Click
        spPuertos.WriteLine("c")
    End Sub

    Private Sub btnvenoff2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnvenoff2.Click
        spPuertos.WriteLine("d")
    End Sub

    Private Sub btnvenon3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnvenon3.Click
        spPuertos.WriteLine("e")
    End Sub

    Private Sub btnvenoff3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnvenoff3.Click
        spPuertos.WriteLine("f")
    End Sub

    Private Sub btnvenon4_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnvenon4.Click
        spPuertos.WriteLine("g")
    End Sub

    Private Sub btnvenoff4_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnvenoff4.Click
        spPuertos.WriteLine("h")
    End Sub

    Private Sub btnvenon5_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnvenon5.Click
        spPuertos.WriteLine("i")
    End Sub

    Private Sub btnvenoff5_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnvenoff5.Click
        spPuertos.WriteLine("j")
    End Sub

    Private Sub btnvenon6_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnvenon6.Click
        spPuertos.WriteLine("k")
    End Sub

    Private Sub btnvenoff6_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnvenoff6.Click
        spPuertos.WriteLine("l")
    End Sub

    Private Sub btnvenon7_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnvenon7.Click
        spPuertos.WriteLine("m")
    End Sub

```

```

    Private Sub btnvenoff7_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnvenoff7.Click
        spPuertos.WriteLine("n")
    End Sub

    Private Sub btnvenon8_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnvenon8.Click
        spPuertos.WriteLine("o")
    End Sub

    Private Sub btnvenoff8_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnvenoff8.Click
        spPuertos.WriteLine("p")
    End Sub

    Private Sub btnvenon9_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnvenon9.Click
        spPuertos.WriteLine("q")
    End Sub

    Private Sub btnvenoff9_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnvenoff9.Click
        spPuertos.WriteLine("r")
    End Sub

    Private Sub btnvenon10_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnvenon10.Click
        spPuertos.WriteLine("s")
    End Sub

    Private Sub btnvenoff10_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnvenoff10.Click
        spPuertos.WriteLine("u")
    End Sub

    Private Sub btnvenon11_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnvenon11.Click
        spPuertos.WriteLine("v")
    End Sub

    Private Sub btnvenoff11_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnvenoff11.Click
        spPuertos.WriteLine("w")
    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        spPuertos.WriteLine("x")
    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
        spPuertos.WriteLine("y")
    End Sub

```

```

End Sub

Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
    spPuertos.WriteLine("(")
End Sub

Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button4.Click
    spPuertos.WriteLine(")")
End Sub

Private Sub Button6_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button6.Click
    spPuertos.WriteLine(";")
End Sub

Private Sub Button5_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button5.Click
    spPuertos.WriteLine("?")
End Sub

Private Sub Button7_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button7.Click
    spPuertos.WriteLine("z")
End Sub

Private Sub Button8_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button8.Click
    spPuertos.WriteLine("#")
End Sub

Private Sub Button9_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button9.Click
    spPuertos.WriteLine("$")
End Sub

Private Sub Button10_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button10.Click
    spPuertos.WriteLine("%")
End Sub

Private Sub Button11_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button11.Click
    spPuertos.WriteLine("&")
End Sub

Private Sub Button12_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button12.Click
    spPuertos.WriteLine("=")
End Sub
End Class

Public Class Form5

```

```

Dim h1 As String
Dim m1 As String
Dim h2 As String
Dim m2 As String
Dim h3 As String
Dim m3 As String
Dim h4 As String
Dim m4 As String
Dim h5 As String
Dim m5 As String
Dim h6 As String
Dim m6 As String
Dim h7 As String
Dim m7 As String
Dim h8 As String
Dim m8 As String
Dim h9 As String
Dim m9 As String
Dim h10 As String
Dim m10 As String
Dim h11 As String
Dim m11 As String
Dim h12 As String
Dim m12 As String
Dim h13 As String
Dim m13 As String
Dim h14 As String
Dim m14 As String
Dim h15 As String
Dim m15 As String
Dim h16 As String
Dim m16 As String
Dim h17 As String
Dim m17 As String

Private Sub Form5_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

End Sub

Private Sub btnDeterminarconexion_Click_1(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles btnDeterminarconexion.Click
    cbopuertos.Items.Clear()
    For Each PuertoDisponibles As String In
My.Computer.Ports.SerialPortNames
        cbopuertos.Items.Add(PuertoDisponibles)
    Next
    If cbopuertos.Items.Count > 0 Then
        cbopuertos.Text = cbopuertos.Items(0)
        MessageBox.Show("SELECCIONE EL PUERTO DE CONEXION")
        btnconectar.Enabled = True
    End If
End Sub

```

```

Else
    MessageBox.Show("NO SE A SELECCIONADO NINGUN PUERTO")
    btnconectar.Enabled = False

End If
End Sub

Private Sub btnconectar_Click_1(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnconectar.Click
    If btnconectar.Text = "Conectar" Then
        spPuertos.PortName = cbopuertos.Text
        btnconectar.Text = "Desconectar"
        btnconectar.Enabled = True

        spPuertos.Open()
    ElseIf btnconectar.Text = "Desconectar" Then
        btnconectar.Text = "Conectar"
        btnconectar.Enabled = False

        spPuertos.Close()

    End If
End Sub

Private Sub btnprog1_Click_1(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnprog1.Click
    h1 = hour1.Text
    m1 = minute1.Text
    boxh1.Text = hour1.Text & ":" & minute1.Text()
    spPuertos.WriteLine("A")
    spPuertos.WriteLine(h1)
    spPuertos.WriteLine(m1)
    MsgBox("HORARIO 1 PROGRAMADO")
End Sub

Private Sub Button2_Click_1(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
    h2 = hour2.Text
    m2 = minute2.Text
    boxh2.Text = hour2.Text & ":" & minute2.Text()
    spPuertos.WriteLine("B")
    spPuertos.WriteLine(h2)
    spPuertos.WriteLine(m2)
    MsgBox("HORARIO 2 PROGRAMADO")
End Sub

Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
    h3 = hour3.Text
    m3 = minute3.Text
    boxh3.Text = hour3.Text & ":" & minute3.Text()
    spPuertos.WriteLine("C")
    spPuertos.WriteLine(h3)
    spPuertos.WriteLine(m3)
    MsgBox("HORARIO 3 PROGRAMADO")

```

```

End Sub

Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button4.Click
    h4 = hour4.Text
    m4 = minute4.Text
    boxh4.Text = hour4.Text & ":" & minute4.Text()
    spPuertos.WriteLine("D")
    spPuertos.WriteLine(h4)
    spPuertos.WriteLine(m4)
    MsgBox("HORARIO 4 PROGRAMADO")
End Sub

Private Sub Button5_Click_1(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button5.Click
    h5 = hour5.Text
    m5 = minute5.Text
    boxh5.Text = hour5.Text & ":" & minute5.Text()
    spPuertos.WriteLine("E")
    spPuertos.WriteLine(h5)
    spPuertos.WriteLine(m5)
    MsgBox("HORARIO 5 PROGRAMADO")
End Sub

Private Sub Button6_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button6.Click
    h6 = hour6.Text
    m6 = minute6.Text
    boxh6.Text = hour6.Text & ":" & minute6.Text()
    spPuertos.WriteLine("F")
    spPuertos.WriteLine(h6)
    spPuertos.WriteLine(m6)
    MsgBox("HORARIO 6 PROGRAMADO")
End Sub

Private Sub Button7_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button7.Click
    h7 = hour7.Text
    m7 = minute7.Text
    boxh7.Text = hour7.Text & ":" & minute7.Text()
    spPuertos.WriteLine("G")
    spPuertos.WriteLine(h7)
    spPuertos.WriteLine(m7)
    MsgBox("HORARIO 7 PROGRAMADO")
End Sub

Private Sub Button8_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button8.Click
    h8 = hour8.Text
    m8 = minute8.Text
    boxh8.Text = hour8.Text & ":" & minute8.Text()
    spPuertos.WriteLine("H")
    spPuertos.WriteLine(h8)
    spPuertos.WriteLine(m8)

```

```

        MsgBox("HORARIO 8 PROGRAMADO")
    End Sub

    Private Sub Button9_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button9.Click
        h9 = hour9.Text
        m9 = minute9.Text
        boxh9.Text = hour9.Text & ":" & minute9.Text()
        spPuertos.WriteLine("I")
        spPuertos.WriteLine(h9)
        spPuertos.WriteLine(m9)
        MsgBox("HORARIO 9 PROGRAMADO")
    End Sub

    Private Sub Button17_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button17.Click
        h10 = hour10.Text
        m10 = minute10.Text
        boxh10.Text = hour10.Text & ":" & minute10.Text()
        spPuertos.WriteLine("J")
        spPuertos.WriteLine(h10)
        spPuertos.WriteLine(m10)
        MsgBox("HORARIO 10 PROGRAMADO")
    End Sub

    Private Sub Button16_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button16.Click
        h11 = hour11.Text
        m11 = minute11.Text
        boxh11.Text = hour11.Text & ":" & minute11.Text()
        spPuertos.WriteLine("K")
        spPuertos.WriteLine(h11)
        spPuertos.WriteLine(m11)
        MsgBox("HORARIO 11 PROGRAMADO")
    End Sub

    Private Sub Button15_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button15.Click
        boxh12.Text = hour12.Text & ":" & minute12.Text()
        h12 = hour12.Text
        m12 = minute12.Text
        spPuertos.WriteLine("L")
        spPuertos.WriteLine(h12)
        spPuertos.WriteLine(m12)
        MsgBox("HORARIO 12 PROGRAMADO")
    End Sub

    Private Sub Button14_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button14.Click
        boxh13.Text = hour13.Text & ":" & minute13.Text()
        h12 = hour13.Text
        m12 = minute13.Text
        spPuertos.WriteLine("M")
        spPuertos.WriteLine(h13)

```

```

        spPuertos.WriteLine(m13)
        MsgBox("HORARIO 13 PROGRAMADO")
    End Sub

    Private Sub Button13_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button13.Click
        boxh14.Text = hour14.Text & ":" & minute14.Text()
        h14 = hour14.Text
        m14 = minute14.Text
        spPuertos.WriteLine("N")
        spPuertos.WriteLine(h14)
        spPuertos.WriteLine(m14)
        MsgBox("HORARIO 14 PROGRAMADO")
    End Sub

    Private Sub Button12_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button12.Click
        boxh15.Text = hour15.Text & ":" & minute15.Text()
        h15 = hour15.Text
        m15 = minute15.Text
        spPuertos.WriteLine("O")
        spPuertos.WriteLine(h15)
        spPuertos.WriteLine(m15)
        MsgBox("HORARIO 15 PROGRAMADO")
    End Sub

    Private Sub Button11_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button11.Click
        boxh16.Text = hour16.Text & ":" & minute16.Text()
        h16 = hour16.Text
        m16 = minute16.Text
        spPuertos.WriteLine("P")
        spPuertos.WriteLine(h16)
        spPuertos.WriteLine(m16)
        MsgBox("HORARIO 16 PROGRAMADO")
    End Sub

    Private Sub Button10_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button10.Click
        boxh17.Text = hour17.Text & ":" & minute17.Text()
        h17 = hour17.Text
        m17 = minute17.Text
        spPuertos.WriteLine("Q")
        spPuertos.WriteLine(h17)
        spPuertos.WriteLine(m17)
        MsgBox("HORARIO 17 PROGRAMADO")
    End Sub

    Private Sub Button18_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button18.Click
        spPuertos.WriteLine("@")
    End Sub
End Class

```

2.3 Marco Teórico Conceptual

- Domótica:

Se entiende como el conjunto de técnicas utilizadas para la automatización de las gestiones en una vivienda o recinto. Con la finalidad de hacer una vida más sencilla lejos de las acciones repetitivas al usuario.

- Inmótica:

Es la incorporación de automatizaciones en ambientes industriales o de oficina con la automatización integral de inmuebles con alta tecnología. La centralización de los datos del edificio, posibilita supervisar y controlar confortablemente desde una PC, los estados de funcionamiento o alarmas de los sistemas que componen la instalación.

- Sistema:

Es un conjunto de procesos o elementos entrelazados, con un medio para formar una totalidad dirigida a la realización de un objetivo común, este puede llegar a ser físico o abstracto.

- Ambientación:

Disposición adecuada de un lugar con la una finalidad en común, muchas veces conocido como el entorno de un determinado espacio.

- Visual Basic:

Es un lenguaje de programación y una aplicación para el desarrollo de programas ejecutables basado en BASIC, desarrollado por Microsoft, especial para la programación orientada a objetos y orientado a eventos

- Conmutador:

Es un dispositivo eléctrico o electrónico que se encarga de dar paso o de cortar el paso a los electrones, generalmente son de uso manual y son utilizados en las instalaciones eléctricas convencionales en las casas y en los edificios, etc.

- Topología Bus:

Los elementos (Romero, Vázquez, & Castro, 2011) comparten la misma línea o bus de comunicación. Cada elemento suele estar identificado por una dirección única y se pueden comunicar dos elementos de forma simultánea.

- Topología Anillo:

Los elementos (Romero, Vázquez, & Castro, 2011) se encuentran interconectados formando un anillo cerrado, por lo que la información pasa por todos los elementos. Sus principales ventajas son: control sencillo y mínimo cableado.

- Topología Árbol:

Es una (Romero, Vázquez, & Castro, 2011) topología que mezcla parte de las anteriores, en particular de la estrella y del bus, y permite además el establecimiento de una jerarquía entre los elementos de la red.

- Controlador Central:

El controlador central en Domótica es el centro del sistema en la topología tipo estrella, es el encargado de procesar la información de entrada como la información de salida, y es regido por un software de gobierno o firmware, el cual tiene las indicaciones específicas a seguir.

- Centralizado:

El término centralizado es utilizado para dar referencia a que todo está entrelazado y tiene un fin común y todo está regido desde un punto en específico.

- Interface de Usuario:

Es la vía en la que se puede comunicar el usuario ya sea con una máquina o con un sistema, este integra todos los puntos de contacto entre usuario y el equipo generalmente no contiene complejidad alguna pues suelen ser fáciles de entender y accionar.

- Arquitectura Descentralizada:

Como el (Romero, Vázquez, & Castro, 2011) nombre lo indica es justamente la arquitectura opuesta a la centralizada. En la arquitectura descentralizada todos los elementos del sistema disponen de inteligencia, en el sentido de que son totalmente independientes. El sistema debe disponer de un bus compartido que permita la comunicación de los elementos.

- Arquitectura Distribuida:

La (Romero, Vázquez, & Castro, 2011) idea de la arquitectura distribuida es mejorar las dos arquitecturas anteriores, para ello el elemento de control se sitúa próximo al elemento a controlar. Ahora no existe un único elemento de control que gobierna todo el sistema, sino que existen varios elementos entre los que se reparte la tarea de control. Estos nuevos elementos de control se denominan nodos, y a ellos se conectan los elementos básicos

- Señales Eléctricas:

Es un tipo de señal generada por fenómenos electromagnéticos, esta puede ser de tipo Análoga y tomar un valor variante de forma continua en el tiempo, o de tipo Digital tomar solo dos valores discretos.

- Relé:

Es un dispositivo electromecánico. Que cumple la función de un conmutador y funciona controlado por un circuito eléctrico en el que, por medio de una bobina y un electroimán, se acciona un juego de uno o varios contactos que permiten abrir o cerrar otros circuitos eléctricos independientes, Todo mediante un pulso eléctrico enviado a la bobina.

- Arduino:

Es una plataforma abierta ideal para realización de prototipos con finalidades educativas, profesionales, Artísticas y de ocio, está basada en software y hardware

de tipo flexible, posee en su circuitería un microcontrolador y una interface de programación.

- **Conexión Serie:**

Consiste en el envío de un solo bit de información a la vez de manera secuencial, o sea bit a bit, a un ritmo acordado entre el emisor y el receptor, utilizado generalmente para la comunicación entre periféricos.

2.4 Documentación Técnica

Arduino Mega R3

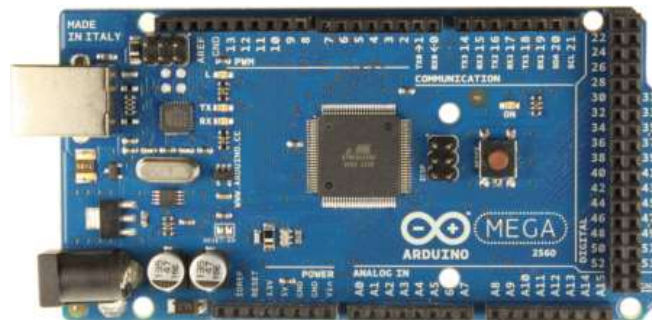


Imagen 2.4.1 imagen ARDUINO MEGA R3

Generalidades

El Arduino Mega 2560 es una placa electrónica basada en el microprocesador Atmega2560. Cuenta con 54 entradas / salidas digitales pines (de los cuales 14 se pueden utilizar como salidas PWM), 16 entradas analógicas, 4 UARTs (puertos serie de hardware), un oscilador de cristal de 16 MHz, una conexión USB, un conector de alimentación, una cabecera ICSP, y un botón de reinicio. Contiene todo lo necesario

para apoyar el microcontrolador, basta con conectarlo a un ordenador con un cable USB o el poder con un adaptador de CA a CC o batería para empezar

Alimentación

Los Arduino Mega puede ser alimentado a través de la conexión USB o con una fuente de alimentación externa. La fuente de alimentación se selecciona automáticamente.

Externa (no USB) de potencia puede venir con un adaptador de corriente alterna a corriente continua (pared-verruga) o la batería. El adaptador se puede conectar al conectar un centro de 2.1mm-positivo clavija en jack de alimentación de la placa. Conduce de una batería se pueden insertar en los encabezados de pines Gnd y Vin del conector de alimentación.

La junta puede operar en un suministro externo de 6 a 20 voltios. Si se proporcionan menos de 7V, sin embargo, el pin de 5V puede proporcionar menos de cinco voltios y el tablero puede ser inestable. Si se utiliza más de 12V, el regulador de voltaje se puede sobrecalentar y dañar la placa. El rango recomendado es 7 a 12 voltios.

Los pines de alimentación son los siguientes:

- **VIN.** La tensión de entrada a la placa Arduino cuando se utiliza una fuente de alimentación externa (en lugar de 5 voltios de la conexión USB o de otra fuente de alimentación regulada). Se puede suministrar tensión a través de esta clavija, o, si el suministro de tensión a través de la toma de poder, acceder a él a través de esta clavija.
- **5V.** Este pin como salida una 5V regulada por el regulador en el tablero. La junta se puede suministrar corriente, ya sea a partir de la entrada de

alimentación (7 - 12 V), el conector USB (5V), o el pasador de VIN de la junta (7-12V). El suministro de tensión a través de los pines de 5V o 3.3V no pasa por el regulador, y puede dañar la placa.

- **3.3 V.** Una tensión de alimentación 3,3 generado por el regulador incluido. Consumo de corriente máxima es de 50 mA.
- **GND.** pins de tierra.
- **IOREF.** Este pin en la placa Arduino proporciona la tensión de referencia con el que el microcontrolador opera. Un escudo correctamente configurado puede leer el voltaje pin IOREF y seleccione la fuente de alimentación adecuada o habilitar traductores de tensión en las salidas para trabajar con el 5V o 3.3V.

Memoria

El Atmega2560 tiene 256 KB de memoria flash para almacenar el código (de 8 KB que se utiliza para el cargador de arranque), 8 KB de SRAM y 4 KB de memoria EEPROM

Entradas y salidas

Cada uno de los 54 pines digitales en los Mega se puede utilizar como una entrada o salida, utilizando pinMode (), digitalWrite () , y digitalRead () funciones. Funcionan a 5 voltios. Cada pin puede proporcionar o recibir un máximo de 40 mA y tiene una pull-up interna resistor (desconectado de manera predeterminada) de 20-50 kOhms. Además, algunos pines tienen funciones especializadas:

- **Serie: 0 (RX) y 1 (TX); serie 1: 19 (RX) y 18 (TX); serie 2: 17 (RX) y 16 (TX); serie 3: 15 (RX) y 14 (TX).** Se utiliza para recibir (RX) y transmitir

(TX) datos serie TTL. Clavijas 0 y 1 están también conectados a los correspondientes pasadores de la ATmega16U2 de USB a TTL chip de serie.

- **Las interrupciones externas: 2 (interrupción. 0), 3 (alarma 1), 18 (interrupción 5), 19 (interrupción 4), 20 (interrupción 3), y 21 (interrupción 2)** Estos pines pueden ser configurados para activar una interrupción en un valor bajo, un flanco ascendente o descendente, o un cambio en el valor.
- **PWM: 2 a 13 y 44 a 46** Proporcionar 8-bit de salida PWM con la analogWrite () función.
- **SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS)** Estos pines compatible con la comunicación SPI utilizando la biblioteca de SPI . Los pines SPI también desatado en la cabecera ICSP, que es físicamente compatible con el Uno, Duemilanove y Diecimila.
- **LED: 13.** Hay un built-in LED conectado al pin digital 13. Cuando el pin es valor HIGH, el LED está en encendido, cuando la pin es LOW, es apagado.
- **TWI: 20 (SDA) y 21 (SCL)** Apoyo TWI comunicación a través de la biblioteca wire. Tenga en cuenta que estos pines no están en la misma ubicación que los pines de inmersión recíproca en el Duemilanove o Diecimila.

El Mega2560 tiene 16 entradas analógicas, cada una de las cuales proporcionan 10 bits de resolución (es decir, 1024 valores diferentes). Por defecto se mide desde el

suelo a 5 voltios, aunque es posible cambiar el extremo superior de su rango usando el pin AREF y `analogReference ()` la función.

Hay un par de los otros pines en el circuito:

- **AREF.** Voltaje de referencia para las entradas analógicas. Se utiliza con `analogReference ()`.
- **Restablecer.** Trae esta línea LOW para reiniciar el microcontrolador. Normalmente se utiliza para agregar un botón de reinicio a los escudos que bloquean el uno en el tablero.

Comunicación

El Arduino Mega2560 tiene un número de instalaciones para la comunicación con un ordenador, Arduino otro, u otros microcontroladores.

El ATmega2560 proporciona cuatro UART para TTL (5V) de comunicación serie. Un ATmega16U2 en los canales bordo de uno de estos a través de USB y proporciona un puerto COM virtual con el software en el equipo (máquinas con Windows se necesita un archivo. `inf`, pero lo hará máquinas OSX y Linux reconocer la tarjeta como un puerto COM automáticamente. El software de Arduino incluye un monitor de serie que permite simples datos de texto que se envían desde y hacia el tablero. El RX y TX LED en el tablero parpadea cuando los datos se transmiten a través de la ATmega8U2/ATmega16U2 conexión chip y USB al ordenador (pero no para la comunicación en serie en los pines 0 y 1).

Características

Microcontroladores	Atmega2560
Tensión de funcionamiento	5V
Voltaje de entrada (recomendado)	7-12V
Voltaje de entrada (limites)	6-20V
Digital I / O Pins	54 (de los cuales 15 proporcionan salida PWM)
Pines de entrada analógica	16
Corriente por I DC/ O pin	40 mA
Corriente DC por Pin 3.3V	50 mA
Memoria Flash	256 KB de los cuales 8 KB utilizadas por bootloader
SRAM	8 KB
EEPROM	4 KB
Velocidad del reloj	16 MHz

Módulo de Relés

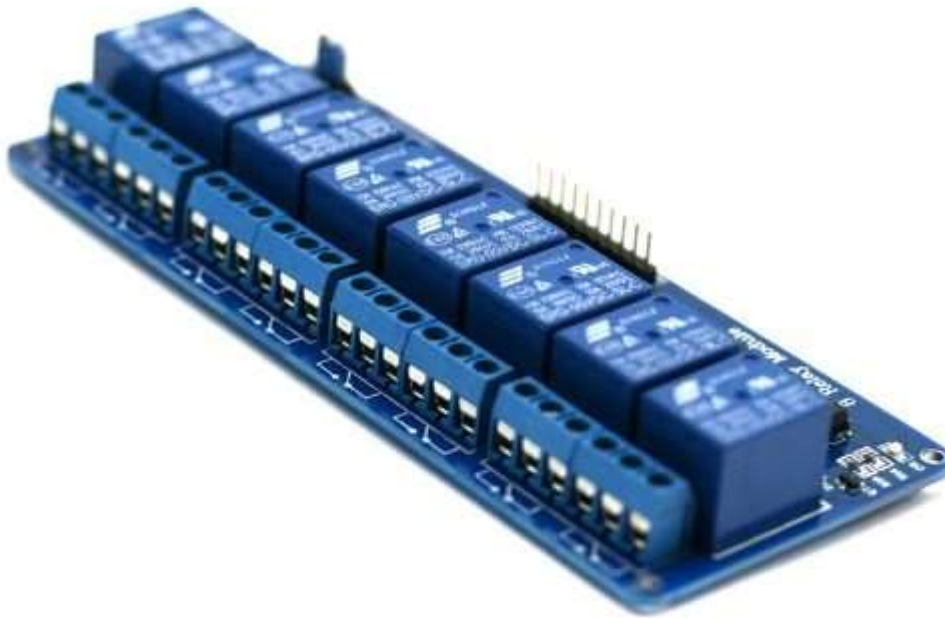


Imagen 2.4.2 Modulo de 8 Relés para conmutación

El relé es un sencillo interruptor mecánico on/off. Se activa cuando la entrada llega a 5V y se desactiva cuando la entrada es 0V. Puede ser controlado a través de la función `digitalWrite()` de Arduino.

El módulo dispone de tres conexiones, llamadas COM, NC e NO. NC significa “NORMALMENTE CERRADO”. Esto significa que cuando el relé no recibe la señal (LOW o 0V de Arduino), el circuito conectado está abierto, pero si se aplican 5V (por definición el pin está en HIGH), el circuito conectado está cerrado. NO significa “NORMALMENTE ABIERTO”, y funciona al revés: si se aplican los 5V el circuito se abre, mientras que se cierra con 0V. Los relés pueden sustituir los interruptores manuales. Desconectado los cables del interruptor y conectándolos a los terminales

COM y NO. Cuando se activa el relé el circuito se cierra y la corriente puede llegar al aparato que se está controlando.

TIPO	CARACTERISTICAS
Resistencia de contacto	100 mW max.
Tiempo de funcionamiento	10 ms máx.
Tiempo de liberación	5 mseg máx.
Rigidez dieléctrica	1500VAC 50/60HZ (1 minuto)
Entre bobina y contacto	1000VAC 50/60HZ (1 minuto)
Resistencia de aislamiento	100 MW mín. (500 VCC)
Max. Encendido / apagado	
mecánicamente	300 operaciones / min
eléctricamente	30 operaciones / min
Temperatura ambiente	-40 ° C a +85 ° C
Humedad de funcionamiento	1. 45 a 85% RH
vibración	
resistencia	10 a 55 Hz Amplitud 1,5 mm doble
Error de operación	10 a 55 Hz Amplitud 1,5 mm doble
choque	
resistencia	100G Min.
Error de operación	10G Min.
esperanza de vida	
mecánicamente	107 operaciones. Min. (Sin carga)
eléctricamente	105 operaciones. Min. (a tensión nominal de bobina)
Peso	aprox. 10grs.

CAPÍTULO III: DESARROLLO DE LA SOLUCIÓN

3.1 Propuesta de Solución

En base a las problemáticas anteriormente mencionadas en el capítulo uno, surgen propuestas técnicas y teóricas orientadas a la solución de las mismas, con énfasis a las áreas de gestión de un Sistema Domótico tales como el confort y ahorro energético.

Siempre y cuando estas sean soluciones viables para el desarrollo e implementación del proyecto, siendo así se delimitaron soluciones no factibles en aspectos técnicos para el manejo del Sistema Domótico.

Utilizando únicamente aquellas que ayudarán a la institución al no despilfarro de los recursos, contribuyendo así en el ahorro de energía eléctrica de la institución, ya que por medio de la implementación del Sistema Domótico se podrá sectorizar y controlar que áreas del instituto tendrán activos su sistema de ventilación, evitando que los alumnos los dejen encendidos malgastando a los recursos eléctricos con los que cuenta la institución, al igual que dinamizar el control de timbre y las luces del auditorio, gestionando de forma directa el confort del personal de mantenimiento ya que este no tendrá que supervisar de grado en grado si los ventiladores están encendidos o apagados, tampoco tendrá que preocuparse por las demoras con el timbre pues mediante la interface de usuario esta será programada para que suene a los horarios indicados por la institución evitando la pérdida de horas clases.

La concepción de dicho proyecto surge de observar las necesidades de la institución ya que este no posee la cantidad necesaria de personal de mantenimiento para coordinar algunas acciones como revisar de grado en grado si han apagado los conmutadores manuales o tocar el timbre en los horarios estipulados entre otras necesidades.

Como todo Sistema Domótico necesita de ciertos componentes que lo integren, para su funcionamiento óptimo, y que cumplan con las necesidades del usuario, es por ello que se seleccionaron componentes tales como: Tarjeta de desarrollo para aplicaciones con microcontroladores Atmega Arduino MEGA 2560, módulo integrado de relés de 8 canales, modulo integrado de relé de un canal, modulo integrado para circuito integrado DS1307 RTC, un circuito integrado driver para display MAX7219 y un Display de cuatro dígitos y cuatro conectores hembra RJ45.

Todos los elementos antes mencionados conforman la unidad de control, dicha unidad estará empotrada en una caja rectangular que contendrá al frente un reloj a tiempo real sincronizado con el reloj de la interface de usuario y dentro de esta caja se encontrará el controlador, el sistema de acondicionamiento de alimentación del circuito electrónico y los conectores RJ45 de tipo hembra, estos se encargarán de llevar la señal de activación y desactivación a los relés mediante cable UTP categoría 5e.

Dicha caja se encuentra instalada en un cuarto en el costado izquierdo del auditorio del Instituto nacional de Aguilares, el cual es habilitado para tener como centro de mando, en él se encuentra una computadora con sistema operativo Windows.

Además esta computadora cuenta con una interface de usuario la cual se conecta con el controlador para administrar desde un solo punto todos los dispositivos conectados a él, dicha interface de usuario es una aplicación ejecutable en sistemas operativos Windows, desarrollada en el lenguaje de programación BASIC diseñada en el desarrollador Visual Basic.NET, cuenta con una interface gráfica sencilla fácil de manejar y bien estructurada para dinamizar en gran manera el control sectorizado de la ambientación y el timbrado de la institución, la aplicación cuenta en su plano principal con tres botones, cada uno dirigido a una de las áreas a controlar, al presionar cualquiera de estos botones se abren las diferentes opciones para controlar la ventilación, las luminarias o el timbre.

Para la programación del timbre y la funcionabilidad del mismo este sistema Domótico cuenta con la característica de guardar en la memoria EPROM del controlador principal la hora ingresada desde la interface de usuario, luego de ser almacenada esta puede ser regrabada al gusto del usuario.

Dicha hora ingresada por el usuario será comparada con la hora del RTC y en base a esta comparación, sonara el timbre que avisa a los alumnos de la institución la hora de receso o de ingreso

Para sincronizar el reloj del sistema Domótico con el reloj de la interface de usuario basta con ingresar a la sección de “control de programación de timbre” el cual tiene la opción un botón con el nombre “sincronizar” y presionarlo; esto para llevar de forma sincronizada tanto la hora de la interface de usuario y la hora que se muestra en el controlador principal.

La creación de una unidad de control que se encargará de manejar desde un solo punto de forma centralizada mediante la utilización de una computadora, es una ruta viable para la solución de las necesidades antes mencionadas.

Para el desarrollo de dicha unidad de control, es necesario inicializar con el diseño del circuito impreso pues en este irán empotrados los diferentes componentes electrónicos que integran el Sistema Domótico, esta unidad de control cuenta con una conexión serie vía USB, dicha conexión se encarga de la comunicación entre la interface de usuario ubicada en la computadora y el controlador principal, y también cuenta con acondicionamiento de salida hacia los actuadores (relés).

Para el desarrollo de firmware o software de control se utilizó el software de desarrollo Arduino ID, el cual utiliza un lenguaje de programación llamado C like el cual es derivado del lenguaje C y el lenguaje C++, el software de control incluye algunas librerías utilizadas para el control del RTC, el reloj físico de la caja donde irá empotrado el Sistema Domótico y para almacenar los horarios en que el timbre debe

de activarse, contiene variables unas para almacenar los datos enviados para la programación del timbre y otras que guardan el valor recibido para la activación de los actuadores, determina la conexión serie entre la computadora y el controlador principal y luego posee comparaciones entre los datos de entrada y las acciones a tomar.

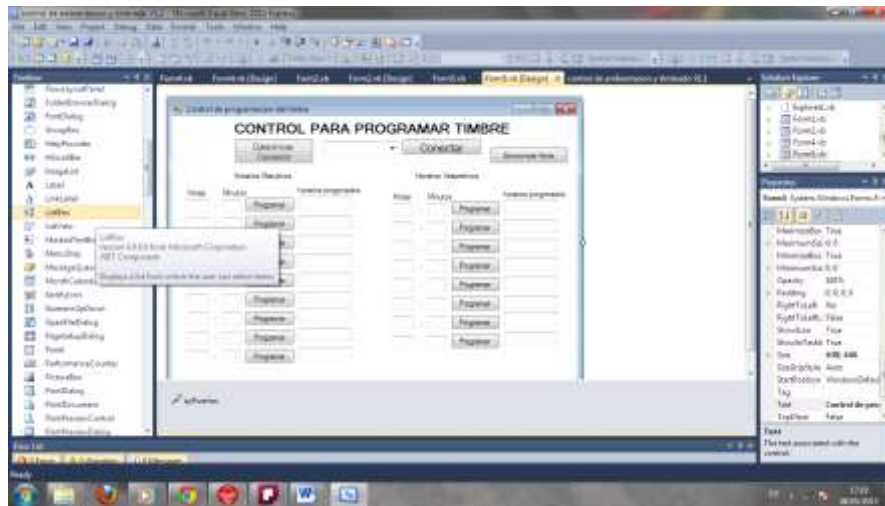
En el diseño de esquemas eléctricos y circuitos impresos se utilizaron dos programas enfocados al desarrollo de los mismos el CadSoft EAGLE PCB Design y el entorno de desarrollo Fritzing.

Para el uso óptimo del Sistema Domótico se desarrolló un manual de usuario, con la finalidad de capacitar a quien utilizara al final el sistema instalado en la institución, este describe paso a paso las instrucciones a seguir para conectar la interface de usuario con el controlador principal, la función de cada botón en la interface de usuario, la forma en la que se maneja cada dispositivo tanto para el control de la ventilación como para el control de las luminarias, y también incluye las instrucciones a seguir para programar de forma adecuada el timbre.

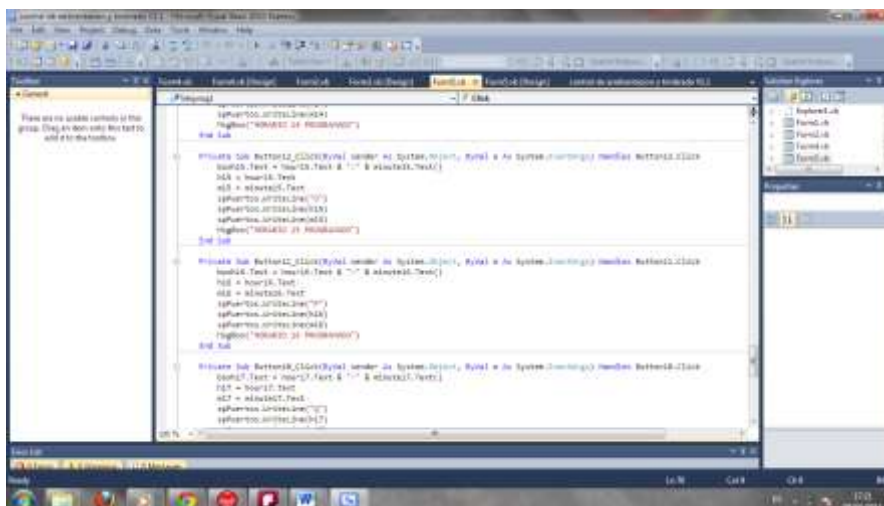
Para la creación del manual de usuario se utilizaron herramientas de diseño gráfico tales como el adobe photoshop el cual permite adecuar de forma precisa la información plasmada en el manual de una forma gráfica amigable a la hora de leerlo.

3.1.2 Desarrollo de la solución.

Como se mencionó anteriormente se utilizó el programa Visual Basic 2010



El cual permite trabajar mediante formularios gráficos para que luego estos sean desarrollados individualmente a la hora de programar

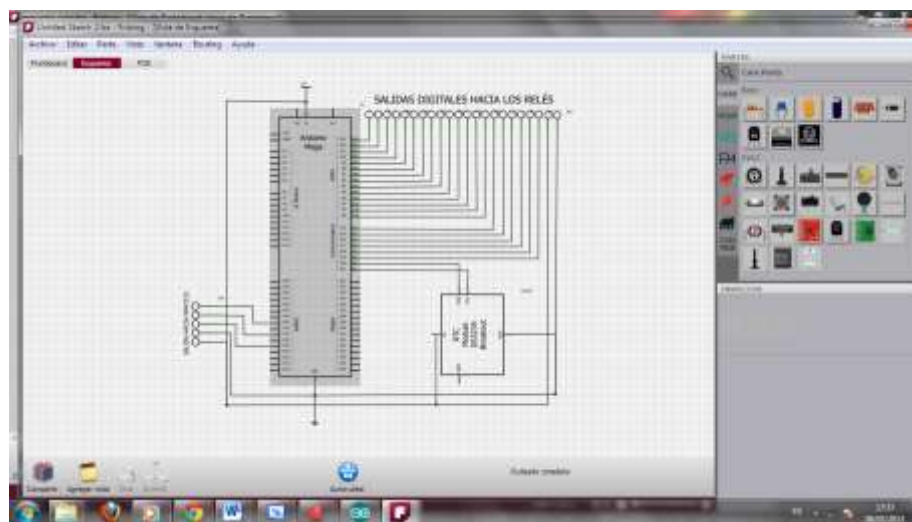


[illegible]

Arduino IDE es un software de distribución libre, que compila el sketch para luego ser grabados en la memoria volátil del Arduino.



Para el desarrollo de los diagramas se utilizó Fritzing, ya que este contiene librerías con el esquema de Arduino y permite trabajar de forma adecuada y ordenada para planificar el diagrama de conexión del circuito del sistema Domótico

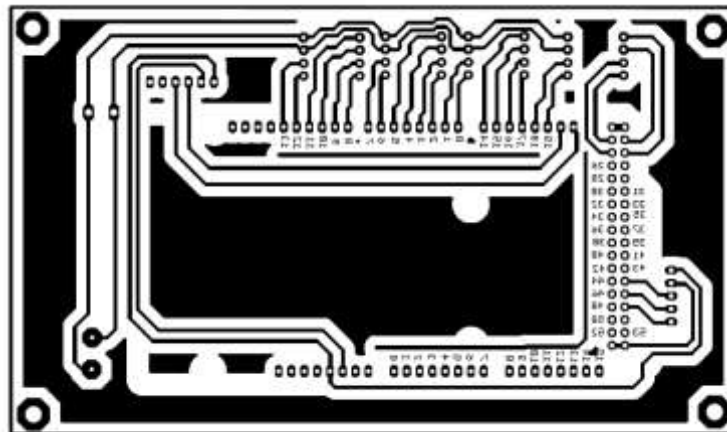


Para la creación del manual se utilizó el photoshop.

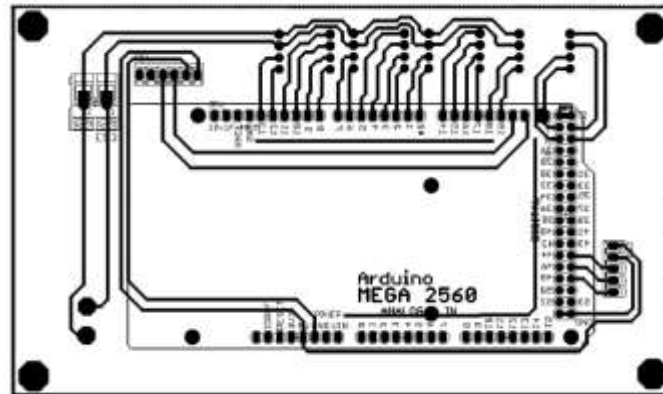


3.1.3 Construcción de la solución

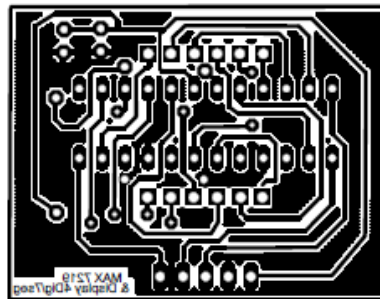
Diseño del PCB principal de montaje del Arduino



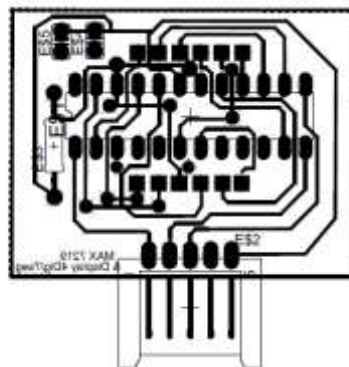
Máscara de componentes



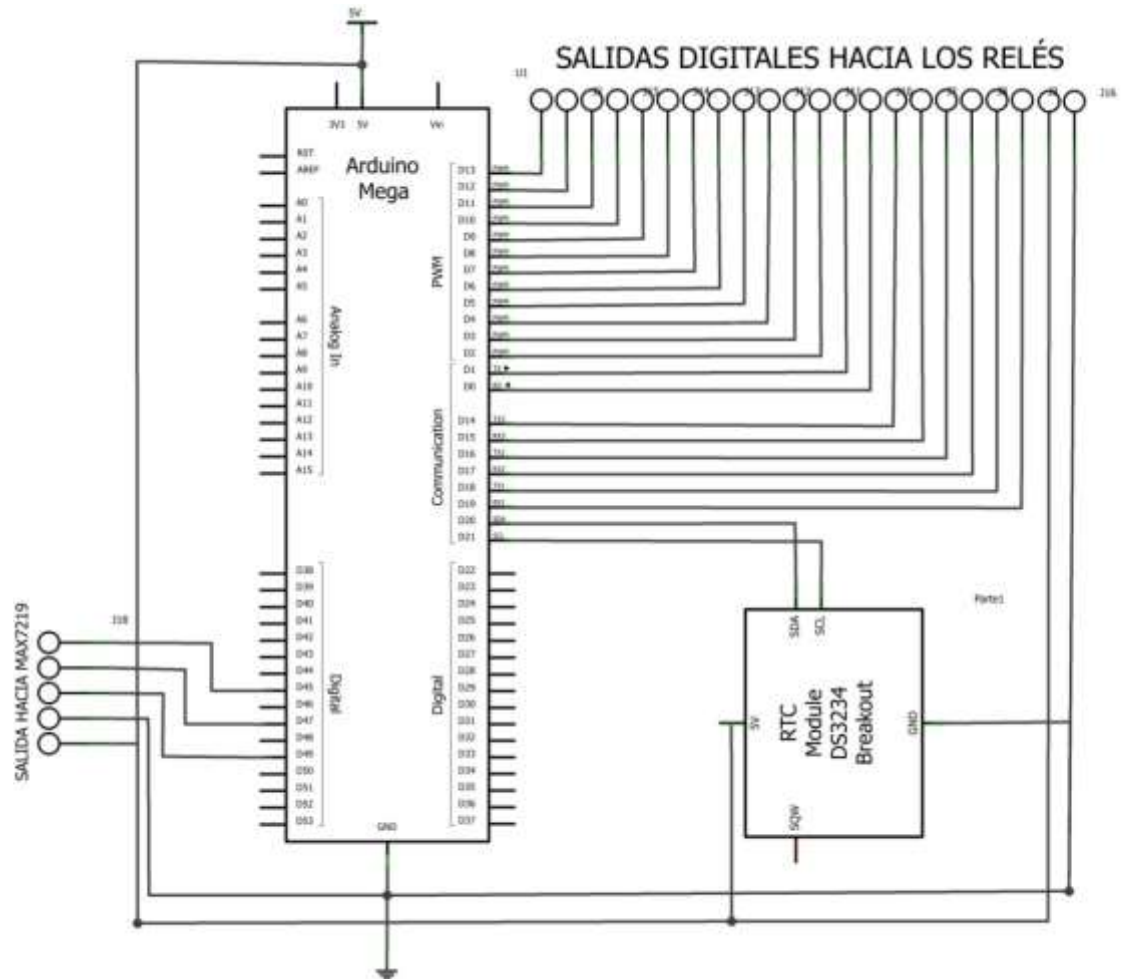
Diseño de PCB para el MAX7219 y el Display de cuatro dígitos



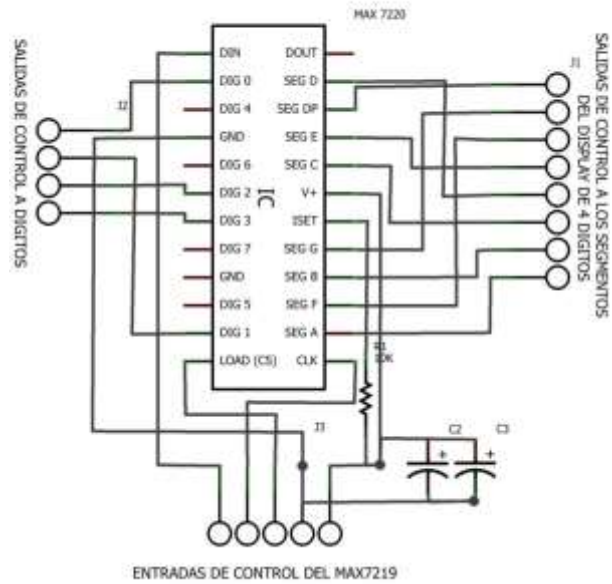
Máscara de componentes del MAX7219 y el Display de cuatro dígitos



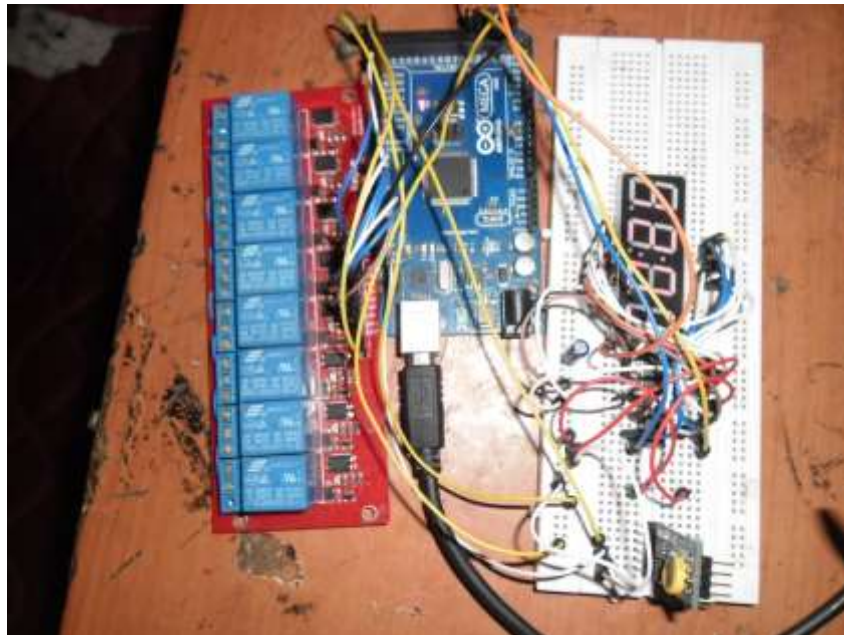
Circuito general del Sistema Domótico



Circuito del reloj externo con el MAX7219 y Display de cuatro dígitos



Montaje en breadboard del Sistema Domótico



Circuito principal del Sistema Domótico empotrada en la caja principal



3.2 Conclusiones

Como resultado del proyecto antes presentado, es posible concluir que existe un beneficio sorprendente, amplio y amigable al consumo eléctrico en la implementación de sistemas de domótica como beneficio a labores de tipo repetitivas, es notable destacar que diferentes ramas de la informática como la programación, topologías de

red ,conocimiento de hardware y sistemas operativos se hacen necesarios, así también por el lado de la electrónica se hace necesario el conocimiento de componentes eléctricos, medidas eléctricas y circuitos electrónicos. De esta manera se tiene como resultado sistemas que cumplan funciones que benefician operaciones repetitivas.

Cabe mencionar que el proyecto presentado con el propósito de controlar dispositivos eléctricos, gestionando áreas de confort y ahorro energético que es administrado de forma centralizada y con la tarea que solventa de manera sencilla las tareas repetitivas del Instituto Nacional de Aguilares, ha sido cumplido satisfactoriamente, dando una solución óptima programable y automatizable al problema planteado y logrando cumplir cada uno de los objetivos propuestos así como ciertos parámetros para la realización del proyecto como producto final. Así el problema que en un inicio se presentó y que se centró en dar la solución a un tipo de problema que incomodaba tanto al personal administrativo como al alumnado y principalmente en un enfoque de automatización de componentes eléctricos.

Ahora da innovación y forma parte de la infraestructura del Instituto Nacional de Aguilares como parte de las muchas posibles soluciones competentes que se le pueden dar inconvenientes o circunstancias adversas que de igual forma innovan y muestran el potencial beneficioso que se tiene el área de la informática y la electrónica.

3.3 Recomendaciones

Recomendaciones para la institución.

El personal directivo y administrativo del Instituto Nacional de Aguilares una vez concluida la instalación en su totalidad del sistema antes mencionado debe procurar tener los lineamientos adecuados tanto con los alumnos y los componentes del control de ambientación y timbrado para optimizar su vida útil, diseñando un procedimiento de su personal para la vigilancia de los puntos terminales o componentes eléctricos ya que como se mencionó anteriormente en la situación problemática, los alumnos forman parte en el deterioro de los inmuebles de la institución por esta razón se hace necesario el diseño de una vigilancia.

La administración de la institución debe motivar a cada uno de los alumnos a dar un uso y cuidado adecuado de la infraestructura de la institución especialmente a los equipos que son parte de la institución esto forma parte importante de las recomendaciones, así como extender los estudios en este proyecto para que los mismos estudiantes tomen la iniciativa de desarrollar sistemas que cumplan características similares o mejoras al mismo y retomar este proyecto en días futuros.

Recomendaciones para el equipo.

- Mantenerlo a un nivel de temperatura adecuado.
- Desarrollar un sistema de periódico de mantenimiento.
- Proporcionarle un suministro eléctrico adecuado. (usar UPS)
- No configurar los controles sin previo conocimiento de estos.

Bibliografía

Andres, N. (9 de abril de 2010). *tipos de.org*. Obtenido de <http://www.tiposde.org/cotidianos/568-tipos-de-manuales/>

Colmenares, L. (11 de 04 de 2010). *Sistemas, tipos y clasificacion*. Recuperado el 24 de 03 de 2013, de <http://informatica-colegiom.foroactivos.net/t13-sistemas-tipos-y-clasificacion>

Moya, J. M. (2010). *Huidobro y Asociados*. Obtenido de http://books.google.com.gt/books?id=V6lzqqDcfF8C&printsec=frontcover&dq=domotica&hl=en&sa=X&ei=fLpJUfqBI4Si2wXitlGYDQ&redir_esc=y#v=onepage&q&f=false

Romero, C., Vázquez, F., & Castro, C. d. (2011). *Domotica e Inmotica vivienda y edificios intelignerte 3ª Edicion*. México D.F.: Alfaomega Grupo Editor, S.A. .

ANEXOS

Matriz de Congruencia

MATRIZ DE CONGRUENCIA		
Tema: Automatización del control de ambientación y timbrado desde una aplicación para PC desarrollado para sistemas operativos Windows como apoyo a las labores repetitivas del Instituto Nacional de Aguilares.		
Enunciado del problema: ¿Sera posible crear un sistema Domótico basado en sistemas digitales programables, desarrollado para ser controlado desde una computadora para la automatización del Instituto Nacional de Aguilares que controle la ambientación de áreas específicas de dicho recinto?		
Objetivo general: Desarrollar un sistema de automatización que controle la ambientación y el timbrado desde una aplicación para PC, desarrollada para sistemas operativos Windows, como apoyo a las labores repetitivas del Instituto Nacional de Aguilares.		
Objetivo específico 1: Crear un circuito electrónico basado en sistemas digitales programables con aplicación en la Domótica que sea controlado desde una computadora del Instituto Nacional de Aguilares.	Objetivo específico 2: Adecuar e instalar el sistema Domótico en las instalaciones del Instituto Nacional de Aguilares	Objetivo específico 3: Diseñar un manual digital e impreso para el manejo óptimo tanto del programa como de la instalación.
Alcance 1: Desarrollar un circuito electrónico controlado desde una computadora para el manejo de la ambientación del instituto nacional de Aguilares.	Alcance 2: Adecuación e Instalación del Sistema Domótico demostración de que el circuito creado funciona de forma adecuada	Alcances 3: Diseñar un manual digital e impreso para el manejo óptimo del Sistema Domótico instalado
Promesa 1: Circuito eléctrico basado en ARDUINO con una interface de control de voltajes medios.	Promesa 2: Adecuación e Instalación del Sistema Domótico en las instalaciones del Instituto Nacional de Aguilares	Promesa 3: crear un manual digital e impreso

Firmware de control guardado en la memoria de arduino desarrollado en arduino ID

```
/////////////////////////////////CONTROL DE AMBIENTACION Y TIMBRADO/////////////////////////////////
/////////////////////////////////DKS_DESING/////////////////////////////////
/////////////////////////////////UNIVERSIDAD TECNOLOGICA DE EL SALCADOR/////////////////////////////////
/////////////////////////////////DESARROLLADO POR KEVIN RAUDA //////////////////////////////////
/////////////////////////////////EN COLABORACION CON EL ING. OTONIEL FLOREZ CORTEZ/////////////////////////////////
/////////////////////////////////gordo.rauda@gmail.com/////////////////////////////////
```

```
#include <EEPROM.h> //agregando libreria EEPROM
#include <Wire.h>
#include "RTCLib.h" //agregando libreria para el uso de Reloj a Tiempo Real (RTC)
RTC_DS1307 RTC;
```

```
#include "LedControl.h" //agregando libreria para el control de Max7219
LedControl DS= LedControl(45,47,49,1); // Definiendo pines de control Arduino - max
```

```
//Dimencionando variables para imprimir horas y minutos en el display
int horas =0;
int u_horas =0;
int segundo =0;
int d_horas =0;
int minutos =0;
int u_minutos =0;
int d_minutos =0;
```

```
// Dimencionando variables para el ingreso de datos
int ordenIngreso = 1;
int valorLeido = 0;
int dia = 0;
```

```
//Dimensionando variables para almacenar las horas y minutos ingresados en la pc
int ala1hrs = 0; //HORAS DEL ALARMA 1
int ala1min = 0; //MINUTOS DE ALARMA 1
int ala2hrs = 0; //HORAS DEL ALARMA 2
int ala2min = 0; //MINUTOS DE ALARMA 2
int ala3hrs = 0; //HORAS DEL ALARMA 3
int ala3min = 0; //MINUTOS DE ALARMA 3
int ala4hrs = 0; //HORAS DEL ALARMA 4
int ala4min = 0; //MINUTOS DE ALARMA 4
int ala5hrs = 0; //HORAS DEL ALARMA 5
int ala5min = 0; //MINUTOS DE ALARMA 5
```

```

int ala6hrs = 0; //HORAS DEL ALARMA 6
int ala6min = 0; //MINUTOS DE ALARMA 6
int ala7hrs = 0; //HORAS DEL ALARMA 7
int ala7min = 0; //MINUTOS DE ALARMA 7
int ala8hrs = 0; //HORAS DEL ALARMA 8
int ala8min = 0; //MINUTOS DE ALARMA 8
int ala9hrs = 0; //HORAS DEL ALARMA 9
int ala9min = 0; //MINUTOS DE ALARMA 9
int ala10hrs = 0; //HORAS DEL ALARMA 10
int ala10min = 0; //MINUTOS DE ALARMA 10
int ala11hrs = 0; //HORAS DEL ALARMA 11
int ala11min = 0; //MINUTOS DE ALARMA 11
int ala12hrs = 0; //HORAS DEL ALARMA 12
int ala12min = 0; //MINUTOS DE ALARMA 12
int ala13hrs = 0; //HORAS DEL ALARMA 13
int ala13min = 0; //MINUTOS DE ALARMA 13
int ala14hrs = 0; //HORAS DEL ALARMA 14
int ala14min = 0; //MINUTOS DE ALARMA 14
int ala15hrs = 0; //HORAS DEL ALARMA 15
int ala15min = 0; //MINUTOS DE ALARMA 15
int ala16hrs = 0; //HORAS DEL ALARMA 16
int ala16min = 0; //MINUTOS DE ALARMA 16
int ala17hrs = 0; //HORAS DEL ALARMA 17
int ala17min = 0; //MINUTOS DE ALARMA 17

```

```

void setup(){

```

```

    // Declarando pines de salida del arduino

```

```

    //sectores del auditorio módulo relay de 8
    pinMode(25, OUTPUT); // ventilador
    pinMode(24, OUTPUT); //timbre
    pinMode(23, OUTPUT); //sec 1
    pinMode(22, OUTPUT); //sec 2
    pinMode(19, OUTPUT); //sec 3
    pinMode(18, OUTPUT); //sec 4
    // grados del auditorio
    pinMode(13, OUTPUT); //2gral a
    pinMode(12, OUTPUT); //2tec a
    pinMode(11, OUTPUT); //2tec b
    pinMode(10, OUTPUT); //2gral b
    pinMode(9, OUTPUT); //1gral c
    pinMode(8, OUTPUT); //3 secre a

```

```

    // grados parte posterior
    pinMode(17, OUTPUT); //aula meca
    pinMode(16, OUTPUT); //1gral b
    pinMode(15, OUTPUT); //2tec c

```

```

pinMode(14, OUTPUT); //3cont b
pinMode(0, OUTPUT); //1 gral a
pinMode(1, OUTPUT); //pin libre

//gados parte delantera
pinMode(2, OUTPUT); //palomar up
pinMode(3, OUTPUT); //palomar1
pinMode(4, OUTPUT); //palomar2
pinMode(7, OUTPUT); //palomar3
pinMode(6, OUTPUT); //pin libre
pinMode(5, OUTPUT); //pin libre

// Escribiendo en la memoria EEPROM del arduino
EEPROM.write(ala1hrs,0);
EEPROM.write(ala1min,1);
EEPROM.write(ala2hrs,2);
EEPROM.write(ala2min,3);
EEPROM.write(ala3hrs,4);
EEPROM.write(ala3min,5);
EEPROM.write(ala4hrs,6);
EEPROM.write(ala4min,7);
EEPROM.write(ala5hrs,8);
EEPROM.write(ala5min,9);
EEPROM.write(ala6hrs,10);
EEPROM.write(ala6min,11);
EEPROM.write(ala7hrs,12);
EEPROM.write(ala7min,13);
EEPROM.write(ala8hrs,14);
EEPROM.write(ala8min,15);
EEPROM.write(ala9hrs,16);
EEPROM.write(ala9min,17);
EEPROM.write(ala10hrs,18);
EEPROM.write(ala10min,19);
EEPROM.write(ala11hrs,20);
EEPROM.write(ala11min,21);
EEPROM.write(ala12hrs,22);
EEPROM.write(ala12min,23);
EEPROM.write(ala13hrs,24);
EEPROM.write(ala13min,25);
EEPROM.write(ala14hrs,26);
EEPROM.write(ala14min,27);
EEPROM.write(ala15hrs,28);
EEPROM.write(ala15min,29);
EEPROM.write(ala16hrs,30);
EEPROM.write(ala16min,31);
EEPROM.write(ala17hrs,31);
EEPROM.write(ala17min,33);
// Abriemndo el puerto serie
Serial.begin(9600);

```

```

Wire.begin();
  RTC.begin();

  DS.shutdown(0,false);
  DS.setIntensity(0,15);
  DS.clearDisplay(0);
  //lcd.begin(16, 2);
}

void loop(){

  reloj();
  //leyendo datos almacenados en memoria EEPROM
  EEPROM.read(ala1hrs);
  EEPROM.read(ala1min);
  EEPROM.read(ala2hrs);
  EEPROM.read(ala2min);
  EEPROM.read(ala3hrs);
  EEPROM.read(ala3min);
  EEPROM.read(ala4hrs);
  EEPROM.read(ala4min);
  EEPROM.read(ala5hrs);
  EEPROM.read(ala5min);
  EEPROM.read(ala6hrs);
  EEPROM.read(ala6min);
  EEPROM.read(ala7hrs);
  EEPROM.read(ala7min);
  EEPROM.read(ala8hrs);
  EEPROM.read(ala8min);
  EEPROM.read(ala9hrs);
  EEPROM.read(ala9min);
  EEPROM.read(ala10hrs);
  EEPROM.read(ala10min);
  EEPROM.read(ala11hrs);
  EEPROM.read(ala11min);
  EEPROM.read(ala12hrs);
  EEPROM.read(ala12min);
  EEPROM.read(ala13hrs);
  EEPROM.read(ala13min);
  EEPROM.read(ala14hrs);
  EEPROM.read(ala14min);
  EEPROM.read(ala15hrs);
  EEPROM.read(ala15min);
  EEPROM.read(ala16hrs);
  EEPROM.read(ala16min);
  EEPROM.read(ala17hrs);
  EEPROM.read(ala17min);

```

```

//declarando condiciones para la comparación de horas y minutos
{
    if (horas == ala1hrs && minutos == ala1min && dia != 1 && dia !=7){
        digitalWrite(24, HIGH);
        delay (15000);
        digitalWrite(24, LOW);
        delay (50000);
    }
    //fin comparacion N°1
    if (horas == ala2hrs && minutos == ala2min && dia != 1 && dia !=7){
        digitalWrite(24, HIGH);
        delay (15000);
        digitalWrite(24, LOW);
        delay (50000);
    }
    //fin comparacion N°2
    if (horas == ala3hrs && minutos == ala3min && dia != 1 && dia !=7){
        digitalWrite(24, HIGH);
        delay (15000);
        digitalWrite(24, LOW);
        delay (50000);
    }
    //fin comparacion N°3
    if (horas == ala4hrs && minutos == ala4min && dia != 1 && dia !=7){
        digitalWrite(24, HIGH);
        delay (15000);
        digitalWrite(24, LOW);
        delay (50000);
    }
    //fin comparacion N°4
    if (horas == ala5hrs && minutos == ala5min && dia != 1 && dia !=7){
        digitalWrite(24, HIGH);
        delay (15000);
        digitalWrite(24, LOW);
        delay (50000);
    }
    //fin comparacion N°5
    if (horas == ala6hrs && minutos == ala6min && dia != 1 && dia !=7){
        digitalWrite(24, HIGH);
        delay (15000);
        digitalWrite(24, LOW);
        delay (50000);
    }
    //fin comparacion N°6
    if (horas == ala7hrs && minutos == ala7min && dia != 1 && dia !=7){
        digitalWrite(24, HIGH);
        delay (15000);
        digitalWrite(24, LOW);
        delay (50000);
    }
}

```

```

}
//fin comparacion N°7
if (horas == ala8hrs && minutos == ala8min && dia != 1 && dia !=7){
    digitalWrite(24, HIGH);
    delay (15000);
    digitalWrite(24, LOW);
    delay (50000);
}
//fin comparacion N°8
if (horas == ala9hrs && minutos == ala9min && dia != 1 && dia !=7){
    digitalWrite(24, HIGH);
    delay (15000);
    digitalWrite(24, LOW);
    delay (50000);
}
//fin comparacion N°9
if (horas == ala10hrs && minutos == ala10min && dia != 1 && dia !=7){
    digitalWrite(24, HIGH);
    delay (15000);
    digitalWrite(24, LOW);
    delay (50000);
}
//fin comparacion N°10
if (horas == ala11hrs && minutos == ala11min && dia != 1 && dia !=7){
    digitalWrite(24, HIGH);
    delay (15000);
    digitalWrite(24, LOW);
    delay (50000);
}
//fin comparacion N°11
if (horas == ala12hrs && minutos == ala12min && dia != 1 && dia !=7){
    digitalWrite(24, HIGH);
    delay (15000);
    digitalWrite(24, LOW);
    delay (50000);
}
//fin comparacion N°12
if (horas == ala13hrs && minutos == ala13min && dia != 1 && dia !=7){
    digitalWrite(24, HIGH);
    delay (15000);
    digitalWrite(24, LOW);
    delay (50000);
}
//fin comparacion N°13
if (horas == ala14hrs && minutos == ala14min && dia != 1 && dia !=7){
    digitalWrite(24, HIGH);
    delay (15000);
    digitalWrite(24, LOW);
    delay (50000);
}

```

```

}
//fin comparacion N°14
if (horas == ala15hrs && minutos == ala15min && dia != 1 && dia !=7){
    digitalWrite(24, HIGH);
    delay (15000);
    digitalWrite(24, LOW);
    delay (50000);
}
//fin comparacion N°15
if (horas == ala16hrs && minutos == ala16min && dia != 1 && dia !=7){
    digitalWrite(24, HIGH);
    delay (15000);
    digitalWrite(24, LOW);
    delay (50000);
}
//fin comparacion N°16
if (horas == ala17hrs && minutos == ala17min && dia != 1 && dia !=7){
    digitalWrite(24, HIGH);
    delay (15000);
    digitalWrite(24, LOW);
    delay (50000);
}
//fin comparacion N°17

// Definiendo acciones a tomar si el puerto serie recibe una orden
if (Serial.available()) {
    delay(100);
    while (Serial.available())
    {
        int inByte = Serial.read();

        if(inByte == '@'){
            RTC.adjust(DateTime(__DATE__, __TIME__));
        }

        if(inByte== '1'){
            digitalWrite(23,1);
        }
        if (inByte=='2'){
            digitalWrite(23,0);
        }
        if(inByte== '3'){
            digitalWrite(22,1);
        }
        if(inByte== '4'){
            digitalWrite(22,0);
        }
        if(inByte== '5'){
            digitalWrite(19,1);
        }
    }
}

```

```

    }
    if(inByte== '6'){
        digitalWrite(19,0);
    }
    if(inByte== '7'){
        digitalWrite(18,1);
    }
    if(inByte== '8'){
        digitalWrite(18,0);
    }
    if(inByte== '9'){
        digitalWrite(23,1);
        digitalWrite(22,1);
        digitalWrite(19,1);
        digitalWrite(18,1);
    }

    if(inByte== '0'){
        digitalWrite(23,0);
        digitalWrite(22,0);
        digitalWrite(19,0);
        digitalWrite(18,0);
    }
}

```

```

// Recepcion de horas y minutos enviados por el puerto serie desde
// la interface de usuario
//ALARMANº1
if(inByte== 'A'){

    while (Serial.available()) {

        if (ordenIngreso ==1)
        {
            valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
            ordenIngreso++;
        }

        if (ordenIngreso == 2)
        {
            valorLeido = Serial.read(); valorLeido = (valorLeido - '0'); //se convierte a su verdadero valor
numérico
            ala1hrs = valorLeido * 10;

            ordenIngreso++;
        }
        if (ordenIngreso == 3)
        {

```

```

    valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
    ala1hrs = valorLeido + ala1hrs;
    ordenIngreso++;
}
if (ordenIngreso == 4)
{
    valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
    ordenIngreso++;
}
if (ordenIngreso == 5)
{
    valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
    ala1min = valorLeido * 10;
    ordenIngreso++;
}
if (ordenIngreso == 6)
{
    valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
    ala1min = valorLeido + ala1min;
    ordenIngreso++;
}
if (ordenIngreso == 7)
{
    valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
    ordenIngreso++;
}
}
ordenIngreso = 1;
}
//FIN ALARMA 1

//ALARMA N°2
if(inByte== 'B'){

    while (Serial.available()) {

        if (ordenIngreso ==1)
        {
            valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
            ordenIngreso++;
        }

        if (ordenIngreso == 2)
        {
            valorLeido = Serial.read(); valorLeido = (valorLeido - '0'); //se convierte a su verdadero valor
numérico
            ala2hrs = valorLeido * 10;

            ordenIngreso++;

```

```

    }
    if (ordenIngreso == 3)
    {
        valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
        ala2hrs = valorLeido + ala2hrs;
        ordenIngreso++;
    }
    if (ordenIngreso == 4)
    {
        valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
        ordenIngreso++;
    }
    if (ordenIngreso == 5)
    {
        valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
        ala2min = valorLeido * 10;
        ordenIngreso++;
    }
    if (ordenIngreso == 6)
    {
        valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
        ala2min = valorLeido + ala2min;
        ordenIngreso++;
    }
    if (ordenIngreso == 7)
    {
        valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
        ordenIngreso++;
    }
}
ordenIngreso = 1;
}

```

```

//ALARMANº3
if(inByte== 'C'){

    while (Serial.available()) {

        if (ordenIngreso ==1)
        {
            valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
            ordenIngreso++;
        }

        if (ordenIngreso == 2)
        {
            valorLeido = Serial.read(); valorLeido = (valorLeido - '0'); //se convierte a su verdadero valor
            numerico

```

```

    ala3hrs = valorLeido * 10;

    ordenIngreso++;
}
if (ordenIngreso == 3)
{
    valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
    ala3hrs = valorLeido + ala3hrs;
    ordenIngreso++;
}
if (ordenIngreso == 4)
{
    valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
    ordenIngreso++;
}
if (ordenIngreso == 5)
{
    valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
    ala3min = valorLeido * 10;
    ordenIngreso++;
}
if (ordenIngreso == 6)
{
    valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
    ala3min = valorLeido + ala3min;
    ordenIngreso++;
}
if (ordenIngreso == 7)
{
    valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
    ordenIngreso++;
}
}
ordenIngreso = 1;
}

//ALARMANº4
if(inByte== 'D'){

    while (Serial.available()) {

        if (ordenIngreso ==1)
        {
            valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
            ordenIngreso++;
        }

        if (ordenIngreso == 2)
        {

```

```

        valorLeido = Serial.read(); valorLeido = (valorLeido - '0'); //se convierte a su verdadero valor
numérico
        ala4hrs = valorLeido * 10;

        ordenIngreso++;
    }
    if (ordenIngreso == 3)
    {
        valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
        ala4hrs = valorLeido + ala4hrs;
        ordenIngreso++;
    }
    if (ordenIngreso == 4)
    {
        valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
        ordenIngreso++;
    }
    if (ordenIngreso == 5)
    {
        valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
        ala4min = valorLeido * 10;
        ordenIngreso++;
    }
    if (ordenIngreso == 6)
    {
        valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
        ala4min = valorLeido + ala4min;
        ordenIngreso++;
    }
    if (ordenIngreso == 7)
    {
        valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
        ordenIngreso++;
    }
}
ordenIngreso = 1;
}
//ALARMAN°5
if(inByte== 'E'){

    while (Serial.available()) {

        if (ordenIngreso ==1)
        {
            valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
            ordenIngreso++;
        }

        if (ordenIngreso == 2)

```

```

    {
        valorLeido = Serial.read(); valorLeido = (valorLeido - '0'); //se convierte a su verdadero valor
numérico
        ala5hrs = valorLeido * 10;

        ordenIngreso++;
    }
    if (ordenIngreso == 3)
    {
        valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
        ala5hrs = valorLeido + ala5hrs;
        ordenIngreso++;
    }
    if (ordenIngreso == 4)
    {
        valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
        ordenIngreso++;
    }
    if (ordenIngreso == 5)
    {
        valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
        ala5min = valorLeido * 10;
        ordenIngreso++;
    }
    if (ordenIngreso == 6)
    {
        valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
        ala5min = valorLeido + ala5min;
        ordenIngreso++;
    }
    if (ordenIngreso == 7)
    {
        valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
        ordenIngreso++;
    }
}
ordenIngreso = 1;
}
//ALARMAN°6
if(inByte== 'F'){

    while (Serial.available()) {

        if (ordenIngreso ==1)
        {
            valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
            ordenIngreso++;
        }
    }
}

```

```

    if (ordenIngreso == 2)
    {
        valorLeido = Serial.read(); valorLeido = (valorLeido - '0'); //se convierte a su verdadero valor
numérico
        ala6hrs = valorLeido * 10;

        ordenIngreso++;
    }
    if (ordenIngreso == 3)
    {
        valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
        ala6hrs = valorLeido + ala6hrs;
        ordenIngreso++;
    }
    if (ordenIngreso == 4)
    {
        valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
        ordenIngreso++;
    }
    if (ordenIngreso == 5)
    {
        valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
        ala6min = valorLeido * 10;
        ordenIngreso++;
    }
    if (ordenIngreso == 6)
    {
        valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
        ala6min = valorLeido + ala6min;
        ordenIngreso++;
    }
    if (ordenIngreso == 7)
    {
        valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
        ordenIngreso++;
    }
}
ordenIngreso = 1;
}
//ALARMANº7
if(inByte== 'G'){

    while (Serial.available()) {

        if (ordenIngreso ==1)
        {
            valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
            ordenIngreso++;
        }
    }
}

```

```

    if (ordenIngreso == 2)
    {
        valorLeido = Serial.read(); valorLeido = (valorLeido - '0'); //se convierte a su verdadero valor
numérico
        ala7hrs = valorLeido * 10;

        ordenIngreso++;
    }
    if (ordenIngreso == 3)
    {
        valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
        ala7hrs = valorLeido + ala7hrs;
        ordenIngreso++;
    }
    if (ordenIngreso == 4)
    {
        valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
        ordenIngreso++;
    }
    if (ordenIngreso == 5)
    {
        valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
        ala7min = valorLeido * 10;
        ordenIngreso++;
    }
    if (ordenIngreso == 6)
    {
        valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
        ala7min = valorLeido + ala7min;
        ordenIngreso++;
    }
    if (ordenIngreso == 7)
    {
        valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
        ordenIngreso++;
    }
}
ordenIngreso = 1;
}
//ALARMANº8
if(inByte== 'H'){

    while (Serial.available()) {

        if (ordenIngreso ==1)
        {
            valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
            ordenIngreso++;

```

```

    }

    if (ordenIngreso == 2)
    {
        valorLeido = Serial.read(); valorLeido = (valorLeido - '0'); //se convierte a su verdadero valor
numérico
        ala8hrs = valorLeido * 10;

        ordenIngreso++;
    }
    if (ordenIngreso == 3)
    {
        valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
        ala8hrs = valorLeido + ala8hrs;
        ordenIngreso++;
    }
    if (ordenIngreso == 4)
    {
        valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
        ordenIngreso++;
    }
    if (ordenIngreso == 5)
    {
        valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
        ala8min = valorLeido * 10;
        ordenIngreso++;
    }
    if (ordenIngreso == 6)
    {
        valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
        ala8min = valorLeido + ala8min;
        ordenIngreso++;
    }
    if (ordenIngreso == 7)
    {
        valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
        ordenIngreso++;
    }
}
ordenIngreso = 1;
}
//ALARMANº9
if(inByte=='I'){

    while (Serial.available()) {

        if (ordenIngreso ==1)
        {
            valorLeido = Serial.read(); valorLeido = (valorLeido - '0');

```

```

ordenIngreso++;
}

if (ordenIngreso == 2)
{
    valorLeido = Serial.read(); valorLeido = (valorLeido - '0'); //se convierte a su verdadero valor
numérico
    ala9hrs = valorLeido * 10;

    ordenIngreso++;
}
if (ordenIngreso == 3)
{
    valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
    ala9hrs = valorLeido + ala9hrs;
    ordenIngreso++;
}
if (ordenIngreso == 4)
{
    valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
    ordenIngreso++;
}
if (ordenIngreso == 5)
{
    valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
    ala9min = valorLeido * 10;
    ordenIngreso++;
}
if (ordenIngreso == 6)
{
    valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
    ala9min = valorLeido + ala9min;
    ordenIngreso++;
}
if (ordenIngreso == 7)
{
    valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
    ordenIngreso++;
}
}
ordenIngreso = 1;
}
//ALARMAN*10
if(inByte=='J'){

    while (Serial.available()) {

        if (ordenIngreso ==1)
        {

```

```

        valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
        ordenIngreso++;
    }

    if (ordenIngreso == 2)
    {
        valorLeido = Serial.read(); valorLeido = (valorLeido - '0'); //se convierte a su verdadero valor
numérico
        ala10hrs = valorLeido * 10;

        ordenIngreso++;
    }
    if (ordenIngreso == 3)
    {
        valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
        ala10hrs = valorLeido + ala10hrs;
        ordenIngreso++;
    }
    if (ordenIngreso == 4)
    {
        valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
        ordenIngreso++;
    }
    if (ordenIngreso == 5)
    {
        valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
        ala10min = valorLeido * 10;
        ordenIngreso++;
    }
    if (ordenIngreso == 6)
    {
        valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
        ala10min = valorLeido + ala10min;
        ordenIngreso++;
    }
    if (ordenIngreso == 7)
    {
        valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
        ordenIngreso++;
    }
}
ordenIngreso = 1;
}
//ALARMAN°11
if(inByte== 'K'){

    while (Serial.available()) {

        if (ordenIngreso ==1)

```

```

    {
        valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
        ordenIngreso++;
    }

    if (ordenIngreso == 2)
    {
        valorLeido = Serial.read(); valorLeido = (valorLeido - '0'); //se convierte a su verdadero valor
numérico
        ala11hrs = valorLeido * 10;

        ordenIngreso++;
    }
    if (ordenIngreso == 3)
    {
        valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
        ala11hrs = valorLeido + ala11hrs;
        ordenIngreso++;
    }
    if (ordenIngreso == 4)
    {
        valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
        ordenIngreso++;
    }
    if (ordenIngreso == 5)
    {
        valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
        ala11min = valorLeido * 10;
        ordenIngreso++;
    }
    if (ordenIngreso == 6)
    {
        valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
        ala11min = valorLeido + ala11min;
        ordenIngreso++;
    }
    if (ordenIngreso == 7)
    {
        valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
        ordenIngreso++;
    }
}
ordenIngreso = 1;
}
//ALARMAN°12
if(inByte== 'L'){

    while (Serial.available()) {

```

```

    if (ordenIngreso == 1)
    {
        valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
        ordenIngreso++;
    }

    if (ordenIngreso == 2)
    {
        valorLeido = Serial.read(); valorLeido = (valorLeido - '0'); //se convierte a su verdadero valor
numérico
        ala12hrs = valorLeido * 10;

        ordenIngreso++;
    }
    if (ordenIngreso == 3)
    {
        valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
        ala12hrs = valorLeido + ala12hrs;
        ordenIngreso++;
    }
    if (ordenIngreso == 4)
    {
        valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
        ordenIngreso++;
    }
    if (ordenIngreso == 5)
    {
        valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
        ala12min = valorLeido * 10;
        ordenIngreso++;
    }
    if (ordenIngreso == 6)
    {
        valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
        ala12min = valorLeido + ala12min;
        ordenIngreso++;
    }
    if (ordenIngreso == 7)
    {
        valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
        ordenIngreso++;
    }
}
ordenIngreso = 1;
}
//ALARMAN°13
if(inByte== 'M'){

    while (Serial.available()) {

```

```

    if (ordenIngreso == 1)
    {
        valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
        ordenIngreso++;
    }

    if (ordenIngreso == 2)
    {
        valorLeido = Serial.read(); valorLeido = (valorLeido - '0'); //se convierte a su verdadero valor
numérico
        ala13hrs = valorLeido * 10;

        ordenIngreso++;
    }
    if (ordenIngreso == 3)
    {
        valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
        ala13hrs = valorLeido + ala13hrs;
        ordenIngreso++;
    }
    if (ordenIngreso == 4)
    {
        valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
        ordenIngreso++;
    }
    if (ordenIngreso == 5)
    {
        valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
        ala13min = valorLeido * 10;
        ordenIngreso++;
    }
    if (ordenIngreso == 6)
    {
        valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
        ala13min = valorLeido + ala13min;
        ordenIngreso++;
    }
    if (ordenIngreso == 7)
    {
        valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
        ordenIngreso++;
    }
}
ordenIngreso = 1;
}
//ALARMAN°14
if(inByte== 'N'){

```

```

while (Serial.available()) {

    if (ordenIngreso ==1)
    {
        valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
        ordenIngreso++;
    }

    if (ordenIngreso == 2)
    {
        valorLeido = Serial.read(); valorLeido = (valorLeido - '0'); //se convierte a su verdadero valor
        numerico
        ala14hrs = valorLeido * 10;

        ordenIngreso++;
    }
    if (ordenIngreso == 3)
    {
        valorLeido = Serial.read();valorLeido = (valorLeido - '0');
        ala14hrs = valorLeido + ala14hrs;
        ordenIngreso++;
    }
    if (ordenIngreso == 4)
    {
        valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
        ordenIngreso++;
    }
    if (ordenIngreso == 5)
    {
        valorLeido = Serial.read();valorLeido = (valorLeido - '0');
        ala14min = valorLeido * 10;
        ordenIngreso++;
    }
    if (ordenIngreso == 6)
    {
        valorLeido = Serial.read();valorLeido = (valorLeido - '0');
        ala14min = valorLeido + ala14min;
        ordenIngreso++;
    }
    if (ordenIngreso == 7)
    {
        valorLeido = Serial.read();valorLeido = (valorLeido - '0');
        ordenIngreso++;
    }
}
ordenIngreso = 1;
}
//ALARMAN°15
if(inByte== 'O'){

```

```

while (Serial.available()) {

  if (ordenIngreso ==1)
  {
    valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
    ordenIngreso++;
  }

  if (ordenIngreso == 2)
  {
    valorLeido = Serial.read(); valorLeido = (valorLeido - '0'); //se convierte a su verdadero valor
numérico
    ala15hrs = valorLeido * 10;

    ordenIngreso++;
  }
  if (ordenIngreso == 3)
  {
    valorLeido = Serial.read();valorLeido = (valorLeido - '0');
    ala15hrs = valorLeido + ala15hrs;
    ordenIngreso++;
  }
  if (ordenIngreso == 4)
  {
    valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
    ordenIngreso++;
  }
  if (ordenIngreso == 5)
  {
    valorLeido = Serial.read();valorLeido = (valorLeido - '0');
    ala15min = valorLeido * 10;
    ordenIngreso++;
  }
  if (ordenIngreso == 6)
  {
    valorLeido = Serial.read();valorLeido = (valorLeido - '0');
    ala15min = valorLeido + ala15min;
    ordenIngreso++;
  }
  if (ordenIngreso == 7)
  {
    valorLeido = Serial.read();valorLeido = (valorLeido - '0');
    ordenIngreso++;
  }
}
ordenIngreso = 1;
}
//ALARMAN°16

```

```

if(inByte== 'P'){

    while (Serial.available()) {

        if (ordenIngreso ==1)
        {
            valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
            ordenIngreso++;
        }

        if (ordenIngreso == 2)
        {
            valorLeido = Serial.read(); valorLeido = (valorLeido - '0'); //se convierte a su verdadero valor
numerico
            ala16hrs = valorLeido * 10;

            ordenIngreso++;
        }
        if (ordenIngreso == 3)
        {
            valorLeido = Serial.read();valorLeido = (valorLeido - '0');
            ala16hrs = valorLeido + ala16hrs;
            ordenIngreso++;
        }
        if (ordenIngreso == 4)
        {
            valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
            ordenIngreso++;
        }
        if (ordenIngreso == 5)
        {
            valorLeido = Serial.read();valorLeido = (valorLeido - '0');
            ala16min = valorLeido * 10;
            ordenIngreso++;
        }
        if (ordenIngreso == 6)
        {
            valorLeido = Serial.read();valorLeido = (valorLeido - '0');
            ala16min = valorLeido + ala16min;
            ordenIngreso++;
        }
        if (ordenIngreso == 7)
        {
            valorLeido = Serial.read();valorLeido = (valorLeido - '0');
            ordenIngreso++;
        }
    }
ordenIngreso = 1;
}

```

```

//ALARMAN°17
if(inByte== 'Q'){

    while (Serial.available()) {

        if (ordenIngreso ==1)
        {
            valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
            ordenIngreso++;
        }

        if (ordenIngreso == 2)
        {
            valorLeido = Serial.read(); valorLeido = (valorLeido - '0'); //se convierte a su verdadero valor
numerico
            ala17hrs = valorLeido * 10;

            ordenIngreso++;
        }
        if (ordenIngreso == 3)
        {
            valorLeido = Serial.read();valorLeido = (valorLeido - '0');
            ala17hrs = valorLeido + ala17hrs;
            ordenIngreso++;
        }
        if (ordenIngreso == 4)
        {
            valorLeido = Serial.read(); valorLeido = (valorLeido - '0');
            ordenIngreso++;
        }
        if (ordenIngreso == 5)
        {
            valorLeido = Serial.read();valorLeido = (valorLeido - '0');
            ala17min = valorLeido * 10;
            ordenIngreso++;
        }
        if (ordenIngreso == 6)
        {
            valorLeido = Serial.read();valorLeido = (valorLeido - '0');
            ala17min = valorLeido + ala17min;
            ordenIngreso++;
        }
        if (ordenIngreso == 7)
        {
            valorLeido = Serial.read();valorLeido = (valorLeido - '0');
            ordenIngreso++;
        }
    }
ordenIngreso = 1;

```

```

}

if(inByte== 'a'){
    digitalWrite(13,1);
}
if (inByte=='b'){
    digitalWrite(13,0);
}
if(inByte== 'c'){
    digitalWrite(12,1);
}
if (inByte=='d'){
    digitalWrite(12,0);
}
if(inByte== 'e'){
    digitalWrite(11,1);
}
if (inByte=='f'){
    digitalWrite(11,0);
}
if(inByte== 'g'){
    digitalWrite(10,1);
}
if (inByte=='h'){
    digitalWrite(10,0);
}
if(inByte== 'i'){
    digitalWrite(9,1);
}
if (inByte=='j'){
    digitalWrite(9,0);
}
if(inByte== 'k'){
    digitalWrite(8,1);
}
if (inByte=='l'){
    digitalWrite(8,0);
}

// graods posteriores
if(inByte== 'm'){
    digitalWrite(17,1);
}
if (inByte=='n'){
    digitalWrite(17,0);
}
if(inByte== 'o'){
    digitalWrite(16,1);
}

```

```

if (inByte=='p'){
    digitalWrite(16,0);
}
if(inByte== 'q'){
    digitalWrite(15,1);
}
if (inByte=='r'){
    digitalWrite(15,0);
}
if(inByte== 's'){
    digitalWrite(14,1);
}
if (inByte=='u'){
    digitalWrite(14,0);
}
if(inByte== 'v'){
    digitalWrite(0,1);
}
if (inByte=='w'){
    digitalWrite(0,0);
}

// aulas palomar
if(inByte== 'x'){
    digitalWrite(2,1);
}
if (inByte=='y'){
    digitalWrite(2,0);
}
if(inByte== 'z'){
    digitalWrite(3,1);
}
if (inByte=='#'){
    digitalWrite(3,0);
}
if(inByte== '$'){
    digitalWrite(4,1);
}
if (inByte=='%'){
    digitalWrite(4,0);
}
if(inByte== '&'){
    digitalWrite(7,1);
}
if (inByte=='='){
    digitalWrite(7,0);
}
}

```

```

}
}
}

// Secuencia para que el RTC guarde las horas y minutos en variables
// Para luego ser enviadas al Max7219
void reloj(){

    DateTime now = RTC.now();
    dia = now.day();
    horas = now.hour();
    u_horas = horas% 10;
    d_horas = horas/ 10;

    minutos = now.minute();
    u_minutos = minutos% 10;
    d_minutos = minutos/10;

    // Imprimiendo en el Display las horas y minutos del RTC
    DS.setDigit(0, 1, u_horas, true);
    DS.setDigit(0, 0, d_horas, true);
    DS.setDigit(0, 3, u_minutos, true);
    DS.setDigit(0, 2, d_minutos, true);

}

```

Carta de Aceptacion



INSTITUTO NACIONAL DE AGUILARES
Final calle la Ronda, Col las Salinas
Teléfono: 2124-7515

San Salvador, 4 de marzo del 2012.

A Quien Interese.

Reciba un cordial saludo de parte del Instituto Nacional de Aguilares, deseándole éxitos en su vida la laboral y personal.

Por medio de la presente dejamos constar que autorizamos la realización, del proyecto de graduación en nuestra institución, el cual tiene como tema: "Automatización del control de ambientación y timbrado desde una aplicación para PC desarrollado para sistemas operativos Windows 7 como apoyo a las labores repetitivas del Instituto Nacional de Aguilares". Avalado por el consejo Directivo Escolar.

Sin más que agradecer nos despedimos deseándoles Bendiciones y triunfos en sus proyectos.

Kevin David Rauda Solis

28-2651-2011

Hernán Ulises Alonzo Ortiz

28-2322-2011

Arnoldo Fabricio Molina Lemus

28-3437-2011



Wilber Antonio Hernández Reyes

Sub-Director INA

Carta de Agradecimiento



INSTITUTO NACIONAL DE AGUILARES
Final calle la Ronda
Colonia Salinas



San Salvador 3 de junio de 2013

A quien interese.

Reciba un cordial saludo, esperando tenga éxito en sus actividades diarias.

Por medio de la presente yo, Wilber Antonio Hernández Reyes desempeñando el cargo de Sub-director del Instituto Nacional de Aguilares, encargado de velar por la realización del proyecto anteriormente autorizado.

Hago constar y doy fe, que los alumnos: Kevin David Rauda, Hernán Ulises Ortiz y Arnoldo Fabricio Lemus, estudiantes de la universidad tecnológica, aspirantes al título de Técnicos en ingeniería Hardware. Desarrollaron un sistema de automatización de las labores previamente establecidas por la junta directiva de maestros y vistas como necesidades a solventar.

Cumpliendo de esta manera los objetivos previamente expresados a mi persona, tales como:

- El desarrollo de sistema de automatización.
- Implementación de dicho sistema.
- Un manual para el uso del mismo.

Lo anterior mente dicho fue realizado en el plazo establecido por los alumnos. Agradecemos grandemente como institución a los alumnos de la Universidad Tecnológica de El Salvador por tomarnos en cuenta, ya que dicho sistema, ayudará a ser más eficiente las labores diarias que se llevan a cabo en la institución

Sin nada mas que agregar me despido deseandoles bendiciones y éxitos. Atentamente.

Wilber Antonio Hernández Reyes
Sub-Director
INA

Por este medios los miembros del jurado evaluador del trabajo de graduación titulado:

AUTOMATIZACIÓN DEL CONTROL DE AMBIENTACIÓN Y TIMBRADO DESDE UNA APLICACIÓN PARA PC DESARROLLADO PARA SISTEMAS OPERATIVOS WINDOWS, COMO APOYO A LAS LABORES REPETITIVAS DEL INSTITUTO DE AGUILARES.

Presentado por:

- 1- ARNOLDO FABRICIO MOLINA LEMUS
- 2- HERNÁN ULISES ALONZO ORTIZ
- 3- KEVIN DAVID RAUDA SOLIS

Después de la defensa oral, hemos revisado el trabajo escrito y por lo tanto damos por Aprobado y finalizado el trabajo.

San Salvador, a los 24 días del mes de junio de 2013

F_____

Ing. María Eva Carranza

F_____

Ing. Omar Otoniel Flores

F_____

Ing. Julio César Rosales