

**Universidad Nacional de Ingeniería**

**Facultad de Electrotecnia y Computación**

**FEC**

**INFORME DE TRABAJO MONOGRÁFICO PARA OPTAR AL TITULO DE  
INGENIERO ELECTRÓNICO**

**Diseño e implementación de un prototipo de timbre  
electrónico programable para el correcto control de  
las horas clases en el centro educativo Monte Sinaí**

Elaborado por:

Br. Sarahi Yaoska Sandoval Sandoval.....2013-62325.

Br. Cristian Ernesto Solórzano Cano.....2013-61571.

Tutor:

Ing. Álvaro Antonio Gaitán.

Managua-Nicaragua

Marzo 2021

## Dedicatoria

Este trabajo Monográfico está dedicado primeramente a Dios quien ha sido mi guía y me dio la sabiduría para poder finalizar mi carrera y proveyó a mi madre todo lo que necesitaba en el transcurso de mis estudios.

A mi madre Ligia Johanna Cano Cruz quien ha sido mi soporte, porque siempre me ha dado su apoyo incondicional, por su coraje para sacarme adelante a pesar de las adversidades ya que gracias a ella he podido alcanzar cada meta propuesta, por formarme con buenos valores y ser una persona de bien. Agradezco a todas las personas que han sido parte de mi formación y que me han apoyado en los momentos más difíciles de mi vida.

Cristian Solórzano Cano

Agradezco a Dios por darme salud y por guiarme en cada paso que doy, por darme sabiduría para finalizar mi carrera y bendecir a cada de una de las personas que me apoyaron en el transcurso de mis estudios.

Gracias a mi madre María Elena Sandoval Ríos por ser mi sostén y por qué siempre me ha inspirado a ser una mejor persona, porque nunca dijo no cuando necesitaba algo para mis estudios, porque con su esfuerzo y dedicación hizo posible que yo pudiera finalizar mis estudios, por su amor incondicional y sobre todo por su fortaleza porque a pesar de las circunstancias adversas siempre permaneció firme apoyándome en todo lo que necesitaba, porque me motivaba a cumplir cada meta y está al pendiente que no me falte nada, gracias a ella soy una mujer de bien.

Agradezco a mis hermanos que siempre me alentaron y apoyaron para seguir adelante.

Sarahi Sandoval Sandoval

## Agradecimientos

Agradecemos principalmente a Dios por permitirnos finalizar nuestros estudios, a nuestros padres que han sido pilar fundamental para que nosotros estemos culminando nuestra carrera.

A la Sra. Martha Quiroz y Lic. Eveling Sánchez porque nos dieron la confianza y nos permitieron implementar nuestro proyecto en su centro de estudios.

Le damos gracias al tutor Ing. Álvaro Gaitán quien nos orientó y fue nuestro apoyo durante la elaboración de nuestro proyecto y por tenernos la paciencia para poder finalizar nuestra tesis.

## Resumen

El propósito de este trabajo monográfico es brindar una alternativa al centro escolar monte Sinaí para el correcto control de horas clases permitiendo al docente tener un mejor desempeño en sus horarios laborales al no estar activando manualmente el timbre, sirviendo este de ayuda para la problemática del centro. Para implementar este proyecto en el instituto primeramente se recopiló información en conjunto con la directora para saber con exactitud cómo abordar el tema.

El sistema es autónomo de tal forma que podrá brindar la hora y fecha en tiempo real para posterior activar el timbre a una hora establecida, estas horas son programadas de acuerdo al horario a convenir del centro y se tendrá a disposición una aplicación de manera que él o la responsable de verificar el correcto uso del sistema puede modificar los horarios de clase ante cualquier eventualidad.

En este documento se presenta el diseño del prototipo y las etapas en la que se ha ido construyendo así también la toma de decisiones de los componentes utilizados en el sistema

Este prototipo es manipulable para el usuario y de fácil acceso para realizar modificaciones en los horarios ya establecidos, solamente se inicia el encendido del mismo y automáticamente queda operando. En el transcurso de su elaboración se ejecutaron diferentes pruebas para evitar inconsistencias en su funcionamiento.

## Índice

<b>I.- Introducción.....</b>	<b>- 1 -</b>
<b>II. Justificación.....</b>	<b>- 2 -</b>
<b>III.- Objetivos. ....</b>	<b>- 4 -</b>
<b>1. General. ....</b>	<b>- 4 -</b>
<b>2. Específicos. ....</b>	<b>- 4 -</b>
<b>CAPÍTULO 1: MARCO TEÓRICO. ....</b>	<b>- 5 -</b>
<b>1.1. Timbre eléctrico.....</b>	<b>- 5 -</b>
<b>1.2. Sistemas Electrónicos Embebidos. ....</b>	<b>- 6 -</b>
<b>1.3. Estrategias de control.....</b>	<b>- 8 -</b>
<b>1.4. Sistemas de Control Automático.....</b>	<b>- 9 -</b>
<b>1.5. Componentes fundamentales de los sistemas de control. ....</b>	<b>- 9 -</b>
<b>1.6. Unidad de control y procesamiento.....</b>	<b>- 10 -</b>
<b>1.6.1. Microcontrolador.....</b>	<b>- 11 -</b>
<b>1.6.2. Reloj en Tiempo Real RTC).....</b>	<b>- 12 -</b>
<b>1.6.3. Actuadores.....</b>	<b>- 13 -</b>
<b>1.7. Interfaz Gráfica. ....</b>	<b>- 13 -</b>
<b>1.7.1. Human Machine Interface (HMI).....</b>	<b>- 13 -</b>
<b>1.7.2. Software para diseño de interfaces. ....</b>	<b>- 14 -</b>
<b>1.7.2.1. MIT App Inventor.....</b>	<b>- 14 -</b>
<b>1.7.2.2. Visual Basic.....</b>	<b>- 15 -</b>
<b>1.8. Sistemas de Comunicación.....</b>	<b>- 17 -</b>
<b>1.8.1. Dispositivos de Comunicación Inalámbrica ....</b>	<b>- 17 -</b>
<b>1.8.1.1. Módulo bluetooth HC-05 y HC-06.....</b>	<b>- 18 -</b>
<b>1.8.1.2. Módulo WIFI ESP-8266.....</b>	<b>- 22 -</b>
<b>1.8.2. Módulo transmisor-receptor asíncrono universal (UART).....</b>	<b>- 25 -</b>
<b>1.8.2.1. Protocolo RS232.....</b>	<b>- 26 -</b>
<b>1.9. Sistemas de respaldo de energía. ....</b>	<b>- 27 -</b>
<b>1.9.1. Inversores de corriente. ....</b>	<b>- 28 -</b>
<b>CAPITULO II: ANÁLISIS Y PRESENTACIÓN DE LOS RESULTADOS.....</b>	<b>- 31 -</b>
<b>2.1. Etapa de análisis. ....</b>	<b>- 32 -</b>
<b>2.2. Diseño del sistema. ....</b>	<b>- 34 -</b>

2.2.1.	Esquema general del sistema de control. ....	- 34 -
2.2.2.	Selección de los componentes principales del sistema. ....	- 35 -
2.3.	Selección de la unidad de control. ....	- 36 -
2.3.1.	Selección de Display.....	- 38 -
2.3.2	Teclado matricial 4x4. ....	- 40 -
2.3.3.	Reloj en tiempo real RTC .....	- 41 -
2.4.	Sistema de comunicación.....	- 42 -
2.4.1.	Selección del dispositivo para comunicación inalámbrica.....	- 42 -
2.4.2.	Comunicación controlador -- ESP01 y ESP01—PC O SMARTPHONE.-	44 -
2.4.3.	Algoritmo de las interfaces para PC y Smartphone. ....	- 47 -
2.4.4.	Diseño de la interfaz para Smartphone.....	- 48 -
2.4.5.	Diseño de la interfaz para equipos de computo.....	- 51 -
2.5.	Algoritmo del firmware para el microcontrolador. ....	- 53 -
2.6.	Fuente de Alimentación.....	- 56 -
2.7.	Sistema de respaldo energético. ....	- 57 -
2.7.1.	Inversor DC-AC de 12VDC a 120Vac.....	- 57 -
2.7.2.	Diseño del inversor de corriente DC-AC de onda cuadrada. ....	- 57 -
2.8	Implementación y evaluación del proyecto. ....	- 68 -
2.8.1.	Ensamble del prototipo. ....	- 68 -
2.8.2.	Evaluación del funcionamiento del prototipo. ....	- 71 -
2.9.	Resultados obtenidos.....	- 72 -
2.10.	Costos del prototipo .....	- 73 -
<b>CAPITULO III: CONCLUSIONES Y RECOMENDACIONES.....</b>		<b>- 74 -</b>
3.1.	CONCLUSIONES.....	- 74 -
3.2	RECOMENDACIONES .....	- 75 -
IV.-	Bibliografía .....	- 77 -

## Lista de figuras

Figura 1. Diagrama eléctrico de un timbre .....	- 5 -
Figura 2. Diagrama de control ON/OFF lazo abierto.....	- 9 -
Figura 3. Estructura genérica de un microcontrolador. ....	- 11 -
Figura 4. Logo MIT App Inventor. ....	- 15 -
Figura 5. Logo Visual Basic 6.0. ....	- 16 -
Figura 6. Módulo bluetooth HC-05 y HC-06.....	- 18 -
Figura 7. Asignación de pines ESP-01. ....	- 24 -
Figura 8. Comunicación serial. ....	- 27 -
Figura 9. Diagrama básico de un inversor. ....	- 29 -
Figura 10. Formar de onda generada por inversores.....	- 30 -
Figura 11. Izquierda: Entrevista a la directora del centro. ....	- 33 -
Derecha: Centro escolar Mote Sináí.....	- 33 -
Figura 12. Diagrama de bloque del proyecto implementado.....	- 34 -
Figura 13. Display LCD 20x4.....	- 38 -
Figura 14. Módulo i2C .....	- 39 -
Figura 15. Teclado de membrana 4x4. ....	- 40 -
Figura 16. Módulo RTC DS-1307 .....	- 41 -
Figura 17. Módulo wifi ESP-01. ....	- 44 -
Figura 19. Algoritmo de las interfaces para PC y Smartphone.....	- 47 -
Figura 20. Algoritmo de las funciones para las interfaces.....	- 47 -
Figura 21. Extensión TCP/IP "ClientSocketAI2Ext.aix" .....	- 50 -
Figura 22. Interfaz "Microsoft WinSock Control 6.0".....	- 52 -
Figura 23. Interfaz para configuración mediante PC.....	- 53 -
Figura 24. Algoritmo del firmware para el microcontrolador.....	- 54 -

Figura 25. Algoritmo del firmware para el microcontrolador.....	- 54 -
Figura 26. Adaptador AC/DC 5V.....	- 56 -
Figura 27. Bloque de la estructura interna del SG3524..	- 59 -
Figura 28. Configuración amplificador diferencial del AE interno del SG3524.....	- 60 -
Figura 29. Diseño del oscilador para el inversor de onda cuadrada con el SG3524. -	61 -
Figura 30. Circuito de protección contra picos de voltaje.....	- 61 -
Figura 31. Red de corriente de arranque para el SG3524..	- 62 -
Figura 32. Circuito de protección contra batería baja.....	- 63 -
Figura 33. Etapa de potencia para el inversor de onda cuadrada.....	- 64 -
Figura 34. Vista de un núcleo tipo E-I.....	- 65 -
Figura 35. Diseño del inversor de onda cuadrada con el integrado SG3524.....	- 66 -
Figura 36. Diagrama esquemático del prototipo, software proteus 8.9.....	- 68 -
Figura 37. Diseño del prototipo en su PCB, software proteus 8.9.....	- 68 -
Figura 38. Diseño del prototipo en 3D, software proteus 8.9. ....	- 69 -
Figura 39. Caja de estanque de paredes lisas PVC.....	- 69 -
Figura 40. Controlador para timbre escolar implementado. ....	- 72 -

## **Lista de tablas**

Tabla 1. Pines para establecer comunicación.....	- 33 -
Tabla 2. Selección de pines.....	- 36 -
Tabla 3. Comparación de costos de dispositivos para comunicación inalámbrica.....	- 40 -
Tabla 4. Niveles de voltaje y corriente para el funcionamiento del sistema.....	- 51 -
Tabla 5. Costos del prototipo.....	- 58 -

## **I.- Introducción.**

El centro educativo monte Sinaí es un colegio privado, el cual recibe ayuda monetaria año con año, de personas extranjeras para que este siga en pie, fundado en 1994 con una matrícula actual de 249 estudiantes, ubicado en el Bo Buenos Aires, portón del cementerio loma linda 4C al N 2C al O, cuenta con 8 docentes los cuales se encargan de impartir las clases en la modalidad primaria incluyendo la directora del centro.

Este proyecto consiste en independizar la activación de un timbre escolar del recurso humano, de manera que se le facilite a cada docente el cumplimiento de los horarios ya establecido para cada cambio de clase, evitando retrasos y creando un ambiente de puntualidad al activarse automáticamente.

Este dispositivo tiene un amplio campo de aplicaciones, no solo para instituciones educativas, sino también para la industria de manufactura, el sector agropecuario y en problemáticas domesticas por ejemplo el encendido de las luminarias de forma automática debido a que el método de activación por fotorresistencias tiene un corto tiempo de vida útil.

El documento tiene a disposición información sobre los dispositivos que se puedan utilizar en el desarrollo y construcción del prototipo, también destaca algunos proyectos que se han realizado en diversos países en los que se proponen diseños similares al de este trabajo.

Se pretende realizar a partir de tres etapas, la primera está basada en la recopilación de información para el diseño del prototipo y entrevista al personal docente para definir los alcances de dicho diseño, la segunda establece cuáles serán los alcances del proyecto, diseño de cada etapa que conforme el prototipo y la compra de los dispositivos establecidos para realizar las pruebas necesarias, como tercera se unificará el sistema de control para posterior realizar las pruebas correspondientes para verificar el cumplimiento de los alcances propuestos en la etapa anterior.

## **II. Justificación.**

Estos dispositivos tienen un amplio campo de aplicaciones en donde el factor tiempo sea de gran importancia. En México se comercializan para aplicaciones en las industrias textiles, para el anuncio de inicio y culminación de la jornada laboral; en el sector agrícola para el control de activación de los sistemas de riego y alimentación de los animales de corral, etc. Y por último para controlar el encendido de las luminarias de manera más precisa y eficaz ya que los sistemas convencionales basados en fotorresistencias presentan averías con mucha frecuencia debido a que dicho dispositivo es expuesto a las condiciones medio ambientales y con el tiempo estos se dañan convirtiéndolo en un método muy ineficaz.

Lo que se pretende es dar alternativas a muchas instituciones tanto industriales como educativas en donde no se cuenta con personal pendiente de los cambios de clases o finalización de la jornada laboral y de su debido anuncio por medio de timbres o campanas eléctricas, haciendo de este un sistema ineficaz y en consecuencia provoca la pérdida de tiempo valioso entre cada actividad. Proponemos un modelo de automatización para dicho timbre haciendo de este un sistema autónomo e independiente de la acción manual de un operario.

Dicha propuesta se cree de gran importancia pues este viene a beneficiar a la presentación del centro y directamente al personal encargado de la manipulación de dicho timbre, ya que puede aprovechar este tiempo para realizar otras tareas de mayor importancia y también se crearía un ambiente de puntualidad entre el personal de la institución.

En este caso sería de mucha utilidad ya que la situación actual del centro acarrea ciertos problemas debido a que la única persona encargada de realizar los cambios de clase de forma manual, es la misma directora del centro, cabe destacar que ella también imparte clases, causando retrasos en el horario ya establecido; por esta razón puede perder la noción del tiempo

por el hecho de estar impartiendo clases o atendiendo algún problema interno del centro, de tal manera que los tiempos estipulados carecen de precisión.

Toda la información que se recopile o genere se deja a disposición de la institución donde se implemente y de la Universidad Nacional de Ingeniería, el desarrollo e implementación de esta propuesta llenará de gran satisfacción a los autores porque estarían contribuyendo a la solución de un problema de carácter social.

### **III.- Objetivos.**

#### **1. General.**

- Diseñar e implementar un prototipo de timbre electrónico programable para el correcto control de los horarios establecidos para la jornada académica del centro escolar Monte Sinaí.

#### **2. Específicos.**

- Establecer las especificaciones técnicas y de diseño óptimas para un sistema de automatización para el timbre escolar.
- Desarrollar un firmware usando el lenguaje de programación adecuado para el sistema de control que registre las horas de cambio de clases y que realice la activación del timbre.
- Implementar dispositivos, tales como PC, Móvil, teclado y LCD, para visualizar y configurar local y remotamente los parámetros que administrará el sistema de control.
- Diseñar una interfaz gráfica para establecer comunicación entre un ordenador y el controlador para acceder a las funciones de activación del timbre y actualización de alarmas, fecha y hora.
- Implementar un sistema de respaldo eléctrico haciendo uso de otras fuentes de energía en caso de fallar el suministro en la red eléctrica comercial.
- Realizar pruebas para verificar el buen funcionamiento del sistema de control y evaluar si cumple con los requerimientos para su implementación en el centro escolar.
- Elaborar un manual de usuario que describa las funciones integradas en el sistema con el fin de facilitar al personal docente la manipulación y configuración de dicho dispositivo.

## CAPÍTULO 1: MARCO TEÓRICO.

### 1.1. Timbre eléctrico.

Un timbre eléctrico es un dispositivo capaz de producir una señal sonora al pulsar un interruptor. Su funcionamiento es gracias a fenómenos relacionados al electromagnetismo. (Araneda & Chandia, 2017)

Consiste en un circuito eléctrico compuesto por un generador, un interruptor y un electroimán. La armadura del electroimán está unida a una pieza metálica llamada martillo, que puede golpear una campana pequeña. (Araneda & Chandia, 2017)

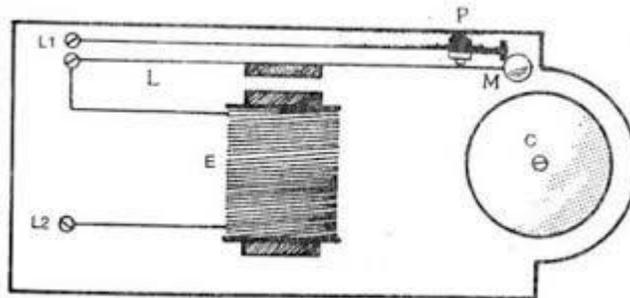


Figura 1. Diagrama eléctrico de un timbre. Fuente: <https://www.comohacer.eu/como-hacer-un-timbre-electrico/>

Al cerrar el pulsador, circula corriente a través del embobinado del electroimán creando un campo magnético el cual atrae la armadura. El martillo, soldado a la armadura, golpea la campana produciendo el sonido. Al abrir el pulsador dicha corriente cesa y el campo magnético desaparece, y un muelle devuelve la armadura a su posición original para interrumpir el sonido.

Existen algunos timbres que no tienen campana, ya que les basta únicamente la vibración de los contactos transmitida a la caja del timbre. En ocasiones se les nombra zumbadores por el sonido que producen; normalmente se les utilizan en oficinas, escuelas e institutos para avisar que

es la hora de cambiar o finalizar ciertas actividades o si hay algún evento natural (Terremotos o sismos) o producidos en el caso de un incendio. Joseph Henry, científico estadounidense y primer secretario de la institución Smithsonian, fue el inventor del timbre eléctrico en 1831.

La automatización de estos aparatos eléctricos se ha venido desarrollando a lo largo de los años para cubrir ciertas necesidades de escuelas, fábricas, oficinas, terminales de transporte, etc y que requieren de un temporizador versátil y de bajo costo con funcionalidad avanzada. (Araneda & Chandia, 2017)

## **1.2. Sistemas Electrónicos Embebidos.**

Un sistema embebido (S.E), es un conjunto de electrónica, informática y hasta mecánica, diseñado para la realización de una función, aplicación o tarea específica. En algunos casos estos sistemas embebidos se encuentran dentro de un sistema de mayor escala y en la mayoría de estos sistemas los componentes se encuentran incluidos en la placa base del dispositivo. (Estrada, Barahona & Rodríguez, 2012)

Algo muy notable en estos sistemas son el precio y el consumo energético. Esto debido a que los sistemas embebidos se fabrican por unidad, decenas o por millones de unidades, logrando así reducir costos. Los sistemas embebidos suelen usar una unidad de procesos y una memoria pequeña para reducir los precios y consumo eléctrico.

Usualmente estos sistemas cuentan con una unidad de control basado en microprocesadores o microcontroladores y la tarea o función que realizan esta especificada por el firmware o programa de aplicación, alojado en la memoria del sistema. La velocidad dependerá de los elementos que se utilicen, ya que se pretende con estos sistemas simplificar la arquitectura haciendo posible que el dispositivo realice dicha tarea de forma práctica y rápida.

Un S.E en general debe poseer ciertas características indispensables con las funciones adecuadas para hacer posible su buen funcionamiento. Entre estas podemos encontrar:

- Integran hardware y software, es decir, el hardware es diseñado para un único software.
- Trabajan en tiempo real, en la mayoría de los casos dichas aplicaciones deben trabajar al tiempo, esto quiere decir que nuestro sistema debe responder instantáneamente a cualquier cambio en sus entradas (sensores, interruptores, etc.), realizando algún proceso o modificando su entorno.
- Bajo costo económico, debido a que son sistemas relativamente pequeños sus costos de elaboración suelen ser muy bajos, lo que garantiza un gasto económico accesible. Estos sistemas son muy eficientes en lo que comprende el consumo de potencial por su reducido tamaño.

En el mundo actual los sistemas embebidos están inmersos dentro de nuestra vida diaria. Sus aplicaciones abarcan muchos sistemas y aparatos que son utilizados por los seres humanos en el diario vivir, mencionaremos alguno de ellos a continuación:

Hogares: están presentes en electrodomésticos como lavadoras, televisores inteligentes, sistemas de audio, sistemas de climatización, teléfonos celulares, etc.

Oficinas: podemos encontrarlos en cámaras digitales, sistemas de alarma, sistemas de control de acceso, aires acondicionados, entre otros.

Transporte: en los automóviles modernos encontraremos una gran cantidad de S.E en su infraestructura. Para controlar muchas partes del vehículo se utilizan sistemas muy precisos como son los frenos ABS, el sistema de inyección de combustible, sistema de luces y señalización, aire

acondicionado, limpia parabrisas, y muchos más. (Estrada, Barahona & Rodríguez, 2012)

Y se pueden mencionar más aplicaciones de estos sistemas tanto en medicina, comunicaciones, robótica y domótica. (Estrada, Barahona & Rodríguez, 2012)

### **1.3. Estrategias de control.**

Un sistema de control es un arreglo de componentes cuyo objetivo es comandar o regular la respuesta de una parte del proceso, conocida como planta, sin que el operador intervenga en forma directa sobre sus elementos de salida. El operador manipula únicamente magnitudes de baja potencia denominadas consignas, mientras que el sistema de control a través de los accionamientos conectados en sus salidas, se encarga de producir los cambios energéticos en la planta. (Daneri, 2008)

Podemos utilizar dos tipos de sistemas de control, el primero es un sistema de control en lazo abierto que se puede definir como aquel sistema en el que la acción de control se efectuará independientemente del comportamiento de la o las salidas. El segundo son los sistemas en lazo cerrado en los que se puede considerar que las señales de salidas si tienen cierta influencia en las acciones de control. (Daneri, 2008)

Existen varias estrategias de control entre las cuales se puede implementar el control ON-OFF. Es simple y no hay parámetros que elegir. El control ON-OFF a menudo funciona al mantener la variable de proceso próxima al punto de consigna, pero generalmente resultará en un sistema donde las variables oscilan. Es común realizar algunas modificaciones bien introduciendo una histéresis o una zona muerta. (Åström & Hägglund, 2009)



Figura 2. Diagrama de control ON/OFF lazo abierto. Fuente: autor.

#### 1.4. Sistemas de Control Automático.

El control automático de procesos constituye un caso particular de la automatización, que se puede definir como: “la suspensión total o parcial de la intervención humana en la ejecución de tareas agrícolas, industriales, domesticas, administrativas o científicas”. La Real Academia de Ciencias Exactas Físicas y Naturales define la “automática” como: “el estudio de los métodos y procedimientos cuya finalidad es la sustitución del operador humano por un operador artificial en la generación de una tarea física o mental previamente programada”. (Ortega, S., 2015).

#### 1.5. Componentes fundamentales de los sistemas de control.

Un sistema de control es una interconexión de componentes que forman una configuración del sistema que proporcionará una respuesta deseada. (Dorf, Bishop, S.F)

Los sistemas de control constan de tres elementos básicos: Entradas, Salidas y unidad de control.

En sus entradas se toman muestras de los captadores (sensores o transductores) y algunas medidas de referencias. En sus salidas se conectan actuadores y la unidad de control es la encargada de procesar la información proveniente de los sensores y de su comparación con las referencias y así realizar las órdenes programadas para los actuadores.

## **1.6. Unidad de control y procesamiento.**

En los últimos años se ha producido un importante cambio en la concepción y estructura de los sistemas de control, habiéndose pasado de los sistemas analógicos a los digitales. La utilización de microprocesadores (o microcontroladores) y del ordenador en el control de procesos ha permitido, no solo la obtención de mejores características y mayores prestaciones en el control que las que se obtendrían con un sistema analógico, sino que además posibilita funciones de adquisición de datos y tratamiento de los datos del proceso (comprobaciones, tendencias, promedios, etc.). Los controladores digitales están muy por encima de los analógicos en cuanto a precisión desde el punto de vista de ruidos internos y efectos de deriva. (Ortega, S., 2015).

La elección de la unidad de control depende del tipo de aplicación, estando condicionada por la complejidad del sistema a controlar, la velocidad de respuesta del sistema, las condiciones ambientales, presupuesto de ejecución, etc.

La ingeniería de control se ha desarrollado a través del tiempo. En el pasado los humanos eran el método para controlar los sistemas. En tiempos recientes a la electricidad los primeros controles eléctricos fueron los relevadores. Los relevadores permiten encender o apagar un circuito eléctrico sin necesidad de un interruptor manual. Fue inventado por Joseph Henry en 1835. (Ortega, S., 2015).

Existen diversos dispositivos que pueden funcionar como unidad de control de procesos entre los que se encuentran y que son muy utilizados tenemos: PIC's, Arduinos y PLC's.

### 1.6.1. Microcontrolador

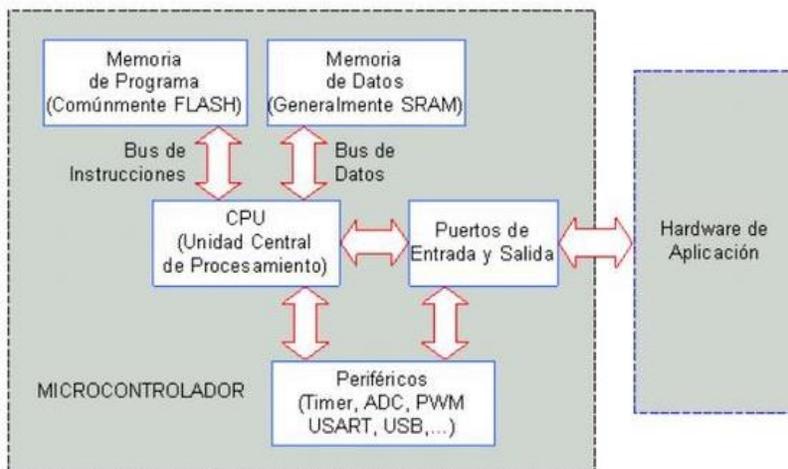


Figura 3. Estructura genérica de un microcontrolador. Fuente: <https://aprendiendoarduino.wordpress.com/2016/06/26/microcontroladores-2/>

Un microcontrolador es un circuito integrado o chip que incluye en su interior las tres unidades funcionales de una computadora: CPU, Memoria y Unidades de E/S, es decir, se trata de un computador completo en un solo circuito integrado.

Dando una explicación más sencilla, es una microcomputadora con la cual disponemos de todo, en pequeñas cantidades, para hacer uso en aplicaciones que requieran automatización, sin la necesidad de tener toda una computadora funcionando.

Son diseñados para disminuir el costo económico y el consumo de energía de un sistema en particular. Por eso el tamaño de la CPU, la cantidad de memoria y los periféricos incluidos dependerán de la aplicación. (Chi S., B. Elías, S.F.)

### **1.6.2. Reloj en Tiempo Real RTC)**

Un Reloj en tiempo real (en inglés, real-time clock, RTC) es un reloj de ordenador, incluido en un circuito integrado, que mantiene la hora actual. Aunque el término normalmente se refiere a dispositivos en ordenadores personales, servidores y sistemas embebidos, los RTC están presentes en la mayoría de los aparatos electrónicos que necesitan guardar el tiempo exacto. (Wikipedia, 2016)

El término se usa para evitar la confusión con los relojes hardware ordinario que sólo son señales que dirigen circuitos digitales, y no cuentan el tiempo en unidades humanas. Los RTC no deben ser confundidos con la computación en tiempo real (en inglés, real-time computing), que comparte su acrónimo de tres letras, pero que no se refiere directamente a la hora del día.

Aunque controlar el tiempo puede hacerse sin un RTC, usar uno tiene beneficios:

- Bajo consumo de energía (importante cuando está funcionando con una pila).
- Libera de trabajo al sistema principal para que pueda dedicarse a tareas más críticas.
- Algunas veces más preciso que otros métodos.

La mayoría de los RTC usan un oscilador de cristal, pero algunos usan la frecuencia de la fuente de alimentación. En muchos casos la frecuencia del oscilador es 32.768 kHz. Ésta es la misma frecuencia usada en los relojes de cuarzo, y por las mismas razones, que la frecuencia es exactamente  $2^{15}$  ciclos por segundo, que es un ratio muy práctico para usar con circuitos de contadores binarios simples. (Wikipedia, 2016)

### **1.6.3. Actuadores**

Se define como actuador a todo aquel dispositivo que transforma la energía eléctrica que recibe en otro tipo de energía, normalmente mecánica o térmica que ejecuta finalmente las condiciones de funcionamiento del sistema, que previamente ha establecido la Unidad de Control Electrónico (SEAT S.A., S.F.).

### **1.7. Interfaz Gráfica.**

Proporciona al operador las funciones de control y supervisión de la planta. El proceso que se supervisará se representa mediante sinópticos gráficos almacenados en el ordenador y generados desde un editor. Los sinópticos están formados por un fondo fijo y varias zonas activas que cambian dinámicamente de formas y colores, según los valores leídos en la planta o en respuesta a las acciones del operador. (Pérez L., E., 2014)

#### **1.7.1. Human Machine Interface (HMI).**

HMI significa "Human Machine Interface", es decir es un sistema que permite interactuar a la persona con la máquina. Tradicionalmente estos sistemas consistían en paneles compuestos por indicadores y comandos, tales como luces pilotos, indicadores digitales y analógicos, registradores, pulsadores, selectores y otros que se interconectaban con la maquina o proceso. En la actualidad, dado que las máquinas y procesos en general están implementadas con controladores y otros dispositivos electrónicos que dejan disponibles puertos de comunicación, es posible contar con sistemas HMI más poderosos y eficaces, además de permitir una conexión más sencilla y económica con el proceso o máquinas. (Cobo, R., S.F.)

## 1.7.2. Software para diseño de interfaces.

En la red hay infinidad de herramientas para construir nuestros prototipos, desde programas profesionales hasta pequeñas aplicaciones web que nos permiten tener un universo de opciones para diseñar interfaces para usuarios finales.

**Javier & J. (2017)** explican que una interfaz es como la ropa que nos ponemos al salir de casa, y que están compuesta de botones, gráficos, íconos y fondos que tienen una apariencia visual diferente en cada sistema operativo. Es también el medio que tiene el usuario de interactuar con el corazón funcional de la app.

### 1.7.2.1. MIT App Inventor.

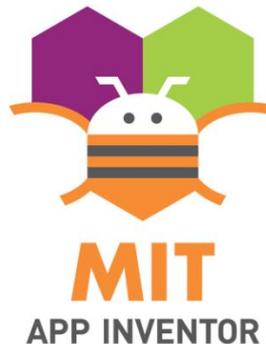
**Raúl (2017)** explica en su blog que App Inventor es un entorno de desarrollo para la elaboración de aplicaciones destinadas al sistema operativo Android. El usuario puede, de forma visual y a partir de un conjunto de herramientas básicas, ir enlazando una serie de bloques para crear la App. El sistema es gratuito y se puede utilizar fácilmente desde la web del MIT. Las App creadas están limitadas por su simplicidad, aunque permiten cubrir un gran número de necesidades básicas en un dispositivo móvil.

**Raúl** continúa narrando que dicha plataforma fue desarrollada inicialmente por el MIT y fue cogido por Google, para que cualquier persona con interés pueda crearse sus propias aplicaciones móviles.

En 2012 Google dio por finalizado el proyecto pero por suerte liberó el código fuente para que cualquiera pudiera usarlo y cedió su desarrollo al MIT para que el proyecto no quedase en el olvido y siguiese recibiendo mejoras.

En el sitio oficial de Google el **Ing. Mark Friedman (2010)** postea que App Inventor es una nueva herramienta de Google Labs que facilita a cualquier

persona (programadores y no programadores, profesionales y estudiantes) la creación de aplicaciones móviles para dispositivos con tecnología Android.



*Figura 4. Logo MIT App Inventor. Fuente: <http://appinventor.mit.edu/>*

#### **1.7.2.2. Visual Basic.**

Visual Basic es un lenguaje de programación orientado a objetos creado por la Microsoft. Este lenguaje incorpora todas las herramientas necesarias para la creación de cualquier aplicación para Windows. Con este lenguaje se puede crear desde una simple calculadora hasta un procesador de texto de la talla de Word, o una hoja de cálculo como Excel, o bien, cualquier aplicación que se le ocurra al programador.

Visual Basic es un lenguaje visual que se origina del lenguaje de programación Basic. La programación en Visual Basic se basa en un ambiente de desarrollo totalmente gráfico, que facilita la creación de interfaces gráficas, y en cierta medida, también la programación misma. Todos los programas que realicemos en Visual Basic serán por defecto, programas con todas las características de una aplicación típica de Windows.

El entorno integrado de desarrollo (EID) de Visual Basic, posee un sin número de herramientas que hacen de la programación de aplicaciones en Visual Basic mucho más fácil e interactiva. Este entorno incluye elementos tales como: barra de menús, barra de controles, barra de herramientas,

ventana de propiedades, ventana de proyectos, depurador, formularios, etc. Estos elementos los podemos apreciar en el momento que ejecutamos Visual Basic en nuestro ordenador.

Hoy en día, se encuentra en el mercado la versión .NET de Visual Basic. Con Visual Basic .NET, Visual Basic sigue siendo la herramienta más productiva para la creación de aplicaciones que se ejecutan en el sistema operativo Microsoft Windows. Esta versión, incluye todas las herramientas de programación rápida de aplicaciones que los programadores esperan de Microsoft, como la creación, con arrastrar y colocar, de aplicaciones para Windows que aprovechan totalmente las bases de datos y los servicios Web XML. También Visual Basic .NET, ofrece características de Internet móvil que permiten a los programadores crear una interfaz Web móvil única. Esta versión no solo es más potente, sino que también, ofrece un entorno gráfico cien por ciento mejorado, en relación a las versiones anteriores de Visual Basic. El editor de código y otros elementos del EID poseen nuevas características y mejoras que facilitan la lectura y escritura de los procedimientos escritos en la aplicación. (Rodríguez, M. & Rodríguez, A., 2008)

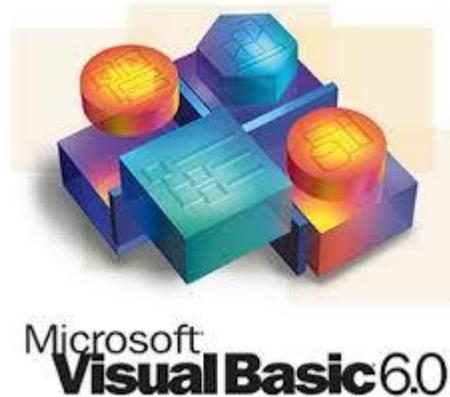


Figura 5. Logo Visual Basic 6.0. Fuente: [https://en.wikipedia.org/wiki/Visual\\_Basic](https://en.wikipedia.org/wiki/Visual_Basic)

## **1.8. Sistemas de Comunicación**

El objetivo de un sistema electrónico de comunicaciones, es transferir información de un lugar a otro. Por consiguiente, se puede decir que las comunicaciones electrónicas son la transmisión, recepción y procesamiento de información entre dos o más lugares, mediante circuitos electrónicos. Todas las formas de comunicación se deben convertir a energía electromagnética antes de ser propagadas a través de un sistema electrónico de comunicaciones.

Aunque los conceptos y principios fundamentales de las comunicaciones electrónicas han cambiado poco desde su introducción, los métodos y circuitos con que se realizan han sufrido grandes cambios. En los años recientes, los transistores y los circuitos integrados lineales han simplificado el diseño de los circuitos de comunicación electrónica, permitiendo así la miniaturización, mejor eficiencia y confiabilidad y costos generales menores. (Tomasi, 2003)

### **1.8.1. Dispositivos de Comunicación Inalámbrica**

Las redes inalámbricas se pueden formar con tan solo dos dispositivos, se pueden establecer redes 'ad hoc' comparables a las redes 'par a par' en una red cableada, donde ambos dispositivos funcionarán como clientes y servidores. De esta forma se tiene conectividad, pero la tasa de transferencia es baja y la seguridad es mínima, sumándose otro problema adicional cuando se utilizan dispositivos inalámbricos de diferentes fabricantes, porque suelen ser incompatibles entre sí. El problema de incompatibilidad se resuelve instalando un punto de acceso (AP, Access Point) para que actúe como concentrador en la red WLAN (Wireless Local Area Network).

Los AP (Access Point) tienen antenas para brindar conectividad inalámbrica a un área específica que recibe el nombre de celda. Esto es como en la telefonía celular donde las radio-bases son los puntos de acceso de la red

celular y el área que dan cobertura las celdas. Según la ganancia de las antenas instaladas en el AP y el lugar donde se instaló, el tamaño de la celda puede ser muy grande. Ejemplo: Las celdas de la telefonía celular son muy amplias porque su radio-base (AP) es estratégicamente ubicada y tienen antenas con una buena ganancia.

El estándar IEEE 802.11b define dos componentes; una estación inalámbrica, la cual puede ser una PC o una Laptop con una tarjeta de red inalámbrica (NIC - Network Interface Card), y un Punto de Acceso (AP - Access Point), el cual actúa como puente entre la estación inalámbrica y la red cableada. (Revista Varitek Universal, 2010).

#### 1.8.1.1. Módulo bluetooth HC-05 y HC-06

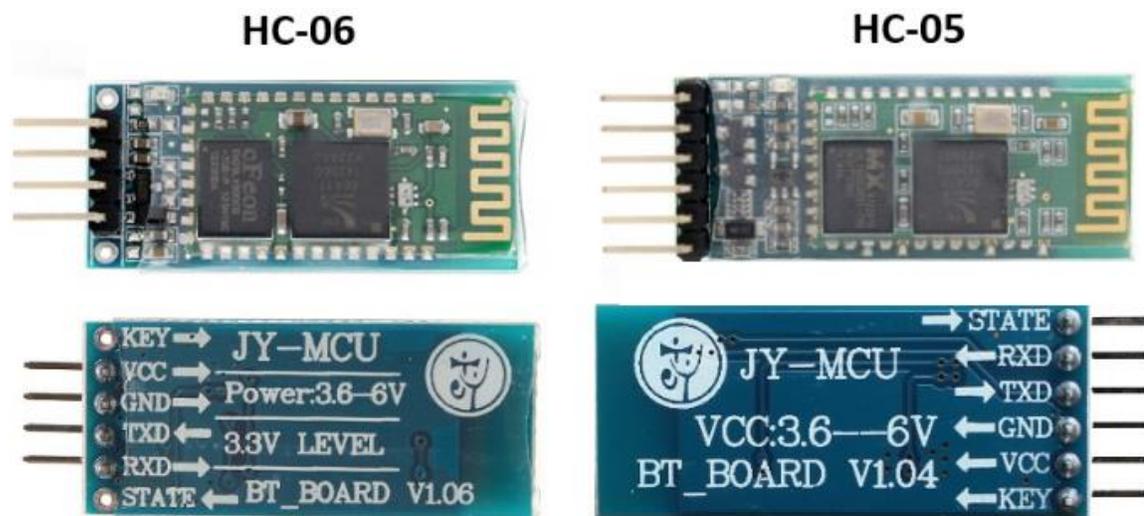


Figura 6. Módulo bluetooth HC-05 y HC-06. Fuente:  
<https://sites.google.com/site/bluetoothypic16f877amicrocode/>

El bluetooth es una tecnología para la transferencia de voz y datos con la que estamos más familiarizados pues está presente en la mayoría de dispositivos electrónicos con los que no relacionamos día con día, desde los

teléfonos móviles hasta reproductores con altavoces integrados para escuchar música.

Entre las principales características de esta tecnología, encontramos:

- Opera en la banda ISM libre entre los 2.402 y los 2,480GHz, por lo que no requiere de ninguna licencia.
- Su alcance depende de la potencia de emisión con la que disponga el equipo transmisor y si estamos en interior o al aire libre.
- Está orientada a la conexión punto a punto o red de malla de pocos nodos.
- Velocidad máxima de transmisión de datos 2Mbps.
- Dificultad para atravesar ciertos obstáculos como paredes.

Su uso está principalmente enfocado a ordenadores de sobremesa, portátiles, dispositivos móviles o de salida de audio, manos libres, dispositivos deportivos o de domótica e incluso en juguetes, tecnología médica o industria.

Los módulos de bluetooth HC-05 y HC-06 son Módulos muy populares para aplicaciones con microcontroladores PIC y Arduino. Se trata de dispositivos relativamente económicos y que habitualmente se venden en un formato que permite insertarlos en un protoboard y cablearlo directamente a cualquier microcontrolador. Estos módulos son bastantes sencillos en cuanto a recursos y configuración de manera que pueden usarse con el microcontrolador de nuestra preferencia. Ambos módulos son excelentes para proyectos escolares, de investigación e incluso para su integración en equipos o prototipos en el campo profesional. (GeekFactory, 2014)

Los módulos HC-05 y HC-06 son elementos que te permitirán comunicar tu microcontrolador por protocolos bluetooth con cualquier otro elemento que cuente con el mismo protocolo, desde un celular hasta un portátil o incluso otro módulo bluetooth conectado a otro microcontrolador. La principal diferencia entre estos dos módulos reside en que el HC-05 puede ser

configurado para ser usado como dispositivo maestro o esclavo, mientras que el HC-06 sólo puede ser usado como esclavo. (Arrieta, V., 2018)

Según **Rubén, (2014)** la única diferencia real entre ambos dispositivos reside en su firmware que ya viene cargado de fábrica, y que para diferenciarlos a la hora de adquirir uno le otro es que el HC-05 tiene dos pines extras etiquetados uno como “Key” o “EN” y el otro como “State”. También se pueden diferenciar en cómo se identifica cada módulo al conectarse a otro dispositivo; el hc-05 se identifica como “HC-05” y el HC-06 se identifica como “Linvor” o “HC-06”.

### **Especificaciones y características del HC-06**

- Voltaje de operación: 3.3 V a 5 V
- Consumo de corriente: 30 mA a 40 mA
- Configuración: Comandos AT
- Chip de radio: CSR BC417143
- Baudios por defecto: 9600, 8, 1, n.
- Versión o firmware: 3.0-20170609
- Modo: Esclavo
- Frecuencia: 2.4 GHz, banda ISM
- Potencia de emisión: 4 dBm, clase 2
- Modulación: GFSK
- Antena: Incorporada en PCB
- Alcance: 5 m a 10 m
- Sensibilidad:  $\leq -81$  dBm a 0.1% BER
- Velocidad: Asíncronica: 2 Mbps (max.)/160 kbps,
- Velocidad sincrónica: 1 Mbps/1 Mbps

- Seguridad: Autenticación y encriptación (Password por defecto: 1234)
- Perfiles: Puerto serial Bluetooth
- Dimensiones: 1.52 cm x 3.57 cm
- Temperatura de operación (Max): 75 °C
- Temperatura de operación (Min): -25 °C
- Interfaz: Bluetooth - Puerto serie UART TTL

### **Especificaciones y características del HC-05**

- Funciona como dispositivo maestro y esclavo bluetooth
- Configurable mediante comandos AT
- Frecuencia de operación: 2.4 GHz Banda ISM
- Modulación: GFSK (Gaussian Frequency Shift Keying)
- Potencia de transmisión: 4dBm, Class 2
- Sensibilidad: -84dBm @ 0.1% BER
- Seguridad: Autenticación y encriptación
- Distancia de hasta 10 metros en condiciones óptimas
- Voltaje de Operación: 3.6 VDC a 6 VDC
- Consumo Corriente: 30 mA a 50mA
- Chip: BC417143
- Versión o firmware: 3.0-20170609
- Interfaz: Serial TTL
- Antena: Integrada en el PCB

- Seguridad: Autenticación y encriptación (Contraseña por defecto: 0000 o 1234)

### **1.8.1.2. Módulo WIFI ESP-8266**

WIFI viene de Wireless Fidelity o bien Fidelidad Inalámbrica, se trata de una tecnología inalámbrica de transmisión de datos para internet basada en ondas de radio, concretamente la tecnología WIFI utiliza las frecuencias de 2.4 y 5 GHz.

Entre las principales características del WiFi hay que destacar:

- Opera en la banda de frecuencia de 2.4 GHz y 5GHz
- Su alcance depende de muchos factores pero es mucho mayor al del Bluetooth.
- Permite la conexión a Internet de diferentes dispositivos y puede usarse también para conectar dispositivos entre sí dentro de una red.
- Velocidad máxima de transmisión de datos 9,6Gbps para WiFi 6(Estándar 802.11ax).
- Dificultad para atravesar ciertos obstáculos y puede encontrar interferencias con otras ondas que emiten en las mismas frecuencias.
- Su uso es quizás el más extendido y de ahí que a día de hoy podamos encontrar un montón de dispositivos conectados.

El módulo WiFi ESP8266 es un auto-contenedor SOC, con pila integrada protocolo TCP/IP que puede dar acceso a cualquier microcontrolador a su red WiFi. El ESP8266 es capaz de acoger ya sea una aplicación o la descarga de todas las funciones de red Wi-Fi desde otro procesador de aplicaciones. Cada módulo ESP8266 viene pre-programado con un conjunto de firmware de comando AT, es decir, sólo tiene que conectar esto a su

dispositivo Arduino y obtener aproximadamente la misma cantidad de WiFi como ofrece WiFi Shield.

Este módulo tiene una potente capacidad suficiente a bordo de procesamiento y almacenamiento que le permite integrarse con los sensores y dispositivos específicos de la aplicación a través de sus GPIO con un desarrollo mínimo por adelantado y la carga mínima durante el tiempo de ejecución. Su alto grado de integración en el chip permite la circuitería externa mínima, incluido el módulo front-end, está diseñado para ocupar la misma área de PCB. El ESP8266 soporta APSD para aplicaciones VoIP y las interfaces de Co-Existence Bluetooth, contiene un auto-calibrado RF que permite que funcione en todas las condiciones, y no requiere de partes externas de RF, así lo señala en su descripción **SDR, SanDoRobotics, (S.F.)**

Según **Hernández, (2016)** debido a los sectores a los que va enfocado, dispositivos del IoT y móviles, el ESP8266 requiere de una gestión de energía eficaz. Dispone de una arquitectura de bajo consumo que trabaja en 3 modos.

- Active mode o modo activo: a pleno rendimiento.
- Sleep mode o modo dormido: solo el RTC (Real Time Clock) está activo para mantener la sincronización. Se queda en modo alerta de los posibles eventos que le hagan despertar. Mantiene en memoria los datos de conexión y así no hace falta volver a establecer la conexión con la WiFi. Consume entre 0,6 mA y 1 mA.
- Deep Sleep o modo en sueño profundo: el RTC está encendido pero no operativo. Debe pasar por el modo dormido antes de despertar. Hay que llevar especial cuidado con los datos ya que en este estado es como si estuviera apagado y todos los datos que no estén almacenados se pierden. Consume alrededor de 20  $\mu$ A.

Hay una fuente casi ilimitada de información disponible para el ESP8266. En la sección Documentos a continuación, encontrará muchos recursos para

ayudarlo en el uso de la ESP8266, incluso instrucciones sobre cómo transformar a este módulo en una solución IoT (Internet of Things).

## ESP-01.

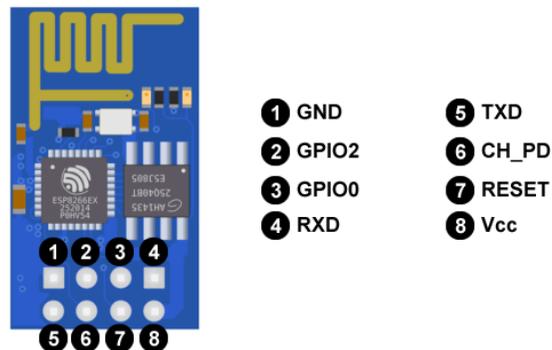


Figura 7. Asignación de pines ESP-01. Fuente: <https://programarfacil.com/podcast/como-configurar-esp01-wifi-esp8266/>

La creciente tendencia por conectar cualquier dispositivo a Internet ha hecho que surjan diversas plataformas para controlar los mismos y registrar sus acciones. Los microcontroladores ESP8266EX de Espressif, se trata de un chip integrado con conexión WiFi y compatible con el protocolo TCP/IP. El objetivo principal es dar acceso a cualquier microcontrolador a una red. La variante más común de los microcontroladores ESP8266 es el ESP01. De este modelo, existen dos versiones. Se diferencian en el color de la placa: azul y negro, respectivamente. La primera versión ofrece 512kB de memoria flash, mientras que la segunda ofrece 1024kB. Ambas versiones poseen dos pines GPIO (General Purpose Input and Output).

Se trata del módulo más popular aunque en muchas ocasiones, no es el más conveniente. El módulo ESP-01 tiene un precio muy reducido. En la actualidad no es el más utilizado, ha sido desbancado por el ESP-12. Tiene disponible dos pines GPIO digitales para controlar sensores y actuadores. (Hernández, L., 2016)

También se puede llegar a utilizar para uso general los pines Rx y Tx si no se utilizan para la comunicación a través del puerto serie. Se puede programar a través de un adaptador serie/USB o con el cableado adecuado, a través de Arduino.

#### **Características del ESP8266:**

- 802.11 b/g/n
- Wi-Fi Direct (P2P), soft-AP
- Pila de protocolos integrado TCP/IP
- Switch integrado TR, balun, LNA, amplificador de potencia y adaptadora de red.
- PLL integrados, reguladores, DCXO y unidades de administración de energía
- Potencia de salida en el modo 802.11b +19.5dBm
- Apagado de corriente de fuga <10uA
- Memoria flash de 1 MB
- CPU de 32 bits de baja potencia integrado, podría ser utilizado como procesador de aplicaciones
- SDIO 1.1 / 2.0, SPI, UART
- STBC, 1 x 1 MIMO, 2 x 1 MIMO
- A-MPDU & A-MSDU agregación y 0,4ms intervalo de guarda
- Despertador y transmisor de paquetes menores 2 ms
- Consumo en espera menor a 1.0mW (DTIM3)

#### **1.8.2. Módulo transmisor-receptor asíncrono universal (UART).**

UART es un módulo de hardware que traduce los datos de paralelo a serial para ser transmitidos, las UARTs son usadas comúnmente en conjunto con estándares de comunicación como EIA, RS-232, RS-422 o RS-485. La designación "universal" indica que el formato de los datos y las velocidades de transmisión pueden ser configurados.

UART es normalmente un circuito integrado individual usado para comunicaciones de un sistema de cómputo, son normalmente incluidas en microcontroladores. Un UART dual, o DUART, combina dos UART en un solo chip. Actualmente estos circuitos pueden comunicarse de manera sincrónica y asincrónica y son conocidos como USART. Las funciones principales de chip UART son: manejar las interrupciones de los dispositivos conectados al puerto serie y convertir los datos en formato paralelo, transmitidos al bus de sistema, a datos en formato serie, para que puedan ser transmitidos a través de los puertos y viceversa.

#### **1.8.2.1. Protocolo RS232.**

Una de las principales razones por las que el RS-232 ha sobrevivido durante más de cinco décadas es que se trata de una señal útil, pero de bajo nivel y rudimentaria, con unas directrices operativas bastante flexibles. En 1962, la única aplicación del estándar de comunicación RS-232 era la conexión de máquinas de escribir electromecánicas y sus ordenadores anfitriones centrales/módems, comúnmente conocidos como sistemas de «teletipo».

Cuando posteriormente se desarrollaron máquinas electrónicas más avanzadas, las adaptaciones patentadas llevaron a asignaciones de pines, conectores y niveles de tensión de señal no estándar. Por ejemplo, la especificación original pedía un conector DB-15, pero en los últimos 30 años la mayoría de los productos RS-232 han adoptado un conector DB-9 (técnicamente llamado DE-9M).

Los 'datos' que se envían a través de las líneas RS232 son simplemente impulsos de tensión positivos (+) y negativos (-) relativos a una referencia de tierra. Un grupo de pulsos +/- enviados por un dispositivo son cuidadosamente cronometrados por el dispositivo receptor y decodificados en lo que los ajustes de hardware consideran paquetes de bits de datos. En otras palabras, la norma RS-232 sólo define un marco eléctrico general

relativamente flexible para transmitir y recibir impulsos eléctricos. Lo que uno hace con todos estos pulsos depende en última instancia del hardware conectado. Cosas como la codificación de caracteres, el espaciado, los bits de inicio, los bits de parada, el orden de bits, la detección de errores, la velocidad de transmisión de bits, etc. no son responsabilidad del ámbito RS-232 y son establecidas por los circuitos conectados del usuario, normalmente en forma de un puerto de comunicación serial y sus chips y transistores asociados.

El trabajo del puerto COM es dar sentido a los pulsos en nombre del ordenador o periférico conectado. Para referencia, un sistema RS-232 debe transmitir de un dispositivo (enviado en su pin Tx), a un dispositivo receptor (recibiendo en su pin Rx) y viceversa. Debido a que el sistema tiene tres cables y dos canales de comunicación distintos, se considera un sistema «Full Duplex». Los datos pueden transmitirse al mismo tiempo que se reciben. (Jenkins, A., S.F.)

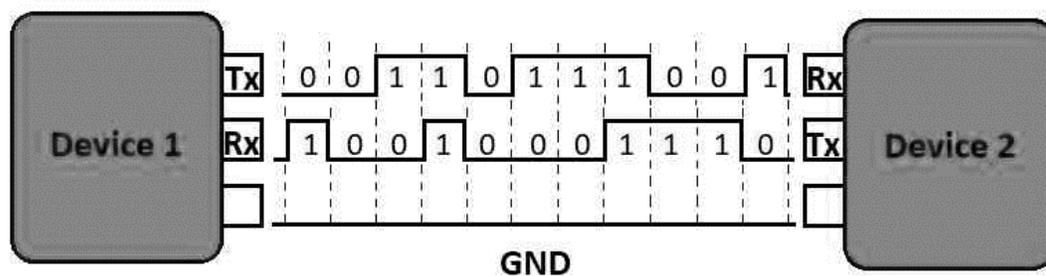


Figura 8. Comunicación serial. Fuente:  
<https://www.signalgen.com/electronica/comunicacion-serie/>

### 1.9. Sistemas de respaldo de energía.

Un sistema de respaldo eléctrico es un conjunto de dispositivos eléctricos y electrónicos que permite garantizar un suministro de energía sin interrupciones al momento de fallar el flujo de corriente eléctrica en el alambrado público. Estos poseen dos componentes sustanciales según su

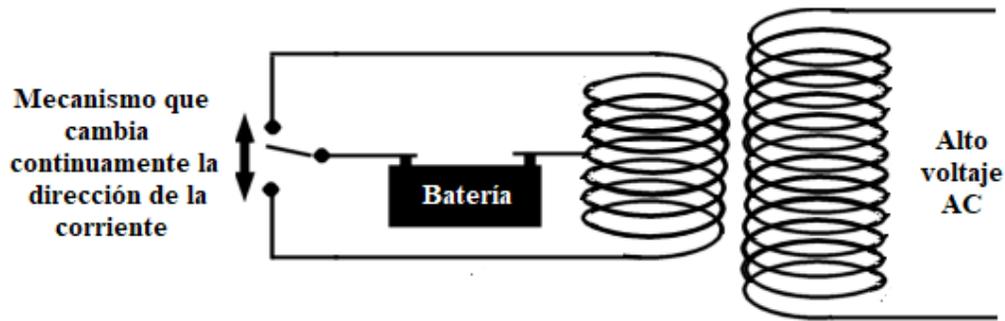
aplicación: un inversor (en caso de necesitar corriente alterna) y un banco de baterías.

Ofrecen muchos beneficios entre los que se podrían resaltar que funcionan como protectores ante apagones repentinos o variaciones súbitas de voltaje, no necesitan de combustibles ni produce gases que dañen el medio ambiente, no requieren de mantenimientos frecuentes y son de larga vida útil; sin embargo tienen varias limitantes pues no se deben usar para equipos de muy alto consumo energético como lavadoras, secadoras, etc. y son diseñados para un tiempo limitado de suministro según la carga que se conecte.

En muchas ocasiones se necesitan de una fuente de energía que sirva para recargar el banco de baterías, entre las más usadas para aplicaciones en hogares u oficinas se encuentra los sistemas de energía fotovoltaicas. (Almacén vulcano S.A.S, 2016)

### **1.9.1. Inversores de corriente.**

Un inversor es un dispositivo electrónico capaz de transformar una corriente continua (DC) en una corriente alterna (AC) a un voltaje y frecuencia determinados. Por ejemplo, si tenemos que alimentar un electrodoméstico que funciona en corriente alterna 120V (frecuencia 60Hz) pero no tenemos a disposición la corriente alterna de red, gracias al inversor, igualmente podemos alimentarlo, utilizando una fuente de corriente continua, como una batería de 12V (DC).



*Figura 9. Diagrama básico de un inversor. Fuente: [mptsolar.com/es/esquema-funcionamiento-inversor.html](http://mptsolar.com/es/esquema-funcionamiento-inversor.html)*

Por lo tanto, es indispensable usarlo para alimentar a través de corriente continua, los dispositivos eléctricos que funcionan en corriente alterna. Los inversores se utilizan mayormente en sistemas fotovoltaicos aislados (autónomos) para alimentar dispositivos eléctricos de casas aisladas, refugios de montaña, casas rodantes y barcos, y también en sistemas fotovoltaicos conectados a la red para introducir la corriente producida por la planta directamente en la red eléctrica de distribución.

Los inversores también se utilizan en muchas otras aplicaciones, desde grupos de continuidad hasta controladores de velocidad de motores eléctricos, desde conmutadores de energía a la iluminación.

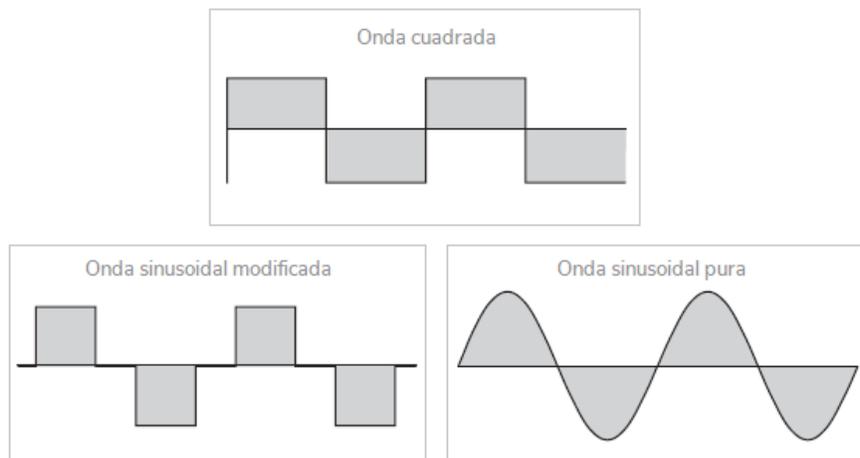
El término inversor también se puede utilizar para referirse a un grupo "rectificador-inversor", alimentado por corriente alterna y utilizado para variar el voltaje y la frecuencia de la corriente alterna en la salida en función de la corriente de entrada (por ejemplo, para la alimentación de particulares máquinas de operación).

Los inversores más comunes utilizados para alimentar cargas de corriente alterna son de tres tipos:

- Inversores de onda cuadrada.
- Inversores de onda sinusoidal modificada.

- Inversores de onda sinusoidal pura.

Los inversores de onda cuadrada son adecuados para el suministro de cargas puramente resistivas. Los inversores de onda sinusoidal modificada son adecuados para cargas resistivas y capacitivas, pero con cargas inductivas pueden producir ruido. Finalmente, los inversores de onda sinusoidal pura son aptos para todo tipo de cargas porque reproducen fielmente una onda sinusoidal igual a la de nuestra red eléctrica doméstica. (MPPTSOLAR, s.f.)



*Figura 10. Formas de onda generada por inversores. Fuente: [mpptsolar.com/es/esquema-funcionamiento-inversor.html](http://mpptsolar.com/es/esquema-funcionamiento-inversor.html)*

## **CAPITULO II: ANÁLISIS Y PRESENTACIÓN DE LOS RESULTADOS.**

En este capítulo presentaremos los resultados obtenidos del diseño, experimentación del prototipo y el análisis en base a la información obtenida mediante las técnicas e instrumentos de estudios que nos permitieron destacar las variables que han influido significativamente en la implementación del proyecto.

El proceso de investigación es de tipo aplicado ya que se está ofreciendo una posible solución a un problema bastante común en las instituciones públicas y privadas, lo que se pretende con el prototipo del timbre electrónico programable es dar una alternativa a los docentes que laboran en el centro, para que mejoren su sistema de control en cuanto a horarios de clases y puedan realizar cualquier tipo de actividad de manera que éste no genere ningún atraso ya que la implementación del mismo viene a favorecer al instituto por ser un sistema autónomo.

Para definir los alcances del proyecto se abordaron entrevistas al personal docente; partiendo de esa información se llevó a cabo el diseño funcional del dispositivo. Las etapas en la que se desarrolla son; etapa de análisis, etapa de diseño y obtención de resultados.

## 2.1. Etapa de análisis.

Inicialmente se realizó una entrevista<sup>1</sup> con la directora del centro **Lic. Eveling Sánchez**, con el fin de conocer y definir la problemática a la que se le quería dar una solución, y por la que se experimentaban atrasos en tareas cotidianas pero de suma importancia como en el caso de impartir las clases sin distracciones.

Dicho centro cuenta con poco personal de trabajo pues sus instalaciones son pequeñas y subsiste de donaciones internacionales a través de diferentes ministerios, debido a esto la directora también se encarga de impartir clases a los estudiantes del centro, por lo que atender la dirección y las horas de clases se vuelve un trabajo muy pesado y muchas veces pierde tiempo de los contenidos que imparte pues debe a la vez estar pendiente de los horarios para el anuncio de los respectivos cambios.

Siendo una tarea muy sencilla, se vuelve tediosa para la dirección pues no se cuenta con más personal para realizarla, por lo que se llegó a un acuerdo<sup>2</sup> sobre dar una solución sencilla y que se dedique a atender esta actividad.

Los requerimientos a cumplir para el diseño del sistema son los siguientes:

- Sistema autónomo de activación para el timbre.
- Las horas de activación sean programables y se puedan ejecutar de manera sencilla para el usuario.
- Que muestre al usuario las horas y minutos en que se dará la activación del timbre.
- Le permita al usuario tener la facilidad de programar las horas de forma remota mediante la PC o un Smartphone con SO Android.
- El prototipo no debe superar significativamente la cantidad de C\$ 3,000.00

---

<sup>1</sup> Anexo 1. Entrevista.

<sup>2</sup> Anexo 2. Acuerdo con el centro educativo.

Al momento de la visita al centro se constató que efectivamente el anuncio de los cambios de horarios se hace de forma manual activando una campana eléctrica mediante un interruptor de ON/OFF<sup>3</sup>.

La directora nos comentaba brevemente que en la actualidad solo anunciaba la hora de entrada y salida de los estudiantes para evitar distracciones a la hora de impartir sus clases, pero que aún no era efectivo debido a que en reiteradas ocasiones a los maestros, incluyéndola, se les va por alto el cambio de clase ocupando así minutos de la siguiente materia, afectando directamente el tiempo de la siguiente clase pues deben reducir el contenido.

Posteriormente se pidió autorización para realizar un chequeo más a detalle de la instalación actual del timbre que trabaja de forma manual. También se necesitó autorización para poder implementar el prototipo una vez terminado, la cual fue aprobada sin ningún inconveniente<sup>4</sup>.

Una vez definida la problemática se investigó de la existencia de equipos que cumplieran con esta tarea, localizando diferentes modelos que podrían ser utilizados<sup>5</sup>. El centro aún no cuenta con ningún sistema que les permita realizar este control de forma autónoma.



*Figura 11. Izquierda: Entrevista a la directora del centro.*

*Derecha: Centro escolar Mote Sinaí. Fuente: autor.*

---

<sup>3</sup>. Anexo 6. Instalación actual del timbre.

<sup>4</sup>. Anexo 2. Acuerdo con el Centro Educativo.

<sup>5</sup>. Anexo 3. Timbres electrónicos programables.

## 2.2. Diseño del sistema.

En este proceso se realiza una descripción de forma secuencial el cual permite visualizar en un diagrama las etapas del sistema que se debe cumplir obligatoriamente para lograr la correcta función en la activación del timbre cuando se detecten los cambios de hora. También se mostraran los elementos que integran este sistema que fueron definidos de acuerdo a los requerimientos establecidos.

Para el desempeño del sistema se utilizó un modelo de construcción de prototipo en el cual se realizaron pruebas y evaluaron los criterios de funcionalidad y accesibilidad para la obtención de los resultados. Estos fueron validados por el usuario final para valorar el cumplimiento de los requerimientos.

### 2.2.1. Esquema general del sistema de control.

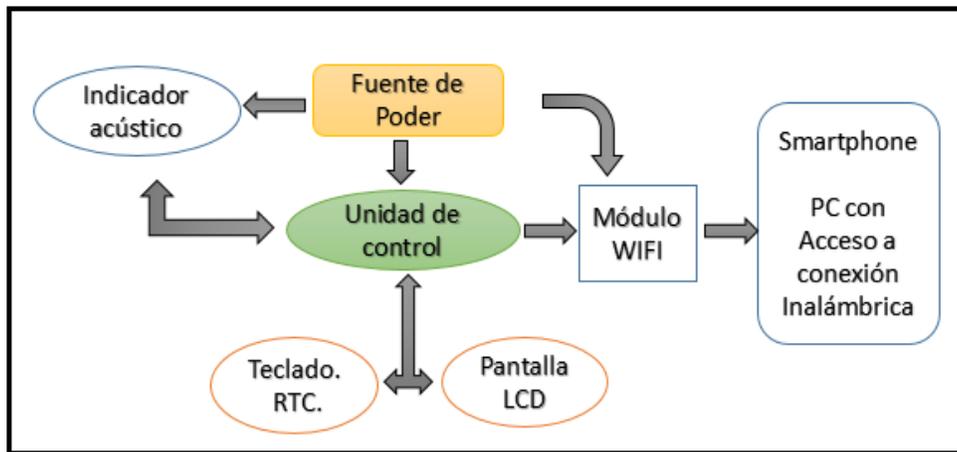


Figura 12. Diagrama de bloque del proyecto implementado. Fuente: Autor.

En esta etapa se muestran todos los aspectos relacionados con el diseño del sistema, empezando por un esquema general y luego describiendo cada bloque que lo conforman. La figura 12 muestra el esquema general del diseño para el correcto control de las horas clases.

En concordancia con la figura, el sistema contará con una unidad de control encargada de monitorear los demás periféricos para el correcto anuncio del cambio de las horas de clase. Funcionará de la siguiente manera: la unidad de control monitoreará los periféricos de entrada tales como teclado, RTC y la conexión inalámbrica a través del módulo WIFI en espera de recibir datos y luego procesarlos o almacenarlos en los registros de memoria de la EEPROM integrada en la unidad de control.

El reloj en tiempo real es el encargado de calcular el tiempo y enviarlo a la unidad de control para que esta se encargue de mostrar la fecha y hora así mismo activar el timbre en los tiempos programados por el usuario.

El usuario dispondrá de un teclado matricial de 4X4 caracteres y de una interfaz tanto para su ordenador como para su Smartphone que le permitirá registrar en tiempo real las horas de activación del timbre, así mismo podrá actualizar la hora y fecha del sistema en el momento que lo desee o crea conveniente. Todo esto será mostrado en un display LCD de 4 filas por 20 caracteres cada fila, para tener una manera visual de verificar cada dato registrado o procesado por la unidad de control, en él también la unidad de control se encargará de mostrar la fecha y hora actual.

### **2.2.2. Selección de los componentes principales del sistema.**

En este proceso de selección se hace mención de todos los criterios y especificaciones que se tomaron en cuenta para elegir cada dispositivo que integrará el prototipo, de manera que cada uno cumpla con las funciones establecidas para que su funcionamiento sea el más óptimo, ya que en el mercado existen diversidad de tecnologías y dispositivos que pueden ser útil en la implementación del proyecto. Sin embargo para su elección se tomará en cuenta que esten disponibles en el mercado local, la efectividad de su sistema, bajo costo, vida útil, entre otros.

### 2.3. Selección de la unidad de control.

El sistema consistía en la activación y desactivación del timbre eléctrico de manera automática para lo que se implementará un microcontrolador. El microcontrolador es el componente más importante de nuestro sistema embebido, ya que es el encargado de interpretar los datos obtenidos de cada periférico que integra el diseño.

Comercialmente existe en el mercado muchas opciones de microcontroladores con diferentes especificaciones, la selección de uno u otro dispositivo queda a criterio del diseñador dependiendo únicamente de los requerimientos que exija su diseño, cabe mencionar que no es bueno la selección de un dispositivo sobre dimensionado pues repercute significativamente a nivel económico.

El primer paso que realizaremos será conocer la cantidad de pines con el que debe contar el microcontrolador, lo siguiente es que cuente con algunos módulos de comunicación integrados como RS232 e I2C.

En la siguiente tabla se muestra la cantidad de pines necesarios para establecer una correcta comunicación entre los periféricos y el microcontrolador.

Tabla 1.

*Pines para establecer comunicación*

<b>Componente</b>	<b>Número de pines</b>	<b>Módulo</b>
LCD	7	GPIO
RTC	2	I2C
Relay	1	GPIO
Teclado 4x4	8	GPIO
ESP8266	2	RS232
<b>Total</b>	<b>20</b>	

En la tabla 1 podemos observar que requerimos un microcontrolador de, al menos 20 pines para poder establecer comunicación entre el microcontrolador y los periféricos.

En el mercado local encontramos microcontroladores de la compañía MICROCHIP, tanto microcontroladores PIC como los de la familia ATMEL. Para tal propósito se utilizará un microcontrolador ATMEL modelo ATMEGA328P integrado en la plataforma arduino en su versión nano, debido a tener ventajas muy importantes para nuestro diseño, ventajas que se mencionaran a continuación.

- No requiere de un programador externo, los PIC si requieren de un programador externo y aunque se pudiera grabar un Bootloader que permita grabarlo del PC necesitamos igualmente cargar el programa mediante un programador y esto para nuestro diseño implicaría en el engrosamiento de los costos por tal razón se descartó el uso de cualquier PIC.
- Su bajo costo en el mercado local. En el mercado local los microcontroladores PIC sus precios oscilan entre los C\$ 200 y C\$ 400, mientras los Atmel y en específico el Atmega328p en su encapsulado DIP-28 ronda los 170 Córdobas y en la plataforma arduino nano cuesta alrededor de los C\$285.

### 2.3.1. Selección de Display.



*Figura 13. Display LCD 20x4. Fuente: Autor.*

Basado en los requerimientos y las consideraciones de diseño para la interacción del sistema con el operador se dispone de un display LCD de 4 filas por 20 caracteres cada fila, para poder visualizar cada dato que se registre o sea procesado por la unidad de control.

En ella también se podrá observar la fecha y hora actual que la unidad de control se encargará de enviar y de esta manera tener una mejor percepción del tiempo transcurrido y así mismo ir visualizando la hora de cada activación del timbre. En la figura 13 podemos observar las características físicas del display.

Se seleccionó esta pantalla porque su tamaño nos permite una mejor visión de los datos procesados, es compatible con Arduino o cualquier Microcontrolador y también con la placa I2C. Posee el controlador LCM2004A el cual permite una excelente funcionalidad.

Características de trabajo de la pantalla LCD

- Voltaje de funcionamiento: 5V
- Consumo de corriente del LCD: aproximadamente 2 mA
- Consumo de la luz de fondo (retroiluminación): 40 mA

## Comunicación I2C

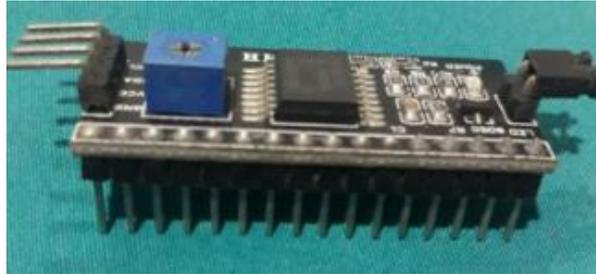


Figura 14. Módulo i2C Fuente: Autor

Cabe mencionar que para reducir un poco los requerimientos en cuanto al número de pines, para el LCD implementaremos un módulo I2C que integra el chip PCF8574 para el que necesitaremos únicamente los dos pines del módulo I2C integrado en la plataforma arduino nano.

Podemos indicar que en una comunicación digital existen distintos dispositivos o elementos. En el caso de I2C se diferencian dos elementos básicos, un MAESTRO y un ESCLAVO.

El **MAESTRO** se encarga de iniciar y parar la comunicación. El **ESCLAVO** I2C, generalmente suele ser un sensor. Este elemento suministra la información de interés al MAESTRO.

En total según la tabla 2 solo utilizaremos 13 pines de un total de 20 que posee la plataforma arduino.

Tabla 2.

*Selección de pines*

Componente	Número de pines	Módulo
LCD_I2C	2	I2C
Relay	1	GPIO
Teclado 4x4	8	GPIO
ESP8266	2	RS232
RTC	2	I2C
Total	13	

### 2.3.2 Teclado matricial 4x4.



Figura 15. Teclado de membrana 4x4. Fuente: Autor

El Teclado matricial de botones plásticos formado por 4 filas y 4 columnas para un total de 16 teclas permite agregar una entrada de usuario e ingresar datos fácilmente, esto nos permitirá ingresar correctamente las horas de clase establecidas. El teclado es de tipo membrana, por lo que entre sus ventajas se encuentra el poco espacio que requiere para ser instalado.

Un teclado matricial 4x4 nos permite controlar actuadores mediante alguna combinación de teclas utilizando arduino y así detectar por código cual es el botón que se está oprimiendo, estos teclados matriciales trabajan normalmente con 8 hilos de conexión pero se puede manejar con 16 pulsadores con ayuda de Arduino.

En el mercado local se encuentran diversidad de dispositivos que permiten ingresar datos que puedan ser interpretados mediante un microcontrolador, también varían en precios y tamaños.

Por su bajo costo y reducido tamaño se seleccionó el teclado de membrana para implementarlo en el prototipo ya que es adecuado para ingresar los datos requeridos para el controlador y que también facilitará al usuario manipularlo cuando se presente algún cambio en los tiempos establecidos en el centro de estudios.

### 2.3.3. Reloj en tiempo real RTC



Figura 16. Módulo RTC DS-1307 Fuente: Autor.

Un reloj en tiempo real (RTC) es un componente que se utiliza principalmente para permitir que un sistema digital mantenga un registro continuado del tiempo en una escala familiar a la percepción humana. Suelen operar a velocidades muy inferiores y consumen mucha menos energía que un reloj para fines generales.

Los RTC son más parecidos a los relojes y calendarios que usamos habitualmente ya que tiene registro de segundos, minutos, horas, día de la semana, fecha, mes y tienen una fuente de alimentación alternativa, por lo que pueden seguir midiendo el tiempo mientras la fuente de alimentación principal está apagada o no está disponible. Normalmente están formados por un resonador de cristal integrado con la electrónica necesaria para contabilizar de forma correcta el paso del tiempo

El reloj en tiempo real que se utilizará en la ejecución del proyecto será el Ds 1307 debido a su bajo costo y bajo consumo de energía, éste nos proporciona una alta estabilidad y precisión en nuestro portotipo, funciona en cualquiera de los formatos de hora 24 o 12 horas con indicador AM/PM, además tiene incorporado un circuito de sensor de tensión que detecta fallas de energía y cambia automáticamente al suministro de batería de respaldo.

## **2.4. Sistema de comunicación.**

En esta sección se describe el proceso de selección de los dispositivos posibles a utilizar para establecer una comunicación inalámbrica entre el ordenador o teléfono celular y el sistema de control.

### **2.4.1. Selección del dispositivo para comunicación inalámbrica.**

Existen diversos dispositivos que permiten establecer comunicación sin la intervención de cables, estos equipos disponen en su interior de tecnologías que utilizan diferentes protocolos de comunicación como es el caso de bluetooth o WiFi.

Para dicha tarea encontramos en el mercado local dispositivos que integran tecnologías bluetooth y WIFI, para establecer comunicación mediante bluetooth están los módulos HC-06 que funciona como esclavo y el HC-05 que funciona como dispositivo maestro o esclavo. Por su parte, para establecer comunicación mediante WIFI encontramos el módulo WIFI ESP-01; dispositivo de reducido tamaño en comparación con otros módulos WIFI e incluso con los HC-06 y HC-05. Para conocer y entender mejor sobre estos dispositivos referirse a la sección 1.8.

Con cualquiera de los dispositivos que utilicemos debemos configurarlos mediante comandos AT, y para esto debemos hacer uso de un terminal serial con protocolo de comunicación RS232, por tanto estableceremos solamente 2 criterios para seleccionar el dispositivo a utilizar que serán costo y distancia de propagación.

Tabla 3.

*Comparación de costos de dispositivos para comunicación inalámbrica.*

<b>Dispositivo</b>	<b>Costo [C\$]</b>	<b>Alcance de propagación [mts]</b>
<b>HC-06</b>	C\$ 385	8 mts aproximadamente
<b>HC-05</b>	C\$ 450	10 mts en las mejores condiciones
<b>ESP-01 (ESP8266)</b>	C\$ 200	30 mts en condiciones óptimas

Los dispositivos que se tomaron a consideración cumplen con las especificaciones requeridas por el sistema de comunicación y son ideales para su implementación, sin embargo en el caso del módulo bluetooth tenemos una desventaja ya que su nivel de alcance es bajo y para el prototipo se necesita de una extensa emisión debido a que el área del colegio donde se implementará el proyecto es de aproximadamente 150 mts<sup>2</sup>.

Se tomó la decisión de utilizar el módulo WiFi Esp8266 ESP-01 ya que en este caso es el más óptimo para realizar esta labor porque nos brinda un mayor alcance de comunicación, y su costo en comparación con los módulos bluetooth es bajo.

## 2.4.2. Comunicación controlador -- ESP01 y ESP01—PC O SMARTPHONE.



Figura 17. Módulo wifi ESP-01 Fuente: Autor.

En esta sección el Esp01 funcionará como un adaptador de protocolos para establecer comunicación entre el controlador y el equipo remoto para configuración (Smartphone o Laptop), es decir, el módulo se encargará de traducir los datos recibidos desde el controlador mediante protocolo RS232, correspondientes a los valores almacenados en los registros de alarma y los traducirá a un formato de red para comunicación TCP/IP y viceversa.

Como se mencionó anteriormente el módulo ESP01 es configurable mediante comandos AT, esto permite al microcontrolador configurar cuantas veces sea necesaria a dicho modulo pues ambos equipo están en constante comunicación a través del puerto serial del  $\mu$ C.

Hay tres modos de operación WiFi en el módulo WiFi ESP8266:

- Modo 1: Station Mode (STA)
- Modo 2: Access Point (AP)
- Modo 3: AP + Station

Para esta aplicación el módulo se configura en modo Access Point. En este modo el módulo actúa como punto de acceso y provee una red WIFI para otras estaciones como móviles o laptops (ver figura 18). Esto se configura enviando a través del puerto serie del  $\mu$ C el siguiente comando: AT + CWMODE=3.



Figura 18. Módulo Esp-01 operando como punto de acceso. Fuente: <https://esp8266-arduino-spanish.readthedocs.io/es/latest/esp8266wifi/readme.html>

Ahora para establecer comunicación entre el módulo y nuestro Smartphone o Laptops se debe configurar el ESP01 como un servidor, esto se logra solamente con el módulo configurado como punto de acceso, ya sea trabajando en el modo 2 o modo 3.

Para configurar el módulo como servidor una vez puesto en marcha el Modo 2 basta con enviar los siguientes comandos AT por el puerto serial del  $\mu$ C:

- Comando 1: AT+ CIPMUX=1; este comando indica que el módulo puede establecer múltiples conexiones (un total de 4).
- Comando 2: AT+ CIPSERVER= 1,1000; le indica al ESP8266 que active el servidor y utilice el puerto 1000 para el envío y recepción de datos por medio del protocolo TCP/IP.

Una vez realizada la configuración, se establece una conexión por sockets entre el módulo y nuestro Smartphone o Laptop.

Los sockets de Internet constituyen el mecanismo para la entrega de paquetes de datos provenientes de la tarjeta de red a los procesos o hilos apropiados. Un socket queda definido por un par de direcciones IP local y remota, un protocolo de transporte y un par de números de puerto local y remoto.

El término socket es también usado como el nombre de una interfaz de programación de aplicaciones (API) para la familia de protocolos de Internet TCP/IP, provista usualmente por el sistema operativo.

Cuando se implementan con el protocolo TCP, los sockets tienen las siguientes propiedades:

- Son orientados a la conexión.
- Se garantiza la transmisión de todos los octetos sin errores ni omisiones.
- Se garantiza que todo octeto llegará a su destino en el mismo orden en que se ha transmitido.

Para que dos procesos puedan comunicarse entre sí mediante sockets es necesario que ambos procesos sean capaces de intercambiarse cualquier secuencia de octetos, es decir, datos relevantes a su finalidad. Para ello son necesarios los dos recursos que originan el concepto de socket:

- Un par de direcciones del protocolo de red (dirección IP, si se utiliza el protocolo TCP/IP), que identifican la computadora de origen y la remota.
- Un par de números de puerto, que identifican a un programa dentro de cada computadora.

Los sockets permiten implementar una arquitectura cliente-servidor. La comunicación debe ser iniciada por uno de los procesos que se denomina programa "cliente". El segundo proceso espera a que otro inicie la comunicación, por este motivo se denomina programa "servidor".

### 2.4.3. Algoritmo de las interfaces para PC y Smartphone.

El algoritmo desarrollado para las interfaces se basó en una sencilla máquina de estados que se muestra en la figura 19 y figura 20, que luego se desarrolló e implementó haciendo uso de las herramientas respectiva para cada dispositivo (PC y Smartphone).

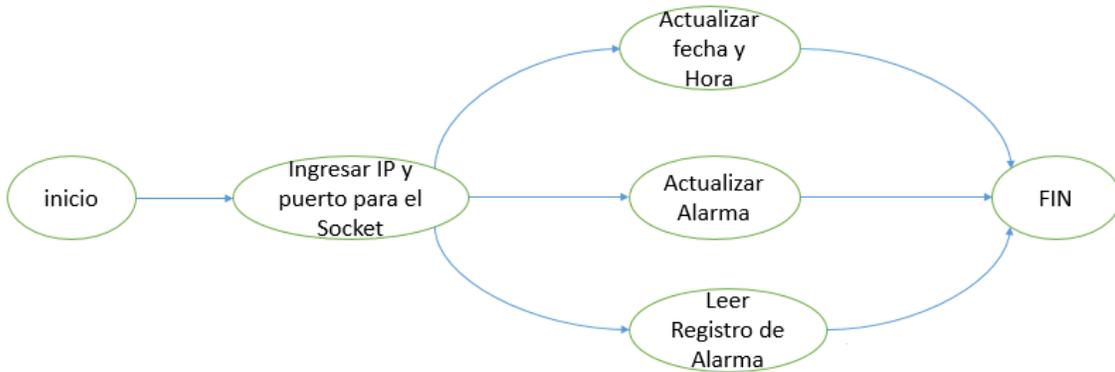


Figura 19. Algoritmo de las interfaces para PC y Smartphone. Fuente. Autor

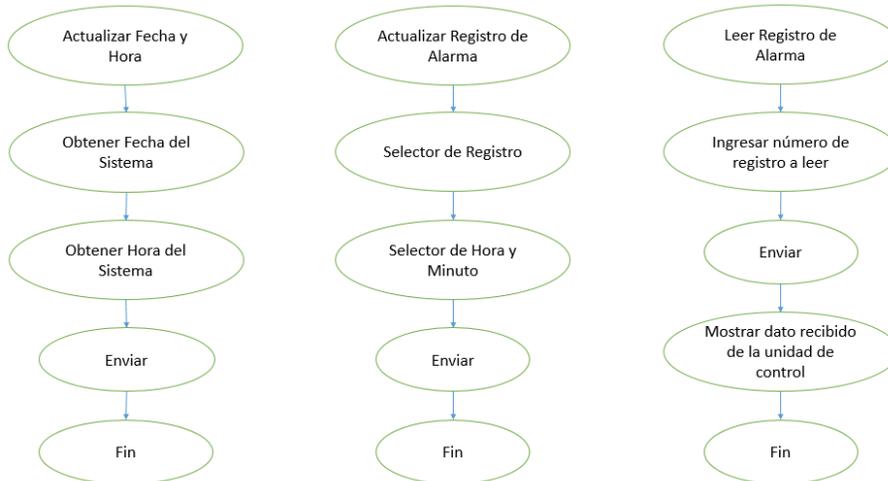


Figura 20. Algoritmo de las funciones para las interfaces. Fuente. Autor

- **Inicio:** Esta etapa se comprende como que el usuario a un no a ingresado a la APP o ejecutado el EXE para la PC.

- **IP y Puerto:** En este estado el usuario deberá ingresar la IP que direcciona al servidor y el puerto por el que compartirán los datos. Una vez verificada la conexión entre el ESP-01 y el otro dispositivo tendrá acceso a las demás funciones, por defecto la IP del servidor será la 192.168.4.1 y el puerto será el 1000.
- **Actualizar fecha y hora:** Aquí el usuario tiene la posibilidad de restablecer la fecha y la hora mediante el dispositivo remoto, el cual toma los valores del sistema y los envía por medio del protocolo TCP/IP al ESP-01 previamente etiquetados para que la unidad de control se encargue de interpretar dichos datos.
- **Actualizar alarma:** acá el usuario puede realizar cambios a las horas y minutos en la que la unidad de control realizará las activaciones del timbre.
- **Leer Alarma:** el usuario puede verificar el correcto guardado de los datos ingresados en el estado anterior.
- **Fin:** este es el último estado del algoritmo, acá se entiende que el usuario ha terminado de realizar las tareas de configuración y ha salido de la app.

Partiendo de lo antes expuesto procedimos a crear tanto la APP para SO Android como el ejecutable para PC con sistemas Windows.

#### **2.4.4. Diseño de la interfaz para Smartphone.**

Para esta etapa se desarrollará por medio de la plataforma MIT App Inventor 2, la elaboración de una aplicación para sistemas operativos Android, la cual permitirá al usuario el correcto control de las actualizaciones tanto de fecha y hora, así como de escritura y lectura de los registros de alarmas para la activación del timbre.

Existen en el mercado multitud de herramientas que facilitan el diseño de apps para móviles: Movincu-be, Goodbarber, 2Stacks, AppNet, etc. Todas ellas son soluciones comerciales pensadas para las empresas que exigen cierto desembolso económico. Facilitan el diseño con una apariencia muy

profesional mediante la elección de una plantilla y la configuración de sus parámetros. Esto limita el potencial para desarrollar el pensamiento computacional de los creadores. La mayoría de estas soluciones tienen limitaciones en la versión gratuita en cuanto al número de apps, marca de agua, conectividad, distribución a clientes. etc. La flexibilidad en el diseño de apps es mayor con MIT App Inventor aunque es cierto que la estética de las apps hay que trabajarla mucho más.

Como se menciona en el apartado 2.4.2 el encargado de recibir y transmitir datos será el módulo ESP-01, este recibirá del Smartphone una serie de datos codificados y etiquetados, el módulo se encargara de traducir estos datos al protocolo RS-232 y la unidad de control se encarga de responder las peticiones o acciones solicitadas por el usuario a través de la app.

También se menciona que para establecer comunicación entre el módulo ESP-01 y nuestro Smartphone haremos uso del protocolo TCP/IP, y mediante sockets estableceremos el envío y recepción de datos.

Para lograr dicho objetivo nos topamos con un inconveniente y es que en su versión anterior y la actual el MIT App inventor no cuenta con ningún bloque para la comunicación a través de sockets, sin embargo la nueva versión permite agregar extensiones que crean un universo aún más grande de posibles aplicaciones, solo basta con tener conocimientos básico de programación en JAVA y podremos desarrollar nuestras propias extensiones.

Para el caso de las comunicaciones por sockets a través del protocolo TCP/IP encontramos en el foro de MIT App Inventor la extensión **ClientSocketAI2Ext.aix** desarrollada por Jean-Rodolphe Letert.

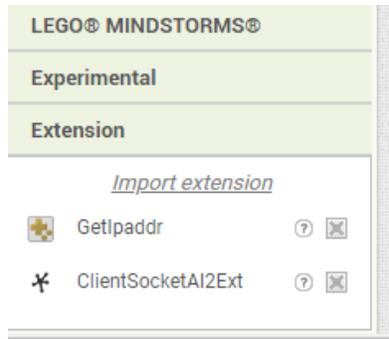


Figura 21. Extensión TCP/IP "ClientSocketAI2Ext.aix". Fuente: Autor

El diseño de la aplicación se lleva a cabo mediante tres procesos:

- Diseño de pantallas. Se crean las distintas ventanas o pantallas que contendrá la aplicación. En ellas se situaron los componentes que integran cada ventana: imágenes, botones, textos, etc. y se configuran sus propiedades.
- Edición de bloques. Se programa de forma visual e intuitiva el flujo de funcionamiento del programa utilizando bloques. Cada objeto dispone de unos métodos específicos que es posible invocar personalizando sus parámetros de llamada.
- Generación del app. Al finalizar las fases de diseño y programación, se genera el instalador APK de la aplicación.

Para el caso de la APP para sistemas Android, obtener la fecha y la hora se realizará mediante un botón respectivamente para cada opción y se integra un botón más para realizar la transmisión de los datos a la unidad de control.

Para los registros de alarma y lectura se cuentan con 9 registros en total, dicho registros podrán ser revisados y configurados a placer por el usuario.

#### **2.4.5. Diseño de la interfaz para equipos de computo.**

La interfaz gráfica es el medio con que el usuario puede comunicarse con el sistema de control, con ayuda de esta el usufructuario podrá acceder a cada una de las funciones integradas en la unidad de control. El modelo de la interfaz se basa en una única pantalla que cuenta con diferentes botones de opciones para que el usuario interactúe con el sistema. Al igual que en la interfaz para Smartphone esta será diseñada mediante tres proceso y contará con las mismas opciones para la configuración de la fecha y la hora, y leer o escribir sobre los registros de horas de activación del timbre.

Los procesos en los que se llevará acabo el diseño de la interfaz se describen a continuación:

- **Diseño de pantalla o formulario:** En este se colocan todos los controles de los que dispone Visual Basic para crear una aplicación. Cada uno de los elementos gráficos que forman parte de una aplicación de Windows es un tipo de control: botones, cajas de texto, cajas de diálogo, barras de desplazamiento, menús, gráficos, cajas de verificación, y muchos otros elementos son controles para Visual Basic.
- **Diseño de código:** En esta etapa se definen el comportamiento de cada procedimiento. Los procedimientos se encuentran dentro de los módulos, y contienen el conjunto de instrucciones que se ejecutan cuando el usuario realice algún evento sobre los objetos que colocamos en el formulario.
- **Generación del archivo exe:** Una vez terminados los procesos anteriores se genera el archivo exe.

La segunda interfaz diseñada para este prototipo cuenta con un campo para el ingreso de la dirección IP al servidor que será el módulo ESP-01 y el puerto por el cual se comunicarán ambos dispositivos (PC-ESP). Para lograr comunicación por sockets Visual Basic cuenta con un componente llamado **Microsoft WinSock Control 6.0** que permite realizar dicha comunicación.

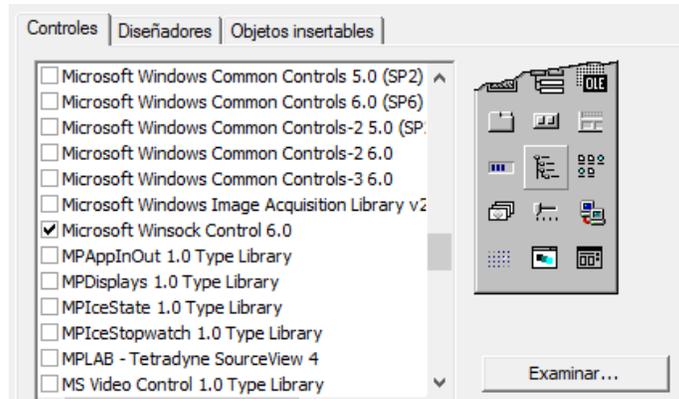


Figura 22. Interfaz "Microsoft WinSock Control 6.0". Fuente: Autor

Una vez que el equipo establece comunicación con el módulo, para indicarle al usuario que ya puede hacer uso de las demás funciones utilizamos un CheckBox el que mostrará un check cuando el PC se conecte con el servidor.

Cuando se verifique dicha conexión el usuario dispone de 4 botones que permitirán configurar las distintas funciones del sistema. El botón actualizar dota al usuario de la posibilidad de actualizar la hora y la fecha del sistema de control, el segundo botón **conf. Alarma** envía a la unidad de control el registro seleccionado con los componentes **OptionButton** etiquetados como Alarma 1 hasta Alarma 9, también a este dato se agrega la hora y minuto seleccionados en los **ComboBox** señalados como hora y fecha en la esquina inferior derecha del **Form**.

El siguiente botón fue etiquetado como Leer Alarmas el cual como su nombre indica permitirá al usuario acceder a los datos almacenado en los registros de alarma, para ello deberá antes seleccionar el registro que quiera leer desde los **OptionButton** y luego dar click en dicho botón. Por último se añadió un botón etiquetado con el nombre activar, este botón servirá como botón de pánico permitiendo así la activación manual del timbre en el momento requerido por el usuario, sin alterar los registros de alarma ya almacenados en la unidad de control.

En la figura 23 se muestra una imagen del diseño final de la interfaz y los botones con los que el usuario puede tener un correcto control de las funciones del sistema.



Figura 23. Interfaz para configuración mediante PC. Fuente. Autor

## 2.5. Algoritmo del firmware para el microcontrolador.

Tomando en cuenta los requerimientos del sistema de control y diseñado el sistema de comunicación ya tenemos todo para diseñar el firmware de nuestro sistema de control que comprende el proceso de lectura y la transferencia de datos hacia los periféricos que conforman nuestro proyecto.

Para el diseño del firmware vamos a considerar el algoritmo mostrado en la figura 24 y la figura 25, dicho algoritmo está diseñado para ser implementado en el microcontrolador ATmega328p integrado en la plataforma arduino nano.

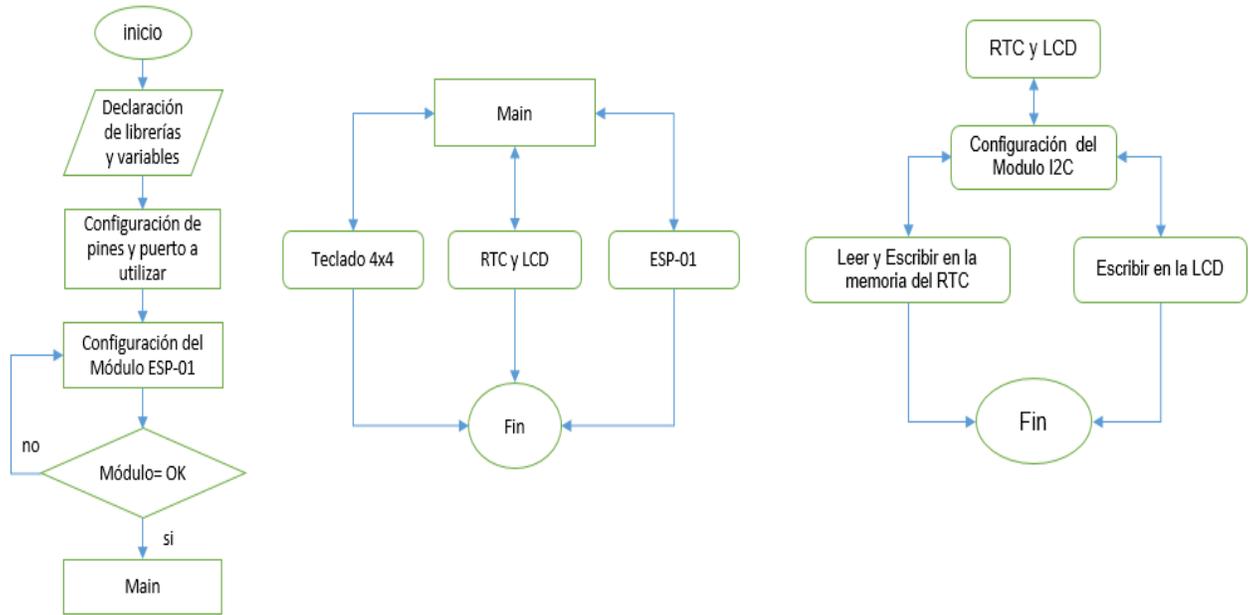


Figura 24. Algoritmo del firmware para el microcontrolador. Fuente. Autor

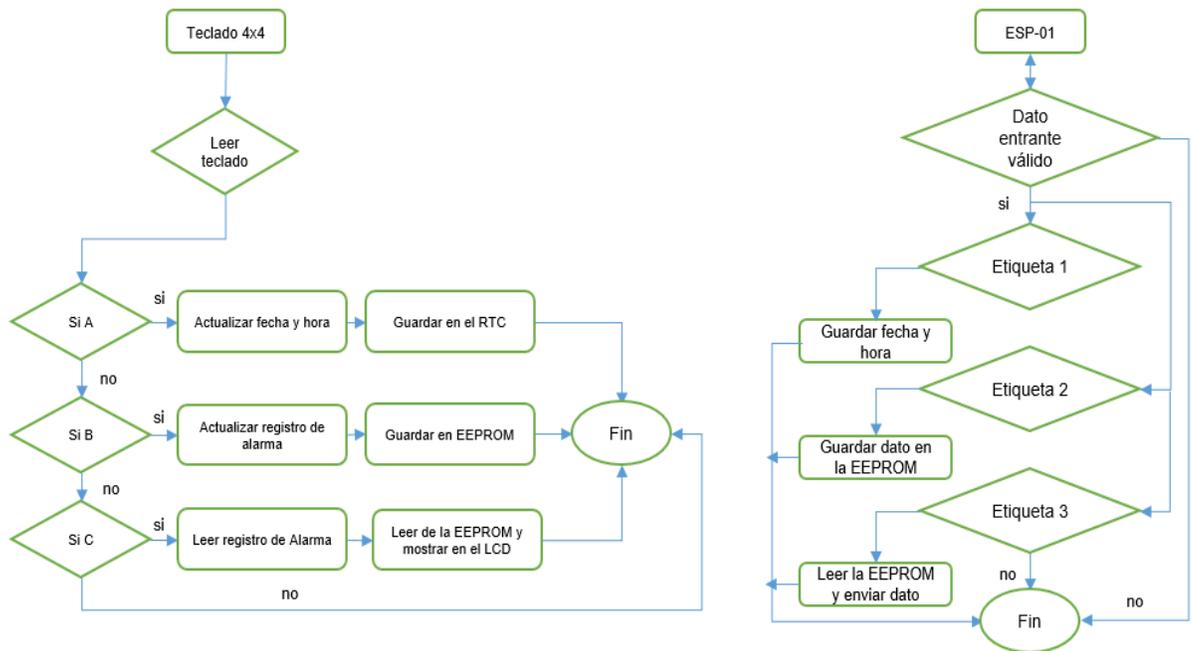


Figura 25. Algoritmo del firmware para el microcontrolador. Fuente. Autor

En primera instancia declaramos todas las librerías y variables que serán utilizadas a lo largo del firmware, luego configuramos los puertos y pines de entrada y salidas, una vez logrado esto esperamos un tiempo para que el

microcontrolador se encargue de configurar el módulo ESP-01, para finalmente llegar al programa principal.

El programa principal se divide en 3 etapas, las cuales debe nuestro microcontrolador atender de manera continua, la primer etapa estará diseñada para atender el teclado matricial de 4X4 caracteres. La segunda etapa será la carga de establecer correcta comunicación con el reloj en tiempo real y el display LCD, toda esta etapa está diseñada para trabajar mediante el protocolo I2C; y por último la tercera etapa se diseña entorno a la obtención y envío de datos mediante el módulo ESP-01.

En la primera etapa que comprende nuestro programa principal, se deberá configurar 8 pines de entrada-salidas de los cuales 4 serán entradas y 4 salidas. Luego se crea un mapa de caracteres con los que el microcontrolador podrá interpretar que carácter corresponde a la tecla presionada por el usuario y de esta manera permitir ingresar a cada una de las funciones que le permitirán ajustar los parámetros necesarios para la correcta ejecución de las tareas del controlador.

La segunda etapa solo se encargará de almacenar y mostrar datos registrados por el usuario por medio de cualquiera de los periféricos de entrada de datos. Todo esto lo logrará haciendo uso del protocolo I2C.

Por último, el microcontrolador deberá realizar envío y recepción de datos mediante el módulo ESP-01, esta comunicación entre el microcontrolador y el módulo será realizada mediante el protocolo RS-232, para esto se implementó una librería que asigna a cualquier pin del microcontrolador la recepción y transmisión de datos, permitiendo así dejar libre espacio en memoria del microcontrolador, debido a que esta librería no cuenta con muchas funciones con la que sí cuenta la librería que por defecto tiene integrada el software de desarrollo de Arduino. Cabe mencionar que aunque no cuenta con varias funciones, si tiene las necesarias para realizar una correcta comunicación con el ESP-01.

## 2.6. Fuente de Alimentación.

Una fuente de alimentación es el equipo o circuito encargado de generar una corriente estable para conectar equipos electrónicos. Se debe tener cuidado al escogerla, según las características de consumo, estabilidad y eficiencia requeridas.

En la tabla 4 se muestran los niveles de voltaje y corrientes necesarios para el correcto funcionamiento del sistema. De manera que debemos asegurar proporcionar un nivel de voltaje de 5 VDC y una corriente total de 276.5 mA.

Tabla 4. Niveles de voltaje y corriente para el funcionamiento del sistema

Elemento	Voltaje	Corriente
ArduinoNANO	5 VDC	25 mA
LCD-I2C	5 VDC	125 mA
Teclado	5 VDC	15 mA
RTC	5 VDC	1.5 mA
Relay	5 VDC	50 mA
ESP-01	3.3 VDC	60 mA
<b>Total</b>	--	276.5

Por su bajo costo y simplicidad de diseño, decidimos utilizar un adaptador estándar. Dicho adaptador suministra un nivel de voltaje aproximado de 5 VDC y puede manejar corrientes de hasta 1.5 Amperes; no obstante para el módulo WIFI ESP-01 se requiere un nivel de voltaje de 3.3 VDC para lo que adaptaremos un regulador de voltaje.



Figura 26. Adaptador AC/DC 5V. Fuente. Autor

## **2.7. Sistema de respaldo energético.**

Un sistema de respaldo eléctrico o mejor conocido como sistemas de alimentación ininterrumpidos (UPS), se conforma básicamente de un circuito inversor de corriente DC/AC y un cargador de baterías automático, con algunas protecciones adicionales.

El inversor de voltaje es un sistema que convierte la tensión de corriente continua, en un voltaje simétrico de corriente alterna de 120Vrms. El inversor se utiliza en infinidad de aplicaciones, que van desde pequeñas UPS para computadores, hasta aplicaciones industriales de alta potencia.

### **2.7.1. Inversor DC-AC de 12VDC a 120Vac.**

Debido a que el timbre eléctrico es un electroimán este necesita de un campo variable para funcionar correctamente, para que dicho timbre funcione necesitamos suministrarle 120 voltios en corriente alterna, por tanto nuestra fuente de energía en caso de fallar el suministro eléctrico no cumplirá con los requerimiento para poder accionar el timbre. Para eso debemos hacer uso de un inversor de corriente DC a AC que suministre el voltaje requerido.

### **2.7.2. Diseño del inversor de corriente DC-AC de onda cuadrada.**

El diseño del inversor se basó en el integrado SG3524, este integrado es un modulador PWM de propósito general y en su interior cuenta con toda la circuitería necesaria para ser usado como controlador en fuentes conmutadas e inversores de corriente.

Mediante un oscilador interno, el integrado genera un diente de sierra cuya frecuencia depende de dos elementos externos al IC, estos elementos son una resistencia RT y un capacitor CT. La resistencia va conectada entre el pin 6 y GND, el capacitor se conecta entre el pin 7 y GND.

La salida del oscilador interno se compara con una tensión externa  $V_c$ , a través del pin 9 y la salida del comparador pasa a un flip-flop y éste acciona un par de transistores de salida bipolares, de modo que los transistores trabajan con un desfase de 180 grados, es decir, un transistor está activo la mitad del periodo y el otro estará activo la segunda mitad del periodo.

Según el fabricante la frecuencia se puede definir por medio de la siguiente ecuación:

$$f_T = \frac{1.25}{(R_T)(C_T)}$$

También por especificaciones del fabricante tenemos que:

$$1.8K \leq R_T \leq 100K$$

$$0.001\mu F \leq C_T \leq 0.1\mu F$$

$$120Hz \leq f \leq 500KHz$$

Para nuestro diseño hemos definido una frecuencia de trabajo de 60 Hz igual a la de la red comercial, para esto usaremos un capacitor de 150 nF y como resultado de pruebas realizadas al circuito obtenemos un valor para  $R_T$  de aproximadamente 77 Kohms.

El IC cuenta con entradas a un comparador entre los pines 1 y 2. En función del valor de la entrada  $V_c$  (que puede variar entre 1V y 3.5V), en cada transistor de salida, tendremos una señal PWM cuya frecuencia será la mitad del oscilador.

En la siguiente imagen proporcionada en la hoja de datos del IC por el fabricante, podemos apreciar el comparador que integra internamente el SG3524. La señal diente de sierra oscila entre 1V y 3.5V, por tanto los voltajes de comparación a la salida del amplificador de error no se recomienda que superen estos valores.

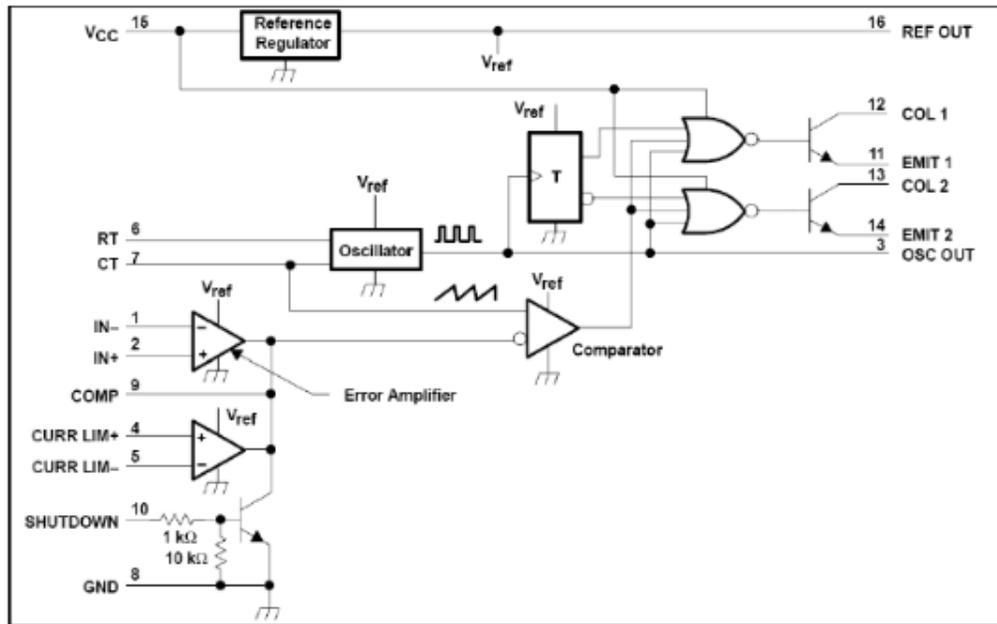


Figura 27. Bloque de la estructura interna del SG3524. Fuente: Texas Instruments.

De manera que se programó un voltaje de referencia en la entrada no inversora (PIN 2) del amplificador de error igual a 2.5V, tomando esta muestra del pin 16 del integrado que proporciona aproximadamente 5V estables. Para lograr los 2.5V se utilizarán dos resistencias del mismo valor, el cual definiremos en 10 Kohms.

Para la entrada inversora debemos proporcionar un nivel de voltaje cercano a los 2.5V configurados en el pin 2, dicho valor lo dejaremos en 2.2V. Para esto se hará un divisor de tensión a partir del voltaje de retroalimentación que se toma de la salida del transformador. El circuito equivalente se muestra en la figura 28.

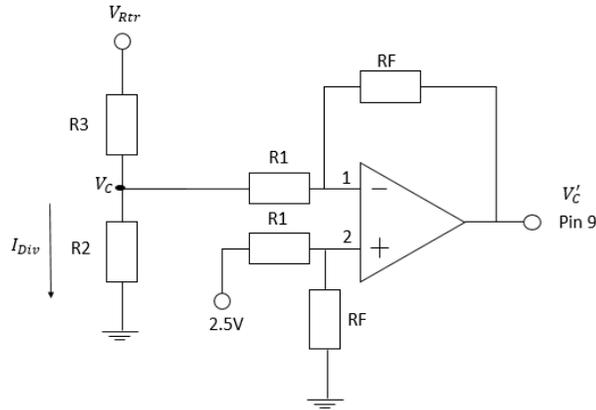


Figura 28. Configuración amplificador diferencial del AE interno del SG3524. Fuente: autor.

Para el voltaje de retroalimentación ( $V_{Rtr}$ ) se utilizó un transformador de baja potencia con un voltaje de secundario igual a 8.5 Vac, que rectificado y filtrado nos entrega 10.5 V<sub>DC</sub>.

Los valores del divisor de tensión y  $R_F$  se calculan a continuación:

$$R_1 = R_2 = 10K\Omega; V_{Rtr} = 10 V_{DC}$$

$$I_{Div} = \frac{V_C}{R_2} = \frac{2.2V}{10Kohms} = 0.22mA$$

$$R_3 = \frac{V_{Rtr} - V_C}{I_{Div}} = \frac{10.5V - 2.2V}{0.22mA} = 37,78K\Omega \cong 39K\Omega$$

$$V_{PIN\ 9} = (2,5V - V_C) \left( \frac{R_F}{R_1} \right) = 2V$$

$$R_F = \frac{(R_1)(V_{PIN\ 9})}{2,5V - V_C} = \frac{(10K\Omega)(2V)}{2,5V - 2,2V} = 66,66K\Omega \cong 68K\Omega$$

En la figura 29, aquí el integrado SG3524 cumple la función de oscilador para generar una onda cuadrada y mediante nuestra red de transistores controlamos el sentido de la corriente para que a la salida del transformador tengamos una onda alterna de semiciclo positivo y negativo.

## Voltaje retroalimentación

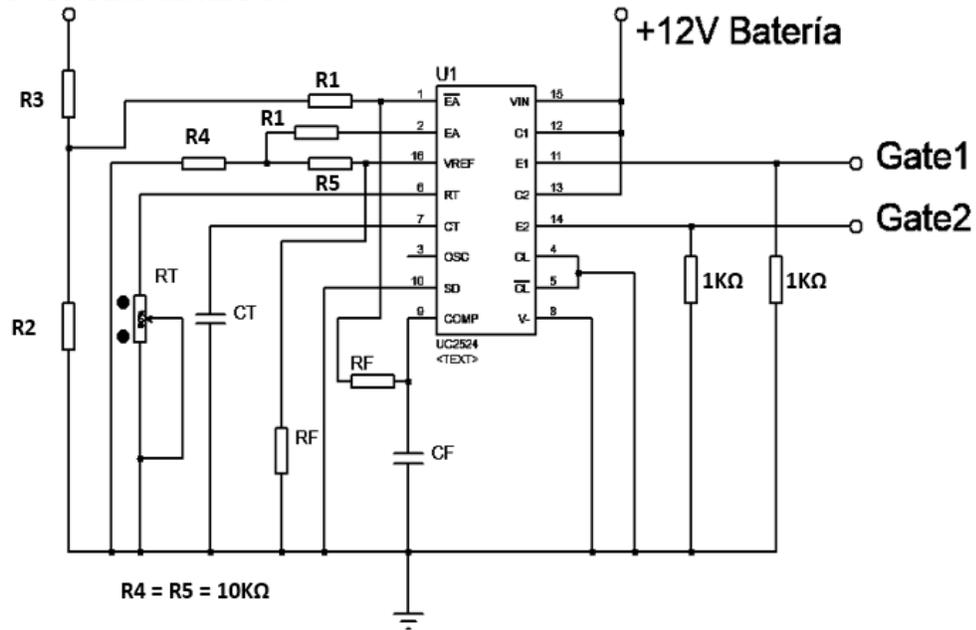


Figura 29. Diseño del oscilador para el inversor de onda cuadrada con el SG3524.

Fuente. Autor

Ahora para proteger a la batería y al mismo tiempo nuestros drivers es necesario crear una red que se encargue de reducir los picos creados al momento de hacer circular corriente de un sentido al otro sobre los devanados. Para ello implementamos una red basada en un rectificador de onda completa que alimenta a un regulador de tensión de 12Vdc, a su vez la salida de este regulador se conecta al pin de alimentación del integrado SG3524. En la figura 30 se muestra el diagrama del diseño de la red de amortiguamiento.

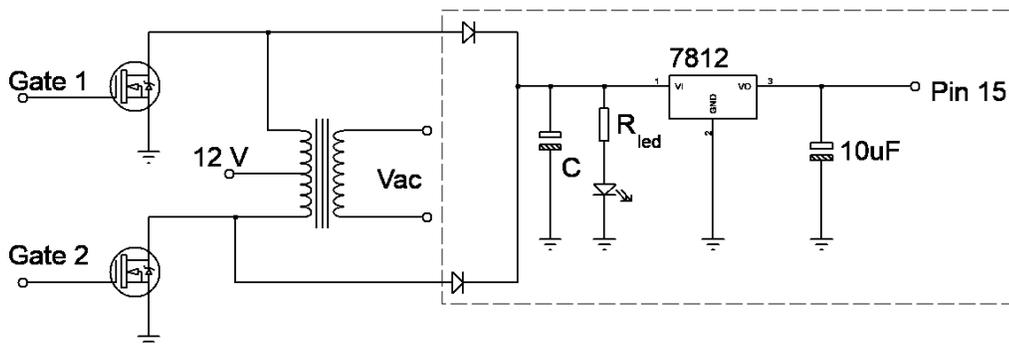


Figura 30. Circuito de protección contra picos de voltaje. Fuente: autor.

El valor del voltaje efectivo de los picos generados se aproxima a los 20 voltios, al ser una red de protección el capacitor de filtrado debe ser suficientemente grande para no dañar ningún circuito conectado después de él, el valor de C se seleccionó en 4700uF a 63 voltios de aislación.

La resistencia en serie con el led, funcionan como ayuda para amortiguar la carga del capacitor, igualmente  $R_{led}$  (figura 30) sirve como limitador de corriente para no dañar el led, de manera que debemos calcular el valor de esta resistencia.

$$R_{led} = \frac{V_{Cap} - V_{led}}{I_{led}} = \frac{20V - 2.5V}{10 mA} = 1750 K\Omega \cong 1.8 K\Omega$$

Ahora para bloquear corriente que pueda entregar la batería al integrado colocaremos una resistencia (figura 31) y un diodo en serie desde la terminal positiva de la batería al pin 15 del SG3524. R debe ser suficientemente pequeña (5.6 ohmios para nuestro diseño) de manera que la corriente que circule por ella permita encender al integrado y una vez esté en marcha nuestra red de amortiguamiento se encargará de suministrar potencia al integrado.

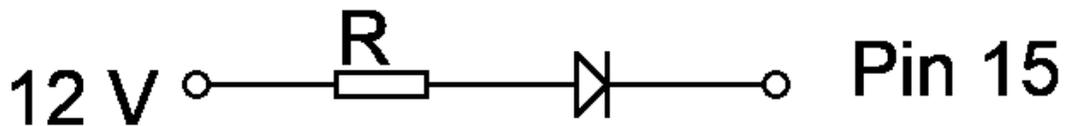


Figura 31. Red de corriente de arranque para el SG3524. Fuente: autor.

Ahora diseñamos el circuito de protección contra batería baja, una batería de 12 voltios se considera totalmente cargada cuando entre sus terminales hay una diferencia de potencia de al menos 12.5 voltios, y completamente descargada hasta los 10.5 voltios.

Por tanto, para nuestro diseño tomamos un nivel de voltaje de descarga igual a 10.8 voltios, en la figura 32 se muestra el circuito diseñado para protección contra batería descargada.

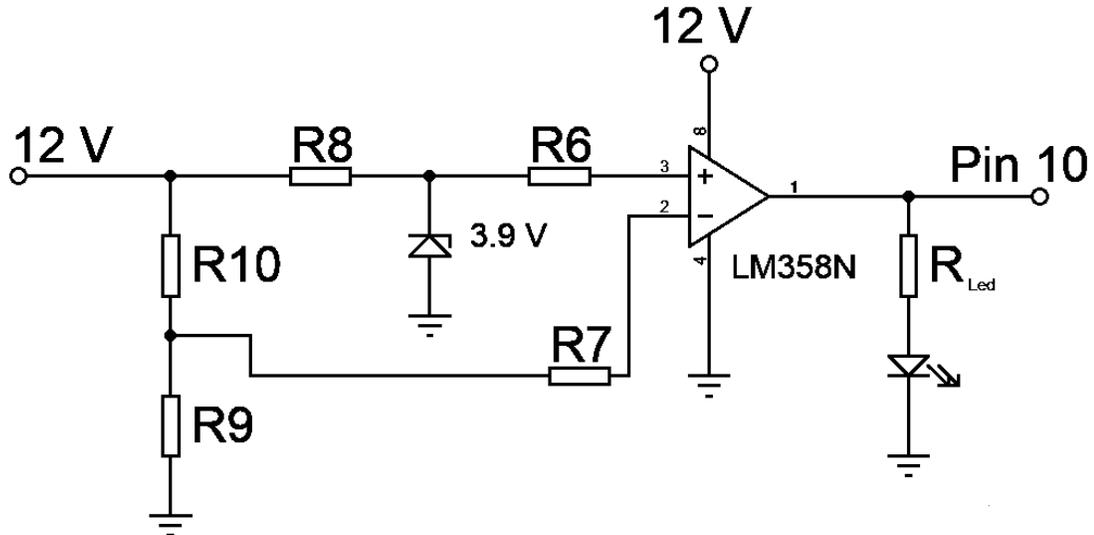


Figura 32. Circuito de protección contra batería baja. Fuente: autor.

Para este circuito R6 y R7 se seleccionaron con valor 10kΩ. El diodo zener establece un valor de referencia fijo para realizar la comparación con el nivel de voltaje que tenga la batería, dicho nivel de tensión será reducido mediante un divisor de voltaje resistivo en donde el voltaje que se compara, con el que fija el diodo zener, será la caída de voltaje en la resistencia R9; R9 se seleccionó con valor de 10KΩ.

$$i_{Div} = \frac{V_{Div}}{10k\Omega} = \frac{3.9V}{10k\Omega} = 0.39 \text{ mA}$$

$$R_{10} = \frac{V_{Bbaja} - V_{Div}}{i_{Div}} = \frac{10.8V - 3.9V}{0.39 \text{ mA}} = 17.7K\Omega \cong 18K\Omega$$

R8 sirve como limitador de corriente y protección para el diodo zener, para este caso el diodo zener es de un 1W, y la corriente la limitamos a 5mA, teniendo estos datos procedemos a calcular el valor de R8.

$$R_8 = \frac{V_{Bcarga} - V_Z}{i} = \frac{12V - 3.9V}{5mA} = 1620\Omega \cong 1.8K\Omega$$

Para la resistencia de protección del diodo led, sabemos que el voltaje de salida del comparador es aproximadamente  $\frac{3}{4} V_{cc}$ , lo que para 12 voltios serian 9 voltios. Por tanto el valor de  $R_{led}$  se calcula de la siguiente manera:

$$R_{LED} = \frac{V_{Comp} - V_{LED}}{i_{LED}} = \frac{9V - 2.4V}{10mA} = 660\Omega \cong 680\Omega$$

La etapa de potencia es la encargada de controlar y entregar una alta corriente para la bobina primaria del transformador. El transformador aunque es muy útil, no hace nada por nada. A medida que aumenta la tensión, la corriente disminuye, pero la potencia permanece igual. En palabras sencillas para obtener 1000 Watts en corriente alterna, necesitaremos suministrar 1000 Watts en corriente continua.

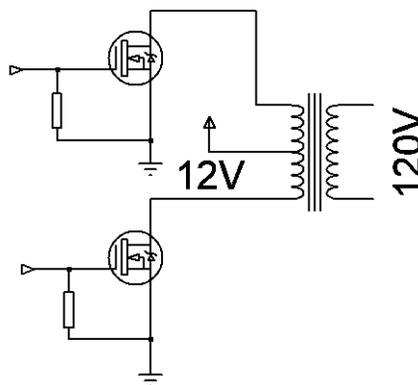


Figura 33. Etapa de potencia para el inversor de onda cuadrada. Fuente: autor

Los transistores y el transformador, determinan cuanta potencia puede suministrar el inversor. Los transistores a utilizar son los IRL540N que pueden manejar potencia de 100 Watts.

El diseño del transformador se realizó utilizando un núcleo de 2.8 cm por 5 cm, acorazado tipo EI (ver figura 34). El propósito de este transformador será de elevar el voltaje de 12 voltios a 120 voltios, por tal razón deberemos realizar los cálculos para el devanado secundario, que será de 12 + 12 voltios o de 24 voltios con TAP central.

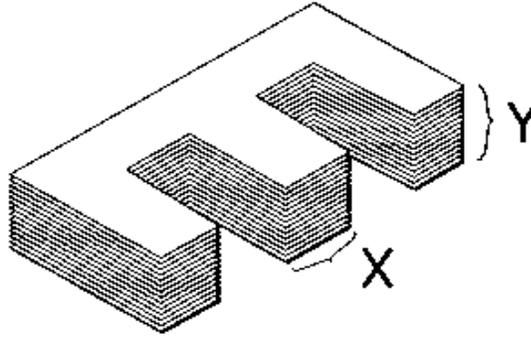


Figura 34. Vista de un núcleo tipo E-I. Fuente: autor.

Para conocer la potencia RMS que puede entregar el transformador solo basta con dividir el área del núcleo por una constante con valor de 1.2 y dicho resultado elevarlo al cuadrado.

$$P_T = \left(\frac{A_N}{1.2}\right)^2 = \left(\frac{(X)(Y)}{1.2}\right)^2 = \left(\frac{2.8cm \times 5cm}{1.2}\right)^2 = \left(\frac{14 cm^2}{1.2}\right)^2 = 136 VA.$$

Teniendo en cuenta este dato podremos determinar el calibre del alambre para el devanado secundario; para ello debemos calcular la corriente que circulara por el secundario, dividiendo la potencia entre el voltaje del devanado y luego ubicar en la tabla AWG el calibre necesario para nuestro bobinado.

$$I_{SEC} = \frac{P_T}{V_{SEC}} = \frac{136 VA}{24 V} = 5.6 A \quad \text{calibre 15AWG}$$

Ahora para determinar el número de espiras o vueltas que debemos enrollar en el devanado lo hacemos de la siguiente manera:

$$\omega_2 = \frac{V_{sec} \times 10^8}{4.44 \times f \times A_N \times \beta_m} = \frac{24V \times 10^8}{4.44 \times 60Hz \times 14cm^2 \times 10000Gauss} = 64.35 vueltas$$

Serán 66 espiras exactamente, debido a que será de tap central debemos enrollar 33 vueltas primero, soldar un cable de tap central y luego las otras 33 vueltas. Para conocer la cantidad de espiras para el devanado primario solo nos basta encontrar el número de vueltas por voltio del devanado secundario y multiplicar dicho valor por el voltaje deseado a la salida de nuestro transformador.

$$\alpha = \frac{\omega_2}{V_2} = \frac{66}{24} = 2.75 \text{ espiras/voltio}$$

$$\omega_1 = (120V)(2.75 \text{ e/v}) = 330 \text{ espiras}$$

Para determinar el calibre del alambre para el primario del transformador se realiza de igual forma que para el secundario. Debemos dividir la potencia entre el voltaje de salida para determinar la corriente que circulará por el primario, luego ubicar en la tabla AWG el calibre de cable adecuado para nuestro bobinado.

$$I_{PRI} = \frac{P_T}{V_{PRI}} = \frac{136 \text{ VA}}{120 \text{ V}} = 1.134 \text{ A} \quad \text{calibre 21AWG}$$

Una vez implementado nuestro diseño del transformador obtuvimos a la salida 115 voltios de corriente alterna aproximadamente, lo que representa un error del 4% con respecto al valor deseado para nuestro diseño.

$$e\% = \frac{V_{estimado} - V_{real}}{V_{real}} \times 100\% = \frac{120V - 115V}{115V} \times 100\% = 4.34\%$$

Con todas las etapas terminadas el circuito diseñado para nuestro inversor de onda cuadrada, se muestra a continuación en la figura 35:

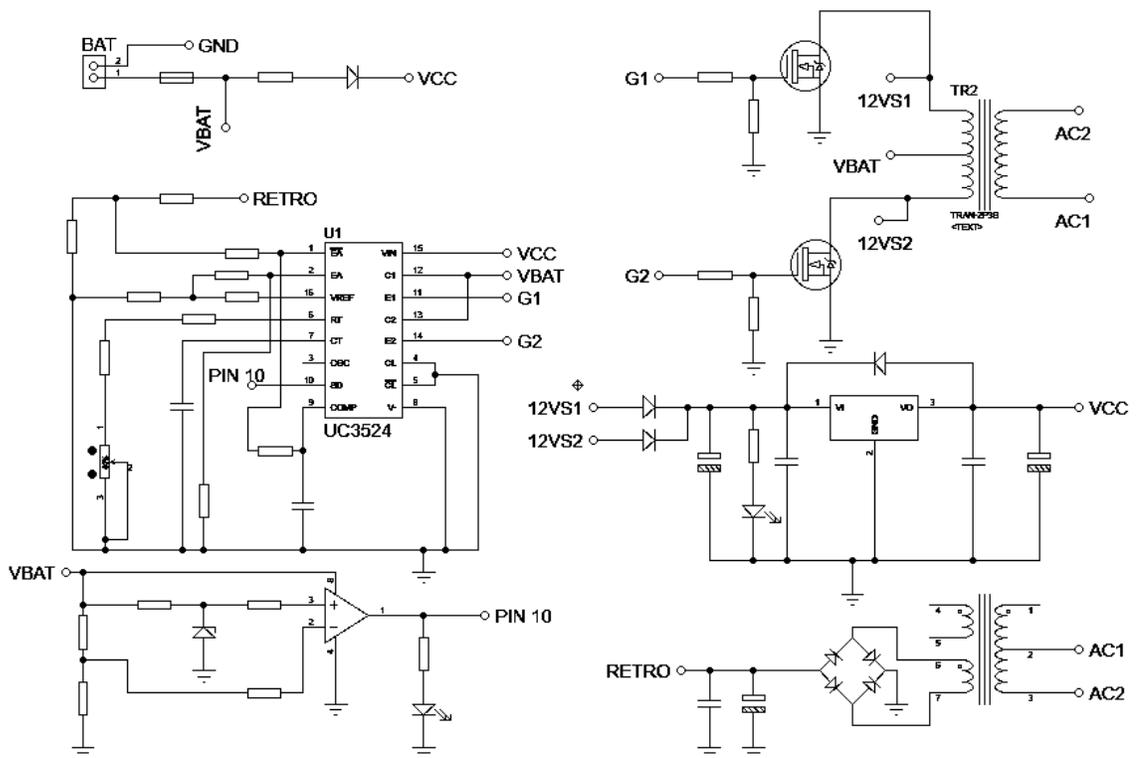


Figura 35. Diseño del inversor de onda cuadrada con el integrado SG3524. Fuente: autor

Finalmente se descartó el uso de dicho diseño porque no es muy recomendado su uso pues una onda cuadrada está cargada de armónicos y con el tiempo tienden a dañar los aparatos eléctricos y el timbre al ser una carga inductiva está más propenso a recibir daños a corto plazo.

Una vez tomada esta decisión, se procedió a investigar sobre las posibles soluciones existentes en el mercado local para dicha tarea, encontrando diversos tipos de equipos encargados de generar electricidad y que cuentan con un sistema armado de carga de las baterías y su respectivo inversor de onda senoidal pura (ideal para cargas inductivas).

En la tienda Era solar, S.A. nos ofrecieron un sistema armado Inversor/Cargador de 600W<sup>6</sup> por un costo aproximado de 1200 dólares, incluido el banco de batería con una carga de 105 Amperes, dicho sistema es el más pequeño en potencia que encontraremos aseguraron los empleados de dicha tienda.

En la empresa Fusión Solar, S.A igualmente ofrecen un sistema armado Inversor/Cargador de 1.5kw por un costo total aproximado a los 1000 dólares, el cual contaría con su cargador de batería y el sistema inversor de corriente así como de su respectivo banco de batería. Cabe señalar que esta empresa actualmente solo ofrece inversores de onda cuadrada de grandes potencias siendo la de 1.5kw la más baja<sup>7</sup>.

Por el alto costo de estos sistemas armados y el tiempo más otros inconvenientes que nos llevaría diseñar un inversor apropiado para la carga que deseamos conectar, hemos decidido no incluir en la implementación de nuestro proyecto dicho sistema de respaldo, quedando esta sección como futura mejora a contemplar por parte del centro educativo.

---

<sup>6</sup>. Anexo 4. Proforma de la empresa Era Solar S.A.

<sup>7</sup>. Anexo 4. Proforma de la empresa Fusión Solar S.A.





Figura 38. Diseño del prototipo en 3D, software proteus 8.9. Fuente: Autor

Una vez terminado el diseño en ARES se procede a la creación de las PCB apoyándonos en el método de planchado<sup>8</sup>; con las pistas planchadas en la tarjeta de cobre se introducen en ácido para obtener el grabado de las pistas de cobre para las conexiones de los diferentes componentes que integran el diseño.

Terminado todo el proceso de planchado y grabado de las pistas de cobre, se procedió a la selección del case que contendrá a nuestro controlador: nuestro circuito tiene las siguientes dimensiones: de largo 145mm y de ancho 5,5mm, por tal razón el case que se selecciono es una cajita de material PVC con las siguientes dimensiones 160mm X 120mm X 80mm. En la figura 39 se observan las características físicas de dicha caja.



Figura 39. Caja de estanque de paredes lisas PVC. Fuente:  
<https://www.vimar.com/es/int/catalog/product/index/code/V55205>

---

<sup>8</sup>. Anexo 6. Proceso de planchado y grabado de la tarjeta pcb.

Una vez ensamblado el prototipo, procedimos a la instalación<sup>9</sup> del mismo y a realizar la entrega de la documentación<sup>10</sup> necesaria para el correcto manejo de nuestro controlador de timbre escolar.

---

<sup>9</sup>. Anexo 7. Instalación del controlador para el timbre.

<sup>10</sup>. Anexo 13. Manual de Usuario.

### **2.8.2. Evaluación del funcionamiento del prototipo.**

En esta fase del diseño del proyecto, luego de instalar<sup>11</sup> nuestro prototipo se procedió a realizar pruebas para garantizar el correcto funcionamiento.

La primera prueba consistía en actualizar la hora y fecha de la unidad de control haciendo uso del teclado 4x4 pudiendo verificar el correcto funcionamiento tanto del teclado como del sistema de control.

La segunda prueba consistía en evaluar el correcto registro de la hora y minuto de activación del timbre así como de su lectura, todo esto mediante el teclado 4x4, los resultados fueron los esperados.

Luego las siguientes pruebas fueron entorno a la correcta comunicación del sistema de control con un Smartphone con sistema Android con nuestra app previamente instalada; los resultados obtenidos aunque satisfactorio se vieron afectados por factores como paredes y árboles, pues en ocasiones se recibieron caracteres que no correspondían a los enviados desde la app o viceversa.

Para validar el correcto funcionamiento del controlador, se evaluaron rubricas de desempeño<sup>12</sup>, en las que se valoraron aspectos como la estética de la instalación, el sistema eléctrico, la facilidad de configuración para el usuario<sup>13</sup> y la correcta comunicación entre el controlador y un Smartphone; obteniendo observaciones solamente en la comunicación con el Smartphone, pues en ocasiones se recibieron datos erróneos y en otras el controlador no atendía las solicitudes del usuario por medio del Smartphone.

En la instalación y realización de las pruebas nos vimos afectados por la situación que atraviesa el mundo con el nuevo virus, limitándonos el ingreso al centro educativo, por tal razón las fechas de pruebas fueron realizadas solamente en Febrero de 2021.

---

11. Anexo 7. Instalación del controlador para el timbre.

12. Anexo 8. Rubricas de desempeño del prototipo.

13. Anexo 13. Manual de Usuario.

## 2.9. Resultados obtenidos.

Como resultado y en base a las directrices que se llevarían a cabo para el diseño del prototipo planteado en los objetivos de nuestro trabajo, obtuvimos un sistema autónomo para la activación del timbre en los horarios de clases del centro educativo Monte Sinaí.

Se obtuvieron resultados satisfactorio de las pruebas aplicadas al diseño para evaluar su buen funcionamiento, el sistema demostró ejecutar casi perfectamente cada función integrada en la unidad de control así como la comunicación entre el Smartphone y PC con el controlador.



Figura 40. Controlador para timbre escolar implementado. Fuente: Autor

## 2.10. Costos del prototipo

En la Tabla 5 se muestra el presupuesto de los costos de los componentes adquiridos para la ejecución de nuestro proyecto, estos fueron adquiridos en el periodo comprendido entre agosto de 2019 hasta septiembre del 2020, el precio con respecto al mercado actual puede presentar variaciones.

**Tabla 5.**

*Costos del prototipo*

Componentes	Cantidad	Precio unitario	Total
Display LCD 2004 I2C	1	C\$400	C\$400
Teclado matricial 4x4 tipo membrana	1	C\$150	C\$150
RTC DS1307	1	C\$170	C\$170
ESP 01	1	C\$200	C\$200
Relay	1	C\$35	C\$35
Arduino nano	1	C\$285	C\$285
Pines conector hembra	2	C\$10	C\$20
Caja plástica para proyecto	1	C\$250	C\$250
Adaptador 5v, 1.5A	1	C\$120	C\$120
Terminal de bloques	2	C\$10	C\$20
Transistor	1	C\$15	C\$15
Regulador 3.3v	1	C\$10	C\$10
Capacitores	2	C\$8	C\$16
Diodo	1	C\$5	C\$5
Gastos varios	--	C\$600	C\$600
Tarjeta de fibra de vidrio	1	C\$140	C\$140
Impresión para PCB	1	C\$30	C\$30
<b>Total de gastos</b>			<b>C\$2,466</b>

## **CAPITULO III: CONCLUSIONES Y RECOMENDACIONES**

### **3.1. CONCLUSIONES**

En base a los objetivos propuestos en la fase inicial de este proyecto y por los resultados obtenidos, concluimos:

- Se concretó el diseño e implementación del prototipo de timbre electrónico programable para el correcto control de las horas de clase en el centro educativo Monte Sinaí.
- Se realizó la implementación de un sistema de comunicación inalámbrico mediante el módulo WiFi ESP8266 para realizar configuraciones al controlador de manera remota mediante PC con sistemas operativo Windows o Smartphone con sistema operativo Android.
- Se desarrolló exitosamente las interfaces que permiten realizar configuraciones de manera remota al controlador mediante PC o Smartphone haciendo uso de las herramientas utilizadas a lo largo de nuestra preparación como fueron Visual Basic 6.0 para la interfaz de PC y MIT App Inventor para la interfaz del Smartphone.
- Por los altos costos de un sistema de respaldo energético comercial y el tiempo y la complejidad de diseñar uno, no se implementó dicha etapa, pero nos dejó importantes conocimientos en cuanto a los datos necesarios para obtener el sistema más adecuado para nuestras necesidades, queda a criterio del centro educativo la obtención del sistema cargador/inversor.
- Se diseñó el manual de usuario para facilitar la utilización del prototipo y se entregó una copia al centro educativo, el cual fue de gran ayuda para comprender el funcionamiento del dispositivo y su correcta manipulación; así lo expresaron las maestras y directora del centro educativo.

- Se realizaron diversas pruebas para corroborar el correcto funcionamiento del prototipo y detectar cualquier anomalía y corregirlo a tiempo, para cumplir con los requerimientos y especificaciones establecidos.

### **3.2 RECOMENDACIONES**

El controlador electrónico para timbres que se ha desarrollado en este trabajo monográfico cumple con todos los requerimientos establecidos en el acuerdo con el centro educativo Monte Sinaí. Sin embargo, este prototipo solo puede registrar la hora y minuto de activación del timbre y estos datos los almacena en la memoria EEPROM del microcontrolador, limitando aún más la cantidad de información almacenable para un mejor desempeño del controlador.

Por lo tanto, a continuación se recomiendan algunas mejoras a implementar sobre este primer prototipo con el objetivo de obtener uno con mejor rendimiento y más prestaciones.

- Implementar un módulo de memoria SD con el objeto de ampliar los datos para la activación del timbre, de manera que el usuario pueda controlar los días de la semana en que se active el timbre así como de un periodo de fechas en los que el controlador activará o no el timbre.
- Mejorar el sistema de comunicación inalámbrico empleando dispositivos de mayor potencia y sensibilidad de manera que se garantice que la información enviada no llegue con errores.
- Implementar un microcontrolador con mayor capacidad de memoria para evitar inconvenientes a la hora de programarlo y que este se vuelva inestable y no ejecute de manera correcta las funciones u optimizar el código hasta reducir lo más posible el uso de la memoria de programa y de datos.

- Hacer uso de otras herramientas para el diseño de interfaces tanto para la PC como para el Smartphone, que dispongan de aspectos estéticos más modernos y prestaciones que faciliten el diseño de las interfaces.

## IV.- Bibliografía

- AlmacénVulcanoSA. (2016). *Sistemas Interconectados a la Red Eléctrica con Respaldo*. Obtenido de [www.gruposaldiarraga.com](http://www.gruposaldiarraga.com): <http://www.gruposaldiarraga.com/respaldo.html>
- Areneda, A., & Chandia, N. (2017). *DESARROLLO DE UN RELOJ/ALARMA PARA LA AUTOMATIZACION DEL TIMBRE ELECTRICO DEL CENTRO DE EDUCACIÓN TECNICA No 9 "ESCUELA INDUSTRIAL", CIPOLLETI*. Obtenido de <https://educacion.rionegro.gov.ar/admarchivos/files/Resultados%20PP/CET%209/Automatizaci%C3%B3n%20del%20timbre.pdf>
- Arrieta, V. (14 de Diciembre de 2018). *MÓDULO BLUETHOOT HC-05 Y HC-06 ARDUINO*. Obtenido de Arca Electrónica: <https://www.arcaelectronica.com/blogs/tutoriales/modulo-bluethoot-hc-05-y-hc-06-arduino>
- Astrom, J. K., & Hagglund, T. (2009). *Control PID Avanzado*.
- Catacora, E., & A., R. (2009). *DISEÑO DE UN PROTOTIPO PARA EL MANEJO AUTOMATIZADO VIA COMPUTADOR, DE UN TIMBRE: TiDiMoBi (Timbre Digital a base de Motores Bipolares)*. Obtenido de [https://www.researchgate.net/publication/282366853\\_Disen%C3%B3\\_de\\_un\\_prototipo\\_para\\_el\\_manejo\\_automatizado\\_via\\_computador\\_de\\_un\\_Timbre\\_TiDiMoBi\\_Timbre\\_Digital\\_a\\_base\\_de\\_Motores\\_Bipolares](https://www.researchgate.net/publication/282366853_Disen%C3%B3_de_un_prototipo_para_el_manejo_automatizado_via_computador_de_un_Timbre_TiDiMoBi_Timbre_Digital_a_base_de_Motores_Bipolares)
- Cobo, R. (S.F.). *El ABC de la Automatización*.
- Cortes Martínez, C., & Vásquez Bohórquez, D. (Abril de 2017). *DESARROLLO DE UN DISPOSITIVO PARA EL REGISTRO Y MONITOREO DE INGRESO DE PERSONAL AL COLEGIO GIMNASIO SABIDURÍA DEL FUTURO Y AUTOMATIZACIÓN DE LAS ALARMAS DE CAMBIO DE CLASE*. Obtenido de <http://repository.udistrital.edu.co/bitstream/11349/6029/9/CortesMartinezCristianSteven2017.pdf>
- Cuello, J. &. (2013-2017). *Aprende a diseñar apps nativas*. Obtenido de <https://appdesignbook.com/es/contenidos/disen%C3%B3-visual-apps-nativas/>
- Daneri, P. A. (2008). *Automatización y Control Industrial*.
- Dorf, B. (S.F.). *Sistemas de Control Moderno 10ma Ed.*
- Estrada, D., Barahona, A., & Rodriguez, M. (Marzo de 2012). *DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE TIMBRE AUTOMATIZADO, COMO APOYO AL CORRECTO CONTROL DE LOS PERIODOS DE HORAS CLASE DURANTE LA JORNADA LABORAL DEL CENTRO ESCOLAR PROFESOR JESÚS LEOCADIO PALENCIA, DE LA CIUDAD DE SAN PABLO TACACHICO*. Obtenido de <https://es.scribd.com/document/117940005/Timbre-Automatico>
- Friedman, M. (12 de Julio de 2010). *App Inventor for Android*. Obtenido de Google Oficial Blog: <https://googleblog.blogspot.com/2010/07/app-inventor-for-android.html>

- Hernández, L. (2016). *ESP8266 todo lo que necesitas saber del módulo WiFi para Arduino*. Obtenido de ProgramaFacil: <https://programarfacil.com/podcast/esp8266-wifi-coste-arduino/>
- Jenkins, A. (s.f.). *El protocolo de comunicacion RS232*. Obtenido de alicat.com: <https://www.alicat.com/es/el-protocolo-de-comunicacion-rs-232-y-su-instrumento-alicat/>
- Letert, J.-R. (01 de Abril de 2020). *TCP/IP Extension*. Obtenido de community.appinventor.mit.edu: <https://community.appinventor.mit.edu/t/tcp-ip-extension/7142>
- MPPTSOLAR. (s.f.). *¿Cómo Funciona un Inversor: Esquema y Funcionamiento?* Obtenido de mpptsolar.com: <https://www.mpptsolar.com/es/esquema-funcionamiento-inversor.html>
- Mujica, C. (2012). *Tutorial Arduino - Conexión Xbee punto a punto*. Obtenido de Geeky Theory: <https://geekytheory.com/tutorial-arduino-comenzando-con-xbee>
- Ortega Serrano, J. (Febrero de 2015). *AUTOMATIZACIÓN DE UN PROTOTIPO DE SISTEMA SONORO CON MINI-PLC PARA EL CAMBIO DE JORNADA EN EL AEIRNNR*. Obtenido de <http://dspace.unl.edu.ec/jspui/handle/123456789/18160>
- Pérez López, E. (2014). *Los sistemas SCADA en la automatización industrial*. Obtenido de <http://www.scielo.sa.cr>: [http://www.scielo.sa.cr/scielo.php?script=sci\\_arttext&pid=S0379-39822015000400003](http://www.scielo.sa.cr/scielo.php?script=sci_arttext&pid=S0379-39822015000400003)
- R., R. (2017). *Inventor de Aplicaciones*. Obtenido de <https://inventordeaplicaciones.es/app-inventor-desde-0/que-es-app-inventor/>
- S.A. (2016). *Reloj en Tiempo Real*. Obtenido de [www.wikipedia.com](http://www.wikipedia.com): [https://es.wikipedia.org/wiki/Reloj\\_en\\_tiempo\\_real](https://es.wikipedia.org/wiki/Reloj_en_tiempo_real)
- S.A. (04 de Febrero de 2007). *Programación de sockets en C de Unix/Linux*. Obtenido de [chuidiang.org](http://www.chuidiang.org): [http://www.chuidiang.org/clinux/sockets/sockets\\_simp.php](http://www.chuidiang.org/clinux/sockets/sockets_simp.php)
- S.A. (15 de Septiembre de 2010). *Medios de Comunicación Inalámbricos. Varitek Universal-Año 3 – Número 9*. Recuperado el 10 de Agosto de 2020, de <https://www.varitek.ec/index.php/temas/tecnologia/medios-de-comunicacion-inalambricos>
- S.A. (02 de Octubre de 2013). *Transmisión Serial - USART*. Obtenido de [wikidot.com](http://wikidot.com): <http://sistdig.wikidot.com/wiki:usart>
- S.A. (21 de febrero de 2014). *Bluetooth-hc-05-y-hc-06-tutorial-de-configuracion*. Obtenido de GeekFactory: <https://www.geekfactory.mx/tutoriales/bluetooth-hc-05-y-hc-06-tutorial-de-configuracion/>
- SanDoRobotics. (s.f.). *Módulo WIFI ESP8266 ESP-01*. Obtenido de SDR, SanDoRobotics: <https://sandorobotics.com/producto/hr0307/>

SEAT, S. (S.F.). *Sensores y Actuadores*, No. 6.

Tomasi, W. (2003). *Sistemas de Comunicaciones Electrónicas*. México: Pearson Educación.

Vallejo, H. D. (2002). Cálculo Simplificado de transformadores. *Saber Electrónica*, 1-6.

# Anexos

Anexo 1. Entrevista realizada a la encargada del centro.

**CENTRO EDUCATIVO MONTE SINAI**

Entrevista

Nombre del entrevistado: Eveling Sánchez Mayorga Fecha: 06/08/2019

1.- Describa el problema que actualmente afecta a la institución en referencia a la activación manual del timbre escolar.

- La Directora, también atiende un grado y este timbre ayudaría al cumplimiento de los horarios de clase.
  - Son muchas las ocupaciones que se desarrollan: Desayuno de los niños, atención a otros niños y este timbre solucionaría el problema.
- 2.- ¿Considera necesario realizar un sistema independiente para la activación del timbre?  
¿Por qué?
- Este sistema facilitaría el cumplimiento del horario de clase, en cuanto al cambio de clases.

3.- ¿Cuáles serían los puntos a tomar en cuenta para la elaboración del diseño de este prototipo?

- La necesidad urgente del centro de un timbre automático.
- Cumplimiento con el horario de clase.
- Organizar mejor el cambio de clase.

4.- ¿Qué beneficio traería para la institución la implementación de este proyecto?

- Cumplimiento en el horario de clase.
- Menos estrés para la Directora, poder atender todas las actividades que se desarrollan en el centro sin estar pendiente de tocar el timbre.

Firma y sello del responsable del Centro Educativo.



## Anexo 2. Acuerdo con el Centro Educativo

### Acuerdo

A los bachilleres Cristian Ernesto Solórzano Cano y Sarahi Yaoska Sandoval Sandoval, el Centro Educativo Monte Sinaí les presentó una problemática que ha causado retrasos a la hora de impartir clases ya que al ser un centro con poco personal de trabajo y debido a que sus instalaciones son pequeñas el cambio en cada periodo de clase se realiza de forma manual, debido a esto los bachilleres propusieron el "Diseño e implementación de un prototipo de timbre electrónico programable para el correcto control de las horas clases en el centro educativo Monte Sinaí" con el cual tanto la institución como los estudiantes obtendrían beneficios, el centro concede el estudio y desarrollo del proyecto bajo los siguientes términos:

1. Se pueden realizar visitas al centro con previo aviso a la Dirección para poder realizar cualquier tipo de pruebas, entrevistas, consultas respetando los horarios preestablecidos y las normas de seguridad.
2. El centro educativo reembolsará el dinero ocupado en las compras de los componentes utilizados en el desarrollo del prototipo, hasta los 3,000 Córdobas.
3. Se debe entregar el prototipo probado e instalado así como un manual de uso del sistema.

Este sistema se registrará bajo los siguientes parámetros

- ❖ Sistema autónomo de activación para el timbre.
- ❖ Las horas de activación sean programables y se puedan ejecutar de manera sencilla para el usuario.
- ❖ Que muestre al usuario las horas y minutos en que se dará la activación del timbre.
- ❖ Le permita al usuario tener la facilidad de programar las horas de forma remota mediante la PC o un Smartphone con SO Android.

El plazo para concluir este proyecto será determinado por los estudiantes.

Firman el acuerdo en la ciudad de Managua a los 07 días del mes de octubre del 2019

Sarahi Sandoval

Br. Sarahi Yaoska Sandoval  
Egresada de Ing. Electrónica  
N° Carnet. 2013-62325

Solorzano

Br. Cristian Ernesto Solórzano  
Egresado de Ing. Electrónica  
N° Carnet. 2013-61571

E. Sánchez

Lic. Eveling Sánchez  
Directora del centro



### Anexo 3. Timbres electrónicos programables.



Nuevo | 5 vendidos

#### Control Automático Timbre Programable Escolar 4-zonas Atb004

\$ 4,060

en 12x \$ 407<sup>23</sup>IVA incluido

[Ver los medios de pago](#)

Envío gratis a todo el país

Conoce los tiempos y las formas de envío.

[Calcular cuándo llega](#)

Devolución gratis

Tienes 30 días desde que lo recibes.

[Conocer más](#)

Vendido por HUMFERNANDEZ2012

No entrega a tiempo

¡Última disponible!



[facebook](#) [Youtube](#) [Pagos en Línea](#) [Ser un Aliado Comercial](#)

[INICIO](#) [QUIENES SOMOS](#) [PRODUCTOS](#) [SERVICIOS](#) [SECTOR SALUD](#) [CONTACTO](#) [NOTICIAS](#) [PAGOS EN LINEA](#)



## CAMPANA ELECTRONICA PROGRAMABLE

Con este equipo usted puede controlar de forma precisa y ordenada el cambio, salida de clases y descansos de su Institución Educativa o planta de trabajo, evitando pérdidas de tiempo al tener que enviar a un estudiante, profesor o personal de la planta para tocar la sirena o timbre. Evite dolores de cabeza o el estrés producido al tener que estar pendiente del reloj en cada cambio de clase o turno de trabajo.





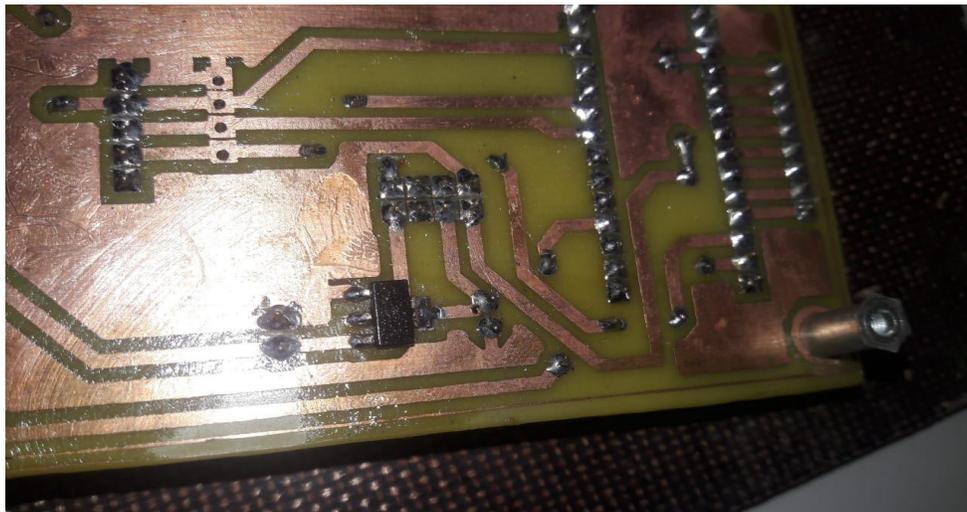




Anexo 5. Imágenes del trabajo realizado antes de la instalación.



Planchado y perforado de la tarjeta pcb para la implementación del prototipo



Parte inferior de la tarjeta donde van soldados los dispositivos

Anexo 6. Instalación actual del timbre.



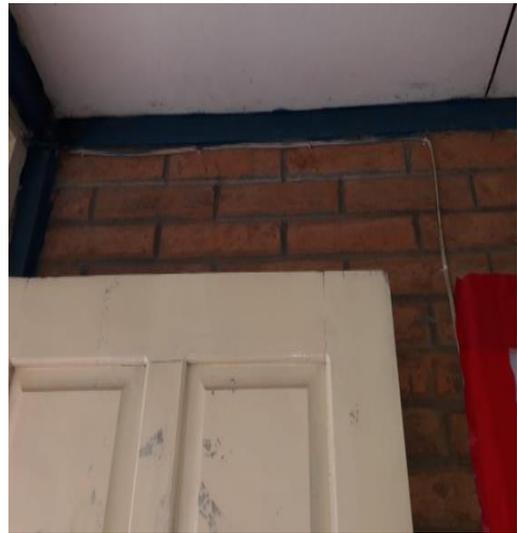
Interruptor para activación de timbre



Alimentación para la activación del timbre



Campana que suena para orientar el cambio de clase



Recorrido del cable de energía

Anexo 7. Instalación del controlador para el timbre.



Perforación de la pared para instalar la caja



Instalación de la caja del controlador



Caja del controlador instalada



Circuiteria interna del controlador de timbre



Instalación final del controlador



Controlador funcionando.

Anexo 8. Rubricas de desempeño del prototipo.

Rúbricas de desempeño con el fin de evaluar la instalación y la función del sistema de control instalado en el centro Educativo Monte Sinaí.

Durante el mes de febrero del año 2021 se realizaron pruebas en las que pudieran determinar si el sistema era funcional por medio de los siguientes criterios:

Criterios de desempeño	Si	No	Observaciones
La caja donde está instalado el sistema es adecuada y no obstruye el espacio de trabajo ni representa ningún tipo de peligro al acercarse a la instalación.	✓		
La configuración del sistema es de fácil manejo para el operador usando la app mediante el celular/pc, así también como el teclado de membrana que posee el sistema.	✓		Por medio del wifi se recibieron datos erróneos y en ocasiones no se atendían las peticiones.
La activación del timbre se cumple y tiene una respuesta rápida ante los tiempos de operación, en que están programados los horarios de clase	✓		
El sistema posee un interruptor que permita el apagado y encendido del mismo para evitar cualquier anomalía cuando no haya personal laborando	✓		
Existe sobrecalentamiento en los cables de conexión del sistema		✓	
La instalación eléctrica del sistema es segura y no hay riesgo de provocar una descarga eléctrica al manipularlo.	✓		
El funcionamiento general del sistema instalado es el esperado	✓		

*E. Schuck*



## Anexo 9. Código grabado en el microcontrolador.

```
C:\Users\Cristian\SkyDrive\Documentos\Tesis\CódigoTesis\Codigotesis\Codigotesis\Codigotesis.ino      martes, 16 de febrero de 2021 17:52
#include <Separador.h>
#include<LiquidCrystal_I2C.h>
#include<EEPROM.h>
#include "Wire.h"
#include<Keypad.h>
#define DS1307_I2C_ADDRESS 0x68 //DIRECCION ARDUINO PARA I2C

Separador x;

char pulsacion;
byte horas, minuto,second, minute, hour, dayOfWeek, dayOfMonth, month, year, pulsacion1;
byte Pins_filas[4]={2, 3, 4, 5};://{9, 8, 7, 6};
byte Pins_cols[4]={6, 7, 8, 9};://{5, 4, 3, 2};
char key[4][4]={
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'}
};
Keypad tecla=Keypad(makeKeymap(key), Pins_filas, Pins_cols, 4, 4);

byte decToBCD(byte val){//convertimos de Decimal a BCD
  return((val/10*16)+(val%10));
}
byte BCDToDec(byte val){//convertimos de BCD a Decimal
  return((val/16*10)+(val%16));
}
byte ASCIItoDEC(byte val){
  return ((val-48));
}

#include "SoftwareSerial.h"
SoftwareSerial EspSerial(10, 11); // RX, TX eran 2 y 3 respectivamente d10 a tx mod y d11 al
{R1K-rx-R2K-gnd}divisor de voltaje

LiquidCrystal_I2C lcd(0x27, 20, 4);

void setup() {
  Wire.begin();
  lcd.begin();
  delay(2000);
  EspSerial.begin(115200);
  delay(200);
  EspSerial.println("AT+CWMODE=2");//EMPIEZA CON ESP
  delay(500);
  EspSerial.println("AT+CIPMUX=1");
  delay(500);
  EspSerial.println("AT+CIPSERVER=1,1000");
  delay(500);
  lcd.print("BIENVENIDOS M. SINAI");
  delay(1000);
  lcd.clear();
}

void loop() {

  const unsigned long intervalo=500;
  unsigned long milliactual=0;
  unsigned long millianterior=0;
  int registro=0;

  inicio:
  pulsacion=tecla.getKey();
  while(pulsacion==0){
    milliactual=millis();
    if(milliactual-millianterior>=intervalo){
      millianterior=milliactual;
      pulsacion=tecla.getKey();
      getrtc();
      if(second==0){
        lcd.setCursor(14,1); lcd.print(" ");
      }
    }
  }
}
```

```

if (minute >= 0 && minute <= 9) { lcd.setCursor (11,1); lcd.print (" "); }
if (hour >= 0 && hour <= 9) { lcd.setCursor (8,1); lcd.print (" "); }
if (dayOfmonth >= 1 && dayOfmonth <= 9) { lcd.setCursor (7,2); lcd.print (" "); }
if (month >= 1 && month <= 9) { lcd.setCursor (11,2); lcd.print (" "); }
if (year >= 0 && year <= 9) { lcd.setCursor (14,2); lcd.print (" "); }
if (dayOfWeek >= 2 && dayOfWeek <= 7 && second == 0) {
    alarmas (hour,minute);
}
}
}

if (EspSerial.available ()) { // check if the ESP module is sending a message
    while (EspSerial.available ()) { // The esp has data so display its output to the serial
        window
        String ca = EspSerial.readString (); // read the next character.
        delay (20);
        if (ca.indexOf ("actualizar") > 0) {
            lcd.clear ();
            dayOfmonth = x.separa (ca, '*', 1).toInt ();
            month = x.separa (ca, '*', 2).toInt ();
            year = x.separa (ca, '*', 3).toInt () - 2000;
            dayOfWeek = x.separa (ca, '*', 4).toInt ();
            hour = x.separa (ca, '*', 5).toInt ();
            minute = x.separa (ca, '*', 6).toInt ();
            second = x.separa (ca, '*', 7).toInt ();
            rtc (dayOfmonth, month, year, dayOfWeek, hour, minute, second);
        }
        if (ca.indexOf ("registro") > 0) {
            lcd.clear ();
            registro = x.separa (ca, '*', 1).toInt ();
            horas = x.separa (ca, '*', 2).toInt ();
            minuto = x.separa (ca, '*', 3).toInt ();
            regEEPROM (registro, horas, minuto);
        }
        if (ca.indexOf ("leer") > 0) {
            lcd.clear ();
            registro = x.separa (ca, '*', 1).toInt ();
            getalarmas (false, registro);
            delay (100);
        }
    }
}
pulsacion = tecla.getKey ();
}

switch (pulsacion) {
    case 'A':
        rtconfig (second);
        pulsacion = 0;
        goto inicio;
        break;
    case 'B':
        setalarmas ();
        pulsacion = 0;
        goto inicio;
        break;
    case 'C':
        lcd.clear ();
        lcd.setCursor (0,0);
        lcd.print ("Digite # de alarma");
        lcd.setCursor (0,1);
        lcd.print ("a leer (1-9):");
        for (int i = 0; i < 1; i++) {
            pulsacion = tecla.getKey ();
            while (pulsacion == 0) {
                pulsacion = tecla.getKey ();
            }
            pulsacion1 = ASCIItoDEC (pulsacion);
            lcd.setCursor (i+12,1);
            lcd.print (pulsacion1);
        }
}

```

```

        getalarmas(true,pulsacion1);
        pulsacion=0;
        goto inicio;
        break;
    default:
        lcd.clear();
        lcd.print("char invalid");
        delay(1000);
        pulsacion=0;
        lcd.clear();
        goto inicio;
    break;
}
}

void rtconfig(byte t){
byte s[2], n[2], c;
minuto:
s[0]=0;s[1]=0;n[0]=0;n[1]=0;c=0;
while(c==0){
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Ingrese Los Min");
lcd.setCursor(0,1);
lcd.print("formato(00-59):");
lcd.setCursor(0,3);
lcd.print("*=Salir #=Siguiente");
for(int i=0; i<2; i++){
minu: pulsacion = tecla.getKey();
while(pulsacion==0){
pulsacion = tecla.getKey();
}
s[i]=ASCIIToDEC(pulsacion);
n[i]=s[i]+48;
lcd.setCursor(i+16,1);
lcd.print(s[i]);
while(n[0]=='*'||n[1]=='*'){ i=0; c=0; goto fin;}
while(n[0]=='#'){ i=0;c=1;goto hora;}
while(n[1]=='#'){ minuto=s[0];i=0; c=1; goto hora;}
while(s[0]>5||s[1]>9){lcd.clear();lcd.print(" Respete Formato");delay(500);goto minuto;}
minuto=s[0]*10 + s[1];
}
for(int i=0; i<1; i++){
pulsacion = tecla.getKey();
while(pulsacion==0){
pulsacion = tecla.getKey();
}
while(pulsacion=='#'){ c=1;goto hora;}
while(pulsacion=='*'){ c=0;goto minuto;}
}
}
delay(250);

hora:
s[0]=0;s[1]=0;n[0]=0;n[1]=0;c=1;
while( c==1){
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Ingrese La Hora");
lcd.setCursor(0,1);
lcd.print("formato(00-23):");
for(int i=0; i<2; i++){
pulsacion = tecla.getKey();
while(pulsacion==0){
pulsacion = tecla.getKey();
}
s[i]=ASCIIToDEC(pulsacion);
n[i]=s[i]+48;
lcd.setCursor(i+16,1);
lcd.print(s[i]);

```

```

if(s[1]>=0){
  lcd.setCursor(0,3);
  lcd.print("*=Atras  #=Siguiente");
  while(n[0]=='*'||n[1]=='*'){ c=0;goto minuto;}
  while(n[0]=='#'){ i=0;c=2;goto diasemana;}
  while(n[1]=='#'){ hour=s[0];i=0; c=2;goto diasemana;}
  while(s[0]>2||s[1]>9){lcd.clear();lcd.print(" Respete Formato");delay(500);goto hora;}
  while(s[0]==2&& s[1]>3){lcd.clear();lcd.print(" Respete Formato");delay(500);goto hora;}
  hour=s[0]*10+s[1];
}
for(int i=0; i<1; i++){
  pulsacion = tecla.getKey();
  while(pulsacion==0){
    pulsacion = tecla.getKey();
  }
  while(pulsacion=='#'){ c=2;goto diasemana;}
  while(pulsacion=='*'){ c=1;goto hora;}
}
}
delay(250);

diasemana:
s[0]=0;s[1]=0;n[0]=0;n[1]=0;c=2;
while( c==2){
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Ingrese Dia de sema");
  lcd.setCursor(0,1);
  lcd.print("form(1_Dom-7_Sab):");
  for(int i=0; i<1; i++){
    pulsacion = tecla.getKey();
    while(pulsacion==0){
      pulsacion = tecla.getKey();
    }
    s[i]=ASCIIToDEC(pulsacion);
    n[i]=s[i]+48;
    lcd.setCursor(i+19,1);
    lcd.print(s[i]);
    if(s[0]>=0){
      lcd.setCursor(0,3);
      lcd.print("*=Atras  #=Siguiente");
      while(n[0]=='*'){ c=1;goto hora;}
      while(n[0]=='#'){ i=0; c=3;goto dia;}
      while(s[0]>7||s[0]<1){lcd.clear();lcd.print(" Respete Formato");delay(500);goto diasemana;}
      dayOfWeek=s[0];
    }
    for(int i=0; i<1; i++){
      pulsacion = tecla.getKey();
      while(pulsacion==0){
        pulsacion = tecla.getKey();
      }
      while(pulsacion=='#'){ c=3;goto dia;}
      while(pulsacion=='*'){ c=2;goto diasemana;}
    }
  }
}
delay(250);

dia:
s[0]=0;s[1]=0;n[0]=0;n[1]=0;c=3;
while( c==3){
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Ingrese Dia de Mes");
  lcd.setCursor(0,1);
  lcd.print("formato(01-31):");
  for(int i=0; i<2; i++){
    pulsacion = tecla.getKey();
    while(pulsacion==0){
      pulsacion = tecla.getKey();
    }
    s[i]=ASCIIToDEC(pulsacion);
  }
}

```

```

n[1]=s[1]+48;
lcd.setCursor(i+16,1);
lcd.print(s[i]);
if(s[1]>=0){
lcd.setCursor(0,3);
lcd.print("*=Atras #=Siguiente");
while(n[0]=='*'||n[1]=='*'){ c=2;goto diasemana;}
while(n[0]=='#'){ i=0;c=4;goto mes;}
while(n[0]=='#'||n[1]=='#'){ dayOfmonth=s[0];i=0; c=4;goto mes;}
while(s[0]>3||s[1]>9){lcd.clear();lcd.print(" Respete Formato");delay(500);goto dia;}
while(s[1]>1&& s[0]==3){lcd.clear();lcd.print(" Respete Formato");delay(500);goto dia;}
dayOfmonth=s[0]*10+s[1];
}
for(int i=0; i<1; i++){
pulsacion = tecla.getKey();
while(pulsacion==0){
pulsacion = tecla.getKey();
}
while(pulsacion=='#'){ c=4;goto mes;}
while(pulsacion=='*'){ c=3;goto dia;}
}
}
delay(250);

mes:
s[0]=0;s[1]=0;n[0]=0;n[1]=0;c=4;
while( c==4){
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Ingrese # de Mes");
lcd.setCursor(0,1);
lcd.print("formato(01-12):");
for(int i=0; i<2; i++){
pulsacion = tecla.getKey();
while(pulsacion==0){
pulsacion = tecla.getKey();
}
s[i]=ASCIIToDEC(pulsacion);
n[1]=s[1]+48;
lcd.setCursor(i+16,1);
lcd.print(s[i]);
if(s[1]>=0){
lcd.setCursor(0,3);
lcd.print("*=Atras #=Siguiente");
while(n[0]=='*'||n[1]=='*'){ c=3;goto dia;}
while(n[0]=='#'){ i=0;c=5;goto anho;}
while(n[1]=='#'){ month=s[0];i=0; c=5;goto anho;}
while(s[0]>1||s[1]>9){lcd.clear();lcd.print(" Respete Formato");delay(500);goto mes;}
while(s[0]==1&& s[1]>2){lcd.clear();lcd.print(" Respete Formato");delay(500);goto mes;}
month=s[0]*10+s[1];
}
for(int i=0; i<1; i++){
pulsacion = tecla.getKey();
while(pulsacion==0){
pulsacion = tecla.getKey();
}
while(pulsacion=='#'){ c=5;goto anho;}
while(pulsacion=='*'){ c=4;goto mes;}
}
}
}
delay(250);

anho:
s[0]=0;s[1]=0;n[0]=0;n[1]=0;c=5;
while( c==5){
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Ingrese Anho");
lcd.setCursor(0,1);
lcd.print("formato(00-99):");
for(int i=0; i<2; i++){

```

```

pulsacion = tecla.getKey();
while(pulsacion==0){
  pulsacion = tecla.getKey();
}
s[i]=ASCIIToDEC(pulsacion);
n[i]=s[i]+48;
while(n[0]!='#')(i=0;c=0;goto save;);
lcd.setCursor(i+16,1);
lcd.print(s[i]);
if(s[i]>=0){
  lcd.setCursor(0,3);
  lcd.print("*=Atras  #=Guardar");
  while(n[0]!='*'||n[1]!='*'){ c=4;goto mes;};
  while(n[1]!='#'){ year=s[0];i=0; c=0;goto save;};
  while(s[0]>9||s[1]>9){lcd.clear();lcd.print(" Respete Formato");delay(500);goto anho;};
  year=s[0]*10+s[1];
}
for(int i=0; i<1; i++){
  pulsacion = tecla.getKey();
  while(pulsacion==0){
    pulsacion = tecla.getKey();
  }
  while(pulsacion=='#'){ c=0;goto save;};
  while(pulsacion=='*'){ c=5;goto anho;};
}
save:
delay(250);
lcd.clear();

rtc(dayOfMonth,month,year,dayOfWeek,hour,minute,second);

fin:
s==0;n==0;
lcd.clear();
delay(1);
}

void rtc(byte dia,byte mes,byte anho,byte Dsem,byte hour,byte minute,byte t){

  Wire.beginTransmission(DS1307_I2C_ADDRESS);
  Wire.write((byte)0x00);
  Wire.write(decToBCD(t));
  Wire.write(decToBCD(minute));
  Wire.write(decToBCD(hour));
  Wire.write(decToBCD(Dsem));
  Wire.write(decToBCD(dia));
  Wire.write(decToBCD(mes));
  Wire.write(decToBCD(anho));
  Wire.endTransmission();
  Wire.beginTransmission(DS1307_I2C_ADDRESS);
  Wire.write((byte)0x00);
  Wire.endTransmission();

}

void getrtc(){

String dayweek[7][1]=("Dom","Lun","Mar","Mie","Jue","Vie","Sab");
String
numonth[12][1]=("Ene","Feb","Mar","Abr","May","Jun","Jul","Ago","Sep","Oct","Nov","Dic");

  Wire.beginTransmission(DS1307_I2C_ADDRESS);
  Wire.write((byte)0x00);
  Wire.endTransmission();
  Wire.requestFrom(DS1307_I2C_ADDRESS, 7);
  second=BCDToDec(Wire.read() &0x7f);
  minute=BCDToDec(Wire.read());
  hour=BCDToDec(Wire.read() &0x3f);
  dayOfWeek=BCDToDec(Wire.read());
  dayOfMonth=BCDToDec(Wire.read());

```

```

month=BCDToDec(Wire.read());
year=BCDToDec(Wire.read());

lcd.setCursor(0,1);
lcd.print("Time:");
lcd.setCursor(7,1);
lcd.print(hour);
lcd.setCursor(10,1);
lcd.print(minute);
lcd.setCursor(13,1);
lcd.print(second);
lcd.setCursor(16,1);
lcd.print(dayweek[1][dayOfWeek-2]);
lcd.setCursor(0,2);
lcd.print("Date:");
lcd.setCursor(7,2);
lcd.print(dayOfMonth);
lcd.setCursor(10,2);
lcd.print(month);
lcd.setCursor(13,2);
lcd.print(year);
lcd.setCursor(16,2);
lcd.print(nummonth[1][month-2]);
lcd.setCursor(9,1);
lcd.print(":");
lcd.setCursor(12,1);
lcd.print(":");
lcd.setCursor(9,2);
lcd.print("/");
lcd.setCursor(12,2);
lcd.print("/");

return hour,minute,dayOfWeek,second,dayOfMonth,month,year;
}

void setalarmas(){
byte registro,alarma[4];
in:
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Digite # de alarma");
lcd.setCursor(0,1);
lcd.print("a actualizar(1-9):");
lcd.setCursor(0,3);
lcd.print("Salir= '*'");
for(int i=0; i<1; i++){
pulsacion = tecla.getKey();
while(pulsacion==0){
pulsacion = tecla.getKey();
}
if(pulsacion>'0' && pulsacion<'A'){
registro=ASCIIToDEC(pulsacion);
lcd.setCursor(i+18,1);
lcd.print(registro);
delay(500);
}
else if(pulsacion=='*'){goto fin;}
else{registro=0; goto in;}
}
all:
alarma[0]=0;alarma[1]=0;alarma[2]=0;alarma[3]=0;
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Digite alarma # ");
lcd.print(registro);
lcd.setCursor(0,1);
lcd.print("(Hora/Min):");
lcd.setCursor(13,1);
lcd.print("/");
for(int i=0; i<4; i++){

```

```

pulsacion = tecla.getKey();
while(pulsacion==0){
  pulsacion = tecla.getKey();
}
  alarma[1]=ASCIIToDEC(pulsacion);
lcd.setCursor(11,1);
lcd.print(alarma[0]);
if(alarma[1]!=0){
  lcd.setCursor(12,1);
  lcd.print(alarma[1]);}
if(alarma[2]!=0){
  lcd.setCursor(14,1);
  lcd.print(alarma[2]);}
if(alarma[3]!=0){
  lcd.setCursor(15,1);
  lcd.print(alarma[3]);}
while(alarma[0]>2||alarma[1]>9||alarma[2]>5||alarma[3]>9){
  lcd.clear();lcd.setCursor(0,1);
  lcd.print("Hor (00-23)-Mi (00-59)");delay(500);goto all;}
while(alarma[0]==2&&alarma[1]>3||alarma[2]>5||alarma[3]>9){
  lcd.clear();lcd.setCursor(0,1);
  lcd.print("Hor (00-23)-Mi (00-59)");delay(500);goto all;}
delay(200);
}
lcd.clear();
horas=alarma[0]*10+alarma[1];
minuto=alarma[2]*10+alarma[3];
regEEPROM(registro,horas,minuto);

fin:
lcd.clear();
delay(10);
}

void regEEPROM(byte registro,byte hora, byte minuto){

byte alarma[4];

  alarma[0]=hora/10;
  alarma[1]=hora%10;
  alarma[2]=minuto/10;
  alarma[3]=minuto%10;

switch(registro){

  case 1:
    for(int address=0;address<4;address++){
      EEPROM.update(address,alarma[address]);}
    delay(100);
    break;

  case 2:
    for(int address=0;address<4;address++){
      EEPROM.update(address+4,alarma[address]);}
    delay(100);
    break;

  case 3:
    for(int address=0;address<4;address++){
      EEPROM.update(address+8,alarma[address]);}
    delay(100);
    break;

  case 4:
    for(int address=0;address<4;address++){
      EEPROM.update(address+12,alarma[address]);}
    delay(100);
    break;

  case 5:
    for(int address=0;address<4;address++){

```

```

        EEPROM.update(address+16, alarma[address]); }
        delay(100);
    break;

    case 6:
        for (int address=0; address<4; address++) {
            EEPROM.update(address+20, alarma[address]); }
        delay(100);
    break;

    case 7:
        for (int address=0; address<4; address++) {
            EEPROM.update(address+24, alarma[address]); }
        delay(100);
    break;

    case 8:
        for (int address=0; address<4; address++) {
            EEPROM.update(address+28, alarma[address]); }
        delay(100);

    break;

    case 9:
        for (int address=0; address<4; address++) {
            EEPROM.update(address+32, alarma[address]); }
        delay(100);
    break;
}
}

void getalarmas(bool key, byte registro){
    String trama="";

    switch(registro){
    case 1:
        horas=EEPROM.read(0)*10 + EEPROM.read(1);
        minuto=EEPROM.read(2)*10 + EEPROM.read(3);
        if (key==true) {
            lcd.setCursor(0,2);
            lcd.print("Hora/Min:");
            lcd.setCursor(10,2);
            lcd.print(horas);
            lcd.print(":");
            lcd.print(minuto);
            delay(2000);
            lcd.clear();
        }
        else{
            trama.concat(String(horas));
            trama.concat(":");
            trama.concat(String(minuto));
            delay(20);
            EspSerial.println("AT+CIPSEND=0,5");
            delay(300);
            EspSerial.println(trama);
        }
    break;
    case 2:
        horas=EEPROM.read(4)*10 + EEPROM.read(5);
        minuto=EEPROM.read(6)*10 + EEPROM.read(7);
        if (key==true) {
            lcd.setCursor(0,2);
            lcd.print("Hora/Min:");
            lcd.setCursor(10,2);
            lcd.print(horas);
            lcd.print(":");
            lcd.print(minuto);
            delay(2000);
        }
    }
}

```

```
    lcd.clear();}
    else{
    trama.concat(String(horas));
    trama.concat(":");
    trama.concat(String(minuto));
    delay(20);
    EspSerial.println("AT+CIPSEND=0,5");
    delay(300);
    EspSerial.println(trama);
    }
break;
case 3:
    horas=EEPROM.read(8)*10 + EEPROM.read(9);
    minuto=EEPROM.read(10)*10 + EEPROM.read(11);
    if(key==true){
    lcd.setCursor(0,2);
    lcd.print("Hora/Min:");
    lcd.setCursor(10,2);
    lcd.print(horas);
    lcd.print(":");
    lcd.print(minuto);
    delay(2000);
    lcd.clear();}
    else{
    trama.concat(String(horas));
    trama.concat(":");
    trama.concat(String(minuto));
    delay(20);
    EspSerial.println("AT+CIPSEND=0,5");
    delay(300);
    EspSerial.println(trama);
    }
break;
case 4:
    horas=EEPROM.read(12)*10 + EEPROM.read(13);
    minuto=EEPROM.read(14)*10 + EEPROM.read(15);
    if(key==true){
    lcd.setCursor(0,2);
    lcd.print("Hora/Min:");
    lcd.setCursor(10,2);
    lcd.print(horas);
    lcd.print(":");
    lcd.print(minuto);
    delay(2000);
    lcd.clear();}
    else{
    trama.concat(String(horas));
    trama.concat(":");
    trama.concat(String(minuto));
    delay(20);
    EspSerial.println("AT+CIPSEND=0,5");
    delay(300);
    EspSerial.println(trama);
    }
break;
case 5:
    horas=EEPROM.read(16)*10 + EEPROM.read(17);
    minuto=EEPROM.read(18)*10 + EEPROM.read(19);
    if(key==true){
    lcd.setCursor(0,2);
    lcd.print("Hora/Min:");
    lcd.setCursor(10,2);
    lcd.print(horas);
    lcd.print(":");
    lcd.print(minuto);
    delay(2000);
    lcd.clear();}
    else{
    trama.concat(String(horas));
    trama.concat(":");
    trama.concat(String(minuto));
```

```
    delay(20);
    EspSerial.println("AT+CIPSEND=0,5");
    delay(300);
    EspSerial.println(trama);
  }
break;
case 6:
  horas=EEPROM.read(20)*10 + EEPROM.read(21);
  minuto=EEPROM.read(22)*10 + EEPROM.read(23);
  if(key==true){
    lcd.setCursor(0,2);
    lcd.print("Hora/Min:");
    lcd.setCursor(10,2);
    lcd.print(horas);
    lcd.print(":");
    lcd.print(minuto);
    delay(2000);
    lcd.clear();
  }
  else{
    trama.concat(String(horas));
    trama.concat(":");
    trama.concat(String(minuto));
    delay(20);
    EspSerial.println("AT+CIPSEND=0,5");
    delay(300);
    EspSerial.println(trama);
  }
break;
case 7:
  horas=EEPROM.read(24)*10 + EEPROM.read(25);
  minuto=EEPROM.read(26)*10 + EEPROM.read(27);
  if(key==true){
    lcd.setCursor(0,2);
    lcd.print("Hora/Min:");
    lcd.setCursor(10,2);
    lcd.print(horas);
    lcd.print(":");
    lcd.print(minuto);
    delay(2000);
    lcd.clear();
  }
  else{
    trama.concat(String(horas));
    trama.concat(":");
    trama.concat(String(minuto));
    delay(20);
    EspSerial.println("AT+CIPSEND=0,5");
    delay(300);
    EspSerial.println(trama);
  }
break;
case 8:
  horas=EEPROM.read(28)*10 + EEPROM.read(29);
  minuto=EEPROM.read(30)*10 + EEPROM.read(31);
  if(key==true){
    lcd.setCursor(0,2);
    lcd.print("Hora/Min:");
    lcd.setCursor(10,2);
    lcd.print(horas);
    lcd.print(":");
    lcd.print(minuto);
    delay(2000);
    lcd.clear();
  }
  else{
    trama.concat(String(horas));
    trama.concat(":");
    trama.concat(String(minuto));
    delay(20);
    EspSerial.println("AT+CIPSEND=0,5");
    delay(300);
    EspSerial.println(trama);
  }
}
```

```

break;
case 9:
  horas=EEPROM.read(32)*10 + EEPROM.read(33);
  minuto=EEPROM.read(34)*10 + EEPROM.read(35);
  if(key==true){
    lcd.setCursor(0,2);
    lcd.print("Hora/Min:");
    lcd.setCursor(10,2);
    lcd.print(horas);
    lcd.print(":");
    lcd.print(minuto);
    delay(2000);
    lcd.clear();
  }
  else{
    trama.concat(String(horas));
    trama.concat(":");
    trama.concat(String(minuto));
    delay(20);
    EspSerial.println("AT+CIPSEND=0,5");
    delay(300);
    EspSerial.println(trama);
  }
break;
default:
  lcd.clear();
  lcd.print("Registros del 1-9");
  delay(2000);
  lcd.clear();
break;
}
}

void alarmas(byte c,byte b){

byte hora[9],minutos[9];

hora[0]=EEPROM.read(0)*10+EEPROM.read(1);
minutos[0]=EEPROM.read(2)*10+EEPROM.read(3);
if(hora[0]==c&&minutos[0]==b){
  lcd.clear();
  lcd.setCursor(0,1);
  lcd.print("  Alarma activa");
  lcd.setCursor(7,2);
  lcd.print(c);
  lcd.setCursor(9,2);
  lcd.print(":");
  lcd.setCursor(10,2);
  lcd.print(b);
  analogWrite(A0,255);
  delay(5000);
  analogWrite(A0,0);
  goto fin;
}
delay(10);

hora[1]=EEPROM.read(4)*10+EEPROM.read(5);
minutos[1]=EEPROM.read(6)*10+EEPROM.read(7);
if(hora[1]==c&&minutos[1]==b){
  lcd.clear();
  lcd.setCursor(0,1);
  lcd.print("  Alarma activa");
  lcd.setCursor(7,2);
  lcd.print(c);
  lcd.setCursor(9,2);
  lcd.print(":");
  lcd.setCursor(10,2);
  lcd.print(b);
  analogWrite(A0,255);
  delay(5000);
  analogWrite(A0,0);
  goto fin;
}

```

```
    }  
    delay(10);  
  
    hora[2]=EEPROM.read(8)*10+EEPROM.read(9);  
    minutos[2]=EEPROM.read(10)*10+EEPROM.read(11);  
    if(hora[2]==c&&minutos[2]==b){  
        lcd.clear();  
        lcd.setCursor(0,1);  
        lcd.print("  Alarma activa");  
        lcd.setCursor(7,2);  
        lcd.print(c);  
        lcd.setCursor(9,2);  
        lcd.print(":");  
        lcd.setCursor(10,2);  
        lcd.print(b);  
        analogWrite(A0,255);  
        delay(5000);  
        analogWrite(A0,0);  
        goto fin;  
    }  
    delay(10);  
  
    hora[3]=EEPROM.read(12)*10+EEPROM.read(13);  
    minutos[3]=EEPROM.read(14)*10+EEPROM.read(15);  
    if(hora[3]==c&&minutos[3]==b){  
        lcd.clear();  
        lcd.setCursor(0,1);  
        lcd.print("  Alarma activa");  
        lcd.setCursor(7,2);  
        lcd.print(c);  
        lcd.setCursor(9,2);  
        lcd.print(":");  
        lcd.setCursor(10,2);  
        lcd.print(b);  
        analogWrite(A0,255);  
        delay(5000);  
        analogWrite(A0,0);  
        goto fin;  
    }  
    delay(10);  
  
    hora[4]=EEPROM.read(16)*10+EEPROM.read(17);  
    minutos[4]=EEPROM.read(18)*10+EEPROM.read(19);  
    if(hora[4]==c&&minutos[4]==b){  
        lcd.clear();  
        lcd.setCursor(0,1);  
        lcd.print("  Alarma activa");  
        lcd.setCursor(7,2);  
        lcd.print(c);  
        lcd.setCursor(9,2);  
        lcd.print(":");  
        lcd.setCursor(10,2);  
        lcd.print(b);  
        analogWrite(A0,255);  
        delay(5000);  
        analogWrite(A0,0);  
        goto fin;  
    }  
    delay(10);  
  
    hora[5]=EEPROM.read(20)*10+EEPROM.read(21);  
    minutos[5]=EEPROM.read(22)*10+EEPROM.read(23);  
    if(hora[5]==c&&minutos[5]==b){  
        lcd.clear();  
        lcd.setCursor(0,1);  
        lcd.print("  Alarma activa");  
        lcd.setCursor(7,2);  
        lcd.print(c);  
        lcd.setCursor(9,2);  
        lcd.print(":");  
        lcd.setCursor(10,2);
```

```
        lcd.print(b);
        analogWrite(A0,255);
        delay(5000);
        analogWrite(A0,0);
        goto fin;
    }
    delay(10);

    hora[6]=EEPROM.read(24)*10+EEPROM.read(25);
    minutos[6]=EEPROM.read(26)*10+EEPROM.read(27);
    if(hora[6]==c&&minutos[6]==b){
        lcd.clear();
        lcd.setCursor(0,1);
        lcd.print(" Alarma activa");
        lcd.setCursor(7,2);
        lcd.print(c);
        lcd.setCursor(9,2);
        lcd.print(":");
        lcd.setCursor(10,2);
        lcd.print(b);
        analogWrite(A0,255);
        delay(5000);
        analogWrite(A0,0);
        goto fin;
    }
    delay(10);

    hora[7]=EEPROM.read(28)*10+EEPROM.read(29);
    minutos[7]=EEPROM.read(30)*10+EEPROM.read(31);
    if(hora[7]==c&&minutos[7]==b){
        lcd.clear();
        lcd.setCursor(0,1);
        lcd.print(" Alarma activa");
        lcd.setCursor(7,2);
        lcd.print(c);
        lcd.setCursor(9,2);
        lcd.print(":");
        lcd.setCursor(10,2);
        lcd.print(b);
        analogWrite(A0,255);
        delay(5000);
        analogWrite(A0,0);
        goto fin;
    }
    delay(10);

    hora[8]=EEPROM.read(32)*10+EEPROM.read(33);
    minutos[8]=EEPROM.read(34)*10+EEPROM.read(35);
    if(hora[8]==c&&minutos[8]==b){
        lcd.clear();
        lcd.setCursor(0,1);
        lcd.print(" Alarma activa");
        lcd.setCursor(7,2);
        lcd.print(c);
        lcd.setCursor(9,2);
        lcd.print(":");
        lcd.setCursor(10,2);
        lcd.print(b);
        analogWrite(A0,255);
        delay(5000);
        analogWrite(A0,0);
        goto fin;
    }
}

fin:
lcd.clear();
delay(1);
}
```

## Anexo 10. Código del form en Visual Basic 6 para aplicación de PC.

```
ProyectoTesis - 1

Private Sub Form_Load()
Dim i As Integer

Pantalla.Enabled = False
Actualizartiempo.Enabled = True
Conectado.Value = 0
Timer2.Enabled = True
For i = 0 To 8
Alarma(i).Value = False
Next i

End Sub

Private Sub Conectar_Click()

Winsock1.Close

Winsock1.RemoteHost = IPDirec.Text
Winsock1.RemotePort = Puerto.Text

Winsock1.Connect

End Sub

Private Sub Winsock1_Connect()

'desplegamos un mensaje en la ventana
Conectar.Visible = False
Conectar.Enabled = False
Desconectar.Visible = True
Conectado.Caption = "Conectado"
Conectado.Value = 1

End Sub

Private Sub IPDirec_KeyPress(KeyAscii As Integer)

If (KeyAscii >= 97) And (KeyAscii < 122) Or (KeyAscii >= 65) And (KeyAscii < 90) Then
MsgBox "No se admite letras" & vbCrLf & "Formato IP:" & vbCrLf & "192.168.0.0 a 192.168.255.255"
KeyAscii = 0
End If

End Sub

Private Sub Puerto_KeyPress(KeyAscii As Integer)

If (KeyAscii = 8) Or (KeyAscii > 47) And (KeyAscii < 58) Then 'usar isnumeric
KeyAscii = KeyAscii
Exit Sub
Else
MsgBox "Solo Ingrese Números " & vbCrLf & "Puertos de 0 a 65535"
KeyAscii = 0
Puerto.Text = ""
Exit Sub
End If

End Sub

Private Sub Desconectar_click()

Conectar.Visible = True
Conectar.Enabled = True
Desconectar.Visible = False
Conectado.Caption = "Desconectado"
Conectado.Value = 0
Winsock1.Close

End Sub

Private Sub Winsock1_Close()

Conectado.Caption = "Desconectado"
Conectado.Value = 0
```

```

ProyectoTesis - 2

    Desconectar.Visible = False
    Winsock1.Close
    Conectar.Enabled = True
    Conectar.Visible = True

End Sub

Private Sub LeerAlarma_Click()
Dim r As Integer

r = texto.Caption
Winsock1.SendData ("leer" & "" & r & vbCrLf) 'para la lectura de las alarmas
Pantalla.Text = "Hora de alarma " & r & ": "
Timer1.Enabled = True
Timer2.Enabled = False
For i = 0 To 8
Alarma(i).Value = False
Next i

End Sub

Private Sub ConfAlarma_Click()
Dim i As Integer
Dim f As String
Dim cadena As String

f = texto.Caption
cadena = "registro" & "" & f & "" & Horas.Text & "" & Minuto.Text & vbCrLf 'para configurar las alar
mas
Winsock1.SendData (cadena)
Timer2.Enabled = False
Timer1.Enabled = True
Pantalla.Text = "Se Envio al registro: " & f & vbCrLf & _
"Hora: " & Horas.Text & ":" & Minuto.Text & vbCrLf

Horas.Text = "HH"
Minuto.Text = "MM"
For i = 0 To 8
Alarma(i).Value = False
Next i

End Sub

Private Sub Botonpanico_Click()

Winsock1.SendData ("activar")

End Sub

Private Sub Actualizartiempo_Click()

fecha = Day(Date) & "" & Month(Date) & "" & Year(Date)
dia = Weekday(Date)
Hora = Hour(Time) & "" & Minute(Time) & "" & Second(Time)
Winsock1.SendData ("actualizar" & "" & fecha & "" & dia & "" & Hora & vbCrLf)

End Sub

Private Sub Alarma_Click(index As Integer)

texto.Caption = index + 1

End Sub

Private Sub Winsock1_DataArrival(ByVal total As Long)
Dim valor As String
'Si existen datos los retira
Winsock1.GetData valor
Timer2.Enabled = False
Timer1.Enabled = True
'Actualiza la ventana con la información recibida
Pantalla.SetStart = Len(Pantalla.Text)
Pantalla.Text = Pantalla.Text & valor

```

```

ProyectoTesis - 3

End Sub

Private Sub timer1_timer()

Timer2.Enabled = True
texto.Caption = "0"

End Sub

Private Sub Timer2_Timer()
Dim cadenal As String
Dim cadena2 As String

Timer1.Enabled = False
fecha = Day(Date) & "/" & Month(Date) & "/" & Year(Date)
dia = Weekday(Date)
Hora = Hour(Time) & ":" & Minute(Time) & ":" & Second(Time)
cadenal = Format(fecha, "long date")
cadena2 = Format(Hora, "long time")
Pantalla.Text = vbCrLf & cadenal & vbCrLf & cadena2 & vbCrLf

End Sub

Private Sub Minuto_KeyDown(KeyCode As Integer, Shift As Integer)

KeyCode = 0
Shift = 0

End Sub

Private Sub Minuto_KeyPress(KeyAscii As Integer)

KeyAscii = 0

End Sub

Private Sub Hora_KeyDown(KeyCode As Integer, Shift As Integer)

KeyCode = 0
Shift = 0

End Sub

Private Sub Hora_KeyPress(KeyAscii As Integer)

KeyAscii = 0

End Sub

```

The image shows a Windows application window titled "Universidad Nacional de Ingeniería". The window contains a form with the following elements:

- IP Address:** A text box containing "192.168.4.1" and a "Desconectar" button with a clock icon.
- Port:** A text box containing "1000" and a "Desconectado" checkbox with a clock icon.
- Buttons:** A vertical stack of buttons: "Actualizar", "Conf. Alarma", "Leer Alarma", and "Activar".
- Alarms:** A grid of radio buttons labeled "Alarma 1" through "Alarma 9".
- Time Selection:** Labels "Hora" and "Minuto" followed by dropdown menus for "HH" and "MM".
- Print Icon:** A small printer icon at the bottom right of the form.

*Diseño del form para la interfaz de PC.*

Anexo 11. Código de bloques de aplicación para Smartphone con sistema Android.

```

inicializar global datos como crear una lista vacía
inicializar global Address como ""
inicializar global Port como ""

cuando Conectar .Clic
ejecutar
    poner global datos a crear una lista vacía
    poner global Address a Address . Texto
    poner global Port a Puerto . Texto
    añadir elementos a la lista lista
        tomar global datos
        tomar global Address
        tomar global Port
    abre otra pantalla con un valor inicial Nombre de la pantalla " Principal "
    Valor inicial tomar global datos

cuando Screen1 .BotónAtrás
ejecutar cerrar la aplicación
    
```

```

inicializar global Address como ""
inicializar global Port como ""
inicializar global Datos como crear una lista vacía

cuando Principal .BotónAtrás
ejecutar cerrar la aplicación

cuando Principal .Inicializar
ejecutar
    poner global Datos a tomar el valor inicial
    poner global Address a seleccionar elemento de la lista tomar global Datos
    índice 1
    poner global Port a seleccionar elemento de la lista tomar global Datos
    índice 2
    poner ClientSocketAI2Ext1 . ServerAddress como tomar global Address
    poner ClientSocketAI2Ext1 . ServerPort como tomar global Port

cuando LeerAlarma .Clic
ejecutar abre otra pantalla con un valor inicial Nombre de la pantalla " LeerAlarma "
    Valor inicial tomar global Datos

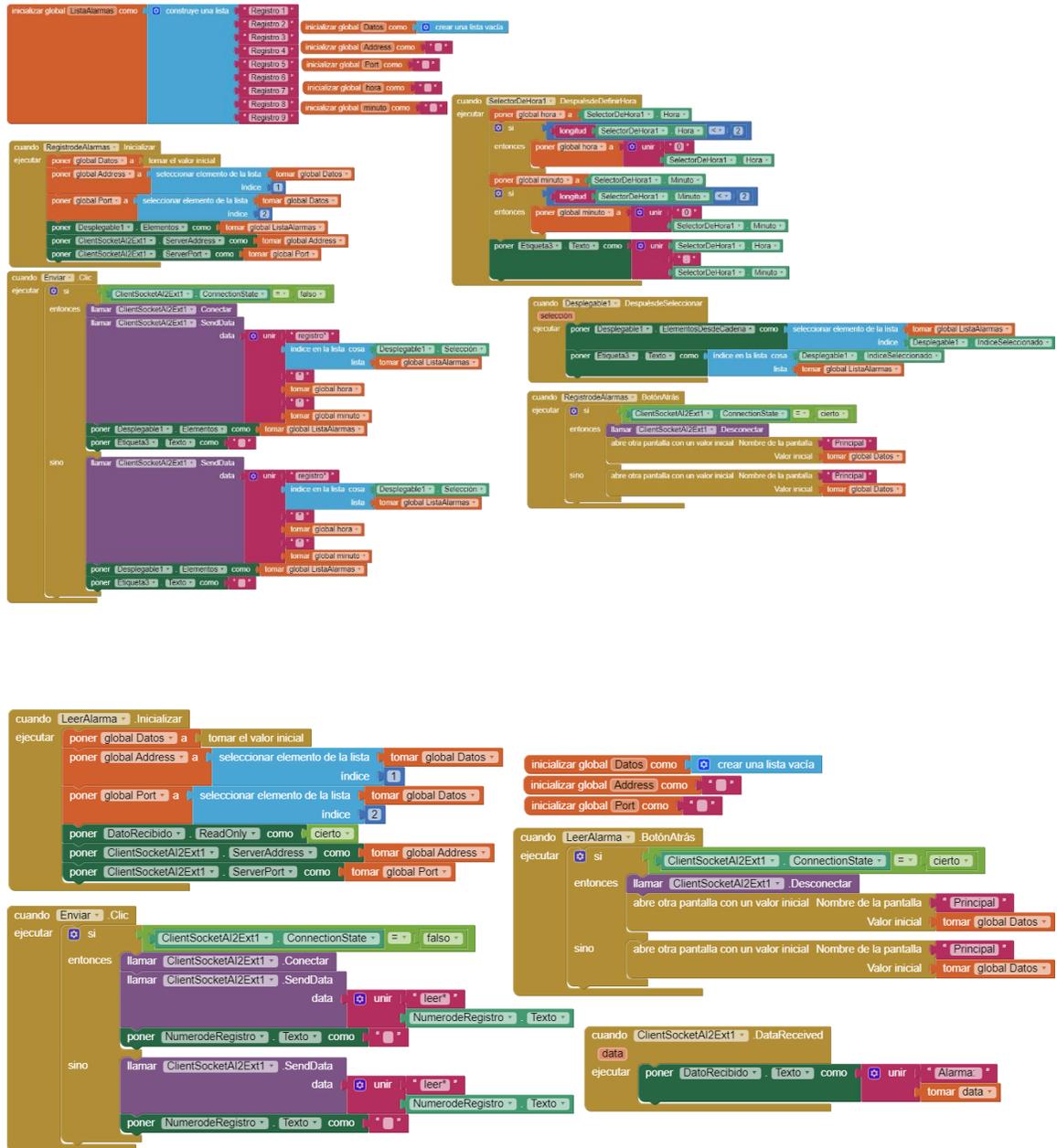
cuando RegistroAlarma .Clic
ejecutar abre otra pantalla con un valor inicial Nombre de la pantalla " RegistrodeAlarmas "
    Valor inicial tomar global Datos

cuando Activar .Clic
ejecutar
    si ClientSocketAI2Ext1 . ConnectionState = falso
    entonces
        llamar ClientSocketAI2Ext1 . Conectar
        llamar ClientSocketAI2Ext1 . SendData
        data activar
    sino
        llamar ClientSocketAI2Ext1 . SendData
        data activar
    
```

## Anexo 11. Código de bloques de aplicación para Smartphone... (Continuación).



## Anexo 11. Código de bloques de aplicación para Smartphone... (Continuación)



Anexo 12. Diseño de las pantallas para la interfaz de Smartphone.

Universidad Nacional de Ingeniería

IP:

Puerto:

**Conectar**

Universidad Nacional de Ingeniería

**Timbre Automático**

**Ajustar Fecha y Hora**

**Registrar Alarma**

**Mostrar Hora de Alarma**

**Activar Timbre**



Universidad Nacional De Ingeniería

**Selector de alarmas**

Registro 1 ▾

**Configuración De Hora**

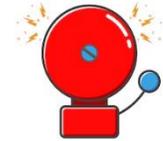
**Selector De Hora**

**Enviar**

LeerAlarma

**Enviar**

Anexo 13. Manual de usuario.



## Manual de instrucciones

*Elaborado por:*

*Br. Sarahi Sandoval Sandoval*

*Br. Cristian Solórzano Cano*

*Lea este manual detenidamente antes de manipular el timbre y consérvelo para consultarlo en el futuro.*

<b>Contenido.....</b>	<b>1</b>
<b>1. INSTALACIÓN DEL CONTROLADOR.....</b>	<b>3</b>
<b>2. CONFIGURACIÓN HACIENDO USO DEL TECLADO DEL CONTROLADOR. ....</b>	<b>5</b>
<b>2.1. ACTUALIZACIÓN DE HORA Y FECHA MEDIANTE TECLADO. ....</b>	<b>6</b>
<b>2.2. CONFIGURACIÓN FECHA MEDIANTE TECLADO.....</b>	<b>8</b>
<b>2.3. REGISTRO DE HORA Y MINUTOS EN QUE SE ACTIVARÁ LA ALARMA MEDIANTE TECLADO.....</b>	<b>10</b>
<b>2.4. LECTURA DE LOS REGISTRO DE ALARMAR PARA ACTIVACIÓN MEDIANTE TECLADO.....</b>	<b>6</b>
<b>2.5. BORRAR REGISTRO DE ALARMA MEDIANTE TECLADO. ....</b>	<b>11</b>
<b>3. CONFIGURACIÓN MEDIANTE PC.....</b>	<b>12</b>
<b>3.1. ACTUALIZAR HORA Y FECHA DESDE EL PC. ....</b>	<b>13</b>
<b>3.2. REGISTRO DE HORAS Y MINUTOS PARA ALARMA DESDE PC.....</b>	<b>15</b>
<b>3.3. LECTURA DE REGISTRO DE ALARMA DESDE LA PC.....</b>	<b>15</b>
<b>4. CONFIGURACIÓN MEDIANTE SMARTPHONE CON SISTEMA ANDROID. ....</b>	<b>17</b>

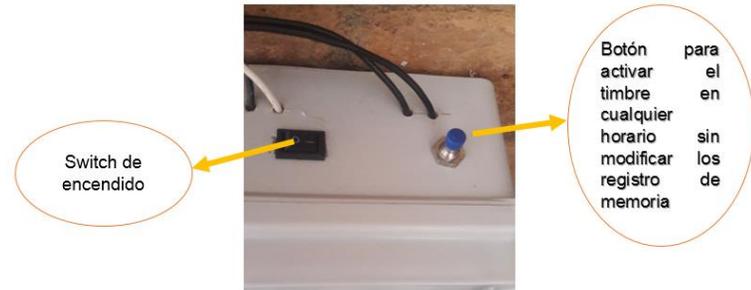
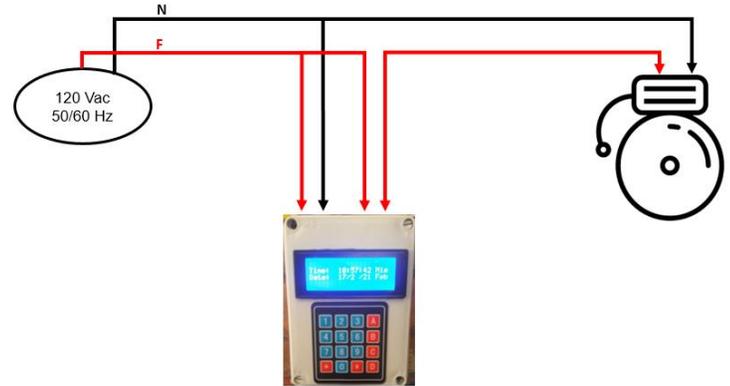
<b>4.1. ACTUALIZAR HORA Y FECHA DESDE SMARTPHONE.. ....</b>	<b>10</b>
<b>4.2. REGISTRO DE HORAS Y MINUTOS PARA ALARMA DESDE SMARTPHONE. ....</b>	<b>10</b>
<b>4.3. LECTURA DE REGISTRO DE ALARMA DESDE SMARTPHONE. ....</b>	<b>22</b>

# Manual de timbre automatizado.

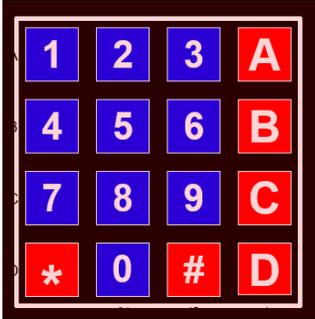
## 1. Instalación del controlador.

Para una correcta instalación del controlador electrónico para el timbre escolar deberá establecer un lugar donde colocara la caja que contiene a nuestro controlador, deber ser un lugar seguro, donde no obstruya el desarrollo normal de las actividades, totalmente seco y fuera del alcance de los menores.

Fijada la cajita se procede a la conexión del timbre y la fuente de suministro eléctrico, para tal caso el suministro eléctrico comercial (120Vac, 50/60 Hz).



## 2. Configuración haciendo uso del teclado del controlador.



Para acceder a las funciones del menú principal se debe presionar cualquiera de las teclas A, B, C y D.

- Tecla A: Actualizar hora y fecha.
- Tecla B: Registrar hora de Alarma.
- Tecla C: Leer un registro de alarma.
- Tecla D: Borrar registro de alarma.

## 2.1. Actualización de hora y fecha mediante teclado.

En esta función el usuario tiene la opción de actualizar la fecha y la hora siguiendo las indicaciones a continuación.

Al presionar la tecla A del teclado el control le mostrará la siguiente pantalla, en la que solicita ingresar los minutos para la hora actual.

```
Ingrese Los Min  
formato(00-59): 25  
  
*=Salir #=Siiguiente
```

```
Ingrese La Hora  
formato(00-23): 12  
  
*=Atras #=Siiguiente
```

Si desea volver a la pantalla de inicio presione (\*), sino después de ingresar el dato, para continuar

con la configuración presione (#). Luego de eso deberá ingresar la hora actual que esta configurado en formato de 24 horas, es decir, después de las 12 md, las horas se deben ingresar como 13 para las 1 de la tarde, 14 paras las 2 de la tarde y así sucesivamente.

Una vez ingresada la hora le pedirá al usuario ingresar el día actual de la semana, los cuales se deberán ingresar de la siguiente manera:

El día 1 corresponde la día Domingo.

El día 2 corresponde al día Lunes.

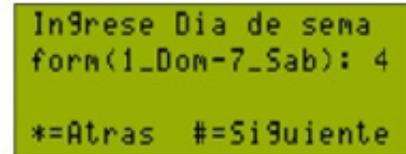
El día 3 corresponde al día Martes.

El día 4 corresponde al día Miércoles.

El día 5 corresponde al día Jueves.

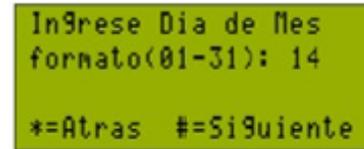
El día 6 corresponde al día Viernes.

El día 7 corresponde al día Sábado.



Terminado este paso, presione en siguiente para configurar la fecha y los datos se vaya actualizando.

## 2.2. Configuración fecha mediante teclado.



Se irá paso a paso ingresando día del mes, el mes y el año. Recordar dar siguiente (#) en cada pantalla mostrada para ir avanzando con la actualización.

En el caso de haber equivocación en el ingreso de los números presionar \* (atrás) y se borra automáticamente, luego presionar siguiente para ingresar nuevamente los datos.

Siguiendo con el proceso ahora se configurará el día del mes y año actual.

```
Ingrese # de Mes
formato(01-12): 09
*=Atras  #=SiGuiente
```

```
Ingrese Año
formato(00-99): 20
*=Atras  #=SiGuiente
```

Una vez terminado este proceso, la configuración quedará lista como se muestra en la siguiente pantalla.

```
Time: 12:25:17 Nie
Date: 14/9 /20 Sep
```

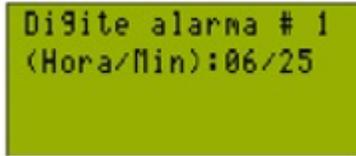
### 2.3. Registro de hora y minutos en que se activará la alarma mediante teclado.

Para acceder a esta función basta con presionar la tecla B en el teclado del dispositivo.

El primer paso será seleccionar el número del registro que desea configurar, para tal propósito el sistema cuenta con 9 registros. Para configurarlo, el usuario deberá escoger un registro del 1 al 9 para ser actualizado.

```
Digite # de alarma
a actualizar(1-9):
Salir= '*'
```

Una vez ingrese el registro de alarma que se desea actualizar, el siguiente paso será configurar la hora en que se activará.

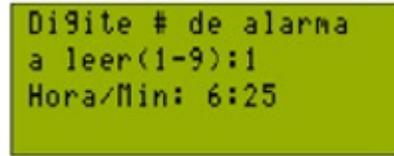


Esto indica que la alarma 1, se activará automáticamente a las 6:25 am. Para realizar activaciones en horarios después de las 12 del medio día se deberá ingresar la hora en formato de 24 horas.

#### **2.4. Lectura de los registros de alarmas para activación mediante teclado.**

Para realizar una lectura de cualquiera de los 9 registros de alarmas con los que cuenta el sistema, el usuario deberá presionar la tecla C.

Luego se le solicitará ingresar el número del registro al que quiere leer. Luego el sistema procederá a mostrar los datos almacenados en el registro.



#### **2.5. Borrar registro de alarma mediante teclado.**

Esta función solo puede ser usada mediante el teclado con el que dispone el controlador. Para acceder a esta función el usuario deberá presionar la tecla D.

Dentro de la función, el controlador mostrará la siguiente pantalla, en donde solicita al usuario digite el número del registro que desea borrar.

```
Seleccione registro
a borrar (1-9):

"*"=Salir
```

Una vez ingreso el número, el controlador procede a realizar dicho borrado, a su vez muestra los datos almacenados al momento de realizar el borrado de dicho registro.

```
Registro # 1: 6:25
Borrando*****
Registro borrado
```

### 3. Configuración mediante Pc.

Para realizar configuración mediante una PC, supondremos que dicha PC cuenta con su tarjeta LAN para conectarse a una red WIFI.

El controlador del timbre para esto cuenta con su propia red, que por defecto tiene por nombre **Farylink\_AEDE08**. Red a la cual se debe conectar nuestra PC. Una vez conectados abrir el ejecutable que fue proporcionado junto con el controlador.



Con el ejecutable abierto, el primer paso para acceder a las funciones del controlador será establecer comunicación con el controlador y para ello solo basta con ingresar en los campo de IP

Address y Port una dirección IP y un puerto que por defecto el ejecutable ya los trae escritos en dichos campos. La dirección IP por defecto es la 192.168.4.1 y el puerto es el 1000.

Llenados estos dos campos deberá presionar con el cursor sobre el botón conectar, una vez presionado se confirmará la conexión con un check sobre el caption desconectado.

### 3.1. Actualizar Hora y Fecha desde el PC.

Una vez conectado con el controlador, el usuario solo deberá dar click sobre el botón Actualizar, el ejecutable toma la hora y la fecha del equipo y lo transmite al controlador del timbre.

### 3.2. Registro de horas y minutos para alarma desde PC.

Para actualizar un registro de activación del timbre es necesario que el usuario selecciones dando

click sobre cualquiera de la lista de alarmas con las que cuenta el ejecutable que en total serán 9.

Luego de eso, seleccionar en los selectores etiquetados como hora y minuto, la hora y minuto en que desea la activación del timbre.



The screenshot shows a configuration interface for alarms. It features a grid of nine radio buttons labeled 'Alarma 1' through 'Alarma 9'. To the right of these buttons are two time selection controls: a 'Hora' (Hour) dropdown menu and a 'Minuto' (Minute) dropdown menu. The 'Minuto' dropdown is currently open, showing a digital clock display with '11:11' and a blue background.

Realizado todos los pasos, solo basta con dar click sobre el botón etiquetado como **conf. Alarma**.

### 3.3. Lectura de registro de alarma desde la PC.

Si el usuario desea realizar una lectura desde el pc de cualquiera de los 9 registro con los que cuenta el sistema deberá seleccionar el registro que

quiere leer dando click en las opciones de la lista de alarmas.



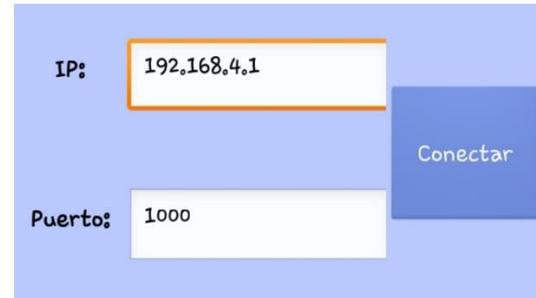
Seleccionado el registro, deberá dar click sobre el botón etiquetado como **leer alarma** y en la pantalla se mostrará el dato solicitado al controlador.

Para activar el timbre el cualquier momento sin tener que acceder a los registros de alarma, el ejecutable cuenta con un botón nombrado como activar, para ello dar click sobre este botón y automáticamente se activará el timbre, sin modificar ningún dato ingresados a los registros.

#### 4. Configuración mediante Smartphone con sistema Android.

Al igual que para la configuración mediante PC se deberá conectar a la red creada por el controlador, que por defecto tiene el nombre de **Farylink-AEDE08**.

Al abrir la App le pedirá llenar dos campos que serán la IP para conectarse con el controlador y un número de puerto para la transmisión de datos. Por defecto ya vienen escritos en los campos la IP y el puerto, solo bastará que el usuario presione el botón **conectar**.



Una vez conectado con el controlador el usuario dispone de los mismos botones que para la configuración mediante PC.



#### 4.1. Actualizar Hora y Fecha desde Smartphone.

Después de conectarse con la red del controlador, el usuario solo deberá dar click sobre el botón Actualizar, el ejecutable toma la hora y la fecha del Smartphone y lo transmite al controlador del timbre.

#### 4.2. Registro de horas y minutos para alarma desde Smartphone.

Para registrar o actualizar la hora de activación del timbre mediante su Smartphone deberá dar click sobre el botón registrar alarma.

Dentro de la función, el usuario tendrá que seleccionar mediante un selector el registro que desea actualizar.



Luego la interfaz dispone de un selector para la hora y otro para los minutos.

Cuando haya seleccionado el registro, la hora y minuto de activación solo bastará con dar click en el botón enviar para que el Smartphone se encargue de hacer llegar estos datos hasta el controlador.

### 4.3. Lectura de registro de alarma desde Smartphone.

Para realizar una lectura de los registros de activación almacenados en el controlador, el usuario de dar click sobre el botón mostrar hora de alarma.



Dentro de la pantalla, dispondrá de un campo de texto para escribir el registro que desea leer; con el registro ya escrito solo basta con dar click sobre el botón enviar y esperar un tiempo para recibir el dato solicitado.