

# **ESCUELA POLITÉCNICA NACIONAL**

## **ESCUELA DE FORMACIÓN DE TECNÓLOGOS**

### **DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO PARA UN SISTEMA ELECTRÓNICO TEMPORIZADO EN PUERTAS Y VENTANAS, EMPLEANDO MÓDULOS ARDUINO, PARA EL RESTAURANTE “EL PAILÓN DE MUSHUÑAN”**

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGA EN ELECTRÓNICA Y TELECOMUNICACIONES**

**JENNY MARIELA TOPÓN GUALLICHICO**

**sajeda\_23@hotmail.com**

**DIRECTOR: ING. FABIO MATÍAS GONZÁLEZ GONZÁLEZ MSc.**

**fabio.gonzalez@epn.edu.ec**

**Quito, Agosto 2017**

## DECLARACIÓN

Yo TOPÓN GUALLICHICO JENNY MARIELA, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normativa institucional vigente.

---

Jenny Mariela Topón Guallichico

## **CERTIFICACIÓN**

Certifico que el presente trabajo fue desarrollado por Jenny Mariela Topón Guallichico, bajo mi supervisión.

---

**Ing. Fabio Matías González González MSc.**

**DIRECTOR DE PROYECTO**

## **AGRADECIMIENTO**

Agradezco a Dios por permitirme culminar esta etapa de mi vida, que a pesar de las diversas dificultades ya logre culminar con mi trabajo.

A mis padres Wilson y Teresa por haberme forjado como la persona que soy y jamás haber desmayado con su apoyo incondicional, para lograr mí meta.

A mis hermanos Santiago y David, por su apoyo en este proceso de mi vida, y por demostrarme que a pesar de las circunstancias siempre estaremos juntos.

A mi pequeño Daniel que me ha dado su comprensión y su alegría para no decaer y seguir adelante en la lucha ante las adversidades.

A mi angelita del cielo Annie, que siempre me dio sus bendiciones y perseverancia para lograr con éxito este proyecto.

A mi esposo Gustavo que con mucha paciencia supo apoyarme y colaborar en los momentos difíciles de mi trabajo.

Finalmente agradezco a mi tutor del proyecto el Ingeniero Fabio González, que creyó en mí y mis capacidades.

Jenny Mariela Topón Guallichico

## **DEDICATORIA**

El presente trabajo está dedicado a mi pequeño ángel aquí en la tierra Daniel y a mi pequeña Annie que siempre me cuida y bendice desde el cielo.

Jenny Topón

## RESUMEN

El presente documento trata de la elaboración de un prototipo de sistema de seguridad temporizado para puertas y ventanas, utilizando el módulo Arduino, dispositivos magnéticos y ultrasónicos; y marcado de clave por medio de un teclado matricial.

El proyecto se basa en implementar un mando central (Arduino y teclado matricial), para permitir al usuario que mediante el ingreso de una clave habilite y deshabilite el sistema de alarma y con ello el control de los dispositivos electrónicos comandados por el módulo Arduino, el cual se ha programado de manera que cuando los sensores magnéticos detecten la presencia de intrusión envíe un aviso sonoro y un aviso visual, hacia una pantalla, de que se ha abierto una puerta o ventana.

El informe escrito del proyecto está organizado en 4 capítulos:

En el capítulo 1 se realiza una investigación de los conceptos básicos de los componentes electrónicos y definiciones que intervendrán para la elaboración del prototipo propuesto.

En el capítulo 2 se realiza la explicación de los parámetros que se consideran para el diseño e implementación del prototipo en el modelo a escala de un restaurante.

En el capítulo 3 se describen las pruebas de funcionamiento y la depuración de errores que fue necesario realizar para obtener el resultado deseado en el sistema de seguridad.

En el capítulo 4 se indican las conclusiones y recomendaciones que se obtuvieron en el diseño e implementación del proyecto.

# ÍNDICE

ÍNDICE DE FIGURAS.....	viii
ÍNDICE DE TABLAS.....	ix
INTRODUCCIÓN.....	x
OBJETIVOS .....	xi
Objetivo General .....	xi
Objetivos Específicos .....	xi
CAPÍTULO 1.....	1
MARCO TEÓRICO .....	1
1.1.    Sistemas de Seguridad.....	1
1.1.1.    Funcionamiento de un Sistema de Seguridad.....	1
1.2.    Plataforma Arduino .....	2
1.2.1.    Microcontrolador .....	2
1.2.2.    Definición de Arduino.....	3
1.2.3.    Beneficios del uso de Arduino .....	3
1.2.4.    Tipos de Módulos Arduino .....	4
1.3.    Pantallas Táctiles.....	7
1.3.1.    Definición.....	7
1.3.2.    Funcionamiento .....	7
1.3.3.    Tipos de pantallas táctiles.....	8
1.4.    Sensores .....	10
1.4.1.    Definición.....	10
1.4.2.    Características de un sensor .....	10
1.4.3.    Clasificación de sensores .....	11
1.5.    Teclado Matricial .....	11
1.5.1.    Definición.....	11
1.5.2.    Funcionamiento .....	12
1.6.    Dispositivo de alerta .....	13
CAPÍTULO 2.....	14
DISEÑO E IMPLEMENTACIÓN DEL PROTOTIPO DEL SISTEMA DE SEGURIDAD .....	14
2.1.    Antecedentes.....	14
2.2.    Determinación de los puntos a ser considerados en el Sistema de Seguridad ..	14

2.2.1.	Descripción del Restaurante.....	14
2.2.2.	Estudio técnico para el sistema de seguridad electrónica.....	16
2.2.3.	Características de seguridad .....	17
2.3.1.	Sensores Magnéticos .....	18
2.3.2.	Sensores de ultrasonido HC-SR04 .....	19
2.3.3.	HMI Nextion Enhanced NX4024K032.....	21
2.3.4.	Controlador Arduino Mega 2560.....	23
2.3.5.	Lenguaje de programación Arduino.....	23
2.3.6.	Teclado Matricial 4x4.....	23
2.4.1.	Diagrama de bloques del hardware .....	25
2.4.2.	Programación de controlador Arduino .....	25
2.4.3.	Programación de HMI.....	29
2.4.4.	Diagrama de Conexiones .....	31
CAPÍTULO 3.....		36
PRUEBAS Y RESULTADOS .....		36
CAPÍTULO 4.....		41
CONCLUSIONES Y RECOMENDACIONES .....		41
4.1.	Conclusiones.....	41
4.2.	Recomendaciones.....	41
BIBLIOGRAFÍA.....		43
ANEXOS.....		45



## ÍNDICE DE FIGURAS

Figura 1.1 Estructura de un sistema de seguridad [1] .....	2
Figura 1.2 Placa módulo Arduino Duemilanove [6] .....	5
Figura 1.3. Placa módulo Arduino Nano [6].....	6
Figura 1.4. Placa módulo Arduino Fio [5] .....	6
Figura 1.5. Placa módulo Arduino Uno [6].....	6
Figura 1.6. Placa módulo Arduino Mega [6] .....	7
Figura 1.7. Funcionamiento de la pantalla táctil por infrarrojos [8] .....	8
Figura 1.8. Componentes pantalla táctil resistiva [8] .....	9
Figura 1.9. Pantalla táctil capacitiva [7].....	9
Figura 1.10. Funcionamiento pantalla táctil de onda acústica superficial [8] .....	10
Figura 1.11. Teclado matricial [10] .....	12
Figura 1.12. Teclado matricial 3x4 conectado a los pines del puerto B. [10] .....	13
Figura 2.1. Áreas del Restaurante .....	15
Figura 2.2. Plano de la ubicación de los sensores .....	18
Figura 2.3. Sensor magnético [13] .....	19
Figura 2.4. Funcionamiento sensor de ultrasonido HC-SR04 [15].....	20
Figura 2.5. Conexión del sensor ultrasonido con Arduino Uno y Mega [14] .....	21
Figura 2.6. HMI Nextion NX4024K032 [16] .....	22
Figura 2.7. Partes de Arduino Mega 2560 [9].....	23
Figura 2.8. Conexión teclado matricial 4x4 con Arduino. [17].....	24
Figura 2.9. Diagrama de bloques del sistema .....	25
Figura 2.10. Creación de un nuevo Sketch en Arduino. ....	26
Figura 2.11. Selección del tipo de tarjeta a utilizar. ....	27
Figura 2.12. Diagrama de flujo del programa. ....	28
Figura 2.13. Selección HMI en Nextion Editor.....	29
Figura 2.14. Selección de orientación HMI.....	30
Figura 2.15. Entorno de la programación del HMI en Nextion. ....	30
Figura 2.16. Menú de páginas y toolbox en Nextion.....	31
Figura 2.17. Diagrama de conexiones .....	32
Figura 2.18. Colocación del imán en puertas y ventanas. ....	33
Figura 2.19. Fijación del sensor magnético.....	33
Figura 2.20. Instalación de sensor ultrasónico .....	34
Figura 2.21. Instalación del buzzer .....	34
Figura 2.22. Cableado de sensores .....	35
Figura 2.23. Instalación Mando Central.....	35
Figura 3.1. Pantalla visualiza mensaje cuando una puerta y ventana están abiertas. ....	37
Figura 3.2. Pantalla visualiza solo PUERTAS y VENTANAS sin mensaje de alerta. ....	38
Figura 3.3. Lectura sensor ultrasónico .....	39
Figura 3.4. Encendido del led de la placa Arduino. ....	39
Figura 3.5. Sistema activado, led rojo encendido.....	40
Figura 3.6. Sistema desactivado, led verde encendido. ....	40

## ÍNDICE DE TABLAS

Tabla 2.1. Determinación de las partes físicas de cada área. ....	16
Tabla 2.2. Distribución por áreas de los sensores.....	17

## **INTRODUCCIÓN**

En la actualidad, obtener un sistema de seguridad es muy importante y necesario para pequeños locales comerciales de los alrededores de la parroquia de Sangolquí. En ocasiones las personas olvidan asegurar correctamente sus puertas y ventanas por la prisa que llevan de realizar sus actividades diarias, siendo éste un factor para que los ladrones ingresen por una ventana o puerta que no estaba asegurada.

Haciendo referencia a la información antes expuesta, en la actualidad los locales comerciales pequeños que están ubicados en los barrios aledaños a la ciudad de Sangolquí, son muy vulnerables ante los robos. Por ello, se plantea el diseño e implementación de un prototipo para sistema de alarma temporizado en puertas y ventanas empleando módulos Arduino, mismos que se interconectarán con sensores magnéticos y de ultrasonido.

Este sistema de alarma permitirá de una u otra forma brindar seguridad y tranquilidad a las personas, no solo porque sus negocios van a estar protegidos, sino también porque el costo de inversión será accesible. En comparación con diseños de alta tecnificación, mismos que tienen un alto costo, y han ocasionado que locales pequeños o que surgen recién no puedan contar con un sistema de seguridad actual, moderno, de bajo costo y fácil manejo.

# OBJETIVOS

## Objetivo General

Implementar un prototipo de un sistema de seguridad electrónico temporizado para puertas y ventanas empleando módulos Arduino, interconectado con sensores de ultrasonido y magnéticos.

## Objetivos Específicos

- Analizar el funcionamiento y programación de hardware y software de la tecnología Arduino.
- Realizar un análisis de los elementos y dispositivos electrónicos a emplearse en el diseño del sistema de seguridad y que sean compatibles con los módulos Arduino.
- Definir los puntos estratégicos que requerirán la utilización de sensores de ultrasonido o magnéticos.
- Diseñar el diagrama circuital con todos los componentes seleccionados como módulos Arduino, sensores de ultrasonido y magnéticos.
- Desarrollar la programación en el software de Arduino para verificar el funcionamiento entre los módulos y los componentes que actúan en el sistema de seguridad.
- Realizar las pruebas necesarias del diseño elaborado en el modelo a escala del restaurante.

# CAPÍTULO 1

## MARCO TEÓRICO

### 1.1. Sistemas de Seguridad

Un sistema de seguridad no es más que un elemento preventivo, ya que no evitará una situación anormal pero sí advierten de ella. Estos sistemas evitan grandes pérdidas y reducen el tiempo de ejecución de las acciones a tomar.

Todo sistema de seguridad debe emitir una alarma a las personas responsables de atender a las situaciones de dicha alerta con el fin de prevenir mayores problemas y pérdidas económicas. En el caso de un sistema instalado en un negocio o inmueble, esta alarma deberá dar aviso al usuario, central de monitoreo o hacia donde esté canalizada la alerta.

Un sistema de seguridad para el hogar o pequeños negocios se compone de varios dispositivos electrónicos (sensores) conectados a un panel de control y éste a su vez estará conectado a una central de monitoreo.

En general, a un sistema de alarma se le denomina, como un conjunto de elementos e instalaciones fundamentales que provean a las personas y a sus bienes materiales protección en caso de un suceso inesperado como robo, sabotaje, incendio, etc.

#### 1.1.1. Funcionamiento de un Sistema de Seguridad

En caso de que un intruso irrumpa en el inmueble, ya sea en el hogar o negocio, el ladrón activará la alarma al momento de abrir una puerta o ventana donde se hallen instalados los sensores de contacto o mediante la activación de los sensores de movimiento. Si dichos inmuebles no cuentan con protección externa en las ventanas (enrejado) se conectan los sensores de ruptura de cristal, estos sensores son activados cuando se rompe el cristal de una ventana o puerta para entrar al inmueble.

Cuando uno o varios de los sensores con los que cuenta el sistema de alarma son activados, harán sonar el dispositivo de la sirena, el ruido que ésta produce hace que el intruso renuncie a seguir con el robo o al menos se tomará poco tiempo dentro del inmueble provocando menos pérdidas materiales.

En la Figura 1.1, se muestra como todos los dispositivos de la alarma son conectados al panel de control ya sea por medio de cables o en forma inalámbrica. En el caso de redes cableadas, generalmente se utilizan dos conductores para alimentación de 12 V y dos conductores para las señales de circuito serie NC (Normalmente Cerrado). El panel de control se encarga de controlar el funcionamiento general del sistema de alarma, tomando información del estado de los distintos dispositivos y accionando eventualmente los sistemas de aviso de la presencia de intrusos en el área protegida. Cabe mencionar que

este sistema de alarma cuenta con una batería interna que sirve para proveer un sistema de alimentación eléctrica ininterrumpida, de manera que ante la falta del suministro eléctrico (normal o provocado por un ladrón), el sistema de alarma contra intrusos continúe brindando protección en forma absolutamente normal.



Figura 1.1 Estructura de un sistema de seguridad [1]

## 1.2. Plataforma Arduino

### 1.2.1. Microcontrolador

Un microcontrolador ( $\mu C$ ), a veces llamado MCU (Micro Controller Unit), es básicamente un ordenador integrado en un solo chip, porque tiene en su interior las tres unidades básicas de un ordenador; es decir, memoria, CPU o microprocesador y periféricos de E/S. [2]

“La mayoría de los microcontroladores modernos no solo integran los elementos básicos mencionados anteriormente, sino que pueden llegar a tener muchas más partes: memoria RAM, memoria flash, temporizadores, decodificadores, conversores A/D y D/A, controladores (DMA, USB, Ethernet, PCI, etc.), UARTs (Universal Asynchronous Receiver-Transmitter).” [2]

Un microcontrolador puede hacer funciones tan simples como hacer titilar una luz o manejar máquinas y robots complejos. Por su versatilidad están presentes en muchos aparatos cotidianos: ordenadores, electrodomésticos, televisiones, reproductores

multimedia, automóviles, dispositivos de red, etc., incluso en las máquinas y robots industriales.

Existen multitud de fabricantes y tipos de microcontroladores, algunos históricos como el primero de ellos, el TMS 1000 de Texas Instruments. Otros tan populares como la familia PIC de la compañía Microchip. Pero los que más interesan en el presente proyecto son los de la compañía ATmel, ya que son los que integran la plataforma Arduino.

### **1.2.2. Definición de Arduino**

Arduino es una pequeña placa de microcontrolador con un puerto USB para conectar al ordenador y diversos zócalos de conexión que se pueden conectar mediante cableado a todo tipo de componentes electrónicos externos, como motores, relés, sensores de luz, diodos láser, altavoces, micrófonos, etc. Se puede alimentar mediante la conexión USB del ordenador o con una pila de 9 V. [3]

La placa se puede controlar directamente desde el ordenador o programarla con éste y posteriormente desconectarla para trabajar de forma autónoma. Puesto que Arduino es un diseño de hardware de código abierto, cualquier persona es libre de tomar los diseños y crear sus propios “clones” de Arduino y venderlos, por lo que el mercado de las placas es bastante competitivo.

Aunque Arduino es un diseño de código abierto para una placa interfaz de microcontrolador, en realidad es más que eso, ya que abarca tanto las herramientas de desarrollo de software que son necesarias para programar la placa Arduino, como la propia placa.

De hecho, existen diferentes diseños de placas Arduino destinados a diferentes tipos de aplicaciones. Todas las placas se pueden programar desde el mismo software de desarrollo de Arduino y, en general, los programas que funcionan en una placa, funcionan en todas.

### **1.2.3. Beneficios del uso de Arduino**

En el mercado existen diversos fabricantes de placas, que incorporan modelos diferentes de microcontroladores, mismos que son comparables y ofrecen una funcionalidad similar a la de las placas Arduino. Estas placas vienen acompañadas de un entorno de desarrollo agradable, cómodo y de un lenguaje de programación sencillo y completo. Por ello, la plataforma Arduino (hardware + software) presenta los siguientes beneficios:

- Es libre y extensible: significa que cualquier persona que tenga conocimiento básico en Arduino, pueda variar el desarrollo del diseño de hardware de las placas, así como en el entorno de desarrollo software y el propio lenguaje de programación, sin problemas. Esto permite que exista un rico “ecosistema” de extensiones, tanto de variantes de placas no oficiales como de librerías software de terceros, que pueden adaptarse mejor a las necesidades concretas.
- “Arduino tiene una gran comunidad: muchas personas lo utilizan, enriquecen la documentación y comparten continuamente sus ideas.” [4]

- “Su entorno de programación es multiplataforma: se puede instalar y ejecutar en sistemas Windows, Mac OS X y Linux. Esto no ocurre con el software de muchas otras placas.” [4]
- El lenguaje de programación es simple y claro: significa que es fácil de aprender y de utilizar, a la vez es flexible y completo para que los usuarios avanzados puedan aprovechar y expresar todas las posibilidades del hardware. Además, cuenta con diversos documentados, mismos que muestran ejemplos detallados, así como variedad de proyectos publicados en diferentes formatos.
- Las placas Arduino son baratas: actualmente dentro del mercado ecuatoriano ya es flexible encontrarlas en diferentes lugares que se dedican a la venta de dispositivos electrónicos. Su costo varía de 13 a 28 dólares, dependiendo del tipo de módulo.
- “Las placas Arduino son reutilizables y versátiles: reutilizables porque se puede aprovechar la misma placa para varios proyectos (ya que es muy fácil de desconectarla, reconectarla y reprogramarla), y versátiles porque las placas Arduino proveen varios tipos diferentes de entradas y salidas de datos, los cuales permiten capturar información de sensores y enviar señales a actuadores de múltiples formas.” [4]

#### 1.2.4. Tipos de Módulos Arduino

En la actualidad existen diferentes versiones de placas o módulos Arduino, a continuación se detallan algunas de las versiones de mayor relevancia dentro de la plataforma Arduino:

**Arduino Duemilanove:** “El Arduino Duemilanove es una placa con microcontrolador basada en el ATmega168 o el ATmega328, tiene 14 pines con entradas/salidas digitales (6 de las cuales pueden ser usadas como salidas PWM), 6 entradas analógicas, un cristal oscilador a 16Mhz, conexión USB, entrada de alimentación, una cabecera ISCP, y un botón de reset.” [2] [5]

Cuenta con las siguientes características:

- Alimentación vía conexión USB o con una fuente de alimentación externa, con un rango recomendado de 7 a 12 voltios.
- Tiene 32Kb de memoria flash para almacenar código (2Kb son usados para el arranque del sistema (bootloader).
- 2 Kb de memoria SRAM y 1Kb memoria EEPROM.
- Proporciona comunicación vía serie UART TTL (5V), disponible a través de los pines digitales 0(RX) y 1(TX).
- Un chip FTDI FT232RL integrado en la placa canaliza la comunicación serial a través del USB y los drivers FTDI (incluidos en el software de Arduino) proporcionan un puerto serie virtual en el ordenador.

En la Figura 1.2, se muestra el módulo Arduino Duemilanove, anteriormente descrito.



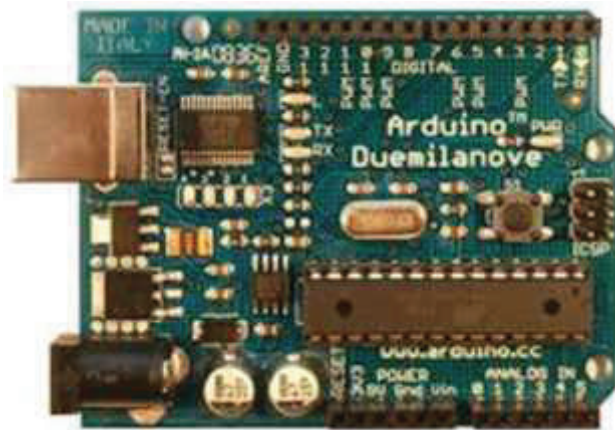


Figura 1.2 Placa módulo Arduino Duemilanove [6]

**Arduino Nano:** “El Arduino Nano es una pequeña y completa placa basada en el ATmega328 (Arduino Nano 3.0) o ATmega168 (Arduino Nano 2.x) que se usa conectándola a una protoboard. No posee conector para alimentación externa, y funciona con un cable USB Mini-B en lugar del cable estándar” [5]. La fuente de alimentación es seleccionada automáticamente a aquella con mayor tensión.

El chip FTDI FT232RL que posee el Nano solo es alimentado si la placa está siendo alimentada usando el cable USB. Como resultado, cuando se utiliza una fuente externa (no USB), la salida de 3.3V (la cual es proporcionada por el chip FTDI) no está disponible y los pines 1 y 0 parpadearán si los pines digitales 0 o 1 están a nivel alto.

El ATmega168 posee 16Kb de memoria flash para almacenar el código (de los cuales 2Kb son usados por el bootloader); el ATmega 328 posee 32Kb, (también con 2 Kb usados por el bootloader). El ATmega168 posee 1Kb de SRAM y 512 bytes de EEPROM; el ATmega328 posee 2 Kb de SRAM y 1Kb de EEPROM.

El Arduino Nano posee varios métodos para la comunicación con un PC, con otro Arduino u otros microcontroladores. El ATmega168 y el ATmega328 poseen un módulo UART que funciona con TTL (5V) lo que permite una comunicación vía serie, usando los pines 0 (RX) y 1 (TX). El chip FTDI FT232RL en la placa hace de puente a través del USB para la comunicación serial y los controladores FTDI incluidos con el software de Arduino, con lo proveen al PC de un puerto “com virtual” para el software en el PC. [5]

Se caracteriza por tener un microcontrolador ATmega 168, con 6 salidas digitales PWM, 6 entradas analógicas, con memoria Flash de 16Kb y 2Kb más reservados para el gestor de arranque, memoria SRAM de 1Kb y memoria EEPROM de 512 bytes, en la Figura 1.3, se visualiza la módulo Arduino Nano.

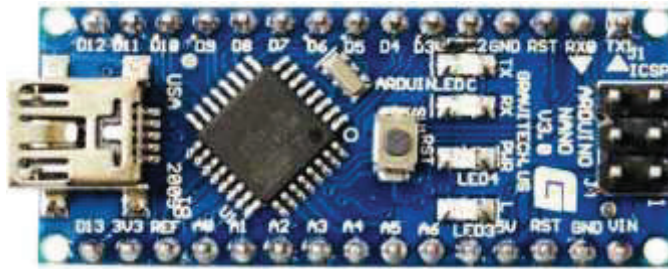


Figura 1.3. Placa módulo Arduino Nano [6]

**Arduino Fio:** Arduino basado en un microcontrolador ATmega328p. Trabaja a 8 MHz y 3.3V. Tiene 14 pines de entrada/salida digitales (6 PWM), 8 pines de entrada analógicas e integra tanto un conector para la batería y su correspondiente módulo de carga, como un slot para poder instalar un módulo de comunicaciones xBee. Tiene UART TTL e interrupciones, lo que permite ponerlo en modo Sleep. También se puede poner el xBee en modo Sleep, reduciendo el consumo total. Además posee tanto TWI (I2C) como SPI. Una ventaja importante de este dispositivo es el bajo consumo en Sleep y el poder programarlo mediante xBee, sin necesidad de conectarlo físicamente al ordenador.

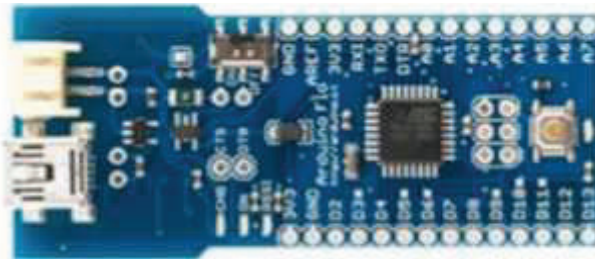


Figura 1.4. Placa módulo Arduino Fio [5]

**Arduino Uno:** Se caracteriza por tener un microcontrolador ATmega328P a 16Mhz, puerto USB, alimentación de 5V, memoria SRAM de 2Kb, memoria flash de 32Kb y 0.5Kb de reserva para el gestor de arranque, 1Kb de memoria EEPROM, 14 pines digitales y 6 con PWM, así como 6 salidas analógicas.

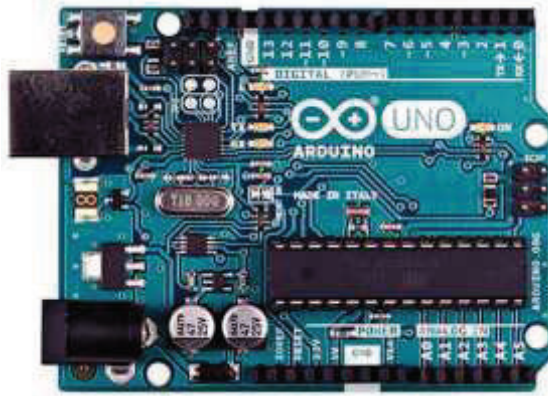


Figura 1.5. Placa módulo Arduino Uno [6]

**Arduino Mega:** Se caracteriza por tener un microcontrolador ATmega1280 o ATmega2560 a 16Mhz, alimentación de 5V, memoria EEPROM de 4Kb, 8Kb de memoria SRAM y 256Kb de memoria Flash con 8Kb reservados para el gestor de arranque. Las entradas analógicas son 16 y posee 15 salidas digitales.

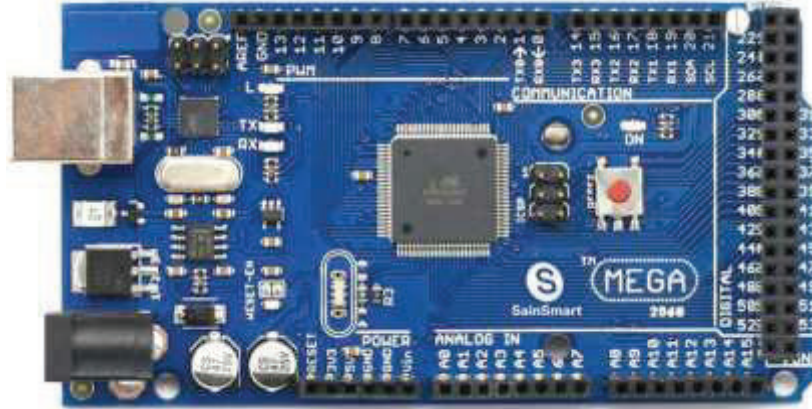


Figura 1.6. Placa módulo Arduino Mega [6]

## 1.3. Pantallas Táctiles

### 1.3.1. Definición

Una pantalla táctil, es un periférico de entrada y salida de datos para el dispositivo en el que esté instalada, actuando así como intermediario directo entre las órdenes y lo que debe hacer el dispositivo en cuestión.

Las órdenes en cuestión pueden ser dadas como su propio nombre indica, de manera táctil, o también mediante un lápiz o Stylus, que según sea la pantalla, deberá ser de tecnología óptica.

### 1.3.2. Funcionamiento

Una pantalla táctil es algo parecido a un teclado invisible pegado al frente de un monitor. Para entender cómo funciona, primero es vital saber cómo trabaja un teclado ordinario, en donde básicamente, cada tecla es un interruptor eléctrico. Cuando se presiona una de ellas, se completa un circuito y fluye una corriente, la cual varía dependiendo de la tecla que se oprima y así es como el equipo se da cuenta de lo que se está escribiendo.

Las pantallas táctiles intentan lograr algo similar a los teclados normales, pero obviamente no pueden utilizar interruptores, membranas de plástico y demás materiales, ya que de lo contrario obstruirían la visibilidad de lo que se muestra en pantalla. Por lo que deben utilizar trucos más astutos para detectar el tacto de manera completamente invisible.

Las pantallas táctiles funcionan de diferentes maneras. Algunas sólo pueden detectar un dedo a la vez y se confunden demasiado si se oprimen dos lugares diferentes al mismo tiempo, mientras que otras sí pueden hacerlo.

### 1.3.3. Tipos de pantallas táctiles

Existen varios tipos de pantallas táctiles, y funcionan de manera totalmente diferente según la tecnología usada y las características técnicas; así, se pueden detallar a continuación algunas de ellas:

- **Por infrarrojos:** En la Figura 1.7 se muestra como está compuesta la pantalla y su funcionamiento. Cuenta con detectores de infrarrojos y diodos emisores de luz infrarroja incrustados en la carcasa, que al tocar con el dedo se interrumpe la emisión y recepción del haz de luz causando una pérdida en la señal de los detectores de infrarrojos.

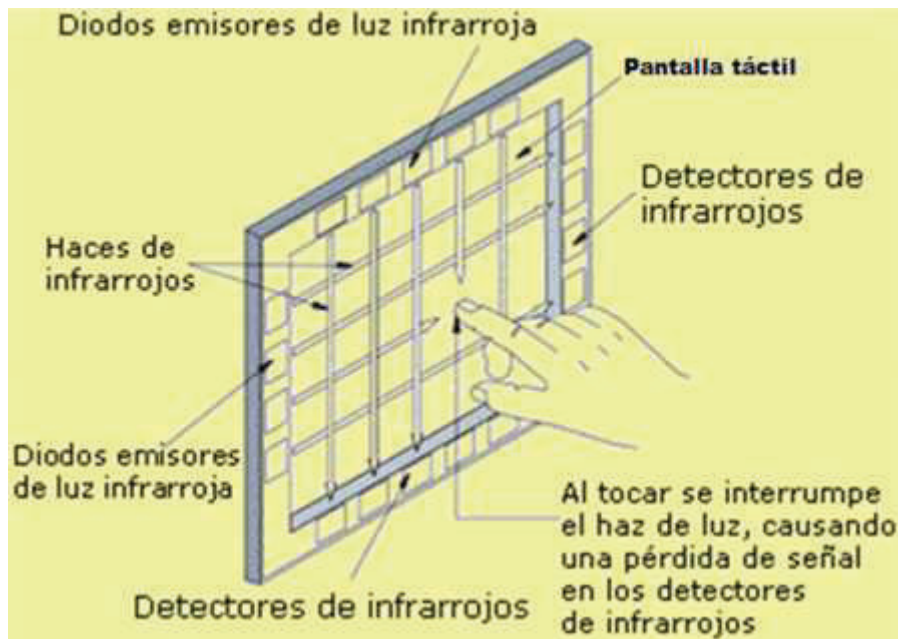


Figura 1.7. Funcionamiento de la pantalla táctil por infrarrojos [8]

- **Resistivas:** En la Figura 1.8 se muestran los componentes y el funcionamiento de este tipo de pantallas. Consta de dos capas de material conductor transparente ligeramente separadas entre sí, que al pulsar sobre la propia pantalla, estas dos partes se unen, donde un sistema electrónico detecta el contacto y es capaz de, midiendo la resistencia, saber el punto exacto del contacto.

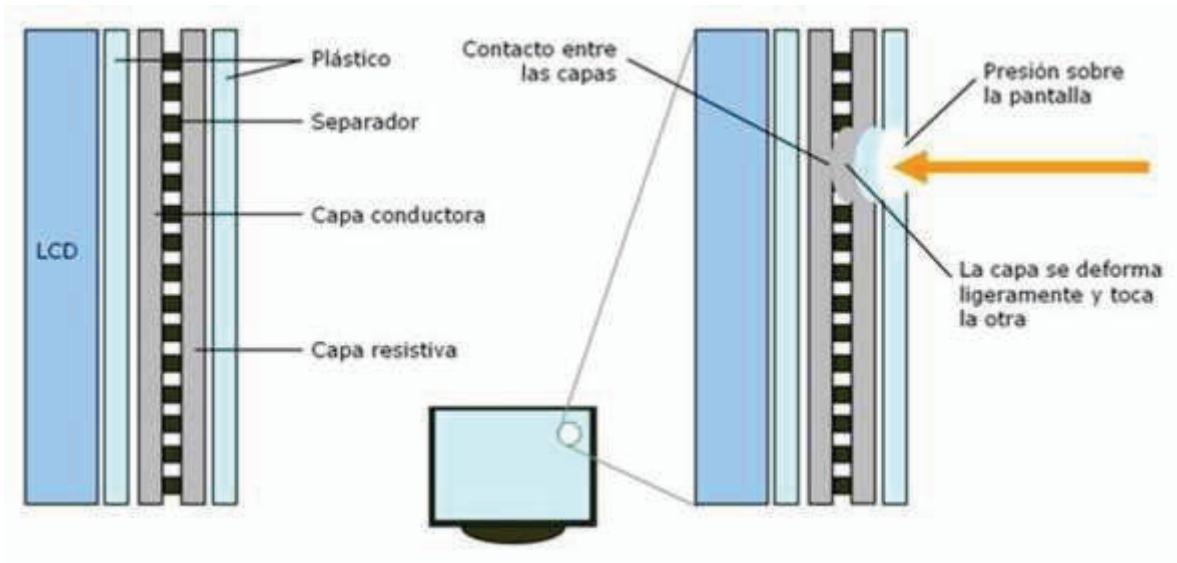


Figura 1.8. Componentes pantalla táctil resistiva [8]

- **Capacitivas:** Este tipo de pantalla se muestra en la Figura 1.9 con sus componentes y funcionamiento. Posee una capa conductora que al aplicar una tensión en cada una de las cuatro esquinas, aparece un campo eléctrico uniforme, que al momento de pulsar la pantalla genera un campo eléctrico de baja tensión, para determinar el lugar de pulsación.

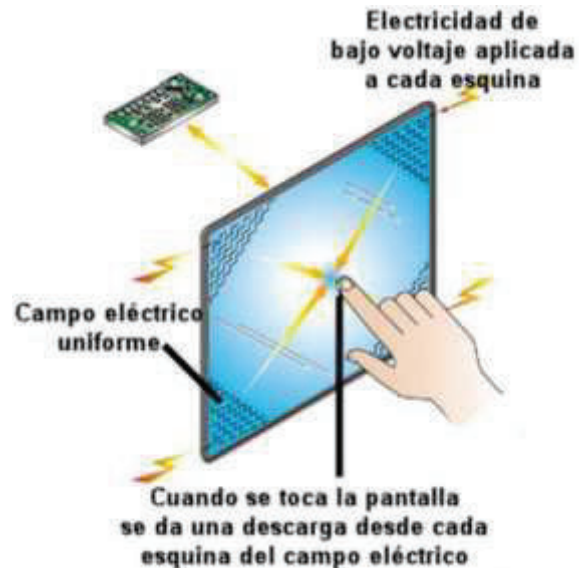


Figura 1.9. Pantalla táctil capacitiva [7]

- **De Onda Acústica Superficial:** A través de la superficie del cristal, se transmiten unas ondas acústicas que son inaudibles para los humanos. En la Figura 1.10 se muestra como está compuesta la pantalla y su funcionamiento, que al tocarla se absorbe parte de la energía acústica atenuando así la intensidad de la señal,

debido a que los transductores emiten ondas acústicas que se encuentran en los ejes de la pantalla.

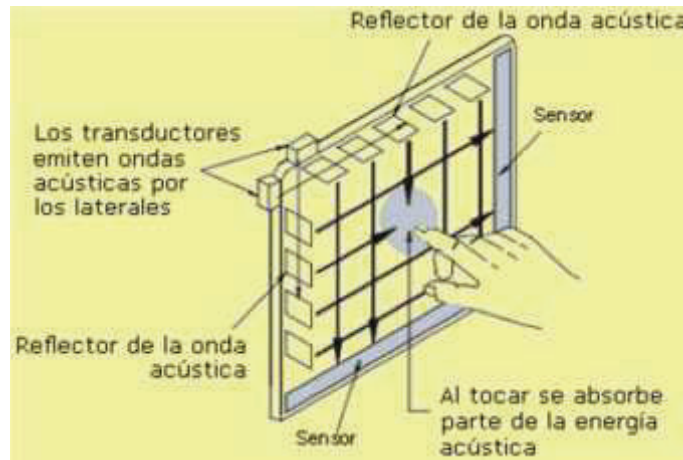


Figura 1.10. Funcionamiento pantalla táctil de onda acústica superficial [8]

## 1.4. Sensores

### 1.4.1. Definición

“Un sensor es un dispositivo diseñado para recibir información de una magnitud del exterior y transformarla en otra magnitud, normalmente eléctrica, que se pueda cuantificar y manipular” [8].

Los sensores, básicamente, transforman o reproducen un determinado fenómeno físico (pueden ser también químicos o bien fisicoquímicos) en una “señal” que puede ser una unidad de medida (como en el caso de los termómetros o barómetros), en sonidos (como los sistemas de seguridad por alarma) o en acciones específicas (por ejemplo, cuando se abre la puerta de un cajero automático mediante el ingreso de una tarjeta magnética). Este proceso se conoce en el ámbito de las ciencias como transducción, lo que constituye en la conversión de un dato en una información en un "lenguaje" diferente.

Otro ejemplo bien conocido y de uso cotidiano son los sensores utilizados en sistemas de alarmas, que se instalan en comercios u hogares para brindar seguridad a la propiedad en casos de robos o asaltos. Los sensores de movimiento utilizados en estos casos, permiten detectar la entrada o el movimiento de una persona en uno o más ambientes del hogar o comercio, una vez que se ha activado el sistema de seguridad por alarma.

### 1.4.2. Características de un sensor

Se describen las características técnicas más relevantes de un sensor a continuación:

- Aceptable calidad de medida
- Precisión
- Offset o desviación de cero
- Linealidad o correlación lineal.

- Sensibilidad
- Rapidez en respuesta
- Derivas
- Repetitividad

Un sensor procesa la magnitud que se quiere medir o controlar por medio de un transductor, para convertir en otra que facilita su medida. Para que los valores detectados puedan ser entendidos por las personas este dispositivo deberá ser conectado a un indicador, como un computador o un display,

### **1.4.3. Clasificación de sensores**

Los sensores de mayor interés se pueden clasificar como se detalla a continuación:

- Sensores mecánicos: para medir cantidades como posición, forma, velocidad, fuerza, torque, presión, vibración, deformación y masa.
- Sensores eléctricos: para medir voltaje, corriente, carga y conductividad.
- Sensores magnéticos: para medir campo, flujo y permeabilidad magnética.
- Sensores térmicos: para medir temperatura, flujo, conductividad y calor específico.
- Otros tipos como acústicos, ultrasónicos, químicos, ópticos, de radiación, láser y de fibra óptica.

De acuerdo con su aplicación, un sensor puede estar formado por materiales metálicos, no metálicos, orgánicos o inorgánicos, y por fluidos, gases, plasmas o semiconductores. Al usar características especiales de esos materiales, los sensores convierten la cantidad o propiedad medida en una señal analógica o digital.

## **1.5. Teclado Matricial**

### **1.5.1. Definición**

El teclado matricial es ensamblado en forma de matriz; “es un simple arreglo de botones conectados en filas y columnas, de modo que se pueden leer varios botones con el mínimo número de pines requeridos” [9], como se ilustra en la Figura 1.11.

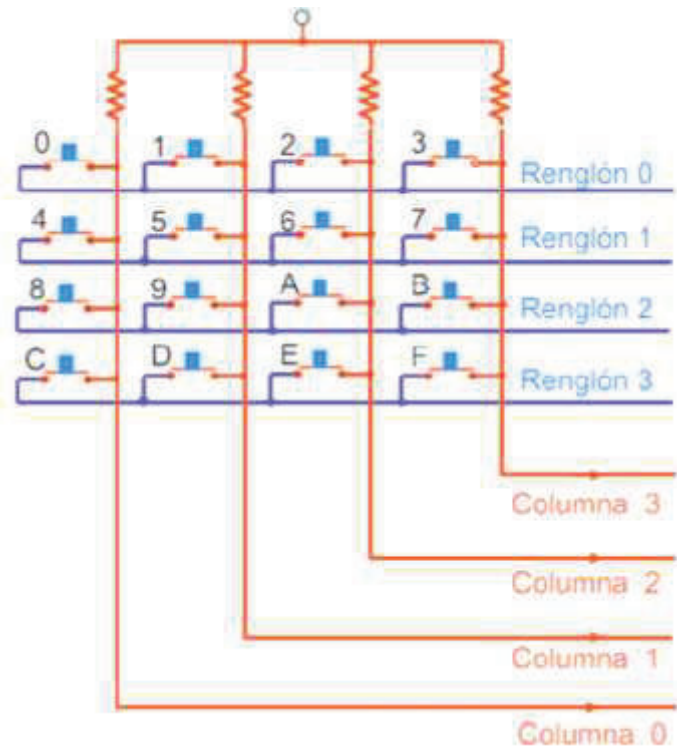


Figura 1.11. Teclado matricial [10]

### 1.5.2. Funcionamiento

Un teclado matricial 4x4 ocupa 4 líneas para las filas y otras 4 líneas para las columnas, lo que permite leer 16 teclas utilizando sólo 8 líneas de un microcontrolador.

Cuando no se ha oprimido ninguna tecla, no hay conexión entre filas y columnas. Mientras que si se oprime una tecla, se hace una conexión entre la columna y la fila de la tecla.

Muchos teclados comerciales ya traen incluido su decodificador, que escanea el teclado y si, una tecla es presionada, regresa un número que identifica la tecla. El decodificador mostrado tiene 8 entradas, las 4 entradas "X" son conectadas a las 4 columnas del teclado y las 4 entradas "Y" son conectadas a las 4 filas.

Cuando se oprime una tecla, el código de 4 bits de la tecla (con 16 teclas, los códigos están entre 0000 y 1111 en binario), aparecerá en las 4 líneas de salida y la línea de datos disponible (DA) se pone en BAJO. Si se conecta a una línea de interrupción, el microprocesador será interrumpido cuando se oprima alguna tecla. La rutina de la interrupción, entonces lee los 4 bits y procesa el dato.

El chip del decodificador se encarga de eliminar el rebote de las teclas, lo que libera al programador de esta responsabilidad, esto es una ventaja al usar un chip decodificador.



El principio de funcionamiento de un teclado matricial es muy sencillo. Básicamente cuando se pulsa un botón en el teclado, se une una fila con una columna.

Por ejemplo, al presionar la tecla “1”, se estará conectando la columna 1 con la fila 1; si se pulsa la tecla “4”, se está conectando nuevamente la columna 1, esta vez con la fila 2; si se pulsa la tecla “9”, entonces se está conectando la columna 3 con la fila 3.

Existen diversas formas de conectar e interpretar el funcionamiento de un teclado matricial. En el diagrama de la Figura 1.12 se puede apreciar un teclado matricial 3x4 conectado a los pines del puerto B.

Puerto B	TrisB	Teclado 3x4
RB0	0 (Salida)	Columna 1
RB1	0 (Salida)	Columna 2
RB2	0 (Salida)	Columna 3
RB3	1 (Entrada)	Fila 1
RB4	1 (Entrada)	Fila 2
RB5	1 (Entrada)	Fila 3
RB6	1 (Entrada)	Fila 4
RB7	1 (Entrada)	Sin Conexión

Figura 1.12. Teclado matricial 3x4 conectado a los pines del puerto B. [10]

## 1.6. Dispositivo de alerta

La mayoría de los sistemas de alarma y advertencia utilizan el sonido como medio de atraer la atención del usuario.

Las sirenas electrónicas son empleadas como alarmas acústicas. “El sonido de la sirena se genera electrónicamente y se conecta a un parlante a través de un amplificador. Actualmente, el circuito generador del sonido y el altoparlante vienen acoplados en el mismo sistema” [11]. En la actualidad dentro del mercado se pueden encontrar sirenas electrónicas de 12 voltios de corriente continua y de 110 voltios de corriente alterna.

Generalmente, las sirenas se ubican en el exterior de las viviendas o locales, el ruido que emiten alertará a los pobladores cuando haya intrusos en el lugar.

## **CAPÍTULO 2**

### **DISEÑO E IMPLEMENTACIÓN DEL PROTOTIPO DEL SISTEMA DE SEGURIDAD**

El presente capítulo detalla el diseño e implementación de los elementos para el sistema de seguridad electrónica del prototipo. Adicionalmente, se analiza los antecedentes y parámetros que intervienen, se establecen los requerimientos necesarios para el sistema, además se realiza la selección de los materiales y dispositivos electrónicos que se emplearán.

#### **2.1. Antecedentes**

Actualmente se ha visto que es importante contar con sistemas de seguridad para pequeños locales de los alrededores de la parroquia de Sangolquí, tanto por el descuido de los dueños al momento de asegurar sus negocios como por la inseguridad que existe en el sector.

Ante el hecho expuesto se ha tomado como referencia al Restaurante “El Pailón de Mushuñan”, un pequeño local comercial que se encuentra ubicado en la Parroquia de Sangolquí, Barrio Mushuñan, calle Inés Gangotena y Chimbo, para elaborar un prototipo de sistema de seguridad temporizado para puertas y ventanas, mismo que podrá ser de utilidad para otros pequeños negocios que existen en el sector.

Se ha tomado como referencia este negocio porque a los pocos meses de empezar su funcionamiento fue objeto de robo. Debido al descuido de los dueños que se olvidaron de cerrar correctamente la ventana de uno de los baños. Aunque, este es un punto no tan vulnerable, los ladrones se dieron sus modos e ingresaron por este lugar y sustrajeron algunos artículos. De ahí, que es necesario dar un aviso antes de que los dueños salgan para que verifiquen que tanto las puertas, como las ventanas queden cerradas correctamente, como también proteger al lugar cuando los dueños no se encuentren.

#### **2.2. Determinación de los puntos a ser considerados en el Sistema de Seguridad**

##### **2.2.1. Descripción del Restaurante**

Se describe brevemente las áreas que constituyen al Restaurante “El Pailón de Mushuñan”. Realizando un análisis técnico de los distintos espacios, con la finalidad de determinar los sitios vulnerables a intrusiones y al olvido de ser aseguradas correctamente.

El local comercial está conformado de las siguientes áreas:

- Hall
- Área de parrilla
- Área de cocina 1
- Área de cocina 2
- Salón
- Servicios higiénicos

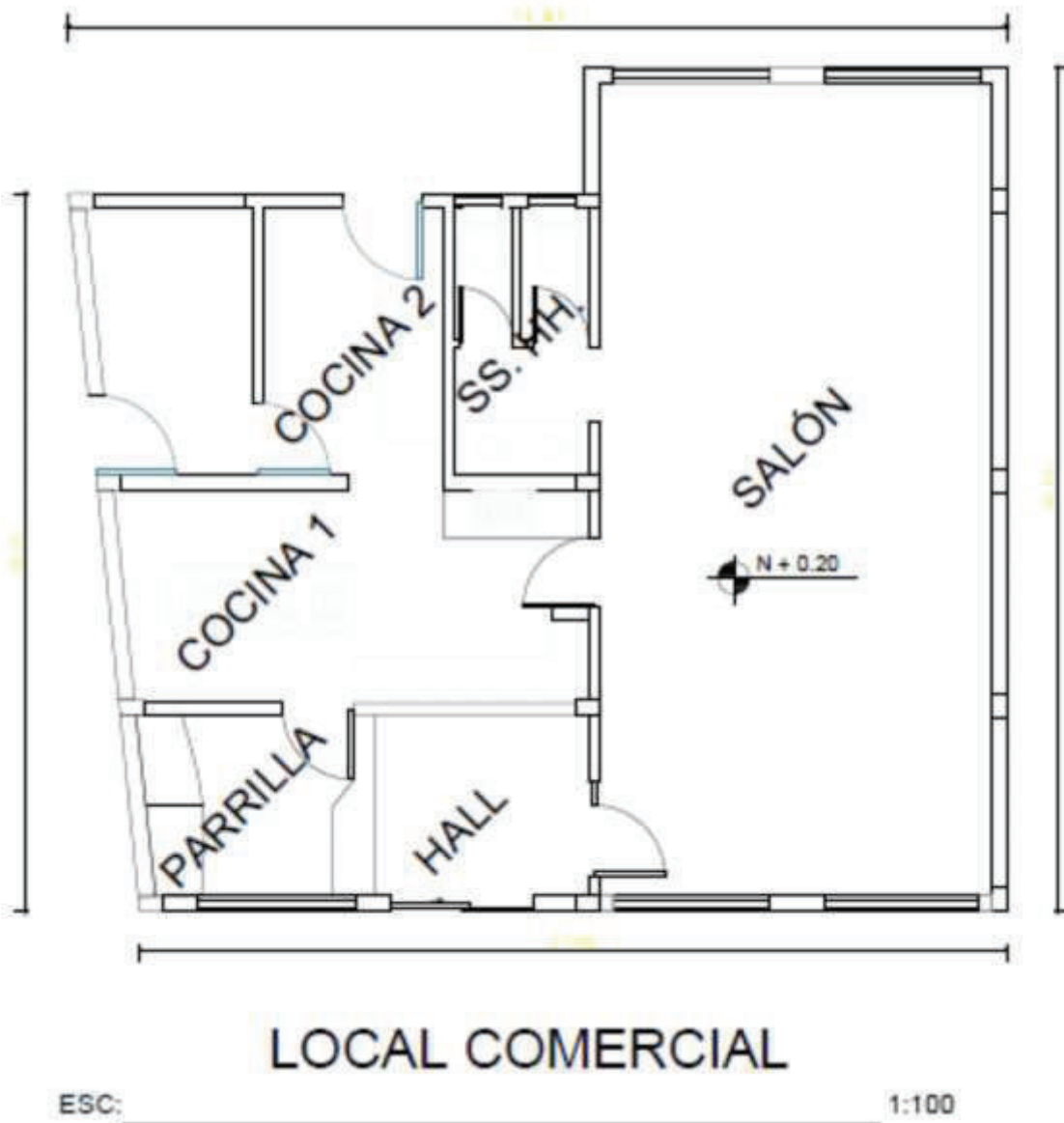


Figura 2.1. Áreas del Restaurante

En la Figura 2.1, se presenta el plano con los lugares, en los que se va asignar el sistema de seguridad electrónica estas son: Hall, Parrilla, Cocina 1, Cocina 2, Salón y Servicios Higiénicos.

Debido a que el área de Cocina 1 se encuentra en medio de dos sitios ya considerados en el sistema de seguridad, ésta se descarta; pero posteriormente en esta zona se podrán instalar sensores de gas y humo.

### 2.2.2. Estudio técnico para el sistema de seguridad electrónica

Luego de la identificación de los espacios que conforman el Restaurante, tomado en referencia para el diseño e implementación del prototipo, se efectúa un estudio técnico para determinar los puntos vulnerables, así como la ubicación de los diferentes sensores y dispositivo de alerta.

Para todas las áreas, el estudio técnico comprende el análisis de la tipología física de cada una de las puertas, ventanas pequeñas y ventanas con vidrios grandes, así como equipos, implementos, materiales que contiene el local comercial y requerimientos de los dueños.

En la Tabla 2.1, se muestra el número de puertas y ventanas que conforman cada área, así como los elementos y equipos que existen en cada una de ellas.

Número Puertas	Número Ventanas	Área	Denominación	Equipos y elementos
1	1	Hall	Puerta de vidrio corrediza Ventana grande de vidrio	Caja registradora, computadora, equipo de sonido
1	1	Parrilla	Ventana con vidrio grande Puerta de madera	Equipo para parrilla.
1	1	Cocina 1	Puerta de madera Ventana grande de vidrio	Cocinas, implementos de cocina, etc.
2	1	Cocina 2	Puertas metálicas Ventana pequeña no se abre	Frigoríficos, vajilla, licuadoras, implementos de cocina.
1	4	Salón	Puerta de madera Ventanas con vidrio grande	Mesas, sillas, instrumentos musicales, implementos de decoración.
2	2	SS.HH.	Puertas de madera Ventanas pequeñas	Implementos de baño.

Tabla 2.1. Determinación de las partes físicas de cada área.

### 2.2.3. Características de seguridad

De acuerdo con lo especificado en la Tabla 2.1, se determina los sitios vulnerables en donde se colocarán los sensores magnéticos y de ultrasonido.

Área	Tipo Sensor		Posición de sensores	Denominación
	Magnético(M)	Ultrasonido(U)		
<b>Hall</b>	1	1	Puerta de vidrio corrediza Parte inferior del mesón cerca a la puerta de vidrio	M1 U2
<b>Parrilla</b>	1	1	Ventana con vidrio grande Pared lateral izquierda	M4 U1
<b>Cocina 2</b>	2	0	Puertas metálicas	M2, M3
<b>Salón</b>	4	3	Ventanas con vidrios grandes Paredes lateral izquierda y derecha	M5, M6, M7, M8 U3, U4, U5
<b>SS.HH</b>	2	0	Ventanas pequeñas	M9, M10
<b>TOTAL</b>	<b>10</b>	<b>5</b>		

Tabla 2.2. Distribución por áreas de los sensores.

De la Tabla 2.2, se determina emplear 10 sensores tipo magnético que se ubicarán en puertas y ventanas para detectar el cierre y apertura de las mismas, tanto para el caso de intrusiones como para el caso en que los dueños olviden cerrar una de ellas.

Además, se han considerado cinco sensores tipo ultrasonido, que se colocarán en la parte interna del Salón, Parrilla y Hall. Estos sensores servirán para monitorear en el caso de que exista alguna irrupción del sistema por rotura de vidrio o perforación de paredes. También se censará constantemente la presencia en la entrada principal.

La posición en la que los sensores serán instalados en el sistema de seguridad electrónica en el modelo a escala del Restaurante, se indica en la Figura 2.2, además se especifica el lugar del dispositivo de alerta que para este caso será un buzzer; así como la unidad de control del sistema (Módulo Arduino y teclado).



Figura 2.2. Plano de la ubicación de los sensores

## 2.3. Descripción de los elementos instalados en el sistema de seguridad electrónico

Para determinar qué tipos de dispositivos se van a emplear dentro del sistema, se tomó en cuenta las características técnicas más adaptables para el prototipo del sistema de seguridad. Además se busca eficacia y reducción del costo económico.

### 2.3.1. Sensores Magnéticos

Este tipo de elemento electrónico detecta un cambio dentro del campo magnético mismo que se produce por la variación de alguna magnitud física. Los sensores magnéticos

están basados en el efecto Hall, “El efecto Hall se produce cuando se ejerce un campo magnético transversal sobre un cable por el que circulan cargas. Como la fuerza magnética ejercida sobre ellas es perpendicular al campo magnético y a su velocidad, las cargas son impulsadas hacia un lado del conductor y se genera en él un voltaje transversal”. [12] [10]

De ahí que el contacto magnético que se emplea en este trabajo, está conformado por dos unidades necesariamente hermanadas en una posición determinada, y ante la separación de las dos piezas se produce un cambio mecánico en los contactos de una de ellas, lo cual informa de un cambio de estado, pasando de abierto a cerrado o viceversa.

Su principal característica es ser un dispositivo de estado sólido, que no tiene partes móviles, es compatible con otros circuitos analógicos y digitales, teniendo un margen de temperatura amplio, buena receptividad y frecuencia de funcionamiento relativamente alta (100 KHz). De ahí que en la Figura 2.3, se muestra el sensor magnético que se emplea en este trabajo.



Figura 2.3. Sensor magnético [13]

Características:

- Dimensiones: 23 x 14 x 6 mm
- Estado del Reed Switch: Normalmente Abierto (NO). Se cierra al estar cerca de un campo magnético (imán)
- Voltaje máximo: 100 VDC
- Corriente máxima: 0,5 Amperes
- Distancia de operación: 15 – 25 mm
- Características del cable: 24 AWG, 350 mm de largo
- Color carcasa: Blanco
- Material carcasa: ABS

### 2.3.2. Sensores de ultrasonido HC-SR04

Un sensor de ultrasonido es un dispositivo para medir distancias. En la Figura 2.4 se muestra su funcionamiento, que se basa en el envío de un pulso de alta frecuencia, no audible por el ser humano. Este pulso rebota en los objetos cercanos y es reflejado hacia

el sensor, midiendo el tiempo entre pulsos y conociendo la velocidad del sonido, se puede estimar la distancia del objeto contra cuya superficie impactó el pulso de ultrasonido. El rango de medición teórico del sensor HC-SR04 es de 2 cm a 400 cm, con una resolución de 0,3 cm.

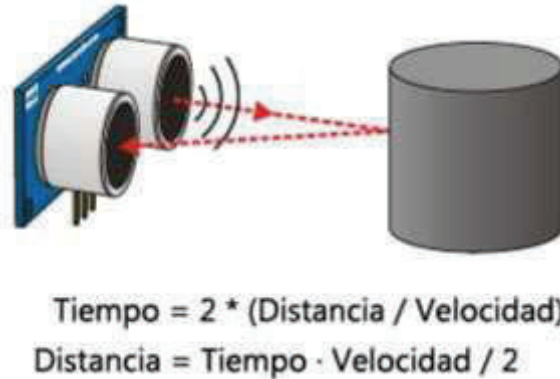


Figura 2.4. Funcionamiento sensor de ultrasonido HC-SR04 [15]

El sensor se basa simplemente en medir el tiempo entre el envío y la recepción de un pulso de ultrasonido. Se sabe que la velocidad del sonido es 343 m/s en condiciones de temperatura 20 °C, 50% de humedad, presión atmosférica a nivel del mar. Transformando unidades resulta:

$$343 \frac{m}{s} \cdot 100 \frac{cm}{m} \cdot \frac{1}{1000000} \frac{s}{\mu} = \frac{1}{29,2} \frac{cm}{\mu}$$

Ecuación 1.

Es decir, el sonido tarda 29,2 microsegundos en recorrer un centímetro. Por tanto, se puede obtener la distancia a partir del tiempo entre la emisión y recepción del pulso mediante la siguiente ecuación.

$$D (cm) = \frac{T (\mu)}{29,2 \times 2}$$

Ecuación 2.

Esquema:

Se conectan los pines Trigger y Echo a pines digitales y la respectiva alimentación del sensor a 5 V. La conexión del sensor ultrasónico con los módulos Arduino es similar, así en la Figura 2.5 se muestra la conexión con el Arduino Uno y Mega, los pines que se empleen para conectar el Trigger y Echo dependerá de las necesidades que requiera el diseño.



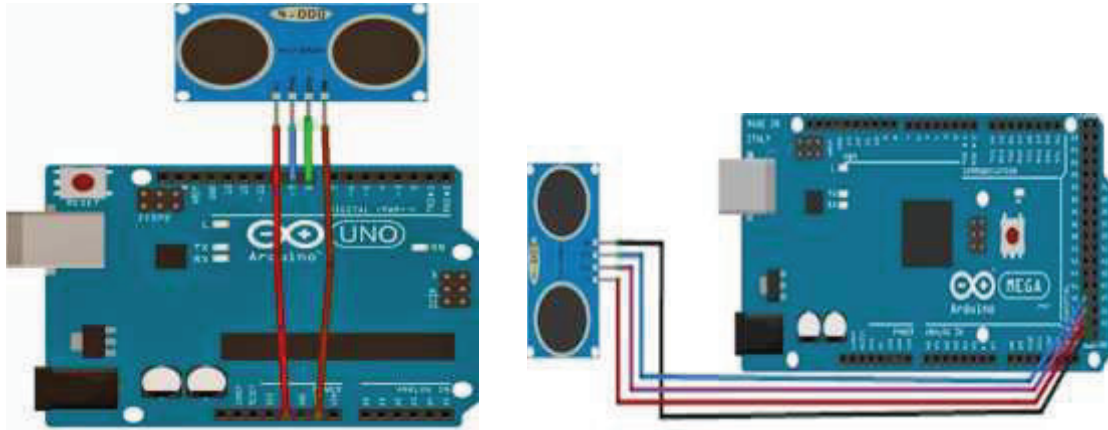


Figura 2.5. Conexión del sensor ultrasonido con Arduino Uno y Mega [14]

Características:

- Voltaje de funcionamiento: 5V (DC)
- Corriente estática: <2 mA
- Corriente de trabajo: 15 mA
- Frecuencia de trabajo: 40 KHz
- Señal de salida: frecuencia de la señal eléctrica, 5 V de alto nivel, bajo nivel de 0 V
- Ángulo Eficaz: <15°
- Distancia de detección: 2 cm - 450 cm
- Resolución: 0,3 cm
- Ángulo de medición: 30°
- Disparo de la señal de entrada: pulso TTL 10  $\mu$ s
- Echo señal de salida: señal PWL de TTL
- VCC = +5 VDC
- Trig = Entrada de Trigger de sensor
- Echo = Salida de Echo de sensor
- GND = GND

### 2.3.3. HMI Nextion Enhanced NX4024K032

Nextion Enhanced NX4024K032 es una potente pantalla HMI TFT de 3.2 ", con 16 MB de espacio de almacenamiento de datos Flash, EEPROM de 1024 bytes, RAM de 3.584 bytes. Con el soporte de GPIO, para controlar dispositivos externos.

Es una solución de interfaz humana (HMI) que proporciona una interfaz de control y visualización entre un humano y un proceso, una máquina, una aplicación o un dispositivo. Se aplica principalmente al Internet de las Cosas (IoT) o al campo de la electrónica de consumo. Es la mejor solución para reemplazar el tradicional LCD y el tubo LED Nixie.

Nextion incluye una parte de hardware (una serie de tarjetas TFT) y una parte de software propia del editor de Nextion. La tarjeta Nextion TFT utiliza sólo un puerto serie para comunicarse. Permite a los usuarios evitar las molestias del cableado. Es fácil adaptar el

HMI de la familia Nextion a los proyectos existentes, ya que los usuarios sólo necesitan proporcionarle un UART, en la Figura 2.6 se muestra la pantalla con sus cuatro cables que utiliza para la conexión con el Arduino.

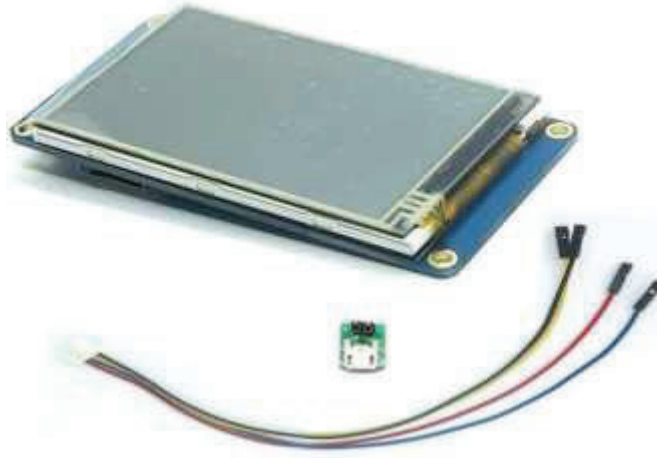


Figura 2.6. HMI Nextion NX4024K032 [16]

#### Características:

- Compatible con RTC incorporado
- Soporta GPIO
- Interfaz para tarjeta SD: tarjeta máxima 32 G Micro TF/SD (formato de archivo FAT32)
- Espacio de almacenamiento de datos Flash: 16 MB
- EEPROM: 1.024 bytes
- RAM: 3.584 bytes
- Buffer de instrucción: 1.024 bytes
- Color: 65K (65.536) colores
- Resolución: 400 × 240 píxeles
- Brillo ajustable: 0 ~ 180 nit- el intervalo de ajuste es 1%

#### Especificación:

- Área visual: 69,60 mm (L) × 41,76 mm (W)
- Tamaño del diseño: 95 (L) x 47,6 (W) x 5,55 (H)
- Tipo de toque: Resistivo
- Luz de fondo: LED
- Vida útil de la retroiluminación (media):> 30.000 horas
- Peso neto: 42,5 g
- Fuente de alimentación recomendada: 5 V 500 mA DC

### 2.3.4. Controlador Arduino Mega 2560

El controlador que se eligió para la implementación del proyecto es un Arduino modelo MEGA 2560, ya que cumple con las necesidades del sistema en cuanto a entradas y salidas que se necesitaron para poder implementar el sistema.

En la Figura 2.7, se detallan las partes que forman al Arduino Mega 2560, alimentación de 5 voltios que se la hace en el conector USB, 54 pines digitales de entrada/salida (de los cuales 15 se pueden utilizar como salidas PWM) y 16 entradas analógicas, 4 UARTs (puertos serie de hardware) que permiten la comunicación serial, un conector de alimentación para fuente externa, una cabecera ICSP, y un botón de reinicio. Además cuenta con un controlador que posee un microprocesador ATmega 2560, con un oscilador de 16 MHz, memoria de 4Kb de EEPROM, 8 Kb de SRAM y 256 Kb de flash.

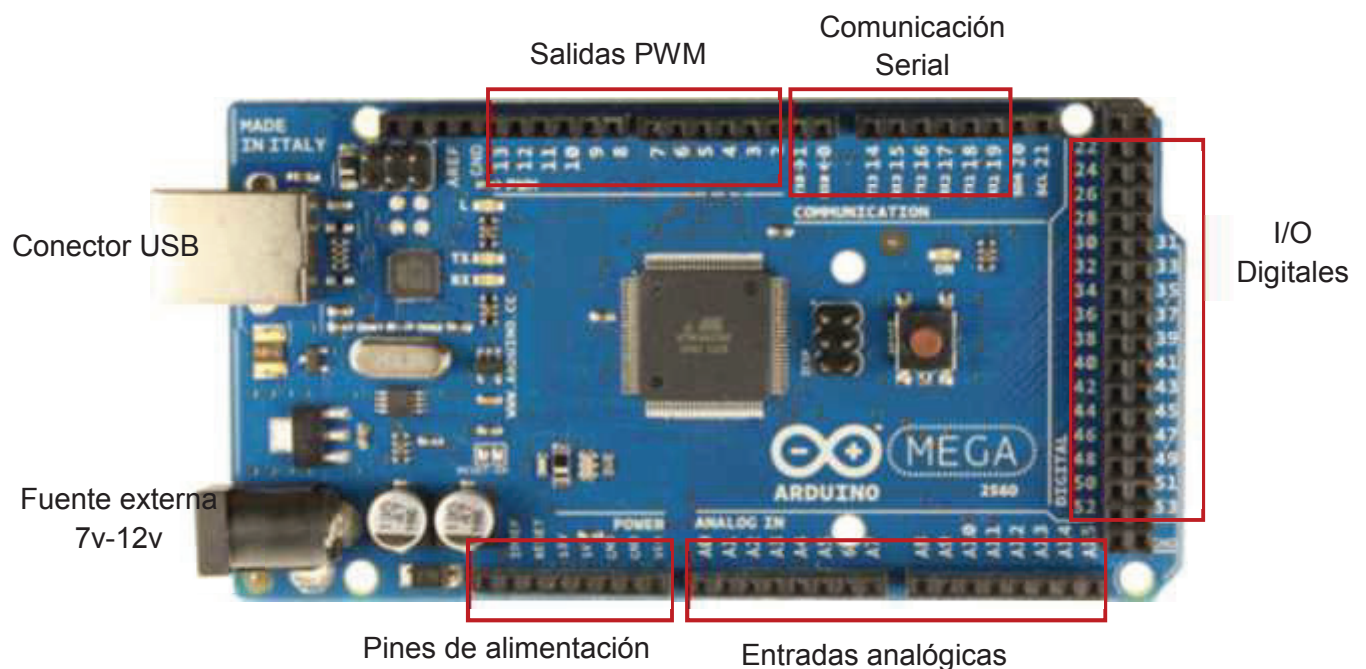


Figura 2.7. Partes de Arduino Mega 2560 [9]

### 2.3.5. Lenguaje de programación Arduino

El lenguaje Arduino (Arduino Programming Language) es un lenguaje de programación de alto nivel similar a otros lenguajes de programación, pero muy sencillo y fácil de aprender. Se basa en Wiring, una plataforma de código abierto que a su vez se apoya sobre el lenguaje de programación Processing.

### 2.3.6. Teclado Matricial 4x4

El teclado matricial 4x4 proporciona una interfaz sencilla de entrada de datos. Sus usos pueden ser tan variados como aplicaciones, que precisen de la introducción manual de

datos, puedan ocurrírsele al diseñador. Desde una calculadora, pasando por una cerradura codificada, un termostato programable, etc.

Esquema: En la Figura 2.8 se muestra la conexión del teclado matricial con el Arduino Uno y Mega, se observa que la conexión es similar para ambos, tomando en cuenta que los pines que se usen dependerá de los requerimientos del diseño.

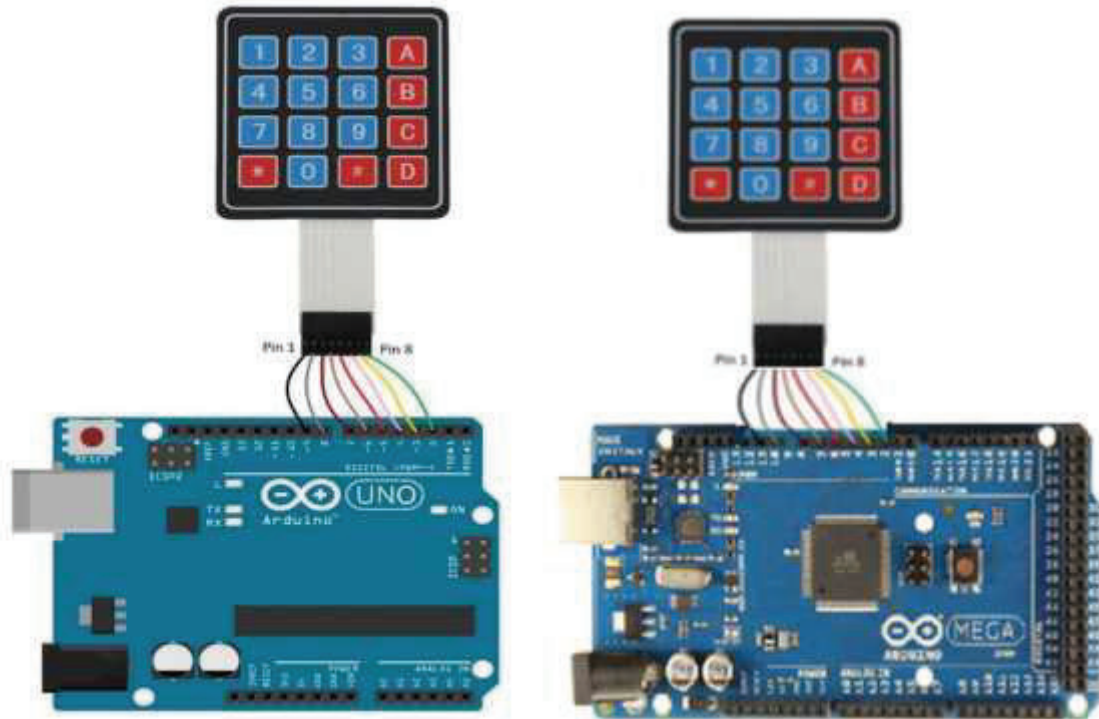


Figura 2.8. Conexión teclado matricial 4x4 con Arduino. [17]

Características:

- 16 botones con organización matricial (4 filas x 4 columnas)
- Teclado tipo membrana
- Mayor resistencia al agua y al polvo
- Auto adhesivo en la parte de atrás
- Tiempo de rebote (Bounce time):  $\leq 5$  ms
- Máximo voltaje operativo: 24 V DC
- Máxima corriente operativa: 30 mA
- Resistencia de aislamiento: 100 M $\Omega$  (@ 100 V)
- Voltaje que soporta el dieléctrico: 250 VRMS (@ 60Hz, por 1 min)
- Expectativa de vida: 1.000.000 de operaciones
- Dimensiones del pad: 6,9 x 7,7 cm aprox.
- Cable de cinta plana de 8.3 cm de largo aprox. (incluido el conector)
- Conector tipo DuPont hembra de una fila y 8 contactos con separación estándar 0,1" (2,54mm)
- Temperatura de operación: 0 a 50°C

## 2.4. Diseño del sistema de seguridad

En esta sección se detallan todos los elementos electrónicos que se utilizarán en la seguridad electrónica del lugar, la interfaz gráfica que permite visualizar una alerta cuando una puerta o ventana están abiertas, así como el uso de un teclado que facilita la activación y desactivación del sistema, acorde con el diseño e implementación.

El desarrollo del hardware de este proyecto, dará a conocer cómo interactúa la señal de los sensores magnéticos y de ultrasonido con el Arduino y el HMI Nextion, así como la comunicación para visualizar en el HMI el funcionamiento del sistema.

### 2.4.1. Diagrama de bloques del hardware

De manera global, el proyecto consta de un solo sistema para el procesamiento de datos como se muestra en la Figura 2.9, el cual está formado por el mando central (módulo Arduino Mega y teclado), bloque de sensores (sensores magnéticos y ultrasonido) y bloque indicador (buzzer y pantalla Nextion).

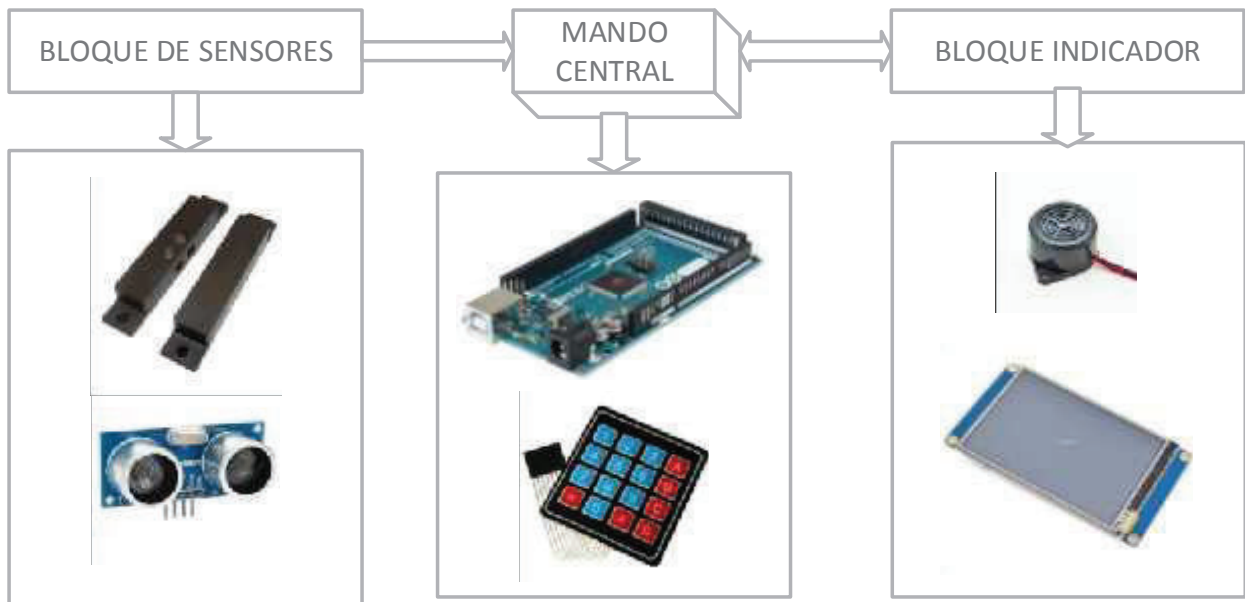


Figura 2.9. Diagrama de bloques del sistema

Para el desarrollo del sistema se realizaron varias pruebas de conexión entre el Arduino, sensores magnéticos y de ultrasonido. También se probó la comunicación entre el HMI y Arduino, que permite visualizar con un mensaje cuando una puerta o ventana están abiertas y dar un aviso, al momento de activar el sistema, que algunos de los sensores magnéticos colocados en puertas y ventanas no se han cerrado.

### 2.4.2. Programación de controlador Arduino

El entorno que proporciona Arduino es de fácil manejo para programar la placa controladora, para la configuración de la comunicación con el HMI y para la conexión con

los sensores magnéticos y de ultrasonido. Este software proporciona distintas ayudas al momento de realizar el proyecto.

#### 2.4.2.1. Configuración de la placa Arduino Mega

Antes de llevar a cabo la programación de la placa controladora Arduino es necesario saber el proceso correcto que va a utilizar el sistema para proceder con la programación de cada una de las etapas ya que es de suma importancia cada detalle del proceso para que no interfiera en su correcto desempeño. Luego de tener claro el proceso que se va a realizar, así como las variables de entrada y salida, estas variables se ingresan en el entorno de programación. Se puede utilizar cualquier estilo de programación que permita el entorno Arduino.

El proceso que se llevó a cabo para programar el Arduino Mega empieza con la creación de un nuevo Sketch, el cual genera un archivo de origen necesario para la respectiva configuración y programación, como se muestra en la Figura 2.10.

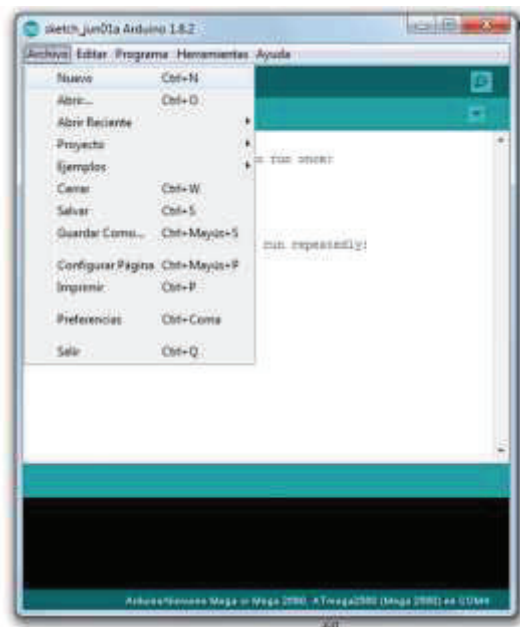


Figura 2.10. Creación de un nuevo Sketch en Arduino.

Una vez creado un nuevo proyecto, el programa despliega una ventana que es el entorno de programación, en donde se escoge el tipo de tarjeta controladora que se va a utilizar y la configuración del dispositivo, en este caso se utiliza la placa Arduino Mega, tal como se indica en la Figura 2.11.

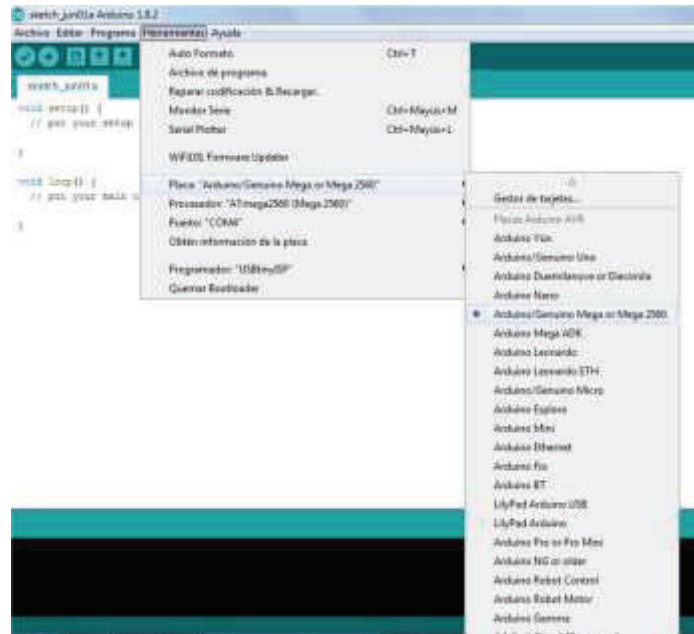


Figura 2.11. Selección del tipo de tarjeta a utilizar.

#### 2.4.2.2. Diagrama de flujo del programa

El siguiente paso luego de configurar la tarjeta controladora es ingresar las variables con sus respectivas direcciones, identificando el tipo de variable que se utilizará en el proceso de programación.

Una vez declaradas las variables que se va a utilizar en el programa, se puede proceder a realizar la programación para el proceso. El programa está llevado al lenguaje de programación Arduino siguiendo cada etapa consecutivamente.

La comunicación entre el Arduino Mega y la pantalla HMI Nextion, se la llevará a cabo empleando una velocidad de comunicación de 9600 baudios, ya que esta es la más óptima para este tipo de dispositivo.

En la siguiente etapa se especifica cada una de las instrucciones que debe procesar el módulo Arduino, como la activación del sistema mediante el ingreso de una clave por medio de un teclado. Además, cuando un sensor magnético se encuentre en estado HIGH (se encuentra haciendo contacto), con lo que el dispositivo de alerta no da ningún tipo de sonido, mientras que para el caso en que el sensor esté en el estado LOW (no hace contacto) existe una alerta visual-sonora por medio del buzzer y la pantalla, el mismo que avisa a los dueños que alguna de las puertas o ventanas están abiertas.

En esta etapa también se tiene la programación de los sensores de ultrasonido, que empiezan su funcionamiento cuando el sistema está activado y se mantienen detectando aún cuando el sistema esté apagado. Por ser un prototipo, se opta por emplear este tipo

de sensores, ya que para el uso real en el sitio se deberá emplear sensores de presencia infrarrojos.

En la Figura 2.12, se muestra el diagrama de flujo del programa donde se representa el proceso que debe seguir el módulo Arduino cuando se active uno de los sensores magnéticos o de ultrasonido, dando un aviso de alerta a través del sonido de un buzzer y el mensaje de “ABIERTA” ya sea para el caso de una puerta o ventana.

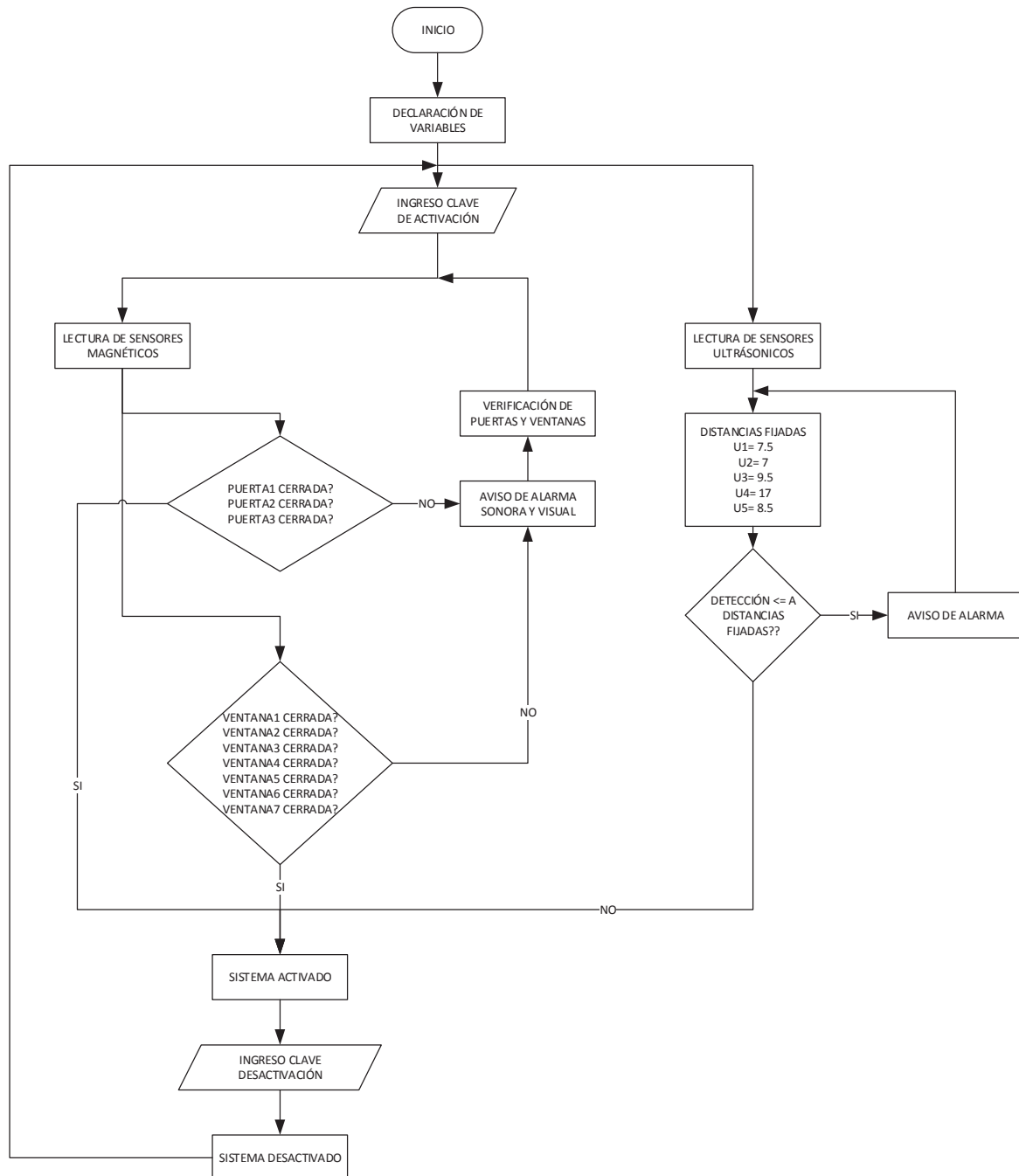


Figura 2.12. Diagrama de flujo del programa.



### 2.4.3. Programación de HMI

La interfaz de máquina humana HMI, permite la comunicación entre personas y máquinas. Estos sistemas se componen de paneles que poseen indicadores y comandos, tales como luces pilotos, indicadores digitales y analógicos, pulsadores, selectores y otros que se interconectan con la máquina, de forma virtual. Este tipo de proceso se desarrolla sobre pantallas programadas; puesto que las HMI poseen puertos de comunicación, que facilitan una conexión más sencilla y económica con los procesos o máquinas.

#### 2.4.3.1. Configuración del HMI Nextion

La configuración de la pantalla HMI NX4024K032 se realiza mediante el software libre “Nextion Editor” v0.35, en él se realiza toda la parte visual de la pantalla es decir aquí se diseñan los menús, se agregan toolbox como: botones, cuadros de texto, figuras, barras, punteros, entre otros.

Como primera configuración se tiene la selección del modelo de HMI en el cual se detalla el tamaño de la pantalla y la calidad de imagen, en este caso se utiliza una pantalla de 3,2 pulgadas con una calidad de imagen de 400x240 pixeles.

En la Figura 2.13 se muestra el menú de selección del tipo de pantalla HMI.

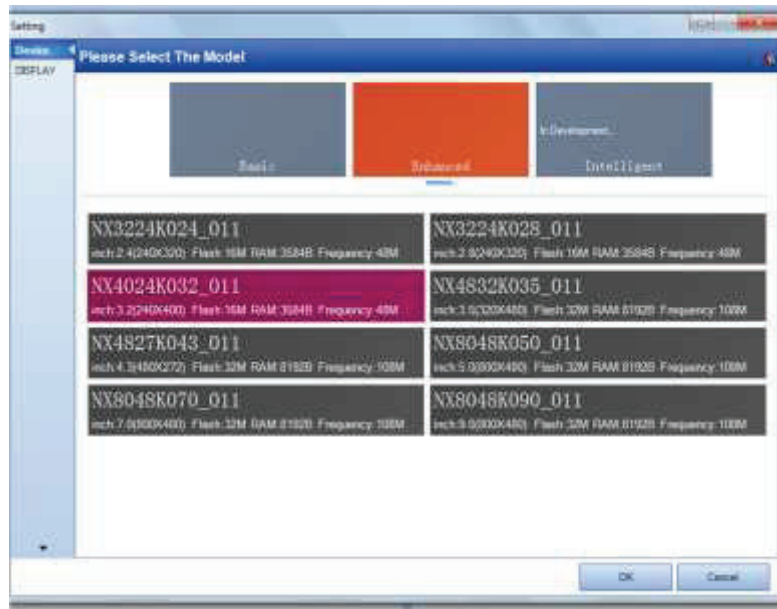


Figura 2.13. Selección HMI en Nextion Editor

A continuación se debe seleccionar la orientación de la pantalla, pudiendo elegir entre cuatro opciones: Horizontal a 0 grados, Vertical a 90 grados, Horizontal a 180 grados y Vertical a 270 grados, como se muestra en la Figura 2.14. Para la HMI se selecciona Horizontal a 90 grados que es la más adecuada para su empleo en el proyecto.

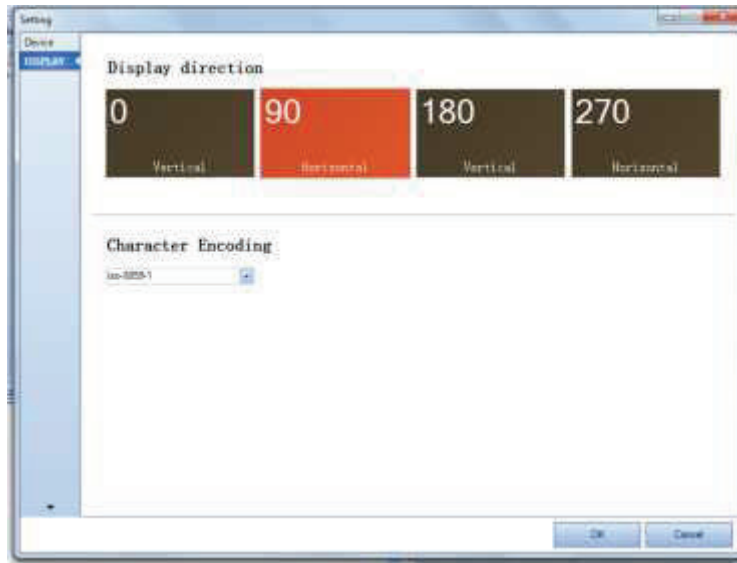


Figura 2.14. Selección de orientación HMI

#### 2.4.3.2. Programación del HMI Nextion

Una vez configurada la pantalla, se procede a su desarrollo, en este caso Nextion Editor proporciona una interfaz muy intuitiva y con varios complementos fáciles de usar y de configurar. La Figura 2.15 muestra el entorno de la programación.

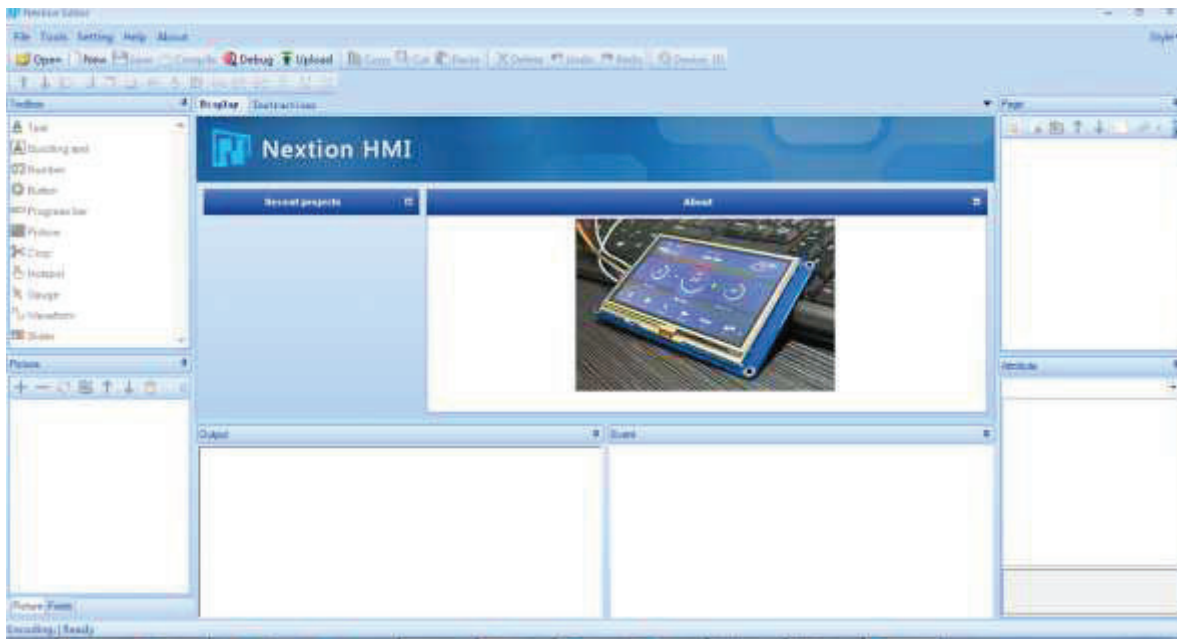


Figura 2.15. Entorno de la programación del HMI en Nextion.

Para empezar con el desarrollo de la HMI se necesitan las imágenes que se vayan a colocar en los menús, donde es importante que estas imágenes tengan una buena calidad y sean del tamaño de la pantalla es decir de 400x240 pixeles, al ser de otra forma las

imágenes no se verán completas en la misma o se verán distorsionadas. Otro elemento necesario antes de programar la HMI es la creación de fuentes de texto; es decir, los tipos de letra que se utilizará.

La Figura 2.16 detalla los diferentes toolbox de configuración presentes en el programa.

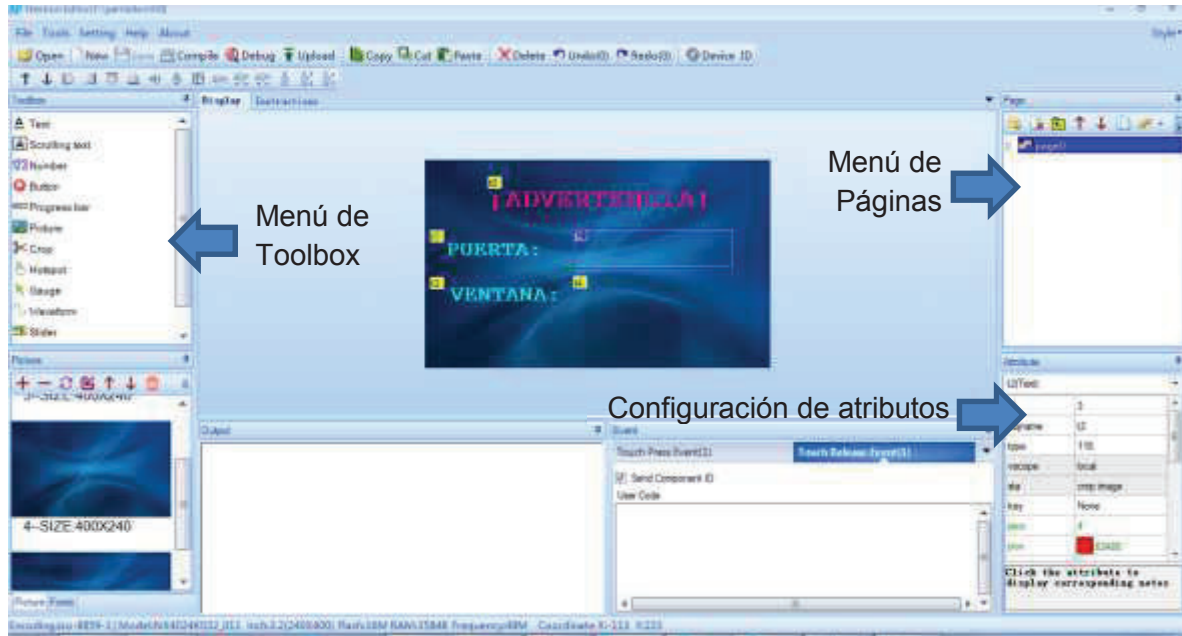


Figura 2.16. Menú de páginas y toolbox en Nextion.

#### 2.4.4. Diagrama de Conexiones

La elaboración del diagrama de conexiones se realizó en el Software Proteus 8 Professional, donde se seleccionan todos los componentes que conforman el sistema; así como la representación de la interconexión de cada uno de los sensores con el Arduino, Teclado Matricial y Pantalla HMI.

En la Figura 2.17 se muestra el diagrama desarrollado.

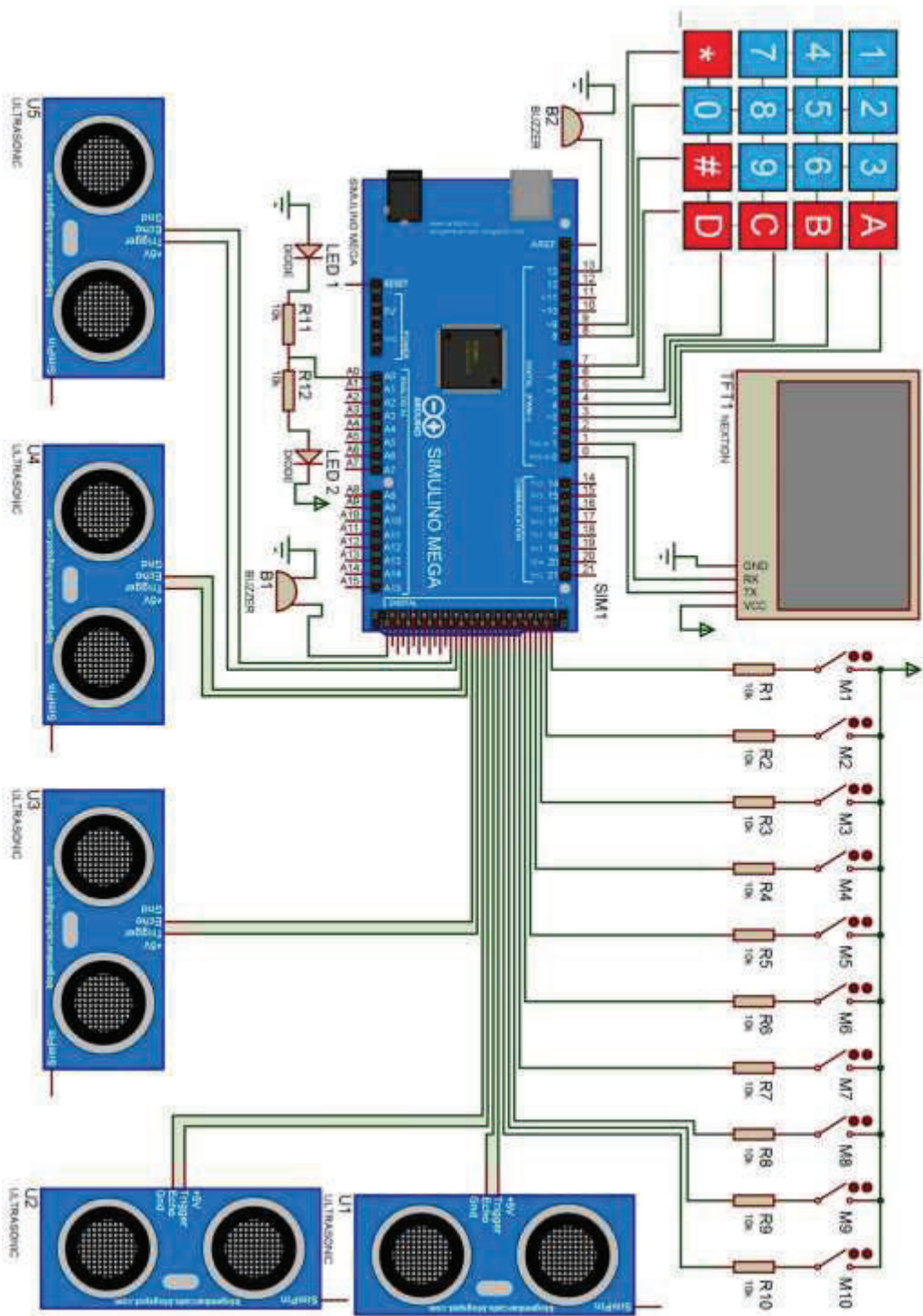


Figura 2.17. Diagrama de conexiones

## 2.5. Implementación del sistema en el modelo a escala del restaurante

En esta sección se muestra la instalación de los diferentes dispositivos electrónicos utilizados, tales como: sensores, buzzer y el Mando Central del Sistema en el modelo a escala del Restaurante, tomando en cuenta el estudio técnico anteriormente realizado.

## 2.6. Instalación de sensores magnéticos

Se instalaron diez sensores magnéticos, de acuerdo con lo especificado en la Tabla 2.2. Como este elemento consta de dos partes: transmisor e imán, se colocará una de las partes en puertas y ventanas. Mientras que la parte del imán será colocada en el marco de las mismas, de manera que al abrirse, se separan los dos elementos y se activará el sistema. En la Figura 2.18 se muestra la colocación y fijación del imán dentro del marco de las ventanas y puertas. Mientras que en la Figura 2.19, se muestra la fijación de los sensores.



Figura 2.18. Colocación del imán en puertas y ventanas.



Figura 2.19. Fijación del sensor magnético

## 2.7. Instalación de sensores ultrasónicos

Se instalaron cinco sensores ultrasónicos, de acuerdo a la Tabla 2.2, estos sensores serán los encargados de cubrir el interior del lugar en caso de que exista intrusión por ruptura de vidrio o perforación de paredes, así como en el caso de que exista presencia de alguien mientras el sistema se encuentra activado. En la Figura 2.20 se muestra la instalación del sensor ultrasónico en una pared de la maqueta.



Figura 2.20. Instalación de sensor ultrasónico

## 2.8. Instalación de buzzer

En la Figura 2.21 se muestra la instalación de un buzzer para dar la señal de alerta sobre la existencia de alguna anomalía dentro del local comercial.

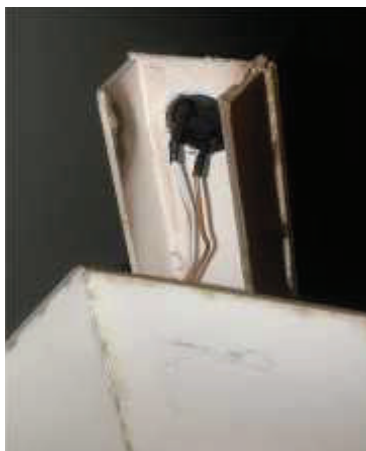


Figura 2.21. Instalación del buzzer

## 2.9. Acoplamiento de sensores a módulo Arduino

Toda la interconexión de los sensores con el módulo Arduino se realiza mediante cableado y se fija en la base de la maqueta. En la Figura 2.22, se observa el cableado de los sensores magnéticos y de ultrasonido.

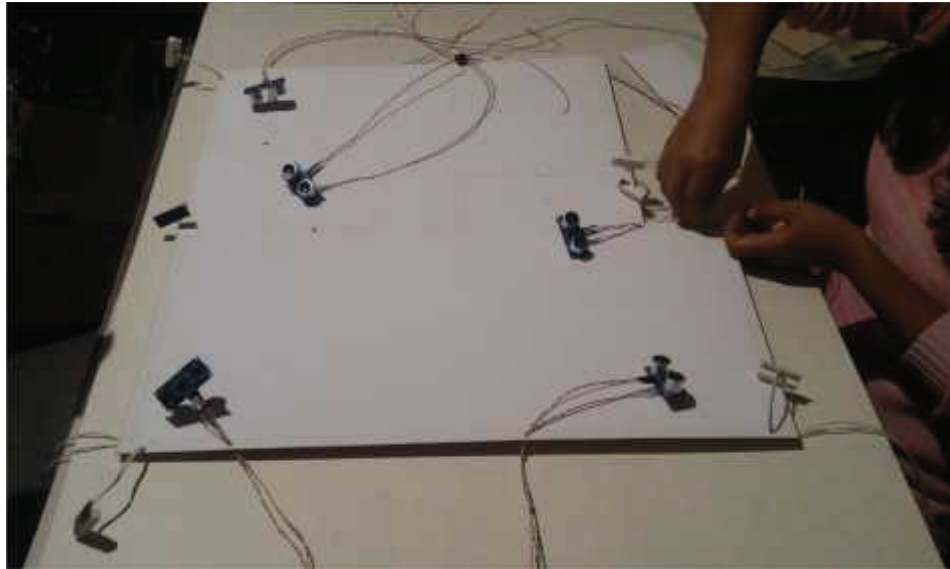


Figura 2.22. Cableado de sensores

## 2.10. Instalación del Mando Central del Sistema

La placa de Arduino conjuntamente con el teclado y la pantalla se instalaron en un lugar visible del prototipo para la demostración, tal como se muestra en la Figura 2.23.



Figura 2.23. Instalación Mando Central

## CAPÍTULO 3

### PRUEBAS Y RESULTADOS

Para evaluar los resultados obtenidos con el sistema se realizó varias pruebas en el mismo. Primero se efectuaron pruebas en hardware y posteriormente en software, tanto del módulo Arduino como del HMI, del sistema de seguridad electrónica implementado en la maqueta.

#### 3.1. Pruebas de Hardware y Software

Mediante el uso de sensores implementados en el sistema de seguridad dentro del modelo a escala del Restaurante, se efectúa pruebas de funcionamiento de:

- Sensores magnéticos
- Sensores ultrasónicos
- Teclado matricial
- Módulo Arduino
- Pantalla Nextion

Mediante las señales que provengan de los sensores del sistema de alarma, el bloque del mando central comenzará con su funcionamiento.

Estas pruebas se realizaron con cada uno de los sensores magnéticos colocados en puertas y ventanas, si se abre un sensor determinado el sistema da una aviso de alerta sonora y visual en la pantalla. Así como cuando se olvide de cerrar correctamente alguna puerta o ventana.

En la Figura 3.1, se observa que tanto una puerta como ventana están abiertas, por lo que en la pantalla se visualiza en PUERTAS “ALERTA ABIERTA” y en VENTANAS “ALERTA ABIERTA”.





Figura 3.1. Pantalla visualiza mensaje cuando una puerta y ventana están abiertas.

Para el caso en el que las puertas y ventanas estén cerradas, en la pantalla solamente se visualizará “PUERTAS” y “VENTANAS”, sin ningún mensaje adjunto de aviso de que una de ellas esté abierta, como se muestra en la Figura 3.2.



Figura 3.2. Pantalla visualiza solo PUERTAS y VENTANAS sin mensaje de alerta.

Los sensores ultrasónicos colocados en las paredes de la maqueta son configurados a las distancias adecuadas, para que al momento que el sistema está activado o desactivado emitan una señal; es decir, éstos estarán monitorizando a cada momento el lugar, dando solamente un aviso sonoro si detectan interrupción en sus distancias fijadas. Esto se puede observar en la misma pantalla de programación del módulo Arduino donde muestra las lecturas de distancias, así como la comunicación con la pantalla para la emisión del mensaje.

En la Figura 3.3, se muestra la captura de pantalla con las lecturas de las distancias de los sitios en los que se colocaron los sensores ultrasónicos y la comunicación con la pantalla Nextion.

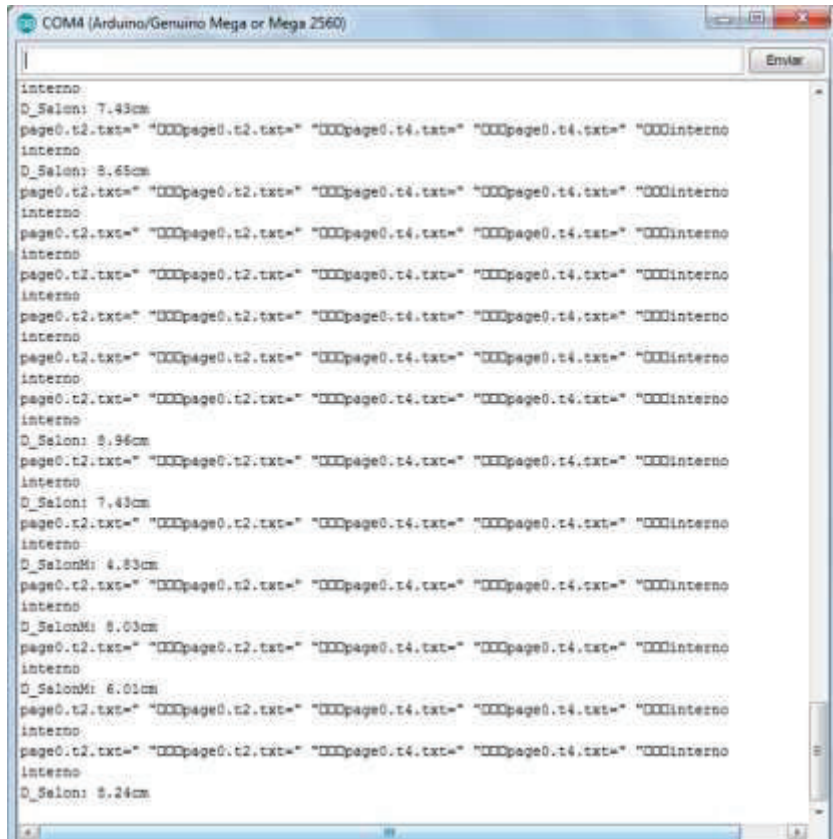


Figura 3.3. Lectura sensor ultrasónico

El funcionamiento y transmisión de las señales que ingresan al módulo Arduino mediante comunicación serial a la pantalla Nextion, se verifica por medio del led incorporado en la placa del módulo, el cual se enciende cuando detecta transmisión de datos. En este caso, como la velocidad de transmisión empleada es 9600 baudios, el led empieza a titilar, indicando que existe transferencia de información entre ambos dispositivos, como se puede observar en la Figura 3.4.



Figura 3.4. Encendido del led de la placa Arduino.

## 3.2. Resultados

Luego de realizado el estudio técnico y de los elementos existentes dentro del mercado para un sistema de alarmas, se implementó un prototipo de seguridad que emplea dispositivos de apertura, detección y alarma, mismos que fueron acoplados al módulo Arduino.

El sistema implementado permite mediante el ingreso de una clave, que se la realiza por medio de un teclado matricial, activar y asegurar el sitio, acción que se verifica mediante el encendido de un led de color rojo (LED 1), como se muestra en la Figura 3.5.



Figura 3.5. Sistema activado, led rojo encendido

De la misma forma, por medio del teclado matricial se desactiva el sistema para el ingreso al lugar, se enciende el led de color verde (LED 2), como se muestra en la Figura 3.6.



Figura 3.6. Sistema desactivado, led verde encendido.

## CAPÍTULO 4

### CONCLUSIONES Y RECOMENDACIONES

#### 4.1. Conclusiones

- El proyecto de implementación de un prototipo de sistema de seguridad temporizado para puertas y ventanas, puede ser implementado para pequeños locales que empiecen su funcionamiento, ya que no requiere de mucha inversión económica.
- El desarrollo de sistemas basados en módulos Arduino es bastante simple. Actualmente, es muy fácil encontrar estos módulos en el mercado local a un costo razonable
- La depuración de los programas en el módulo Arduino, en particular de las rutinas de comunicación con cualquier tipo de dispositivo que se le acople, se facilita por el encendido y titilación del led incorporado en la placa Arduino, que nos permite verificar el intercambio y procesamiento de las señales.
- La comunicación entre el microcontrolador y los dispositivos electrónicos, sensores y pantalla se facilita porque la programación se la realiza en el lenguaje propio de Arduino, mismo que es muy similar al lenguaje C.
- Al inicio del desarrollo del proyecto se iba a controlar la activación y desactivación del sistema con una pantalla táctil; sin embargo, por problemas de comandos en el Arduino no se logró la comunicación entre estos dispositivos, lo cual hizo que se deba utilizar un teclado para este fin.
- Se puede mejorar fácilmente el sistema implementado, ya que al estar basado en un módulo Arduino, para la incorporación de nuevas funcionalidades bastará con cambiar la programación de dicho módulo.

#### 4.2. Recomendaciones

- El módulo Arduino empleado en el sistema debe trabajar con un voltaje continuo de 5 voltios, para grabar el programa en su microcontrolador.
- El módulo Arduino no debe recibir un voltaje mayor a 12 V ni inferior a 7 V cuando se emplea una fuente externa, esto podría causar graves daños.

- Aplicar correctamente la ecuación para el cálculo de las distancias que intervienen en la medición de los sensores ultrasónicos, puesto que eso facilita la calibración adecuada de los sensores y así el sistema funcione como se requiere.
- Es recomendable trabajar a 9600 baudios como velocidad de comunicación, ya que sino esto causa dificultades al momento de la transmisión y recepción de los datos.
- Para establecer una aplicación real de este sistema, se deberá emplear sensores de presencia infrarrojos.

## BIBLIOGRAFÍA

- [1] Tutorial Sistemas Electrónicos de Seguridad. (2015, Mayo) Tecnologías de la Seguridad. [Online]. <http://serviciostc.com/category/sistemas-intrusion/>
- [2] Isacc P.E. (2013) Ciencia y Tecnología. [Online]. <http://architecnologia.blogspot.com.es/>
- [3] Monk Simon,. Madrid, España: Estribor, 30 Proyectos con Arduino.
- [4] Torrente Artero O., Arduino Curso Práctico de Formación , Primera ed. México: Alfaomega, 2013.
- [5] Arduino: Tecnología para todos. Tecnología para todos. [Online]. <http://arduinodhtics.weebly.com/historia.html>
- [6] REYES MENA D., "DISEÑO DE LA RED DE SENSORES INALÁMBRICOS MEDIANTE LA EVALUACIÓN DEL DESEMPEÑO DE LAS PLATAFORMAS ARDUINO Y WASPMOTE BAJO EL PROTOCOLO DE COMUNICACIONES ZIGBEE PARA EL MONITOREO Y CONTROL DE LOS EMBALSES DE AGOYÁN Y PISAYAMBO", 2014, ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO.
- [7] Culturacion. Culturacion. [Online]. <http://culturacion.com/que-son-y-como-funcionan-las-pantallas-tactiles/>
- [8] Ling Adrián. (2015, Noviembre) UNOCERO. [Online]. <https://www.unocero.com/2015/11/11/como-funciona-una-pantalla-tactil/>
- [9] ARDUINO. (2017) ARDUINO. [Online]. <https://www.arduino.cc/en/Main/ArduinoBoardMega2560>
- [10] Racoon Team. (2013) Microelectrónica Programable HD. [Online]. <http://6im7rt.blogspot.com/2013/04/v-behaviorurldefaultvmlo.html>
- [11] Ortega Edwin. (2014, junio) SlideShare. [Online]. [https://es.slideshare.net/hernancardenas9400/clasificacion-de-sensores-36167798?next\\_slideshow=1](https://es.slideshare.net/hernancardenas9400/clasificacion-de-sensores-36167798?next_slideshow=1)
- [12] 5Hertz Electronica. (2014, Abril) 5Hertz Electronica. [Online]. <http://5hertz.com/tutoriales/?p=623>
- [13] Fast tecnología y calidad. Fast tecnología y calidad. [Online]. <http://dfast.cl/sensores/113-sensor-magnetico-reed-switch.html>

- [14] Torrente Artero Oscar, "ARDUINO CURSO PRACTICO DE FORMACIÓN," in *ARDUINO CURSO PRACTICO DE FORMACIÓN*. México: Alfaomega, 2013, pp. 440-451.
- [15] Techmake. Techmake. [Online]. <http://www.techmake.com/tutorialhcsr04>
- [16] Intelligent Systems Co.Ltd. (2017) ITEAD C.C. [Online]. <https://www.itead.cc/nextion-nx4024k032.html>
- [17] Max Electrónica. Max Electrónica potenciamos tus proyectos. [Online]. <http://www.maxelectronica.cl/sensores/113-sensor-magnetico-reed-switch.html>
- [18] Córdova S. Valladolid D., ALARMA PARA EL HOGAR Y PEQUEÑOS NEGOCIOS CON ENLACE INALÁMBRICO A LA CENTRAL DE MONITOREO, MARZO 2010, UNIVERSIDAD AUTONOMA DE CIUDAD JÚAREZ.
- [19] Crnl. López P. Crnl. Chafra M., REDUCCIÓN DEL ÍNDICE DELICUENCIAL DEL CANTÓN RUMIÑAHUI-SANGOLQUÍ, 2013, <http://repositorio.espe.edu.ec/bitstream/21000/8132/1/T-ESPE-047684.pdf>.
- [20] El Telégrafo, "Policía instalará 10 sistemas de botones de auxilio en el cantón Rumiñahui," Abril 2014, <http://www.eltelegrafo.com.ec/noticias/justicia/1/policia-instalara-10-sistemas-de-botones-de-auxilio-en-el-canton-ruminahui>. [Online]. <http://www.eltelegrafo.com.ec/noticias/justicia/1/policia-instalara-10-sistemas-de-botones-de-auxilio-en-el-canton-ruminahui>
- [21] Mendoza F., "No hay coordinación contra la inseguridad," *Ultimas noticias*, Feb. 2013, <http://www.ultimasnoticias.ec/noticias/13154-no-hay-coordinacion-contrala-inseguridad.html>.
- [22] Molina Jose Luis. (2002-2017) [www.profesormolina.com.ar](http://www.profesormolina.com.ar). [Online]. <http://www.profesormolina.com.ar/index.htm>
- [23] Duarte Quintana Pilar. Pilar Duarte Quintana Tecnologia. [Online]. <http://pilarduartetecnologia.blogspot.com>
- [24] Toasa S. Alomoto E., Construcción de una alarma utilizando Microcontroladores, dispositivos inalambricos y automarcado telefónico para la Iglesia Renuevo de Sangolquí, 2013, <http://bibdigital.epn.edu.ec/bitstream/15000/5615/1/CD-4642.pdf>.
- [25] Gaceta educativa - Lostipos.com. Equipo de redacción. (2015, Septiembre) Los tipos.com. [Online]. <http://www.lostipos.com/de/sensores.html>



## **ANEXOS**

ANEXO A: Manual de usuario

ANEXO B: Programa del sistema de seguridad

ANEXO C: Datos técnicos módulo Arduino Mega 2560

ANEXO D: Datos técnicos sensor ultrasónico

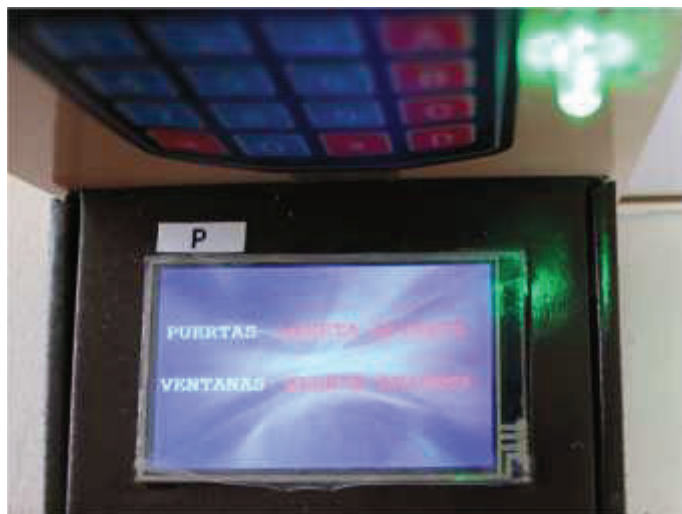
ANEXO E: Datos técnicos teclado matricial

## ANEXO A: Manual de usuario

1. Ingreso de clave de activación: pulsar una contraseña de cuatro dígitos, para que el sistema quede activado, mismo que se indica por el encendido de un led rojo, y en la pantalla no se visualiza ninguna advertencia.



2. Si en caso de pulsar más de cuatro dígitos el sistema se bloquea, para ello pulsar cuatro veces la tecla "D" para resetear el número de pulsos y así poder volver a ingresar la contraseña.
3. Para desactivar el sistema ingresar la misma clave de cuatro dígitos, se enciende el led verde queda encendido.
4. En el caso de estar el sistema activado led rojo encendido y exista una intrusión en el sistema, se dará un aviso sonoro y visual. Para lo cual primero se procederá desactivando la alarma, encendido led verde, para visualizar en la pantalla si son las ventanas o puertas las que se abrieron.



**ANEXO B:** Programa del sistema de seguridad elaborado en la plataforma Arduino.

```
#include <EEPROM.h> // Librería EEPROM incluida.
#include <Keypad.h> //llama a la librería Keypad.h

char contraseña[]="1234"; //Aquí se escribe la contraseña de 4 dígitos
char código="DDDD"; //Cadena donde se guardarán los caracteres de las teclas
presionadas
int position=0; //variable que se incrementará al presionar las teclas
const byte ROWS = 4; //Número de filas del teclado que se está usando
const byte COLS = 4; //Número de columnas del teclado que se está usando

//DECLARACIÓN DE SENSORES MAGNÉTICOS
int puerta1=22, puerta2=23, puerta3=24, ventana1=25, ventana2=26, ventana3=27;
int ventana4=28, ventana5=29, ventana6=30, ventana7=31;

//DECLARACIÓN DE SENSORES ULTRASÓNICOS
int Trigger1=32; int Echo1=33; int Trigger2=34; int Echo2=35; int Trigger3=36;
int Echo3=37; int Trigger4=38; int Echo4=39; int Trigger5=40; int Echo5=41;
//DECLARACIÓN DE VARIABLES
float tiempo1, distancia1; float tiempo2, distancia2; float tiempo3, distancia3;
float tiempo4, distancia4; float tiempo5, distancia5; byte sirena=53; byte buzzer=13;
char hexaKeys[ROWS][COLS] = //Aquí se pondrá la disposición de los caracteres tal
cual están en nuestro teclado
{
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'}
};

byte rowPins[ROWS] = {2, 3, 4, 5}; //Se selecciona los pines en el Arduino donde irán
conectadas las filas
byte colPins[COLS] = {6, 7, 8, 9}; //Se selecciona los pines en el Arduino donde irán
conectadas las columnas
Keypad customKeypad = Keypad(makeKeymap(hexaKeys), rowPins, colPins, ROWS,
COLS); //inicializa el teclado
byte leds=A0; byte estado=0; //Pin de salida para los leds de estado de la alarma.
byte cntPul=0; //Contador de pulsaciones de botón del teclado
char recibirDato; char key; boolean claveBien=false; boolean claveMal=false;
```

```

void setup()
{
  pinMode (puerta1,INPUT); pinMode (puerta2,INPUT); pinMode (puerta3,INPUT);
  pinMode (ventana1,INPUT); pinMode (ventana2,INPUT);
  pinMode (ventana3,INPUT); pinMode (ventana4,INPUT);
  pinMode (ventana5,INPUT); pinMode (ventana6,INPUT);
  pinMode (ventana7,INPUT);
  pinMode(Echo1,INPUT); pinMode(Trigger1,OUTPUT); pinMode(Echo2,INPUT);
  pinMode(Trigger2,OUTPUT); pinMode(Echo3,INPUT);
  pinMode(Trigger3,OUTPUT); pinMode(Echo4,INPUT);
  pinMode(Trigger4,OUTPUT); pinMode(Echo5,INPUT);
  pinMode(Trigger5,OUTPUT); pinMode(sirena,OUTPUT);
  pinMode(leds, OUTPUT); //Pin revicion de sensores
  pinMode(buzzer,OUTPUT);
  Serial.begin(9600); //inicializar puerto serie
  Serial.print("0");

}
void loop()
{
  Ultrasonico1(); Ultrasonico2(); Ultrasonico3(); Ultrasonico4(); Ultrasonico5();
  int Puerta1=digitalRead(puerta1); int Puerta2=digitalRead(puerta2);
  int Puerta3=digitalRead(puerta3); int Ventana1=digitalRead(ventana1);
  int Ventana2=digitalRead(ventana2); int Ventana3=digitalRead(ventana3);
  int Ventana4=digitalRead(ventana4); int Ventana5=digitalRead(ventana5);
  int Ventana6=digitalRead(ventana6); int Ventana7=digitalRead(ventana7);

while(estado==0){
  puertoSerie();
  digitalWrite(leds,LOW); digitalWrite(sirena,LOW);
  if(digitalRead (puerta1)==LOW||digitalRead(puerta2)==LOW||digitalRead(puerta3)==LOW)
  Serial.print("page1.t2.txt=");
  Serial.write(0x22);
  Serial.print("ABIERTAS");
  Serial.write(0x22);
  Serial.write(0xff);
  Serial.write(0xff);
  Serial.write(0xff);
}
else
{
  printPuertaCerrada();
}
}
/*/ digitalWrite(13,HIGH);

```

```

if(Puerta1==LOW||Puerta2==LOW||Puerta3==LOW){
printPuertaA();
Serial.println("Alarma Activada");}
else{
    printPuertaB();
    Serial.println("Casa Segura");*/
    }
if(digitalRead (ventana1) == LOW || digitalRead (ventana2) == LOW || digitalRead
(ventana3) == LOW || digitalRead (ventana4) == LOW || digitalRead (ventana5) == LOW
||digitalRead (ventana6) == LOW || digitalRead (ventana7) == LOW)
{
    //printVentanaA();
    //Serial.println("Alarma Activada");
Serial.print("page1.t6.txt=");
Serial.write(0x22);
Serial.print("ABIERTAS");
Serial.write(0x22);
Serial.write(0xff);
Serial.write(0xff);
Serial.write(0xff);
    }
    else
    {
printVentanaCerrada();
//Serial.println("Casa Segura");
    }
else{
printPuertaCerrada();
printVentanCerrada();
void clavePuk()
clavePin();
if(claveBien||claveMal||(Serial.available())>0){
key=keypad.getKey();
If(key=='*'||key=='#'){
char customKey = customKeypad.getKey(); //se guarda en la variable customKey el
caracter de la tecla presionada
if (customKey != NO_KEY) //se evalúa si se presionó una tecla
{
codigo[cont]=customKey; //se guarda caracter por caracter en el arreglo codigo[]
Serial.print(codigo[cont]); //se imprime en el puerto serie la tecla presionada
cont=cont+1;
Serial.println(cont); //se incrementa la variable cont
if(cont==4) //si ya fueron presionadas 4 teclas se evalúa si la contraseña es correcta
{

```

```

cont=0;
if(codigo[0]==contrasena[0]&&codigo[1]==contrasena[1]&&codigo[2]==contrasena[2]&
&codigo[3]==contrasena[3]) {
digitalWrite(13,LOW);
digitalWrite(Act_Alarma,HIGH);}
else if(customKey=='D')
{
digitalWrite(13,LOW);
digitalWrite(Act_Alarma,LOW);
cont=0; //resetear a 0 la variable cont
}
}
If(Serial.available(>0){
recibirDato=Serial.read();
if(recibirDato=='1')
int interno=analogRead(A0);
if(interno==0){
Serial.println("interno");
}else{
if(Puerta1==LOW||Puerta2==LOW||Puerta3==LOW){
flash(20);
printPuertaAbierta();
Serial.println("Alarma Activada");
}else{
printPuertaB();
//Serial.println("Casa Segura");
}
if(interno==0){
Serial.println("interno");
}
else{
if(Ventana1==LOW||Ventana2==LOW||Ventana3==LOW||Ventana4==LOW||Ventana5
==LOW||Ventana6==LOW||Ventana7==LOW)
{
flash(20);
printVentanaAbierta();
//Serial.println("Alarma Activada");
}else{
printVentanaCerrada();
//Serial.println("Casa Segura");
}
}
}

```

```

//_____FUNCIONES EXTERNAS

void flash (int duracion)
{
    digitalWrite(sirena,HIGH); //Prende el LED
    delay(duracion); //Tiempo
    digitalWrite(sirena,LOW); //Apaga el LED
    delay(duracion); //Tiempo
}

void printPuertaAbierta()
{
    for(int a=0; a<2; a++)
    {
        Serial.print("page1.t2.txt="); // variable de la pantalla nextion
        Serial.write(0x22);
        Serial.print("ABIERTAS"); //solo manda el simbolo de porcentaje
        Serial.write(0x22);
        Serial.write(0xff);
        Serial.write(0xff);
        Serial.write(0xff);
    }
}

void printPuertaCerrada()
{
    for(int a=0; a<2; a++)
    {
        Serial.print("page0.t2.txt="); // variable de la pantalla nextion
        Serial.write(0x22);
        Serial.print(" "); //solo manda el simbolo de porcentaje
        Serial.write(0x22);
        Serial.write(0xff);
        Serial.write(0xff);
        Serial.write(0xff);
    }
}

void printVentanaAbierta()
{
    for(int a=0; a<2; a++)
    {
        Serial.print("page1.t6.txt="); // variable de la pantalla nextion
        Serial.write(0x22);
        Serial.print("ABIERTAS"); //solo manda el simbolo de porcentaje
        Serial.write(0x22);
        Serial.write(0xff);
    }
}

```

```

Serial.write(0xff);
Serial.write(0xff);
}
}
void printVentanaCerrada()
{
for(int a=0; a<2; a++)
{
Serial.print("page1.t6.txt="); // variable de la pantalla nextion
Serial.write(0x22);
Serial.print(" "); //solo manda el simbolo de porcentaje
Serial.write(0x22);
Serial.write(0xff);
Serial.write(0xff);
Serial.write(0xff);
}
}
void Ultrasonico1(){
digitalWrite(Trieger1,LOW);
delayMicroseconds(2);
digitalWrite(Trieger1,HIGH);
delayMicroseconds(10);
digitalWrite(Trieger1,LOW);
tiempo1=pulseIn(Echo1,HIGH);
distancia1=(tiempo1/2)/29.2;
if(distancia1>0 && distancia1<=7)
{
Serial.print("D_Parrilla: ");
Serial.print(distancia1);
Serial.println("cm");
delay(500);
estado=3;
Serial.print("3");
//for(int a=0; a<10; a++)
Delay(1000);
flash(50);
}
}
}
void Ultrasonico2(){
digitalWrite(Trieger2,LOW);
delayMicroseconds(2);
digitalWrite(Trieger2,HIGH);
delayMicroseconds(10);

```



```

    digitalWrite(Trigger2,LOW);
    tiempo2=pulseIn(Echo2,HIGH);
    distancia2=(tiempo2/2)/29.2;
    if(distancia2>0 && distancia2<=5)
    {
        Serial.print("D_Principal: ");
        Serial.print(distancia2);
        Serial.println("cm");
        delay(500);
        digitalWrite(altavoz,HIGH);
        delay(500);
        digitalWrite(altavoz,LOW);
        delay(500);
        //for(int a=0; a<10; a++)
        //flash(50);
    }
}

void Ultrasonico3(){
    digitalWrite(Trigger3,LOW);
    delayMicroseconds(2);
    digitalWrite(Trigger3,HIGH);
    delayMicroseconds(10);
    digitalWrite(Trigger3,LOW);
    tiempo3=pulseIn(Echo3,HIGH);
    distancia3=(tiempo3/2)/29.2;
    if(distancia3>0 && distancia3<=7)
    {
        Serial.print("D_SalonF: ");
        Serial.print(distancia3);
        Serial.println("cm");
        delay(500);
        estado=3;
        Serial.print("3");
        Delay(1000);
        //for(int a=0; a<10; a++)
        //flash(50);
    }
}

void Ultrasonico4(){
    digitalWrite(Trigger4,LOW);
    delayMicroseconds(2);
    digitalWrite(Trigger4,HIGH);
    delayMicroseconds(10);

```

```

    digitalWrite(Triple4,LOW);
    tiempo4=pulseIn(Echo4,HIGH);
    distancia4=(tiempo4/2)/29.2;
    if(distancia4>0 && distancia4<=9)
    {
    Serial.print("D_SalonM: ");
    Serial.print(distancia4);
    Serial.println("cm");
    delay(500);
    estado=3;
    Serial.print("3");
    Delay(1000);
    //for(int a=0; a<10; a++)
    //flash(50);
        }
    }
}
void Ultrasonico5(){
    digitalWrite(Triple5,LOW);
    delayMicroseconds(2);
    digitalWrite(Triple5,HIGH);
    delayMicroseconds(10);
    digitalWrite(Triple5,LOW);
    tiempo5=pulseIn(Echo5,HIGH);
    distancia5=(tiempo5/2)/29.2;
    if(distancia5>0 && distancia5<=5.5)
    {
    Serial.print("D_Salon: ");
    Serial.print(distancia5);
    Serial.println("cm");
    delay(500);
    estado=3;
    Serial.print("3");
    Delay(1000);
    //for(int a=0; a<10; a++)
    //flash(50);
        }
    }
}
//_____

```