



**“Implementación de un sistema automático para determinar el aforo
actual en el Restaurante Sr. Pincho”**

Moreta Manzano, Stalin Fabricio

Departamento de Eléctrica y Electrónica

Carrera de Tecnología Superior en Automatización e Instrumentación

Monografía, previo a la obtención del título de Tecnólogo Superior en
Automatización e Instrumentación

Ing. Guerrero Rodríguez, Lucía Eliana

Latacunga, 07 de abril del 2021



DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE TECNOLOGÍA SUPERIOR EN AUTOMATIZACIÓN E INSTRUMENTACIÓN

CERTIFICACIÓN

Certifico que la monografía, “**Implementación de un sistema automático para determinar el aforo actual en el restaurante Sr. Pincho**” fue realizado por el señor **Moreta Manzano, Stalin Fabricio**, la cual ha sido revisada y analizada en su totalidad por la herramienta de verificación de similitud de contenido; por lo tanto cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, razón por la cual me permito acreditar y autorizar para que lo sustente públicamente.

Latacunga, 06 de abril del 2021



Plumado, electrónico y firmado por:
LUCIA ELIANA
GUERRERO
RODRIGUEZ

Ing. Guerrero Rodríguez, Lucía Eliana

C. C.: 0501878649



Document Information

Analyzed document	MONOGRAFÍA_SR_MORETA_STALIN.docx (D100831533)
Submitted	4/7/2021 4:39:00 AM
Submitted by	Guerrero Rodriguez Lucia Eliana
Submitter email	leguerrero6@espe.edu ec
Similarity	5%
Analysis address	leguerrero6.espe@analysis.arkund.com

Sources included in the report

W	URL: https://blog.educacionit.com/2018/08/27/arduino-y-raspberry-pi-que-son-y-cuales-so ... Fetched: 4/7/2021 4:40:00 AM	 1
W	URL: https://www.xataka.com/basics/que-arduino-como-funciona-que-puedes-hacer-uno Fetched: 4/7/2021 4:40:00 AM	 6



Escanea el código QR para:
LUCIA ELIANA
GUERRERO
RODRIGUEZ

ING. GUERRERO RODRIGUEZ, LUCIA ELIANA
C.C. 0501878649



DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA
CARRERA DE TECNOLOGÍA SUPERIOR EN AUTOMATIZACIÓN E
INSTRUMENTACIÓN

RESPONSABILIDAD DE AUTORÍA

Yo, **Moreta Manzano, Stalin Fabricio**, con cédula de ciudadanía N° **1803384765**, declaro/declaramos que el contenido, ideas y criterios de la monografía: **“Implementación de un sistema automático para determinar el aforo actual en el restaurante Sr. Pincho”** es de mi autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos, y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Latacunga, 07 de abril del 2021

A handwritten signature in blue ink, reading 'Stalin Fabricio Moreta Manzano', is written over a horizontal line. The signature is written in a cursive style.

Moreta Manzano, Stalin Fabricio

C.C.: 1803384765



**DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA
CARRERA DE TECNOLOGÍA SUPERIOR EN AUTOMATIZACIÓN E
INSTRUMENTACIÓN**

AUTORIZACIÓN DE PUBLICACIÓN

Yo, **Moreta Manzano, Stalin Fabricio** autorizo a la Universidad de las Fuerzas Armadas ESPE publicar la monografía: **“Implementación de un sistema automático para determinar el aforo actual en el restaurante Sr. Pincho”** en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi responsabilidad.

Latacunga, 03 de marzo del 2021

Moreta Manzano, Stalin Fabricio

C.C.: 1803384765

DEDICATORIA

A mis padres por haberme forjado como la persona que soy en la actualidad, con cada uno de los valores que he demostrado a lo largo de este proyecto de superación personal como profesional, muchos de mis logros se los debo a ustedes entre los que se incluye este, ya que aquí se ve reflejado su esfuerzo diario por siempre darme el apoyo incondicional.

Me formaron con valores desde las primeras instancias, además de reglas y con algunas libertades, siempre enseñándome la distinción de lo bueno y lo malo, al final de cuentas todo lo aportado en mi de todas las formas está dando fruto.

Me dieron la motivación constantemente sin descanso para llegar al objetivo que todos tenemos, por ello eh logrado alcanzar mis anhelos. A las personas que siempre de una u otra manera han sabido brindarme su apoyo de todas las maneras posibles de realizar el objetivo planteado.

MORETA MANZANO, STALIN FABRICIO

AGRADECIMIENTO

Gracias a Dios por permitirme realizar este objetivo, tener y disfrutar a mi familia, ya que ha sido de gran apoyo en las decisiones tomadas en mi formación personal como profesional.

El camino siempre ha sido lleno de dificultades, obstáculos, pero a la final lo importante es el resultado final, pero sin lugar a dudas, gracias al amor, bondad, apoyo, lo complicado se ha convertido en algo sencillo.

A todos los docentes quienes han sabido impartir la catedra de cada nivel, como consiguiente a sus enseñanzas se refleja el presente trabajo.

No ha sido sencillo el camino hasta ahora, pero gracias a sus aportes, a su amor, a su inmensa bondad y apoyo, lo complicado de lograr esta meta se ha notado menos. Les agradezco y hago presente mi gran afecto hacia ustedes, mi hermosa familia.

MORETA MANZANO, STALIN FABRICIO

Tabla de contenidos	
Carátula.....	1
Certificación.....	2
Urkund.....	3
Responsabilidad de autoría.....	4
Autorización de publicación.....	5
Dedicatoria.....	6
Agradecimiento.....	7
Tabla de contenidos.....	8
Índice de tablas.....	11
Índice de figuras.....	12
Resumen.....	14
Abstrac.....	15
Generalidades.....	16
Introducción.....	16
Antecedentes.....	16
Justificación e importancia.....	18
Planteamiento del problema.....	18
Objetivos.....	19
<i>Objetivo general.....</i>	19
<i>Objetivos específicos.....</i>	19

Alcance	20
Marco Teórico	21
<i>Arduino</i>	21
Sensor E18-D80NK.....	26
<i>Circuito integrado ULN 2803 A</i>	27
Optoacoplador 4N25	29
Regulador de voltaje LM7805	31
Transistor NPN 2n3904	33
Diodos LED.....	35
Condensadores	37
Resistencia eléctrica.....	39
Desarrollo del proyecto	43
Desarrollo	43
Programación	43
Diseño del display indicador	61
Conexión Arduino – ULN2803A – Display Indicador	64
Conexión Arduino – Sensores	65
Montaje de elementos en la placa.....	68
Comprobación del funcionamiento.....	69
Conclusiones y recomendaciones	74
Conclusiones.....	74

Recomendaciones.....	74
Bibliografía.....	74
Anexos	76

Índice de tablas

Tabla 1 <i>Especificaciones técnicas del Arduino Mega</i>	25
Tabla 2 <i>Especificaciones técnicas sensor E18-D80NK</i>	27
Tabla 3 <i>Especificaciones técnicas ULN 2803</i>	29
Tabla 4 <i>Especificaciones técnicas 4N25</i>	31
Tabla 5 <i>Especificaciones técnicas LM7805</i>	33
Tabla 6 <i>Especificaciones técnicas del transistor 2N3904</i>	34
Tabla 7 <i>Especificaciones de voltaje de LEDES</i>	37
Tabla 8 <i>Código de colores para resistencias</i>	41

Índice de figuras

Figura 1	<i>Ilustración placa Arduino Mega 2560 REV 3.....</i>	23
Figura 2	<i>Descripción de entradas y salidas de Arduino Mega.....</i>	24
Figura 3	<i>Representación física Arduino E18-D80NK y disposición de pines.....</i>	26
Figura 4	<i>Disposición de pines ULN 2803.....</i>	28
Figura 5	<i>Disposición de pines 4N25</i>	30
Figura 6	<i>Disposición de pines LM7805.....</i>	32
Figura 7	<i>Disposición de pines 2N3904.....</i>	34
Figura 8	<i>Simbología y polaridad diodo led.....</i>	36
Figura 9	<i>Representación física de condensadores</i>	38
Figura 10	<i>Representación simbólica.....</i>	39
Figura 11	<i>Representación física de una resistencia.....</i>	40
Figura 12	<i>Especificación de las bandas de la resistencia</i>	41
Figura 13	<i>Diseño de display indicador</i>	61
Figura 14	<i>Diseño de pistas para display indicador.....</i>	62
Figura 15	<i>Diseño y montaje de los Leds en la placa de baquelita.....</i>	63
Figura 16	<i>Parte posterior display 7 segmentos.....</i>	64
Figura 17	<i>Conexión Arduino – Circuito integrado ULN2803A–Display Indicador...65</i>	65
Figura 18	<i>Conexión Arduino a sensores E18 – D80NK</i>	66
Figura 19	<i>Conexión de Arduino a LCD</i>	67
Figura 20	<i>Diseño de pistas para LCD – Arduino, y sistema de control.....</i>	68
Figura 21	<i>Montaje de los elementos en la placa madre</i>	69
Figura 22	<i>Visualización de funcionamiento.....</i>	70
Figura 23	<i>Representación display indicador</i>	71
Figura 24	<i>Indicador de número máximo establecido.....</i>	72
Figura 25	<i>Representación display indicador</i>	72

Figura 26 *Diseño previo del panel de control*..... 73

Resumen

El presente proyecto investigativo y práctico tiene como objetivo implementar un sistema automático para determinar el aforo actual en el restaurante Sr. Pincho". Se utilizó una placa Arduino MEGA de 54 pines digitales que funcionan como entrada y salida, donde se recibirá la señal de dos sensores de proximidad infrarrojos ajustables E18-D80K (S1, S2), que detectan objetos sin necesidad de contacto en un rango de hasta 80cm. Estos sensores están colocados en la entrada del restaurant para el conteo descendente y ascendente del aforo. La salida del sensor cumple la siguiente secuencia: primera instancia S1 se activa, segunda instancia S1, S2 activados, tercera instancia S1 desactivado, S2 desactivado cuenta el número de persona a la entrada (conteo ascendente). El conteo descendente ocurre de manera contraria a la descrita. La información es presentada en un panel conformado por dos display de 7 segmentos ánodo común y luces indicadoras de color verde para continuar y rojo detenerse, dependiendo del aforo permitido cuando se sobrepase sonará una alarma que no se desactivará al menos que la cantidad de personas baje, este valor se puede modificar mediante un teclado que tiene la opción de restablecer el set point elegido a cero. Los datos son presentados en una pantalla líquida de cristal (LCD), en esta pantalla se muestra el aforo permitido y las personas que se encuentran dentro del local. El proceso de conteo se repite de manera cíclica sin la necesidad de la intervención humana.

Palabras Clave:

- **ARDUINO MEGA.**
- **SENSOR E18-D80NK.**
- **AFORO DE PERSONAS.**
- **CONTADOR ASCENDENTE Y DESCENDENTE.**

Abstrac

The objective of this research and practical project is to implement an automatic system to determine the current capacity at Sr. Pincho restaurant". An Arduino MEGA board with 54 digital pins that function as input and output was used, where the signal of two adjustable infrared proximity sensors E18-D80K (S1, S2) was found, which detect objects without the need for contact in a range of up to 80cm. These sensors are placed at the entrance of the restaurant to count down and up the capacity. The sensor output follows the following sequence: first instance S1 turns on, second instance S1, S2 on, third instance S1 off, S2 off counts the number of person at the input (counting up). Instead, the countdown occurs in the opposite manner as described. The information is presented on a panel made up of two 7-segment common anode displays and green indicator lights to continue and red to stop, depending on the allowed capacity, when it is exceeded, an alarm will sound that will not deactivate unless the number of people drops. , this value can be modified using a keyboard that has the option of resetting the chosen set point to zero. The data is sent on a liquid crystal display (LCD), this screen shows the allowed capacity and the people who are inside the premises. The counting process is repeated cyclically without the need for human intervention.

Key words:

- **ARDUINO MEGA**
- **SENSOR E18-D80NK**
- **CAPACITY OF PEOPLE**
- **UP AND DOWN CUNTER**

CAPÍTULO I

1. Generalidades

1.1. Introducción

El presente proyecto técnico está basado en el estudio y desarrollo de la “Implementación de un sistema automático para determinar el aforo actual en el restaurante Sr. Pincho”, para las cuales se han implementado técnicas de investigación, tales como experimentales, campo y de observación, cuyo objetivo es realizar un sistema automático para determinar el aforo actual en el restaurante Sr. Pincho.

El control de aforo en la actualidad es una medida de bioseguridad adoptadas por el gobierno en curso y el COE nacional, la automatización de este proceso se centra en sí, en controlar el ingreso y salida de personas a un área determinada, para de esta manera vigilar la cantidad de personas que ingresan, mediante una placa madre la cual es el Arduino Mega 2560.

Además, se han implementado varios sistemas para la tabulación en este sistema automático.

1.2. Antecedentes

En el año 2019 en el mes de noviembre se origina el SARSCOV-2 en el país de China, donde la contaminación se propaga, pues hay individuos que viajan a diferentes partes de mundo por negocios, paseo y otros.

Es así como el Ecuador se vio afectado y a finales del mes de marzo del 2020 con una contaminación abrupta, matando a miles de personas y afectando la economía en el país.

Debido a que la población a nivel nacional tiene la necesidad de reactivar la economía de todos los establecimientos para generar ingresos, tomando en cuenta el riesgo del virus SARSCOV-2 que puede ser mortal, han tomado la decisión de abrir los locales comerciales, siguiendo las medidas establecidas por el COE Nacional, como son un aforo establecido en cada uno de los locales en un máximo del 30%.

Es por este acontecimiento y debido a las medidas tomadas por las diferentes entidades del gobierno de turno para salvaguardar la salud de la población en general, se han presentado medidas de precaución como el uso de mascarilla, lavado continuo de las manos con agua y jabón, uso de alcohol y una distancia segura entre personas de como mínimo dos metros.

Debido a que los diferentes países como en el Ecuador, se encuentra atravesando una crisis sanitaria debido a la propagación del virus SARSCOV-2, ha originado que las autoridades de salud tomen medidas que permiten precautelar la proliferación del mencionado virus.

Entre las medidas tomadas se ha determinado que los locales comerciales en base a su tamaño restrinjan el ingreso de personas a una determinada capacidad por lo que se considera importante la implementación de un sistema automático para determinar el número de personas que ingresan y salen del restaurante Sr. Pincho, para controlar el aforo establecido debido a las medidas de bioseguridad que han sido establecidas por varias entidades del gobierno ecuatoriano.

El Sr. Pincho, es un restaurant con 3 años de trayectoria en el mercado, que se vio afectado por dicho virus, la cual debido a la pandemia ocurrida meses atrás, no contaba con clientes para sobrellevar los gastos de inversión en el local, por lo que su economía se vio afectada, y con ello se toma la solución de adquirir un sistema automático para determinar el aforo actual en el restaurante, pues al adquirir este sistema automático ya mencionado se podrá activar la economía del restaurante.

1.3. Justificación e importancia

Se ha planteado una propuesta para dar solución al problema que afecta de manera económica a varios establecimientos que en la actualidad no pueden realizar con normalidad sus actividades, debido a las restricciones planteadas por el COE nacional por la pandemia que se está enfrentando en la actualidad.

Al realizar la “IMPLEMENTACIÓN DE UN SISTEMA AUTOMÁTICO PARA DETERMINAR EL AFORO ACTUAL EN EL RESTAURANTE SR. PINCHO.”, se reactivará mediante el servicio brindado, la economía del restaurante Sr. Picho debido a un control que será realizado para el aforo de personas que ingresan al establecimiento.

1.4. Planteamiento del problema

En la actualidad el país y el mundo se encuentra atravesando una crisis sanitaria debido a la propagación del virus SARSCOV-2, lo que ha originado que las autoridades de salud tomen medidas que permiten precautelar la proliferación del mencionado virus. Entre las medidas tomadas se ha determinado que los locales comerciales en base a su tamaño restrinjan el ingreso de personas a una determinada capacidad por lo que se considera importante la implementación de un

sistema automático para determinar el número de personas que ingresan y salen del restaurante Sr. Pincho para controlar el aforo establecido debido a las medidas de bioseguridad que han sido establecidas por varias entidades del gobierno ecuatoriano.

Por dicho virus los locales comerciales no pueden realizar sus actividades con normalidad, debido a las medidas de prevención dadas por la salud que el gobierno de turno ha dispuesto para los habitantes a nivel nacional. El problema se genera por medio de dicho virus. Ya que, al no darse una solución prudente de manera inmediata, nuevamente se podría optar por un cierre temporal de los locales comerciales debido a la proliferación del virus que se encuentra latente en todas partes. Y esto lleva de la mano a que el Restaurant Sr. Pincho por consiguiente cierre y no mejore su economía y recuperación de los recursos que son invertidos para el servicio de las personas.

Además, pueden surgir nuevamente focos de contagios ya que existe cierto porcentaje de personas que no presentan síntomas al verse afectado por el virus SARSCOV-2.

1.5. Objetivos

1.5.1. Objetivo general

- Implementar un sistema automático para determinar el aforo actual en el restaurante Sr. Pincho.

1.5.2. Objetivos específicos

- Establecer información básica y detallada referente a tarjetas de control que permitan realizar un conteo de personas mediante sensores implementados.

- Analizar la afluencia de personas que se da en el restaurante Sr. Pincho para un control adecuado del ingreso y salida de personas.
- Implementar un sistema automático para determinar el aforo actual en el restaurante Sr. Pincho.

1.6. Alcance

Este proyecto tiene como finalidad implementar un sistema automático para determinar el aforo actual en el restaurante Sr. Pincho, donde mediante control de sensores y un indicador por segmentos realizado mediante led's de alto brillo se indicará el conteo del ingreso y salida de personas. Este proceso automático es realizado mediante Arduino Mega 2560, para la cual se emplearon técnicas de investigación sobre lo que refiere a librerías, y líneas de programación para declaración de variables como de control de sensores además de botones para controlar el número máximo de ingreso en el cual permiten modificar las opciones.

CAPÍTULO II

2. Marco Teórico

2.1. Arduino

Arduino es una plataforma de prototipos electrónica de código abierto (open-source) basada en hardware y software flexibles y fáciles de usar. Está pensado para artistas, diseñadores, como hobby y para cualquier interesado en crear objetos o entornos interactivos. Arduino puede sentir el entorno mediante la recepción de entradas desde una variedad de sensores y puede afectar a su alrededor mediante el control de luces, motores y otros artefactos (Arduino, 2021).

El microcontrolador de la placa se programa usando el Arduino Programming Language (basado en Wiring 1) y el Arduino Development Environment (basado en Processing 2). Los proyectos de Arduino pueden ser autónomos o se pueden comunicar con software en ejecución en un ordenador (por ejemplo, con Flash, Processing, MaxMSP, etc.). Las placas se pueden ensamblar a mano o encargarse preensambladas; el software se puede descargar gratuitamente. Los diseños de referencia del hardware (archivos CAD) están disponibles bajo licencia open-source, por lo que es libre de adaptarlas a sus necesidades. Arduino recibió una mención honorífica en la sección Digital Communities del Ars Electronica Prix en 2006 (Herrador, 2009).

Arduino es una plataforma de creación de electrónica de código abierto, la cual está basada en hardware y software libre, flexible y fácil de utilizar para los creadores y desarrolladores. Esta plataforma permite crear diferentes tipos de microordenadores de una sola placa a los que la comunidad de creadores puede darles diferentes tipos de uso (Fernández, 2020).

Para poder entender este concepto, primero se debe entender los conceptos de hardware libre y el software libre. El hardware libre son los dispositivos cuyas

especificaciones y diagramas son de acceso público, de manera que cualquiera puede replicarlos. Esto quiere decir que Arduino ofrece las bases para que cualquier otra persona o empresa pueda crear sus propias placas, pudiendo ser diferentes entre ellas, pero igualmente funcionales al partir de la misma base. (Fernández, 2020).

El software libre son los programas informáticos cuyo código es accesible por cualquiera, para que quien quiera pueda utilizarlo y modificarlo. Arduino ofrece la plataforma Arduino IDE (Entorno de Desarrollo Integrado), que es un entorno de programación con el que cualquiera puede crear aplicaciones para las placas Arduino, de manera que se les puede dar todo tipo de utilidades (Fernández, 2020).

El proyecto nació en 2003, cuando varios estudiantes del Instituto de Diseño Interactivo de Ivrea, Italia, con el fin de facilitar el acceso y uso de la electrónica y programación. Lo hicieron para que los estudiantes de electrónica tuviesen una alternativa más económica a las populares BASIC Stamp, unas placas que por aquel entonces valían más de cien dólares, y que no todos se podían permitir (Fernández, 2020).

El resultado fue Arduino, una placa con todos los elementos necesarios para conectar periféricos a las entradas y salidas de un microcontrolador, y que puede ser programada tanto en Windows como macOS y GNU/Linux. Un proyecto que promueve la filosofía 'learning by doing', que viene a querer decir que la mejor manera de aprender es cacharreando (Fernández, 2020).

El Arduino es una placa basada en un microcontrolador ATMEL. Los microcontroladores son circuitos integrados en los que se pueden grabar instrucciones, las cuales se escriben con el lenguaje de programación que puede utilizar en el entorno Arduino IDE. Estas instrucciones permiten crear programas que interactúan con los circuitos de la placa (Fernández, 2020).

El microcontrolador de Arduino posee lo que se llama una interfaz de entrada, que es una conexión en la que se conectan en la placa diferentes tipos de periféricos. La información de estos periféricos que conecte se trasladará al microcontrolador, el cual se encargará de procesar los datos que le lleguen a través de ellos (Fernández, 2020).

El tipo de periféricos que puede utilizar para enviar datos al microcontrolador depende en gran medida de qué uso le esté pensando dar.

Pueden ser cámaras para obtener imágenes, teclados para introducir datos, o diferentes tipos de sensores (Fernández, 2020).

También cuenta con una interfaz de salida, que es la que se encarga de llevar la información que se ha procesado en el Arduino a otros periféricos. Estos periféricos pueden ser pantallas o altavoces en los que se puede reproducir los datos procesados, pero también pueden ser otras placas o controladores (Fernández, 2020).

Figura 1

Ilustración placa Arduino Mega 2560 REV 3



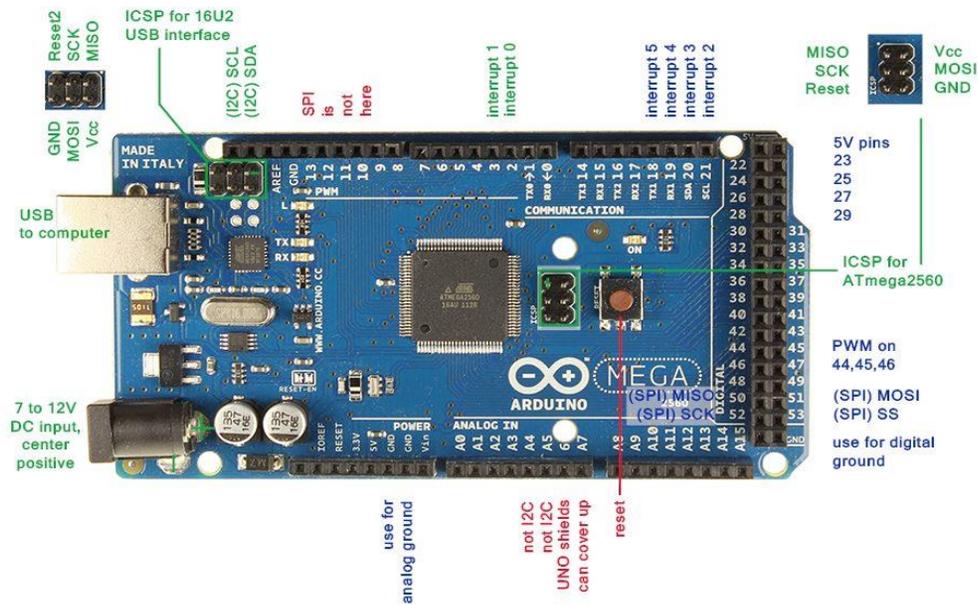
Nota. En esta imagen se muestra el diseño externo del Arduino mega 2560.

Tomado de (Arduino, 2021).

Los datos entradas y salidas se observan a continuación en la Figura 2, además se visualiza todos los elementos que interactúan, también se especifican las características más importantes del Arduino en la Tabla 1.

Figura 2

Descripción de entradas y salidas de Arduino Mega



Nota. En esta imagen se muestra la localización de pines de conexión externas.

Tomado de (Arduino, 2021).

Tabla 1*Especificaciones técnicas del Arduino Mega*

Característica	Descripción
Microcontrolador	Atmega2560
Tensión de trabajo	5V
Tensión de entrada (recomendada)	7-12V
Tensión de entrada (límite)	6-20V
Pines Digitales I/O	54 (de los cuales 15 proporcionan salida PWM)
Pines de entradas Analógicas	16
DC Corriente por Pin I/O	20 mA
DC Corriente por Pin 3.3V	50mA
Memoria Flash	25 KB de los cuales 8 KB se usan por el bootloader
SRAM	8 KB
EEPROM	4 KM
Velocidad del reloj	16 MHz
Largo	101.52 mm
Ancho	53.3 mm
Peso	37 g

Nota. Esta tabla muestra las características del Arduino Mega. Tomado de (Arduino, 2021).

2.2. Sensor E18-D80NK

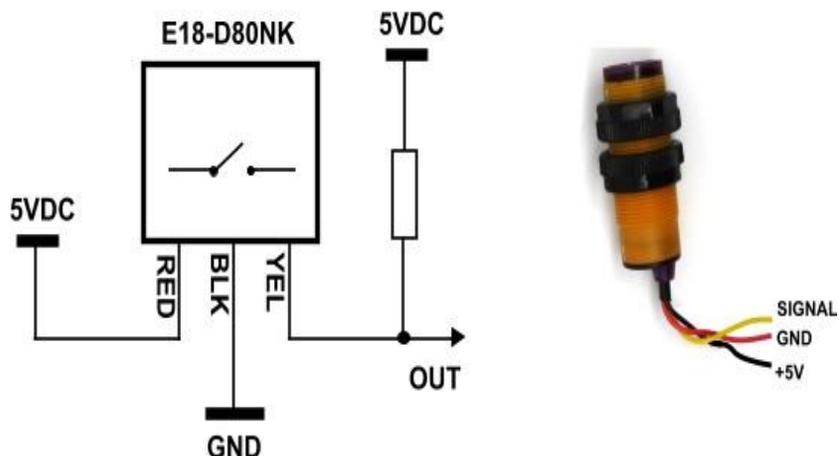
El E18-D80NK es un sensor óptico reflexivo diseñado para la detección de presencia de objetos, su distancia de detección es ajustable, lo que permite utilizarlo en una amplia gama de aplicaciones. Puede usarse en detección y conteo de personas, robots de sumo, detección de objetos, aplicaciones de automatización (Martínez Hidalgo, 2020).

El sensor E18-D80NK es ideal para utilizarse en aplicaciones de proyectos con Arduino y microcontroladores PIC, que funcionen con 5 voltios de corriente continua directa (Martínez Hidalgo, 2020).

Se debe tener en cuenta que este sensor no dará como medición de distancia un nivel de tensión analógico, sino que entregará un nivel lógico alto o bajo, dependiendo de la sensibilidad ajustada en el potenciómetro de la parte trasera del mismo sensor (Martínez Hidalgo, 2020).

Figura 3

Representación física sensor E18-D80NK y disposición de pines.



Nota. Esta imagen muestra los pines de salida y entrada del sensor E18-D80NK.

Tomado de (Martínez Hidalgo, 2020).

Tabla 2*Especificaciones técnicas sensor E18-D80NK*

Característica	Descripción
Tensión de operación	5v DC
Rango de detección	3 a 80 cm (ajustable)
Corriente de salida máxima	100mA
Salida	Tipo NPN normalmente abierto
Dimensiones	18mm D * 68mm L
Material	Polímero
Temperatura de trabajo	-25 a 70 °C

Nota. Esta tabla tiene información técnica del sensor de proximidad infrarrojo E18-D80NK. Tomado de (Martínez Hidalgo, 2020).

2.3. Circuito integrado ULN 2803 A

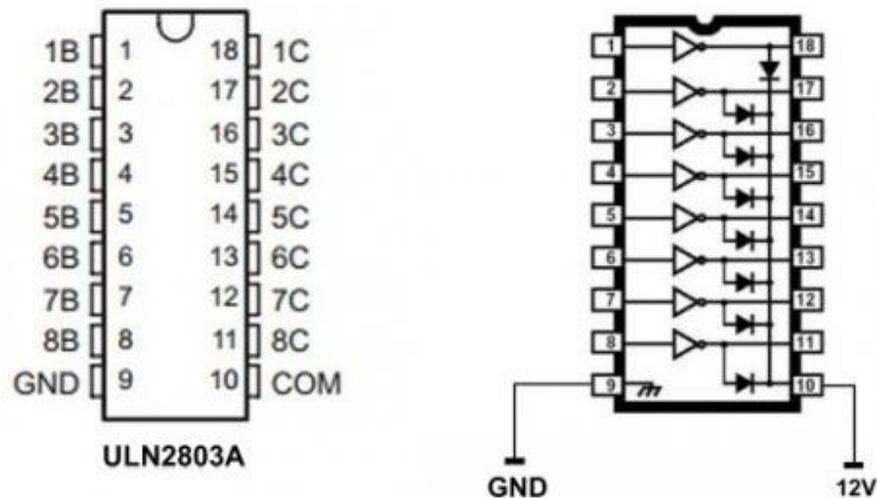
El circuito integrado ULN2803 es un arreglo de transistores Darlington que ofrecen 8 salidas de alta corriente, para impulsar dispositivos como: relay, motores, solenoides, bombillas de filamento, series de leds, display, etc. (Saldarriaga Cardona, 2012).

Otras de sus características principales son:

- Voltaje de alimentación: Típicamente el ULN2803 se puede utilizar con voltajes de 5, 9 y 12 voltios, pero también puede soportar voltajes hasta de 50v (Saldarriaga Cardona, 2012).
- El “voltaje de alimentación” se hace por el pin 10 (COM), que más bien se trata de un punto común para todos los cátodos de los diodos clamp. La conexión a tierra (GND) se encuentra en el pin 9 (Electrónica, 2016).
- Corriente: La corriente máxima que puede obtener por salida del ULN2803 es de 500mA, pero la misma se puede incrementar si se utilizan las salidas y entradas en paralelo (Saldarriaga Cardona, 2012).

Figura 4

Disposición de pines ULN 2803



Nota. En la figura se indica la distribución de pines del circuito integrado ULN2803A. Tomado de (Saldarriaga Cardona, 2012)

Tabla 3*Especificaciones técnicas ULN 2803*

Característica	Descripción
Transistores NPN Darlington en emisor común	8
Diodos de supresión de voltajes inversos	8
Salida	0.5A (500mA) Max.
Voltaje de sostenimiento a la salida	-0.5V a 50V
Potencia máxima	1.47 W
Resistencia de entrada a la base	2.7 k Ω
Voltaje de entrada	--0.5V a 30V
Entradas compatibles	TTL t CMOS de 5V
Encapsulados	DIP 18 pines

Nota. Esta tabla describe las características del circuito integrado ULN 2803.

Tomado de (Saldarriaga Cardona, 2012).

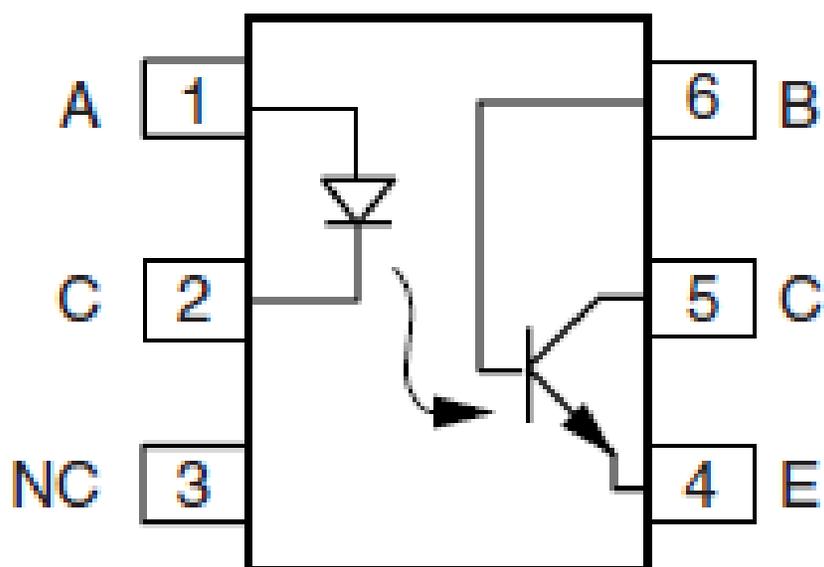
2.4. Optoacoplador 4N25

Un optoacoplador, también llamado optoaislador o aislador acoplado ópticamente, es un dispositivo de emisión y recepción que funciona como un interruptor activado mediante la luz emitida por un diodo led que satura un componente optoelectrónico, normalmente en forma de fototransistor o fototriac.

De este modo se combinan en un solo dispositivo semiconductor, un foto-emisor y un fotorreceptor cuya conexión entre ambos es óptica. Estos elementos se encuentran dentro de un encapsulado que por lo general es del tipo DIP. Se suelen utilizar para aislar eléctricamente a dispositivos muy sensibles.

Figura 5

Disposición de pines 4N25



Nota. En la imagen se observa la disposición de pines del circuito integrado 4N25.

Tomada de (Malvino, 2000).

Tabla 4*Especificaciones técnicas 4N25*

Característica	Descripción
Salida	Fototransistor
Voltaje de aislamiento	2500 VRMS
Corriente del LED en directo máx.	80mA
Voltaje inverso del LED máx.	6V
Voltaje colector-emisor máx.	30V
Corriente de colector máx.	100mA
Tiempo de respuesta	3us típico
CTR min	20%
Encapsulado	DIP de 6 pines

Nota. Esta tabla describe las características del circuito integrado 4N25. Tomado de (Malvino, 2000).

2.5. Regulador de voltaje LM7805

Un regulador de voltaje o tensión como el LM7805 es un dispositivo que es capaz de modificar una señal de tensión que obtiene a su entrada y ofrece una señal diferente de voltaje a su salida (Malvino, 2000).

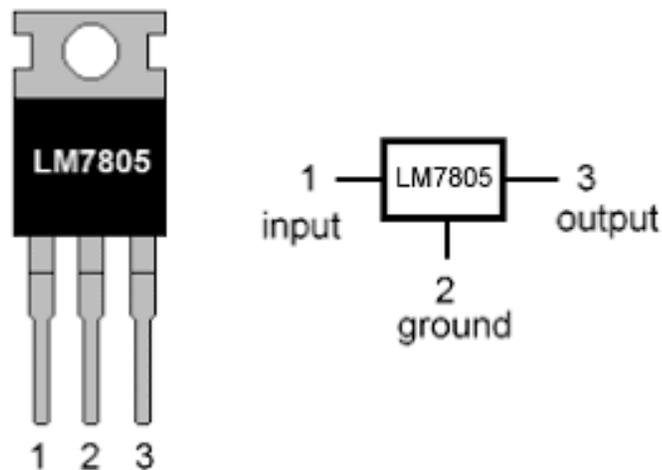
En esa salida, el voltaje suele ser inferior y con unas características determinadas que se requieren para evitar riesgos o para el que el circuito que alimenta funcione de forma adecuada, si es sensible a variaciones de voltaje (Malvino, 2000).

El regulador de tensión cuenta con un circuito interno con una serie de resistencias y transistores bipolares, conectados de tal forma que permiten afinar la señal del voltaje de una forma adecuada (Malvino, 2000).

Figura 6

Disposición de pines LM7805

LM7805 PINOUT DIAGRAM



Nota. En esta imagen se muestra la distribución de pines del circuito integrado LM7805. Tomado de (Malvino, 2000).

Tabla 5*Especificaciones técnicas LM7805*

CARACTERÍSTICA	DESCRIPCIÓN
Vout	5V
Vin-Vout	202V – 30V
Temperatura de operación	0 – 125 °C
I máx. de salida	1A

Nota. Esta tabla contiene las características técnicas de circuito integrado LM7805 tomado de (Malvino, 2000).

2.6. Transistor NPN 2n3904

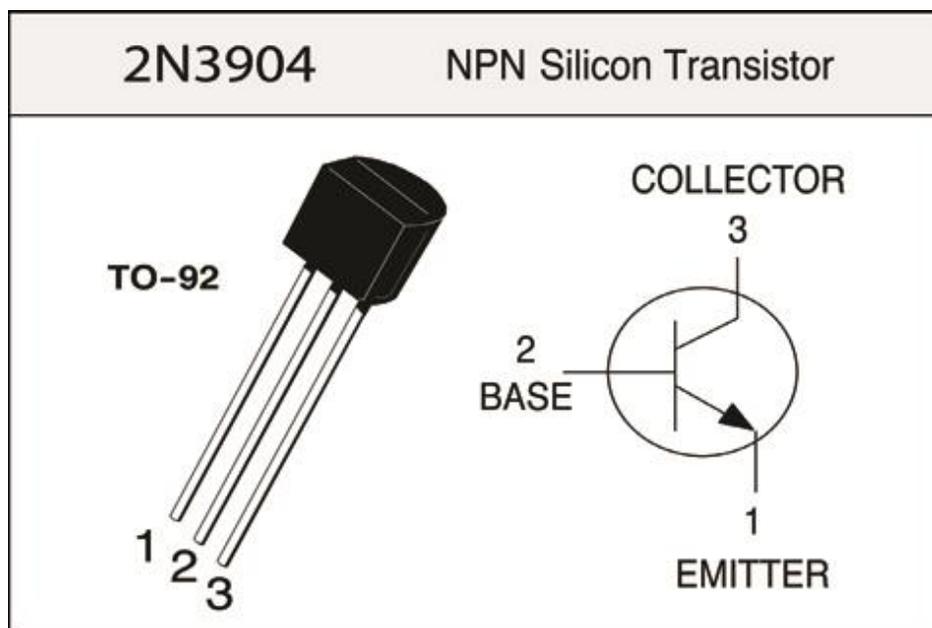
El transistor 2N3904 es de pequeña señal. El 2N3904 es un transistor tipo NPN de conmutación rápida, corto apague y baja tensión de saturación, adecuado para la conmutación y amplificación.

El transistor es un dispositivo electrónico semiconductor utilizado para entregar una señal de salida en respuesta a una señal de entrada (Raymundo Barrales, 2019).

Actualmente se encuentra prácticamente en todos los aparatos electrónicos de uso diario tales como radios, televisores, reproductores de audio y video, relojes de cuarzo, computadoras, lámparas, tomógrafos, teléfonos celulares, aunque casi siempre dentro de los llamados circuitos integrados (Raymundo Barrales, 2019).

Figura 7

Disposición de pines 2N3904



Nota. Esta imagen muestra el tipo de transistor y distribución de pines. Tomada de (Raymundo Barrales, 2019).

Tabla 6

Especificaciones técnicas del transistor 2N3904

CARACTERÍSTICA	DETALLE
Transistor bipolar	NPN, 40V, TO-92
Polaridad del transistor	NPN
Voltaje V (br)	40 V
Transición de frecuencia ft	300 MHz
Disipación de potencia Pd	625 mW

CARACTERÍSTICA	DETALLE
DC Corriente del colector	200 mA
Ganancia de corriente continua hFE	100
Rango temperatura	-55 °C a 150 ° C
Colector Emisor Tensión Sat	300mV en $I_c = 10 \text{ mA}$
DC ganancia corriente	30 en $I_c = 100 \text{ mA}$
Encapsulado	TO-92

Nota. Esta tabla contiene las características técnicas del transistor 2N3904.

Tomado de (Raymundo Barrales, 2019).

2.7. Diodos LED

Los LEDs o diodos de emisión de luz siempre se han utilizado en el mundo de la tecnología como indicadores lumínicos, ya que son dispositivos semiconductores que emiten luz cuando por el circula una corriente eléctrica (Déleg, 2015).

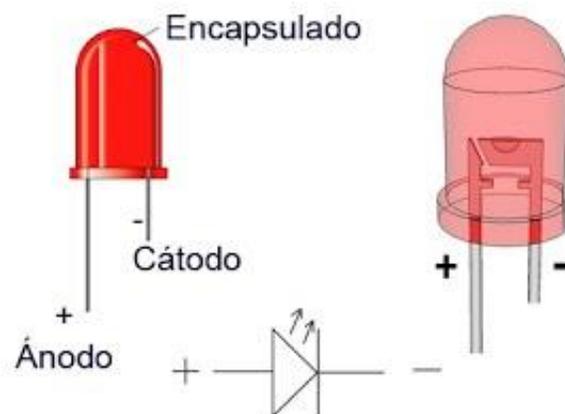
La tecnología LED ha dado su paso más reciente con los LEDs de luz blanca que son los que pueden sustituir a las bombillas de luz, principalmente porque consumen menos energía eléctrica la cual se enfoca al cuidado del medio ambiente. En cuanto a las pantallas LED, hay varios modelos de LEDs distintos como los OLED (leds orgánicos) que están fabricados con materiales polímeros orgánicos semiconductores, que, si bien no tienen la misma eficiencia, son aún

mucho más baratos de fabricar y de una desintegración mejor para el medio ambiente. Este ensayo da una reseña histórica sobre la tecnología led. También se exponen aquí su principio de funcionamiento y cuáles son sus ventajas al ser aplicado en cualquier necesidad humana y sus desventajas que en comparación con las ventajas son mínimas (Déleg, 2015).

LED se define por sus siglas como diodo emisor de luz, no es más que un pequeño chip de material semiconductor, que cuando es atravesado por una corriente eléctrica, en sentido apropiado, emite luz monocromática sin producir calor, es decir un componente electrónico semiconductor, con polaridad por lo que se usará en funciones de señalización, estética y, actualmente iluminación. Su estructura consta de un hilo muy fino, entre el cátodo y el ánodo, que podría dar apariencia de fragilidad, pero no es así; porque no tiene que ponerse incandescente (de hecho, apenas se calienta) y no está al aire, sino incrustado dentro del epoxi (Déleg, 2015).

Figura 8

Simbología y polaridad diodo led



Nota. Esta imagen muestra la polaridad, diseño físico y simbólico de los diodos LED. Tomada de (Déleg, 2015).

Tabla 7*Especificaciones de voltaje de LEDS*

COLOR	LED
Rojo	1.8 a 2.2 V
Anaranjado	2.0 a 2.2 V
Amarillo	2.0 a 2.4 V
Verde	2.0 a 3.5 V
Azul	3.5 a 3.8 V
Blanco	3.6 a 4.0 V

Nota. Esta tabla contiene la tensión que depende de cada color de led. Tomada de (Déleg, 2015)

2.8. Condensadores

Un condensador eléctrico también conocido como capacitor, proveniente del nombre equivalente en inglés, es un dispositivo pasivo, utilizado en electricidad y electrónica, capaz de almacenar energía sustentando un campo eléctrico (Routledge, 1997).

Está formado por un par de superficies conductoras, generalmente en forma de láminas o placas, en situación de influencia total, esto es, que todas las líneas de campo eléctrico que parten de una van a parar a la otra, separadas por un material dieléctrico o por vacío (Atienza, 1997).

Las placas, sometidas a una diferencia de potencial, adquieren una determinada carga eléctrica, positiva en una de ellas y negativa en la otra, siendo nula la variación de carga total (Routledge, 1997).

Aunque desde el punto de vista físico un condensador no almacena carga ni corriente eléctrica, más bien simplemente energía mecánica latente, al ser introducido en un circuito, se comporta en la práctica como un elemento capaz de almacenar la energía eléctrica (Atienza, 1997).

Figura 9

Representación física de condensadores.



Nota. En esta imagen se muestra el diseño físico de los capacitores. Tomada de (Atienza, 1997).

A continuación, se muestra la clasificación de condensadores

1. Condensadores fijos

- Condensador de cerámica
- Condensador de lámina de plástico
- Condensador de mica

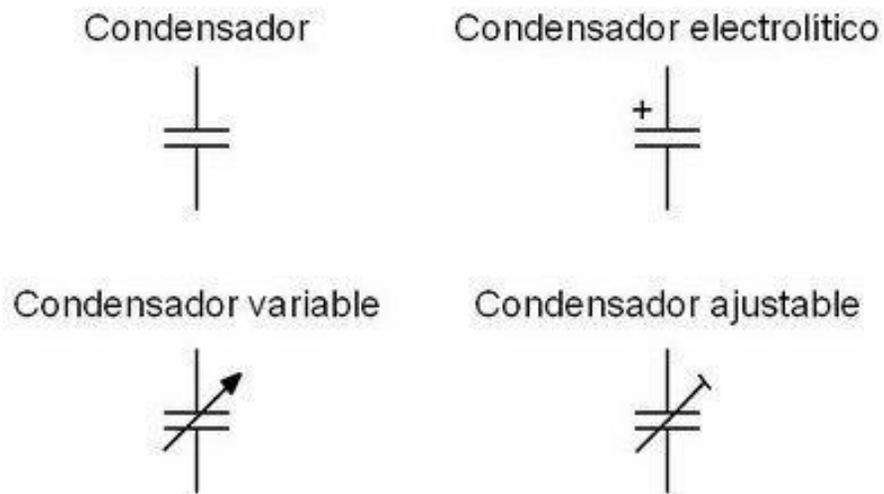
- Capacitores de poliéster
- Condensadores electrolíticos
- Condensadores de tantalio

2. Condensadores variables

- Capacitores variables giratorios
- Capacitores ajustables Trimmer

Figura 10

Representación simbólica



Nota. Esta imagen muestra la representación de cada tipo de condensadores.

Tomada de (Routledge, 1997)

2.9. Resistencia eléctrica

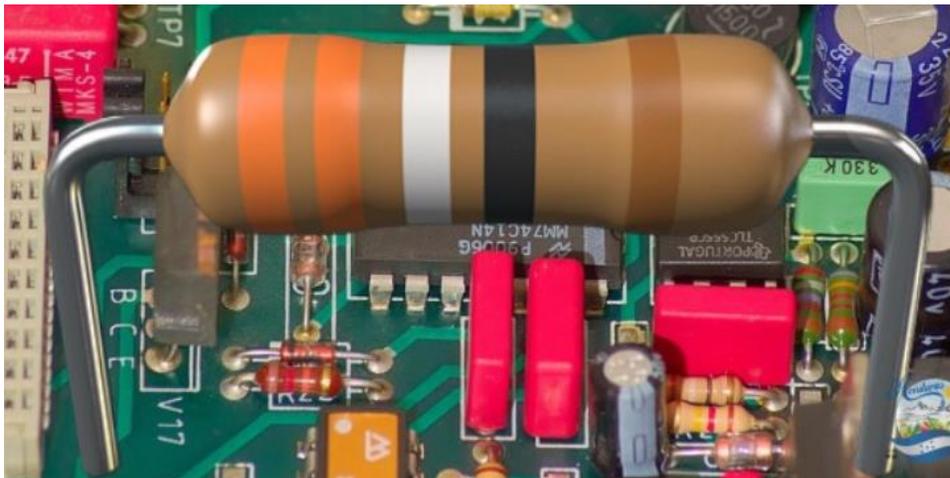
Se le denomina resistencia eléctrica a la oposición al flujo de corriente a través de un conductor (Young, 2020).

La unidad de resistencia en el Sistema Internacional es el ohmio, que se representa con la letra griega omega (Ω), en honor al físico alemán Georg Simon Ohm, quien descubrió el principio que ahora lleva su nombre (Young, 2020).

La resistencia tiene diferentes valores como: ohmio, kilo ohmio, giga ohmio, entre muchos valores más. Para la verificación de las mismas se utiliza un código de colores con sus respectivas valencias (Young, 2020).

Figura 11

Representación física de una resistencia.



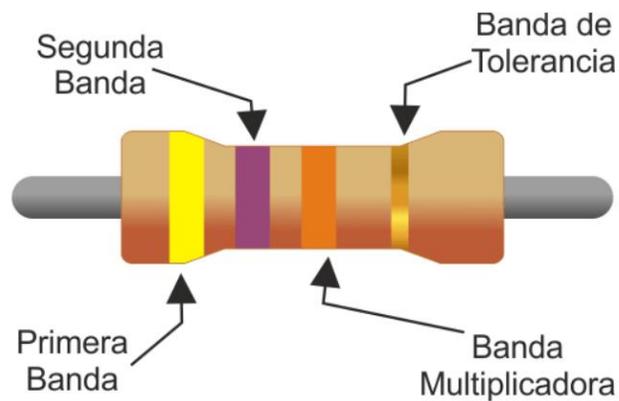
Nota. En esta imagen se muestra la representación física de una resistencia.

Tomada de (M.I. Juan Carlos Cedeño Vázquez, 2016).

A continuación, se muestra como identificar las bandas de una resistencia para el reconocimiento e identificación de las diferentes resistencias según el color de cada una de ellas.

Figura 12

Especificación de las bandas de la resistencia



Nota. En esta imagen se muestra las bandas de una resistencia. Tomada de (M.I. Juan Carlos Cedeño Vázquez, 2016).

Como siguiente paso se muestra el código de colores para la identificación de las resistencias.

Tabla 8

Código de colores para resistencias.

Color	Banda 1	Banda 2	Banda 3	Tolerancia
NEGRO	0	0	X 1 Ω	
MARRÓN	1	1	X 10 Ω	+/- 1%
ROJO	2	2	X 100 Ω	+/- 2%
NARANJA	3	3	X 1000 Ω	
AMARILLO	4	4	X 10,000 Ω	
VERDE	5	5	X 100,000 Ω	
AZUL	6	6	X 1,000,000 Ω	
VIOLETA	7	7	X 10,000,000 Ω	
GRIS	8	8	X 100,000,000 Ω	

Color	Banda 1	Banda 2	Banda 3	Tolerancia
BLANCO	9	9	X 1,000,000,000 Ω	
DORADO	-----	-----	X 0,1 Ω	+/- 5%
PLATEADO	-----	-----	X 0,01 Ω	+/-10%
	SIN BANDA			+/- 20%

Nota. Los datos para realizar esta tabla fueron tomados de (M.I. Juan Carlos

Cedeño Vázquez, 2016)

CAPÍTULO III

3. Desarrollo Del Proyecto

El propósito del desarrollo del proyecto consiste en definir los detalles para la “Implementación del sistema automático para determinar el aforo actual en el restaurante Sr. Pincho”.

El aplicativo en si consiste principalmente en cumplir un sistema automático de tabulación, es por ello, que el sistema implementado permitirá realizar un control de las personas que ingresan y salen del lugar.

3.1. Desarrollo

Para la implementación del presente proyecto se consideró la utilización de la placa Arduino, que fue programada en el lenguaje de programación C++ de la placa.

Se realizó el código de programación en el software Arduino para poder ejecutar el proceso automático de la tabulación, en la cual mediante el sensor registra el ingreso o salida de personas, el indicador muestra la cantidad de personas que se encuentran en el establecimiento. A continuación, se detallan las líneas de programación.

3.2. Programación

```
// Declaración de LCD
#include <LiquidCrystal.h>           // Librería LCD
// Declaración de variables
// Variables de entrada
int leds1 = 36;                      // Retorno de pulsos S1
int leds2 = 37;                      // Retorno de pulsos S2
```

```

int contadorprincipal =0;           // Variable de conteo
String pulsadores;                 // Cadena de caracteres sensores
const int sensor_asc = 2;          // Entrada sensor S1
const int sensor_des = 3;          // Entrada sensor S2
int estado_sensor_asc ;            // Estado del sensor S1 (1-0)
int estado_sensor_des;             // Estado del sensor S2 (1-0)

// Pines LCD
const int rs = A5, en = A4, d4 = A3, d5 = A2, d6 = A1, d7 = A0;
LiquidCrystal lcd (rs, en, d4, d5, d6, d7); // Configuración pines `LCD
// SET POINT (AFORO)
const int pulsante_asc = 22;       // Entrada para el contador de pulsos ascendentes
const int pulsante_des = 24;       // Entrada para el contador de pulsos descendentes
const int ledinsensor1 = 50;        //Indicador de retorno sensor S1
const int ledinsensor2 = 52;        //Indicador de retorno sensor S2
//VARIABLES CONTEO ASCENDENTE SET POINT
int contadorseleccion = 0;         // Variable conteo Set Point(# de veces presionado)
int estado_pulsante_asc = 0;        // Estado actual del pulsador Aumentar (1-0)
int estadoanteriorpulsante_asc = 1; // Estado anterior del pulsador (1-0)
// VARIABLES PARA EL CONTADOR DESCENDENTE SET POINT
int estado_pulsante_des = 0;        // Estado actual del pulsador Decrementar (1-0)
//Estado anterior del pulsador Decrementar (1-0)
int estadoanteriorpulsante_des = 1;
//VARIABLES DE RETORNO SET POINT
const int pulsador_asc = 38;        // Entrada de pulsos set point Aumentar
const int pulsador_des = 39;        // Entrada de pulsos set point Decrementar
//VARIABLES PARA EL CONTADOR ASCENDENTE S1
int contador = 0;                   // Variable conteo Sensores(# de veces presionado)
int estado_pulsador_asc = 0;        // Estado actual S1(1-0)
int lastButtonState_asc = 1;        // Estado anterior de S1(1-0)

```

```

//VARIABLES PARA EL CONTADOR DESCENDENTE S2
int estado_pulsador_des = 0;           // Estado actual del S2(1-0)
int lastButtonState_des = 1;         // Estado anterior del S2(1-0)

//VARIABLES PARA INDICADOR DE AFORO
const int sir = 44;                   // Pin Sirena
const int ledaforomax = 46;          // Indicador de aforo máximo
const int ledaforoafirmativo = 42;   // Indicador de aforo continúe

//VARIABLES PINES PARA DISPLAY 1` (UNIDADES)
int a = 20;
int b = 19;
int c = 18;
int d = 17;
int e = 16;
int f = 15;
int g = 14;

//VARIABLES PINES PARA DISPLAY 2` (DECENAS)
int A = 7;
int B = 8;
int C = 9;
int D = 10;
int E = 11;
int F = 12;
int G = 13;

//ARRAY DISPLAY 7 SEGMENTOS
int uno [7] = {a, b, c, d, e, f, g};   // UNIDADES
int dos [7] = {A, B, C, D, E, F, G};   // DECENAS

//CONFIGURACION DE ARRAY DISPLAY 7 SEGMENTOS
//Ánodo común
//Valor LOW (0) encender segmento HIGH (1) apagar:

```

```

// Inversor ULN2803
int unidad0 [7] = {1, 1, 1, 1, 1, 1, 0}; //= #0
int unidad1 [7] = {0, 1, 1, 0, 0, 0, 0}; //= #1
int unidad2 [7] = {1, 1, 0, 1, 1, 0, 1}; //= #2
int unidad3 [7] = {1, 1, 1, 1, 0, 0, 1}; //= #3
int unidad4 [7] = {0, 1, 1, 0, 0, 1, 1}; //= #4
int unidad5 [7] = {1, 0, 1, 1, 0, 1, 1}; //= #5
int unidad6 [7] = {1, 0, 1, 1, 1, 1, 1}; //= #6
int unidad7 [7] = {1, 1, 1, 0, 0, 0, 0}; //= #7
int unidad8 [7] = {1, 1, 1, 1, 1, 1, 1}; //= #8
int unidad9 [7] = {1, 1, 1, 1, 0, 1, 1}; //= #9

//DECENAS
int decena0 [7] = {1, 1, 1, 1, 1, 1, 0}; //= #0
int decena1 [7] = {0, 1, 1, 0, 0, 0, 0}; //= #1
int decena2 [7] = {1, 1, 0, 1, 1, 0, 1}; //= #2
int decena3 [7] = {1, 1, 1, 1, 0, 0, 1}; //= #3
int decena4 [7] = {0, 1, 1, 0, 0, 1, 1}; //= #4
int decena5 [7] = {1, 0, 1, 1, 0, 1, 1}; //= #5
int decena6 [7] = {1, 0, 1, 1, 1, 1, 1}; //= #6
int decena7 [7] = {1, 1, 1, 0, 0, 0, 0}; //= #7
int decena8 [7] = {1, 1, 1, 1, 1, 1, 1}; //= #8
int decena9 [7] = {1, 1, 1, 1, 0, 1, 1}; //= #9

int counter = 0; // Contador interno global (0-99)
int contadorD = 0; // Contador decenas (0-9)

void setup () {
  lcd. begin(16, 2); // Inicio LCD
}

//DECLARACION DE ENTRADAS
//Entradas con Resistencia de pull_up interno
pinMode (sensor_asc, INPUT_PULLUP); // Declaración como Entrada S1.

```

```

pinMode (sensor_des, INPUT_PULLUP);           // Declaración como Entrada S2.
// Declaración como Entrada Pulsador Aumentar
pinMode (pulsante_asc, INPUT_PULLUP);
// Declaración como Entrada Pulsador Decrementar
pinMode (pulsante_des, INPUT_PULLUP);
// Declaración como Entrada pulsos set point Aumentar
pinMode (pulsador_asc, INPUT_PULLUP);
// Declaración como Entrada pulsos set point Decrementar
pinMode (pulsador_des, INPUT_PULLUP);
// DECLARACION DE SALIDAS
pinMode(leds1,OUTPUT);           // Declaración como Salida indicador S1.
pinMode (leds2 ,OUTPUT);        // Declaración como Salida indicador S2.
pinMode(ledinsensor1,OUTPUT);   // Declaración como Salida continúe flecha.
pinMode (ledinsensor2 ,OUTPUT); // Declaración como Salida stop x.
pinMode (sir, OUTPUT);          // Declaración como Salida sirena
for (byte i = 0; i < 7; i++) {  // iterar con un bucle for dentro del array
pinMode (uno[i], OUTPUT);       // Declaración como Salida display Unidades
pinMode (dos[i], OUTPUT);      // Declaración como Salida sirena display decenas
}
}
void loop() {
//CONTADOR SENSORES
//LECTURA SENSOR ASCENDENTE
If (digitalRead(sensor_asc)==HIGH){ // Lectura sensor 1 S1
digitalWrite(ledinsensor1,HIGH);    // Enciende indicador de retorno del s1
}
Else
{
digitalWrite (ledinsensor1, LOW);    //Apaga indicador de retorno del S1
}
}

```

```

}

//LECTURA SENSOR DESCENDENTE

if (digitalRead(sensor_des)==HIGH){    // Lectura sensor S2

digitalWrite(ledinsensor2,HIGH);        // Enciende indicador de retorno del S2

}

else{

digitalWrite(ledinsensor2,LOW);         // Apaga indicador de retorno S2

}

estado_sensor_asc = digitalRead(sensor_asc);

//Lee valor del pin2 y asigna al estado del sensor asc (0 o 1)

estado_sensor_des = digitalRead(sensor_des);

//Lee valor del pin3 y asigna al estado del sensor des (0 o 1)

//CONDICIONES DE SENSORES SALIDA (CONTADOR DESCENDENTE)

//PRIMER INSTANCIA

//CASO 1: S1 Encendido, S2 Apagado (Estado persona en la puerta saliendo)

if (estado_sensor_asc == HIGH && estado_sensor_des== LOW && pulsadores ==

"")

{

pulsadores="saliendo";

}

//SEGUNDA INSTANCIA

//CASO 2: S1 Encendido, S2 Encendido (Estado persona en la puerta saliendo

mitad)

if (estado_sensor_asc == HIGH && estado_sensor_des == HIGH && pulsadores ==

"saliendo")

{

pulsadores="saliendo_mitad";

}

//TERCER INSTANCIA

//CASO 3: S1 apagado, S2 Encendido (Estado persona en la puerta casi_salio )

```

```

if (estado_sensor_asc == LOW && estado_sensor_des== HIGH && pulsadores ==
"saliendo_mitad")
{
pulsadores="casi_salio";
}

if (estado_sensor_asc== LOW && estado_sensor_des== LOW && pulsadores ==
"casi_salio")
{
pulsadores="";
contadorprincipal= contadorprincipal-1;      // Decrementa una persona
digitalWrite(leds2,HIGH);                    // Enciende Retorno de pulsos S2
analogWrite(sir,220);                        // Enciende sirena
delay(500);                                  // Espera medio segundo
}
else
{
digitalWrite(leds2,LOW);                     // Apagar Retorno de pulsos S2
analogWrite(sir,0);                          // Apagar sirena
}

//CONDICIONES DE SENSORES ENTRADA (CONTADOR ASCENDENTE)
//PRIMER INSTANCIA
//CASO 1: S1 Apagado, S2 Encendido (Estado persona en la puerta entrando)
if (estado_sensor_asc == LOW && estado_sensor_des== HIGH && pulsadores ==
"")
{
pulsadores="entrando";
}

//SEGUNDA INSTANCIA
//CASO 2: S1 Encendido, S2 Encendido (Estado persona en la puerta
entrando_mitad)

```

```

if (estado_sensor_asc == HIGH && estado_sensor_des == HIGH && pulsadores ==
"entrando")
{
pulsadores="entrando_mitad";
}

//TERCER INSTANCIA

//CASO 3: S1 Encendido, S2 apagado (Estado persona en la puerta casi_entro)
if (estado_sensor_asc == HIGH && estado_sensor_des== LOW && pulsadores ==
"entrando_mitad")
{
pulsadores="casi_entro";
}

if (estado_sensor_asc== LOW && estado_sensor_des == LOW && pulsadores ==
"casi_entro")
{
pulsadores="";
contadorprincipal= contadorprincipal+1; // Incrementa una persona
digitalWrite (leds1, HIGH); // Enciende Retorno de pulsos S1
analogWrite(sir,255); // Enciende sirena
delay(500); // Espera medio segundo
}
else
{
digitalWrite(leds1,LOW); // Apaga Retorno de pulsos S1
analogWrite(sir,0); // Apagar sirena
}

// CONTADOR AFORO SET POINT

// Lee valor del pin22 y asigna al estado del Pulsador aumentar (0 o 1)
estado_pulsante_asc = digitalRead(pulsante_asc);
estado_pulsante_des = digitalRead(pulsante_des);

```

```

// Lee valor del pin24 y asigna al estado del Pulsador disminuir (0 o 1)
// Comparar el estado del botón a su estado anterior
if (estado_pulsante_asc != estadoanteriorpulsante_asc)
{
// Si el estado fue cambiado, incremente el conteo
if (estado_pulsante_asc == HIGH)
{
// si el estado fue cambiado, incremente el conteo
// si el estado actual es alto, entonces que pase de off a on:
if (contadorseleccion < 99)
{
contadorseleccion++;
Serial.println("+");
Serial.print("numero de veces presionado");
Serial.println(contadorseleccion);
delay(100);
}
}
}

estadoanteriorpulsante_asc = estado_pulsante_asc;
// comparar el estado del botón a su estado anterior
if (estado_pulsante_des != estadoanteriorpulsante_des ) {
// si el estado fue cambiado, decrementa el conteo
// si el estado actual es alto, entonces que pase de off a on:
if (estado_pulsante_des == HIGH)
{
if (contadorseleccion > 0)
{
contadorseleccion--;

```

```

Serial.println("-");
Serial.print("numero de veces presionado");
Serial.println(contadorseleccion);
delay(100);
}
}
}

//Guarda último estado actual como el último estado para próximo bucle
estadoanteriorpulsante_des = estado_pulsante_des;
// CONTADOR DISPLAY
// Lee valor del pin 38 y asigna al estado del Pulsador aumentar (0 o 1)
estado_pulsador_asc = digitalRead(pulsador_asc);
// Lee valor del pin39 y asigna al estado del Pulsador decrementar (0 o 1)
estado_pulsador_des = digitalRead(pulsador_des);
// Comparar el estado del botón a su estado anterior
if (estado_pulsador_asc != lastButtonState_asc)
{
// Si el estado fue cambiado, incremente el conteo
if (estado_pulsador_asc == HIGH)
{
// Si el estado actual es alto, entonces que pase de off a on:
if (counter < contadorseleccion)
{
contador++;
counter++;
Serial.print("CONTEO= ");
Serial.println(contador);
Serial.print("CONTEO_sis= ");
Serial.println(counter);
}
}
}
}

```

```

delay(100);

if (counter == 10 || counter == 20 || counter == 30 || counter == 40 || counter == 50 ||
counter == 60 || counter == 70 || counter == 80 || counter == 90)

{
//Cuando sea una decena incrementa 1
contadorD++;
delay (100);
}
}
}
}

// comparar el estado del botón a su estado anterior
lastButtonState_asc = estado_pulsador_asc;
// si el estado fue cambiado, decrementa el conteo
if (estado_pulsador_des != lastButtonState_des )
{
// si el estado actual es alto, entonces que pase de off a on:
if (estado_pulsador_des == HIGH)
{
//No permite el conteo negativo en LCD
if (counter > 0 )
{
contador--;
counter--;
Serial.print("CONTEO= ");
Serial.println(contador);
Serial.print("CONTEO_sis= ");
Serial.println(counter);
delay(100);
}
}
}
}
}

```

```

if (counter == 9 || counter == 19 || counter == 29 || counter == 39 || counter == 49 ||
counter == 59 || counter == 69 || counter == 79 || counter == 89) {

contadorD--;          //Cuando sea una decena decrementa

delay(100);

}

//Resta 1 del display de decenas y lo vuelve nueve

if (contador == -1)

{

//El contador de unidades empieza en 9

contador = 9;

}

}

}

}

//AFORO MAXIMO/MINIMO

//Cuando el contador interno sea igual al set point seleccionado

if (counter == contadorseleccion)

{

//Diferente de cero

if (counter != 0)

{

//encienda un tono de detenerse

analogWrite(sir,255);

//enciende indicador de stop x

digitalWrite(ledaforomax,HIGH);

}

}

else

{

```

```

//Apaga el tono de stop
analogWrite(sir,0);

//Apaga el indicador de stop x
digitalWrite(ledaforomax,LOW);
}

// Enciende indicador de siga flecha
if (counter == 0){//Contador igual a 0
digitalWrite (ledaforoafirmativo, HIGH);
}

if (contadorseleccion > counter)
{
//si el set point mayor contador entonces
digitalWrite(ledaforoafirmativo,HIGH);
}

else
{
//Apaga el indicador de siga flecha
digitalWrite(ledaforoafirmativo,LOW);
}

//guarda el último estado actual como el ultimo estado para el próximo bucle
lastButtonState_des = estado_pulsador_des;

if (contador == 10)
{
contador = 0;
}

//CASOS PARA ENCENDER DISPLAY UNIDADES
switch (contador)
{
// ENCIENDE #0

```

```
case 0:
for (byte i = 0; i < 7; i++) {
digitalWrite (uno[i], unidad0[i]);
}
break;
// ENCIENDE #1
case 1:
for (byte i = 0; i < 7; i++) {
digitalWrite (uno[i], unidad1[i]);
}
break;
// ENCIENDE #2
case 2:
for (byte i = 0; i < 7; i++) {
digitalWrite (uno[i], unidad2[i]);
}
break;
// ENCIENDE #3
case 3:
for (byte i = 0; i < 7; i++) {
digitalWrite (uno[i], unidad3[i]);
}
break;
// ENCIENDE #4
case 4:
for (byte i = 0; i < 7; i++) {
digitalWrite (uno[i], unidad4[i]);
}
break;
```

```
// ENCIENDE #5
case 5:
for (byte i = 0; i < 7; i++) {
digitalWrite (uno[i], unidad5[i]);
}
break;
// ENCIENDE #6
case 6:
for (byte i = 0; i < 7; i++) {
digitalWrite (uno[i], unidad6[i]);
}
break;
// ENCIENDE #7
case 7:
for (byte i = 0; i < 7; i++) {
digitalWrite (uno[i], unidad7[i]);
}
break;
// ENCIENDE #8
case 8:
for (byte i = 0; i < 7; i++) {
digitalWrite (uno[i], unidad8[i]);
}
break;
// ENCIENDE #9
case 9:
for (byte i = 0; i < 7; i++) {
digitalWrite (uno[i], unidad9[i]);
}
```

```
break;
}
// CASOS PARA ENCENDER DISPLAY DECENAS
switch (contadorD) {
//ENCIENDE #0
case 0:
for (byte i = 0; i < 7; i++) {
digitalWrite (dos[i], decena0[i]);
}
break;
//ENCIENDE #1
case 1:
for (byte i = 0; i < 7; i++) {
digitalWrite (dos[i], decena1[i]);
}
break;
//ENCIENDE #2
case 2:
for (byte i = 0; i < 7; i++) {
digitalWrite (dos[i], decena2[i]);
}
break;
// ENCIENDE #3
case 3:
for (byte i = 0; i < 7; i++) {
digitalWrite (dos[i], decena3[i]);
}
break;
//ENCIENDE #4
```

```
case 4:
for (byte i = 0; i < 7; i++) {
digitalWrite (dos[i], decena4[i]);
}
break;
//ENCIENDE #5
case 5:
for (byte i = 0; i < 7; i++) {
digitalWrite (dos[i], decena5[i]);
}
break;
//ENCIENDE #6
case 6:
for (byte i = 0; i < 7; i++) {
digitalWrite (dos[i], decena6[i]);
}
break;
//ENCIENDE #7
case 7:
for (byte i = 0; i < 7; i++) {
digitalWrite (dos[i], decena7[i]);
}
break;
//ENCIENDE #8
case 8:
for (byte i = 0; i < 7; i++) {
digitalWrite (dos[i], decena8[i]);
}
break;
```

```

//ENCIENDE #9
case 9:
for (byte i = 0; i < 7; i++) {
digitalWrite (dos[i], decena9[i]);
}
break;
}

//DEFINICION DE AFORO
if(counter >= 40){ // SI AFORO MAYOR IGUAL A 40
if (counter >= 40) { // SI EL CONTADOR INTERNO MAYOR IGUAL A 40
digitalWrite(sir, 1); //ENCIENDE SIRENA
}
else {
digitalWrite(sir, 0); //APAGA sirena
}
}

lcd.clear(); //Limpia el LCD
lcd.setCursor(0,0); //Línea 00
lcd.print("AFORO PERMIT:"); //Despliega en Lcd el mensaje Aforo permitido
lcd.setCursor(14,0); //línea 14'0
lcd.print(contadorseleccion); //Despliega el valor del contador de selección

lcd.setCursor(0,1); //línea 0-1
lcd.print("AFORO ACTUAL:"); //Despliega en lcd el mensaje Aforo actual
lcd.setCursor(14,1); //línea 14-1
lcd.print( counter); //Despliega el valor del contador interno
delay (100); //tiempo de espera un segundo
}

```

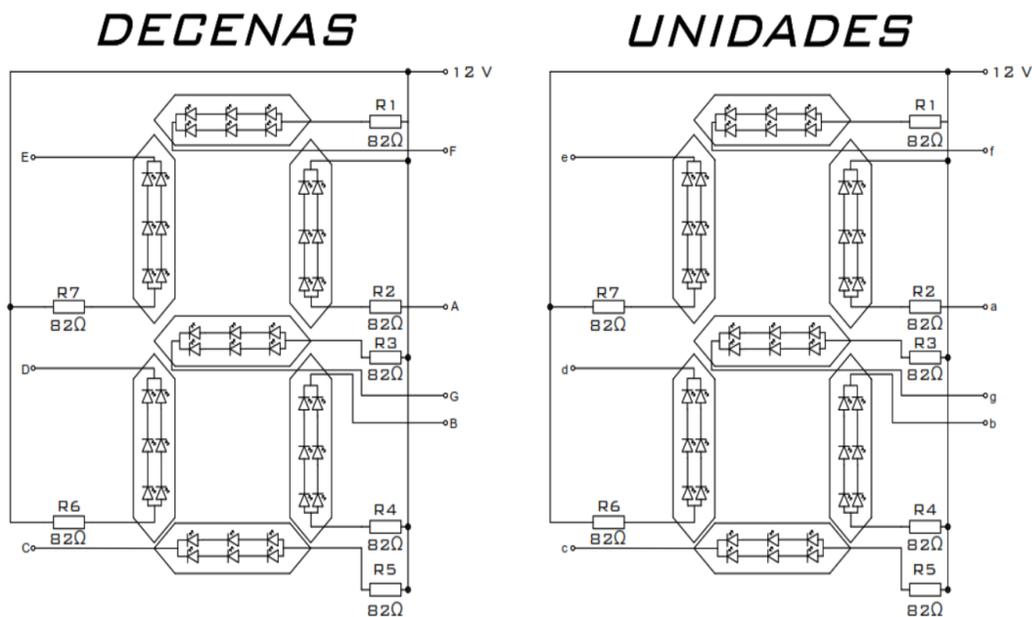
En la programación se explica en cada línea de código el uso de las sentencias como de las secuencias para el funcionamiento del sistema automático implementado.

3.3. Diseño del display indicador

A continuación, se muestra el diseño realizado para el adecuado funcionamiento del display indicador, el cual ha sido elaborado con leds de alta luminosidad color verde, para lo cual se ha tomado la forma de un display de 7 segmentos, cada uno con su debida nomenclatura para la conexión desde la placa Arduino hasta ellos.

Figura 13

Diseño de display indicador.



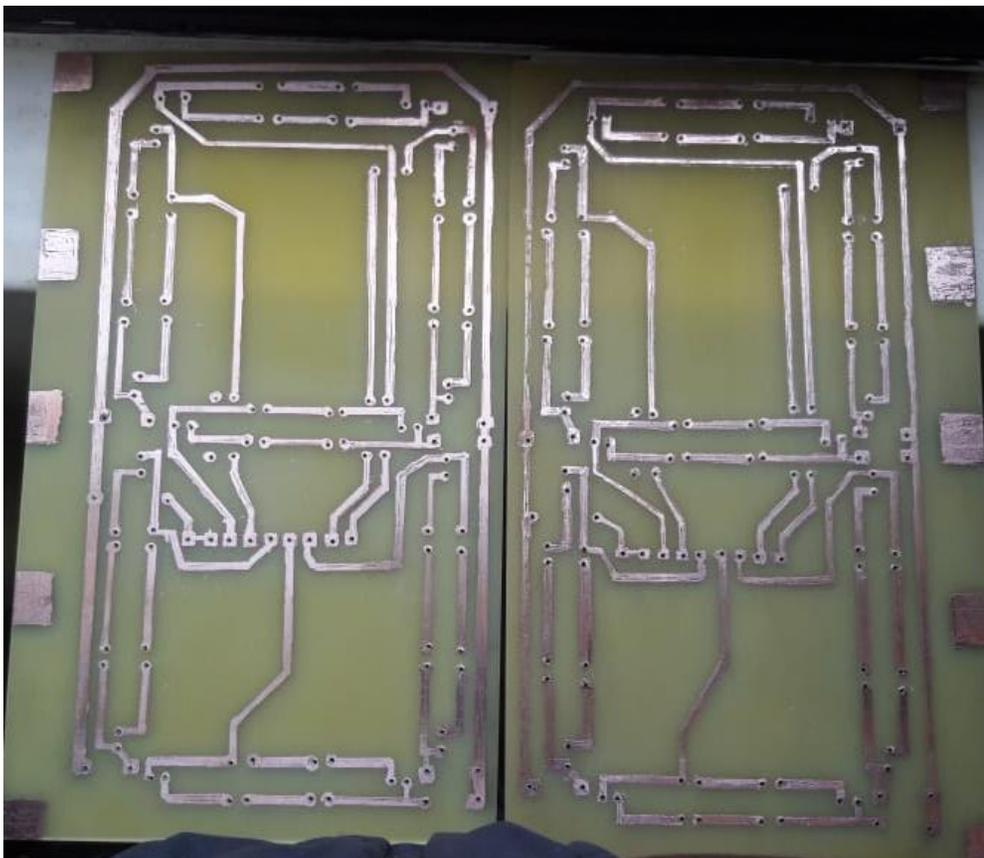
Nota. Esta imagen muestra la representación de cada uno de los display instalados con sus respectivos segmentos y nomenclatura para unidades y decenas.

Posterior a la implementación de los display de 7 segmentos de unidades y decenas respectivamente, se realiza el diseño de la placa en baquelita.

En la figura número 14 se muestra el diseño de las pistas en baquelita, para la disposición de segmentos y leds, respectivamente.

Figura 14

Diseño de pistas para display indicador.

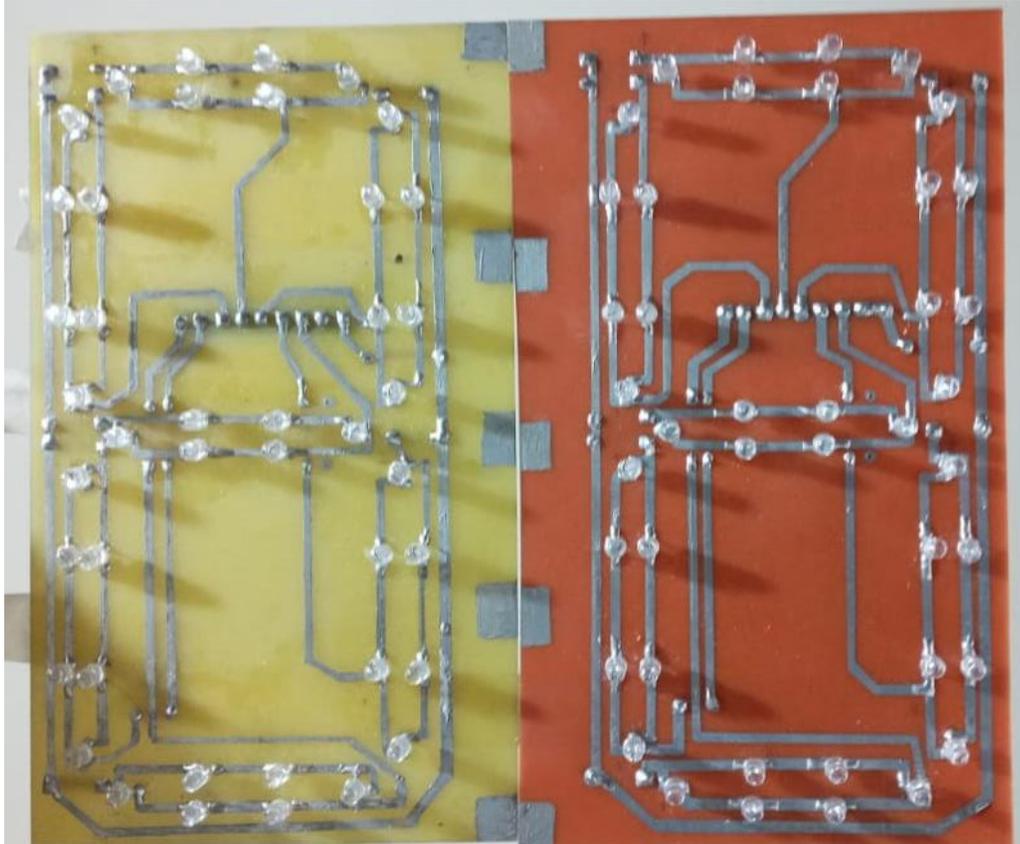


Nota. En esta imagen se muestra las pistas para los leds de alta luminosidad, de acuerdo con cada uno de los segmentos destinados para la programación.

En la figura 15 se indica el diseño y ensamblaje de los diodos leds, que forman cada uno de los segmentos del display indicador.

Figura 15

Diseño y montaje de los Leds en la placa de baquelita.



Nota. Esta imagen muestra la disposición de los leds en los display por segmentos indicados en la imagen anterior de unidades y decenas.

En la figura 16 se muestra la parte posterior de la placa implementada con disposición para cada uno de los segmentos correspondientes a conectar en la placa Arduino.

Figura 16

Parte posterior display 7 segmentos



Nota. En esta imagen se muestra la disposición de los pines.

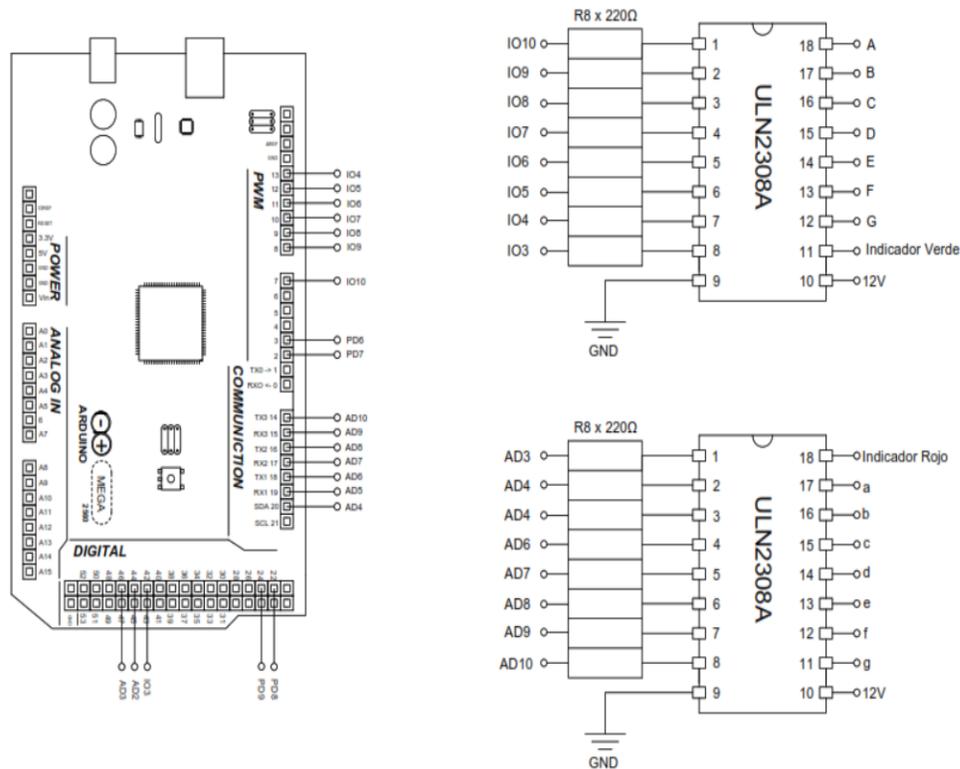
Una vez elaboradas las placas de baquelita, cada una de ellas se conectaron a cada pin del Arduino. De acuerdo a la asignación realizada en la programación.

3.4. Conexión Arduino – ULN2803A – Display Indicador

En la figura 17, se muestra la conexión que se realizó desde el Arduino hasta el circuito integrado ULN2803A, para que a la salida se realice el encendido de cada uno de los segmentos de acuerdo a la programación, donde se ejecuta la tabulación, para el conteo de número de personas que ingresan al local.

Figura 17

Conexión Arduino – Circuito integrado ULN2803A – Display Indicador



Nota. En esta imagen se muestra los pines de conexión desde el Arduino hacia los circuitos integrados.

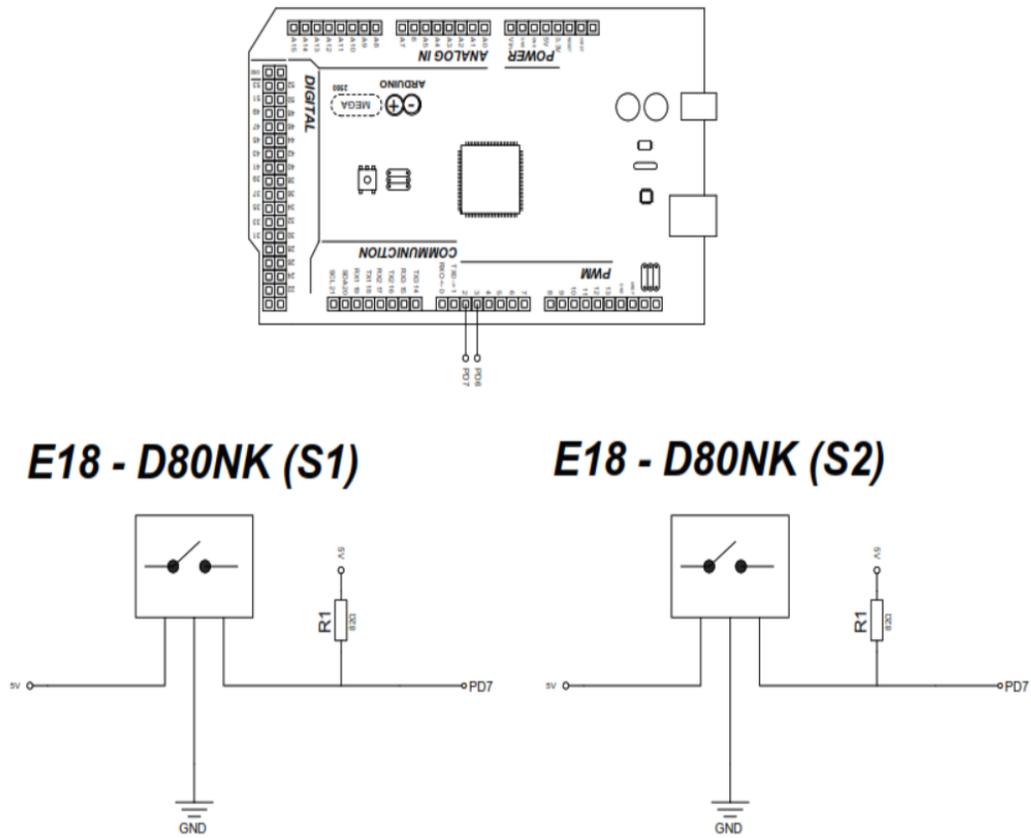
Cabe indicar que la conexión de los pines establecidos desde el Arduino hasta las resistencias de manera respectiva está determinada mediante la programación, de igual forma la salida para cada uno de los segmentos indicados anteriormente.

3.5. Conexión Arduino – Sensores

La conexión que a continuación se presenta es de acuerdo a los pines establecidos en los códigos de programación, con su alimentación respectiva.

Figura 18

Conexión Arduino a sensores E18 – D80NK

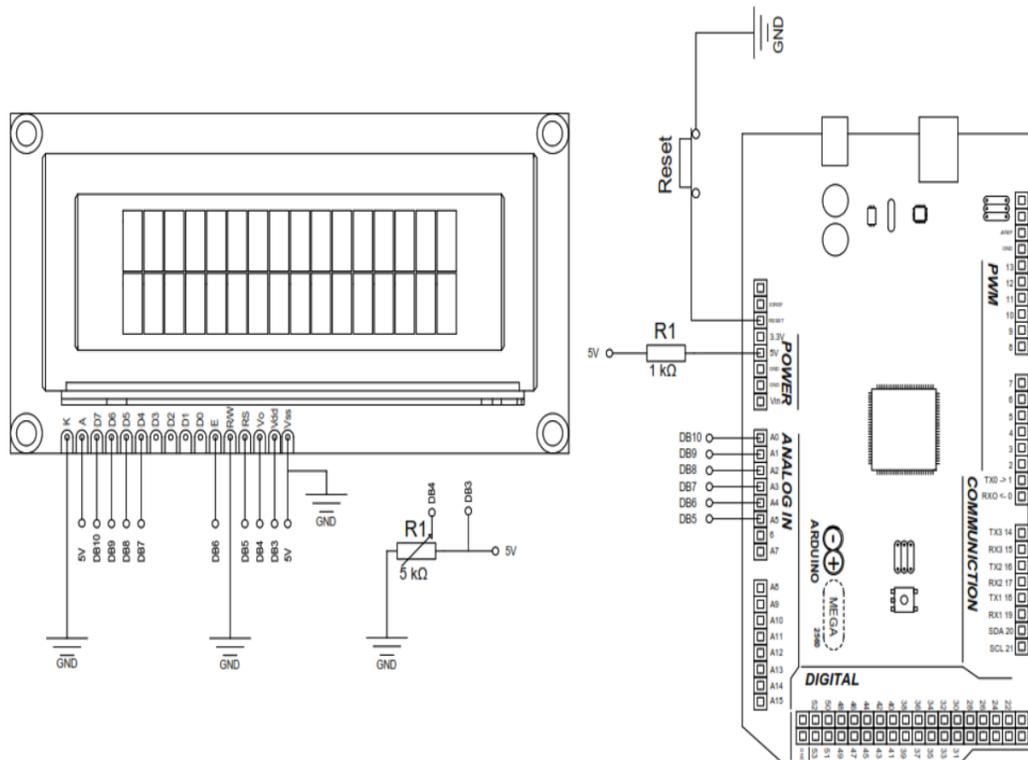


Nota. En la imagen se muestra la conexión para los dos sensores, con los pines establecidos mediante la programación.

La conexión de los pines se determinó en la programación de manera correspondiente que se realice el senso; ingreso y salida de las personas.

Figura 19

Conexión de Arduino a LCD



Nota. En esta imagen se muestra la disposición de conexión desde el Arduino al LCD.

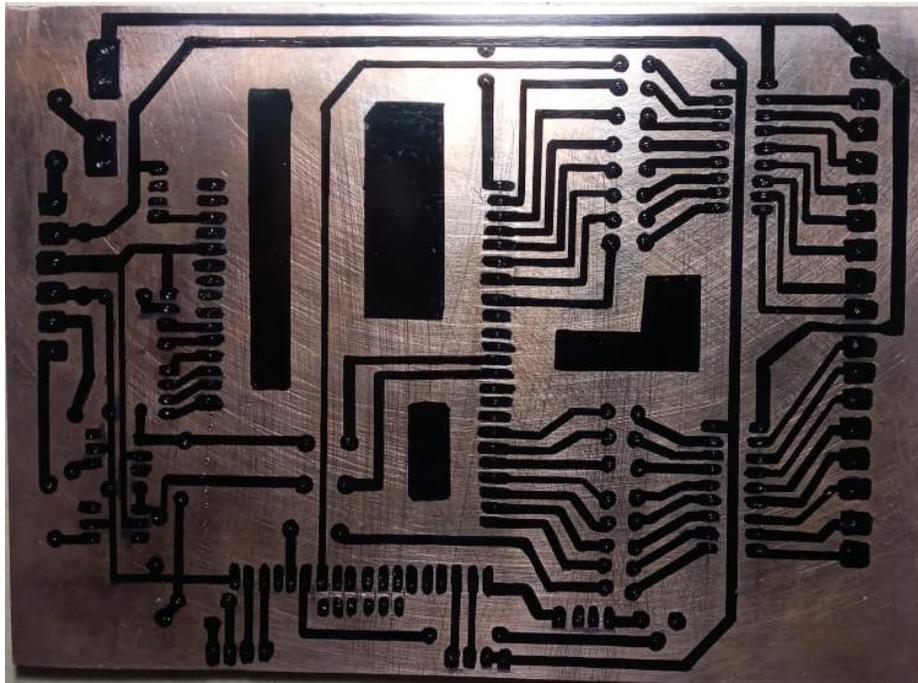
A continuación, se muestra el diseño de las pistas, para cada uno de los elementos adicionales para el control como:

- Pulsadores
- Sensores
- LCD
- Led's Indicadores

Estos elementos van en la caja de control en la cual se indica, la activación de los sensores, modificación para el aforo actual permitido, y el botón reset.

Figura 20

Diseño de pistas para LCD – Arduino, y sistema de control.



Nota. En esta imagen se muestra la disposición de las pistas para el sistema de control implementado.

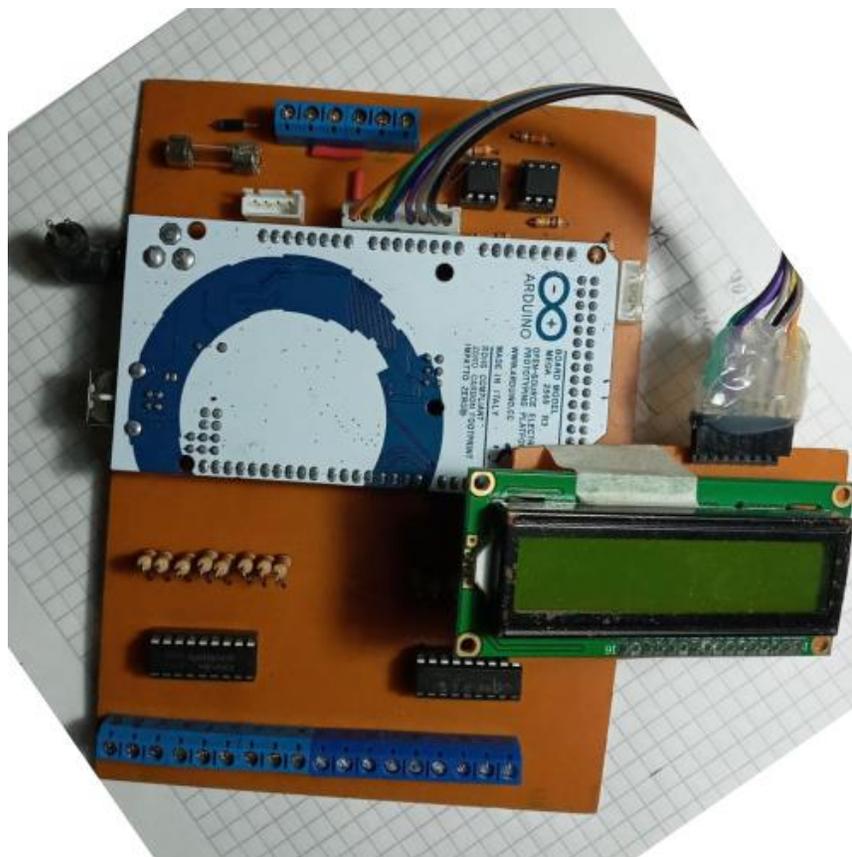
La implementación de cada uno de los elementos es de suma importancia para este sistema automático implementado, como el Arduino y una pantalla líquida (LCD) para la visualización y control del aforo.

3.6. Montaje de elementos en la placa.

En la figura 21 se muestra el montaje del LCD a la placa madre de Arduino correspondiente con cada uno de los elementos ya mencionados, para el control y funcionamiento explicado.

Figura 21

Montaje de los elementos en la placa madre.



Nota. En esta imagen se muestra el montaje del Arduino con los circuitos integrados, LCD.

3.7. Comprobación del funcionamiento.

En la figura 22 se muestra el LCD con los botones de manejo, por el cual se realiza el control principal para el aforo que se permite en el establecimiento.

Figura 22

Visualización de funcionamiento.



Nota. En esta imagen de comprobación se muestra el aforo permitido y actual, además de los botones con los cuales permite modificar las opciones.

En la figura 23 se presenta el aforo actual en el indicador como en el LCD, para tener un control automático mediante los dos medios dando el mismo valor en el LCD como en el indicador.

Figura 23

Representación display indicador.



Nota. En esta imagen se muestra el aforo actual, además de un indicador adicional el cual indica que el acceso es permitido.

En la figura 24 se visualiza el control que se da mediante el panel principal, para la asignación de las personas que pueden ingresar al establecimiento, determinando de esta manera si se cumple con el aforo máximo permitido, debido a las normas establecidas por las entidades de gobierno en turno.

Figura 24

Indicador de número máximo establecido.



Nota. En esta imagen se muestra que el aforo actual llegó al límite.

En la figura 25 se muestra la restricción que enciende la "X" la cual impide el ingreso mediante una alarma, que indica que se ha cumplido con el límite máximo de personas que han ingresado.

Figura 25

Representación display indicador.

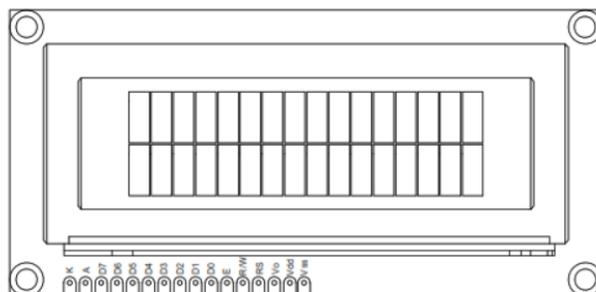


Nota. En esta imagen se muestra que ha llegado al límite y se restringe el ingreso.

En la figura 26 se muestra el diseño realizado previo al montaje de la vista frontal o control.

Figura 26

Diseño previo del panel de control.



Nota. En esta imagen se muestra cómo quedó definido los pulsadores para control del panel.

Concluido con este procedimiento se deja implementado el circuito que permite determinar el aforo actual del restaurante Sr. Pincho.

CAPÍTULO IV

4. Conclusiones y recomendaciones

4.1. Conclusiones

- Se ha implementado un sistema automático para determinar el aforo actual en el restaurante Sr. Pincho, que cumple con las disposiciones emitidas por los organismos de regulación actuales.
- Se estableció información básica y detallada referente a tarjetas de control que permitan realizar un conteo de personas mediante sensores implementados.
- Se analizó la afluencia de personas que se da en el restaurante Sr. Pincho para un control adecuado del ingreso y salida de personas.
- Los sistemas automáticos en la actualidad, son de gran utilidad, y gran aporte para la economía de muchas áreas a nivel nacional.

4.2. Recomendaciones

- Analizar las posibilidades comerciales para la implementación y diseño de un sistema automático.
- Verificar las líneas de programación para que los sensores cumplan con las sentencias de código planteadas.
- Dar a conocer los avances con prestaciones tecnológicas a la comunidad en general, ya que estos pueden ser de gran aporte para obtener ingresos económicos.

4.3. Bibliografía

- Arduino, I. (5 de Febrero de 2021). *IDE Arduino*. Obtenido de IDE Arduino: <https://www.arduino.cc/en/software>. Recuperado el 26 de marzo del 2021
- Atienza, F. B. (1997). *Diccionario politécnico de las lenguas españolas e inglesas*. Barcelona: Ediciones Díaz de Santos.
- Déleg, M. (2015). *Tecnología LED*. Cuenca, Azuay Ecuador: Universidad Politécnica Salesiana.
- Electrónica, K. (8 de Marzo de 2016). *Blogger*. Obtenido de Blogger: <https://www.kitelectronica.com/2016/02/tutorial-circuito-integrado-uln2803.html>. Recuperado el 25 de marzo del 2021
- Fernández, Y. (3 de Agosto de 2020). *Xalaka Basics*. Obtenido de Xalaka Basics: <https://www.xataka.com/basics/que-arduino-como-funciona-que-puedes-hacer-uno>. Recuperado el 26 de marzo del 2021
- Herrador, R. E. (2009). *Guía de Usuarios de arduino*. San Francisco, California, 94105, USA: Creative Commons.
- M.I. Juan Carlos Cedeño Vázquez, I. J. (2016). *Corriente y Resistencia Eléctrica*. Madrid: Trillas.
- Malvino, A. P. (2000). *Principios de Electrónica*. Madrid: McGraw-Hill.
- Martínez Hidalgo, I. G. (2020). *AUNAR*. Bogotá: Autonomía.
- Raymundo Barrales, V. B. (2019). *Laboratorio de Circuitos Eléctricos I*. Bogotá: Ediciones de la U.
- Routledge. (1997). *Diccionario Técnico Inglés*. Gran Bretaña: Routledge.p.104.
- Saldarriaga Cardona, L. M. (2012). *Sistema Electrónico*. Bogotá: Universidad Pedagógica Nacional.
- Young, J. D. (2020). *Diccionario enciclopédico de electrónica*. Madrid: Maeva. Recuperado el 25 de Marzo de 2021

ANEXOS