



UNIVERSIDAD LAICA ELOY ALFARO DE MANABI

FACULTAD DE CIENCIAS INFORMATICAS
CARRERA DE TECNOLOGIAS DE LA INFORMACION

TEMA:

TAREA DE RECUPERACION

NOMBRE:

MACIAS PICO JOSSELYN STEFANY

DOCENTE:

ING. MIKE MACHUCA

MATERIA:

SISTEMAS DIGITALES

FECHA:

FEBRERO 2020

MANTA-MANABÍ-ECUADOR



INDICE

¿QUE SON LOS LEDS?	10
EFICIENCIA Y PARÁMETROS OPERACIONALES:	10
CARACTERISTICAS:	11
PARTES DE UN LED.....	11
CONEXIÓN DE LOS LEDS:.....	12
RESISTENCIAS	12
RESISTENCIAS DE 330Ω	12
RESISTENCIA DE 560 Ω	13
ARDUINO	13
FUNCIONES GENERALES DE CADA PIN:	13
PARTE DE UN ARDUINO:	14
COMO SE CONECTAN VARIOS PINES DE ARDUINO	15
PROGRAMA ARDUINO	15
PULSADOR	16
¿QUE ES UN PULSADOR?	16
COMO CONECTAR UN BOTON A ARDUINO	16
NUMEROS BINARIOS	16
¿COMO CONVERTIR NUMERO DE BINARIOS A DECIMALES?.....	17
NUMEROS HEXADECIMALES	17
CIRCUITO INTEGRADO 74HC595.....	18
PARTES DE UN CIRCUITO INTEGRADO	19
RESISTENCIA DE VARIABLE	20
COMO CONECTAR EL PONTECIOMETRO AL ARDUINO	20
DISPLAY DE 7 SEGMENTOS	20
DISPLAY 7 SEGMENTOS CÁTODO COMÚN	21



DISPLAY 7 SEGMENTOS ÁNODO COMÚN	21
Funcionamiento	22
MATRIZ LED 8X8	23
PALABRAS RESERVADAS	24
PRACTICA 1: ENCENDER Y APAGAR UN LED	27
OBJETIVOS:	27
MATERIALES:	27
DIAGRAMA:	27
ESQUEMA:	28
PROGRAMACION:	28
RESULTADOS:	29
ANEXOS:	29
PRACTICA 2: ENCENDER Y APAGAR VARIOS LEDS	30
OBJETIVOS:	30
MATERIALES:	30
DIAGRAMA:	31
ESQUEMA:	31
PROGRAMACION:	31
RESULTADOS:	33
ANEXOS:	33
PRACTICA 3: ENCENDER Y APAGAR VARIOS LEDS EN TIEMPOS DIFERENTES	34
OBJETIVO:	34
MATERIALES:	34
DIAGRAMA:	35
ESQUEMA:	35
PROGRAMACION:	35



RESULTADOS:	37
ANEXOS:	38
PROYECTO 4: ENCENDER VARIOS LEDS CON FORMA CIRCULAR	38
OBJETIVOS:	38
MATERIALES:	38
DIAGRAMA:	39
ESQUEMA:	39
PROGRAMACION:	39
RESULTADOS:	53
ANEXOS:	53
PROYECTO 5: PRENDER Y APAGAR UN LED MEDIANTE UN PULSADOR	54
OBJETIVOS:	54
MATERIALES:	54
DIAGRAMA:	54
ESQUEMA:	55
PROGRAMACION:	55
RESULTADOS:	56
PROYECTO 6: MEDIANTE UN PULSADOR REALIZAR CAMBIOS DE ANIMACION	56
OBJETIVOS:	56
MATERIALES:	57
DIAGRAMA:	57
ESQUEMA:	58
PROGRAMACION:	58
RESULTADOS:	74
PROYECTO 7: CONECTAR RESISTENCIA VARIABLE	74
OBJETIVOS:	74



MATERIALES:	74
DIAGRAMA:	75
ESQUEMA:	75
PROGRAMACION:	76
RESULTADO:	76
PROYECTO 8: CONTROLAR EL RETARDO DEL LED CON RESISTENCIA	
VARIABLE	77
OBJETIVO:	77
MATERIALES:	77
DIAGRAMA:	78
ESQUEMA:	78
PROGRAMACION:	79
RESULTADOS:	79
PROYECTO 9: VARIAR NUMEROS MEDIANTE UNA RESISTENCIA VARIABLE 80	
OBJETIVOS:	80
MATERIALES:	80
DIAGRAMA:	80
ESQUEMA:	81
PROGRAMACION:	81
RESULTADOS:	82
PROYECTO 10: PRENDER LED MEDIANTE NUMERO INGRESADO 82	
OBJETIVOS:	82
MATERIALES:	82
DIAGRAMA:	83
ESQUEMA:	83
PROGRAMACION:	84
RESULTADOS:	86



PROYECTO 11: UTILIZAR POTENCIOMETRO PARA INDICAR LA TEMPERATURA DE UNA BODEGA	86
OBJETIVOS:	86
MATERIALES:	86
DIAGRAMA:	86
ESQUEMA:	87
PROGRAMACION:	87
RESULTADO:	89
PRACTICA 12: CONTADOR BINARIO DE 8 BITS.....	90
OBJETIVOS:	90
MATERIALES:	90
DIAGRAMA:	91
ESQUEMA:	91
PROGRAMACION:	91
RESULTADOS:	92
ANEXOS:	93
PRACTICA 13: JUEGO DE NUMEROS ALEATORIOS MEDIANTE NUMEROS BINARIOS	93
OBJETIVOS:	93
MATERIALES:	93
DIAGRAMA:	94
ESQUEMA:	94
PROGAMACION:	94
RESULTADOS:	96
ANEXOS:	97
PRACTICA 14: CONTADOR HEXADECIMAL DE UN BIT EN UN DISPLAY DE 7 SEGMENTOS AUTOMATICAMENTE	97



OBJETIVOS:	97
MATERIALES:	98
DIAGRAMA:	98
ESQUEMA:	99
PROGRAMACION:	99
RESULTADOS:	105
ANEXOS:	105
PRACTICA 15: CONTADOR HEXADECIMAL DE UN BIT POR MEDIO DEL DISPLAY DE 7 SEGMENTOS CON UN PULSADOR.....	106
OBJETIVOS:	106
MATERIALES:	106
DIAGRAMA:	107
ESQUEMA:	107
POGRAMACION:	107
RESULTADOS:	112
ANEXOS:	112
PRACTICA 16: MOSTRAR LOS NUMEROS HEXADECIMALES CON REGISTRO DE DESPLAZAMIENTO.....	113
OBJETIVOS:	113
MATERIALES:	113
DIAGRAMA:	113
ESQUEMA:	114
PROGRAMACION:	114
RESULTADOS:	115
ANEXOS:	116
PRACTICA 17: MOSTRAR LOS DATOS DEL 00 AL 99 AUTOMATICAMENTE..	116
OBJETIVOS:	116



MATERIALES:	116
DIAGRAMA:	117
ESQUEMA:	117
PROGRAMACION:	117
RESULTADOS:	119
ANEXOS:	119
PRACTICA 18: CONECTAR UNA MATRIZ LED 8 X 8 CON REGISTRO DE DESPLAZAMIENTO.	120
OBJETIVO:	120
MATERIALES:	120
DIAGRAMA:	120
ESQUEMA:	121
PROGRAMACION:	121
RESULTADOS:	132



MARCO TEORICO



¿QUE SON LOS LEDS?

Un diodo emisor de luz o led (light-emitting diode) es una fuente de luz constituida por un material semiconductor dotado de dos terminales. Se trata de un diodo de unión p-n, que emite luz cuando está activado.

Un led comienza a emitir cuando se le aplica una tensión de 2-3 voltios. En polarización inversa se utiliza un eje vertical diferente al de la polarización directa para mostrar que la corriente absorbida es prácticamente constante con la tensión hasta que se produce la ruptura.

EFICIENCIA Y PARÁMETROS OPERACIONALES:

El color que emite cada diodo LED en particular depende principalmente del material semiconductor que se haya empleado en su fabricación. Cada compuesto químico propio del material semiconductor utilizado en la fabricación de un diodo LED permite la emisión de una luz de un color específico, correspondiente a una determinada longitud de onda del espectro electromagnético.

	COLOR	LONGIRUD DE ONDA	COEFICIENTE DE EFICIENCIA	EFICIENCIA LUMINOSA
	Rojo	$620 < \lambda < 645$	0.39	72
	Anaranjado	$610 < \lambda < 620$	0.29	98
	Verde	$520 < \lambda < 550$	0.15	93
	Cian	$490 < \lambda < 520$	0.26	75
	Azul	$460 < \lambda < 490$	0.35	37
	Radiacion Infraroja	$\lambda > 760$	$\Delta V < 1.63$	
	Amarillo	$570 < \lambda < 610$	$1.63 < \Delta V < 2.03$	
	Violeta	$450 < \lambda < 500$	$2.76 < \Delta V < 4.0$	
	Purpura	Combinacion de dsitintos tipos	$2.48 < \Delta V < 3.7$	
	Ultravioleta	$\lambda < 400$	$3 < \Delta V < 4.1$	



	Rosa	Combinacion de dsitintos tipos	$\Delta V \sim 3.3$	
	Blanco	Espectro amplio	$2.8 < \Delta V < 4.2$	

(Wikipedia, 2020)

CARACTERISTICAS:

Todos los diodos LEDs poseen una construcción sólida. Lo que les permite absorber vibraciones extremas y golpes sin llegar a romperse ni estropearse. El chip emisor de luz es la parte principal de un LED y se encuentra encerrado en una envoltura o cápsula de resina epoxi transparente o también tintada del mismo color de la luz que emiten, pudiendo tener formas y tamaños diferentes. Desde el mismo chip que se encuentra encerrado en el interior de la cápsula parten dos terminales que atraviesan su base y salen al exterior para que se puedan conectar a un circuito eléctrico de corriente directa (C.D.), de forma tal que el LED quede polarizado directamente. Cuando el LED es nuevo, el terminal más corto corresponde al polo negativo (-) del chip, mientras el más largo corresponde al polo positivo (+).

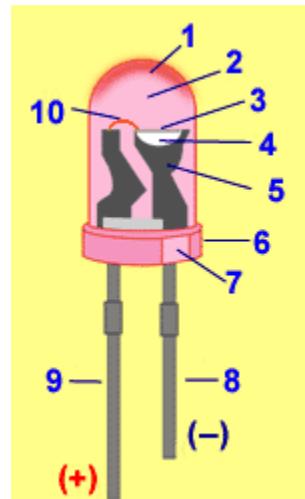


IMAGEN 1 Partes de un LED

PARTES DE UN LED

1. Lente que forma parte del encapsulado o envoltura protectora del LED
2. Encapsulado de resina epoxy
3. Chip o Diodo semiconductor emisor de luz
4. Copa reflectora
5. Yunque
6. Base
7. Marca plana que identifica el terminal correspondiente al catodo del chip
8. Terminal de conexión externa negativo correspondiente al catodo es decir la parte mas corta



9. Terminal de conexión externa positiva correspondiente al anodo es decir la parte mas larga
10. Alambre de oro muy fino conectado al anodo del chip y al terminal positivo de conexión externa. (Alvarez, s.f.)

CONEXIÓN DE LOS LEDS:

Cada LED debe estar conectado a su respectivo pin. El anodo que es la parte positiva del LED tiene que estar conectada a el pin del Arduino el cual debería estar declarada la salida en el programa y a una parte de la resistencia la cual se estará utilizando y la otra parte de esta se conectara al pin de tierra. Así mismo el catodo que es la parte negativa va conectada a tierra en el Arduino

RESISTENCIAS

Las resistencias son uno de los tipos básicos de componentes electrónicos. Tienen dos terminales y un semiconductor, está formada por carbón y otros elementos resistivos.

Una resistencia es un elemento pasivo que disipa energía en forma de calor según la ley de Joule. También establece una relación de proporcionalidad entre la intensidad de corriente que la atraviesa y la tensión medible entre sus extremos, relación conocida como ley de Ohm. En general, una resistencia podrá tener diferente comportamiento en función del tipo de corriente que circule por ella.

Las resistencias son uno de los tipos básicos de componentes electrónicos. Tienen dos terminales y un semiconductor, está formada por carbón y otros elementos resistivos. Un semiconductor es justo lo que parece: algo que conduce la electricidad, pero no tan bien. Mientras que los conductores como el cobre y el oro se utilizan en circuitos para permitir el flujo de electricidad libremente, un semiconductor se utiliza para proporcionar una cierta resistencia al flujo de electricidad. Es por eso que una resistencia tiene este nombre. Cuando la electricidad pasa a través de los semiconductores, parte de ella se convierte en calor.

Cuanto mayor sea el voltaje, mayor es la energía. (Geekbot Electronics, s.f.)

RESISTENCIAS DE 330Ω

Los valores de la resistencia de 330 se muestra en el siguiente cuadro:



IMAGEN	COLOR PRIMERO	COLOR SEGUNDO	MULTIPLICADOR	TOLERANCIA
Four resistors with orange bands.	Naranja = 3	Naranja = 3	Negro = x 1	Dorado = 5%

(Electronica , s.f.)

RESISTENCIA DE 560 Ω

Tiene los colores verde, azul y café con una tolerancia de 5%

IMAGEN	COLOR PRIMERO	COLOR SEGUNDO	MULTIPLICADOR	TOLERANCIA
A resistor with green, blue, and brown bands.	Verde = 5	Azul = 6	Cafe= x 10	Dorado = 5%

(Telesa Online, s.f.)

ARDUINO

El Arduino Uno es una placa de microcontrolador de código abierto basado en el microchip ATmega328P y desarrollado por Arduino.cc.

La placa está equipada con conjuntos de pines de E/S digitales y analógicas que pueden conectarse a varias placas de expansión y otros circuitos. La placa tiene 14 pines digitales, 6 pines analógicos y programables con el Arduino IDE a través de un cable USB tipo B.

Puede ser alimentado por el cable USB o por una batería externa de 9 voltios, aunque acepta voltajes entre 7 y 20 voltios.

El proyecto Arduino se inició en el Interaction Design Institute Ivrea (IDII) en Ivrea, Italia, en el 2005. (Wikipedia, s.f.)

FUNCIONES GENERALES DE CADA PIN:

- LED: hay un LED incorporado controlado por el pin digital 13. Cuando el pin tiene un valor alto, el LED está encendido, cuando el pin está bajo, está apagado.



- VIN: el voltaje de entrada a la placa Arduino/Genuino cuando se utiliza una fuente de alimentación externa (a diferencia de los 5 voltios de la conexión USB u otra fuente de alimentación regulada).
- 5V: Este pin emite 5V regulado desde el regulador en el tablero. La placa se puede alimentar con el conector de alimentación de CC (7-20 V), el conector USB (5 V) o el pin VIN de la placa (7-20 V).
- 3V3: un suministro de 3,3 voltios generado por el regulador de la placa. El consumo máximo de corriente es de 50 mA.
- GND: Pines de tierra.
- IOREF: este pin en la placa Arduino/Genuino proporciona el voltaje de referencia con la que funciona el microcontrolador. Un blindaje configurado correctamente puede leer el voltaje del pin IOREF y seleccionar la fuente de alimentación adecuada o permitir que los traductores de voltaje en las salidas funcionen con 5V o 3.3V.
- Reset: normalmente se usa para agregar un botón de restablecimiento a los aislantes que bloquean el que está en el tablero.⁶

PARTES DE UN ARDUINO:

1. Potencia USB
2. Conector del adaptador
3. Pin a tierra
4. Pin a 5 v
5. Pin a 3.3 v
6. Pines analógicos
7. Pines digitales
8. PWM (estos pines ~ pueden actuar como digitales normales, pero también se pueden utilizar como modulación por ancho de pulsos).
9. AREF: Soportes de referencias analógicas
10. Botón de reinicio
11. Indicador LED de alimentación
12. Leds RX TX

(Descubre Arduino , s.f.)

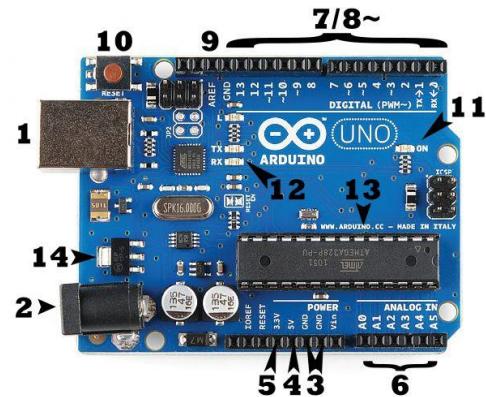


IMAGEN 2 Partes de un Arduino



COMO SE CONECTAN VARIOS PINES DE ARDUINO

Para conectar cada led en un distinto pin del Arduino primero se necesita definir los leds en la segunda parte del programa cada led siendo igual al pin del cual se conectara en el arduino, después en la tercera parte del programa se necesita declarar cada uno de los componentes con un tipo OUTPUT escribiendo `pinMode(variable, OUTPUT);`. En la siguiente parte que es el loop se realizan las líneas de código las cuales se ejecutarán para que se encienda o se apague cada uno de los leds.

PROGRAMA ARDUINO

El programa de Arduino para realizar una programación consta de cuatro partes:

- La primera parte es donde se encuentran las librerías
- La segunda parte es donde se encuentran las Constantes, variables globales y objetos de clases que después se usarán durante la programación
- La tercera parte es donde se encuentra el setup, esta función solo se ejecuta al iniciar el Arduino y no se vuelve a ejecutar más. En esta parte se puede ejecutar pines y ejecutar las operaciones que consideramos oportunas. Ejemplo:

```
void setup() {  
    //declarar entradas y  
    salidas y variables  
  
    pinMode(9, OUTPUT);  
    //cuales serán los pines de  
    entrada y pines de  
    salida  
}
```

- La cuarta parte es la última parte del código esta parte se hace infinitamente y de parte cíclica, es decir que cuando se haya realizado la función setup y la función loop entrará en ejecución y se vuelve a ejecutar otra vez hasta que se apague o desconecte el Arduino. Esta es la parte principal de todo el programa. Ejemplo:

```
void loop() {
```



IMAGEN 3 Logo del programa de Arduino



//LO QUE VA HACER, VA A LEER DEFINITIVAMENTE TODO... TERMINA Y REGRESA

```
digitalWrite(9, HIGH); //ENCIENDA
delay (500); //tiempo de retardo
digitalWrite(9, LOW); // APAGADO
delay (500); //tiempo de retardo
}
```

(Diwo, s.f.)

PULSADOR

¿QUE ES UN PULSADOR?

Un pulsador es un dispositivo simple con dos posiciones, EN encendido y AP apagado (Diario Electronico, s.f.). Ademas, un pulsador es un operador eléctrico que, cuando se oprime, permite el paso de la corriente eléctrica y, cuando se deja de oprimir, lo interrumpe.

COMO CONECTAR UN BOTON A ARDUINO

Un botón puede conectarse a cualquier pin de arduino (digital o analógico, ya que los analógicos usualmente funcionan también como digitales). (Geekfactory , s.f.)



IMAGEN 4 Pulsador

Una patilla debe ir al positivo (5V o a la línea positiva de la protoboard, alimentada por el 5V del arduino). La otra patilla debe tener doble conexión: a una resistencia, y de ahí al negativo o GND, y al pin digital con el que vamos a controlar el pulsador.

(Programoergos, s.f.)

NUMEROS BINARIOS

El sistema binario emplea sólo dos dígitos los cuales son cero (0) y uno (1). A nivel interno este sistema funciona con dos grados diferentes de voltaje, dentro del sistema binario para indicar el apagado equivalente a «cero voltios» se representa por (0) o el encendido equivalente a +5 o +12 voltios representado por el (1). (Definicion , s.f.)



¿COMO CONVERTIR NUMERO DE BINARIOS A DECIMALES?

Para convertir números de binario a decimal necesitamos seguir algunos pasos escritos a continuación:

1. Se escribe el número binario. 10011011_2
2. Escribe las potencias de dos de derecha a izquierda, empieza en 2^0 , dándole un valor de "1", incrementa el exponente en uno en cada potencia. $2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0$
3. Détente cuando la cantidad de elementos de la lista sea igual a la cantidad de dígitos del número binario. 128, 64, 32, 16, 8, 4, 2, 1
4. Escribelos de forma ordenada cada numero de la potencia con cada numero binario.
 $1 \rightarrow 128 \quad 0 \rightarrow 64 \quad 0 \rightarrow 32 \quad 1 \rightarrow 16 \quad 1 \rightarrow 8 \quad 0 \rightarrow 4 \quad 1 \rightarrow 2 \quad 1 \rightarrow 1$
5. Luego eliminas los números que al multiplicar el binario por el decimal te da igual a 0
 $1 \rightarrow 128 \quad 1 \rightarrow 16 \quad 1 \rightarrow 8 \quad 1 \rightarrow 2 \quad 1 \rightarrow 1$
6. Luego sumas todos los valores decimales que te quedan, y esa suma será el resultado de su conversión.

$$128+16+8+2+1 = 155$$

(cv.uoc.edu, s.f.)

NUMEROS HEXADECIMALES

El sistema hexadecimal es el sistema de numeración posicional que tiene como base el 16. Sus números están representados por los 10 primeros dígitos de la numeración decimal, y el intervalo que va del número 10 al 15 están representados por las letras del alfabeto de la 'A' a la 'F'.

Su uso actual está muy vinculado a la informática y ciencias de la computación donde las operaciones de la CPU suelen usar el byte u octeto como unidad básica de memoria. Aunque los circuitos electrónicos digitales y las computadoras utilizan el sistema binario, el trabajar con este sistema de numeración es bastante más complicado, lo que da como resultado una gran posibilidad de cometer errores se trabaja con números binarios demasiado largos.

Este sistema posee dos grandes ventajas en el entorno informático:

- Crea una simplificación en la escritura de los números decimales, ya que cada 4 cifras binarias se representa simplemente por una hexadecimal.



- Cada cifra hexadecimal se puede expresar por 4 cifras binarias, con lo que la transposición entre estos dos sistemas se facilita considerablemente. Para convertir un numero binario a hexadecimal se realizará el mismo proceso pero de forma inversa.

Informacion mas detallada en la siguiente tabla:

DECIMAL	BINARIO	HEXADECIMAL
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

(electrontools, s.f.)

CIRCUITO INTEGRADO 74HC595

El 74HC595 es un registro de desplazamiento de 8 bit con una entrada serie y salida paralelo, la gran utilidad de esto es poder utilizar y controlar 8 salidas con tan solo 3 pines conectados al arduino.



PARTES DE UN CIRCUITO INTEGRADO

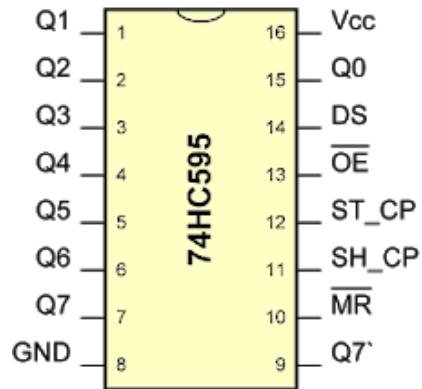


IMAGEN 5 Partes de un Circuito

PIN	SYMBOL	DESCRIPCION
1	Q1	Parallel data output (bit -1)
2	Q2	Parallel data output (bit -2)
3	Q3	Parallel data output (bit -3)
4	Q4	Parallel data output (bit -4)
5	Q5	Parallel data output (bit -5)
6	Q6	Parallel data output (bit -6)
7	Q7	Parallel data output (bit -7)
8	GND	Ground 0 V
9	Q7'	Serial Data output
10	MR	Master Reset (Active Low)
11	SH_CP	Shift Register Clock Input
12	ST_CP	Storage Register Clock Input
13	OE	Output Enable (Active Low)
14	DS	Serial Data input
15	Q0	Parallel data output (bit-8)
16	Vcc	Positive Supply Voltaje



RESISTENCIA DE VARIABLE

Una resistencia ajustable o potenciómetro es una resistencia cuyo valor podemos modificar moviendo su eje o cursor. Entre los extremos del potenciómetro el valor siempre es el mismo; pero entre un extremo y el punto intermedio tendremos una resistencia variable desde 0 al valor especificado. Su símbolo es el de la figura adjunta:



IMAGEN 7n Resistencia Variable

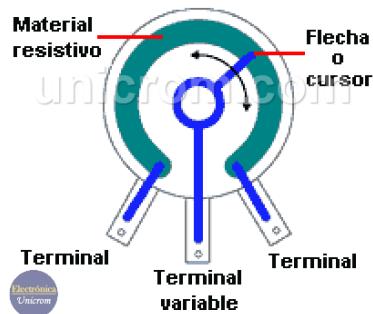


IMAGEN 6 Partes de la
Resistencia Variable

Hay dos tipos principales de potenciómetros y los dos comparten el mismo principio de funcionamiento.

- Lineales: Son aquellos que se controlan deslizando el cursor sobre una pista resistiva.
- Rotatorios: Son aquellos que se controlan girando un eje. Ejemplo: el control de volumen de un equipo de sonido. dentro de este tipo de potenciómetro están los potenciómetros múltiples, que son varios que comparten un mismo eje. (Unicrom, s.f.)

COMO CONECTAR EL PONTECIOMETRO AL ARDUINO

1. Identificar los pines del potenciómetro
2. Conectar el borde izquierdo a tierra
3. El segundo pin se conecta al pin designado en el arduino
4. El tercer borde se conecta a la salida de 5v

DISPLAY DE 7 SEGMENTOS

El display 7 Segmentos es un dispositivo opto-electrónico que permite visualizar números del 0 al 9. Existen dos tipos de display, de cátodo común y de ánodo común.



Es importante mencionar que los display de 7 segmentos, dado que están construidos con diodos LED, requieren una corriente máxima. En otras palabras se requiere colocar una resistencia para limitar la corriente. Dicha resistencia depende de la corriente que se quiera suministrar al LED así como de la caída de voltaje.

El display de 7 segmentos tiene una estructura casi estándar en cuanto al nombre de los segmentos. Para dicho elemento, se cuenta con 7 leds, uno para cada segmento. Para cada segmento, se le asigna una letra desde la «a» hasta la «g». (Hetpro, s.f.)

DISPLAY 7 SEGMENTOS CÁTODO COMÚN

El display cátodo común es aquel que tiene el pin común conectado a los negativos de los LED's (cátodo). Esto significa que este tipo de display se «controla» con '1' s lógicos o con voltaje positivo. El arreglo para un display de cátodo común sería el siguiente:

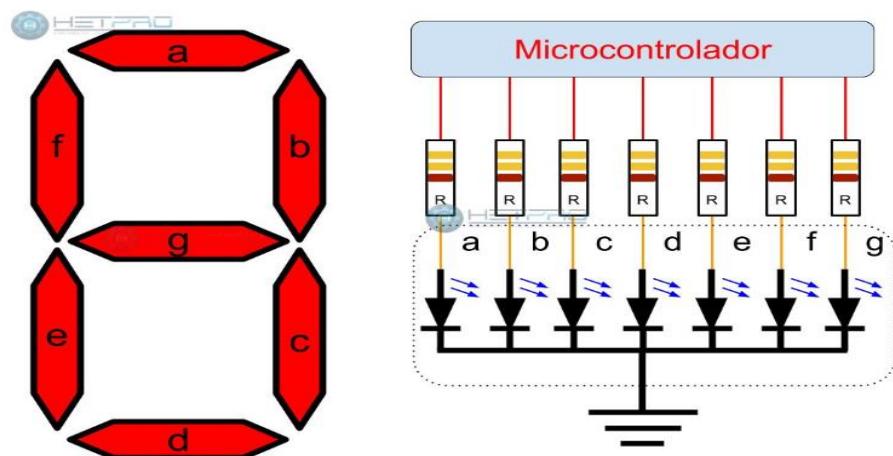


IMAGEN 8 Display Anodo comun

DISPLAY 7 SEGMENTOS ÁNODO COMÚN

El display ánodo común es aquel cuyos ánodos están conectados al mismo punto. Este tipo de display es controlado por ceros, es decir que el microcontrolador o MCU, FPGA o microprocesador, le asigna a cada segmento un cero lógico (también llamada tierra digital). El esquema o diagrama del display de 7 segmentos en ánodo común es: (Heptro, s.f.)

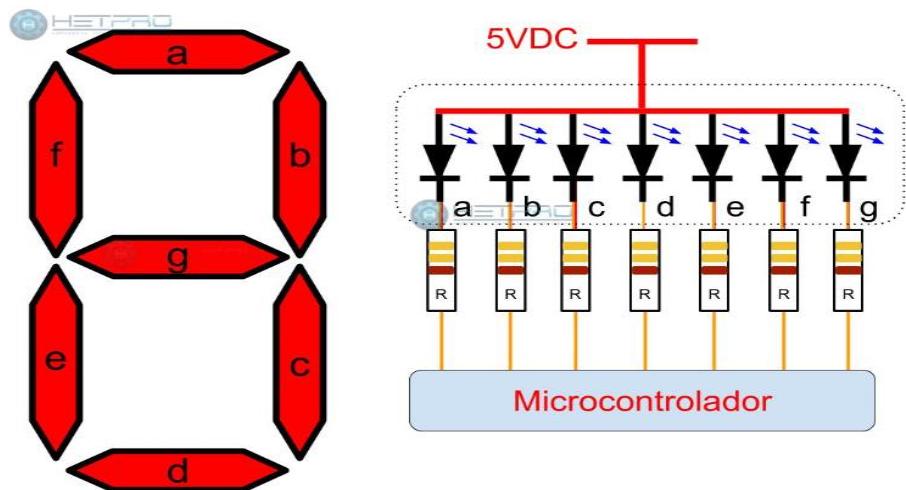


IMAGEN 9 Díplay Anodo Común

Funcionamiento

El funcionamiento del display es tan sencillo que tomando en cuenta las medidas propuestas

Para encender un segmento es necesario poner su estrada en alto (1, para mantenerlo encendido debe estar en bajo (0))

Para ello tenemos dos ejemplos a continuación (Electron Tools, s.f.)

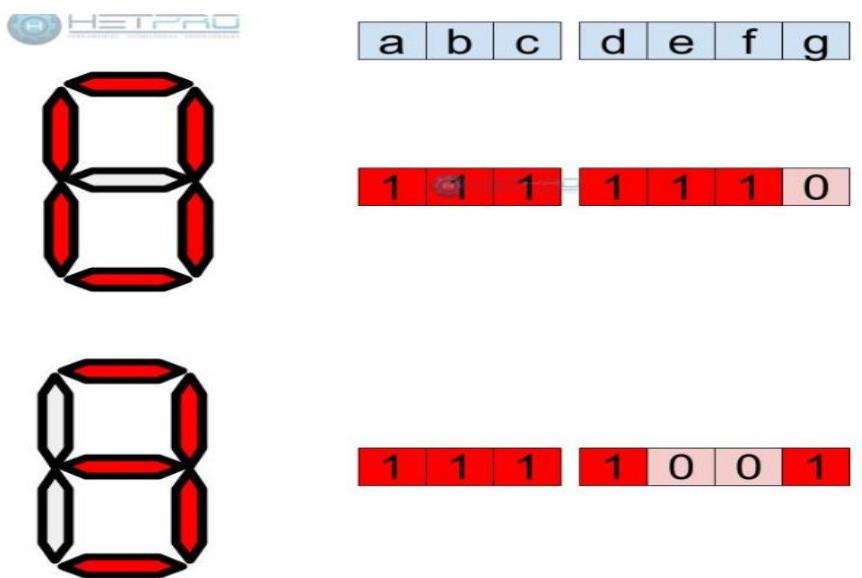


IMAGEN 10 Funcionamiento de los displays



MATRIZ LED 8X8

Una matriz LED (o Array LED) consiste en una matriz de diodos LED normales y corrientes. La podemos encontrar de múltiples maneras y colores. Pueden ir desde sólo un color, a varios colores, incluso una matriz RGB.

La idea, es que como tenemos una matriz de 8×8 podemos dibujar letras y símbolos de la misma.

manera que ya hicimos en el pasado, cuando definimos el símbolo de grado, para el display LCD. Es decir, definiendo matrices de puntos para representar las letras, que luego sacaremos por el Display.

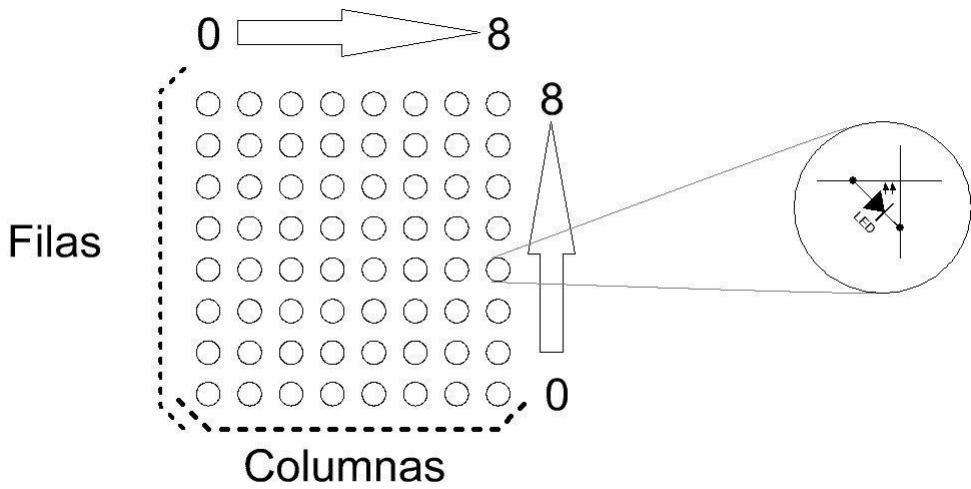
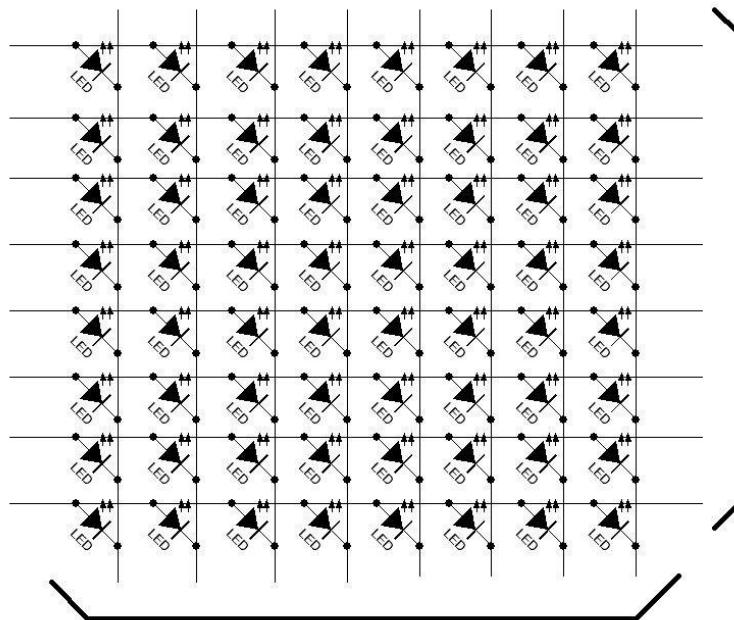


IMAGEN 11 Partes de la matriz



FILAS

Se activarán con transistores conectados al Puerto B del microcontrolador

COLUMNAS

IMAGEN 12 Partes de una matriz

PALABRAS RESERVADAS

int x=Serial.parseInt();

Serial.parseInt(): Busca el siguiente entero valido en la serie entrante

Serial.flush();

Esta espera que se complete la transmisión de los datos salientes

randomSeed(analogRead(0));

randomSeed(): inicializa el generador de números aleatorios con una entrada bastante aleatoria

analogRead(0): genera un número aleatorio

numero = random(255);

random(): es el total de datos que necesita conseguir aleatoriamente

shiftOut (pin14, pin11, bitOrder, x);

shiftOut: Es el que convierte numeros en BIT.

pin14: es el pin que genera cada bit.

pin11: es el que se utiliza para alternar, una vez que el anterior se haya configurado.

bitOrder: Es la dirección con la que se realiza el desplazamiento de los bits estos son:

- **MSBFIRST:** Primero el bit mas significativo, es decir bit mas a a izquierda.
- **LSBFIRST:** Primero el bit menos significativo, es decir bit mas a la derecha.



x: es el valor de los cuales se comenzará a desplazar para convertir el dato.

pinMode(pulsador,INPUT_PULLUP);

INPUT_PULLUP: Este invierte efectivamente el comportamiento del modo ENTRADA, donde ALTO significa que el sensor está apagado y BAJO significa que el sensor está encendido.

attachInterrupt(digitalPinToInterrupt(pulsador),conta,RISING);

digitalPinToInterrupt(pulsador): Número de interrupción, es decir, en que pin se “escuchará” para esperar la interrupción.

conta: función que se llamará en caso de que se produzca la interrupción.

RISING: se ejecuta mientras el valor va de 0 a 1 (o de 0 a 5 Voltios).



**PRACTICAS DEL
PRIMER Y SEGUNDO
PARCIAL DEL TERCER
SEMESTRE**



PRACTICA 1: ENCENDER Y APAGAR UN LED

OBJETIVOS:

- Conocer un poco sobre los leds, su funcionamiento sus polaridades.
- Conocer sobre las resistencias de 330 Ohmios.
- Aprender a usar un Arduino Uno
- Aprender lo básico de programar en Arduino

MATERIALES:

- 1 LED.
- Una resistencia de 220Ω .
- Una tarjeta Arduino Uno.
- Un cable USB impresora.
- Un computador.
- Cables jumper.
- Protoboard.
- Programa Arduino

DIAGRAMA:

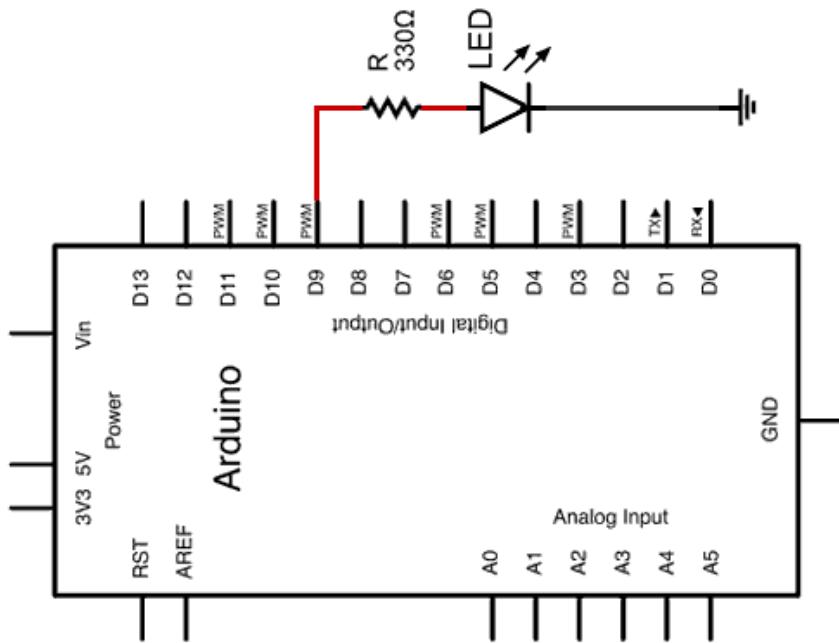
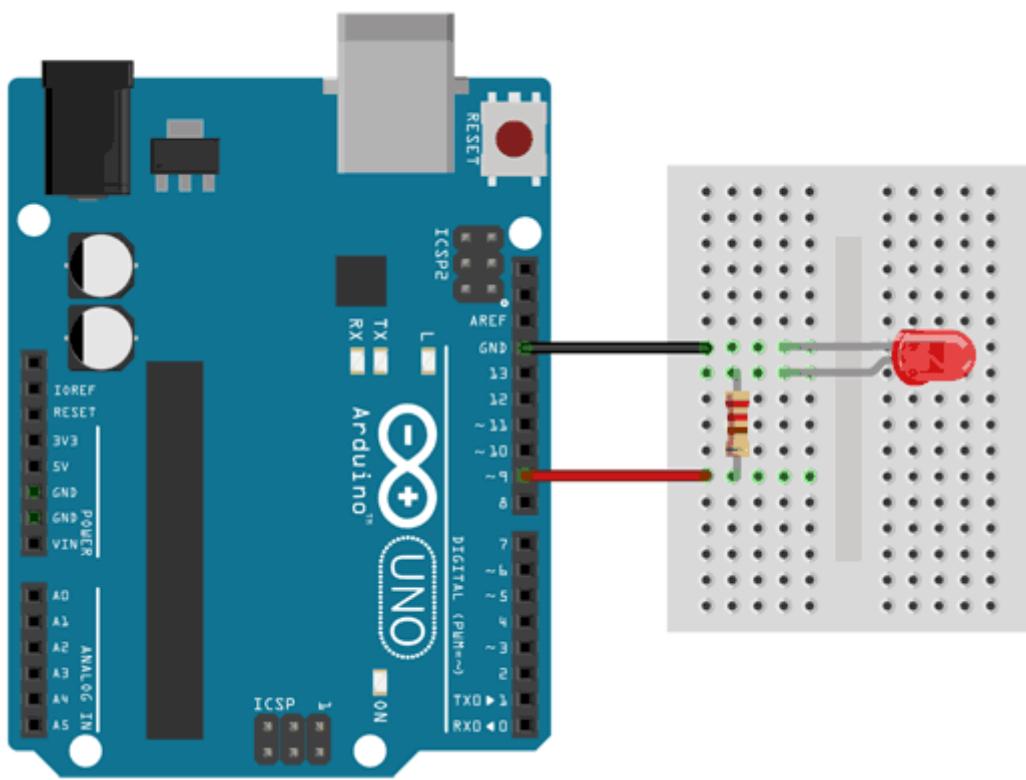


DIAGRAMA 1



ESQUEMA:



ESQUEMA 1

PROGRAMACION:

```
/*
 *
 *      ULEAM-FACCI
 *
 *TECNOLOGIA DE LA INFORMACION
 *
 *      SISTEMAS DIGITALES
 *
 *MACIAS PICO JOSSELYN STEFANY
 *
*****
* *****PRACTICA N° 1 *****
*
*      _____HOLA MUNDO DE ARDUINO_____
*
*PARA TAREA COMENTAR TODO EL CODIGO
*/
void setup() {
```



```
//declarar entradas y salidas y variables
pinMode(9,OUTPUT); //cuales seran los pines de entrada y pines de
salida
}
void loop() {
    //LO QUE VA HACER, VA A LEER DEFINITIVAMENTE TODO... TERMINA Y
REGRESA
    digitalWrite(9, HIGH); //ENCIENDA
    delay (500); //tiempo de retardo
    digitalWrite(9, LOW); // APAGADO
    delay (500); //tiempo de retardo
}
```

RESULTADOS:

Esta practica se realizo con el objetivo de aprender a utilizar un Arduino Uno incluido el programa y asi mismo conocer sus partes y se obtuvo como resultado uno de los primeros circuitos donde se prende y se apaga un led por medio de un delay(); es decir un retardo. No se obtuvo problema con las conexiones ni con el programa.

ANEXOS:

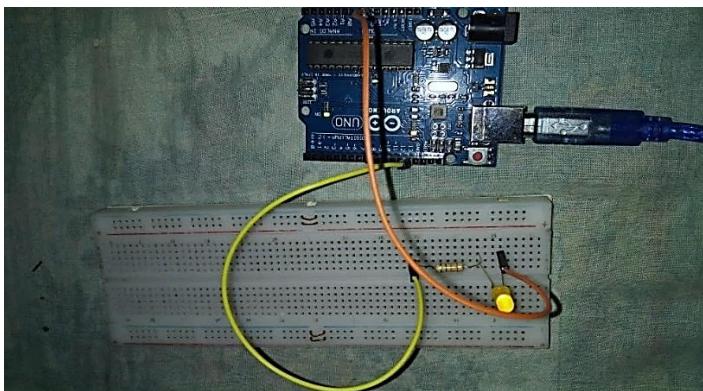


IMAGEN 13 Practica 1 ENCENDIDO

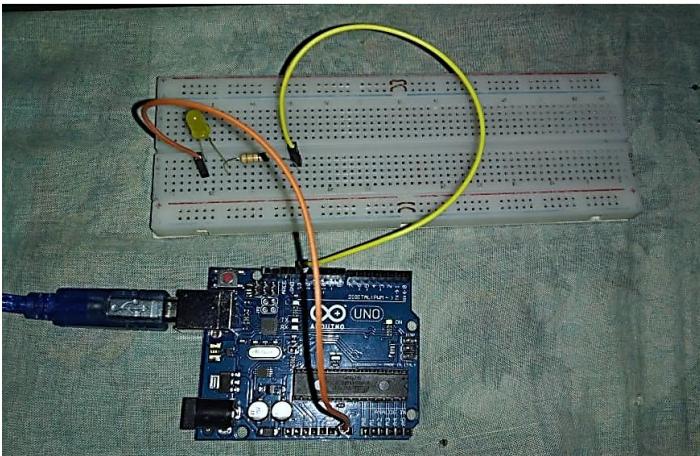


IMAGEN 14 Practica 1 APAGADO

PRACTICA 2: ENCENDER Y APAGAR VARIOS LEDS

OBJETIVOS:

- Conocer sobre las conexiones de los leds
- Mostrar los leds Encendidos y apagados según lo requiera el usuario
- Poner en practica los conocimientos obtenidos en la practica anterior

MATERIALES:

- 5 LED.
- 5 resistencia de 220Ω .
- Una tarjeta Arduino Uno.
- Un cable USB impresora.
- Un computador.
- Cables jumper.
- Protoboard.



DIAGRAMA:

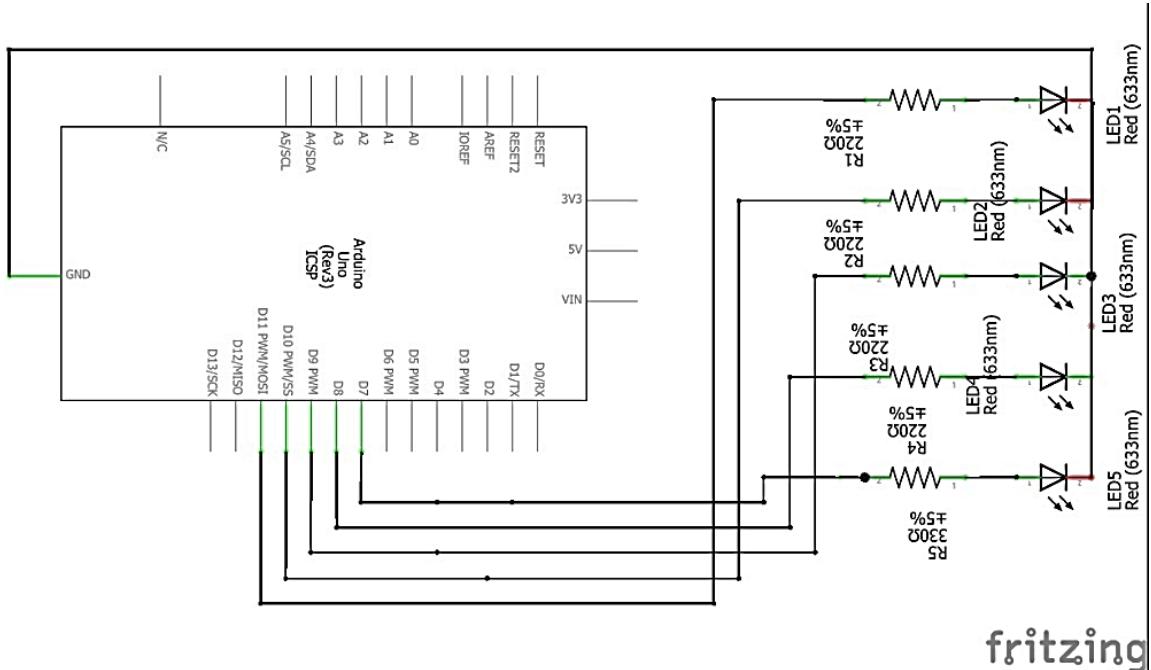
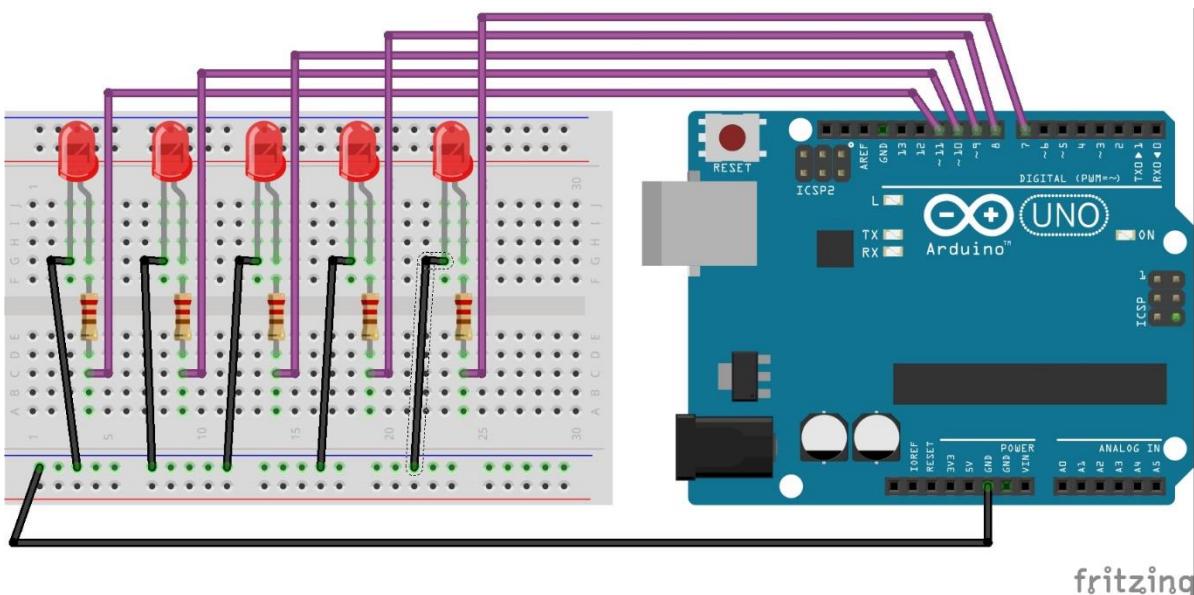


DIAGRAMA 2

ESQUEMA:



ESQUEMA 2

PROGRAMACION:

```
/*
```



* ULEAM-FACCI
*TECNOLOGIA DE LA INFORMACION
* SISTEMAS DIGITALES
*MACIAS PICO JOSSELYN STEFANY
*
*****PRACTICA N° 3 *****
*
* _____ ENCENDIDO DE VARIOS LED'S _____
*
*PARA TAREA COMENTAR TODO EL CODIGO
*/
//declarar los focos

int foco1=7;
int foco2=8;
int foco3=9;
int foco4=10;
int foco5=11;

//declarar el retardo
int retardo = 500;

void setup() {
 //declarar entradas y salidas y variables
 pinMode(foco1,OUTPUT); //cuales seran los pines de salida
 pinMode(foco2,OUTPUT); //cuales seran los pines de salida
 pinMode(foco3,OUTPUT); //cuales seran los pines de salida
 pinMode(foco4,OUTPUT); //cuales seran los pines de salida
 pinMode(foco5,OUTPUT); //cuales seran los pines de salida
}

void loop() {
 //LO QUE VA HACER, VA A LEER DEFINITIVAMENTE TODO... TERMINA Y
 REGRESA
 digitalWrite(foco1, HIGH); //ENCIENDA
 digitalWrite(foco2, HIGH); //ENCIENDA
 digitalWrite(foco3, HIGH); //ENCIENDA



```
    digitalWrite(foco4, HIGH); //ENCIENDA
    digitalWrite(foco5, HIGH); //ENCIENDA
    delay (retardo); //tiempo de retardo
    digitalWrite(foco1, LOW); // APAGADO
    digitalWrite(foco2, LOW); // APAGADO
    digitalWrite(foco3, LOW); // APAGADO
    digitalWrite(foco4, LOW); // APAGADO
    digitalWrite(foco5, LOW); // APAGADO
    delay (retardo); //tiempo de retardo
}
```

RESULTADOS:

Como resultado a esta practica se ha mostrado como encender una secuencia de leds utilizando la tarjeta Arduino el código permite aumentar o disminuir el tiempo en el que enciendan los LEDs y a su vez definir cuantos leds estarán encendidos.

ANEXOS:

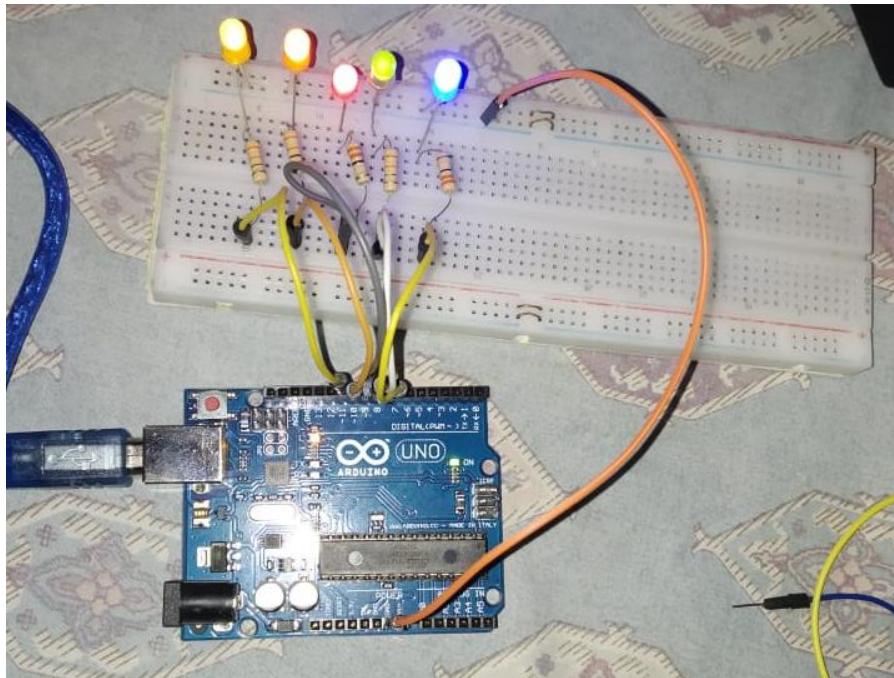


IMAGEN 15 Practica 2 ENCENDIDO

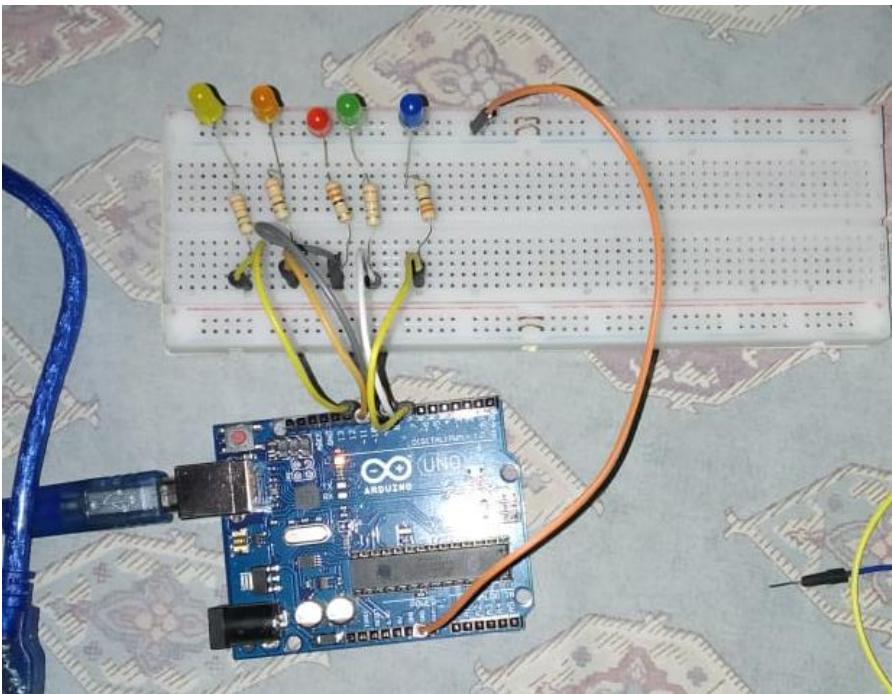


IMAGEN 16 Practica 2 APAGADO

PRACTICA 3: ENCENDER Y APAGAR VARIOS LEDS EN TIEMPOS DIFERENTES

OBJETIVO:

- Mostrar los leds en diferentes tiempos apagados y encendidos en forma de secuencia
- Poner en practica lo conocimientos ya obtenidos durante las enseñanzas

MATERIALES:

- 5 LED.
- 5 resistencia de 220Ω .
- Una tarjeta Arduino Uno.
- Un cable USB impresora.
- Un computador.
- Cables jumper.
- Protoboard.

DIAGRAMA:

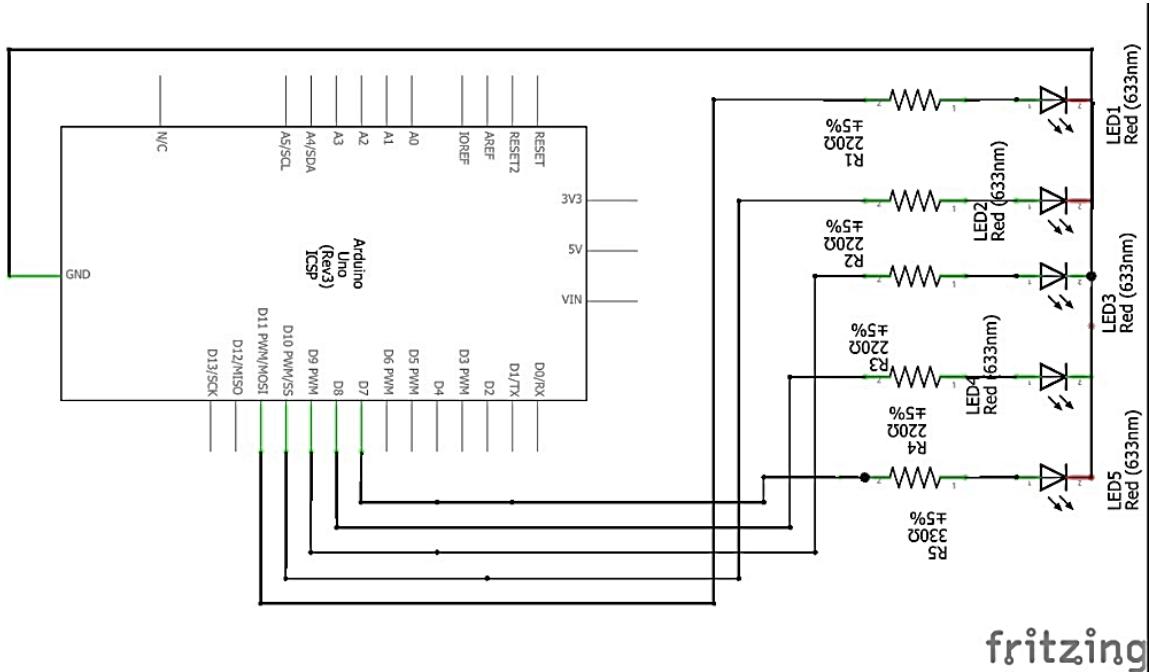
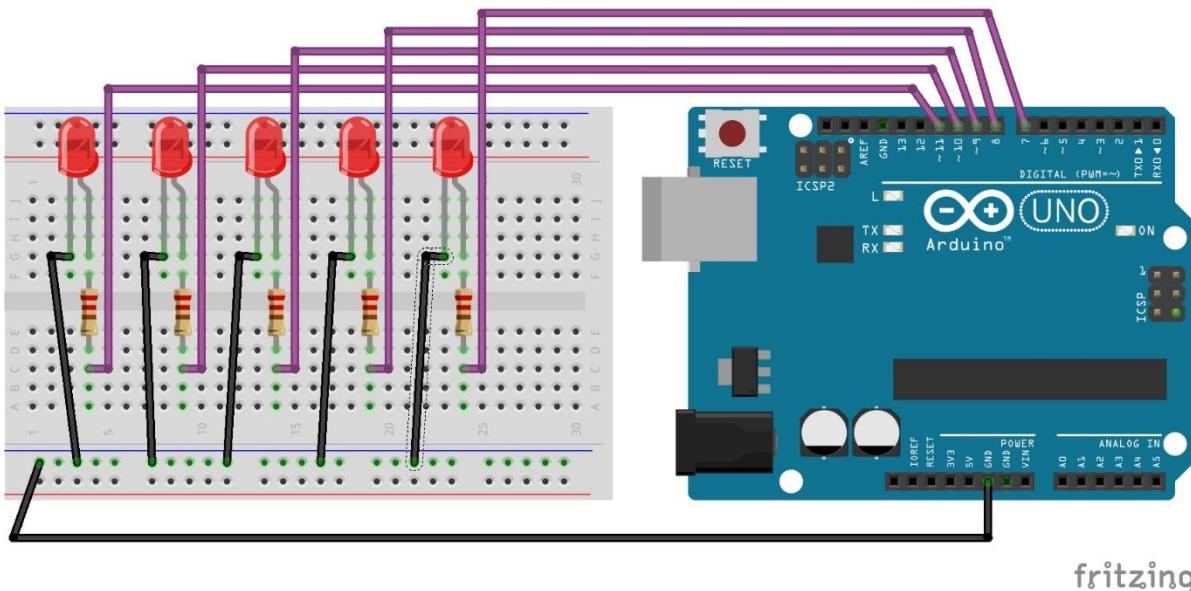


DIAGRAMA 3

ESQUEMA:



ESQUEMA 3

PROGRAMACION:

/*



```
* ULEAM-FACCI
*
* TECNOLOGIA DE LA INFORMACION
*
* SISTEMAS DIGITALES
*
* MACIAS PICO JOSSELYN STEFANY
*
*****
***** TAREA EN CASA *****
*
* _____ENCENDIDO DE VARIOS LED's _____
*
*PARA TAREA COMENTAR TODO EL CODIGO
*
*/
//DECLARAR VARIABLES
//declarar los focos
int foco1=7;
int foco2=8;
int foco3=9;
int foco4=10;
int foco5=11;
//declarar el retardo
int retardo1 = 500;
int retardo2 = 1000;
int retardo3 = 1500;
int retardo4 = 2000;
int retardo5 = 2500;
void setup() {
    //declarar entradas y salidas
    pinMode(foco1,OUTPUT); //cuales seran los pines de salida
    pinMode(foco2,OUTPUT); //cuales seran los pines de salida
    pinMode(foco3,OUTPUT); //cuales seran los pines de salida
    pinMode(foco4,OUTPUT); //cuales seran los pines de salida
    pinMode(foco5,OUTPUT); //cuales seran los pines de salida
}
```



```
void loop() {  
    //LO QUE VA HACER, VA A LEER DEFINITIVAMENTE TODO... TERMINA Y  
    REGRESA
```

ESQUEMA 4

```
//ESTO ES PARA QUE LOS FOCOS SE PRENDAN  
digitalWrite(foco1, HIGH); //ENCIENDA  
delay (retardo1); //tiempo de retardo  
digitalWrite(foco2, HIGH); //ENCIENDA  
delay (retardo2); //tiempo de retardo  
digitalWrite(foco3, HIGH); //ENCIENDA  
delay (retardo3); //tiempo de retardo  
digitalWrite(foco4, HIGH); //ENCIENDA  
delay (retardo4); //tiempo de retardo  
digitalWrite(foco5, HIGH); //ENCIENDA  
delay (retardo5); //tiempo de retardo  
//ESTO ES PARA QUE LOS FOCOS SE APAGUEN  
digitalWrite(foco5, LOW); //APAGADO  
delay (retardo5); //tiempo de retardo  
digitalWrite(foco4, LOW); //APAGADO  
delay (retardo4); //tiempo de retardo  
digitalWrite(foco3, LOW); //APAGADO  
delay (retardo3); //tiempo de retardo  
digitalWrite(foco2, LOW); //APAGADO  
delay (retardo2); //tiempo de retardo  
digitalWrite(foco1, LOW); //APAGADO  
delay (retardo1); //tiempo de retardo
```

```
}
```

RESULTADOS:

En esta practica se ha desarrollado un código que permita encender y apagar una secuencia de LEDS para los cuales se ha utilizado un retardo diferente para cada uno, haciendolo asi de una forma mas fácil y mas rápida. No se obtuvo problema ni en el circuito ni en la parte del código.



ANEXOS:

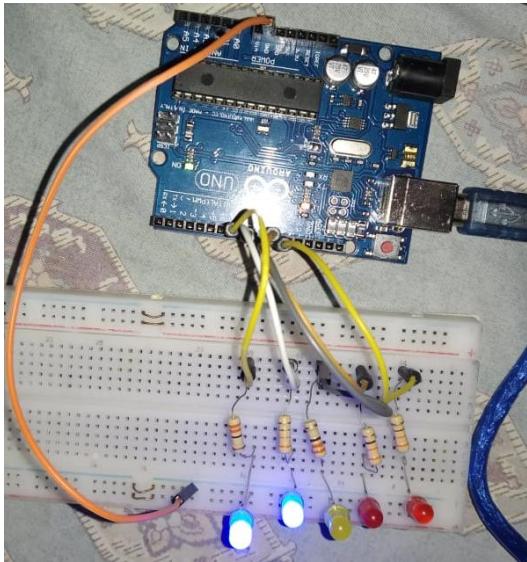


IMAGEN 17ENCENDIDO DE LED

PROYECTO 4: ENCENDER VARIOS LEDS CON FORMA CIRCULAR

OBJETIVOS:

- Poner en practica los conocimientos obtenidos en las practicas anteriores
- Encender un led al comenzar y al terminar, luedo de eso encender dos leds juntos después de cada retardo.

MATERIALES:

- 8 LED.
- 8 resistencia de 330Ω .
- Una tarjeta Arduino Uno.
- Un cable USB impresora.
- Un computador.
- Cables jumper.
- Protoboard.



DIAGRAMA:

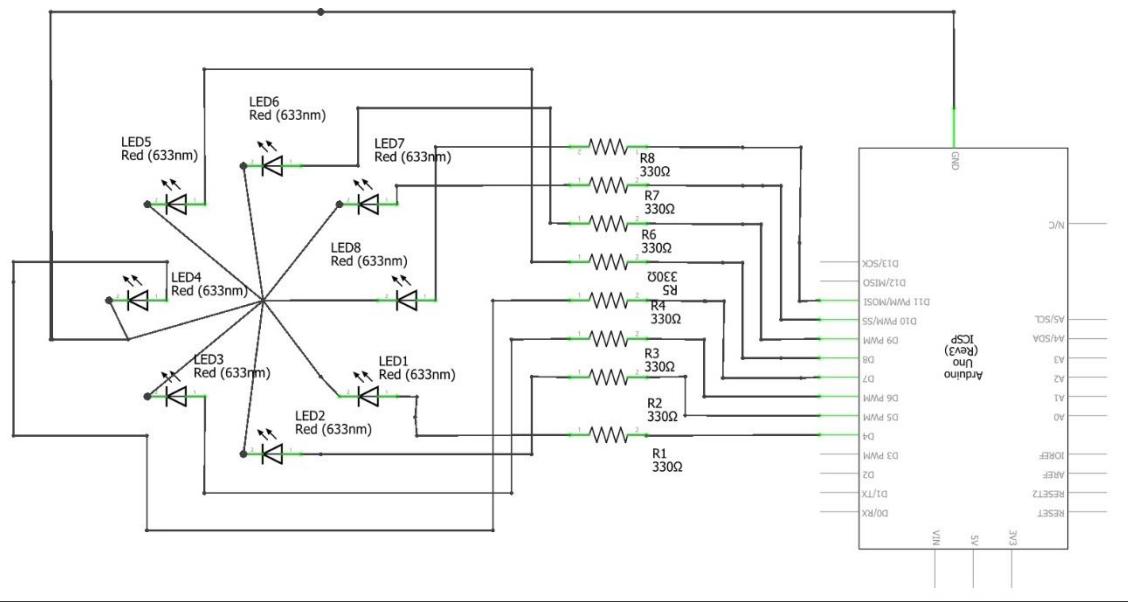
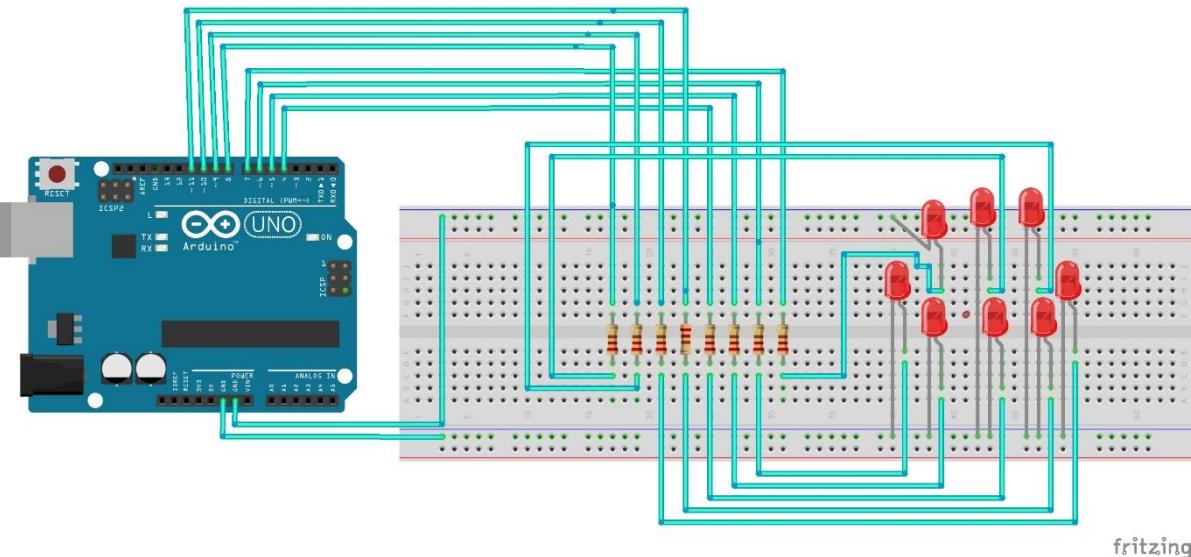


DIAGRAMA 4

ESQUEMA:



ESQUEMA 5

PROGRAMACION:

```
/*
 *          ULEAM-FACCI
 *      TECNOLOGIA DE LA INFORMACION
 *      SISTEMAS DIGITALES
```



```
*      MACIAS PICO JOSSELYN STEFANY
*
*****
***** PRACTICA N° 4 *****
*
* _____ ENCENDIDO CIRCULAR DE VARIOS LED's _____
*/
```

```
//DECLARAR VARIABLES
```

```
//declarar los focos
```

```
int foco1=4;
int foco2=5;
int foco3=6;
int foco4=7;
int foco5=8;
int foco6=9;
int foco7=10;
int foco8=11;
//declarar el retardo
int retardo= 100;
```

```
void setup() {
```

```
//declarar entradas y salidas
```

```
pinMode(foco1,OUTPUT); //cuales seran los pines de salida
pinMode(foco2,OUTPUT); //cuales seran los pines de salida
pinMode(foco3,OUTPUT); //cuales seran los pines de salida
pinMode(foco4,OUTPUT); //cuales seran los pines de salida
pinMode(foco5,OUTPUT); //cuales seran los pines de salida
pinMode(foco6,OUTPUT); //cuales seran los pines de salida
pinMode(foco7,OUTPUT); //cuales seran los pines de salida
pinMode(foco8,OUTPUT); //cuales seran los pines de salida
```



}

```
void loop() {  
  
    //LO QUE VA HACER, VA A LEER DEFINITIVAMENTE TODO... TERMINA Y  
    REGRESA  
  
    //ESTO ES PARA QUE LOS FOCOS SE PRENDAN  
  
    digitalWrite(foco1, LOW); //ENCIENDA  
    digitalWrite(foco2, LOW); //ENCIENDA  
    digitalWrite(foco3, LOW); //ENCIENDA  
    digitalWrite(foco4, LOW); //ENCIENDA  
    digitalWrite(foco5, LOW); //ENCIENDA  
    digitalWrite(foco6, LOW); //ENCIENDA  
    digitalWrite(foco7, LOW); //ENCIENDA  
    digitalWrite(foco8, LOW); //ENCIENDA  
    delay (retardo); //tiempo de retardo  
  
    digitalWrite(foco1, HIGH); //APAGADO  
    digitalWrite(foco2, LOW); //ENCIENDA  
    digitalWrite(foco3, LOW); //ENCIENDA  
    digitalWrite(foco4, LOW); //ENCIENDA  
    digitalWrite(foco5, LOW); //ENCIENDA  
    digitalWrite(foco6, LOW); //ENCIENDA  
    digitalWrite(foco7, LOW); //ENCIENDA  
    digitalWrite(foco8, LOW); //ENCIENDA  
    delay (retardo); //tiempo de retardo  
  
    digitalWrite(foco1, HIGH); //ENCIENDA  
    digitalWrite(foco2, HIGH); //APAGADO  
    digitalWrite(foco3, LOW); //ENCIENDA  
    digitalWrite(foco4, LOW); //ENCIENDA  
    digitalWrite(foco5, LOW); //ENCIENDA
```



```
digitalWrite(foco6, LOW); //ENCIENDA
digitalWrite(foco7, LOW); //ENCIENDA
digitalWrite(foco8, LOW); //ENCIENDA
delay (retardo); //tiempo de retardo

digitalWrite(foco1, LOW); //ENCIENDA
digitalWrite(foco2, HIGH); //ENCIENDA
digitalWrite(foco3, HIGH); //ENCIENDA
digitalWrite(foco4, LOW); //ENCIENDA
digitalWrite(foco5, LOW); //ENCIENDA
digitalWrite(foco6, LOW); //ENCIENDA
digitalWrite(foco7, LOW); //ENCIENDA
digitalWrite(foco8, LOW); //ENCIENDA
delay (retardo); //tiempo de retardo

digitalWrite(foco1, LOW); //ENCIENDA
digitalWrite(foco2, LOW); //ENCIENDA
digitalWrite(foco3, HIGH); //ENCIENDA
digitalWrite(foco4, HIGH); //ENCIENDA
digitalWrite(foco5, LOW); //ENCIENDA
digitalWrite(foco6, LOW); //ENCIENDA
digitalWrite(foco7, LOW); //ENCIENDA
digitalWrite(foco8, LOW); //ENCIENDA
delay (retardo); //tiempo de retardo

digitalWrite(foco1, LOW); //ENCIENDA
digitalWrite(foco2, LOW); //ENCIENDA
digitalWrite(foco3, LOW); //ENCIENDA
digitalWrite(foco4, HIGH); //ENCIENDA
digitalWrite(foco5, HIGH); //ENCIENDA
digitalWrite(foco6, LOW); //ENCIENDA
digitalWrite(foco7, LOW); //ENCIENDA
digitalWrite(foco8, LOW); //ENCIENDA
```



```
delay (retardo); //tiempo de retardo

digitalWrite(foco1, LOW); //ENCIENDA
digitalWrite(foco2, LOW); //ENCIENDA
digitalWrite(foco3, LOW); //ENCIENDA
digitalWrite(foco4, LOW); //ENCIENDA
digitalWrite(foco5, HIGH); //ENCIENDA
digitalWrite(foco6, HIGH); //ENCIENDA
digitalWrite(foco7, LOW); //ENCIENDA
digitalWrite(foco8, LOW); //ENCIENDA
delay (retardo); //tiempo de retardo

digitalWrite(foco1, LOW); //ENCIENDA
digitalWrite(foco2, LOW); //ENCIENDA
digitalWrite(foco3, LOW); //ENCIENDA
digitalWrite(foco4, LOW); //ENCIENDA
digitalWrite(foco5, LOW); //ENCIENDA
digitalWrite(foco6, HIGH); //ENCIENDA
digitalWrite(foco7, HIGH); //ENCIENDA
digitalWrite(foco8, LOW); //ENCIENDA
delay (retardo); //tiempo de retardo

digitalWrite(foco1, LOW); //ENCIENDA
digitalWrite(foco2, LOW); //ENCIENDA
digitalWrite(foco3, LOW); //ENCIENDA
digitalWrite(foco4, LOW); //ENCIENDA
digitalWrite(foco5, LOW); //ENCIENDA
digitalWrite(foco6, LOW); //ENCIENDA
digitalWrite(foco7, HIGH); //ENCIENDA
digitalWrite(foco8, HIGH); //ENCIENDA
delay (retardo); //tiempo de retardo

digitalWrite(foco1, HIGH); //ENCIENDA
```



```
digitalWrite(foco2, LOW); //ENCIENDA
digitalWrite(foco3, LOW); //ENCIENDA
digitalWrite(foco4, LOW); //ENCIENDA
digitalWrite(foco5, LOW); //ENCIENDA
digitalWrite(foco6, LOW); //ENCIENDA
digitalWrite(foco7, LOW); //ENCIENDA
digitalWrite(foco8, HIGH); //ENCIENDA
delay (retardo); //tiempo de retardo

digitalWrite(foco1, LOW); //ENCIENDA
digitalWrite(foco2, LOW); //ENCIENDA
digitalWrite(foco3, LOW); //ENCIENDA
digitalWrite(foco4, LOW); //ENCIENDA
digitalWrite(foco5, LOW); //ENCIENDA
digitalWrite(foco6, LOW); //ENCIENDA
digitalWrite(foco7, LOW); //ENCIENDA
digitalWrite(foco8, LOW); //ENCIENDA
delay (retardo); //tiempo de retardo

digitalWrite(foco1, HIGH); //APAGADO
digitalWrite(foco2, LOW); //ENCIENDA
digitalWrite(foco3, LOW); //ENCIENDA
digitalWrite(foco4, LOW); //ENCIENDA
digitalWrite(foco5, LOW); //ENCIENDA
digitalWrite(foco6, LOW); //ENCIENDA
digitalWrite(foco7, LOW); //ENCIENDA
digitalWrite(foco8, LOW); //ENCIENDA
delay (retardo); //tiempo de retardo

digitalWrite(foco1, HIGH); //ENCIENDA
digitalWrite(foco2, HIGH); //APAGADO
digitalWrite(foco3, LOW); //ENCIENDA
digitalWrite(foco4, LOW); //ENCIENDA
```



```
digitalWrite(foco5, LOW); //ENCIENDA
digitalWrite(foco6, LOW); //ENCIENDA
digitalWrite(foco7, LOW); //ENCIENDA
digitalWrite(foco8, LOW); //ENCIENDA
delay (retardo); //tiempo de retardo
```

```
digitalWrite(foco1, LOW); //ENCIENDA
digitalWrite(foco2, HIGH); //ENCIENDA
digitalWrite(foco3, HIGH); //ENCIENDA
digitalWrite(foco4, LOW); //ENCIENDA
digitalWrite(foco5, LOW); //ENCIENDA
digitalWrite(foco6, LOW); //ENCIENDA
digitalWrite(foco7, LOW); //ENCIENDA
digitalWrite(foco8, LOW); //ENCIENDA
delay (retardo); //tiempo de retardo
```

```
digitalWrite(foco1, LOW); //ENCIENDA
digitalWrite(foco2, LOW); //ENCIENDA
digitalWrite(foco3, HIGH); //ENCIENDA
digitalWrite(foco4, HIGH); //ENCIENDA
digitalWrite(foco5, LOW); //ENCIENDA
digitalWrite(foco6, LOW); //ENCIENDA
digitalWrite(foco7, LOW); //ENCIENDA
digitalWrite(foco8, LOW); //ENCIENDA
delay (retardo); //tiempo de retardo
```

```
digitalWrite(foco1, LOW); //ENCIENDA
digitalWrite(foco2, LOW); //ENCIENDA
digitalWrite(foco3, LOW); //ENCIENDA
digitalWrite(foco4, HIGH); //ENCIENDA
digitalWrite(foco5, HIGH); //ENCIENDA
digitalWrite(foco6, LOW); //ENCIENDA
digitalWrite(foco7, LOW); //ENCIENDA
```



```
digitalWrite(foco8, LOW); //ENCIENDA
delay (retardo); //tiempo de retardo

digitalWrite(foco1, LOW); //ENCIENDA
digitalWrite(foco2, LOW); //ENCIENDA
digitalWrite(foco3, LOW); //ENCIENDA
digitalWrite(foco4, LOW); //ENCIENDA
digitalWrite(foco5, HIGH); //ENCIENDA
digitalWrite(foco6, HIGH); //ENCIENDA
digitalWrite(foco7, LOW); //ENCIENDA
digitalWrite(foco8, LOW); //ENCIENDA
delay (retardo); //tiempo de retardo

digitalWrite(foco1, LOW); //ENCIENDA
digitalWrite(foco2, LOW); //ENCIENDA
digitalWrite(foco3, LOW); //ENCIENDA
digitalWrite(foco4, LOW); //ENCIENDA
digitalWrite(foco5, LOW); //ENCIENDA
digitalWrite(foco6, HIGH); //ENCIENDA
digitalWrite(foco7, HIGH); //ENCIENDA
digitalWrite(foco8, LOW); //ENCIENDA
delay (retardo); //tiempo de retardo

digitalWrite(foco1, LOW); //ENCIENDA
digitalWrite(foco2, LOW); //ENCIENDA
digitalWrite(foco3, LOW); //ENCIENDA
digitalWrite(foco4, LOW); //ENCIENDA
digitalWrite(foco5, LOW); //ENCIENDA
digitalWrite(foco6, LOW); //ENCIENDA
digitalWrite(foco7, HIGH); //ENCIENDA
digitalWrite(foco8, HIGH); //ENCIENDA
delay (retardo); //tiempo de retardo
```



```
digitalWrite(foco1, HIGH); //ENCIENDA
digitalWrite(foco2, LOW); //ENCIENDA
digitalWrite(foco3, LOW); //ENCIENDA
digitalWrite(foco4, LOW); //ENCIENDA
digitalWrite(foco5, LOW); //ENCIENDA
digitalWrite(foco6, LOW); //ENCIENDA
digitalWrite(foco7, LOW); //ENCIENDA
digitalWrite(foco8, HIGH); //ENCIENDA
delay (retardo); //tiempo de retardo

digitalWrite(foco1, LOW); //ENCIENDA
digitalWrite(foco2, LOW); //ENCIENDA
digitalWrite(foco3, LOW); //ENCIENDA
digitalWrite(foco4, LOW); //ENCIENDA
digitalWrite(foco5, LOW); //ENCIENDA
digitalWrite(foco6, LOW); //ENCIENDA
digitalWrite(foco7, LOW); //ENCIENDA
digitalWrite(foco8, LOW); //ENCIENDA
delay (retardo); //tiempo de retardo

//RETROCEDA

digitalWrite(foco1, LOW); //ENCIENDA
digitalWrite(foco2, LOW); //ENCIENDA
digitalWrite(foco3, LOW); //ENCIENDA
digitalWrite(foco4, LOW); //ENCIENDA
digitalWrite(foco5, LOW); //ENCIENDA
digitalWrite(foco6, LOW); //ENCIENDA
digitalWrite(foco7, LOW); //ENCIENDA
digitalWrite(foco8, HIGH); //ENCIENDA
delay (retardo); //tiempo de retardo

digitalWrite(foco1, LOW); //ENCIENDA
```



```
digitalWrite(foco2, LOW); //ENCIENDA
digitalWrite(foco3, LOW); //ENCIENDA
digitalWrite(foco4, LOW); //ENCIENDA
digitalWrite(foco5, LOW); //ENCIENDA
digitalWrite(foco6, LOW); //ENCIENDA
digitalWrite(foco7, HIGH); //ENCIENDA
digitalWrite(foco8, HIGH); //ENCIENDA
delay (retardo); //tiempo de retardo

digitalWrite(foco1, LOW); //ENCIENDA
digitalWrite(foco2, LOW); //ENCIENDA
digitalWrite(foco3, LOW); //ENCIENDA
digitalWrite(foco4, LOW); //ENCIENDA
digitalWrite(foco5, LOW); //ENCIENDA
digitalWrite(foco6, HIGH); //ENCIENDA
digitalWrite(foco7, HIGH); //ENCIENDA
digitalWrite(foco8, LOW); //ENCIENDA
delay (retardo); //tiempo de retardo

digitalWrite(foco1, LOW); //ENCIENDA
digitalWrite(foco2, LOW); //ENCIENDA
digitalWrite(foco3, LOW); //ENCIENDA
digitalWrite(foco4, LOW); //ENCIENDA
digitalWrite(foco5, HIGH); //ENCIENDA
digitalWrite(foco6, HIGH); //ENCIENDA
digitalWrite(foco7, LOW); //ENCIENDA
digitalWrite(foco8, LOW); //ENCIENDA
delay (retardo); //tiempo de retardo

digitalWrite(foco1, LOW); //ENCIENDA
digitalWrite(foco2, LOW); //ENCIENDA
digitalWrite(foco3, LOW); //ENCIENDA
digitalWrite(foco4, HIGH); //ENCIENDA
```



```
digitalWrite(foco5, HIGH); //ENCIENDA
digitalWrite(foco6, LOW); //ENCIENDA
digitalWrite(foco7, LOW); //ENCIENDA
digitalWrite(foco8, LOW); //ENCIENDA
delay (retardo); //tiempo de retardo

digitalWrite(foco1, LOW); //ENCIENDA
digitalWrite(foco2, LOW); //ENCIENDA
digitalWrite(foco3, HIGH); //ENCIENDA
digitalWrite(foco4, HIGH); //ENCIENDA
digitalWrite(foco5, LOW); //ENCIENDA
digitalWrite(foco6, LOW); //ENCIENDA
digitalWrite(foco7, LOW); //ENCIENDA
digitalWrite(foco8, LOW); //ENCIENDA
delay (retardo); //tiempo de retardo

digitalWrite(foco1, LOW); //ENCIENDA
digitalWrite(foco2, HIGH); //ENCIENDA
digitalWrite(foco3, HIGH); //ENCIENDA
digitalWrite(foco4, LOW); //ENCIENDA
digitalWrite(foco5, LOW); //ENCIENDA
digitalWrite(foco6, LOW); //ENCIENDA
digitalWrite(foco7, LOW); //ENCIENDA
digitalWrite(foco8, LOW); //ENCIENDA
delay (retardo); //tiempo de retardo

digitalWrite(foco1, HIGH); //ENCIENDA
digitalWrite(foco2, HIGH); //ENCIENDA
digitalWrite(foco3, LOW); //ENCIENDA
digitalWrite(foco4, LOW); //ENCIENDA
digitalWrite(foco5, LOW); //ENCIENDA
digitalWrite(foco6, LOW); //ENCIENDA
digitalWrite(foco7, LOW); //ENCIENDA
```



```
digitalWrite(foco8, LOW); //ENCIENDA
delay (retardo); //tiempo de retardo

digitalWrite(foco1, HIGH); //ENCIENDA
digitalWrite(foco2, LOW); //ENCIENDA
digitalWrite(foco3, LOW); //ENCIENDA
digitalWrite(foco4, LOW); //ENCIENDA
digitalWrite(foco5, LOW); //ENCIENDA
digitalWrite(foco6, LOW); //ENCIENDA
digitalWrite(foco7, LOW); //ENCIENDA
digitalWrite(foco8, HIGH); //ENCIENDA
delay (retardo); //tiempo de retardo

digitalWrite(foco1, LOW); //ENCIENDA
digitalWrite(foco2, LOW); //ENCIENDA
digitalWrite(foco3, LOW); //ENCIENDA
digitalWrite(foco4, LOW); //ENCIENDA
digitalWrite(foco5, LOW); //ENCIENDA
digitalWrite(foco6, LOW); //ENCIENDA
digitalWrite(foco7, LOW); //ENCIENDA
digitalWrite(foco8, HIGH); //ENCIENDA
delay (retardo); //tiempo de retardo

digitalWrite(foco1, LOW); //ENCIENDA
digitalWrite(foco2, LOW); //ENCIENDA
digitalWrite(foco3, LOW); //ENCIENDA
digitalWrite(foco4, LOW); //ENCIENDA
digitalWrite(foco5, LOW); //ENCIENDA
digitalWrite(foco6, LOW); //ENCIENDA
digitalWrite(foco7, HIGH); //ENCIENDA
digitalWrite(foco8, HIGH); //ENCIENDA
delay (retardo); //tiempo de retardo
```



```
digitalWrite(foco1, LOW); //ENCIENDA
digitalWrite(foco2, LOW); //ENCIENDA
digitalWrite(foco3, LOW); //ENCIENDA
digitalWrite(foco4, LOW); //ENCIENDA
digitalWrite(foco5, LOW); //ENCIENDA
digitalWrite(foco6, HIGH); //ENCIENDA
digitalWrite(foco7, HIGH); //ENCIENDA
digitalWrite(foco8, LOW); //ENCIENDA
delay (retardo); //tiempo de retardo
```

```
digitalWrite(foco1, LOW); //ENCIENDA
digitalWrite(foco2, LOW); //ENCIENDA
digitalWrite(foco3, LOW); //ENCIENDA
digitalWrite(foco4, LOW); //ENCIENDA
digitalWrite(foco5, HIGH); //ENCIENDA
digitalWrite(foco6, HIGH); //ENCIENDA
digitalWrite(foco7, LOW); //ENCIENDA
digitalWrite(foco8, LOW); //ENCIENDA
delay (retardo); //tiempo de retardo
```

```
digitalWrite(foco1, LOW); //ENCIENDA
digitalWrite(foco2, LOW); //ENCIENDA
digitalWrite(foco3, LOW); //ENCIENDA
digitalWrite(foco4, HIGH); //ENCIENDA
digitalWrite(foco5, HIGH); //ENCIENDA
digitalWrite(foco6, LOW); //ENCIENDA
digitalWrite(foco7, LOW); //ENCIENDA
digitalWrite(foco8, LOW); //ENCIENDA
delay (retardo); //tiempo de retardo
```

```
digitalWrite(foco1, LOW); //ENCIENDA
digitalWrite(foco2, LOW); //ENCIENDA
digitalWrite(foco3, HIGH); //ENCIENDA
```



```
digitalWrite(foco4, HIGH); //ENCIENDA
digitalWrite(foco5, LOW); //ENCIENDA
digitalWrite(foco6, LOW); //ENCIENDA
digitalWrite(foco7, LOW); //ENCIENDA
digitalWrite(foco8, LOW); //ENCIENDA
delay (retardo); //tiempo de retardo

digitalWrite(foco1, LOW); //ENCIENDA
digitalWrite(foco2, HIGH); //ENCIENDA
digitalWrite(foco3, HIGH); //ENCIENDA
digitalWrite(foco4, LOW); //ENCIENDA
digitalWrite(foco5, LOW); //ENCIENDA
digitalWrite(foco6, LOW); //ENCIENDA
digitalWrite(foco7, LOW); //ENCIENDA
digitalWrite(foco8, LOW); //ENCIENDA
delay (retardo); //tiempo de retardo

digitalWrite(foco1, HIGH); //ENCIENDA
digitalWrite(foco2, HIGH); //ENCIENDA
digitalWrite(foco3, LOW); //ENCIENDA
digitalWrite(foco4, LOW); //ENCIENDA
digitalWrite(foco5, LOW); //ENCIENDA
digitalWrite(foco6, LOW); //ENCIENDA
digitalWrite(foco7, LOW); //ENCIENDA
digitalWrite(foco8, LOW); //ENCIENDA
delay (retardo); //tiempo de retardo

digitalWrite(foco1, HIGH); //ENCIENDA
digitalWrite(foco2, LOW); //ENCIENDA
digitalWrite(foco3, LOW); //ENCIENDA
digitalWrite(foco4, LOW); //ENCIENDA
digitalWrite(foco5, LOW); //ENCIENDA
digitalWrite(foco6, LOW); //ENCIENDA
```



```
digitalWrite(foco7, LOW); //ENCIENDA
digitalWrite(foco8, HIGH); //ENCIENDA
delay (retardo); //tiempo de retardo

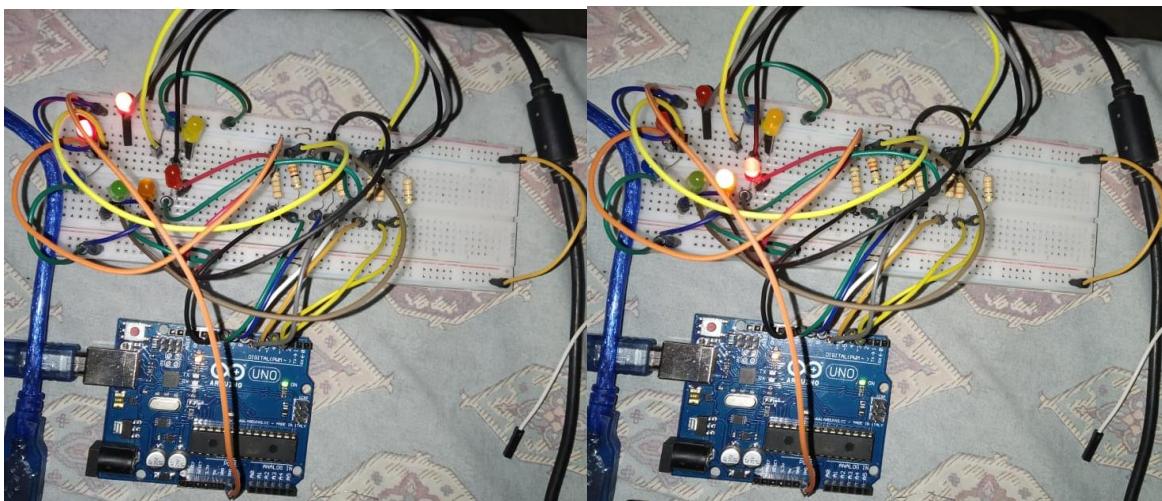
digitalWrite(foco1, LOW); //ENCIENDA
digitalWrite(foco2, LOW); //ENCIENDA
digitalWrite(foco3, LOW); //ENCIENDA
digitalWrite(foco4, LOW); //ENCIENDA
digitalWrite(foco5, LOW); //ENCIENDA
digitalWrite(foco6, LOW); //ENCIENDA
digitalWrite(foco7, LOW); //ENCIENDA
digitalWrite(foco8, LOW); //ENCIENDA
delay (retardo); //tiempo de retardo

}
```

RESULTADOS:

En esta practica se realizaron varias líneas de código por cada cambio de animacion en cada cambio deben prender dos leds y al finalizar se apagan todos y se vuelve hacer lo mismo en sentido contrario. No se obtuvieron problemas ni con el circuito, ni con el código. El retardo s epuede modificar para que sea mas rápido o mas lento al encender los siguientes LEDs

ANEXOS:





PROYECTO 5: PRENDER Y APAGAR UN LED MEDIANTE UN PULSADOR

OBJETIVOS:

- Aprender a usar un pulsador para encender y apagar un led
- Verificar conexiones
- Poner en practica conocimientos obtenidos

MATERIALES:

- Arduino Uno
- Cable USB de arduino
- Protoboard
- 1 led
- 2 resistencia de 330 ohms
- 1 pulsador
- Cables jumper

DIAGRAMA:

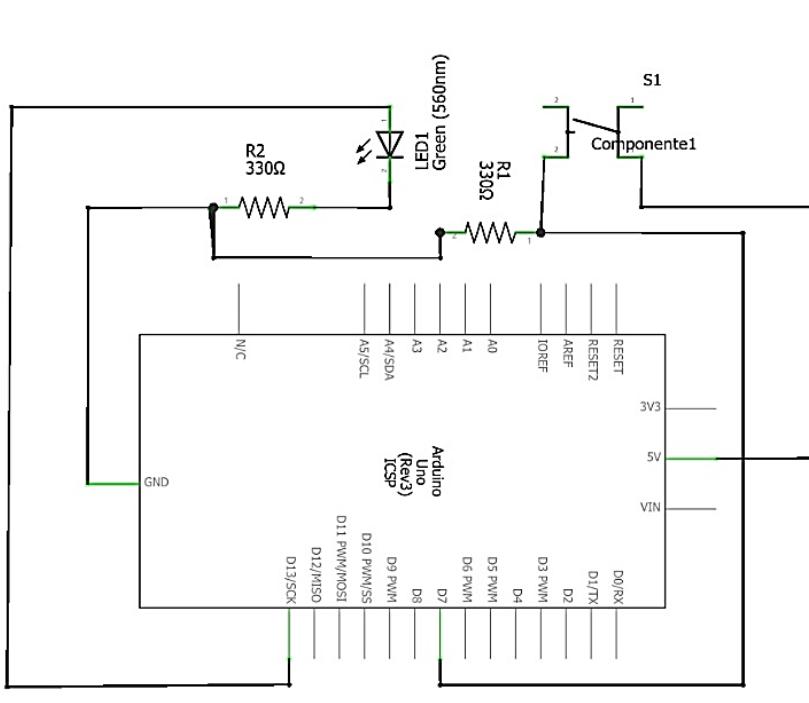
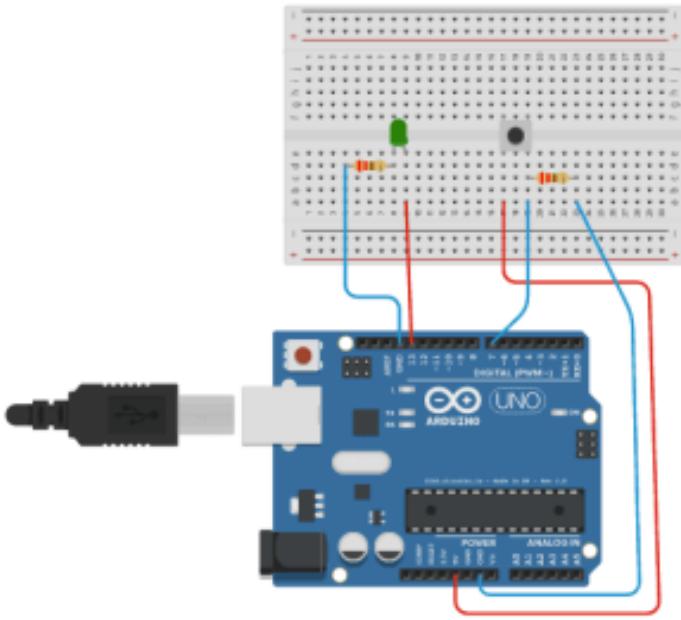


DIAGRAMA 5



ESQUEMA:



ESQUEMA 6

PROGRAMACION:

```
/*
 *          ULEAM-FACCI
 *      TECNOLOGIA DE LA INFORMACION
 *      SISTEMAS DIGITALES
 *      MACIAS PICO JOSSELYN STEFANY
 *
 ***** PRACTICA N° 6 *****
 */
//DECLARAR VARIABLES
//declarar los focos
int pulsador=7;
int led=13;
int pulso=0;;
int retardo=200;
void setup() {
    //declarar entradas y salidas
```



```
pinMode(led,OUTPUT); //cuales seran los pines de salida
pinMode(pulsador,OUTPUT); //cuales seran los pines de salida
}

void parpadeo(){
    digitalWrite(led, HIGH);
    delay(retardo);
    digitalWrite(led, LOW);
    delay(retardo);
}

void loop() {
    pulso= digitalRead( pulsador);
    if (pulso==HIGH) {
        parpadeo();
    }else{
        digitalWrite(led, LOW);
        delay(50);
    }
}
```

RESULTADOS:

En esta practica se utilizo un nuevo componente que es llamado pulsador para el cual se utilizo un if en donde se realiza un parpadeo solo si se utiliza el pulsador caso contrario solo queda encendido. Además este pulsador también trabaja con una resistencia de 330 ohmios conectada en la parte de donde sale el pin negativo

PROYECTO 6: MEDIANTE UN PULSADOR REALIZAR CAMBIOS DE ANIMACION

OBJETIVOS:

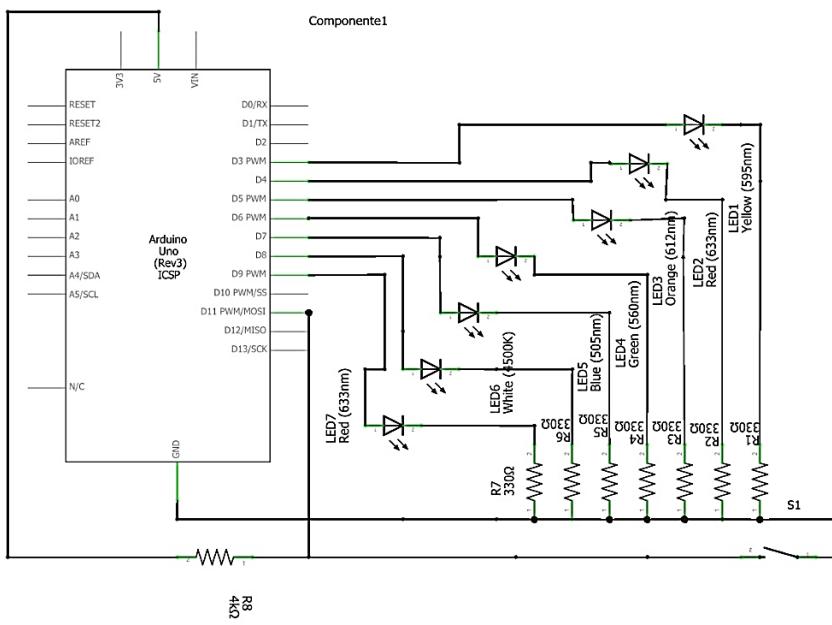
- Demostrar los conocimientos obtenidos en clase
- Realizar métodos para cada animación y así podría ser más factible trabajar
- Con ayuda del pulsador, realice cada animación en el preciso momento



MATERIALES:

- 7 resistencia de $330\ \Omega$
- Un arduino uno
- Un protoboard
- Cable para conectar arduino
- Varios cables jumper
- 7 leds
- 1 pulsador
- 1 resistencia de $4\ k\Omega$

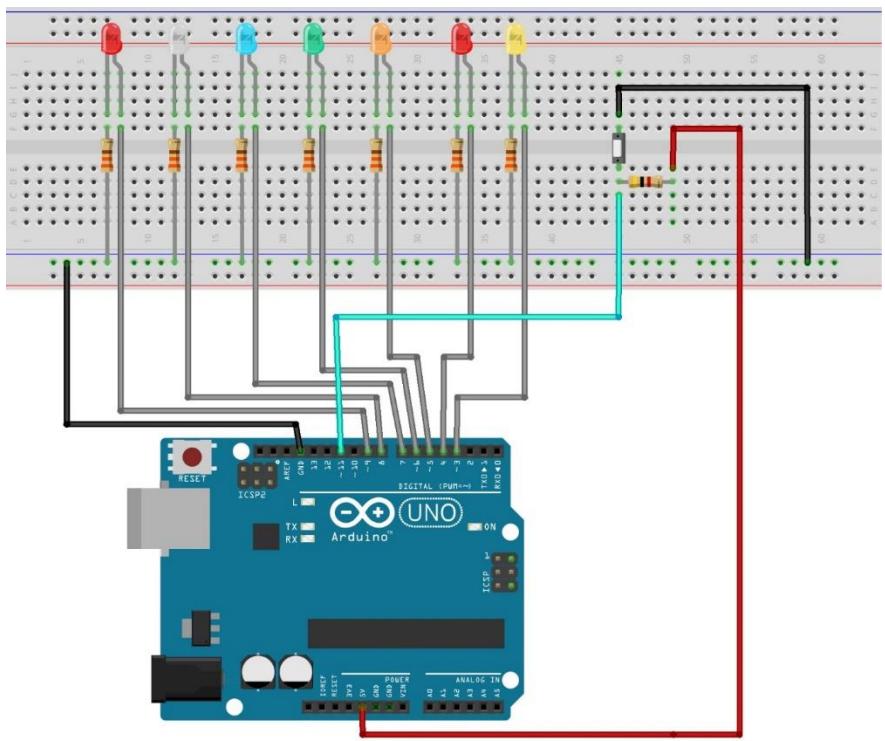
DIAGRAMA:



fritzing

DIAGRAMA 6

ESQUEMA:



ESQUEMA 7

PROGRAMACION:

```
/*
*
*
*          ULEAM-FACCI
*
*      TECNOLOGIA DE LA INFORMACION
*
*      SISTEMAS DIGITALES
*
*      MACIAS PICO JOSSELYN STEFANY
*
*****
*      TAREA N° 2 *****
*
*      _____ ANIMACIONES CON 7 LED _____
*
*PARA TAREA COMENTAR TODO EL CODIGO
*
*/

```



```
int led1 =3;
int led2 =4;
int led3 =5;
int led4 =6;
int led5 =7;
int led6 =8;
int led7 =9;
int pulsador = 11;
int retardo =500;
int pulso=0;

void setup() {
    pinMode(led1,OUTPUT); //cuales seran los pines de salida
    pinMode(led2,OUTPUT); //cuales seran los pines de salida
    pinMode(led3,OUTPUT); //cuales seran los pines de salida
    pinMode(led4,OUTPUT); //cuales seran los pines de salida
    pinMode(led5,OUTPUT); //cuales seran los pines de salida
    pinMode(led6,OUTPUT); //cuales seran los pines de salida
    pinMode(led7,OUTPUT); //cuales seran los pines de salida
    pinMode(pulsador,OUTPUT); //cuales seran los pines de salida
}

void animacion1(){
    //PRIMERA PULSACION
    digitalWrite(led1, HIGH); //ENCIENDA
    digitalWrite(led2, LOW); //ENCIENDA
    digitalWrite(led3, LOW); //ENCIENDA
    digitalWrite (led4, LOW); //ENCIENDA
    digitalWrite (led5, LOW); //ENCIENDA
    digitalWrite (led6, LOW); //ENCIENDA
    digitalWrite (led7, HIGH); //ENCIENDA
    delay (retardo); //tiempo de retardo
    digitalWrite (led1, LOW); //ENCIENDA
    digitalWrite (led2, HIGH); //ENCIENDA
    digitalWrite (led3, LOW); //ENCIENDA
```



```
digitalWrite (led4, LOW); //ENCIENDA
digitalWrite (led5, LOW); //ENCIENDA
digitalWrite (led6, HIGH); //ENCIENDA
digitalWrite (led7, LOW); //ENCIENDA
delay (retardo); //tiempo de retardo
digitalWrite (led1, LOW); //ENCIENDA
digitalWrite (led2, LOW); //ENCIENDA
digitalWrite (led3, HIGH); //ENCIENDA
digitalWrite (led4, LOW); //ENCIENDA
digitalWrite (led5, HIGH); //ENCIENDA
digitalWrite (led6, LOW); //ENCIENDA
digitalWrite (led7, LOW); //ENCIENDA
delay (retardo); //tiempo de retardo
digitalWrite (led1, LOW); //ENCIENDA
digitalWrite (led2, LOW); //ENCIENDA
digitalWrite (led3, LOW); //ENCIENDA
digitalWrite (led4, HIGH); //ENCIENDA
digitalWrite (led5, LOW); //ENCIENDA
digitalWrite (led6, LOW); //ENCIENDA
digitalWrite (led7, LOW); //ENCIENDA
delay (retardo); //tiempo de retardo
digitalWrite (led1, LOW); //ENCIENDA
digitalWrite (led2, LOW); //ENCIENDA
digitalWrite (led3, LOW); //ENCIENDA
digitalWrite (led4, LOW); //ENCIENDA
digitalWrite (led5, LOW); //ENCIENDA
digitalWrite (led6, LOW); //ENCIENDA
digitalWrite (led7, LOW); //ENCIENDA
delay (retardo); //tiempo de retardo
digitalWrite (led1, LOW); //ENCIENDA
digitalWrite (led2, LOW); //ENCIENDA
digitalWrite (led3, LOW); //ENCIENDA
digitalWrite (led4, HIGH); //ENCIENDA
```



```
digitalWrite(led5, LOW); //ENCIENDA
digitalWrite(led6, LOW); //ENCIENDA
digitalWrite(led7, LOW); //ENCIENDA
delay(retardo); //tiempo de retardo
digitalWrite(led1, LOW); //ENCIENDA
digitalWrite(led2, LOW); //ENCIENDA
digitalWrite(led3, HIGH); //ENCIENDA
digitalWrite(led4, LOW); //ENCIENDA
digitalWrite(led5, HIGH); //ENCIENDA
digitalWrite(led6, LOW); //ENCIENDA
digitalWrite(led7, LOW); //ENCIENDA
delay(retardo); //tiempo de retardo
digitalWrite(led1, LOW); //ENCIENDA
digitalWrite(led2, HIGH); //ENCIENDA
digitalWrite(led3, LOW); //ENCIENDA
digitalWrite(led4, LOW); //ENCIENDA
digitalWrite(led5, LOW); //ENCIENDA
digitalWrite(led6, HIGH); //ENCIENDA
digitalWrite(led7, LOW); //ENCIENDA
delay(retardo); //tiempo de retardo
digitalWrite(led1, HIGH); //ENCIENDA
digitalWrite(led2, LOW); //ENCIENDA
digitalWrite(led3, LOW); //ENCIENDA
digitalWrite(led4, LOW); //ENCIENDA
digitalWrite(led5, LOW); //ENCIENDA
digitalWrite(led6, LOW); //ENCIENDA
digitalWrite(led7, HIGH); //ENCIENDA
delay(retardo); //tiempo de retardo
digitalWrite(led1, LOW); //ENCIENDA
digitalWrite(led2, LOW); //ENCIENDA
digitalWrite(led3, LOW); //ENCIENDA
digitalWrite(led4, LOW); //ENCIENDA
digitalWrite(led5, LOW); //ENCIENDA
```



```
digitalWrite(led6, LOW); //ENCIENDA
digitalWrite(led7, LOW); //ENCIENDA
delay (retardo); //tiempo de retardo
}

void animacion2(){
    //SEGUNDA PULSACION
    digitalWrite(led1, HIGH); //ENCIENDA
    digitalWrite(led2, LOW); //ENCIENDA
    digitalWrite(led3, LOW); //ENCIENDA
    digitalWrite(led4, LOW); //ENCIENDA
    digitalWrite(led5, LOW); //ENCIENDA
    digitalWrite(led6, LOW); //ENCIENDA
    digitalWrite(led7, HIGH); //ENCIENDA
    delay (retardo); //tiempo de retardo
    digitalWrite(led1, HIGH); //ENCIENDA
    digitalWrite(led2, HIGH); //ENCIENDA
    digitalWrite(led3, LOW); //ENCIENDA
    digitalWrite(led4, LOW); //ENCIENDA
    digitalWrite(led5, LOW); //ENCIENDA
    digitalWrite(led6, HIGH); //ENCIENDA
    digitalWrite(led7, HIGH); //ENCIENDA
    delay (retardo); //tiempo de retardo
    digitalWrite(led1, HIGH); //ENCIENDA
    digitalWrite(led2, HIGH); //ENCIENDA
    digitalWrite(led3, HIGH); //ENCIENDA
    digitalWrite(led4, LOW); //ENCIENDA
    digitalWrite(led5, HIGH); //ENCIENDA
    digitalWrite(led6, HIGH); //ENCIENDA
    digitalWrite(led7, HIGH); //ENCIENDA
    delay (retardo); //tiempo de retardo
    digitalWrite(led1, HIGH); //ENCIENDA
    digitalWrite(led2, HIGH); //ENCIENDA
    digitalWrite(led3, HIGH); //ENCIENDA
```



```
digitalWrite(led4, HIGH); //ENCIENDA
digitalWrite(led5, HIGH); //ENCIENDA
digitalWrite(led6, HIGH); //ENCIENDA
digitalWrite(led7, HIGH); //ENCIENDA
delay (retardo); //tiempo de retardo
digitalWrite(led1, LOW); //ENCIENDA
digitalWrite(led2, LOW); //ENCIENDA
digitalWrite(led3, LOW); //ENCIENDA
digitalWrite(led4, HIGH); //ENCIENDA
digitalWrite(led5, LOW); //ENCIENDA
digitalWrite(led6, LOW); //ENCIENDA
digitalWrite(led7, LOW); //ENCIENDA
delay (retardo); //tiempo de retardo
digitalWrite(led1, LOW); //ENCIENDA
digitalWrite(led2, LOW); //ENCIENDA
digitalWrite(led3, HIGH); //ENCIENDA
digitalWrite(led4, HIGH); //ENCIENDA
digitalWrite(led5, HIGH); //ENCIENDA
digitalWrite(led6, LOW); //ENCIENDA
digitalWrite(led7, LOW); //ENCIENDA
delay (retardo); //tiempo de retard
digitalWrite(led1, LOW); //ENCIENDA
digitalWrite(led2, HIGH); //ENCIENDA
digitalWrite(led3, HIGH); //ENCIENDA
digitalWrite(led4, HIGH); //ENCIENDA
digitalWrite(led5, HIGH); //ENCIENDA
digitalWrite(led6, HIGH); //ENCIENDA
digitalWrite(led7, LOW); //ENCIENDA
delay (retardo); //tiempo de retard
digitalWrite(led1, HIGH); //ENCIENDA
digitalWrite(led2, HIGH); //ENCIENDA
digitalWrite(led3, HIGH); //ENCIENDA
digitalWrite(led4, HIGH); //ENCIENDA
```



```
digitalWrite(led5, HIGH); //ENCIENDA
digitalWrite(led6, HIGH); //ENCIENDA
digitalWrite(led7, HIGH); //ENCIENDA
delay (retardo); //tiempo de retardo
}

void animacion3(){
    digitalWrite(led1, HIGH); //ENCIENDA
    digitalWrite(led1, LOW); //ENCIENDA
    delay(retardo);
    digitalWrite(led2, HIGH); //ENCIENDA
    digitalWrite(led2, LOW); //ENCIENDA
    delay(retardo);
    digitalWrite(led3, HIGH); //ENCIENDA
    digitalWrite(led3, LOW); //ENCIENDA
    delay(retardo);
    digitalWrite(led4, HIGH); //ENCIENDA
    digitalWrite(led4, LOW); //ENCIENDA
    delay(retardo);
    digitalWrite(led5, HIGH); //ENCIENDA
    digitalWrite(led5, LOW); //ENCIENDA
    delay(retardo);
    digitalWrite(led6, HIGH); //ENCIENDA
    digitalWrite(led6, LOW); //ENCIENDA
    delay(retardo);
    digitalWrite(led7, HIGH); //ENCIENDA
    digitalWrite(led7, LOW); //ENCIENDA
    delay (retardo); //tiempo de retardo
    digitalWrite(led7, LOW); //ENCIENDA
    digitalWrite(led7, HIGH); //ENCIENDA
    delay(retardo);
    digitalWrite(led6, LOW); //ENCIENDA
    digitalWrite(led6, HIGH); //ENCIENDA
    delay(retardo);
```



```
digitalWrite(led5, LOW); //ENCIENDA
digitalWrite(led5, HIGH); //ENCIENDA
delay(retardo);
digitalWrite(led4, LOW); //ENCIENDA
digitalWrite(led4, HIGH); //ENCIENDA
delay(retardo);
digitalWrite(led3, LOW); //ENCIENDA
digitalWrite(led3, HIGH); //ENCIENDA
delay(retardo);
digitalWrite(led2, LOW); //ENCIENDA
digitalWrite(led2, HIGH); //ENCIENDA
delay(retardo);
digitalWrite(led1, LOW); //ENCIENDA
digitalWrite(led1, HIGH); //ENCIENDA
delay (retardo); //tiempo de retardo
}
void animacion4(){
//CUARTA PULSACION
digitalWrite(led1, LOW); //ENCIENDA
digitalWrite(led2, LOW); //ENCIENDA
digitalWrite(led3, LOW); //ENCIENDA
digitalWrite(led4, LOW); //ENCIENDA
digitalWrite(led5, LOW); //ENCIENDA
digitalWrite(led6, LOW); //ENCIENDA
digitalWrite(led7, LOW); //ENCIENDA
delay (retardo); //tiempo de retardo
digitalWrite(led1, LOW); //ENCIENDA
digitalWrite(led2, LOW); //ENCIENDA
digitalWrite(led3, LOW); //ENCIENDA
digitalWrite(led4, HIGH); //ENCIENDA
digitalWrite(led5, HIGH); //ENCIENDA
digitalWrite(led6, HIGH); //ENCIENDA
digitalWrite(led7, LOW); //ENCIENDA
```



```
delay (retardo); //tiempo de retardo
digitalWrite(led1, LOW); //ENCIENDA
digitalWrite(led2, HIGH); //ENCIENDA
digitalWrite(led3, HIGH); //ENCIENDA
digitalWrite(led4, HIGH); //ENCIENDA
digitalWrite(led5, LOW); //ENCIENDA
digitalWrite(led6, LOW); //ENCIENDA
digitalWrite(led7, LOW); //ENCIENDA
delay (retardo); //tiempo de retardo
digitalWrite(led1, HIGH); //ENCIENDA
digitalWrite(led2, HIGH); //ENCIENDA
digitalWrite(led3, LOW); //ENCIENDA
digitalWrite(led4, LOW); //ENCIENDA
digitalWrite(led5, LOW); //ENCIENDA
digitalWrite(led6, HIGH); //ENCIENDA
digitalWrite(led7, HIGH); //ENCIENDA
delay (retardo); //tiempo de retardo
digitalWrite(led1, LOW); //ENCIENDA
digitalWrite(led2, LOW); //ENCIENDA
digitalWrite(led3, LOW); //ENCIENDA
digitalWrite(led4, HIGH); //ENCIENDA
digitalWrite(led5, HIGH); //ENCIENDA
digitalWrite(led6, HIGH); //ENCIENDA
digitalWrite(led7, LOW); //ENCIENDA
delay (retardo); //tiempo de retardo
digitalWrite(led1, HIGH); //ENCIENDA
digitalWrite(led2, HIGH); //ENCIENDA
digitalWrite(led3, LOW); //ENCIENDA
digitalWrite(led4, LOW); //ENCIENDA
digitalWrite(led5, LOW); //ENCIENDA
digitalWrite(led6, HIGH); //ENCIENDA
digitalWrite(led7, HIGH); //ENCIENDA
delay (retardo); //tiempo de retardo
```



```
digitalWrite(led1, LOW); //ENCIENDA
digitalWrite(led2, LOW); //ENCIENDA
digitalWrite(led3, LOW); //ENCIENDA
digitalWrite(led4, HIGH); //ENCIENDA
digitalWrite(led5, HIGH); //ENCIENDA
digitalWrite(led6, HIGH); //ENCIENDA
digitalWrite(led7, LOW); //ENCIENDA
delay (retardo); //tiempo de retardo
digitalWrite(led1, LOW); //ENCIENDA
digitalWrite(led2, HIGH); //ENCIENDA
digitalWrite(led3, HIGH); //ENCIENDA
digitalWrite(led4, HIGH); //ENCIENDA
digitalWrite(led5, LOW); //ENCIENDA
digitalWrite(led6, LOW); //ENCIENDA
digitalWrite(led7, LOW); //ENCIENDA
delay (retardo); //tiempo de retardo
digitalWrite(led1, HIGH); //ENCIENDA
digitalWrite(led2, HIGH); //ENCIENDA
digitalWrite(led3, LOW); //ENCIENDA
digitalWrite(led4, LOW); //ENCIENDA
digitalWrite(led5, LOW); //ENCIENDA
digitalWrite(led6, HIGH); //ENCIENDA
digitalWrite(led7, HIGH); //ENCIENDA
delay (retardo); //tiempo de retardo
digitalWrite(led1, LOW); //ENCIENDA
digitalWrite(led2, LOW); //ENCIENDA
digitalWrite(led3, LOW); //ENCIENDA
digitalWrite(led4, LOW); //ENCIENDA
digitalWrite(led5, LOW); //ENCIENDA
digitalWrite(led6, LOW); //ENCIENDA
digitalWrite(led7, LOW); //ENCIENDA
delay (retardo); //tiempo de retardo
}
```



```
void animacion5(){

    digitalWrite(led1, LOW); //ENCIENDA
    digitalWrite(led2, LOW); //ENCIENDA
    digitalWrite(led3, HIGH); //ENCIENDA
    digitalWrite(led4, LOW); //ENCIENDA
    digitalWrite(led5, HIGH); //ENCIENDA
    digitalWrite(led6, LOW); //ENCIENDA
    digitalWrite(led7, LOW); //ENCIENDA
    delay (retardo); //tiempo de retardo
    digitalWrite(led1, HIGH); //ENCIENDA
    digitalWrite(led2, LOW); //ENCIENDA
    digitalWrite(led3, HIGH); //ENCIENDA
    digitalWrite(led4, LOW); //ENCIENDA
    digitalWrite(led5, HIGH); //ENCIENDA
    digitalWrite(led6, LOW); //ENCIENDA
    digitalWrite(led7, HIGH); //ENCIENDA
    delay (retardo); //tiempo de retardo
    digitalWrite(led1, HIGH); //ENCIENDA
    digitalWrite(led2, LOW); //ENCIENDA
    digitalWrite(led3, HIGH); //ENCIENDA
    digitalWrite(led4, HIGH); //ENCIENDA
    digitalWrite(led5, HIGH); //ENCIENDA
    digitalWrite(led6, LOW); //ENCIENDA
    digitalWrite(led7, HIGH); //ENCIENDA
    delay (retardo); //tiempo de retardo
    digitalWrite(led1, HIGH); //ENCIENDA
    digitalWrite(led2, HIGH); //ENCIENDA
    digitalWrite(led3, HIGH); //ENCIENDA
    digitalWrite(led4, HIGH); //ENCIENDA
    digitalWrite(led5, HIGH); //ENCIENDA
    digitalWrite(led6, HIGH); //ENCIENDA
    digitalWrite(led7, HIGH); //ENCIENDA
    delay (retardo); //tiempo de retardo
```



```
digitalWrite(led1, LOW); //ENCIENDA
digitalWrite(led2, LOW); //ENCIENDA
digitalWrite(led3, LOW); //ENCIENDA
digitalWrite(led4, LOW); //ENCIENDA
digitalWrite(led5, LOW); //ENCIENDA
digitalWrite(led6, LOW); //ENCIENDA
digitalWrite(led7, LOW); //ENCIENDA
delay (retardo); //tiempo de retardo
}

void animacion6(){
    digitalWrite(led1, LOW); //ENCIENDA
    digitalWrite(led2, HIGH); //ENCIENDA
    digitalWrite(led3, LOW); //ENCIENDA
    digitalWrite(led4, LOW); //ENCIENDA
    digitalWrite(led5, LOW); //ENCIENDA
    digitalWrite(led6, LOW); //ENCIENDA
    digitalWrite(led7, LOW); //ENCIENDA
    delay (retardo); //tiempo de retardo
    digitalWrite(led1, HIGH); //ENCIENDA
    digitalWrite(led2, HIGH); //ENCIENDA
    digitalWrite(led3, HIGH); //ENCIENDA
    digitalWrite(led4, LOW); //ENCIENDA
    digitalWrite(led5, LOW); //ENCIENDA
    digitalWrite(led6, LOW); //ENCIENDA
    digitalWrite(led7, LOW); //ENCIENDA
    delay (retardo); //tiempo de retardo
    digitalWrite(led1, HIGH); //ENCIENDA
    digitalWrite(led2, HIGH); //ENCIENDA
    digitalWrite(led3, HIGH); //ENCIENDA
    digitalWrite(led4, LOW); //ENCIENDA
    digitalWrite(led5, LOW); //ENCIENDA
    digitalWrite(led6, LOW); //ENCIENDA
    digitalWrite(led7, LOW); //ENCIENDA
}
```



```
delay (retardo); //tiempo de retardo
digitalWrite(led1, HIGH); //ENCIENDA
digitalWrite(led2, HIGH); //ENCIENDA
digitalWrite(led3, HIGH); //ENCIENDA
digitalWrite(led4, HIGH); //ENCIENDA
digitalWrite(led5, LOW); //ENCIENDA
digitalWrite(led6, LOW); //ENCIENDA
digitalWrite(led7, HIGH); //ENCIENDA
delay (retardo); //tiempo de retardo
digitalWrite(led1, HIGH); //ENCIENDA
digitalWrite(led2, HIGH); //ENCIENDA
digitalWrite(led3, HIGH); //ENCIENDA
digitalWrite(led4, HIGH); //ENCIENDA
digitalWrite(led5, HIGH); //ENCIENDA
digitalWrite(led6, HIGH); //ENCIENDA
digitalWrite(led7, HIGH); //ENCIENDA
delay (retardo); //tiempo de retardo
digitalWrite(led1, LOW); //ENCIENDA
digitalWrite(led2, LOW); //ENCIENDA
digitalWrite(led3, LOW); //ENCIENDA
digitalWrite(led4, LOW); //ENCIENDA
digitalWrite(led5, HIGH); //ENCIENDA
digitalWrite(led6, HIGH); //ENCIENDA
digitalWrite(led7, LOW); //ENCIENDA
delay (retardo); //tiempo de retardo
digitalWrite(led1, LOW); //ENCIENDA
digitalWrite(led2, LOW); //ENCIENDA
digitalWrite(led3, LOW); //ENCIENDA
digitalWrite(led4, HIGH); //ENCIENDA
digitalWrite(led5, LOW); //ENCIENDA
digitalWrite(led6, LOW); //ENCIENDA
digitalWrite(led7, HIGH); //ENCIENDA
delay (retardo); //tiempo de retardo
```



```
digitalWrite(led1, HIGH); //ENCIENDA
digitalWrite(led2, LOW); //ENCIENDA
digitalWrite(led3, HIGH); //ENCIENDA
digitalWrite(led4, LOW); //ENCIENDA
digitalWrite(led5, LOW); //ENCIENDA
digitalWrite(led6, LOW); //ENCIENDA
digitalWrite(led7, LOW); //ENCIENDA
delay (retardo); //tiempo de retardo
digitalWrite(led1, HIGH); //ENCIENDA
digitalWrite(led2, HIGH); //ENCIENDA
digitalWrite(led3, HIGH); //ENCIENDA
digitalWrite(led4, HIGH); //ENCIENDA
digitalWrite(led5, HIGH); //ENCIENDA
digitalWrite(led6, HIGH); //ENCIENDA
digitalWrite(led7, HIGH); //ENCIENDA
delay (retardo); //tiempo de retardo
}

void animacion7(){
    digitalWrite(led1, LOW); //ENCIENDA
    digitalWrite(led2, HIGH); //ENCIENDA
    digitalWrite(led3, LOW); //ENCIENDA
    digitalWrite(led4, HIGH); //ENCIENDA
    digitalWrite(led5, LOW); //ENCIENDA
    digitalWrite(led6, HIGH); //ENCIENDA
    digitalWrite(led7, LOW); //ENCIENDA
    delay (retardo); //tiempo de retardo
    digitalWrite(led1, LOW); //ENCIENDA
    digitalWrite(led2, LOW); //ENCIENDA
    digitalWrite(led3, HIGH); //ENCIENDA
    digitalWrite(led4, LOW); //ENCIENDA
    digitalWrite(led5, HIGH); //ENCIENDA
    digitalWrite(led6, LOW); //ENCIENDA
    digitalWrite(led7, LOW); //ENCIENDA
```



```
delay (retardo); //tiempo de retardo
digitalWrite(led1, HIGH); //ENCIENDA
digitalWrite(led2, LOW); //ENCIENDA
digitalWrite(led3, LOW); //ENCIENDA
digitalWrite(led4, LOW); //ENCIENDA
digitalWrite(led5, LOW); //ENCIENDA
digitalWrite(led6, LOW); //ENCIENDA
digitalWrite(led7, LOW); //ENCIENDA
delay (retardo); //tiempo de retardo
digitalWrite(led1, HIGH); //ENCIENDA
digitalWrite(led2, LOW); //ENCIENDA
digitalWrite(led3, LOW); //ENCIENDA
digitalWrite(led4, LOW); //ENCIENDA
digitalWrite(led5, LOW); //ENCIENDA
digitalWrite(led6, LOW); //ENCIENDA
digitalWrite(led7, HIGH); //ENCIENDA
delay (retardo); //tiempo de retardo
digitalWrite(led1, HIGH); //ENCIENDA
digitalWrite(led2, LOW); //ENCIENDA
digitalWrite(led3, HIGH); //ENCIENDA
digitalWrite(led4, LOW); //ENCIENDA
digitalWrite(led5, HIGH); //ENCIENDA
digitalWrite(led6, LOW); //ENCIENDA
digitalWrite(led7, HIGH); //ENCIENDA
delay (retardo); //tiempo de retardo
digitalWrite(led1, HIGH); //ENCIENDA
digitalWrite(led2, HIGH); //ENCIENDA
digitalWrite(led3, HIGH); //ENCIENDA
digitalWrite(led4, LOW); //ENCIENDA
digitalWrite(led5, HIGH); //ENCIENDA
digitalWrite(led6, HIGH); //ENCIENDA
digitalWrite(led7, HIGH); //ENCIENDA
delay (retardo); //tiempo de retardo
```



```
digitalWrite(led1, HIGH); //ENCIENDA
digitalWrite(led2, HIGH); //ENCIENDA
digitalWrite(led3, HIGH); //ENCIENDA
digitalWrite(led4, HIGH); //ENCIENDA
digitalWrite(led5, HIGH); //ENCIENDA
digitalWrite(led6, HIGH); //ENCIENDA
digitalWrite(led7, HIGH); //ENCIENDA
delay (retardo); //tiempo de retardo
digitalWrite(led1, LOW); //ENCIENDA
digitalWrite(led2, LOW); //ENCIENDA
digitalWrite(led3, LOW); //ENCIENDA
digitalWrite(led4, LOW); //ENCIENDA
digitalWrite(led5, LOW); //ENCIENDA
digitalWrite(led6, LOW); //ENCIENDA
digitalWrite(led7, LOW); //ENCIENDA
delay (retardo); //tiempo de retardo
}

void loop() {
if ( digitalRead(pulsador)==1) {
pulso++;
Serial.println(pulso);
if (digitalRead(pulsador)==0) {
pulso++;
Serial.println(pulso);
}
if (pulso==1) {
animacion1();
}
if (pulso==2) {
animacion2();
}
if (pulso==3) {
animacion3();
}
```



```
        }

        if (pulso==4) {
            animacion4();
        }

        if (pulso==5) {
            animacion5();
        }

        if (pulso==6) {
            animacion6();
        }

        if (pulso==7) {
            animacion7();
        }
    }
}
```

RESULTADOS:

En esta practica se realizo un código con el cual por medio de un pulsador muestra varias animaciones para esto se incluye algunos if, donde en cada uno de estos se ingresa una animacion para que cada vez que se use el pulsador se cambie de animacion. Se debe esperar hasta que termine una animacion para que se realice la siguiente. El pulsador se puede conectar en algunos de los pines analógicos o en los digitales

PROYECTO 7: CONECTAR RESISTENCIA VARIABLE

OBJETIVOS:

- Conocer como funciona una resistencia variable y como se conecta al arduino
- Demostrar los conocimientos aprendidos
- Investigar sobre los diferentes problemas en los que se podria utilizar una resistencia variable

MATERIALES:

- Una Resistencia variable
- Un arduino



- Cable para conectar arduino
- Un computador
- Un protboard
- Varios cables jumper

DIAGRAMA:

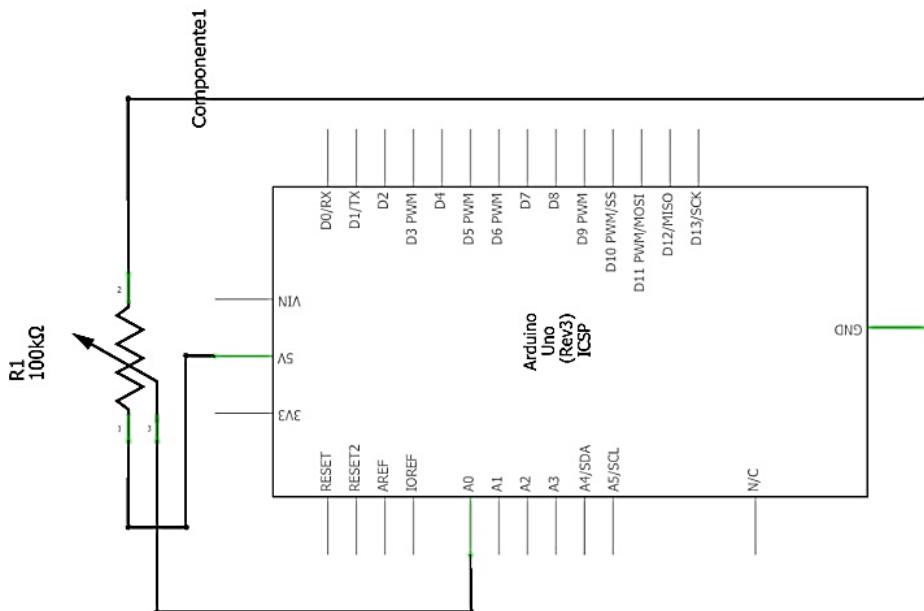
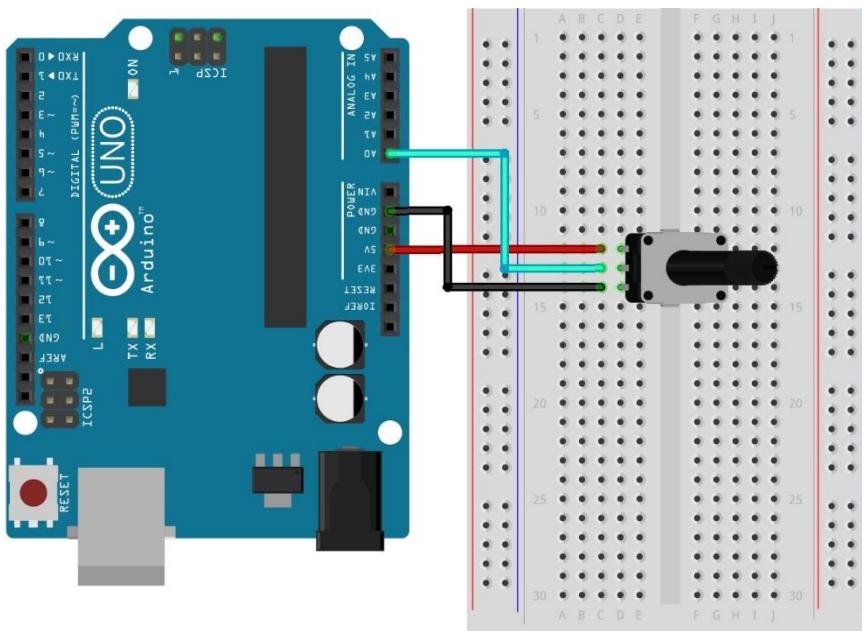


DIAGRAMA 7

ESQUEMA:



ESQUEMA 8



PROGRAMACION:

```
/*
 *
 *
 *          ULEAM-FACCI
 *
 *      TECNOLOGIA DE LA INFORMACION
 *
 *      SISTEMAS DIGITALES
 *
 *      MACIAS PICO JOSSELYN STEFANY
 *
 *
***** PRACTICA N° 6 *****
*
*      _____RESISTENCIA VARIABLE _____
*
*/
void setup() {
    //declarar entradas y salidas
    Serial.begin(9600);
}

void loop() {
    int Sensor = analogRead (A0);
    Serial.println (Sensor);
    delay(1);
}
```

RESULTADO:

En esta practica se aprendio a utilizar y conectar una resistencia variable y se tuvo en cuenta que la resistencia variable tiene tres pines el primero es el que se conecta con el positivo, el segundo con el pin de el Arduino y el tercero con GND que es tierra, conociendo esto no se tuvo problema alguno para realizar las conexiones. En este proyecto se utilizo el monitor serie para mostrar los datos de la resistencia variable de forma correspondiente.



PROYECTO 8: CONTROLAR EL RETARDO DEL LED CON RESISTENCIA VARIABLE

OBJETIVO:

- Practicar mediante un led el funcionamiento de una resistencia variable
- Controlar la rapidez con la que se requiere que cambie de apagado a encendido el led

MATERIALES:

- Una resistencia variable
- Un led
- Una resistencia de $330\ \Omega$
- Un arduino uno
- Cable para conectar arduino
- Varios cables jumper
- Un computador
- Un protoboard



DIAGRAMA:

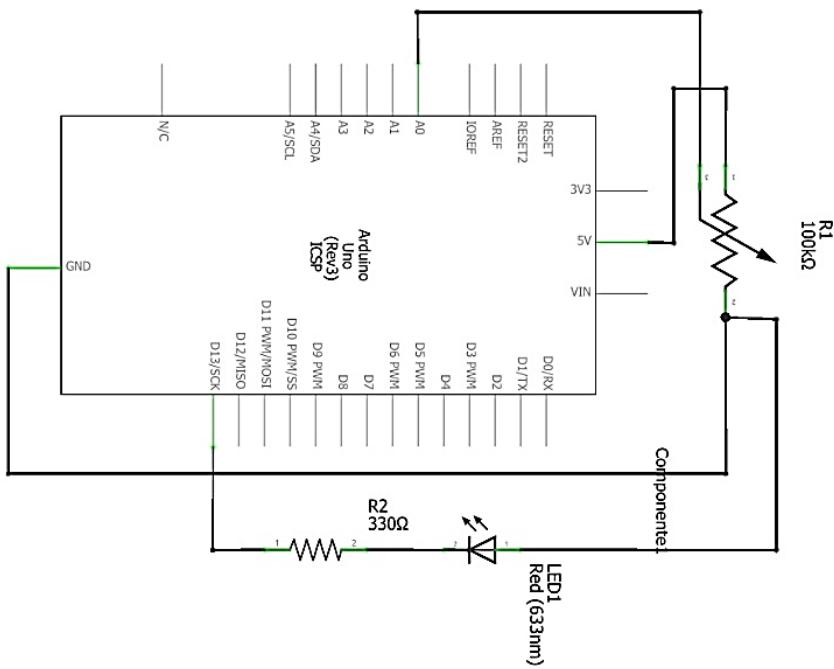
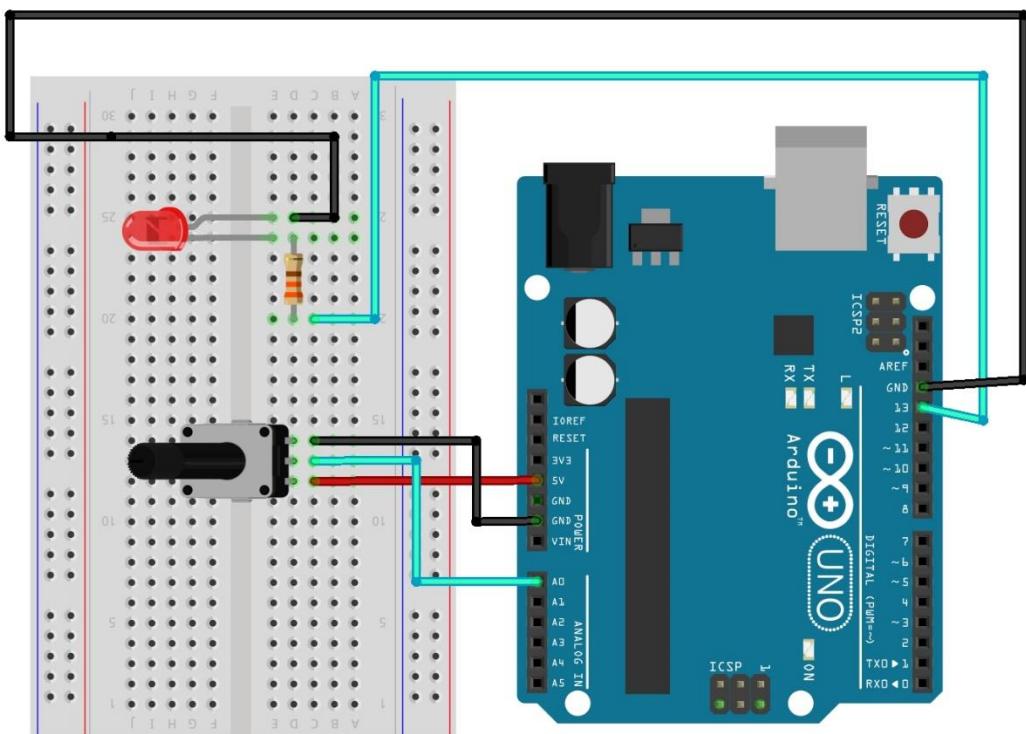


DIAGRAMA 8

ESQUEMA:



ESQUEMA 9



PROGRAMACION:

```
*  
*          ULEAM-FACCI  
*          TECNOLOGIA DE LA INFORMACION  
*          SISTEMAS DIGITALES  
*          MACIAS PICO JOSSELYN STEFANY  
*  
***** PRACTICA N° 8 *****  
*  
*          _____ CONTROLAR RETARDO DE LED MEDIANTE POTENCIOMETRO _____  
*  
*/  
  
int led =13;  
int nuevo =0;  
  
void setup() {  
    //declarar entradas y salidas  
    Serial.begin(9600);  
    pinMode (led, OUTPUT);  
}  
  
void loop() {  
    int Sensor = analogRead (A0);  
    nuevo=Sensor+100;  
    digitalWrite(led,HIGH);  
    delay(nuevo);  
    digitalWrite(led,LOW);  
    delay(nuevo);  
}
```

RESULTADOS:

Como conclusión dentro de esta practica podríamos decir que se ha utilizado una nueva variable para guardar el dero con el que se hará el retardo y es por eso que a este dato se le suma 100 para que encada momento quese vuelva a ejecutar vaya de acuerdo a la resitencia variable.



PROYECTO 9: VARIAR NUMEROS MEDIANTE UNA RESISTENCIA

VARIABLE

OBJETIVOS:

- Practicar como la resistencia variable puede ayudar a variar algunos tipos de datos.
- Utilizar la conversión del dato dado por la variable a un numero exacto

MATERIALES:

- Un arduino uno
- Una resistencia variable
- Cable para conectar arduino
- Varios cables jumper
- 1 protoboard

DIAGRAMA:

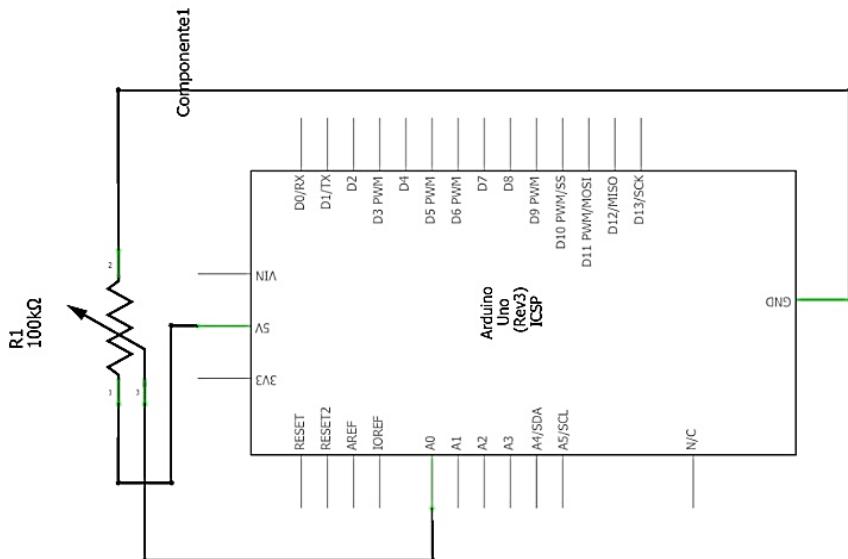
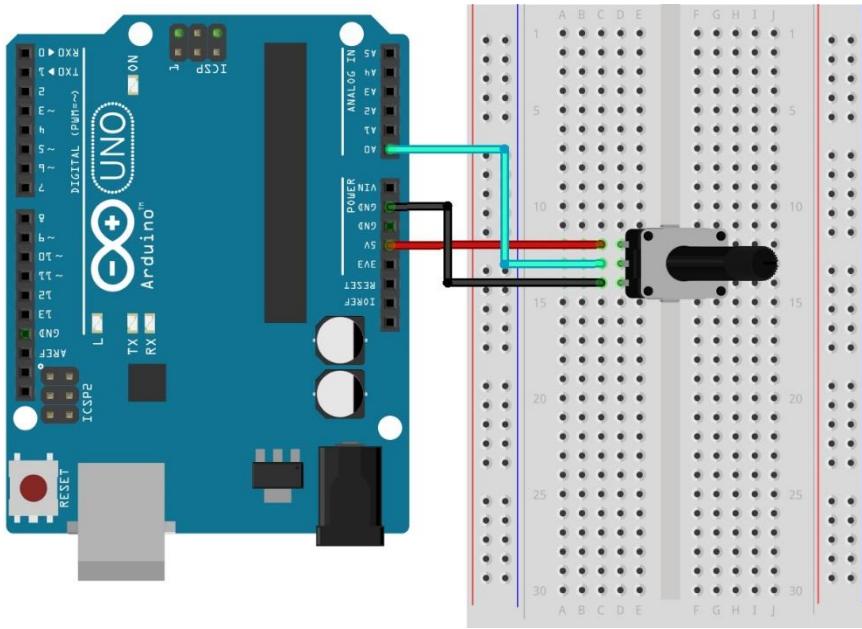


DIAGRAMA 9



ESQUEMA:



ESQUEMA 10

PROGRAMACION:

```
/*
*
*
*          ULEAM-FACCI
*
*      TECNOLOGIA DE LA INFORMACION
*
*      SISTEMAS DIGITALES
*
*      MACIAS PICO JOSSELYN STEFANY
*
*
***** PRACTICA N° 9 *****
*
*
*      _____ENCENDIDO CIRCULAR DE VARIOS LED's _____
*
*
*PARA TAREA COMENTAR TODO EL CODIGO
*
*/
int nuevo =0;
void setup() {
    Serial.begin(9600);
```



```
}

void loop() {
    int Sensor = analogRead (A0);
    int Conver = map(Sensor, 0, 1023, 0, 100);
    Serial.print(" El valor de la resistencia es de aproximadamente ");
    Serial.print(Conver);
    Serial.println (" KILO OHMIOS ");
    delay (20);
}
```

RESULTADOS:

Se ha concluido que de acuerdo a los dos circuitos anteriores que se han realizado con la resistencia variable en esta lo unico que aumenta la practica 7 es que se convierte el dato para dar un valor exato con inicio y final.

PROYECTO 10: PRENDER LED MEDIANTE NUMERO INGRESADO

OBJETIVOS:

- Utilizar el monitor serie para que el usuario pueda ingresar un numero según el led que quiera encender
- Realizar la codificación para que al ingresar el numero se haga aquel proceso mediante un switch

MATERIALES:

- 7 leds
- 7resistencias de 330
- Un arduino uno
- Cable para conectar arduino
- Varios cables jumper
- Un computador
- Un protoboard



DIAGRAMA:

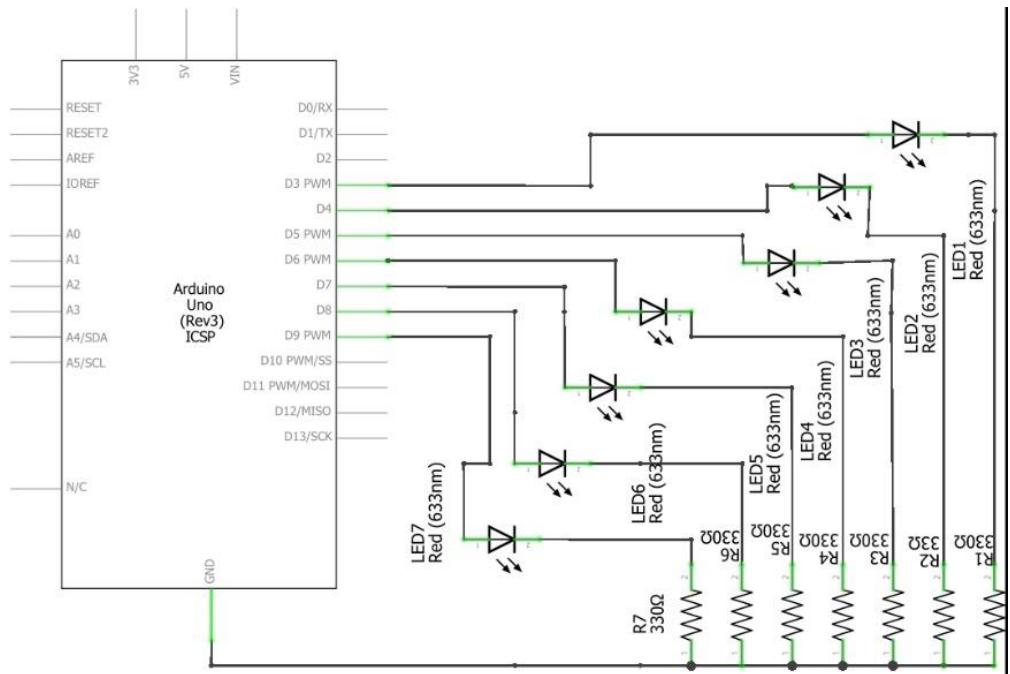
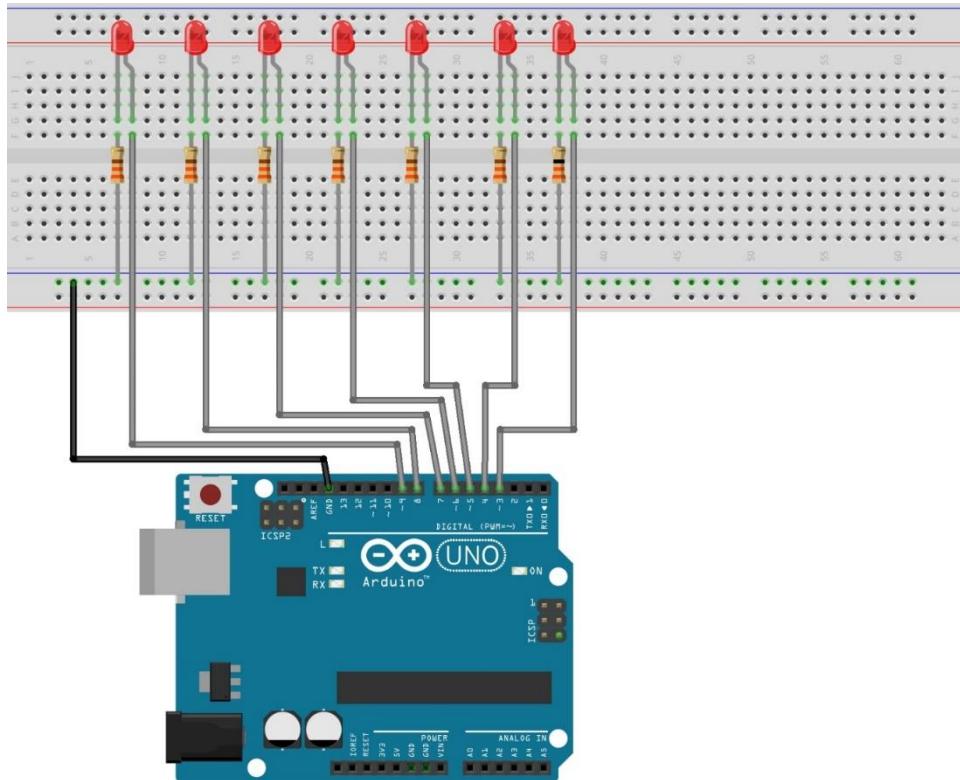


DIAGRAMA 10

ESQUEMA:



ESQUEMA 11



PROGRAMACION:

```
/*
 *
 *          ULEAM-FACCI
 *
 *      TECNOLOGIA DE LA INFORMACION
 *
 *      SISTEMAS DIGITALES
 *
 *      MACIAS PICO JOSSELYN STEFANY
 *
 *
***** PRACTICA N° 10 *****
*
*      _____ ANIMACIONES CON 7 LED _____
*
*PARA TAREA COMENTAR TODO EL CODIGO
*
*/
void setup() {
    //inicializa la comunicacion serial
    Serial.begin(9600);
    //Se usa un for se inicializa el LED Pin
    for (int LED=3; LED<=9; LED++) {
        pinMode(LED, OUTPUT);
    }
}

void loop() {
    //Se le pone lectura serial para que lea por medio de
    //compilacion
    // put your main code here, to run repeatedly:
    //Leer el sensor (posible para poner una comunicacion por medio
    //de bluetooth o inalambrica
    if (Serial.available()>0){ //cuando se tiene un caracter y se
    manda un caracter que va a ser mayor a 0
```



`int lectura= Serial.read(); //lectura en ves inByte (lectura es igual a lo que lea el lector sereal`

```
switch (lectura) {  
    case '1':  
        digitalWrite(3, HIGH);  
        break;  
    case '2':  
        digitalWrite(4, HIGH);  
        break;  
    case '3':  
        digitalWrite(5, HIGH);  
        break;  
    case '4':  
        digitalWrite(6, HIGH);  
        break;  
    case '5':  
        digitalWrite(7, HIGH);  
        break;  
    case '6':  
        digitalWrite(8, HIGH);  
        break;  
    case '7':  
        digitalWrite(9, HIGH);  
        break;  
    case 'x':  
        for (int LED =3; LED<=9; LED++) {  
            digitalWrite(LED, LOW);  
        }  
    }  
}
```



RESULTADOS:

En este proyecto se ha ejecutado un programa dentro del cual se ha realizado mediante un switch para que cada dato que se ingrese este declarado en cada uno de los casos y asi de acuerdo a cada uno de ellos mostrar el encendido del LED pedido estos datos van del 1 al 7 y se ha agregado al código un numero x el cual nos indica que si es una x entonces se apagaran todos los leds.

PROYECTO 11: UTILIZAR POTENCIOMETRO PARA INDICAR LA TEMPERATURA DE UNA BODEGA

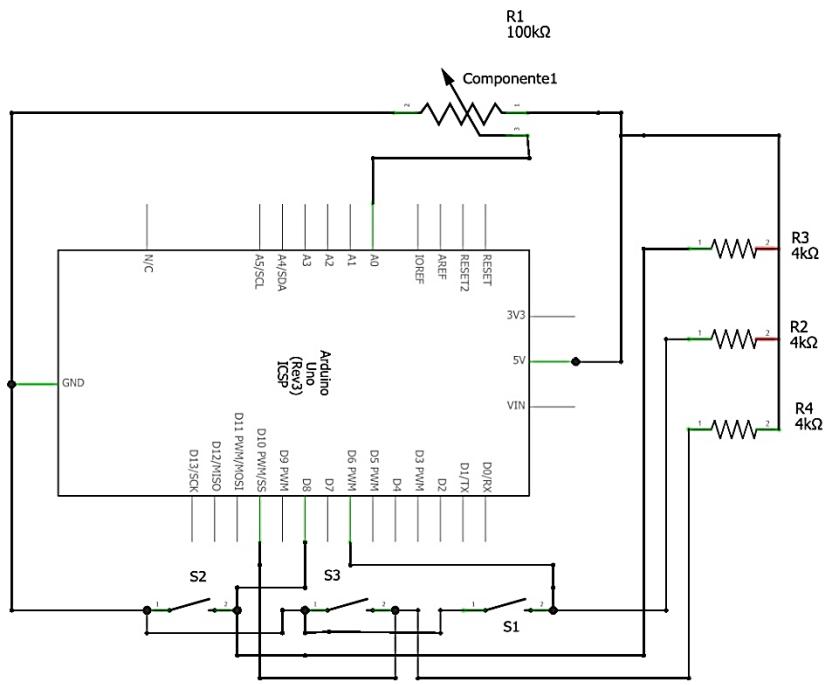
OBJETIVOS:

- Ejecutar el código de manera que mediante los pulsadores pueda decir si la bodega esta vacia o no
- Realizar algunas condiciones if para que indique de acuerdo a la temperatura indicada por el potenciómetro

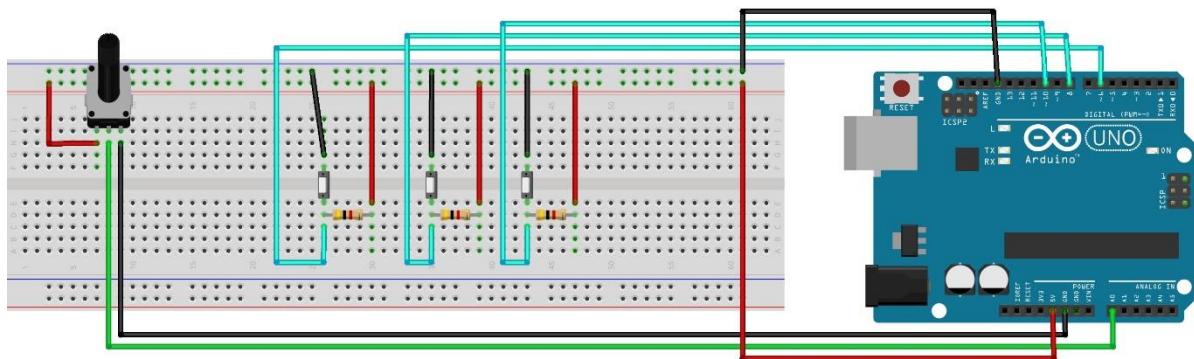
MATERIALES:

- 3 pulsadores
- 3 resistencias 4 kOhmios
- Una resistencia variable
- Un arduino uno
- Varios cables jumper
- Un protoboard
- Cable para conectar arduino
- Un computador

DIAGRAMA:



ESQUEMA:



PROGRAMACION:

```
/* 3 SENsores DIGITALES QUE DIGA BODEGA VACIA
 * BODEGA AL 25% TEMPERATURA DEL MOMENTO
 * BODEGA AL 50% TEMPERATURA AL MOMENTO
 * BODEGA AL 100%
 * SE PUEDE USAR QUE RANGO TIENE EL SENSOR ANALOGICO
 *
 */
int sensor=analogRead(A1);
int conver=map(sensor,0,100,0,10);
```



```
int pulsador1=6;
int pulsador2=8;
int pulsador3=10;

void setup() {
    pinMode(pulsador1, INPUT);
    pinMode(pulsador2, INPUT);
    pinMode(pulsador3, INPUT);
    Serial.begin(9600);

}

void temperatura(){
    sensor=analogRead(A1);
    conver= map(sensor,0,100,0,10);
    if(conver>50){
        Serial.println(" LA TEMPERATURA ALTA... ESTA MUY CALIENTE
ACTIVE VENTILACION ");
        Serial.println("SU TEMPERATURA ES ");
        Serial.println(conver);
    }else{
        if(conver>25){
            Serial.println(" LA TEMPERATURA ESTA NORMAL... ");
            Serial.println("SU TEMPERATURA ES ");
            Serial.println(conver);
        }
        else{
            if(conver==0){
                Serial.println(" LA TEMPERATURA esta en cero... ");
                Serial.println("SU TEMPERATURA ES ");
                Serial.println(conver);
            }
            else{
                Serial.println(" LA TEMPERATURA esta baja... ");
```



```
Serial.println("SU TEMPERATURA ES ");

Serial.println(conver);

}

}

delay(500);

}

void loop() {

if (digitalRead(pulsador1)==HIGH) {

Serial.println(" LA BODEGA ESTA EN 25% ");

temperatura();

}

else{

if(digitalRead(pulsador2)==HIGH) {

Serial.println(" LA BODEGA ESTA EN 50% ");

temperatura();

}

else{

if(digitalRead(pulsador3)==HIGH) {

Serial.println(" LA BODEGA ESTA EN 100% ");

temperatura();

} else{



Serial.println(" LA BODEGA ESTA VACIA ");

temperatura();

}

}

}

delay(500);

}
```

RESULTADO:

Dentro de este programa se ha realizado un método aparte que es donde ese realiza toda la operación para que muestre la temperatura ingresada donde se convierte el dato de la



resistencia variable en un numero exacto y ese de allí se muestra dependiendo las condiciones entre si es mayor o menor al numero convertido.

Al ingresar al Loop que es donde se pone todo lo que se va a ejecutar se realizan otras condiciones donde nos diga sobre las bodegas si esta vacia, en 50% o en 100%

PRACTICA 12: CONTADOR BINARIO DE 8 BITS

OBJETIVOS:

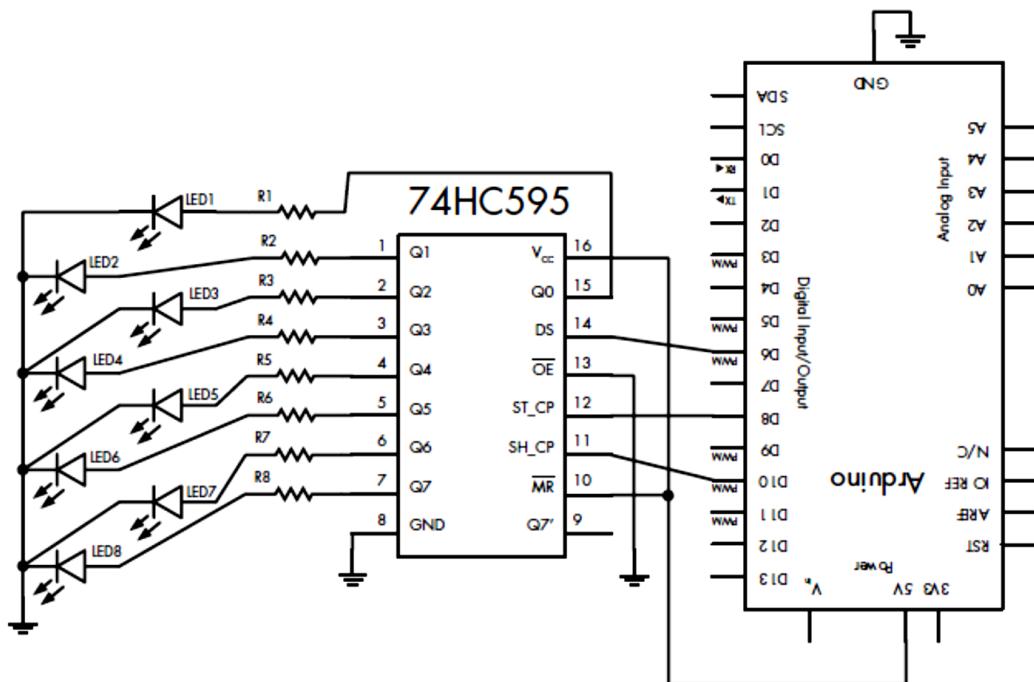
- Ejecute la programación de tal manera que los leds muestren de forma binaria los números comprendidos del 0 al 255.
- Conocer los numeros binarios
- Conocer como es el funcionamiento del Arduino y los puertos de conexión
- Investigar sobre el funcionamiento o las conexiones del circuito 74HC595.

MATERIALES:

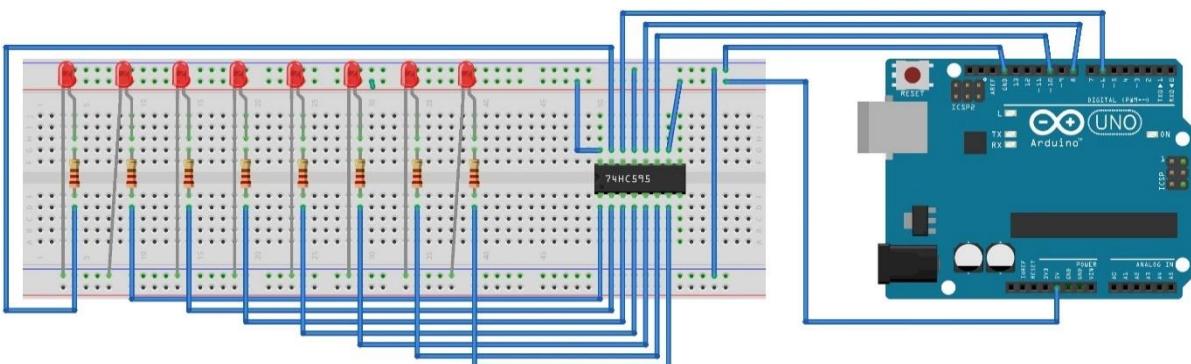
- 1 Arduino
- 8 Leds
- 8 Resistencias de $330\ \Omega$
- 1 Circuito Integrado de 74HC595
- Varios cables Jumper
- 1 Protoboard
- Cable para conectar Arduino
- Computadora
- Programa Arduino



DIAGRAMA:



ESQUEMA:



PROGRAMACION:

```
/*
 *          ULEAM-FACCI
 *  TECNOLOGIA DE LA INFORMACION
 *  SISTEMAS DIGITALES
 *  MACIAS PICO JOSSELYN STEFANY
 *
 **** PROYECTO N°1 ****
 *
 *  ANIMACIONES CON 7 LED
 *
 */
```



```
//Se declaran los pines
int pin14=6;
int pin12=8;
int pin11=10;

void setup() {
    pinMode(pin14,OUTPUT); //cuales serán los pines de salida
    pinMode(pin12,OUTPUT); //cuales serán los pines de salida
    pinMode(pin11,OUTPUT); //cuales serán los pines de salida
}
void loop() {
    Serial.println( " Números comprendidos del 0 al 255 en binarios "
);
//Este for es el que va a mostrar los numeros desde 0 hasta
for (int x=0; x<256; x++) {
    digitalWrite(pin12, LOW);
    shiftOut(pin14, pin11, MSBFIRST, x);
    digitalWrite(pin12, HIGH);
    delay(1000);
}
}
```

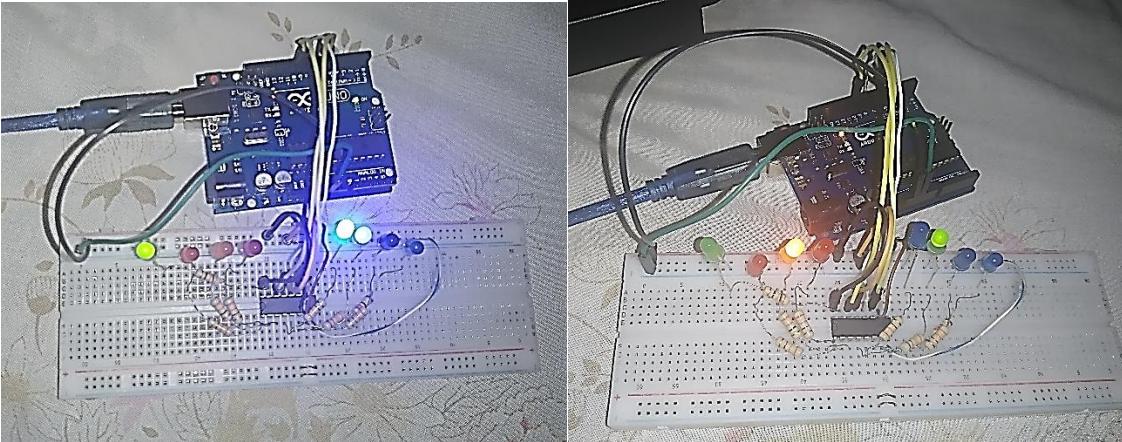
RESULTADOS:

Para lograr obtener los resultados propuestos de acuerdo con nuestro objetivo. Hemos investigado un poco de algunas funciones nuevas como el “shiftOut” que sirve para escribir los bits uno a la vez. Puede empezar desde el bit más o menos significante. Este método es muy útil para manejar el registro del desplazamiento.

Se recomienda a la hora de realizar el circuito en físico, tener en cuenta el lado positivo y el negativo en los leds, así mismo tener no confundirse en los pines del circuito con su parte inicial y final.



ANEXOS:



PRACTICA 13: JUEGO DE NUMEROS ALEATORIOS MEDIANTE NUMEROS BINARIOS

OBJETIVOS:

- Poner en práctica algunos de los conocimientos obtenidos en el primer parcial.
- Mostrar dentro del monitor serie un espacio para ingresar el numero en decimal mostrado por los leds.
- Mostrar un numero aleatorio en binario mediante los leds para que el usuario identifique el numero en decimal.
- Convertir números de binarios a decimales
- Ingresar el dato presentado, en forma decimal por medio del monitor serie.

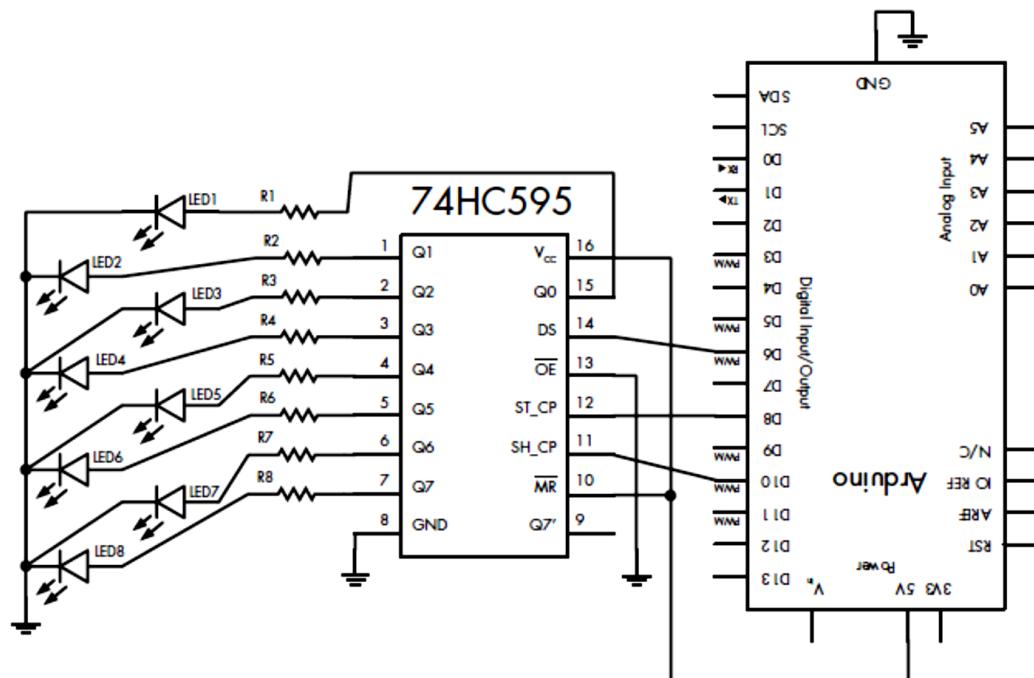
MATERIALES:

- 1 Arduino
- 8 Leds
- 8 Resistencias de $360\ \Omega$
- 1 Circuito Integrado de 74HC595
- Varios cables Jumper
- 1 Protoboard

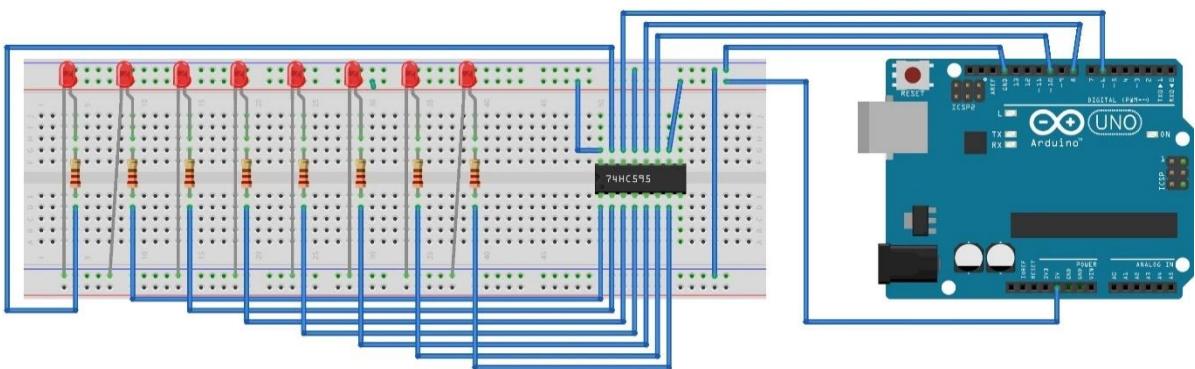


- Cable para conectar Arduino
- Computadora
- Programa Arduino

DIAGRAMA:



ESQUEMA:



PROGAMACION:

```
/*
 *
 *
 * ULEAM-FACCI
 * TECNOLOGIA DE LA INFORMACION
```



```
*           SISTEMAS DIGITALES
*   MACIAS PICO JOSSELYN STEFANY
*
***** PROYECTO N° 2 *****
*
*_REALIZAR PRUEBAS Y COMENTAR CODIGO_
*/
/*
#define DATA 6    // conectado al pin 14 del CI 74HC595
#define LATCH 8   // conectado al pin 12 del CI 74HC595
#define CLOCK 10  // conectado al pin 11 del CI 74HC595

int numero = 0; //declara variable número inicializada en 0
int respuesta = 0; //declara variable respuesta inicializada en 0

void setup(){
pinMode(LATCH, OUTPUT); //Este es una salida conectado a el pin 8 del
arduino
pinMode(CLOCK, OUTPUT); //Este es una salida conectado a el pin 10
del arduino
pinMode(DATA, OUTPUT); //Este es una salida conectado a el pin 6 del
arduino
Serial.begin(9600); //Abre el puerto serie, configura los datos a una
comunicacion de 9600 bps
randomSeed(analogRead(0));      // genera un número aleatorio
analogRead(0)
mostrarNumero(0);
}
//Este es un metodo donde se muestra el numero aleatorio por medio
de numeros binarios por medio de los LEDS
void mostrarNumero(byte b){ //Este recibe el numero dado
aleatoriamente
digitalWrite(LATCH, LOW); //La salida conectada al pin 8 del arduino
estara apagada
shiftOut(DATA, CLOCK, MSBFIRST, b); //Este es donde se convierte el
numero ingresado en Bit
digitalWrite(LATCH, HIGH);//cuando ya sucede el caso anterior la
salida conectada al pin 8 del arduino se enciende
}
//Este es un metodo para obtener el dato que el usuario ingresa
void ObtenerRespuesta(){
Serial.flush(); //Espera que se complete la transmision de los datos
salientes
while (Serial.available() == 0){
//Este metodo es para que no se cambie el numero aleatorio hasta
que no se termine el proceso
}
while(Serial.available() > 0){
int x=Serial.parseInt(); //Busca el siguiente entero valido en la
serie entrante, la funcion terminara
//si se agota el tiempo de espera
if (x>0){
respuesta=x;
}
```



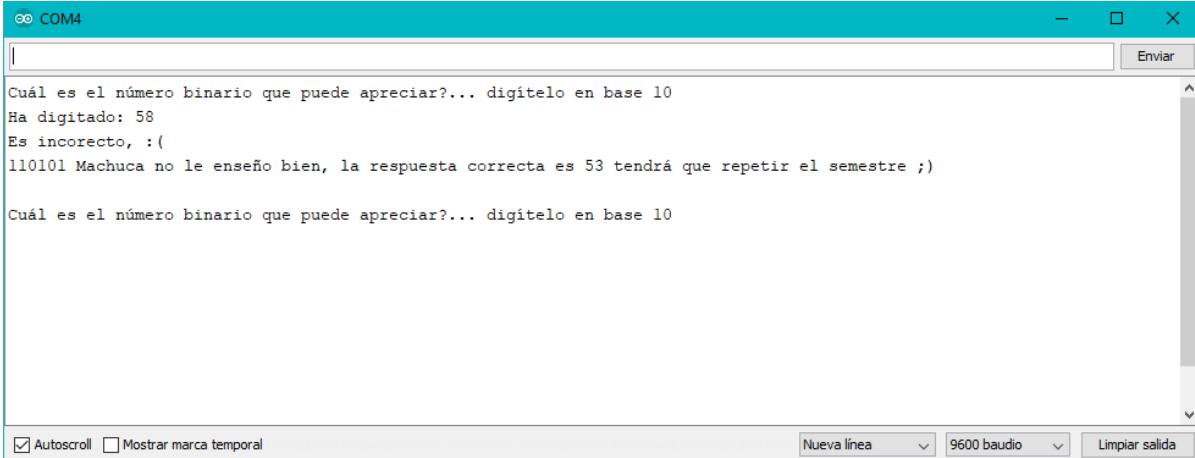
```
//respuesta = respuesta * 10;
//z = Serial.read() - '0';
//respuesta = respuesta + z;
delay(5);
}
}
Serial.print("Ha digitado: ");
Serial.println(respuesta);1
}
void chequeaRespuesta(){
    //Este metodo va a revisar si la respuesta esta correcta o no lo
esta
if (respuesta == numero) { //realiza una condicional y verifica si
la respuesta es correcta
//En este metodo se compara el valor aleatorio con el valor ingresado
por el usuario
// se realiza si la condicion es afirmativa
Serial.print("La respuesta es correcta! ");
Serial.print(respuesta, BIN);
Serial.print( " es igual a " );
Serial.println(numero);
Serial.println();
}
else{ // se realiza si la respuesta es incorrecta
Serial.println( "Es incorrecto, :( " );
Serial.print(numero, BIN);
Serial.print( " Machuca no le enseño bien, la respuesta correcta es
" );
Serial.print(numero);
Serial.println( " tendrá que repetir el semestre ;)" );
Serial.println();
}
respuesta = 0;
delay(1000);
}
void loop (){
numero = random(255); // Solo se obtienen numeros hasta el 256
mostrarNumero(numero); //Se ingresa al metodo de mostrarNumero
Serial.println( "Cuál es el número binario que puede apreciar?...
digítelo en base 10 " );
ObtenerRespuesta(); //Se llama al metodo para obtener la respuesta
chequeaRespuesta(); //Se llama al metodo para chequear la respuesta
}
```

RESULTADOS:

Durante el proceso se realizó algunas pruebas, para poder así darnos cuenta de que una parte del código no estaba bien , en este caso era el método llamado obtener respuesta, ya que en el código dado, al ingresar el número se cambiaba y no salía la respuesta correcta (Se realizaba un cambio con el número ingresado), donde se integró la parte Serial.parseInt() que esta es utilizada para buscar el siguiente entero valido en la serie entrante.



ANEXOS:

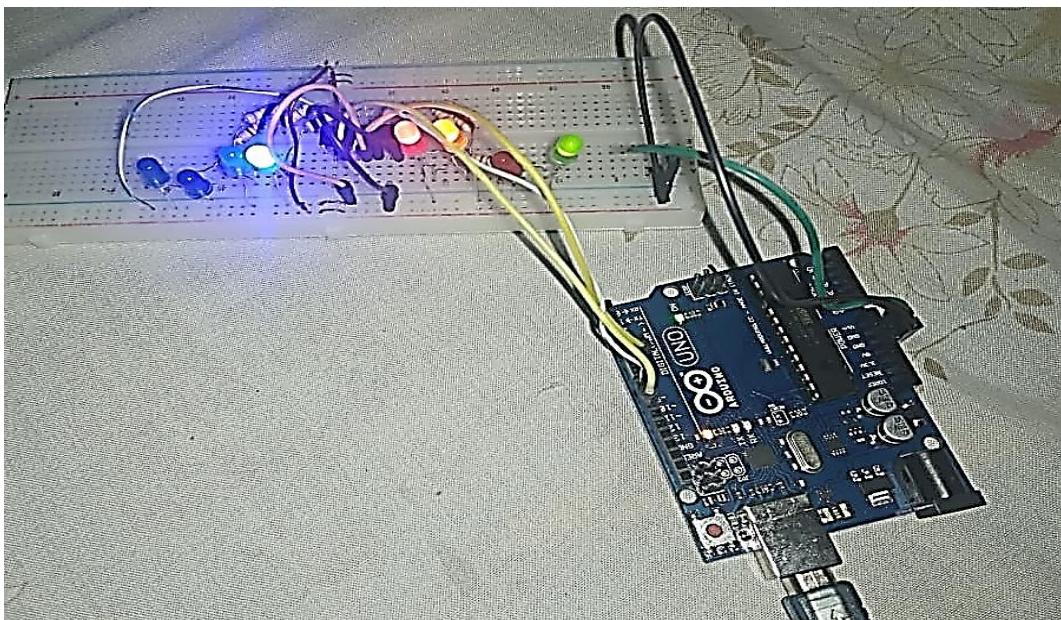


The screenshot shows the Arduino IDE's Serial Monitor window titled "COM4". The text in the window is as follows:

```
Cuál es el número binario que puede apreciar?... digitelo en base 10
Ha digitado: 58
Es incorrecto, :(
110101 Machuca no le enseño bien, la respuesta correcta es 53 tendrá que repetir el semestre ;)

Cuál es el número binario que puede apreciar?... digitelo en base 10
```

At the bottom of the window, there are several buttons and settings: "Autoscroll" (checked), "Mostrar marca temporal" (unchecked), "Nueva línea" (dropdown set to "Nueva línea"), "9600 baudio" (dropdown set to "9600 baudio"), and "Limpiar salida" (button).



PRACTICA 14: CONTADOR HEXADECIMAL DE UN BIT EN UN DISPLAY DE 7 SEGMENTOS AUTOMATICAMENTE

OBJETIVOS:

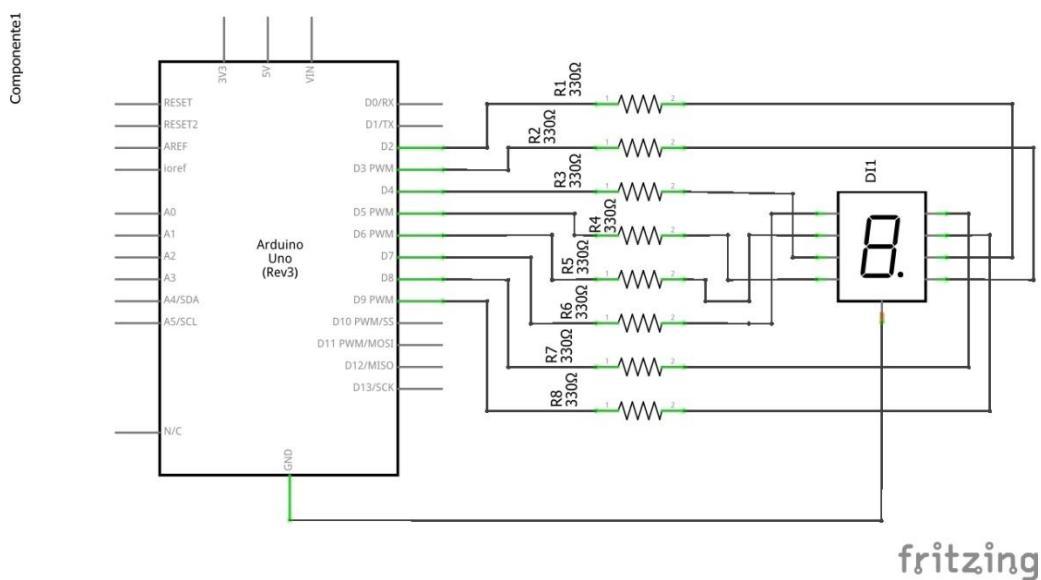
- Mostrar los números hexadecimales ya conocidos
- Conocer acerca de las conexiones que se debe realizar para mostrar los datos en un display de 7 segmentos
- Ejecutar el programa de manera que se muestren los números hexadecimales



MATERIALES:

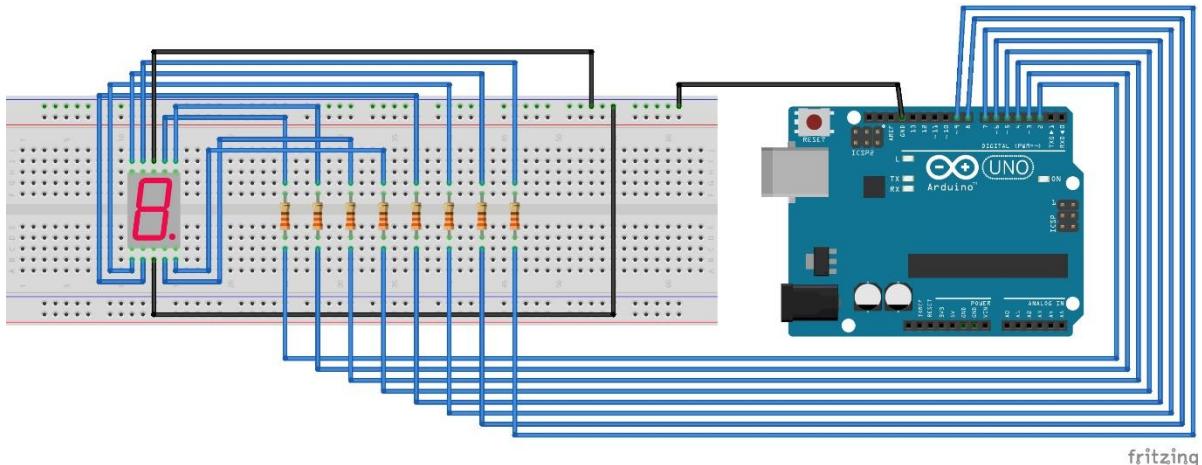
- Un arduino uno
- Cable para conectar arduino
- 1 display de 7 segmentos
- 8 resistencias de $330\ \Omega$
- Un protoboard
- Varios cables jumper
- Un computador
- Programa arduino

DIAGRAMA:





ESQUEMA:



PROGRAMACION:

```
/*
 *      ULEAM-FACCI
 *      TECNOLOGIA DE LA INFORMACION
 *      SISTEMAS DIGITALES
 *      MACIAS PICO JOSSELYN STEFANY
 *
 * ***** PROYECTO N° 3 *****
 *
 * _NUMEROS HEXADECIMALES CON DISPLAY_
 */
//Declarar e iniciar variables
int a=2;
int b=3;
int c=4;
int dp=5;
int d=6;
int e=7;
int f=9;
int g=8;

void setup () {

    pinMode( a, OUTPUT);
    pinMode( b, OUTPUT);
    pinMode( c, OUTPUT);
    pinMode( dp, OUTPUT);
    pinMode( d, OUTPUT);
    pinMode( e, OUTPUT);
    pinMode( f, OUTPUT);
    pinMode( g, OUTPUT);

}

//Metodo para realizar el 0
```



```
void cero() {
    digitalWrite(a,HIGH );
    digitalWrite(b,HIGH );
    digitalWrite(c,HIGH );
    digitalWrite(d,HIGH );
    digitalWrite(e,HIGH );
    digitalWrite(f,HIGH );
    digitalWrite(g,LOW );
    delay(1000);
    digitalWrite(a,LOW );
    digitalWrite(b,LOW );
    digitalWrite(c,LOW );
    digitalWrite(d,LOW );
    digitalWrite(e,LOW );
    digitalWrite(f,LOW );
    digitalWrite(g,LOW );
}

//Metodo para realizar el 1
void uno(){
    digitalWrite(a,LOW );
    digitalWrite(b,HIGH );
    digitalWrite(c,HIGH );
    digitalWrite(d,LOW );
    digitalWrite(e,LOW );
    digitalWrite(f,LOW );
    digitalWrite(g,LOW );
    delay(1000);
    digitalWrite(a,LOW );
    digitalWrite(b,LOW );
    digitalWrite(c,LOW );
    digitalWrite(d,LOW );
    digitalWrite(e,LOW );
    digitalWrite(f,LOW );
    digitalWrite(g,LOW );

}

//Metodo para realizar el 2
void dos(){
    digitalWrite(a,HIGH );
    digitalWrite(b,HIGH );
    digitalWrite(g,HIGH );
    digitalWrite(e,HIGH );
    digitalWrite(d,HIGH );
    delay(1000);

    digitalWrite(a,LOW );
    digitalWrite(b,LOW );
    digitalWrite(c,LOW );
    digitalWrite(d,LOW );
    digitalWrite(e,LOW );
    digitalWrite(f,LOW );
    digitalWrite(g,LOW );
}

//Metodo para realizar el 3
```



```
void tres() {
    digitalWrite(a,HIGH );
    digitalWrite(b,HIGH );
    digitalWrite(g,HIGH );
    digitalWrite(c,HIGH );
    digitalWrite(d,HIGH );
    digitalWrite( e,LOW );
    delay(1000);

    digitalWrite(a,LOW );
    digitalWrite(b,LOW );
    digitalWrite(c,LOW );
    digitalWrite(d,LOW );
    digitalWrite(e,LOW );
    digitalWrite(f,LOW );
    digitalWrite(g,LOW );
}

//Metodo para realizar el 4
void cuatro(){
    digitalWrite(f,HIGH );
    digitalWrite(g,HIGH );
    digitalWrite(b,HIGH );
    digitalWrite(c,HIGH );
    delay(1000);

    digitalWrite(a,LOW );
    digitalWrite(b,LOW );
    digitalWrite(c,LOW );
    digitalWrite(d,LOW );
    digitalWrite(e,LOW );
    digitalWrite(f,LOW );
    digitalWrite(g,LOW );
}

//Metodo para realizar el 5
void cinco () {
    digitalWrite(a,HIGH );
    digitalWrite(f,HIGH );
    digitalWrite(g,HIGH );
    digitalWrite(c,HIGH );
    digitalWrite(d,HIGH );
    delay(1000);

    digitalWrite(a,LOW );
    digitalWrite(b,LOW );
    digitalWrite(c,LOW );
    digitalWrite(d,LOW );
    digitalWrite(e,LOW );
    digitalWrite(f,LOW );
    digitalWrite(g,LOW );
}

//Metodo para realizar el 6
void seis(){
    digitalWrite(a,HIGH );
    digitalWrite(f,HIGH );
```



```
digitalWrite(g,HIGH );
digitalWrite(e,HIGH );
digitalWrite(d,HIGH );
digitalWrite(c,HIGH );
delay(1000);

digitalWrite(a,LOW );
digitalWrite(b,LOW );
digitalWrite(c,LOW );
digitalWrite(d,LOW );
digitalWrite(e,LOW );
digitalWrite(f,LOW );
digitalWrite(g,LOW );
}

//Metodo para realizar el 7
void siete(){
    digitalWrite(a,HIGH);
    digitalWrite(b,HIGH);
    digitalWrite(c,HIGH);
    delay(1000);

    digitalWrite(a,LOW );
    digitalWrite(b,LOW );
    digitalWrite(c,LOW );
    digitalWrite(d,LOW );
    digitalWrite(e,LOW );
    digitalWrite(f,LOW );
    digitalWrite(g,LOW );
}
//Metodo para realizar el 8
void ocho(){
    digitalWrite(a,HIGH );
    digitalWrite(b,HIGH );
    digitalWrite(c,HIGH );
    digitalWrite(g,HIGH );
    digitalWrite(d,HIGH );
    digitalWrite(e,HIGH );
    digitalWrite(f,HIGH );
    delay(1000);

    digitalWrite(a,LOW );
    digitalWrite(b,LOW );
    digitalWrite(c,LOW );
    digitalWrite(g,LOW );
    digitalWrite(d,LOW );
    digitalWrite(e,LOW );
    digitalWrite(f,LOW );
}
//Metodo para realizar el 9
void nueve(){
    digitalWrite(a,HIGH );
    digitalWrite(b,HIGH );
    digitalWrite(c,HIGH );
    digitalWrite(g,HIGH );
```



```
digitalWrite(d,HIGH );
digitalWrite(f,HIGH );
delay(1000);
digitalWrite(a,LOW );
digitalWrite(b,LOW );
digitalWrite(c,LOW );
digitalWrite(d,LOW );
digitalWrite(e,LOW );
digitalWrite(f,LOW );
digitalWrite(g,LOW );
}
//Metodo para realizar el A
void AA(){
digitalWrite(a,HIGH );
digitalWrite(b,HIGH );
digitalWrite(c,HIGH );
digitalWrite(g,HIGH );
digitalWrite(e,HIGH );
digitalWrite(f,HIGH );
delay(1000);
digitalWrite(a,LOW );
digitalWrite(b,LOW );
digitalWrite(c,LOW );
digitalWrite(d,LOW );
digitalWrite(e,LOW );
digitalWrite(f,LOW );
digitalWrite(g,LOW );
}
//Metodo para realizar el B
void BB(){
digitalWrite(c,HIGH );
digitalWrite(g,HIGH );
digitalWrite(d,HIGH );
digitalWrite(e,HIGH );
digitalWrite(f,HIGH );
delay(1000);
digitalWrite(a,LOW );
digitalWrite(b,LOW );
digitalWrite(c,LOW );
digitalWrite(d,LOW );
digitalWrite(e,LOW );
digitalWrite(f,LOW );
digitalWrite(g,LOW );
}
//Metodo para realizar el C
void CC(){
digitalWrite(a,HIGH );
digitalWrite(d,HIGH );
digitalWrite(e,HIGH );
digitalWrite(f,HIGH );
delay(1000);
digitalWrite(a,LOW );
digitalWrite(b,LOW );
digitalWrite(c,LOW );
```



```
digitalWrite(d, LOW );
digitalWrite(e, LOW );
digitalWrite(f, LOW );
digitalWrite(g, LOW );
}

//Metodo para realizar el D
void DD(){
digitalWrite(b, HIGH );
digitalWrite(c, HIGH );
digitalWrite(d, HIGH );
digitalWrite(e, HIGH );
digitalWrite(g, HIGH );

delay(1000);
digitalWrite(a, LOW );
digitalWrite(b, LOW );
digitalWrite(c, LOW );
digitalWrite(d, LOW );
digitalWrite(e, LOW );
digitalWrite(f, LOW );
digitalWrite(g, LOW );
}

//Metodo para realizar el E
void EE(){
digitalWrite(a, HIGH );
digitalWrite(g, HIGH );
digitalWrite(d, HIGH );
digitalWrite(e, HIGH );
digitalWrite(f, HIGH );
delay(1000);
digitalWrite(a, LOW );
digitalWrite(b, LOW );
digitalWrite(c, LOW );
digitalWrite(d, LOW );
digitalWrite(e, LOW );
digitalWrite(f, LOW );
digitalWrite(g, LOW );
}

//Metodo para realizar el F
void FF(){
digitalWrite(a, HIGH );
digitalWrite(g, HIGH );
digitalWrite(e, HIGH );
digitalWrite(f, HIGH );
delay(1000);
digitalWrite(a, LOW );
digitalWrite(b, LOW );
digitalWrite(c, LOW );
digitalWrite(d, LOW );
digitalWrite(e, LOW );
digitalWrite(f, LOW );
digitalWrite(g, LOW );
}
```

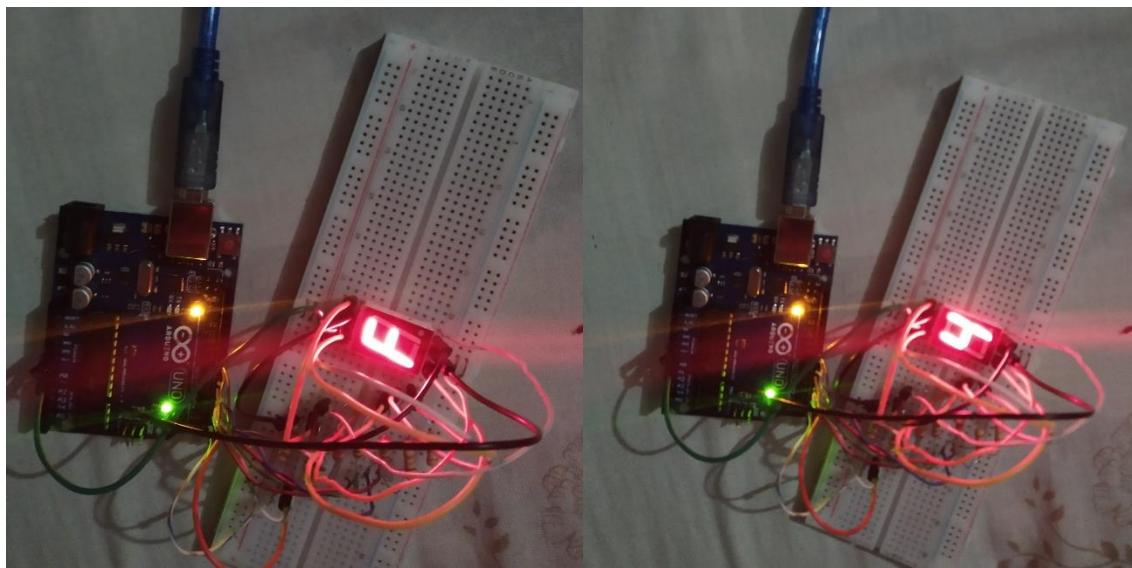


```
void loop() {
//Se llamará a cada método para que se realice lo indicado
    cero();
    uno();
    dos();
    tres();
    cuatro();
    cinco();
    seis();
    siete();
    ocho();
    nueve();
    AA();
    BB();
    CC();
    DD();
    EE();
    FF();
}
```

RESULTADOS:

En este proyecto se realizaon por medio de metodos cada una de las letras que se muestran en el display. Lo complicado fueron las conexiones entre los pines del display y del Arduino, para esto se debe tener en cuenta la ubicación de cada uno de los pines del display en el que debe ir la conexión.

ANEXOS:





PRACTICA 15: CONTADOR HEXADECIMAL DE UN BIT POR MEDIO DEL DISPLAY DE 7 SEGMENTOS CON UN PULSADOR

OBJETIVOS:

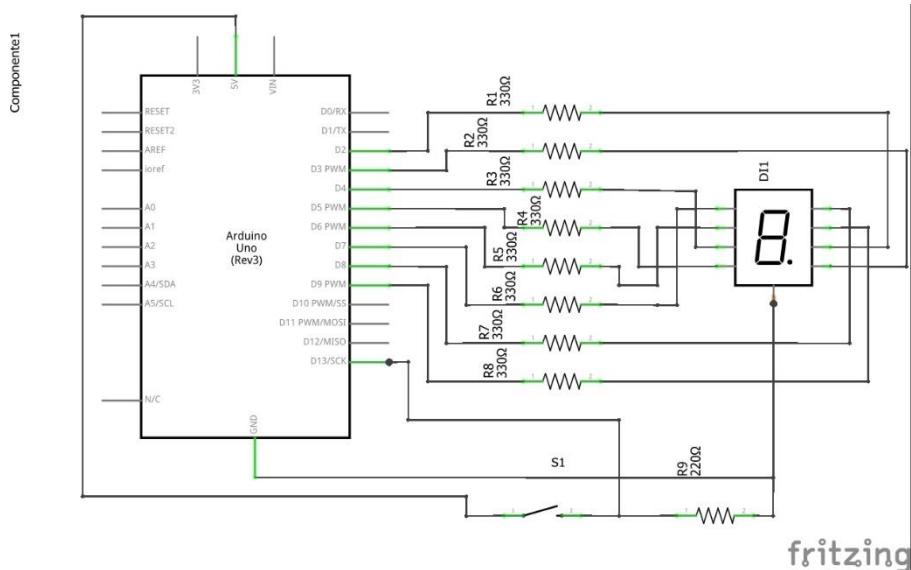
- Ejecutar el programa de modo que cada vez que se haga presente el pulsador se cambie de dato e el display
- Investigar sobre como se realiza la conexión de un pulsador
- Poner en practica los conocimeintos dados
- Conocer como funciona y como se conecta el pulsador.
- Ejecutar la programación mostrando un digito hexadecimal cada vez que se realice un pulsador.
- Poner en practica los conocimientos dados en las practicas anteriores.

MATERIALES:

- 8 Resistencias de $330\ \Omega$
- 1 Display de 7 segmentos
- 1 Arduino
- Varios cables Jumper
- Cable para conectar Arduino
- Computadora
- Programa Arduino
- 1 Protoboard
- 1 Pulsador
- 1 resistencia de $10\ K\Omega$

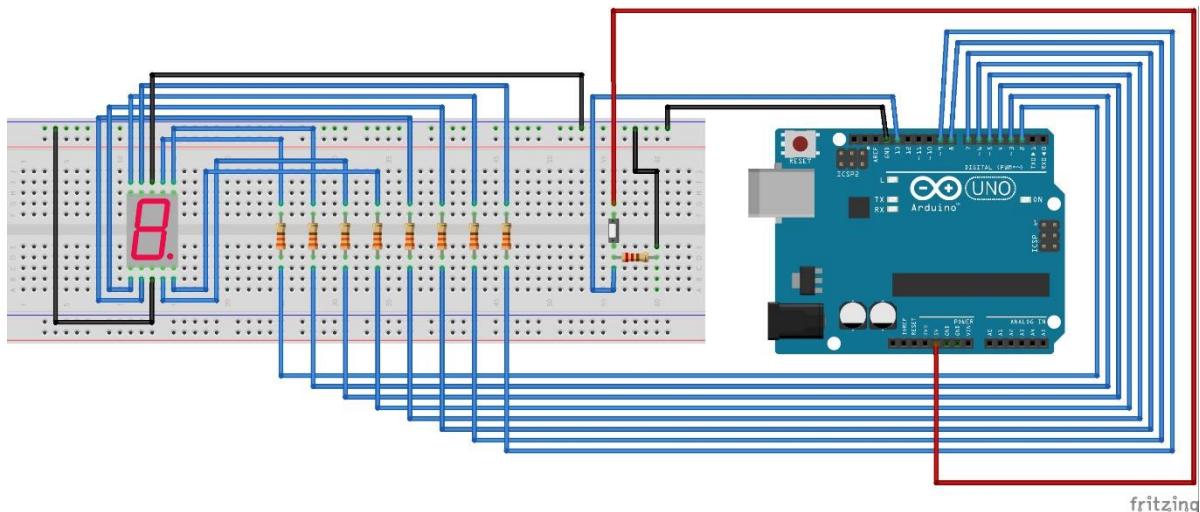


DIAGRAMA:



fritzing

ESQUEMA:



fritzing

PROGRAMACION:

```
/*
 * ULEAM-FACCI
 * TECNOLOGIA DE LA INFORMACION
 * SISTEMAS DIGITALES
 * MACIAS PICO JOSSELYN STEFANY
 *
 * ***** PROYECTO N° 4 *****
 *
 * _ NUMEROS HEXADECIMALES CON DISPLAY _
```



```
/*
int a= 3;
int b=4;
int c=5;
int dp=6;
int d=7;
int e=8;
int f=10;
int g=9;
int pulsador=2;
int pulso=0;

void setup() {
    pinMode( a, OUTPUT );
    pinMode( b, OUTPUT );
    pinMode( c, OUTPUT );
    pinMode( dp, OUTPUT );
    pinMode( d, OUTPUT );
    pinMode( e, OUTPUT );
    pinMode( f, OUTPUT );
    pinMode( g, OUTPUT );
    pinMode( a, LOW );
    pinMode( b, LOW );
    pinMode( c, LOW );
    pinMode( dp, LOW );
    pinMode( d, LOW );
    pinMode( e, LOW );
    pinMode( f, LOW );
    pinMode( g, LOW );
    pinMode(pulsador, INPUT_PULLUP);
    Serial.begin(9600);
    attachInterrupt(digitalPinToInterrupt(pulsador),conta,RISING);
}
void cero(){
    digitalWrite(a,HIGH );
    digitalWrite(b,HIGH );
    digitalWrite(c,HIGH );
    digitalWrite(d,HIGH );
    digitalWrite(e,HIGH );
    digitalWrite(f,HIGH );
    digitalWrite(g,LOW );
}
void uno(){
    digitalWrite(a,LOW );
    digitalWrite(b,HIGH );
    digitalWrite(c,HIGH );
    pinMode( d, LOW );
    pinMode( e, LOW );
    pinMode( f, LOW );
    pinMode( g, LOW );
}
void dos(){
    digitalWrite(a,HIGH );
    digitalWrite(b,HIGH );
```



```
    digitalWrite(g,HIGH );
    digitalWrite(e,HIGH );
    digitalWrite(d,HIGH );
    digitalWrite(c,LOW );
    digitalWrite(f,LOW );
}

void tres(){
    digitalWrite(a,HIGH );
    digitalWrite(b,HIGH );
    digitalWrite(g,HIGH );
    digitalWrite(c,HIGH );
    digitalWrite(d,HIGH );
    digitalWrite(e,LOW );
    digitalWrite(f,LOW );
}

void cuatro(){
    digitalWrite(f,HIGH );
    digitalWrite(g,HIGH );
    digitalWrite(b,HIGH );
    digitalWrite(c,HIGH );
    digitalWrite(a,LOW );
    digitalWrite(d,LOW );
    digitalWrite(e,LOW );
}

void cinco(){
    digitalWrite (a,HIGH );
    digitalWrite (f,HIGH );
    digitalWrite (g,HIGH );
    digitalWrite (c,HIGH );
    digitalWrite (d,HIGH );
    digitalWrite (b,LOW );
    digitalWrite (e,LOW );
}

void seis(){
    digitalWrite (a,HIGH );
    digitalWrite (f,HIGH );
    digitalWrite (g,HIGH );
    digitalWrite (e,HIGH );
    digitalWrite (d,HIGH );
    digitalWrite (c,HIGH );
    digitalWrite (b,LOW );
}

void siete(){
    digitalWrite(a,HIGH );
    digitalWrite(b,HIGH );
    digitalWrite(c,HIGH );
    digitalWrite(d,LOW );
    digitalWrite(e,LOW );
    digitalWrite(f,LOW );
    digitalWrite(g,LOW );
}

void ocho(){
    digitalWrite(a,HIGH );
    digitalWrite(b,HIGH );
```



```
digitalWrite(c,HIGH );
digitalWrite(g,HIGH );
digitalWrite(d,HIGH );
digitalWrite(e,HIGH );
digitalWrite(f,HIGH );
}
void nueve(){
digitalWrite(a,HIGH );
digitalWrite(b,HIGH );
digitalWrite(c,HIGH );
digitalWrite(g,HIGH );
digitalWrite(d,HIGH );
digitalWrite(f,HIGH );
digitalWrite(e,LOW );
}
void AA(){
digitalWrite(a,HIGH );
digitalWrite(b,HIGH );
digitalWrite(c,HIGH );
digitalWrite(g,HIGH );
digitalWrite(e,HIGH );
digitalWrite(f,HIGH );
digitalWrite(d,LOW );
}
void BB(){
digitalWrite(a,LOW );
digitalWrite(b,LOW );
digitalWrite(c,HIGH );
digitalWrite(g,HIGH );
digitalWrite(d,HIGH );
digitalWrite(e,HIGH );
digitalWrite(f,HIGH );
}
void CC(){
digitalWrite(a,HIGH );
digitalWrite(d,HIGH );
digitalWrite(e,HIGH );
digitalWrite(f,HIGH );
digitalWrite(b,LOW );
digitalWrite(c,LOW );
digitalWrite(g,LOW );
}
void DD(){
digitalWrite(a,LOW );
digitalWrite(b,HIGH );
digitalWrite(c,HIGH );
digitalWrite(d,HIGH );
digitalWrite(e,HIGH );
digitalWrite(f,LOW );
digitalWrite(g,HIGH );
}
void EE(){
digitalWrite(a,HIGH );
digitalWrite(g,HIGH );
```



```
digitalWrite(d,HIGH );
digitalWrite(e,HIGH );
digitalWrite(f,HIGH );
digitalWrite(b,LOW );
digitalWrite(c,LOW );
}
void FF(){
digitalWrite(a,HIGH );
digitalWrite(g,HIGH );
digitalWrite(e,HIGH );
digitalWrite(f,HIGH );
digitalWrite(b,LOW );
digitalWrite(c,LOW );
digitalWrite(d,LOW );
}
void conta(){
pulso++;
Serial.print( "numero de veces presionado" );
Serial.println(pulso);
delay(100);
}
void loop() {
/*if ( digitalRead(11)==1) {
pulso++;
if (pulso==17) {
pulso=0;
} */
if (pulso==1){
cero();
}
if (pulso==2){
uno();
}
if (pulso==3){
dos();
}
if (pulso==4){
tres();
}
if (pulso==5){
cuatro();
}
if (pulso==6){
cinco();
}
if (pulso==7){
seis();
}
if (pulso==8){
siete();
}
if (pulso==9){
ocho();
}
}
```

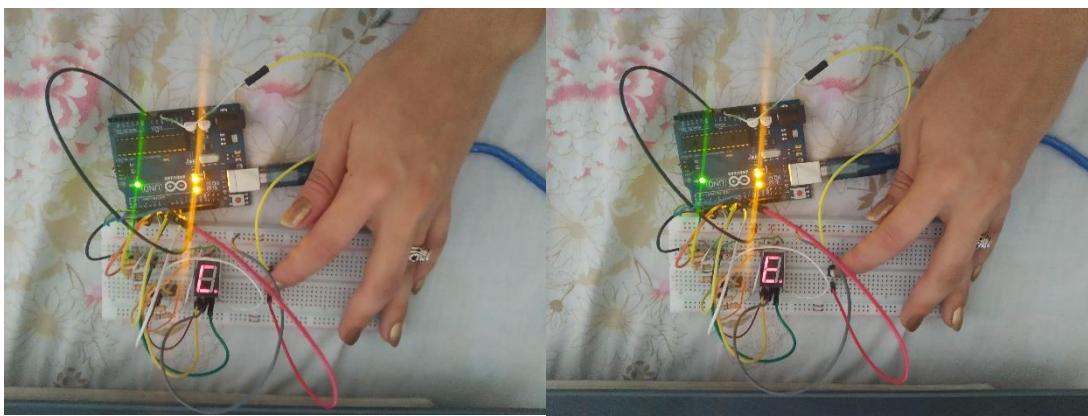


```
if (pulso==10) {
    nueve();
}
if (pulso==11) {
    AA();
}
if (pulso==12) {
    BB();
}
if (pulso==13) {
    CC();
}
if (pulso==14) {
    DD();
}
if (pulso==15) {
    EE();
}
if (pulso==16) {
    FF();
}
}
```

RESULTADOS:

Para este proyecto hemos utilizado la estructura del circuito 3 realizando un método para cada cambio de luces y luego utilizando un método if para cada una de las pulsaciones, en este conocimos dos métodos nuevos que son donde se utiliza INPUT_PULLUP y el de las interrupciones incluyendo el RISING. Para no obtener problemas con las conexiones se necesita ubicar bien las resistencias y los cables e incluso los leds identificar su parte negativa y su parte positiva.

ANEXOS:





PRACTICA 16: MOSTRAR LOS NUMEROS HEXADECIMALES CON REGISTRO DE DESPLAZAMIENTO

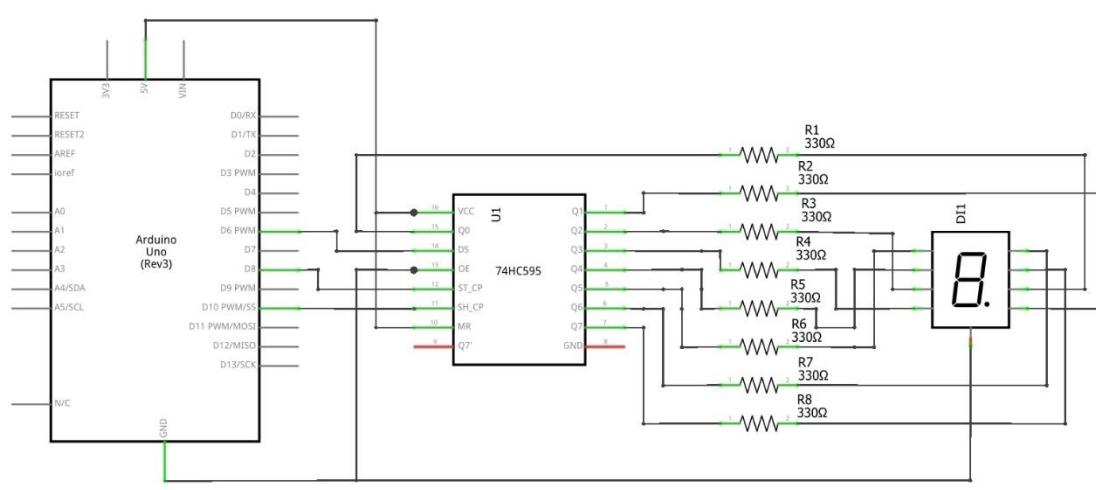
OBJETIVOS:

- Realizar la conexión para poder mostrar los números hexadecimales mediante el tiempo que el usuario desee.
- Conocer las conexiones con las que trabaja el display de siete segmentos
- Poner en práctica los conocimientos obtenidos.
- Conocer el funcionamiento de la resistencia de $560\ \Omega$.

MATERIALES:

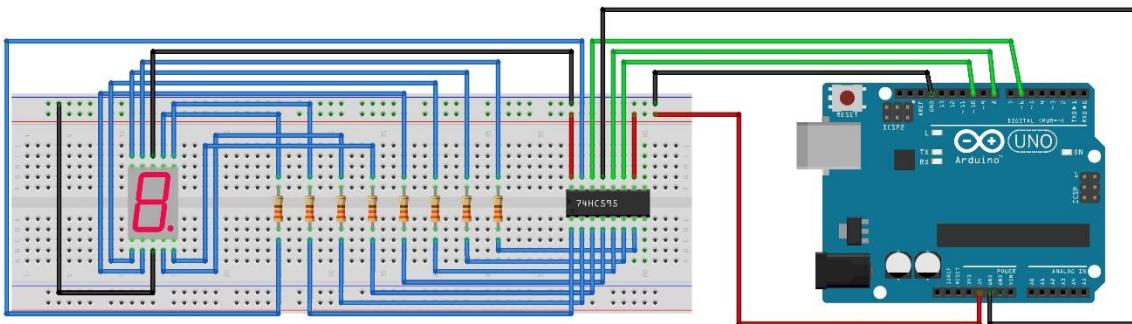
- Un Arduino Uno
- Un circuito integrado 74HC595
- 8 resistencias de 330Ω
- Varios cables jumper
- Cable para conectar arduino
- 1 protoboard
- 1 display de 7 segmentos
- Un computador

DIAGRAMA:





ESQUEMA:



PROGRAMACION:

```
/*
 *          ULEAM-FACCI
 *      TECNOLOGIA DE LA INFORMACION
 *      SISTEMAS DIGITALES
 *  MACIAS PICO JOSSELYN STEFANY
 *
 * ***** PROYECTO N° 5 *****
 *
 * _ NUMEROS HEXADECIMALES CON DISPLAY _
 */
#define latch_pin 8
#define data_pin 6
#define clock_pin 10
//SE HABILITAN PINES COMO SALIDA
void setup () {
    for (int n=6; n<11; n++) pinMode(n,OUTPUT);
}

void loop () {
//SECUENCIA 0 A 256 PARA ENVIAR DATOS POR PIN 6
    for(int sec=0; sec<17;sec++) {
        digitalWrite(latch_pin,HIGH);
        //FUNCION PARA ENVIO DE DATOS AL SHIFT REGISTER
        shiftOut(data_pin,clock_pin, MSBFIRST, animacion(sec) );
        delay(1000);
        digitalWrite(latch_pin,LOW);
    }
}
//ANIMACIÓN PARA PRENDER LEDS DE DISPLAY EN FORMA DE INFINITO
int animacion (int sec_ ){
    int frame =0;
    //SE PRENDE UN LED A LA VEZ
    switch(sec_){
        case 0: frame = B10111111;
                  break;
        case 1: frame = B10000110;
```



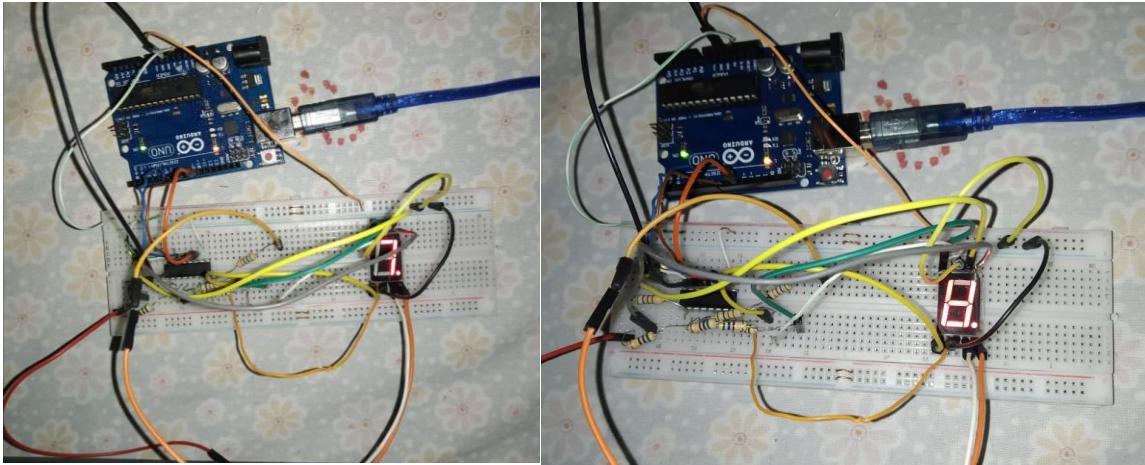
```
        break;
    case 2: frame = B11011011;
        break;
    case 3: frame = B11001111;
        break;
    case 4: frame = B11100110;
        break;
    case 5: frame = B11101101;
        break;
    case 6: frame = B11111101;
        break;
    case 7: frame = B10000111;
        break;
    case 8: frame = B11111111;
        break;
    case 9: frame = B11101111;
        break;
    case 10: frame =B11110111;
        break;
    case 11: frame =B11111100;
        break;
    case 12: frame =B10111001;
        break;
    case 13: frame =B11011110;
        break;
    case 14: frame = B11111001;
        break;
    case 15: frame = B11110001;
        break;
    }
    return frame;
}
```

RESULTADOS:

Como resultado podemos darnos cuenta que la única diferencia es que hemos conectado el display a un circuito integrado para así solo poder conectar tres pines al Arduino y de el circuito integrado sacar 8 pines para cada lado del display de 7 segmentos y esto se ha realizado por medio de un switch para así poder prender cada uno de los LEDs en forma de infinito. Nos damos cuenta que en cada uno de los casos a 8 bits cada uno de ellos va hacia un pin del circuito integrado es así como sabríamos cuáles pines están encendidos y cuáles apagados.



ANEXOS:



PRACTICA 17: MOSTRAR LOS DATOS DEL 00 AL 99 AUTOMATICAMENTE.

OBJETIVOS:

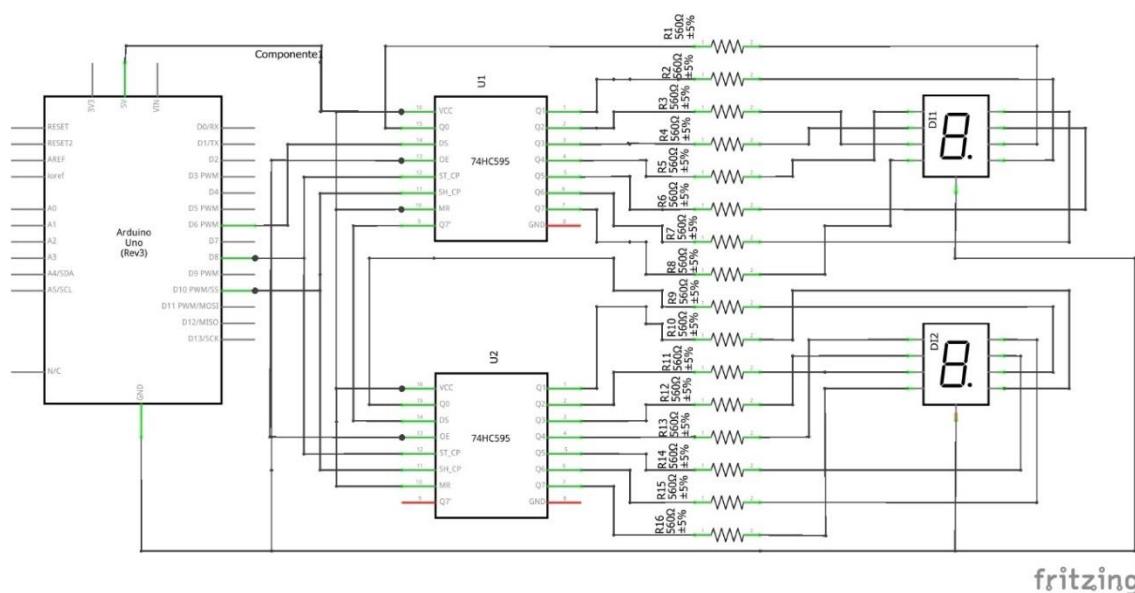
- Poner en practica los conocimientos obtenidos anteriormente.
- Conocer las conexiones para poder conectar dos circuitos integrados y que realicen la misma función
- Investigar acerca de los pines de cada display de 7 segmentos

MATERIALES:

- 2 display de 7 segmentos
- 2 circuitos integrados 74HC595
- 16 resistencias de $560\ \Omega$
- Varios cables jumper
- 1 protoboard
- 1 Arduino Uno
- Cable para conectar arduino
- Un computador

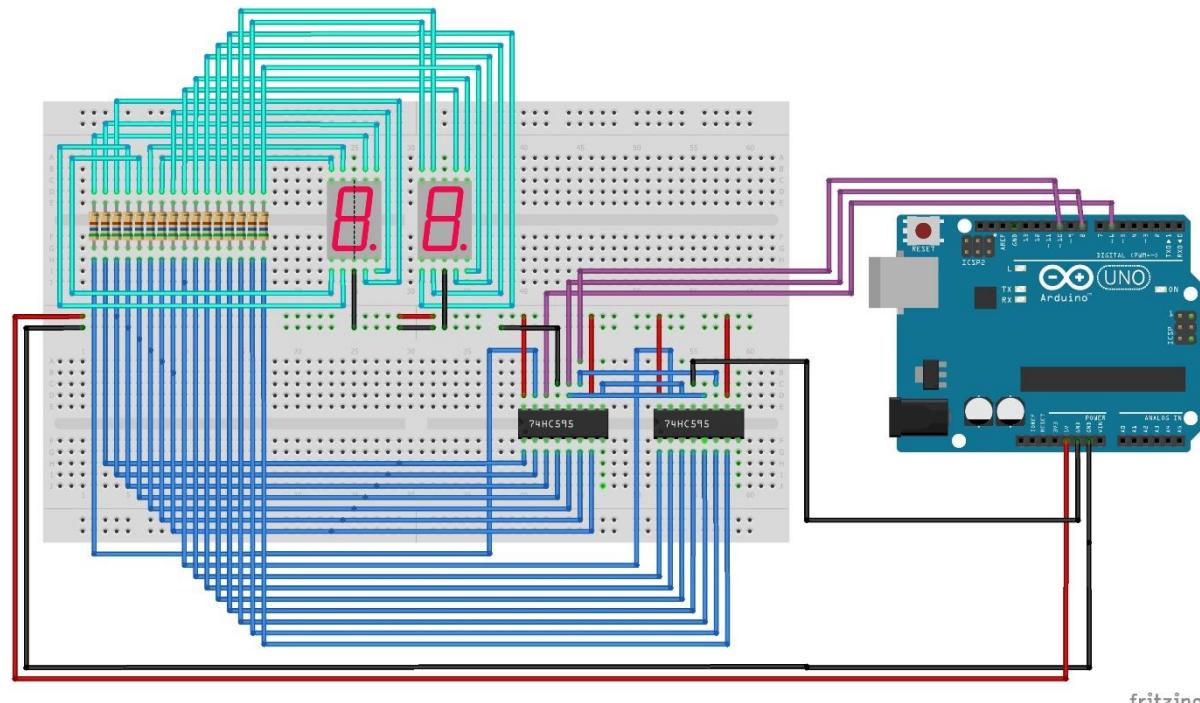


DIAGRAMA:



fritzing

ESQUEMA:



fritzing

PROGRAMACION:

```
/*
 * 
 * ULEAM-FACCI
 * TECNOLOGIA DE LA INFORMACION
 * SISTEMAS DIGITALES
 * MACIAS PICO JOSSELYN STEFANY
 *
```



* ***** PROYECTO N° 6*****

*

* _____ NUMEROS DEL 00 AL 99 _____

*

*/

```
# define latch_pin 8
#define data_pin 6
#define clock_pin 10
//SE HABILITAN PINES COMO SALIDA
void setup () {
    for (int n=6; n<11; n++) pinMode(n,OUTPUT);
}

void loop () {
//SECUENCIA 0 A 256 PARA ENVIAR DATOS POR PIN 6
    for(int sec=0; sec<10;sec++) {
        for(int sec1=0; sec1<10; sec1++){
            digitalWrite(latch_pin,HIGH);
            //FUNCION PARA ENVIO DE DATOS AL SHIFT REGISTER
            shiftOut(data_pin,clock_pin, MSBFIRST, animacion(sec) );
            shiftOut(data_pin,clock_pin, MSBFIRST, animacion(sec1) );
            delay(500);
            digitalWrite(latch_pin,LOW);
        }
    }
}
//ANIMACIÓN PARA PRENDER LEDS DE DISPLAY EN FORMA DE INFINITO
int animacion (int sec_ ){
    int frame =0;
    //SE PRENDE UN LED A LA VEZ
    switch(sec_){
        case 0: frame = B10111111;
                  break;
        case 1: frame = B10000110;
                  break;
        case 2: frame = B11011011;
                  break;
        case 3: frame = B11001111;
                  break;
        case 4: frame = B11100110;
                  break;
        case 5: frame = B11101101;
                  break;
        case 6: frame = B11111101;
                  break;
        case 7: frame = B10000111;
                  break;
        case 8: frame = B11111111;
                  break;
        case 9: frame = B11101111;
                  break;
    }
}
```

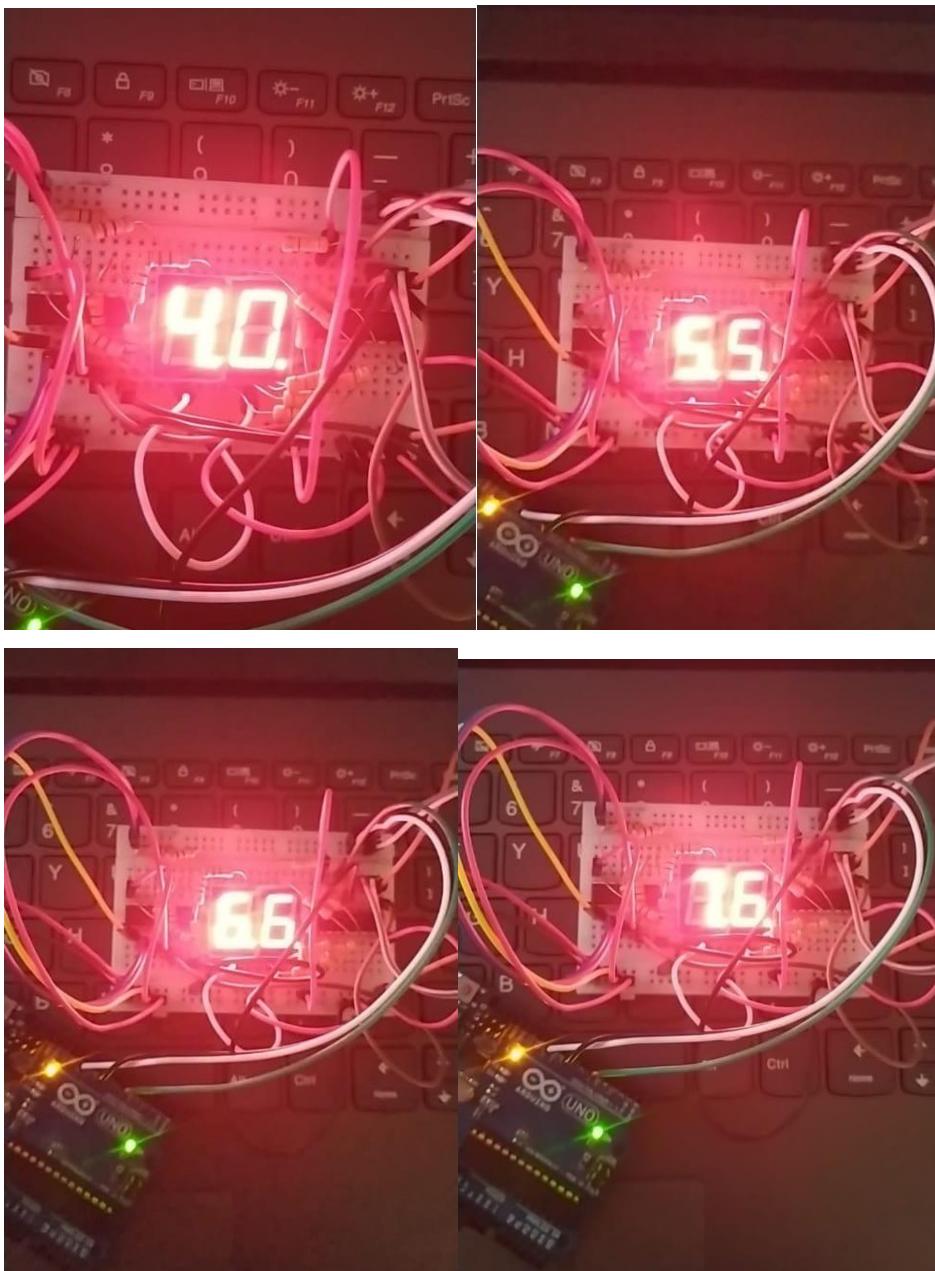


```
    return frame;  
}
```

RESULTADOS:

Para concluir en este programa el único cambio que se hizo fue aumentar dos `for` en la parte del loop para que se enciendan ambos displays incluyendo los registros de desplazamiento y es para eso lo cual se utilizan el `MSBFIRST` y el `LBSFIRST` en cada uno de estos registros de desplazamiento se llama al método de animación cada uno tiene una variable diferente ya que uno es para el primer display y el otro para el segundo.

ANEXOS:





PRACTICA 18: CONECTAR UNA MATRIZ LED 8 X 8 CON REGISTRO DE DESPLAZAMIENTO.

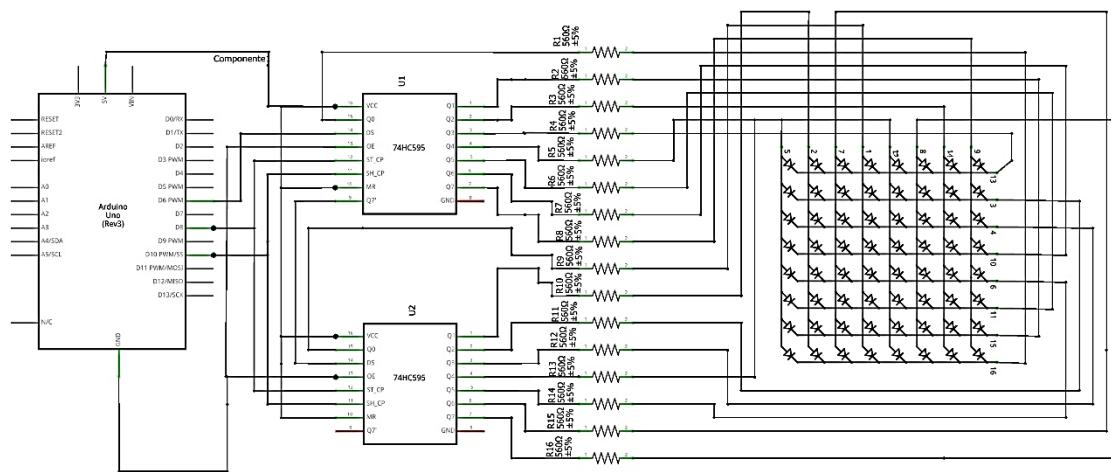
OBJETIVO:

- Conocer el funcionamiento de una matriz 8 X 8
- Investigar sobre las conexiones de una matriz 8 x 8
- Investigar sobre como realizar desplazamientos dentro de la matriz
- Ejecutar el código para que mediante la conexión realizada pueda mostrar su nombre, incluyendo su desplazamiento.

MATERIALES:

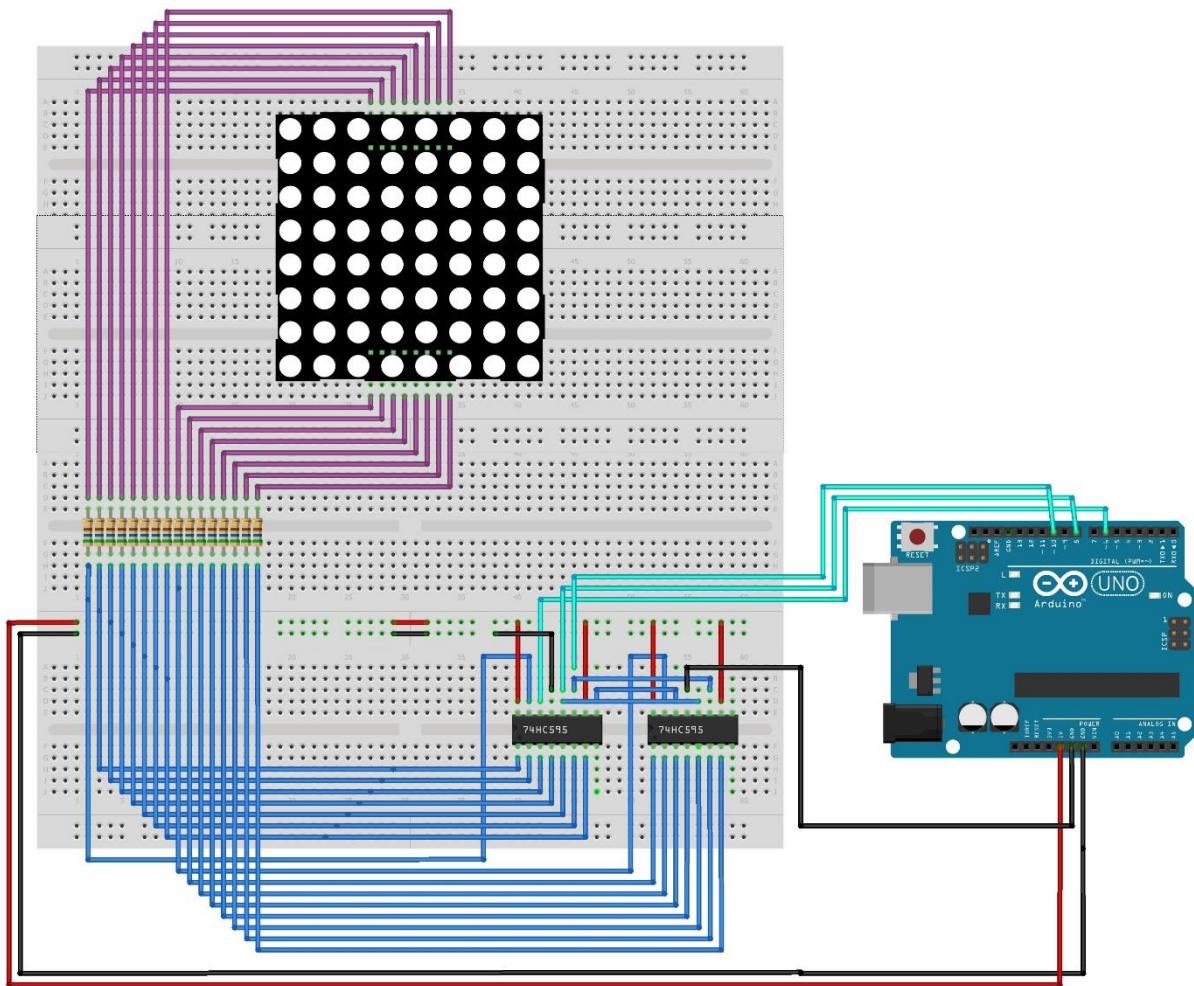
- Una matriz 8 x 8
- Cables jumper
- Un Arduino uno
- 2 circuitos integrados 74HC595
- 16 Resistencias de $560\ \Omega$
- Cable de arduino
- Un computador
- Dos protoboard

DIAGRAMA:





ESQUEMA:



PROGRAMACION:

```
/*
 *      ULEAM-FACCI
 *      TECNOLOGIA DE LA INFORMACION
 *      SISTEMAS DIGITALES
 *      MACIAS PICO JOSSELYN STEFANY
 *
 *      *TAREA*
 *
 ** POROYECTO N°7 **
 *
 *      ____ NOMBRE CON MATRIZ 8 x 8 __
 *
 *      PARA TAREA COMENTAR TODO EL CODIGO
 */
/*Realice la conexión del esquema que se muestra en el circuito No.
5;
```



```
* y ejecute la programación de tal manera que en la matriz de led
8x8
* se pueda mostrar un mensaje de texto (en este caso, su nombre
completo).
*/
//Se definen los pines
#define latch_pin 8
#define data_pin 6
#define clock_pin 10
//Se definen las variables
int a, hold, shift;
//Se define el retardo
int retardo=100;
//Se define el total de los numeros binarios a contar
int binary[] = {254, 253, 251, 247, 239, 223, 191, 127};//ESTO LO
MODIFIQUE/

//Se realiza un metodo para cada letra en una respectiva matriz
byte LetraJ []={
    B00000000,
    B01111110,
    B01111110,
    B00011000,
    B00011000,
    B01111000,
    B01111000,
    B00000000
};

byte LetraO []={

    B00000000,
    B01111110,
    B01111110,
    B01100110,
    B01100110,
    B01111110,
    B01111110,
    B00000000
};

byte LetraS []={

    B00000000,
    B00111100,
    B00100000,
    B00111100,
    B00000100,
    B00000100,
    B00111100,
    B00000000
};

byte LetraE []= {
```



```
B00000000,  
B00000000,  
B00111100,  
B00100000,  
B00111100,  
B00100000,  
B00111100,  
B00000000  
};  
  
byte LetraL []={  
  
    B00000000,  
    B01100000,  
    B01100000,  
    B01100000,  
    B01100000,  
    B01111100,  
    B01111100,  
    B00000000  
};  
  
byte LetraY []={  
  
    B00000000,  
    B00010100,  
    B00010100,  
    B00011100,  
    B00001000,  
    B00001000,  
    B00001000,  
    B00000000  
};  
  
byte LetraN []={  
  
    B00000000,  
    B01000010,  
    B01100010,  
    B01010010,  
    B01001010,  
    B01000110,  
    B01000010,  
    B00000000  
};  
  
byte LetraT []={  
  
    B00000000,  
    B01111110,  
    B01111110,  
    B00011000,  
    B00011000,  
    B00011000,  
    B00011000,
```



```
B00000000
};

byte LetraF []={

    B00000000,
    B00111100,
    B00100000,
    B00111100,
    B00100000,
    B00100000,
    B00100000,
    B00000000
};

byte LetraA []={

    B00000000,
    B00111100,
    B00100100,
    B00100100,
    B00111100,
    B00100100,
    B00100100,
    B00000000
};

byte LetraM []={

    B00000000,
    B01000100,
    B01101100,
    B01010100,
    B01000100,
    B01000100,
    B01000100,
    B00000000
};

byte LetraC []={

    B00000000,
    B00111100,
    B00100000,
    B00100000,
    B00100000,
    B00100000,
    B00111100,
    B00000000
};

byte LetraI []={

    B00000000,
    B01111100,
    B00010000,
    B00010000,
```



```
B00010000,  
B00010000,  
B01111100,  
B00000000  
};  
  
byte LetraP []={  
  
    B00000000,  
    B00111100,  
    B00100100,  
    B00100100,  
    B00111100,  
    B00100000,  
    B00100000,  
    B00000000  
};  
  
//Se realiza un metodo para que pueda mostrar las letras dentro de  
la matriz  
void mostrarA(){  
    //El primer for es para controlar los leds de la matriz por  
columna  
    for(shift=0; shift<8; shift++){  
        //El segundo for es para controlar los leds de la matriz por  
fila  
        for(holder=0; holder<100; holder++){  
            //El cuarto for es para que avancen los bits que se mostraran  
            for(a=0;a<8;a++){  
                //Esto es para que la matriz comience apagada  
                digitalWrite(latch_pin, LOW);  
                //Esta desplaza las filas desde el bit menos significativo  
                primero  
                shiftOut(data_pin, clock_pin, LSBFIRST, binary[a]);  
                //Esta parte desplaza las columnas desde el bit menos  
significativo primero en esta se llama a la matriz que contiene su  
letra  
                shiftOut(data_pin, clock_pin, MSBFIRST, LetraA[a]<<shift);  
                //Y esta es para que muestre la letra  
                digitalWrite(latch_pin, HIGH);  
            }  
        }  
    }  
}  
void mostrarC(){  
    //El primer for es para controlar los leds de la matriz por columna  
    for(shift=0; shift<8; shift++){  
        //El segundo for es para controlar los leds de la matriz por  
fila  
        for(holder=0; holder<100; holder++){  
            //El cuarto for es para que avancen los bits que se mostraran  
            for(a=0;a<8;a++){  
                //Esto es para que la matriz comience apagada  
                digitalWrite(latch_pin, LOW);  
            }  
        }  
    }  
}
```



```
//Esta desplaza las filas desde el bit menos significativo
primero
    shiftOut(data_pin, clock_pin, LSBFIRST, binary[a]);
    //Esta parte desplaza las columnas desde el bit menos
significativo primero en esta se llama a la matriz que contiene su
letra
    shiftOut(data_pin, clock_pin, MSBFIRST, LetraC[a]<<shift);
    //Y esta es para que muestre la letra
    digitalWrite(latch_pin, HIGH);
}
}
}
}

void mostrarE(){
//El primer for es para controlar los leds de la matriz por columna
for(shift=0; shift<8; shift++){
    //El segundo for es para controlar los leds de la matriz por
fila
    for(hold=0; hold<100; hold++){
        //El cuarto for es para que avancen los bits que se mostraran
        for(a=0;a<8;a++){
            //Esto es para que la matriz comience apagada
            digitalWrite(latch_pin, LOW);
            //Esta desplaza las filas desde el bit menos significativo
            primero
                shiftOut(data_pin, clock_pin, LSBFIRST, binary[a]);
                //Esta parte desplaza las columnas desde el bit menos
significativo primero en esta se llama a la matriz que contiene su
letra
                shiftOut(data_pin, clock_pin, MSBFIRST, LetraE[a]<<shift);
                //Y esta es para que muestre la letra
                digitalWrite(latch_pin, HIGH);
            }
        }
    }
}

void mostrarF(){
    //El primer for es para controlar los leds de la matriz por columna
    for(shift=0; shift<8; shift++){
        //El segundo for es para controlar los leds de la matriz por
fila
        for(hold=0; hold<100; hold++){
            //El cuarto for es para que avancen los bits que se mostraran
            for(a=0;a<8;a++){
                //Esto es para que la matriz comience apagada
                digitalWrite(latch_pin, LOW);
                //Esta desplaza las filas desde el bit menos significativo
                primero
                    shiftOut(data_pin, clock_pin, LSBFIRST, binary[a]);
                    //Esta parte desplaza las columnas desde el bit menos
significativo primero en esta se llama a la matriz que contiene su
letra
                    shiftOut(data_pin, clock_pin, MSBFIRST, LetraF[a]<<shift);
                    //Y esta es para que muestre la letra
            }
        }
    }
}
```



```
        digitalWrite(latch_pin, HIGH);
    }
}
}

void mostrarI(){
    //El primer for es para controlar los leds de la matriz por
columna
    for(shift=0; shift<8; shift++){
        //El segundo for es para controlar los leds de la matriz por
fila
        for(holder=0; holder<100; holder++){
            //El cuarto for es para que avancen los bits que se mostraran
            for(a=0;a<8;a++){
                //Esto es para que la matriz comience apagada
                digitalWrite(latch_pin, LOW);
                //Esta desplaza las filas desde el bit menos significativo
                primero
                shiftOut(data_pin, clock_pin, LSBFIRST, binary[a]);
                //Esta parte desplaza las columnas desde el bit menos
                significativo primero en esta se llama a la matriz que contiene su
                letra
                shiftOut(data_pin, clock_pin, MSBFIRST, LetraI[a]<<shift);
                //Y esta es para que muestre la letra
                digitalWrite(latch_pin, HIGH);
            }
        }
    }
}

void mostrarJ(){
    //El primer for es para controlar los leds de la matriz por
columna
    for(shift=0; shift<8; shift++){
        //El segundo for es para controlar los leds de la matriz por
fila
        for(holder=0; holder<100; holder++){
            //El cuarto for es para que avancen los bits que se mostraran
            for(a=0;a<8;a++){
                //Esto es para que la matriz comience apagada
                digitalWrite(latch_pin, LOW);
                //Esta desplaza las filas desde el bit menos significativo
                primero
                shiftOut(data_pin, clock_pin, LSBFIRST, binary[a]);
                //Esta parte desplaza las columnas desde el bit menos
                significativo primero en esta se llama a la matriz que contiene su
                letra
                shiftOut(data_pin, clock_pin, MSBFIRST, LetraJ[a]<<shift);
                //Y esta es para que muestre la letra
                digitalWrite(latch_pin, HIGH);
            }
        }
    }
}

void mostrarL(){
```



```
//El primer for es para controlar los leds de la matriz por columna
for(shift=0; shift<8; shift++){
    //El segundo for es para controlar los leds de la matriz por
    fila
    for(hold=0; hold<100; hold++){
        //El cuarto for es para que avancen los bits que se mostraran
        for(a=0;a<8;a++){
            //Esto es para que la matriz comience apagada
            digitalWrite(latch_pin, LOW);
            //Esta desplaza las filas desde el bit menos significativo
            primero
            shiftOut(data_pin, clock_pin, LSBFIRST, binary[a]);
            //Esta parte desplaza las columnas desde el bit menos
            significativo primero en esta se llama a la matriz que contiene su
            letra
            shiftOut(data_pin, clock_pin, MSBFIRST, LetraL[a]<<shift);
            //Y esta es para que muestre la letra
            digitalWrite(latch_pin, HIGH);
        }
    }
}
void mostrarM(){
    //El primer for es para controlar los leds de la matriz por
    columna
    for(shift=0; shift<8; shift++){
        //El segundo for es para controlar los leds de la matriz por
        fila
        for(hold=0; hold<100; hold++){
            //El cuarto for es para que avancen los bits que se mostraran
            for(a=0;a<8;a++){
                //Esto es para que la matriz comience apagada
                digitalWrite(latch_pin, LOW);
                //Esta desplaza las filas desde el bit menos significativo
                primero
                shiftOut(data_pin, clock_pin, LSBFIRST, binary[a]);
                //Esta parte desplaza las columnas desde el bit menos
                significativo primero en esta se llama a la matriz que contiene su
                letra
                shiftOut(data_pin, clock_pin, MSBFIRST, LetraM[a]<<shift);
                //Y esta es para que muestre la letra
                digitalWrite(latch_pin, HIGH);
            }
        }
    }
}
void mostrarN(){
    //El primer for es para controlar los leds de la matriz por columna
    for(shift=0; shift<8; shift++){
        //El segundo for es para controlar los leds de la matriz por
        fila
        for(hold=0; hold<100; hold++){
            //El cuarto for es para que avancen los bits que se mostraran
            for(a=0;a<8;a){
```



```
//Esto es para que la matriz comience apagada
digitalWrite(latch_pin, LOW);
//Esta desplaza las filas desde el bit menos significativo
primero
    shiftOut(data_pin, clock_pin, LSBFIRST, binary[a]);
    //Esta parte desplaza las columnas desde el bit menos
significativo primero en esta se llama a la matriz que contiene su
letra
    shiftOut(data_pin, clock_pin, MSBFIRST, LetraN[a]<<shift);
    //Y esta es para que muestre la letra
    digitalWrite(latch_pin, HIGH);
}
}
}
}

void mostrarO(){
    //El primer for es para controlar los leds de la matriz por
columna
    for(shift=0; shift<8; shift++){
        //El segundo for es para controlar los leds de la matriz por
fila
        for(hold=0; hold<100; hold++){
            //El cuarto for es para que avancen los bits que se mostraran
            for(a=0;a<8;a++){
                //Esto es para que la matriz comience apagada
                digitalWrite(latch_pin, LOW);
                //Esta desplaza las filas desde el bit menos significativo
primero
                    shiftOut(data_pin, clock_pin, LSBFIRST, binary[a]);
                    //Esta parte desplaza las columnas desde el bit menos
significativo primero en esta se llama a la matriz que contiene su
letra
                    shiftOut(data_pin, clock_pin, MSBFIRST, LetraO[a]<<shift);
                    //Y esta es para que muestre la letra
                    digitalWrite(latch_pin, HIGH);
                }
            }
        }
    }
}

void mostrarP(){
    //El primer for es para controlar los leds de la matriz por
columna
    for(shift=0; shift<8; shift++){
        //El segundo for es para controlar los leds de la matriz por
fila
        for(hold=0; hold<100; hold++){
            //El cuarto for es para que avancen los bits que se mostraran
            for(a=0;a<8;a++){
                //Esto es para que la matriz comience apagada
                digitalWrite(latch_pin, LOW);
                //Esta desplaza las filas desde el bit menos significativo
primero
                    shiftOut(data_pin, clock_pin, LSBFIRST, binary[a]);
```



```
//Esta parte desplaza las columnas desde el bit menos significativo primero en esta se llama a la matriz que contiene su letra
    shiftOut(data_pin, clock_pin, MSBFIRST, LetraP[a]<<shift);
    //Y esta es para que muestre la letra
    digitalWrite(latch_pin, HIGH);
}
}
}
}

void mostrarS(){
    //El primer for es para controlar los leds de la matriz por columna
    for(shift=0; shift<8; shift++){
        //El segundo for es para controlar los leds de la matriz por fila
        for(holder=0; holder<100; holder++){
            //El cuarto for es para que avancen los bits que se mostraran
            for(a=0;a<8;a++){
                //Esto es para que la matriz comience apagada
                digitalWrite(latch_pin, LOW);
                //Esta desplaza las filas desde el bit menos significativo primero
                shiftOut(data_pin, clock_pin, LSBFIRST, binary[a]);
                //Esta parte desplaza las columnas desde el bit menos significativo primero en esta se llama a la matriz que contiene su letra
                shiftOut(data_pin, clock_pin, MSBFIRST, LetraS[a]<<shift);
                //Y esta es para que muestre la letra
                digitalWrite(latch_pin, HIGH);
            }
        }
    }
}

void mostrarT(){
    //El primer for es para controlar los leds de la matriz por columna
    for(shift=0; shift<8; shift++){
        //El segundo for es para controlar los leds de la matriz por fila
        for(holder=0; holder<100; holder++){
            //El cuarto for es para que avancen los bits que se mostraran
            for(a=0;a<8;a++){
                //Esto es para que la matriz comience apagada
                digitalWrite(latch_pin, LOW);
                //Esta desplaza las filas desde el bit menos significativo primero
                shiftOut(data_pin, clock_pin, LSBFIRST, binary[a]);
                //Esta parte desplaza las columnas desde el bit menos significativo primero en esta se llama a la matriz que contiene su letra
                shiftOut(data_pin, clock_pin, MSBFIRST, LetraT[a]<<shift);
                //Y esta es para que muestre la letra
                digitalWrite(latch_pin, HIGH);
            }
        }
    }
}
```



```
        }
    }
}

void mostrarY(){
//El primer for es para controlar los leds de la matriz por columna
for(shift=0; shift<8; shift++){
//El segundo for es para controlar los leds de la matriz por
fila
    for(hold=0; hold<100; hold++){
        //El cuarto for es para que avancen los bits que se mostraran
        for(a=0;a<8;a++){
            //Esto es para que la matriz comience apagada
            digitalWrite(latch_pin, LOW);
            //Esta desplaza las filas desde el bit menos significativo
            primero
                shiftOut(data_pin, clock_pin, LSBFIRST, binary[a]);
            //Esta parte desplaza las columnas desde el bit menos
            significativo primero en esta se llama a la matriz que contiene su
            letra
                shiftOut(data_pin, clock_pin, MSBFIRST, LetraY[a]<<shift);
            //Y esta es para que muestre la letra
            digitalWrite(latch_pin, HIGH);
        }
    }
}
}

void setup() {
//En esta muestra cuales son los pines de salida
pinMode(latch_pin,OUTPUT); //cuales seran los pines de salida
pinMode(data_pin,OUTPUT); //cuales seran los pines de salida
pinMode(clock_pin,OUTPUT); //cuales seran los pines de salida
}

void loop() {
//En esta se llama a cada metodo para mostrar ya en fisico
mostrarJ();
delay(retardo);
mostrarO();
delay(retardo);
mostrarS();
delay(retardo);
mostrarS();
delay(retardo);
mostrarE();
delay(retardo);
mostrarL();
delay(retardo);
mostrarY();
delay(retardo);
mostrarN();
delay(retardo);
mostrarS();
delay(retardo);
mostrarT();
delay(retardo);
```



```
mostrarE();  
delay(retardo);  
mostrarF();  
delay(retardo);  
mostrarA();  
delay(retardo);  
mostrarN();  
delay(retardo);  
mostrarY();  
delay(retardo);  
mostrarM();  
delay(retardo);  
mostrarA();  
delay(retardo);  
mostrarC();  
delay(retardo);  
mostrarI();  
delay(retardo);  
mostrarA();  
delay(retardo);  
mostrarP();  
delay(retardo);  
mostrarI();  
delay(retardo);  
mostrarC();  
delay(retardo);  
mostrarO();  
delay (retardo);  
}
```

RESULTADOS:

En esta ultima practica se puede concluir que se aprendio de todo un poco, lo cual fue de mucha ayuda para el ultimo código de programación, en este se crearon métodos de byte diferentes donde cada uno de estos contiene una letra lo cual es llamado por un nuevo método el cual se denomina como mostrar seguido de la letra que desea. Para finalizar se llama al método de mostrar dentro del loop seguido de un delay para que no se muestren todos juntos.