

# Documentación de Código y Esquema Electrónico

---

## Documentación de trabajo

TCU-629-Ver1.0

### Resumen

El presente documento muestra la documentación relacionada al circuito electrónico diseñado para el funcionamiento del dispensador, en este documento se presenta tanto el diagrama electrónico como la programación en Arduino para distintas propuestas que se han planteado implementar a lo largo del proceso de diseño del dispensador hasta la fecha, así como se indican las razones por las cuales algunas de las propuestas no fueron implementadas finalmente en el dispensador. También, se muestra la documentación relacionada a las versiones del circuito que si han sido implementadas en algún momento, así como la versión más reciente con la que se encuentra el dispensador funcionando.

Fecha ..... Marzo, 2023  
Autor 1 ..... Jose Pablo Valverde Brandford  
Profesor Coordinador ..... Michael Fernández Vega

## Índice

<b>1. Versiones Propuestas.</b>	<b>3</b>
1.1. Versión con buzzer indicador de nivel de alcohol. . . . .	3
1.2. Versión con buzzer indicador de nivel de alcohol y LED indicador de accio- namiento de bomba. . . . .	7
1.3. Versión con Indicador de LEDs para nivel de alcohol. . . . .	11
<b>2. Versión sin potenciómetro (06/01/2023).</b>	<b>15</b>
<b>3. Versión Primer modelo (02/02/2023).</b>	<b>17</b>
<b>4. Versión actual, segundo modelo (23/05/2023).</b>	<b>20</b>

## 1. Versiones Propuestas.

### 1.1. Versión con buzzer indicador de nivel de alcohol.

Esta versión fue propuesta por el Autor 1, consiste en la implementación de un sistema para controlar el nivel de alcohol en gel en el recipiente de almacenamiento, por medio de un sensor ultrasónico colocado en la boca del recipiente se mide la distancia desde la boca del recipiente a la capa superior del alcohol dentro del recipiente, en el momento en que el sensor mida una distancia definida como el fondo de la botella entonces el buzzer activa una alarma que indica que el recipiente se encuentra vacío.

A continuación se enumeran las razones por las que no fue implementada esta versión:

- Dificultad para colocar el sensor en la boca de la botella y que a la vez se deje espacio para realizar el llenado de la misma.
- Posibilidad de que la vida útil del sensor en la botella se reduzca debido a la humedad o al constante manejo y movimiento al que estaría expuesto debido a que por el diseño del dispensador la botella debe sacarse del mismo para rellenarla.

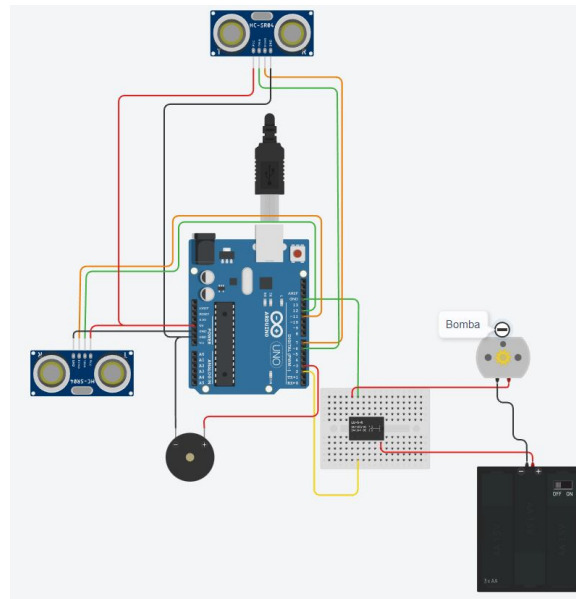
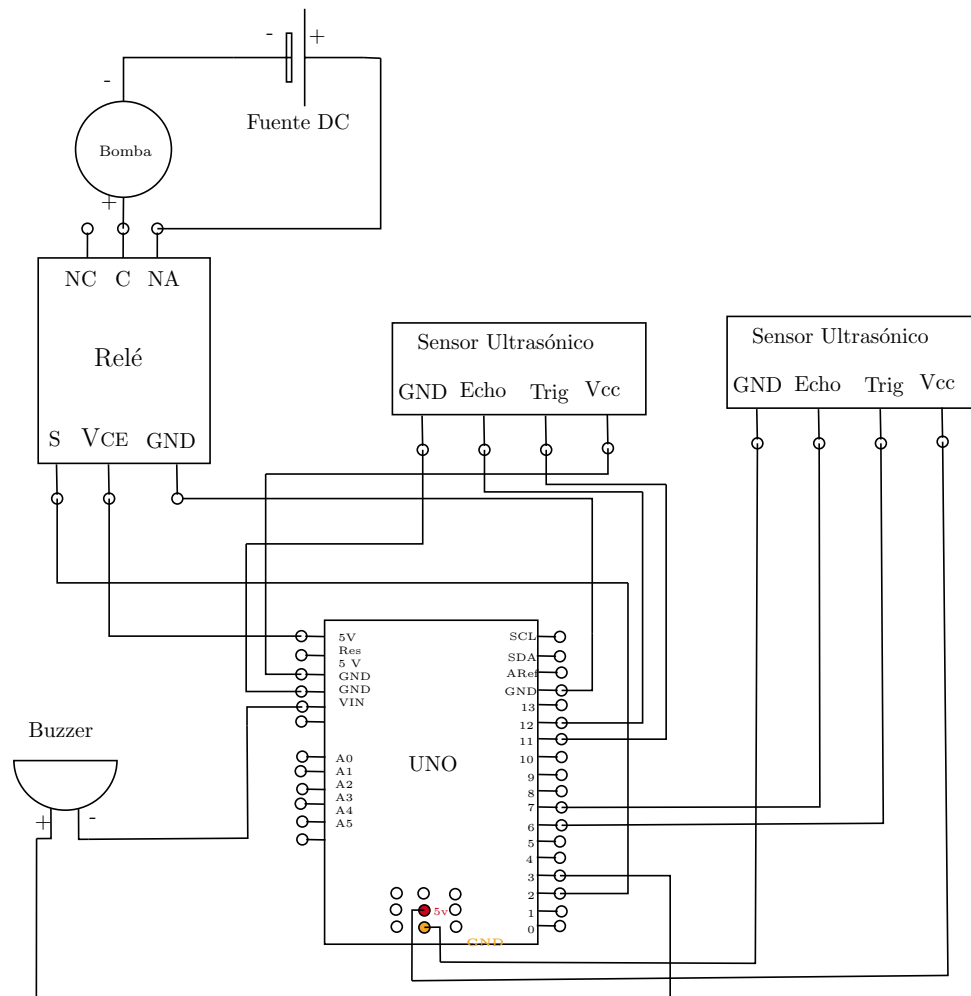


Figura 1: Captura del circuito con buzzer indicador de nivel de alcohol.



```

1
2
3 #include <Wire.h>
4
5 const int buzzer = 3;
6 const int Bomba = 2;
7 int trig = 12; //int sensor dist mano
8 int echo = 11; //echo sensor dist mano
9
10 int trig2 = 6; //int sensor Control liquido
11 int echo2 = 7; //echo sensor Control liquido
12
13
14 float distancia;
15 long duracion;
16
17 int TimeOn; // Tiempo en estado alto se al Sensor Nivel liquido
18 int Nivel; // Determina la distacia segun la velocidad del sonido y la
    duracion TimeOn
19
20 void setup() {
21
22
23     Serial.begin(9600); //iniciar puerto serie
24     pinMode(Bomba , OUTPUT); //definir pin como salida
25     pinMode(trig , OUTPUT);
26     pinMode(echo , INPUT_PULLUP);
27
28     pinMode(trig2,OUTPUT);
29     pinMode(echo2,INPUT_PULLUP);
30
31     pinMode(buzzer ,OUTPUT);
32
33
34 }
35
36 void loop(){
37
38     digitalWrite (trig2,HIGH);
39     delay(1);
40     digitalWrite (trig2,LOW);
41
42     TimeOn= pulseIn(echo2,HIGH); //tiempo que dura la se al del sensor en
        ser emitida y recibida
43     Nivel=float ((TimeOn/2)*0.0343); // Convierte el tiempo anterior en
        una distancia (en CM)
44     Serial.println(Nivel);
45
46     if (Nivel>=250 ) { //en este if se define todo lo relacionado al nivel
        del alcohol m nimo
47
48         digitalWrite(buzzer ,HIGH);
49         delay(1000);
50         digitalWrite(buzzer ,LOW);
51
52
53     }
54
55     //+++++ Sensor mano y bomba
        +++++

```

```
56  digitalWrite(trig , LOW);
57  delayMicroseconds(2);
58  digitalWrite(trig , HIGH);
59  delayMicroseconds(10);
60  digitalWrite(trig , LOW);
61  duracion = pulseIn(echo, HIGH);
62  distancia = float((duracion/2)*0.0343);
63  Serial.print(distancia);
64  Serial.print("cm");
65
66  if(distancia <=100){
67
68      digitalWrite(Bomba, HIGH);
69      delay(1000); // Tiempo que dura la bomba encendida
70      digitalWrite(Bomba, LOW);
71      delay(5000);
72  }
73  else {
74      digitalWrite(Bomba, LOW);
75  }
76
77  delay(500);
78 }
```

Listing 1: Código de programación del circuito anterior

## 1.2. Versión con buzzer indicador de nivel de alcohol y LED indicador de accionamiento de bomba.

Esta versión fue propuesta por el Autor 1, consiste en la implementación del mismo sistema para controlar el nivel de alcohol en gel en el recipiente de almacenamiento mencionado en la versión anterior pero ahora implementando también un diodo LED el cual indica el accionamiento de la bomba. La finalidad de la implementación del diodo LED es que en caso de que el dispensador presente algún problema se pueda identificar rápidamente si el sistema de la bomba está alimentado y funcionando bien y descartar que el problema que presente el dispensador sea propiciado por un fallo en las conexiones de la bomba o el sensor. En otras palabras, el LED sería un indicador de que el sistema electrónico del dispensador se encuentra funcionando correctamente.

A continuación se enumeran las razones por las que no fue implementada esta versión:

- Dificultad para colocar el sensor en la boca de la botella y que a la vez se deje espacio para realizar el llenado de la misma.
- Posibilidad de que la vida útil del sensor en la botella se reduzca debido a la humedad o al constante manejo y movimiento al que estaría expuesto debido a que por el diseño del dispensador la botella debe sacarse del mismo para rellenarla.
- Se considera innecesario la implementación del LED ya que la bomba produce un sonido suficientemente alto como para que ese sonido pueda ser indicador de que el sistema electrónico se encuentra funcionando correctamente.

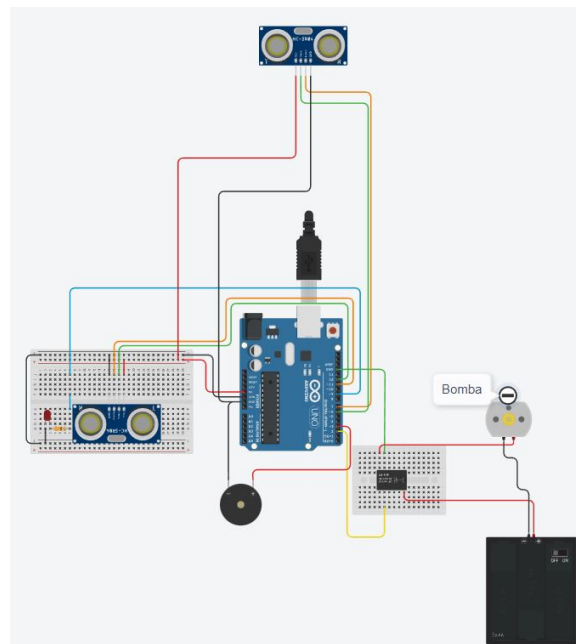
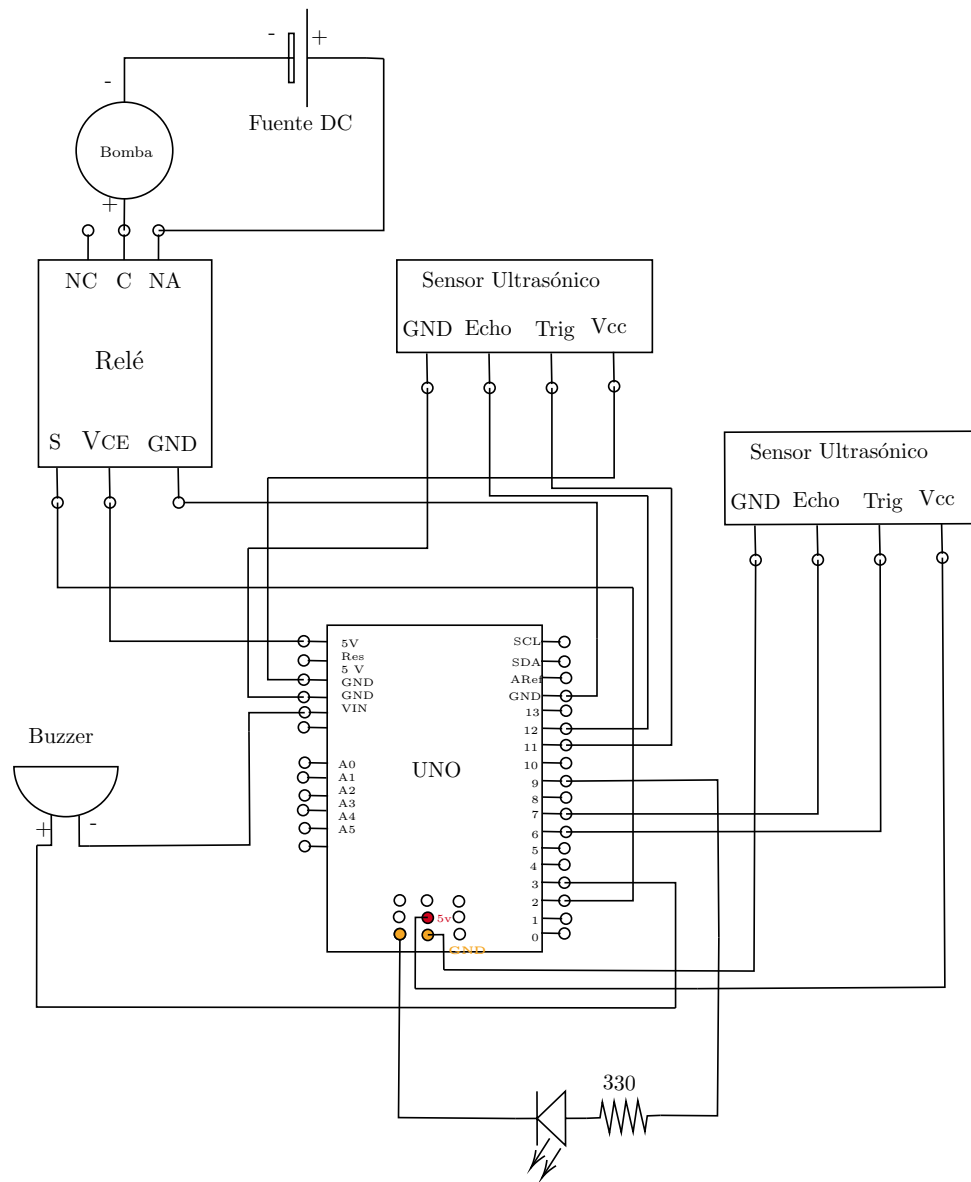


Figura 2: Captura del circuito con buzzer indicador de nivel de alcohol y LED indicador de accionamiento del dispensador.





```
1
2
3 #include <Wire.h>
4
5 const int buzzer = 3;
6 const int Bomba = 2;
7 int trig = 12; //int sensor dist mano
8 int echo = 11; //echo sensor dist mano
9
10 int trig2 = 6; //int sensor Control liquido
11 int echo2 = 7; //echo sensor Control liquido
12
13 int LED_Bomba = 9;
14 float distancia;
15 long duracion;
16
17 int TimeOn; // Tiempo en estado alto se al Sensor Nivel liquido
18 int Nivel; // Determina la distacia segun la velocidad del sonido y la
    duracion TimeOn
19
20
21 void setup() {
22
23
24     Serial.begin(9600); //iniciar puerto serie
25     pinMode(Bomba , OUTPUT); //definir pin como salida
26     pinMode(trig , OUTPUT);
27     pinMode(echo , INPUT_PULLUP);
28     pinMode(LED_Bomba, OUTPUT);
29
30     pinMode(trig2,OUTPUT);
31     pinMode(echo2,INPUT_PULLUP);
32
33     pinMode(buzzer ,OUTPUT);
34
35
36 }
37
38 void loop(){
39
40     digitalWrite (trig2,HIGH);
41     delay(1);
42     digitalWrite (trig2,LOW);
43
44     TimeOn= pulseIn(echo2,HIGH); //tiempo que dura la se al del sensor en
        ser emitida y recibida
45     Nivel=float ((TimeOn/2)*0.0343); // Convierte el tiempo anterior en
        una distancia (en CM)
46     Serial.println(Nivel);
47
48     if (Nivel>=250 ) { //en este if se define todo lo relacionado al nivel
        del alcohol m nimo
49
50         digitalWrite(buzzer ,HIGH);
51         delay(1000);
52         digitalWrite(buzzer ,LOW);
53
54     }
55
56
```

```
57 //+++++ Sensor mano y bomba
58 digitalWrite(trig , LOW);
59 delayMicroseconds(2);
60 digitalWrite(trig , HIGH);
61 delayMicroseconds(10);
62 digitalWrite(trig , LOW);
63 duracion = pulseIn(echo , HIGH);
64 distancia = float((duracion/2)*0.0343);
65 Serial.print(distancia);
66 Serial.print("cm");
67
68 if(distancia <=100){
69     digitalWrite(Bomba, HIGH);
70     digitalWrite(LED_Bomba, HIGH);
71     delay(1000); // Tiempo que dura la bomba encendida
72     digitalWrite(Bomba, LOW);
73     digitalWrite(LED_Bomba, LOW);
74     delay(5000);
75 }
76 else {
77     digitalWrite(Bomba, LOW);
78 }
79
80 delay(500);
81 }
82 }
```

Listing 2: Código de programación del circuito anterior

### 1.3. Versión con Indicador de LEDs para nivel de alcohol.

Esta versión fue propuesta por el Autor 1, consiste en la implementación de un sistema para controlar el nivel de alcohol en gel en el recipiente de almacenamiento por medio de un indicador de tres niveles con LEDs, este sistema utiliza de igual forma que las versiones propuestas anteriores, un sensor ultrasónico para medir el nivel de alcohol en la botella, conforme el sensor colocado en la boca de la botella va midiendo una distancia mayor, el sistema indicador con LEDs va cambiando el LED alimentado, cuando la distancia medida es pequeña, significa que la botella se encuentra relativamente llena, entonces el LED que se encuentra prendido es el verde, cuando la botella se encuentra en un rango medio entonces el LED es amarillo, y cuando la botella se encuentra vacía entonces se prende el LED rojo.

A continuación se enumeran las razones por las que no fue implementada esta versión:

- Dificultad para colocar el sensor en la boca de la botella y que a la vez se deje espacio para realizar el llenado de la misma.
- Posibilidad de que la vida útil del sensor en la botella se reduzca debido a la humedad o al constante manejo y movimiento al que estaría expuesto debido a que por el diseño del dispensador la botella debe sacarse del mismo para rellenarla.
- Se tendría que implementar el uso de una protoboard lo que implicaría un mayor espacio dentro del dispensador para el circuito electrónico.

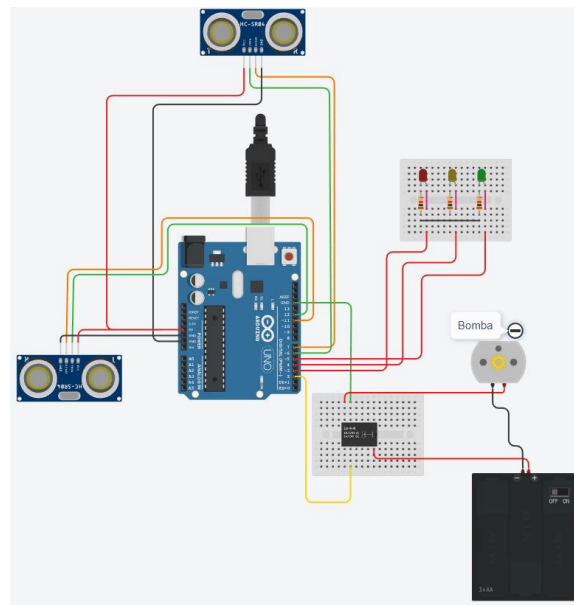
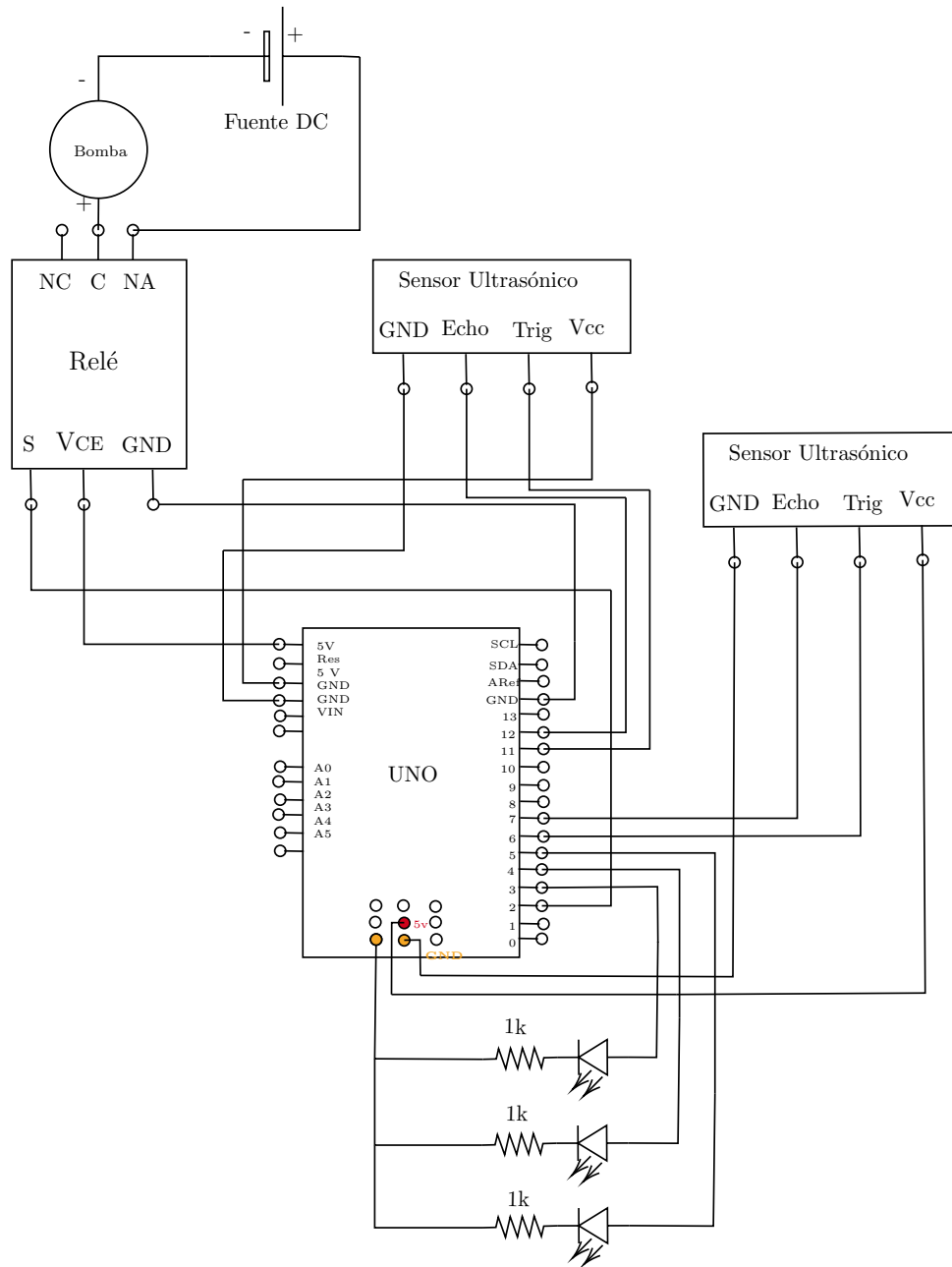


Figura 3: Captura del circuito con circuito de LEDs indicador de nivel de alcohol.



```

1
2
3 #include <Wire.h>
4
5 const int Bomba = 2;
6 int trig = 12; //int sensor dist mano
7 int echo = 11; //echo sensor dist mano
8
9 int trig2 = 6; //int sensor Control liquido

```

```

10 int echo2 = 7; //echo sensor Control liquido
11
12
13
14
15 float distancia;
16 long duracion;
17
18 int LED_Rojo_Ind_Nivel = 3;
19 int LED_Amarillo_Ind_Nivel = 4;
20 int LED_Verde_Ind_Nivel = 5;
21
22
23 int TimeOn; // Tiempo en estado alto se al Sensor Nivel liquido
24 int Nivel; // Determina la distacia segun la velocidad del sonido y la
    duracion TimeOn
25
26
27 void setup() {
28
29
30     Serial.begin(9600); //iniciar puerto serie
31     pinMode(Bomba , OUTPUT); //definir pin como salida
32     pinMode(trig , OUTPUT);
33     pinMode(echo , INPUT_PULLUP);
34
35
36     pinMode(trig2,OUTPUT);
37     pinMode(echo2,INPUT_PULLUP);
38
39     pinMode(LED_Rojo_Ind_Nivel,OUTPUT);
40     pinMode(LED_Amarillo_Ind_Nivel,OUTPUT);
41     pinMode(LED_Verde_Ind_Nivel,OUTPUT);
42
43
44 }
45
46 void loop(){
47
48     //+++++Indicador nivel con tira de LED y
    pantalla +++++
49
50
51     digitalWrite (trig2,HIGH);
52     delay(1);
53     digitalWrite (trig2,LOW);
54
55     TimeOn= pulseIn(echo2,HIGH); //tiempo que dura la se al del sensor en
    ser emitida y recibida
56     Nivel=float ((TimeOn/2)*0.0343); // Convierte el tiempo anterior en
    una distancia (en CM)
57     Serial.println(Nivel);
58
59     if (Nivel>=250 ) { //en este if se define todo lo relacionado al nivel
    del alcohol m nimo
60
61         digitalWrite(LED_Rojo_Ind_Nivel,HIGH);
62         digitalWrite(LED_Amarillo_Ind_Nivel,LOW);
63         digitalWrite(LED_Verde_Ind_Nivel,LOW);
64

```

```

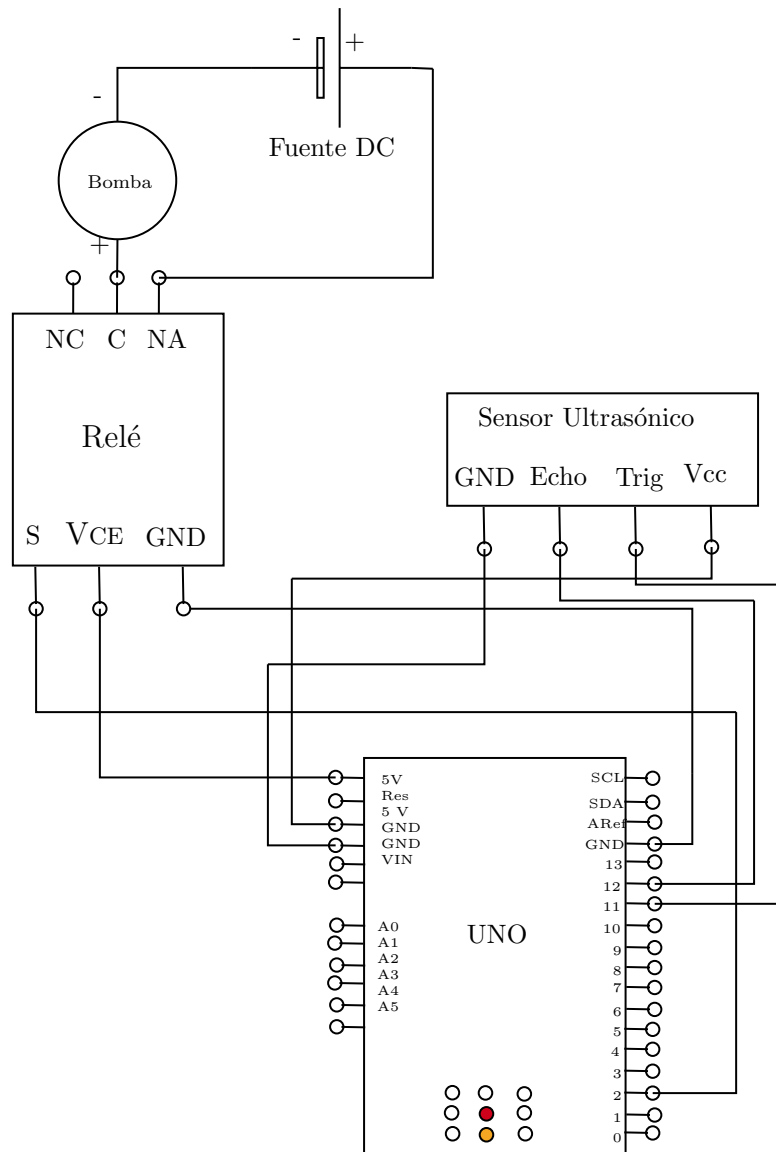
65     }
66     if (Nivel<250 && Nivel>=100) {
67
68         digitalWrite(LED_Rojo_Ind_Nivel,LOW);
69         digitalWrite(LED_Amarillo_Ind_Nivel,HIGH);
70         digitalWrite(LED_Verde_Ind_Nivel,LOW);
71
72     }
73     if (Nivel<100) {
74
75         digitalWrite(LED_Rojo_Ind_Nivel,LOW);
76         digitalWrite(LED_Amarillo_Ind_Nivel,LOW);
77         digitalWrite(LED_Verde_Ind_Nivel,HIGH);
78
79     }
80
81
82     //+++++ Sensor mano y bomba
83     digitalWrite(trig, LOW);
84     delayMicroseconds(2);
85     digitalWrite(trig, HIGH);
86     delayMicroseconds(10);
87     digitalWrite(trig, LOW);
88     duracion = pulseIn(echo, HIGH);
89     distancia = float((duracion/2)*0.0343);
90     Serial.print(distancia);
91     Serial.print("cm");
92
93     if(distancia <=100){
94
95         digitalWrite(Bomba, HIGH);
96         delay(1000); // Tiempo que dura la bomba encendida
97         digitalWrite(Bomba, LOW);
98         delay(5000);
99     }
100     else {
101         digitalWrite(Bomba, LOW);
102     }
103
104     delay(500);
105 }

```

Listing 3: Código de programación del circuito anterior

## 2. Versión sin potenciómetro (06/01/2023).

Esta versión es la más básica para el funcionamiento del dispensador, fue propuesta por los compañeros que empezaron el diseño de este proyecto en semestres anteriores, consiste únicamente en un sensor ultrasónico colocado en la boquilla del dispensador que detecta cuando se acerca la mano, luego es accionada la bomba de diafragma que toma el alcohol del dispensador presente dentro de la carcasa del dispensador y lo dispensa por medio de la boquilla a la mano del usuario, todo controlado por el módulo de Arduino UNO. Esta versión fue implementada y probada con éxito en el dispensador.



```

1 //CREACIN DE LAS CONSTANTES
2 const int rele = 2; // rele que acciona la bomba
3 int trig = 12; // trig del sensor
4 int echo = 11; // echo del sensor
5 float distancia; // distancia de accionamiento del sensor
6 long duracion; // duraci n que se mantiene activa el sensor
7
8 void setup() {
9     Serial.begin(9600); //iniciar puerto serie
10    pinMode(rele , OUTPUT); //definir pin como salida
11    pinMode(trig , OUTPUT); // trig salida
12    pinMode(echo , INPUT); // echo entrada
13 }
14
15 void loop(){
16
17     //C digo del sensor
18     digitalWrite(trig , LOW); // se apaga el sensor
19     delayMicroseconds(2); // duraci n apagado ms
20     digitalWrite(trig , HIGH); // se enciende el sensor
21     delayMicroseconds(10); // duracion encendido en ms
22     digitalWrite(trig , LOW); // se apaga el sensor
23     duracion = pulseIn(echo , HIGH); // guarda el accionar del sensor
24     distancia = float((duracion/2)*0.0343); // transforma el tiempo
        accionado en distancia
25     Serial.print(distancia); // se monitorea la distancia de accionado
26     Serial.print("cm"); // se monitorea la distancia de accionado
27
28     //C digo para el accionar de la bomba
29     if(distancia <=5) // se define el rango de funcionamiento del sensor para
        la bomba
30         digitalWrite(rele , HIGH); // activar la bomba
31     }
32     else if((distancia >=6) && (distancia <= 180)){
33         digitalWrite(rele , LOW); // se desactiva la bomba
34     }
35     else if(distancia>=10){
36         digitalWrite(rele , LOW); // se desactiva la bomba
37     }
38     delay(100); // este delay es para reducir la cantidad de lecturas
        seguidas del sensor
39 }

```

Listing 4: Código de programación del circuito anterior



### 3. Versión Primer modelo (02/02/2023).

Esta versión se basa en la anterior, es en esencia exactamente igual electrónicamente hablando pero ahora por recomendación del profesor coordinado Michael Fernández se implementa un potenciómetro que permite regular la cantidad de alcohol que dispensa el dispositivo diseñado. Esta versión fue implementada y probada con éxito en el dispensador y a la fecha de la última actualización del presente documento, esta es la versión con la que se encuentra funcionando el dispositivo.

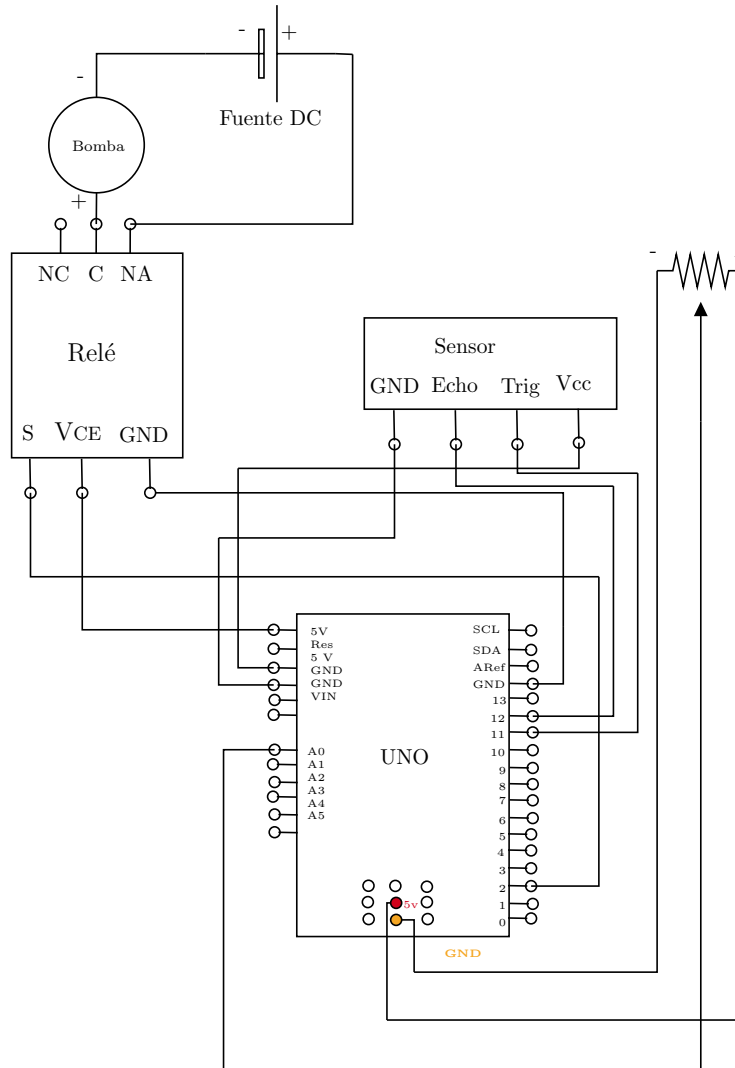


Figura 4: Esquemático eléctrico dispensador de alcohol.

```

1 //CREACIN DE LAS CONSTANTES
2 const int rele = 2; // rele que acciona la bomba
3 int trig = 12; // trig del sensor
4 int echo = 11; // echo del sensor
5 float distancia; // distancia de accionamiento del sensor
6 long duracion; // duraci n que se mantiene activa el sensor
7 int pot; // variable para definir el potenciometro de 0 a 1023
8 int cantidad; // variable para relacionar el potenciometro a la cantidad
   de alcohol
9
10
11 void setup() {
12   Serial.begin(9600); //iniciar puerto serie
13   pinMode(rele , OUTPUT); //definir pin como salida
14   pinMode(trig , OUTPUT); // trig salida
15   pinMode(echo, INPUT); // echo entrada
16 }
17
18 void loop(){
19
20   //C digo para el potencio metro
21   pot = analogRead(A0); // lectura del potenciometro (va de 0 a 1023)
22   cantidad = pot/10.23; // relaci n del potenciometro a la cantidad
   que se quiere (de 0 100)
23   Serial.print("Cantidad es de "); // se imprime el dialogo de
   cantidad
24   Serial.println(cantidad); // se imprime la cantidad (duracion de la
   bomba en milisegundos)
25
26   //C digo del sensor
27   digitalWrite(trig , LOW); // se apaga el sensor
28   delayMicroseconds(2); // duraci n apagado ms
29   digitalWrite(trig , HIGH); // se enciende el sensor
30   delayMicroseconds(10); // duracion encendido en ms
31   digitalWrite(trig , LOW); // se apaga el sensor
32   duracion = pulseIn(echo , HIGH); // guarda el accionar del sensor
33   distancia = duracion/57.8; // transforma el tiempo accionado en
   distancia
34   Serial.print(distancia); // se monitorea la distancia de accionado
35   Serial.println("cm"); //// se monitorea la distancia de accionado
36
37
38   //C digo para el accionar de la bomba
39   if(distancia <=15 && distancia>7){ // se define el rango de
   funcionamiento del sensor para la bomba
40   delay(300) ; // demora para que se accion la bomba al activarse el
   sensor
41   digitalWrite(rele , HIGH); // activar la bomba
42   delay(cantidad); // duraci n que debe durar la bomba activa , esto
   define la cantidad de alcohol
43   digitalWrite(rele , LOW); // se desactiva la bomba
44   delay(10000); // demora para que vuelva a funcionar la bomba en caso
   de utilizarse seguido
45   }
46
47   delay(100); // este delay es para reducir la cantidad de lecturas
   seguidas del sensor
48 }

```

---

Listing 5: Código de programación del pimer modelo

#### 4. Versión actual, segundo modelo (23/05/2023).

A partir de la retroalimentación obtenida a partir del personal del hogar de ancianos luego del periodo de pruebas del primer modelo del dispensador, se evalúan varias opciones de mejora, escogiendo la opción de implementar un sistema indicador de LEDs y buzzer con el fin de que sirva como señal al usuario para que éste sepa cuando el dispensador se encuentra listo para dispensar el alcohol, y cuando el dispositivo se encuentra realizando el delay entre accionamiento, esto permite evitar que el usuario ponga la mano muy seguido y al dispensador no le dé tiempo de dispensar el alcohol.

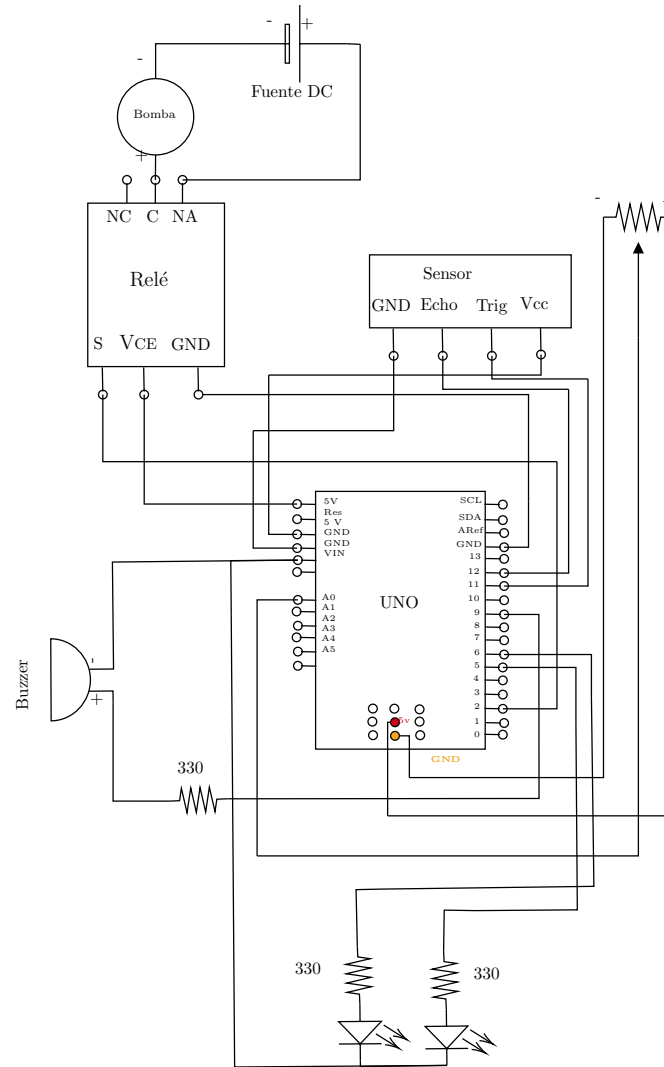


Figura 5: Esquemático eléctrico dispensador de alcohol.

```

1 //CREACIN DE LAS CONSTANTES
2 const int rele = 2; // rele que acciona la bomba
3 int trig = 12; // trig del sensor
4 int echo = 11; // echo del sensor
5 float distancia; // distancia de accionamiento del sensor
6 long duracion; // duraci n que se mantiene activa el sensor
7 int pot; // variable para definir el potenciometro de 0 a 1023
8 int cantidad; // variable para relacionar el potenciometro a la cantidad
  de alcohol
9 int led_rojo = 5;
10 int led_verde= 6;
11 const int buzzer = 9;
12
13 void setup() {
14   Serial.begin(9600); //iniciar puerto serie
15   pinMode(rele , OUTPUT); //definir pin como salida
16   pinMode(trig , OUTPUT); // trig salida
17   pinMode(echo , INPUT); // echo entrada
18   pinMode(led_rojo , OUTPUT);
19   pinMode(led_verde , OUTPUT);
20   pinMode(buzzer , OUTPUT);
21 }
22
23 void loop(){
24
25   //C digo para el potencio metro
26   pot = analogRead(A0); // lectura del potenciometro (va de 0 a 1023)
27   cantidad = pot/10.23; // relaci n del potenciometro a la cantidad
  que se quiere (de 0 100)
28   Serial.print("Cantidad es de = "); // se imprime el dialogo de
  cantidad
29   Serial.println(cantidad); // se imprime la cantidad (duracion de la
  bomba en milisegundos)
30
31   //C digo del sensor
32   digitalWrite(trig , LOW); // se apaga el sensor
33   delayMicroseconds(2); // duraci n apagado ms
34   digitalWrite(trig , HIGH); // se enciende el sensor
35   delayMicroseconds(10); // duracion encendido en ms
36   digitalWrite(trig , LOW); // se apaga el sensor
37   duracion = pulseIn(echo , HIGH); // guarda el accionar del sensor
38   distancia = duracion/57.8; // transforma el tiempo accionado en
  distancia
39   Serial.print(distancia); // se monitorea la distancia de accionado
40   Serial.println("cm"); //// se monitorea la distancia de accionado
41
42   digitalWrite(led_rojo , LOW);
43   digitalWrite(led_verde , HIGH);
44
45   //C digo para el accionar de la bomba
46   if(distancia <=12 && distancia>7){ // se define el rango de
  funcionamiento del sensor para la bomba
47   delay(500) ; // demora para que se accion la bomba al activarse el
  sensor
48   digitalWrite(rele , HIGH); // activar la bomba
49   digitalWrite(led_verde , LOW);
50   digitalWrite(led_rojo , HIGH);
51   delay(cantidad); // duraci n que debe durar la bomba activa , esto
  define la cantidad de alcohol
52   digitalWrite(rele , LOW); // se desactiva la bomba

```

```
53
54     analogWrite (buzzer,5000);
55     delay(3000); // demora para que vuelva a funcionar la bomba en caso
de utilizarse seguido
56     analogWrite (buzzer,0);
57 }
58
59 //delay(100); // este delay es para reducir la cantidad de lecturas
seguidas del sensor
60 }
```

Listing 6: Código de programación más reciente a la fecha