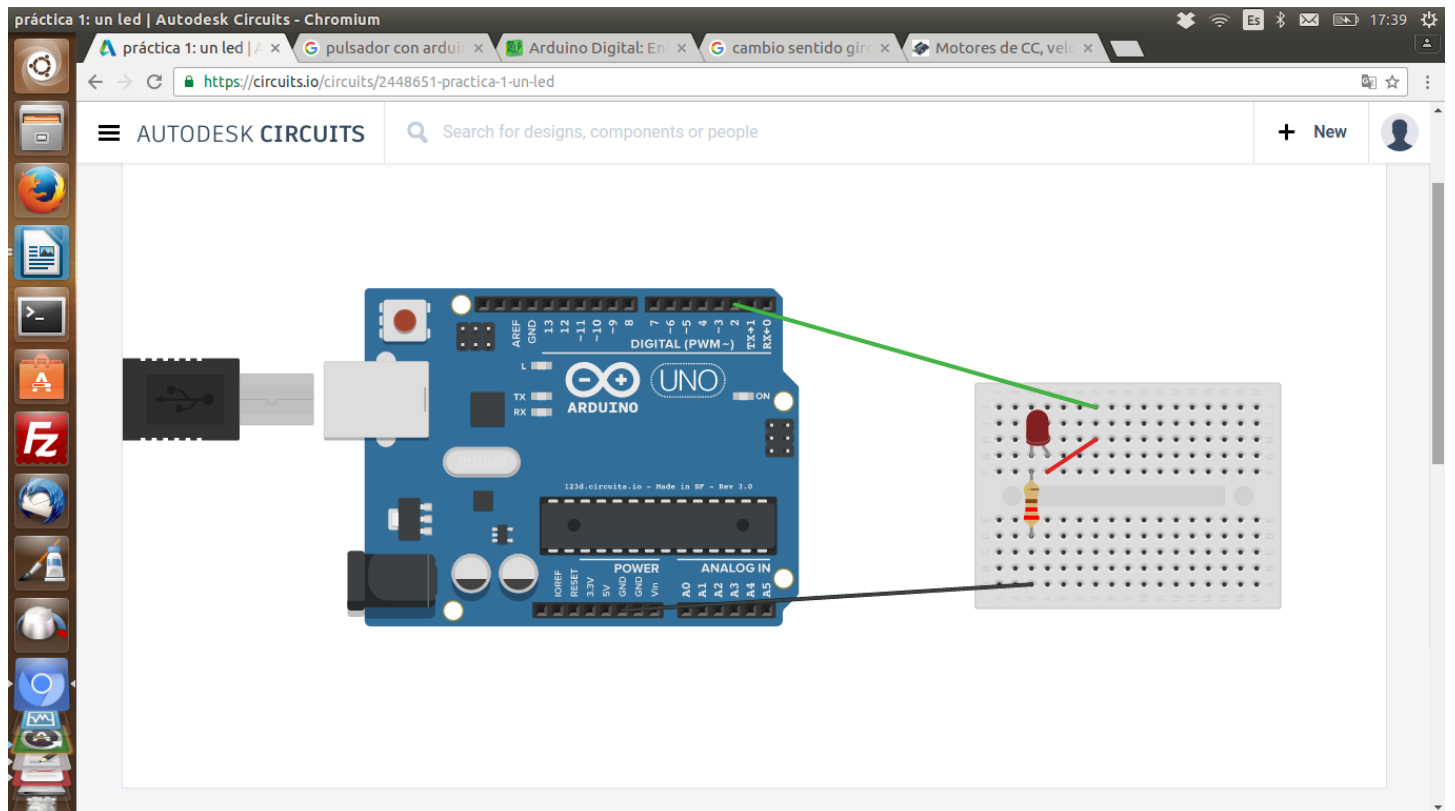


PRÁCTICAS ARDUINO_2017

**PARA CONFIGURAR EL PUERTO ESCRIBIMOS EN EL TERMINAL:
Aplicaciones---accesorios--terminal (si pide contraseña es usuario)**

```
sudo chmod 666 /dev/ttyACM0
```

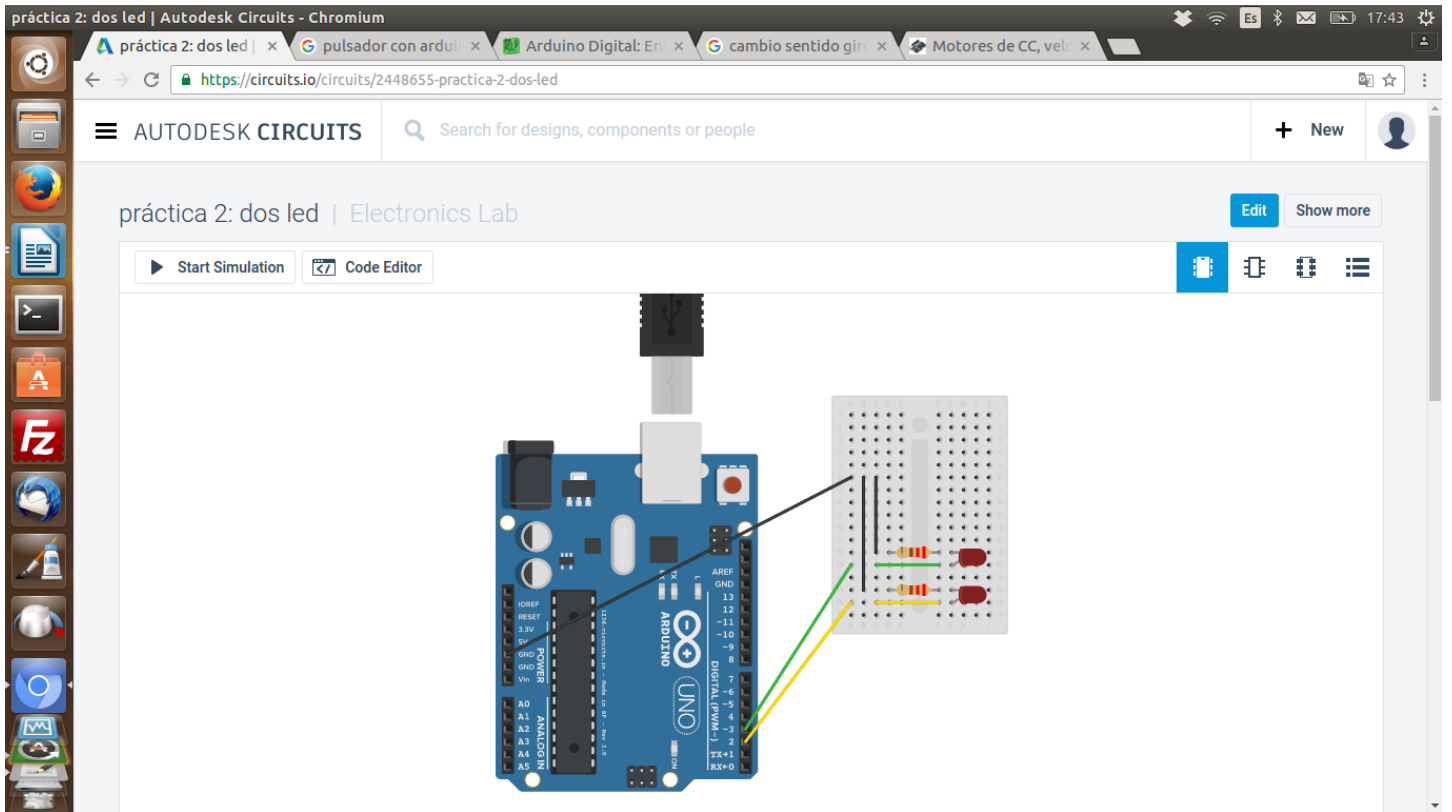
PRÁCTICA 1: ENCENDER UN LED



```
///  
//# Práctica 1: encendemos y apagamos un led con el pin 2 cada segundo  
void setup ()  
{  
  pinMode(2,OUTPUT); //configuramos el pin 2 como salida  
}  
void loop()  
{  
  digitalWrite(2,HIGH); //enciende el pin 2, un segundo, pag 159  
  delay(1000);  
  digitalWrite(2,LOW); //apaga el pin 2 , un segundo  
  delay(1000);  
}
```

PRÁCTICA 1.2: EXPERIMENTA DISMINUYENDO EL TIEMPO ENTRE ENCENDIDOS DEL LEDS, apreciarás como llegará el momento en que prácticamente no se apaga.

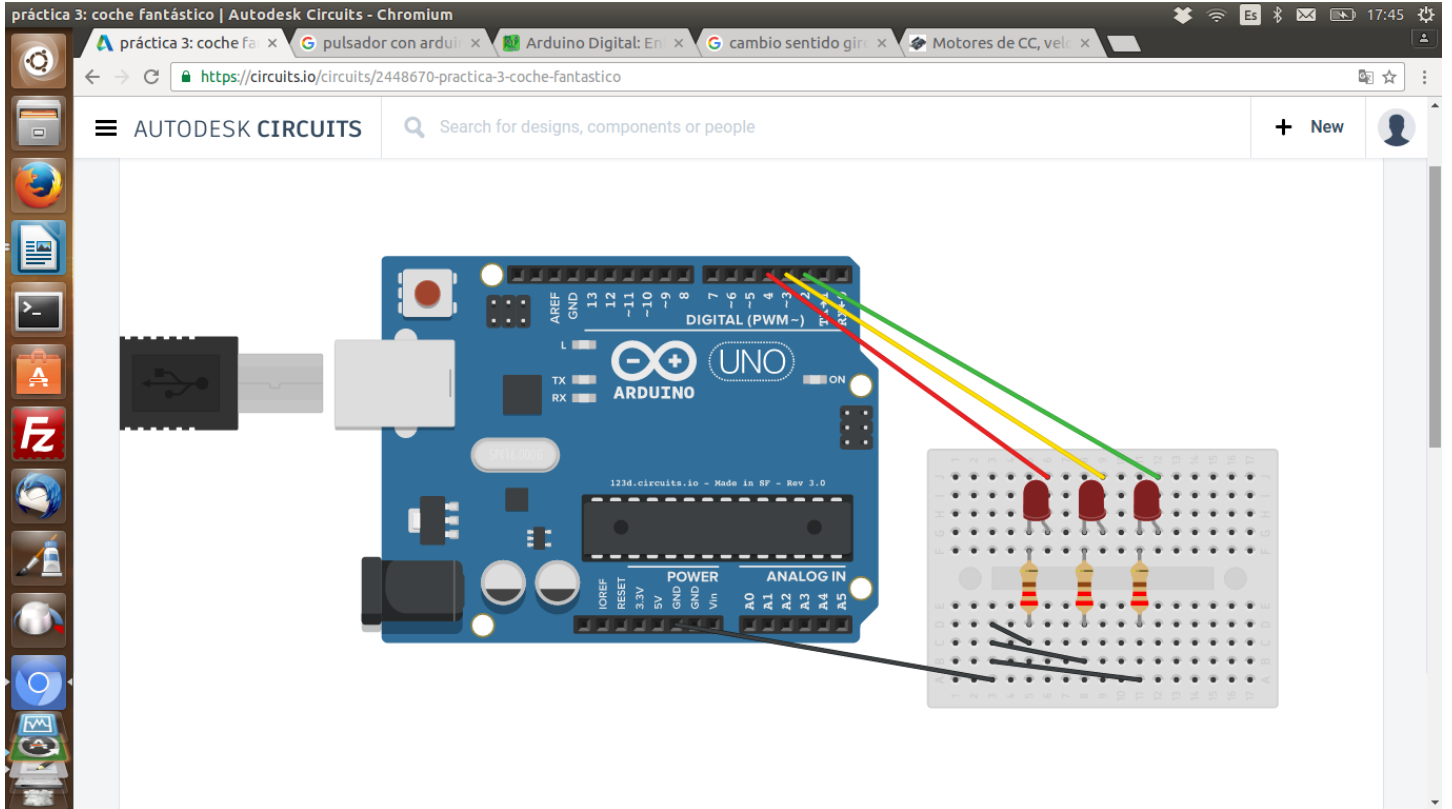
PRÁCTICA 2: DOS LEDS



```
void setup ()
{
  pinMode(2,OUTPUT); //configuramos el pin 2 como salida
  pinMode(3,OUTPUT); // configuramos el pin 7 como salida
}
void loop()
{
  digitalWrite(2,HIGH); //enciende el pin 2, un segundo, pag 159
  digitalWrite(3,LOW); //apaga el pin 7, un segundo
  delay(1000);
  digitalWrite(2,LOW); //apaga el pin 2 , un segundo
  digitalWrite(3,HIGH); //enciende el pin 7 un segudno
  delay(1000);
}
```

PRÁCTICA 2.2: EXPERIMENTA DISMINUYENDO EL TIEMPO ENTRE ENCENDIDOS DEL LEDS, apreciarás como llegará el momento en que prácticamente no se distingue quién está apagado o encendido.

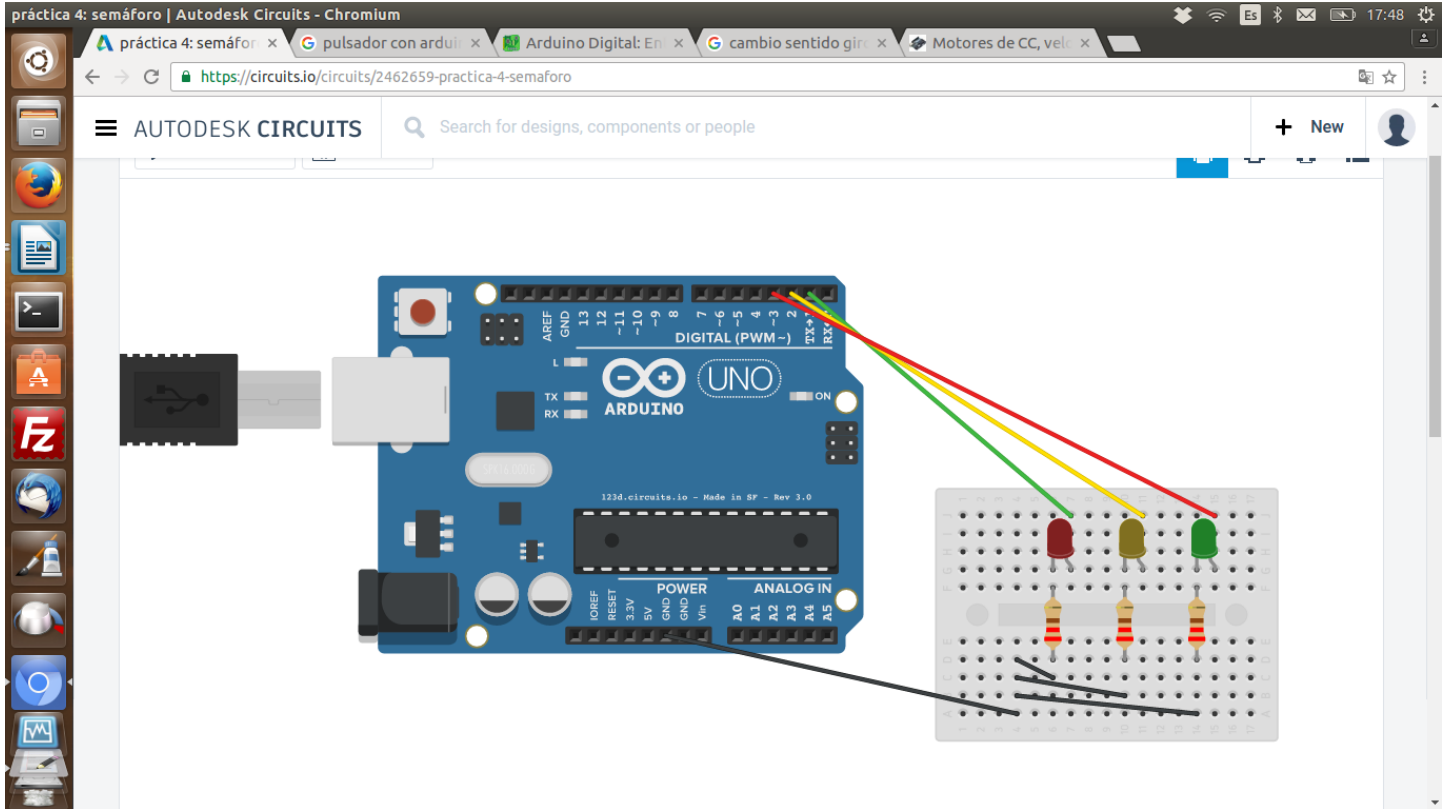
PRÁCTICA 3: COCHE FANTÁSTICO



```
void setup ()
{
  pinMode(2,OUTPUT);// configuramos el pin 2 como salida
  pinMode(3,OUTPUT); //configuramos el pin 3 como salida
  pinMode(4,OUTPUT);// configuramos el pin 4 como salida
}
void loop()
{
  digitalWrite(2,HIGH);//enciende el pin 2, medio segundo, pag 159
  digitalWrite(3,LOW);//apaga el pin 3, medio segundo
  digitalWrite(4,LOW);
  delay(500);
  digitalWrite(2,LOW);//enciende el pin 2, medio segundo, pag 159
  digitalWrite(3,HIGH);//apaga el pin 3, medio segundo
  digitalWrite(4,LOW);
  delay(500);
  digitalWrite(2,LOW);//enciende el pin 2, medio segundo, pag 159
  digitalWrite(3,LOW);//apaga el pin 3, medio segundo
  digitalWrite(4,HIGH);
  delay(500);
  digitalWrite(2,LOW);//enciende el pin 2, medio segundo, pag 159
  digitalWrite(3,HIGH);//apaga el pin 3, medio segundo
  digitalWrite(4,LOW);
  delay(500);
}
```

PRÁCTICA 3.2: EXPERIMENTA DISMINUYENDO EL TIEMPO ENTRE ENCENDIDOS DEL LEDS, apreciarás como evoluciona la transición entre leds.

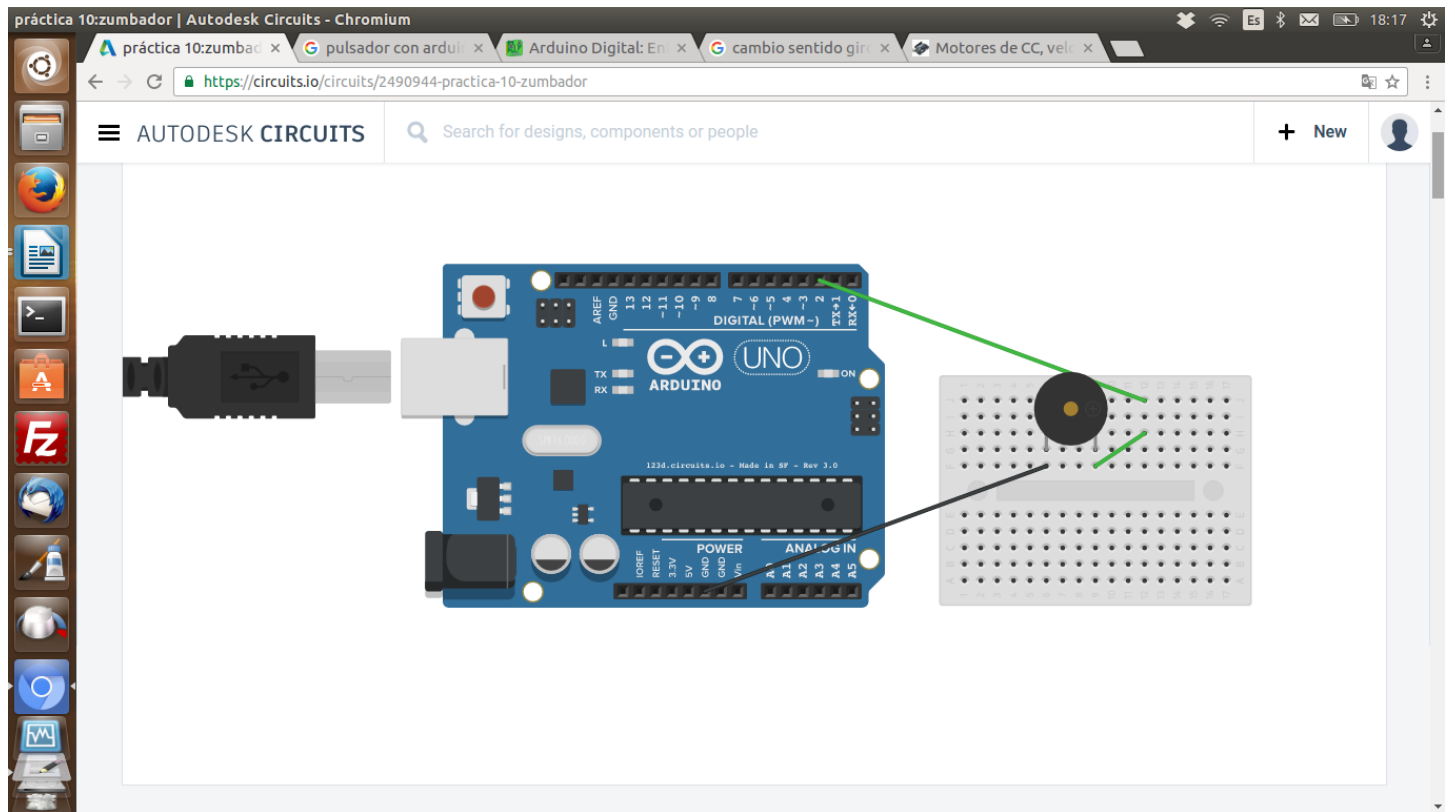
PRÁCTICA 4: SEMÁFORO



```
#define LEDVERDE 1    //Renombramos el pin 1 a LEDVERDE
#define LEDAMARILLO 2 //Renombramos el pin 2 a LEDAMARILLO
#define LEDROJO 3     //Renombramos el pin 3 a LEDROJO
```

```
void setup ()
{
  pinMode(LEDVERDE,OUTPUT); //configuramos el pin LEDVERDE como salida
  pinMode(LEDAMARILLO,OUTPUT); // configuramos el pin LEDAMARILLO como salida
  pinMode(LEDROJO,OUTPUT); //configuramos el pin LEDROJO como salida
}
void loop()
{
  digitalWrite(LEDVERDE,LOW); //enciende el pin LEDAMARILLO,
  digitalWrite(LEDAMARILLO,LOW); //apaga el pin LEDROJO,
  digitalWrite(LEDROJO,HIGH);
  delay(5000);
  digitalWrite(LEDVERDE,HIGH); //enciende el pin LEDAMARILLO,
  digitalWrite(LEDAMARILLO,LOW); //apaga el pin LEDROJO,
  digitalWrite(LEDROJO,LOW);
  delay(5000);
  digitalWrite(LEDVERDE,LOW); //enciende el pin LEDAMARILLO,
  digitalWrite(LEDAMARILLO,HIGH); //apaga el pin LEDROJO,
  digitalWrite(LEDROJO,LOW);
  delay(3000);
}
```

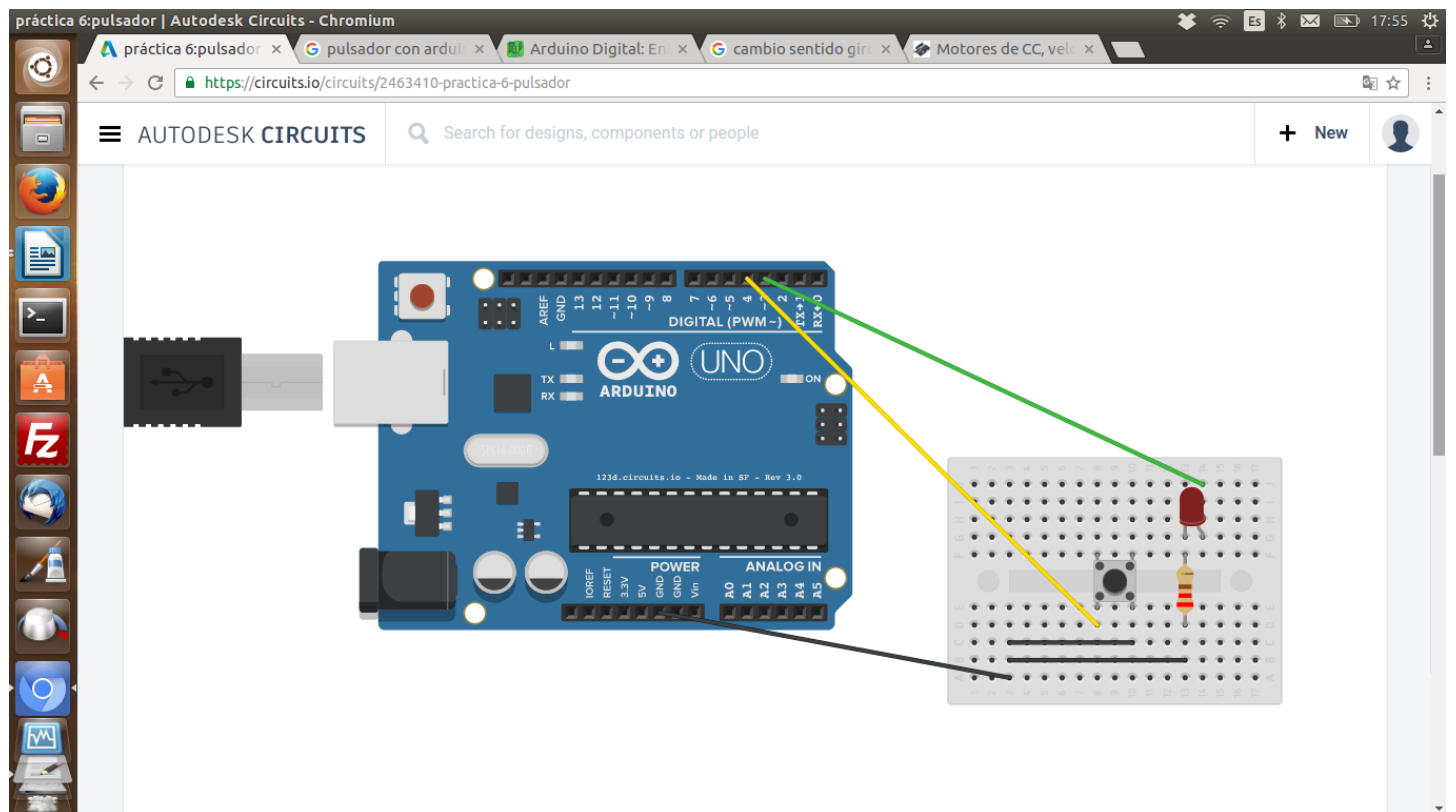
PRÁCTICA 5: ZUMBADOR SIMPLE.



```
## Práctica 5: encendemos y apagamos un zumbador o timbre con el pin 2 cada segundo
void setup ()
{
  pinMode(2,OUTPUT); //configuramos el pin 2 como salida
}
void loop()
{
  digitalWrite(2,HIGH);//enciende el pin 2, un segundo, pag 159
  delay(1000);
  digitalWrite(2,LOW);//apaga el pin 2 , un segundo
  delay(1000);
}
```

PRÁCTICA 5.2: DISEÑA EL PROGRAMA NECESARIO PARA QUE EL ZUMBADOR SUENE S.O.S EN CÓDIGO MORSE, sería dando tres señales eléctricas al pulsador entre delays de apagado , una corta, una larga , una corta (S.O.S).

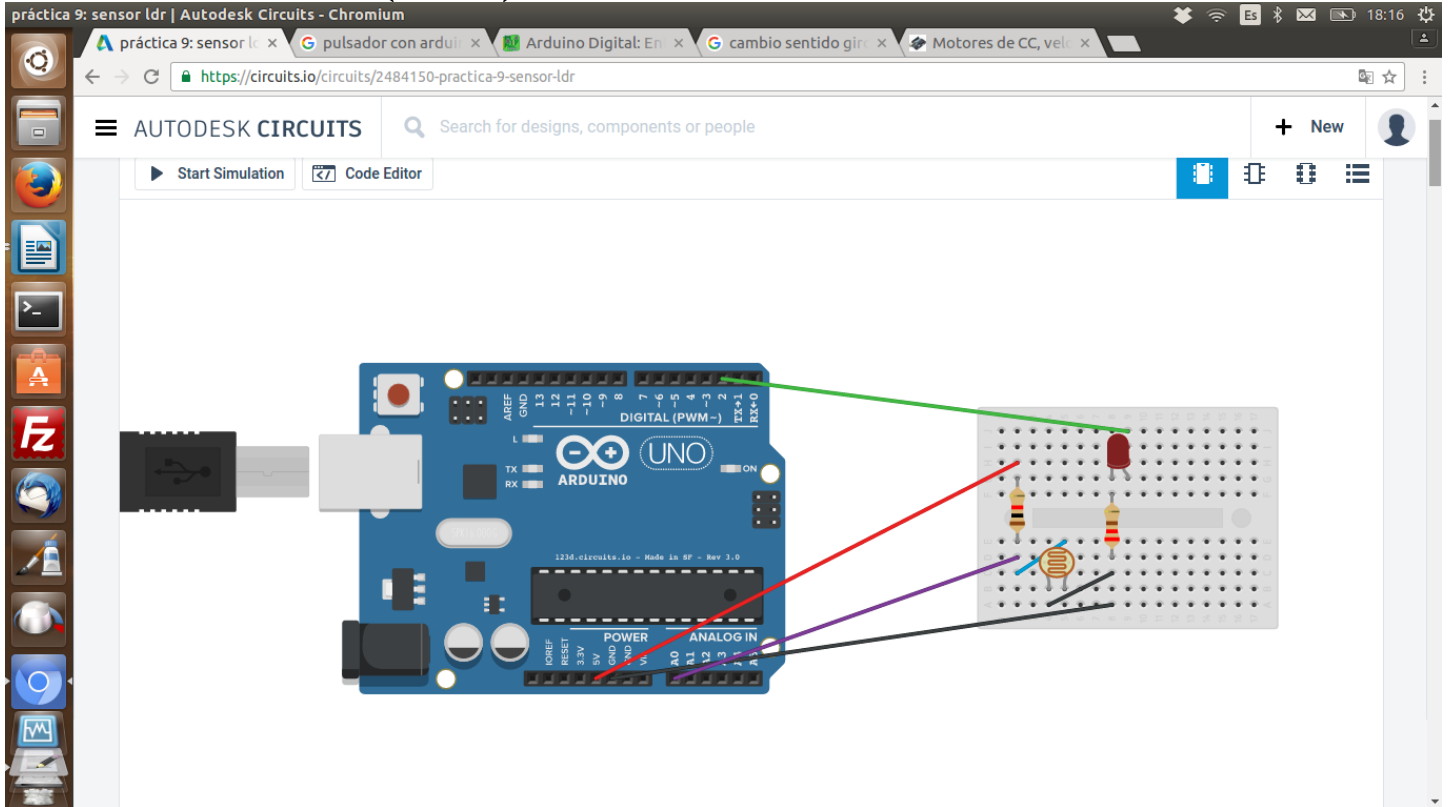
PRÁCTICA 6: PULSADOR



```
void setup ()
{
  Serial.begin(9600);
  //Configuramos el pin PULSADOR como botón de entrada de pull-up
  pinMode(4,INPUT_PULLUP);
  //Configuramos el pin LED como salida
  pinMode(3,OUTPUT);
}
void loop()
{
  int sensorVal = digitalRead(4);

  if (sensorVal == LOW) {
    digitalWrite(3, HIGH);
  }
  else {
    digitalWrite(3, LOW);
  }
}
```

PRÁCTICA 7: SENSOR LDR (DE LUZ).

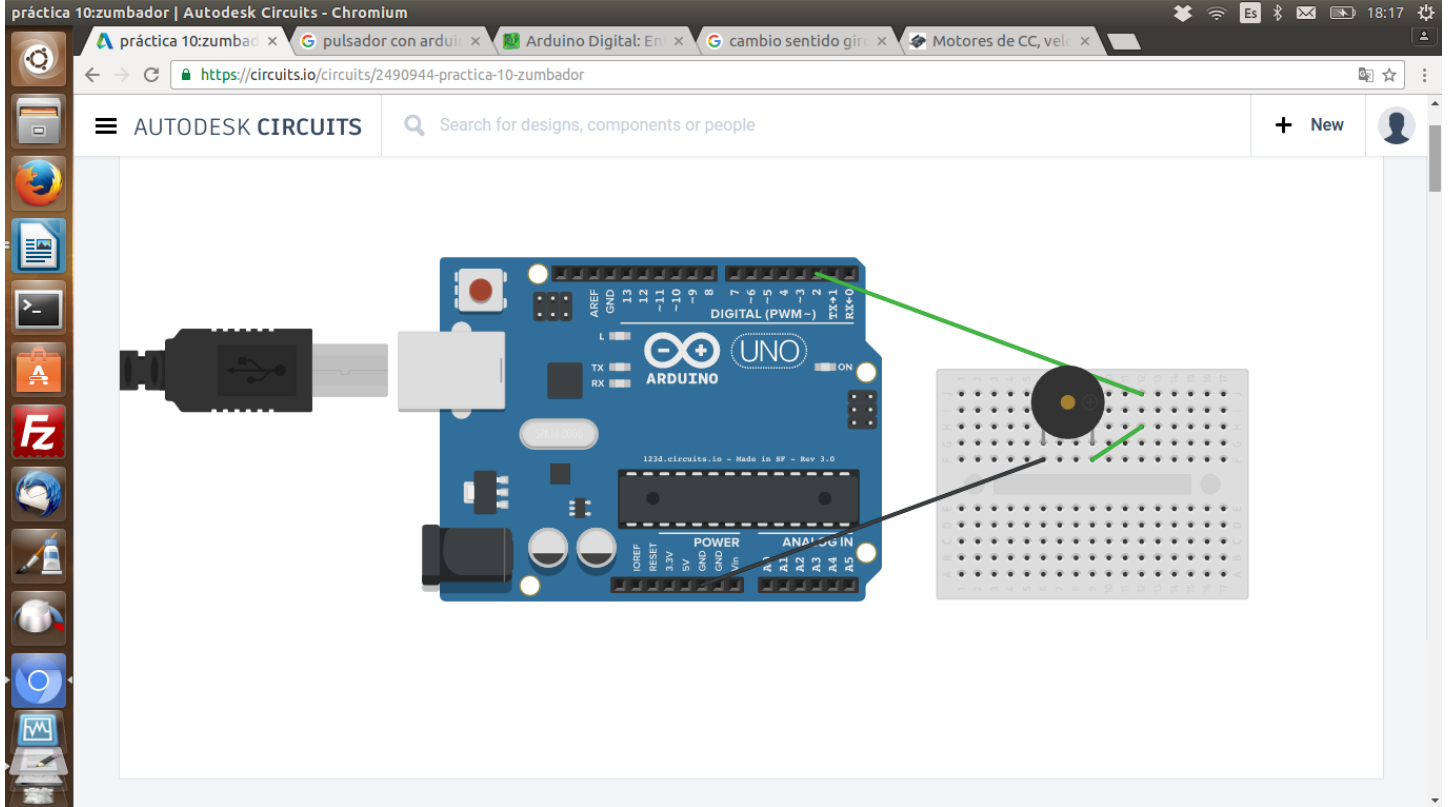


```
#define PIN_LDR A0 //Cambiamos el nombre de la entrada A0
#define PIN_BUZZER 2 //Cambiamos el nombre del pin 2

void setup()
{
  Serial.begin(9600); //inicia comunicacion serial
  pinMode(PIN_LDR,INPUT); //definimos el pin A0 como entrada
  pinMode(PIN_BUZZER,OUTPUT); //definimos el pin 2 como salida
}
void loop()
{
  int ldr =analogRead(PIN_LDR); //tomamos el valor de la LDR como entero comando int
  Serial.println(ldr); //escribe la luminosidad en el serial
  delay(500);

  if(ldr < 500) //si el valor de la ldr es menor que 500 encender led
  {
    digitalWrite(PIN_BUZZER,HIGH); //encendemos led
  }
  else //en el caso contrario , es decir siendo mayor o igual que 500 lo apagamos
  {
    digitalWrite(PIN_BUZZER,LOW); //apagamos el led
  }
}
```

PRÁCTICA 8: ZUMBADOR.

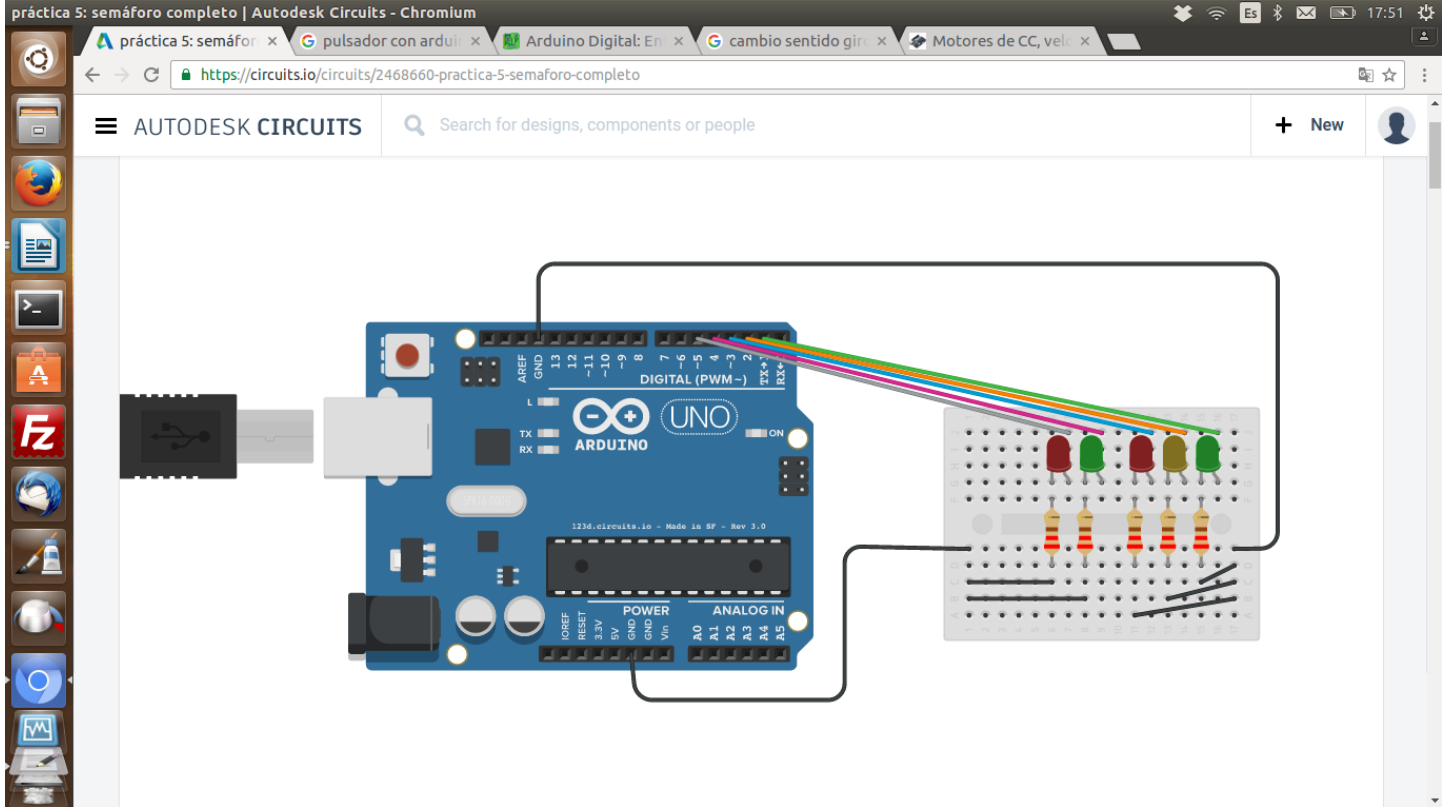


```
#define PINBUZZER 2 //definimos el pin al que conectamos el zumbador
#define TEMPO 120 //definimos el número de notas negras en un minuto
#define Do 262 //definimos las frecuencias de las notas
#define Re 294
#define Mi 320
#define Fa 349
#define Sol 392
#define La 440
#define Si 494
#define Do_alto 523
#define NEGRA 60000/TEMPO //definimos la duracion de las notas en milisegundos
#define REDONDA NEGRA*4
#define BLANCA NEGRA*2
#define CORCHEA NEGRA/2
#define SEMICORCHEA NEGRA/4
```

```
void setup()
{
  tone(PINBUZZER,Do_alto,CORCHEA);
  delay(CORCHEA*1.3); //para diferenciar entre las notas esperamos un 30%

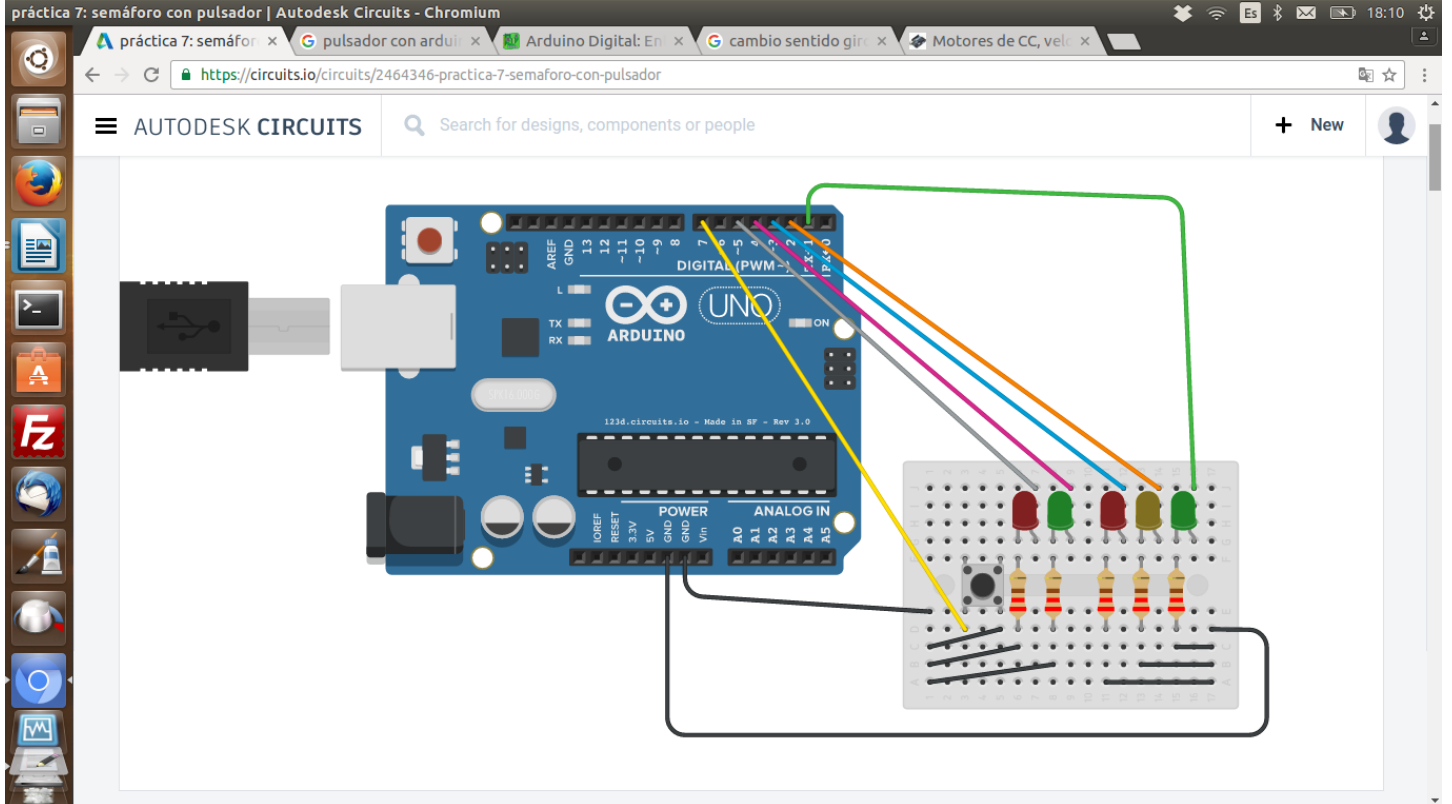
  tone(PINBUZZER,Sol,SEMICORCHEA);
  delay(SEMICORCHEA*1.3);
  tone(PINBUZZER,Sol,SEMICORCHEA);
  delay(SEMICORCHEA*1.3);
  tone(PINBUZZER,La,CORCHEA);
  delay(CORCHEA*1.3);
  tone(PINBUZZER,Sol,CORCHEA);
  delay(CORCHEA*1.3);
  noTone(PINBUZZER); //silencio
  delay(CORCHEA);
  tone(PINBUZZER,Si,CORCHEA);
  delay(CORCHEA*1.3);
  tone(PINBUZZER,Do_alto,CORCHEA);
  delay(CORCHEA*1.3);
}
void loop()
{
}
```


PRÁCTICA 9: SEMÁFORO COMPLETO



```
#define VERDE 1 //Renombramos el pin 1 a VERDE
#define AMARILLO 2 //Renombramos el pin 2 a AMARILLO
#define ROJO 3 //Renombramos el pin 3 a ROJO
#define VERDEPEATON 4 //Renombramos el pin 4 a VERDEPEATON
#define ROJOPEATON 5 //Renombramos el pin 5 a VERDEPEATON
void setup()
{
  pinMode(VERDE,OUTPUT); //configuramos el pin VERDE como salida
  pinMode(AMARILLO,OUTPUT); // configuramos el pin AMARILLO como salida
  pinMode(ROJO,OUTPUT); //configuramos el pin ROJO como salida
  pinMode(VERDEPEATON,OUTPUT); //configuramos el pin led rojo de peaton como salida
  pinMode(ROJOPEATON,OUTPUT); //configuramos el pin led rojo de peaton como salida
}
void loop()
{
  digitalWrite(VERDE,HIGH);
  digitalWrite(AMARILLO,LOW);
  digitalWrite(ROJO,LOW);
  digitalWrite(VERDEPEATON,LOW);
  digitalWrite(ROJOPEATON,HIGH);
  delay(10000);
  digitalWrite(VERDE,LOW);
  digitalWrite(AMARILLO,HIGH);
  digitalWrite(ROJO,LOW);
  digitalWrite(VERDEPEATON,LOW);
  digitalWrite(ROJOPEATON,HIGH);
  delay(5000);
  digitalWrite(VERDE,LOW);
  digitalWrite(AMARILLO,LOW);
  digitalWrite(ROJO,HIGH);
  digitalWrite(VERDEPEATON,HIGH);
  digitalWrite(ROJOPEATON,LOW);
  delay(5000);
  digitalWrite(VERDE,LOW);
  digitalWrite(AMARILLO,LOW);
  digitalWrite(ROJO,HIGH);
  digitalWrite(VERDEPEATON,LOW);
  digitalWrite(ROJOPEATON,LOW);
  delay(500);
}
```

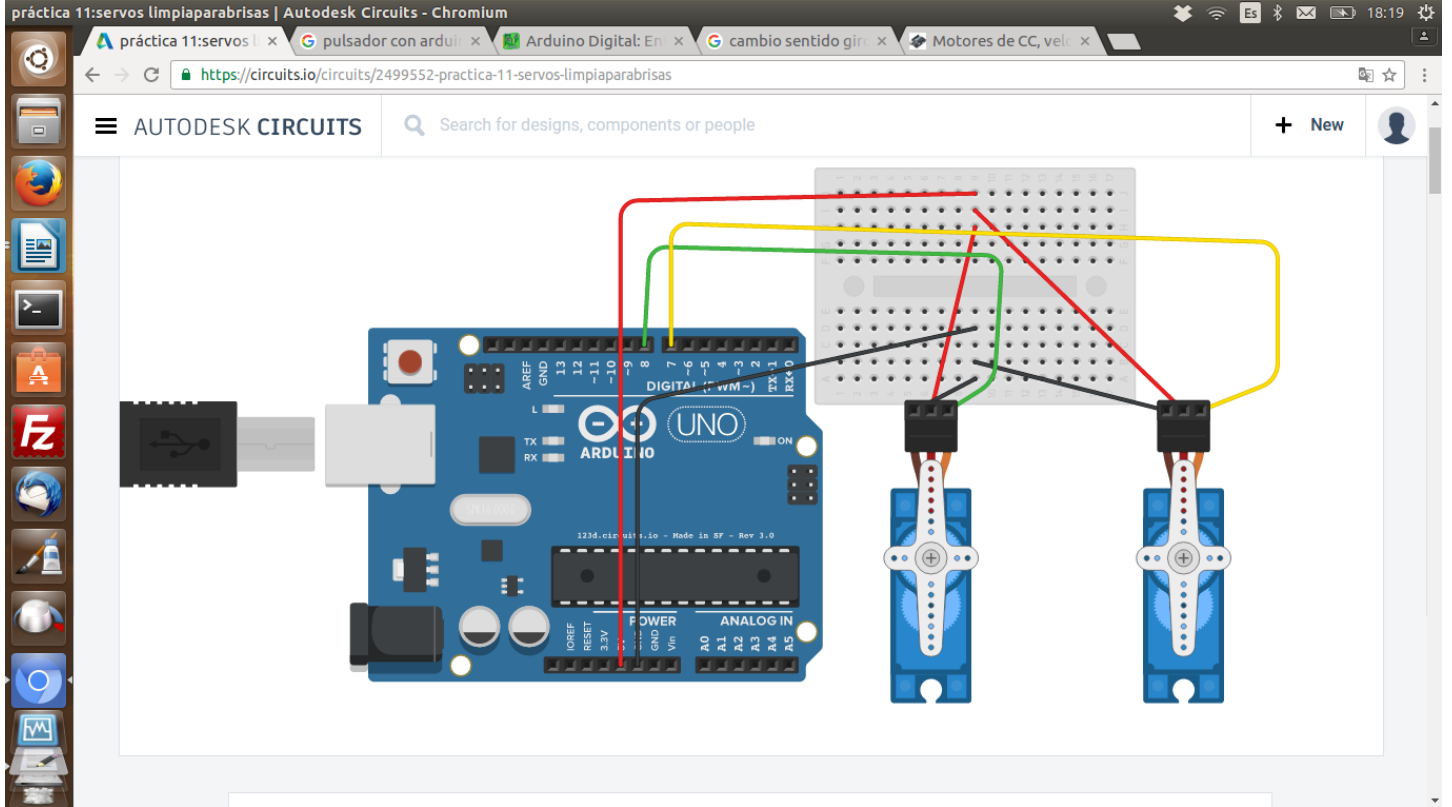

PRÁCTICA 10: SEMÁFORO CON PULSADORES.



```
#define VERDE 1 //Renombramos el pin 1 a VERDE
#define AMARILLO 2 //Renombramos el pin 2 a AMARILLO
#define ROJO 3 //Renombramos el pin 3 a ROJO
#define VERDEPEATON 4 //Renombramos el pin 4 a VERDEPEATON
#define ROJOPEATON 5 //Renombramos el pin 5 a VERDEPEATON
#define PULSADOR 7 //Renombramos el pin 7 a PULSADOR
```

```
void setup ()
{
  pinMode(VERDE,OUTPUT); //configuramos el pin VERDE como salida
  pinMode(AMARILLO,OUTPUT); // configuramos el pin AMARILLO como salida
  pinMode(ROJO,OUTPUT); //configuramos el pin ROJO como salida
  pinMode(VERDEPEATON,OUTPUT); //configuramos el pin led rojo de peaton como salida
  pinMode(ROJOPEATON,OUTPUT); //configuramos el pin led rojo de peaton como salida
  pinMode(PULSADOR,INPUT_PULLUP); //configuramos el pulsador como entrada
}
void loop()
{
  //Guardamos en una variable la lectura del boton
  // En pull-up guardara 1 no pulsado, o pulsado
  int boton = digitalRead(PULSADOR);
  //Usamos una condicional para preguntar si el boton esta pulsado
  if (boton == 1)
  {
    digitalWrite(VERDE,HIGH);
    digitalWrite(AMARILLO,LOW);
    digitalWrite(ROJO,LOW);
    digitalWrite(VERDEPEATON,LOW);
    digitalWrite(ROJOPEATON,HIGH);
    delay(10000);
  }
  else
  {
    digitalWrite(VERDE,LOW);
    digitalWrite(AMARILLO,HIGH);
    digitalWrite(ROJO,LOW);
    digitalWrite(VERDEPEATON,LOW);
    digitalWrite(ROJOPEATON,HIGH);
    delay(5000);
    digitalWrite(VERDE,LOW);
    digitalWrite(AMARILLO,LOW);
    digitalWrite(ROJO,HIGH);
    digitalWrite(VERDEPEATON,HIGH);
    digitalWrite(ROJOPEATON,LOW);
    delay(5000);
  }
}
```


PRÁCTICA 12: LIMPIA CRISTALES CON SERVOMOTORES



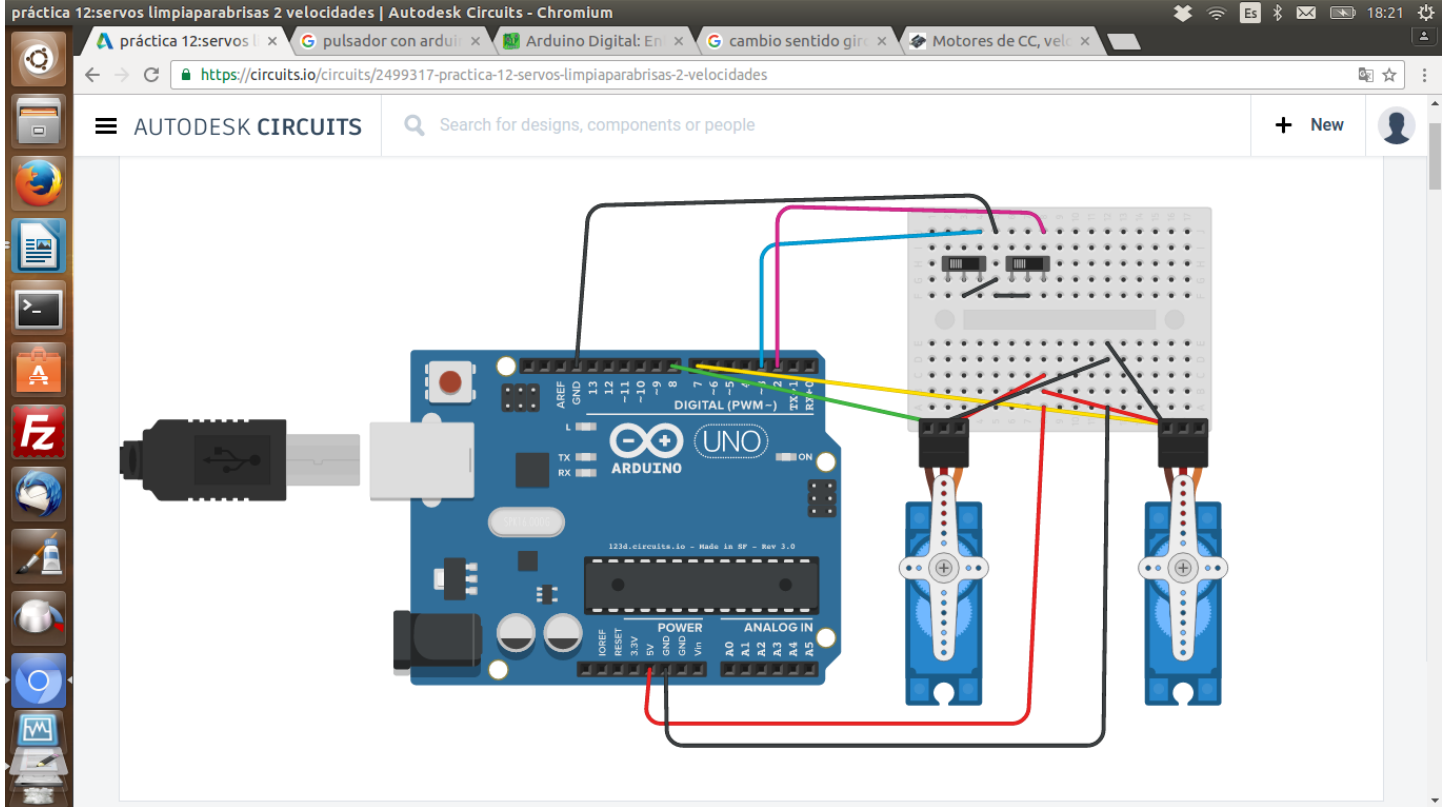
```
#include <Servo.h> //incluye la librería servo
Servo servoX; //llamamos al servo 1 servo X
Servo servoY; //llamamos al servo 2 servo Y
void setup(){
  servoX.attach(7); //le damos señal al servo X con el pin 8
  servoY.attach(8); // le damos señal al servo Y con el pin 9
  servoX.write(0); //le damos orden de reinicio a 0 grados
  servoY.write(0);
}

void loop(){

  servoX.write(150); //le decimos que el servo X vaya a la posición de 150 grados
  servoY.write(150); //le decimos que el servo X vaya a la posición de 150 grados
  delay(1500);
  servoX.write(0); //le decimos que el servo X vaya a la posición de 0 grados
  servoY.write(0); //le decimos que el servo X vaya a la posición de 0 grados
  delay(1500);

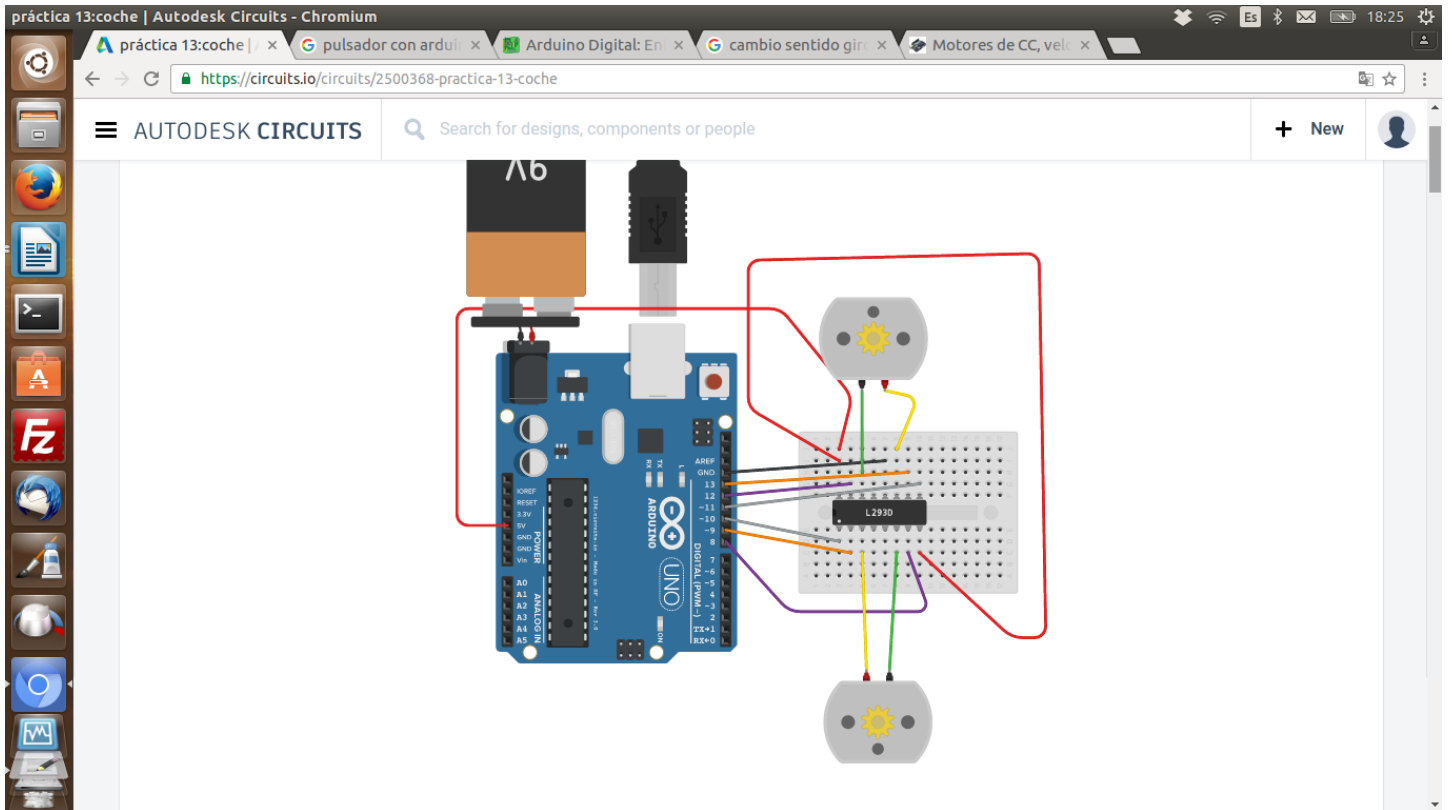
}
```

PRÁCTICA 13: LIMPIACRISTALES CON DOS VELOCIDADES.



```
#include <Servo.h> //incluye la librería servo
Servo servoX; //llamamos al servo 1 servo X
Servo servoY; //llamamos al servo 2 servo Y
void setup(){
  servoX.attach(7); //le damos señal al servo X con el pin 8
  servoY.attach(8); // le damos señal al servo Y con el pin 9
  servoX.write(0); //le damos orden de reinicio a 0 grados
  servoY.write(0);
  pinMode(2,INPUT_PULLUP);
  pinMode(3,INPUT_PULLUP);
}
void loop(){
  int boton1=digitalRead(2);
  int boton2=digitalRead(3);
  if (boton1==0)
  {
    servoX.write(150); //le decimos que el servo X vaya a la posición de 150 grados
    servoY.write(150); //le decimos que el servo X vaya a la posición de 150 grados
    delay(2000);
    servoX.write(0); //le decimos que el servo X vaya a la posición de 0 grados
    servoY.write(0); //le decimos que el servo X vaya a la posición de 0 grados
    delay(2000);
  }
  else if (boton2==0)
  {
    servoX.write(150); //le decimos que el servo X vaya a la posición de 150 grados
    servoY.write(150); //le decimos que el servo X vaya a la posición de 150 grados
    delay(1000);
    servoX.write(0); //le decimos que el servo X vaya a la posición de 0 grados
    servoY.write(0); //le decimos que el servo X vaya a la posición de 0 grados
    delay(1000);
  }
  else
  {
    servoX.write(0); //le decimos que el servo X vaya a la posición de 150 grados
    servoY.write(0); //le decimos que el servo X vaya a la posición de 150 grados
    delay(1000);
  }
}
```

PRÁCTICA 14: COCHE (AQUÍ HAY VARIAS PRÁCTICAS JUNTAS, PRIMERO DIBUJO SIMULADOR Y CON PINES CONTROLES DE VELOCIDAD PERO ESTOS DOS A VECES DAN PROBLEMAS Y POR ELLOS TENGO LAS PRÁCTICAS SIN ELLOS)



ENCHUFAR LA FUENTE Y LUEGO SIN DESCONECTARLA CARGAR EL CÓDIGO CON EL USB (DA PROBLEMAS CON EL ARDUINO GENUINO DEL INSTITUTO Y EL L293D) CONECTAR LA CONEXIÓN VIN ANTIGUA DEL CHIP A 5V.

PRACTICA 1 COCHE: ADELANTE Y ATRÁS

```
#define I1 8 // pin 1 motor 1
#define I2 9 // pin 2 motor 1
#define I3 12 // pin 1 motor 2
#define I4 13 // pin 2 motor 2
```

```
void setup() {
  pinMode(I1, OUTPUT);
  pinMode(I2, OUTPUT);
  pinMode(I3, OUTPUT);
  pinMode(I4, OUTPUT);
}
```

```
void loop() {
  digitalWrite(I1, HIGH);
  digitalWrite(I2, LOW);
  digitalWrite(I3, HIGH);
  digitalWrite(I4, LOW);
  delay(2000);
```

```
digitalWrite(I1, LOW); //ahora para cambiar el sentido de giro cambiamos la polaridad
digitalWrite(I2, HIGH); //corriente y viceversa
digitalWrite(I3, LOW);
digitalWrite(I4, HIGH);
delay(2000);
}
```


PRÁCTICA 2 COCHE: ADELANTE, PARO, ATRÁS, PARO

```
#define I1 8 // pin 1 motor 1
#define I2 9 // pin 2 motor 1
#define I3 12 // pin 1 motor 2
#define I4 13 // pin 2 motor 2
```

```
void setup() {
  pinMode(I1, OUTPUT);
  pinMode(I2, OUTPUT);
  pinMode(I3, OUTPUT);
  pinMode(I4, OUTPUT);
}
```

```
void loop() {
  digitalWrite(I1, HIGH);
  digitalWrite(I2, LOW);
  digitalWrite(I3, HIGH);
  digitalWrite(I4, LOW);
  delay(2000);
```

```
  digitalWrite(I1, LOW);
  digitalWrite(I2, LOW);
  digitalWrite(I3, LOW);
  digitalWrite(I4, LOW);
  delay(1000);
```

```
  digitalWrite(I1, LOW); //ahora para cambiar el sentido de giro cambiamos la polaridad a los motores por
  digitalWrite(I2, HIGH); //corriente y viceversa
  digitalWrite(I3, LOW);
  digitalWrite(I4, HIGH);
  delay(2000);
```

```
  digitalWrite(I1, LOW);
  digitalWrite(I2, LOW);
  digitalWrite(I3, LOW);
  digitalWrite(I4, LOW);
  delay(1000);
}
```

PRÁCTICA COCHE 3 (PARA DELANTE -VUELTA ----REGRESO)

```
#define I1 8 // pin 1 motor 1
#define I2 9 // pin 2 motor 1
#define I3 12 // pin 1 motor 2
#define I4 13 // pin 2 motor 2
```

```
void setup() {
  pinMode(I1, OUTPUT);
  pinMode(I2, OUTPUT);
  pinMode(I3, OUTPUT);
  pinMode(I4, OUTPUT);
}
```

```
void loop() {
  digitalWrite(I1, HIGH);
```

```
digitalWrite(I2, LOW);
digitalWrite(I3, HIGH);
digitalWrite(I4, LOW);
delay(2000);
```

```
digitalWrite(I1, LOW);
digitalWrite(I2, LOW);
digitalWrite(I3, LOW);
digitalWrite(I4, LOW);
delay(1000);
```

```
digitalWrite(I1, LOW);
digitalWrite(I2, HIGH);
digitalWrite(I3, HIGH);
digitalWrite(I4, HIGH);
delay(1200);
```

```
digitalWrite(I1, LOW);
digitalWrite(I2, LOW);
digitalWrite(I3, LOW);
digitalWrite(I4, LOW);
delay(1000);
}
```

PRÁCTICA 4 COCHE: GIRO EN CÍRCULO

```
#define I1 8 // pin 1 motor 1
#define I2 9 // pin 2 motor 1
#define I3 12 // pin 1 motor 2
#define I4 13 // pin 2 motor 2
```

```
void setup() {
  pinMode(I1, OUTPUT);
  pinMode(I2, OUTPUT);
  pinMode(I3, OUTPUT);
  pinMode(I4, OUTPUT);
}
```

```
void loop() {
  digitalWrite(I1, HIGH);
  digitalWrite(I2, LOW);
  digitalWrite(I3, LOW);
  digitalWrite(I4, LOW);
  delay(4000);
```

```
digitalWrite(I1, LOW);
digitalWrite(I2, LOW);
digitalWrite(I3, LOW);
digitalWrite(I4, LOW);
delay(1000);
```

```
digitalWrite(I1, LOW);
digitalWrite(I2, LOW);
digitalWrite(I3, HIGH);
digitalWrite(I4, LOW);
```

```
delay(4000);
```

```
digitalWrite(I1, LOW);  
digitalWrite(I2, LOW);  
digitalWrite(I3, LOW);  
digitalWrite(I4, LOW);  
delay(1000);  
}
```

PRÁCTICA 5 COCHE:SERPIENTE, AVANCE ALTERNATIVO DE CADA UNO DE LOS MOTORES

```
#define I1 8 // pin 1 motor 1  
#define I2 9 // pin 2 motor 1  
#define I3 12 // pin 1 motor 2  
#define I4 13 // pin 2 motor 2
```

```
void setup() {  
  pinMode(I1, OUTPUT);  
  pinMode(I2, OUTPUT);  
  pinMode(I3, OUTPUT);  
  pinMode(I4, OUTPUT);  
}
```

```
void loop() {  
  digitalWrite(I1, HIGH);  
  digitalWrite(I2, LOW);  
  digitalWrite(I3, LOW);  
  digitalWrite(I4, LOW);  
  delay(1000);
```

```
  digitalWrite(I1, LOW);  
  digitalWrite(I2, LOW);  
  digitalWrite(I3, HIGH);  
  digitalWrite(I4, LOW);  
  delay(1000);  
}
```

PRÁCTICA 6 COCHE:DESCRIBIR UN CUADRADO

```
#define I1 8 // pin 1 motor 1  
#define I2 9 // pin 2 motor 1  
#define I3 12 // pin 1 motor 2  
#define I4 13 // pin 2 motor 2
```

```
void setup() {  
  pinMode(I1, OUTPUT);  
  pinMode(I2, OUTPUT);  
  pinMode(I3, OUTPUT);  
  pinMode(I4, OUTPUT);  
}
```

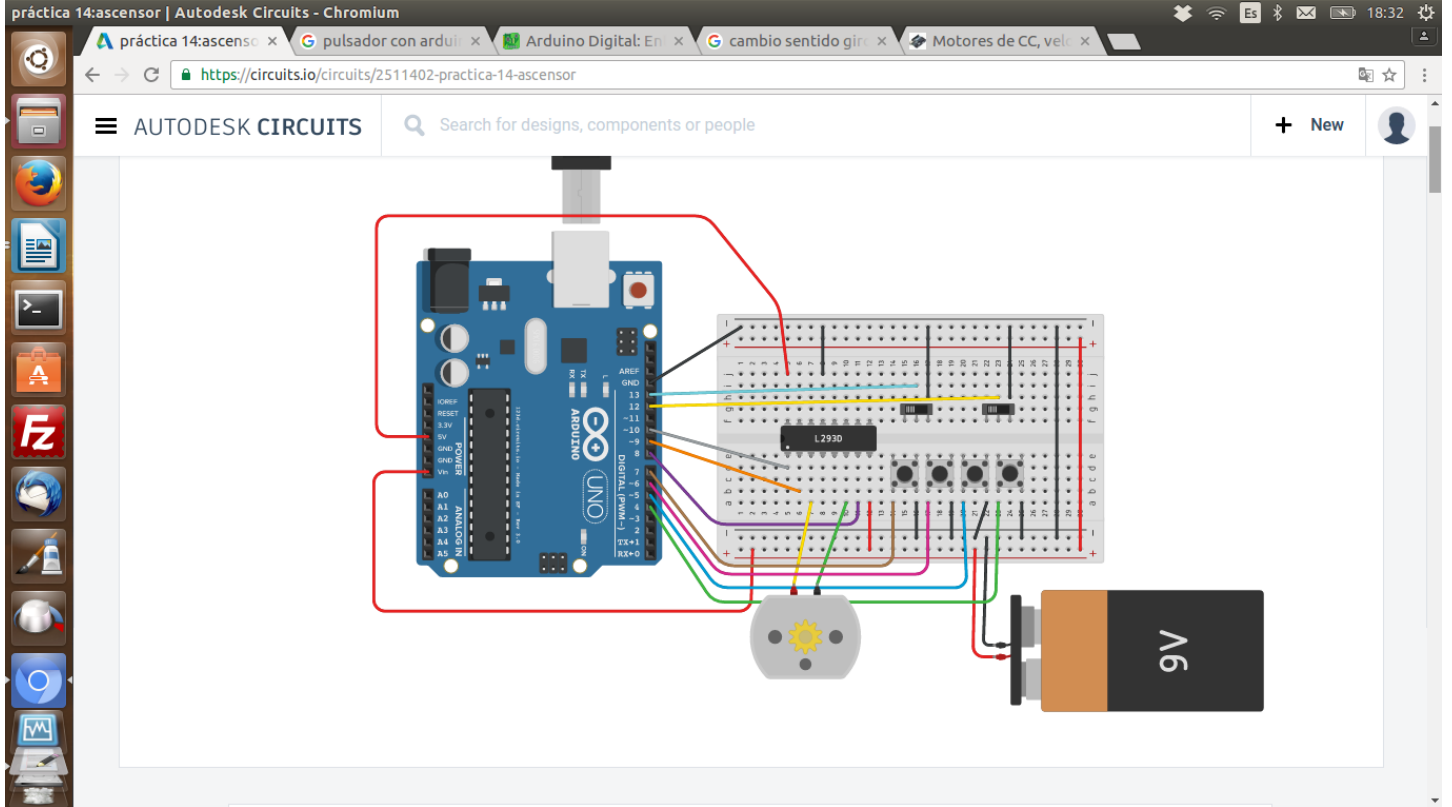
```
void loop() {  
  digitalWrite(I1, HIGH);  
  digitalWrite(I2, LOW);  
  digitalWrite(I3, HIGH);  
  digitalWrite(I4, LOW);  
  delay(2000);
```

```
digitalWrite(I1, LOW);  
digitalWrite(I2, LOW);  
digitalWrite(I3, LOW);  
digitalWrite(I4, LOW);  
delay(1000);
```

```
digitalWrite(I1, LOW);  
digitalWrite(I2, HIGH);  
digitalWrite(I3, LOW);  
digitalWrite(I4, LOW);  
delay(750);
```

```
digitalWrite(I1, LOW);  
digitalWrite(I2, LOW);  
digitalWrite(I3, LOW);  
digitalWrite(I4, LOW);  
delay(1000);  
}
```

PRÁCTICA 15:ASCENSOR COMPLETO.



```
#define E1 10 // Pin para controlar la velocidad de motor 1
#define LLAMADAB 7 //Pin llamada para bajar
#define BAJA 6 // Pin para bajar
#define LLAMADAS 5//Pin llamada para subir
#define SUBE 4 //Pin para subir
#define FINBAJO 13//Pin final de carrera bajo
#define FINARRIBA 12//Pin final de carrera arriba
```

```
#define I1 8 // pin 1 motor 1
#define I2 9 // pin 2 motor 1
```

```
void setup() {
  pinMode(E1, OUTPUT);
  pinMode(I1, OUTPUT);
  pinMode(I2, OUTPUT);
  pinMode(LLAMADAB,INPUT_PULLUP);
  pinMode(LLAMADAS,INPUT_PULLUP);
  pinMode(BAJA,INPUT_PULLUP);
  pinMode(SUBE,INPUT_PULLUP);
  pinMode(FINBAJO,INPUT_PULLUP);
  pinMode(FINARRIBA,INPUT_PULLUP);
}
```

```
void loop() {
  int llamabaja = digitalRead(LLAMADAB);
  int llamasube = digitalRead(LLAMADAS);
  int baja = digitalRead(BAJA);
  int sube = digitalRead(SUBE);
  int finbajo = digitalRead(FINBAJO);
  int finarriba = digitalRead(FINARRIBA);
```

```
if (finbajo==0 && sube==0) //entro pulso subir y subo
{
  analogWrite(E1,150); // velocidad del motor 1 150
  digitalWrite(I1, HIGH);
  digitalWrite(I2, LOW);
  delay(1000);
}

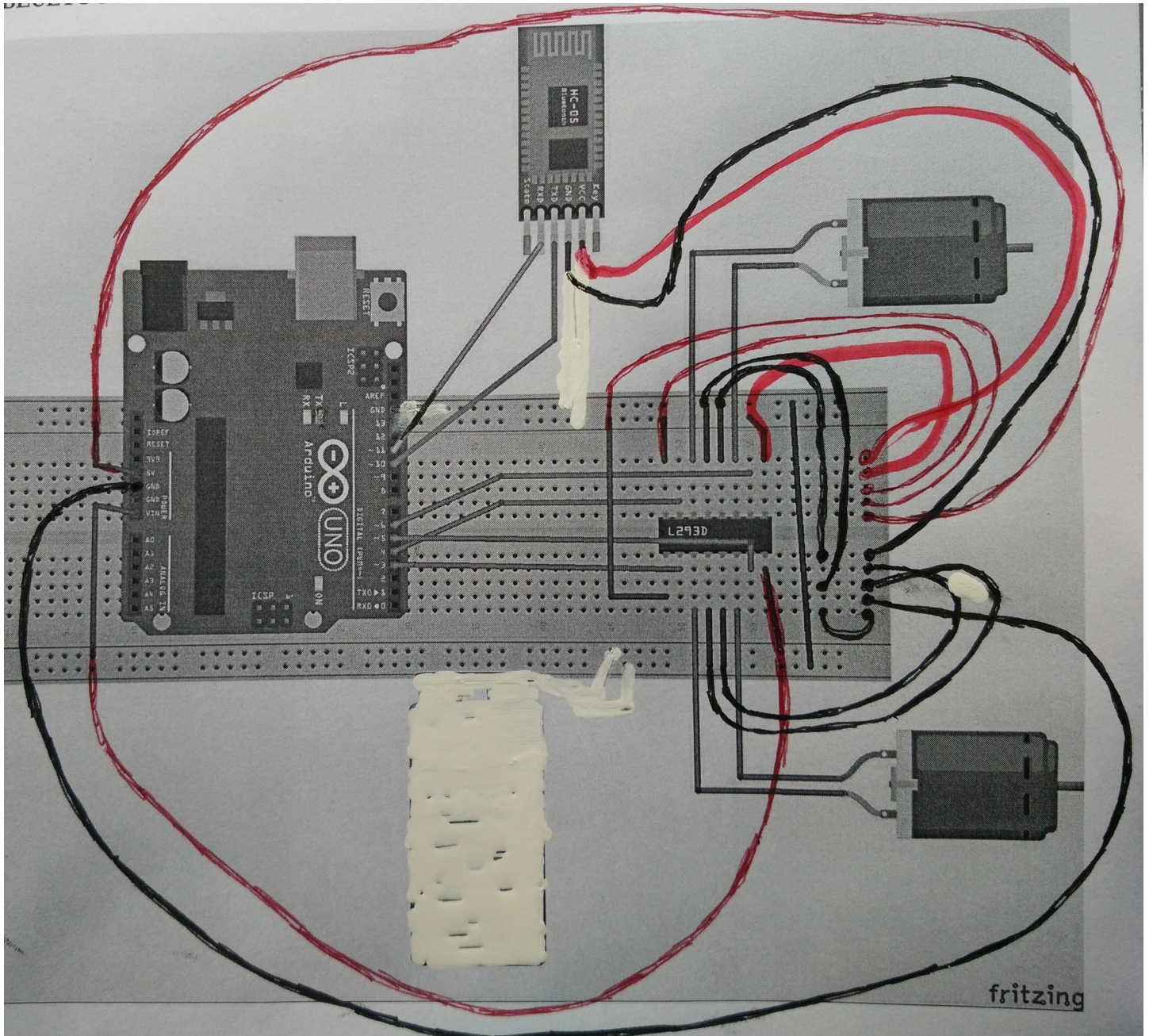
else if (finarriba==0 && baja==0) //entro pulso bajar y bajo
{
  analogWrite(E1,150); // velocidad del motor 1 150
  digitalWrite(I1, LOW);
  digitalWrite(I2, HIGH);
  delay(1000);
}

else if (finarriba==0 && llamabaja==0) //la cabina está arriba, pulso baja y lo hace
{
  analogWrite(E1,150); // velocidad del motor 1 150
  digitalWrite(I1, LOW);
  digitalWrite(I2, HIGH);
  delay(1000);
}

else if (finbajo==0 && llamasube==0) //la cabina está a bajo, la llamo y sube
{
  analogWrite(E1,150); // velocidad del motor 1 150
  digitalWrite(I1, HIGH);
  digitalWrite(I2, LOW);
  delay(1000);
}

else //para cualquier otro caso como solo finarriba o finbajo, paro
{
  analogWrite(E1,0); // paro
  digitalWrite(I1, LOW);
  digitalWrite(I2, HIGH);
  delay(1000);
}
}
```

PRÁCTICA 16: BLUETOOTH: CONFIGURAMOS EL SENSOR BLUETOOTH CON LOS PINES 10 Y 11.



```
#include <SoftwareSerial.h>
```

```
SoftwareSerial miPuertoSerie=SoftwareSerial(10,11);
```

```
int unChar;
```

```
void setup(){
```

```
  miPuertoSerie.begin(9600);
```

```
  pinMode(3, OUTPUT);
```

```
  pinMode(4, OUTPUT);
```

```
  pinMode(5, OUTPUT);
```

```
  pinMode(6, OUTPUT);
```

```
}
```

```
void loop(){
```

```
  char unChar=miPuertoSerie.read();
```

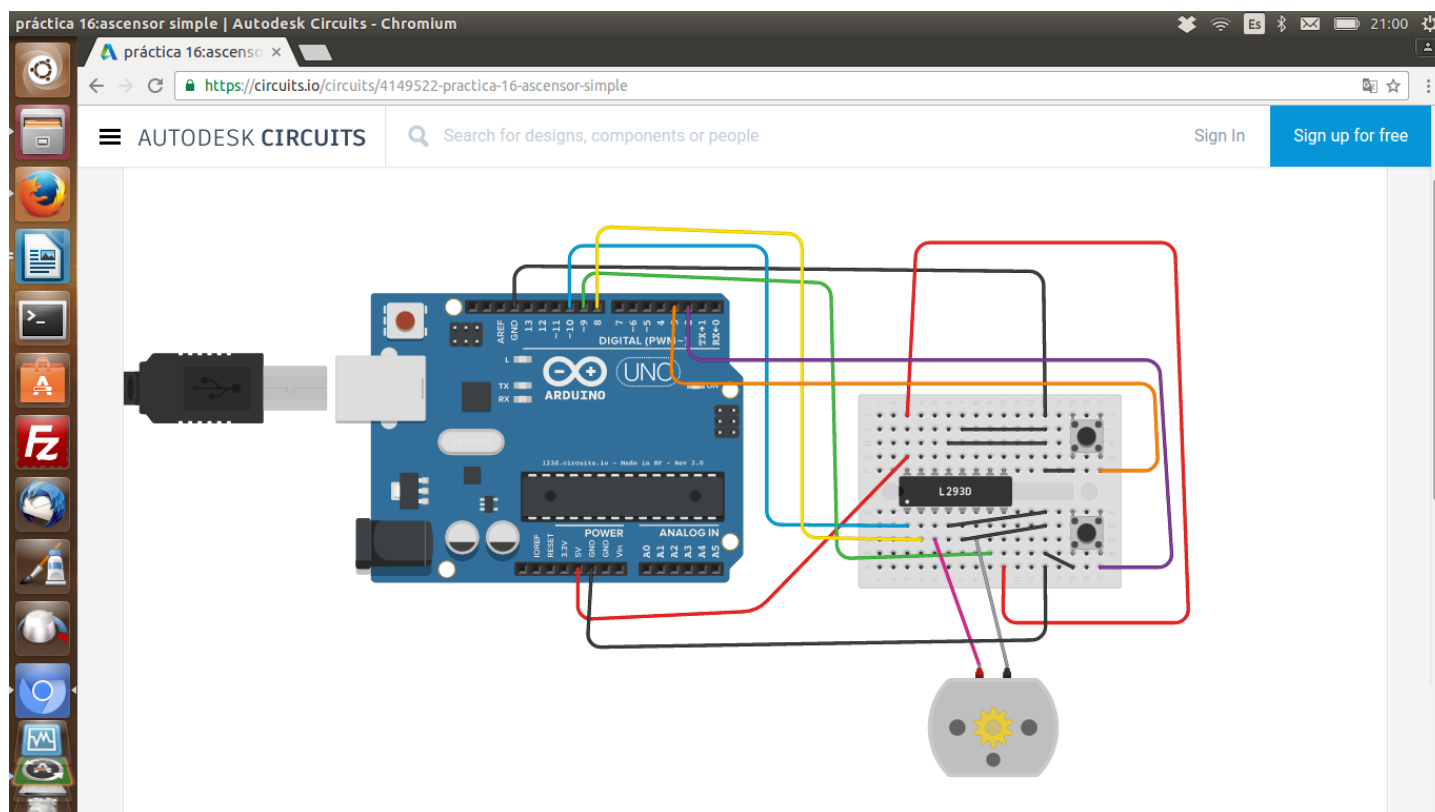
```
  if (unChar>0){
```

```
    switch (unChar){
```

```
    case '2':
        digitalWrite(3, HIGH);
        digitalWrite (4, HIGH);
        digitalWrite(5,LOW);
        digitalWrite(6,LOW);
        delay(100);
        break;
case '1':
    digitalWrite (3,HIGH);
    digitalWrite (4,LOW);
    digitalWrite (5,LOW);
    digitalWrite (6,LOW);
    delay(100);
    break;
case '3':
    digitalWrite (3, LOW);
    digitalWrite (4, HIGH);
    digitalWrite (5, LOW);
    digitalWrite (6, LOW);
    delay(100);
    break;
case '0':
    digitalWrite (3, LOW);
    digitalWrite (4, LOW);
    digitalWrite (5, LOW);
    digitalWrite (6, LOW);
    delay(100);
    break;
case '9':
    digitalWrite(3, LOW);
    digitalWrite(4, LOW);
    digitalWrite(5, HIGH);
    digitalWrite(6,HIGH);
    delay(100);
    break;

}
}
}
```


PRÁCTICA 17:ASCENSOR SENCILLO.



```
#define E1 10 // Pin para controlar la velocidad de motor 1
#define I1 8 // pin 1 motor 1
#define I2 9 // pin 2 motor 1
```

```
void setup() {
  Serial.begin(9600);
  pinMode(2, INPUT_PULLUP);
  pinMode(3, INPUT_PULLUP);
```

```
  pinMode(E1, OUTPUT);
  pinMode(I1, OUTPUT);
  pinMode(I2, OUTPUT);
```

```
}
```

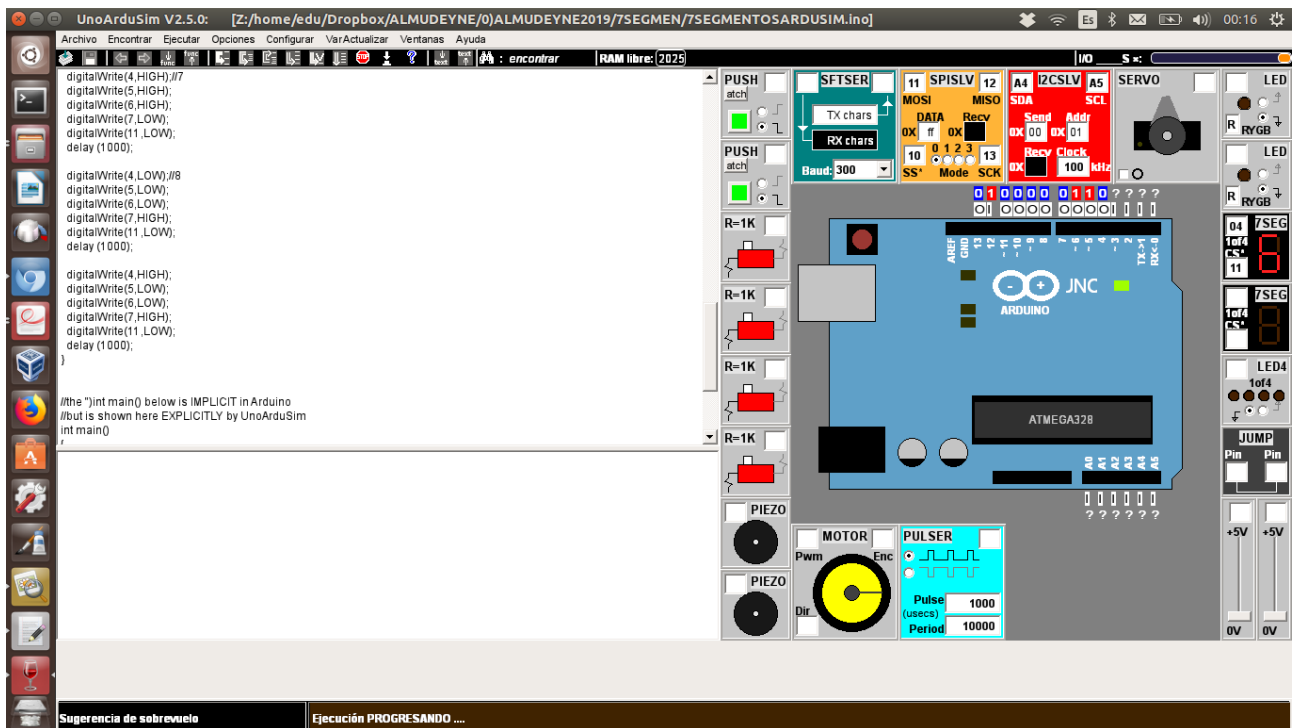
```
void loop() {
  int valorpuls1 = digitalRead(2);
  int valorpuls2 = digitalRead(3);
  Serial.println(valorpuls1);
  Serial.println(valorpuls2);
```

```
  if (valorpuls1 == LOW) {
    analogWrite(E1, 150); // velocidad del motor 1 150
    digitalWrite(I1, HIGH);
    digitalWrite(I2, LOW);
    delay(2000);
  }
```

```
  if (valorpuls2 == LOW) {
```

```
analogWrite(E1, 150); // de damos velocidad de giro al motor 1 150
digitalWrite(I1, LOW); //ahora para cambiar el sentido de giro cambiamos la polaridad a los motores
digitalWrite(I2, HIGH); //corriente y viceversa
delay(2000);
}
else {
analogWrite(E1, 0); // de damos velocidad de giro al motor 1 150
digitalWrite(I1, LOW); //ahora para cambiar el sentido de giro cambiamos la polaridad a los motores
digitalWrite(I2, LOW); //corriente y viceversa
delay(1000);
}
}
```

PRÁCTICA DISPLAY 7 SEGMENTOS



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

```
void setup ()
{
    pinMode(4,OUTPUT);
    pinMode(5,OUTPUT);
    pinMode(6,OUTPUT);
    pinMode(7,OUTPUT);
    pinMode(8,OUTPUT);
    pinMode(9,OUTPUT);
    pinMode(10,OUTPUT);
    pinMode(11,OUTPUT);
}
void loop()
{
    digitalWrite(4,LOW);//0
    digitalWrite(5,LOW);
    digitalWrite(6,LOW);
    digitalWrite(7,LOW);
    digitalWrite(11,LOW);
    delay (1000);

    digitalWrite(4,HIGH);//1
    digitalWrite(5,LOW);
    digitalWrite(6,LOW);
    digitalWrite(7,LOW);
    digitalWrite(11,LOW);
    delay (1000);

    digitalWrite(4,LOW);//2
    digitalWrite(5,HIGH);
    digitalWrite(6,LOW);
    digitalWrite(7,LOW);
    digitalWrite(11,LOW);
    delay (1000);

    digitalWrite(4,HIGH);//3
    digitalWrite(5,HIGH);
    digitalWrite(6,LOW);
    digitalWrite(7,LOW);
    digitalWrite(11,LOW);
    delay (1000);

    digitalWrite(4,LOW);//4
    digitalWrite(5,LOW);
    digitalWrite(6,HIGH);
    digitalWrite(7,LOW);
    digitalWrite(11,LOW);
    delay (1000);

    digitalWrite(4,HIGH);//5
    digitalWrite(5,LOW);
    digitalWrite(6,HIGH);
```

```
digitalWrite(7,LOW);  
digitalWrite(11,LOW);  
delay (1000);
```

```
digitalWrite(4,LOW);//6  
digitalWrite(5,HIGH);  
digitalWrite(6,HIGH);  
digitalWrite(7,LOW);  
digitalWrite(11,LOW);  
delay (1000);
```

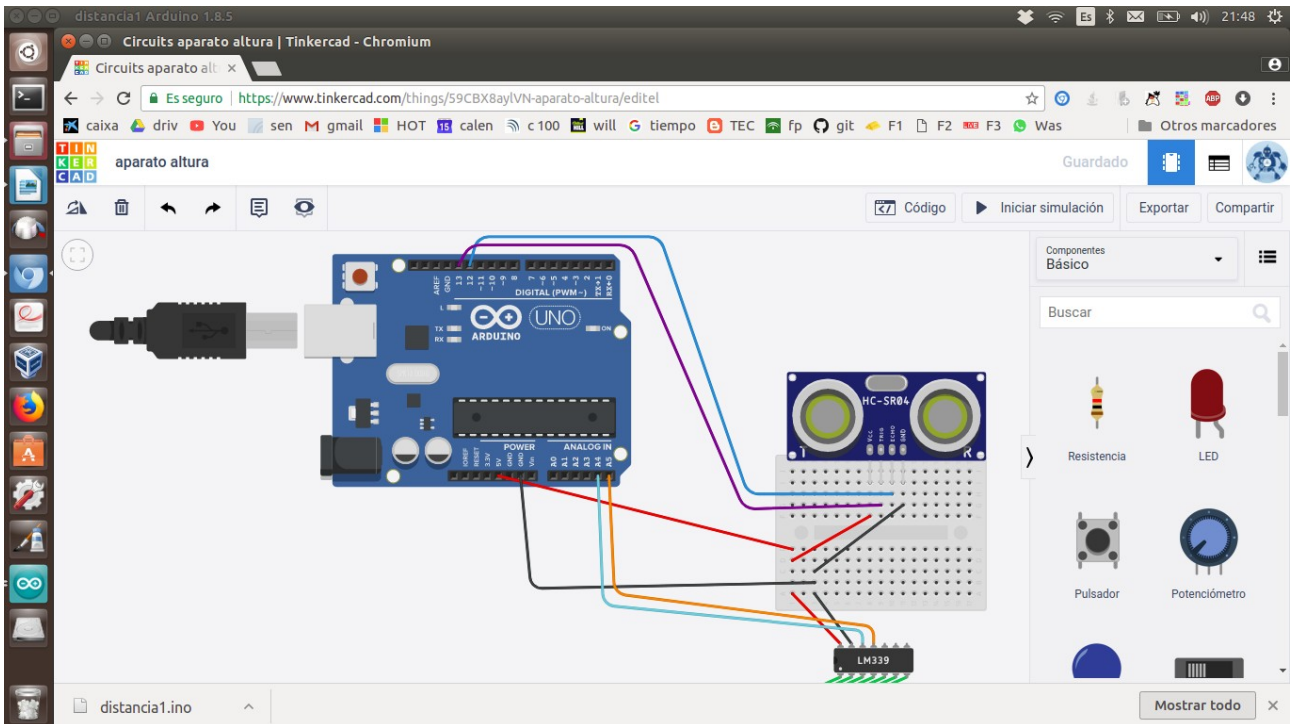
```
digitalWrite(4,HIGH);//7  
digitalWrite(5,HIGH);  
digitalWrite(6,HIGH);  
digitalWrite(7,LOW);  
digitalWrite(11,LOW);  
delay (1000);
```

```
digitalWrite(4,LOW);//8  
digitalWrite(5,LOW);  
digitalWrite(6,LOW);  
digitalWrite(7,HIGH);  
digitalWrite(11,LOW);  
delay (1000);
```

```
digitalWrite(4,HIGH);  
digitalWrite(5,LOW);  
digitalWrite(6,LOW);  
digitalWrite(7,HIGH);  
digitalWrite(11,LOW);  
delay (1000);
```

```
}
```

MEDIDOR DE ALTURA



//código aparato altura

```
#include <Wire.h> // librería de comunicacion por I2C
#include <LCD.h> // librería para funciones de LCD
#include <LiquidCrystal_I2C.h> // librería para LCD por I2C
long distancia0;
long tiempo;
long distancia1;
long distancia2;
```

```
LiquidCrystal_I2C lcd (0x27, 2, 1, 0, 4, 5, 6, 7); // DIR, E, RW, RS, D4, D5, D6, D7, configura los pines del display
```

```
void setup()
{
  Serial.begin (9600); //inicia comunicacion serial
  pinMode(13, OUTPUT); /*activación del pin 13 como salida: para el pulso ultrasónico*/
  pinMode(12, INPUT); /*activación del pin 12 como entrada: tiempo del rebote del ultrasonido*/
```

```
  lcd.setBacklightPin(3,POSITIVE); // puerto P3 de PCF8574 como positivo
  lcd.setBacklight(HIGH); // habilita iluminación posterior de LCD
  lcd.begin(16, 2); // 16 columnas por 2 líneas para LCD 1602A
  lcd.clear(); // limpia pantalla
  Serial.begin (9600); //inicia comunicación serial
```

```

}
void loop()
{
digitalWrite(13,LOW); /* Por cuestión de estabilización del sensor*/
delayMicroseconds(5);
digitalWrite(13, HIGH); /* envío del pulso ultrasónico*/
delayMicroseconds(10);
tiempo=pulseIn(12, HIGH); /* Función para medir la longitud del pulso entrante. Mide el tiempo
que transcurrido entre el envío del pulso ultrasónico y cuando el sensor recibe el rebote, es decir:
desde que el pin 12 empieza a recibir el rebote, HIGH, hasta que deja de hacerlo, LOW, la longitud
del pulso entrante*/
distancia0= int(0.017*tiempo);

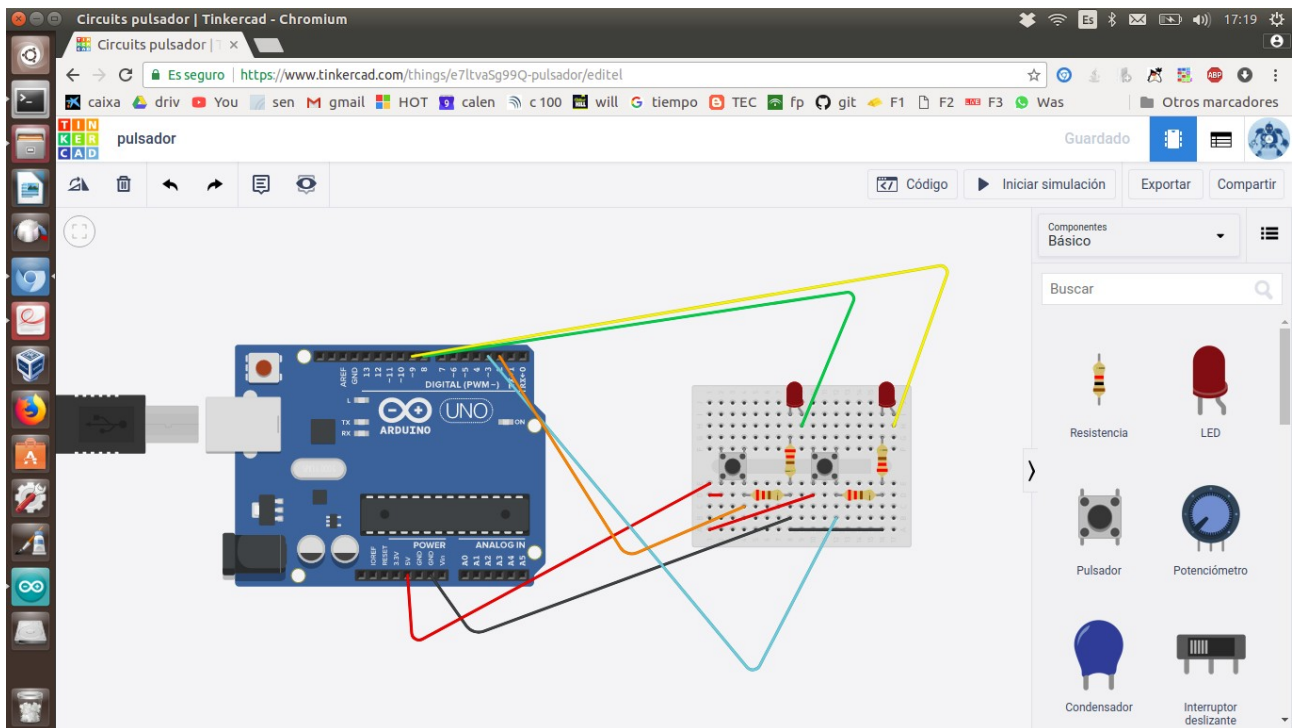
Serial.println("SU ALTURA ES DE ");
Serial.println(200-distancia0); /*colocamos la distancia a dos metros si es menos o más ajustamos el
200*/
Serial.println(" cm");
delay(500);

lcd.setCursor(0, 0); // ubica cursor en columna 0 y línea 1
lcd.print("usted mide");//imprimimos en el lcd la palabra humedo
lcd.setCursor(0, 1); // ubica cursor en columna 0 y línea 0
lcd.print(200-distancia0); // escribe valor de la altura en centímetros
lcd.print(" centímetros "); // escribe el texto

}

```

DOS PULSADORES VERSIÓN ANTIGUA DE IDE ARDUINO (GUADALINEX 2013)



```
const int LED1=8; // pin 1 led1
const int LED2=9; // pin 2 led2
const int BOTON1=2;
const int BOTON2=3;
int val1;
int val2;
void setup() {

    pinMode(BOTON1, INPUT);
    pinMode(BOTON2,INPUT);

    pinMode(LED1, OUTPUT);
    pinMode(LED2, OUTPUT);
}
void loop() {
    val1=digitalRead(BOTON1);
    val2=digitalRead(BOTON2);

    if (val1 == HIGH & val2 == LOW) {

        digitalWrite(LED1, HIGH);
        digitalWrite(LED2, LOW);

    }
    if (val1 == LOW & val2 == HIGH){

        digitalWrite(LED1, LOW); //ahora para cambiar el sentido de giro cambiamos la
polaridad a los motores
        digitalWrite(LED2, HIGH); //corriente y viceversa
```



```
    }  
    else {  
        digitalWrite(LED1, LOW); //ahora para cambiar el sentido de giro cambiamos la  
polaridad a los motores  
        digitalWrite(LED2, LOW); //corriente y viceversa  
    }  
}
```

LCD

Práctica LCD

vídeo youtube que lo explica

descargar la librería con

<https://bitbucket.org/fmalpartida/new-liquidcrystal/downloads/>

o cogerla de la carpeta ldc2 como zip la copio y la pego en la carpeta donde está arduino opt/arduino 1.8.5

entrar en la carpeta del programa arduino en mi caso opt/arduino 1.8.5/library quitar la librería Liquidcrystal y añadir tras extraerla la carpeta de la nueva librería NewLiquidcrystal y luego en Programa...libreria...incluir librería...desde zip

al extraerla se crea una nueva carpeta, los archivos tienen que estar sueltos dentro de la carpeta Newliquidcrystal1.5.1 puede que no los encuentre si no, copiarlos también ahí

CÓDIGO LCD

```
#include <Wire.h> // libreria de comunicacion por I2C
#include <LCD.h> // libreria para funciones de LCD
#include <LiquidCrystal_I2C.h> // libreria para LCD por I2C

LiquidCrystal_I2C lcd (0x27, 2, 1, 0, 4, 5, 6, 7); // DIR, E, RW, RS, D4, D5, D6, D7

void setup()
{
    lcd.setBacklightPin(3,POSITIVE); // puerto P3 de PCF8574 como positivo
    lcd.setBacklight(HIGH); // habilita iluminacion posterior de LCD
    lcd.begin(16, 2); // 16 columnas por 2 lineas para LCD 1602A
    lcd.clear(); // limpia pantalla
}

void loop()
{
    lcd.setCursor(0, 0); // ubica cursor en columna 0 y linea 0
    lcd.print("Edu y Aitor"); // escribe el texto
    lcd.setCursor(0, 1); // ubica cursor en columna 0 y linea 1
    lcd.print(millis() / 1000); // funcion millis() / 1000 para segundos transcurridos
    lcd.print(" seg."); // escribe seg.
}
```

CODIGO HUMEDAD

```
void setup() {
    // put your setup code here, to run once:
```

```

Serial.begin(9600);
pinMode(2, INPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  int val = analogRead(A0);
  Serial.println(val);
  delay(500);
  if(digitalRead(2)==LOW){
    Serial.print("HUMEDO ");
  }
}

```

codigo con sensor de luz,temperatura y humedad

```

float temperatura = 0; //variable para la temperatura con decimales
#include <Wire.h> // libreria de comunicacion por I2C
#include <LCD.h> // libreria para funciones de LCD
#include <LiquidCrystal_I2C.h> // libreria para LCD por I2C

#define PIN_LDR A1 //Cambiamos el nombre de la entrada A1
#define PIN_BUZZER 2 //Cambiamos el nombre del pin 2

LiquidCrystal_I2C lcd (0x27, 2, 1, 0, 4, 5, 6, 7); // DIR, E, RW, RS, D4, D5, D6, D7,configura los
pines del display

void setup()
{
  Serial.begin (9600); //inicia comunicacion serial

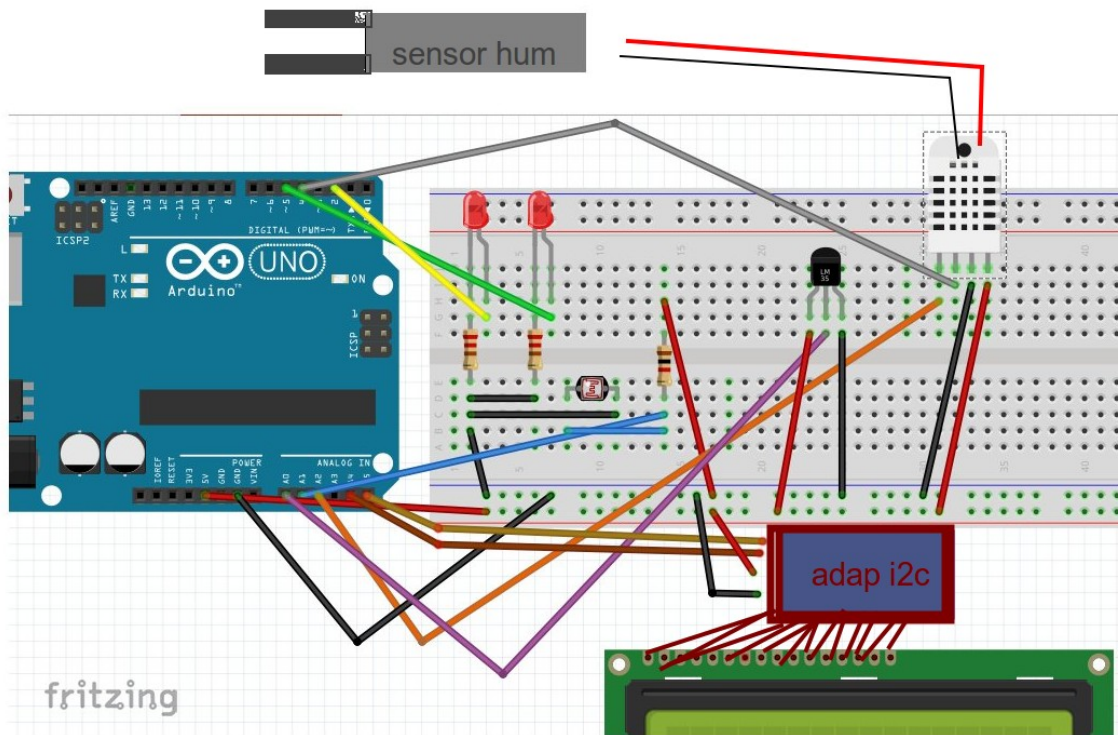
  pinMode(4, INPUT); //definimos pin 4 como entrada
  pinMode(5, OUTPUT); //definimos pin 5 como salida
  lcd.setBacklightPin(3,POSITIVE); // puerto P3 de PCF8574 como positivo
  lcd.setBacklight(HIGH); // habilita iluminacion posterior de LCD
  lcd.begin(16, 2); // 16 columnas por 2 lineas para LCD 1602A
  lcd.clear(); // limpia pantalla
  Serial.begin (9600); //inicia comunicacion serial
  pinMode(PIN_LDR,INPUT); //definimos el pin A0 como entrada
  pinMode(PIN_BUZZER,OUTPUT); //definimos el pin 2 como salida
}
void loop()
{
  //Calcula la temperatura usando como referencia 5v
  temperatura = (5.0 * analogRead(0)*100.0)/1023.0;
  Serial.println (temperatura); //escribe la temperatura en el serial
}

```

```
delay (500); //espera 0.5 segundos para la siguiente medicion
```

```
int humedad = analogRead(A3); //define leyendo la variable de la humedad  
Serial.println(humedad); //imprime en el serial el valor de la humedad  
delay(500); //espera medio segundo hasta la siguiente lectura
```

```
int ldr = analogRead(PIN_LDR); //tomamos el valor de la LDR como entero comando int  
Serial.println (ldr); //escribe la resistencia en el serial  
delay (500); //esperamos medio segundo  
if(digitalRead(4)==LOW){ //si la señal de pin 4 es baja (hay humedad) entonces  
Serial.print("HUMEDO "); //imprimimos en el monitor serie la palabra humedo  
digitalWrite(5,HIGH); // y encendemos un led con el pin 5  
delay(200); //esperamos 0.2 segundos  
digitalWrite(5,LOW); //apagamos el led 5  
delay(200); //esperamos 0.2 , consiguiendo que el led parpadee  
lcd.setCursor(0, 1); // ubica cursor en columna 0 y linea 1  
lcd.print("humedo"); //imprimimos en el lcd la palabra humedo  
}  
else {  
lcd.setCursor(0, 1); // ubica cursor en columna 0 y linea 1  
lcd.print("seco "); //imprimimos en el lcd la palabra seco  
}  
if (ldr > 300) //si el valor de la ldr aumenta cuando disminuye la luz encender el led  
{  
digitalWrite(PIN_BUZZER,HIGH); //encendemos led  
}  
else //en el caso contrario , es decir siendo menor lo apagamos  
{  
digitalWrite(PIN_BUZZER,LOW); //apagamos el led  
}  
lcd.setCursor(0, 0); // ubica cursor en columna 0 y linea 0  
lcd.print(ldr); // escribe valor de lumninosidad  
lcd.print(" lum "); // escribe el texto  
  
lcd.print(temperatura); // escribe valor temperatura  
lcd.print(" C"); // escribe el texto  
  
}
```



LCD SENCILLA (TIPO HOLA MUNDO)

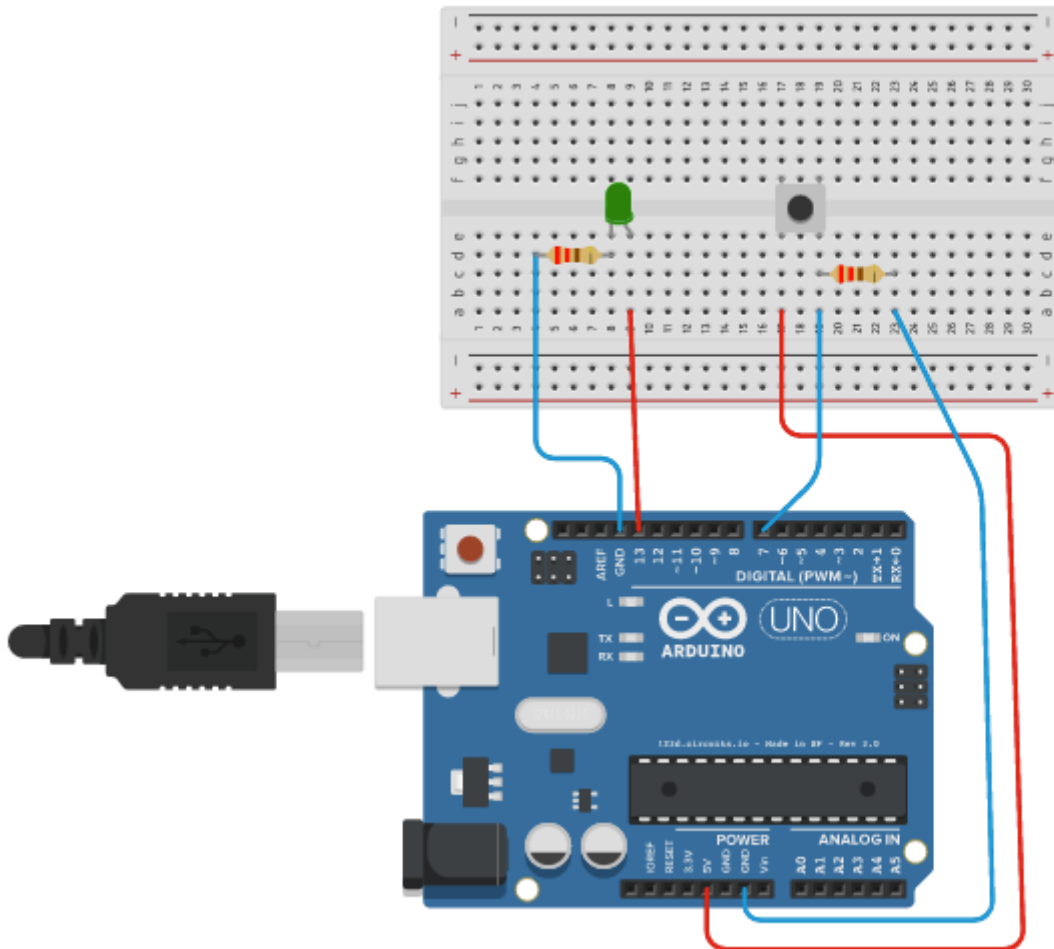
```
#include <LiquidCrystal_I2C.h>
```

```
LiquidCrystal_I2C lcd (0x20,16,2); // si no te sale con esta direccion puedes usar (0x3f,16,2) ||  
(0x27,16,2) ||(0x20,16,2)
```

```
void setup() {  
  lcd.init();  
  lcd.backlight();  
  lcd.clear();  
  lcd.setCursor(0,0);  
  lcd.print("EDU");  
  lcd.setCursor (0,1);  
  lcd.print("AITOR");  
}
```

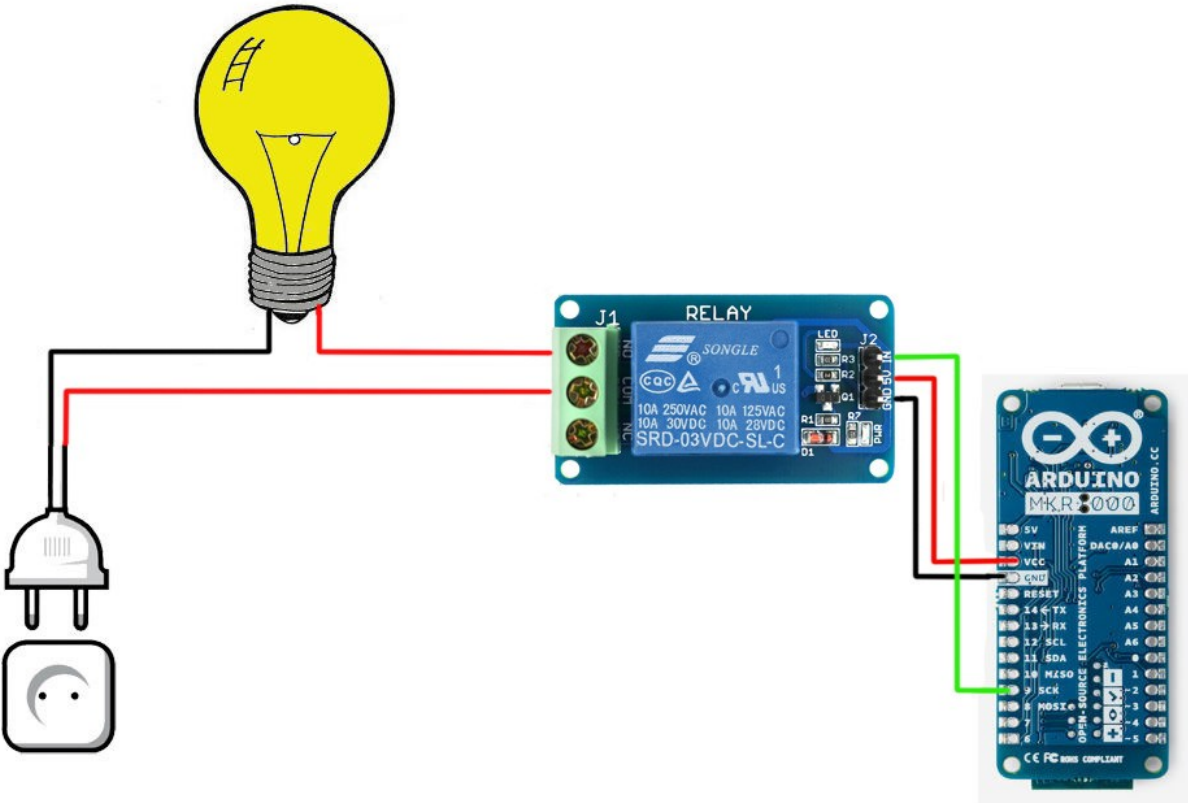
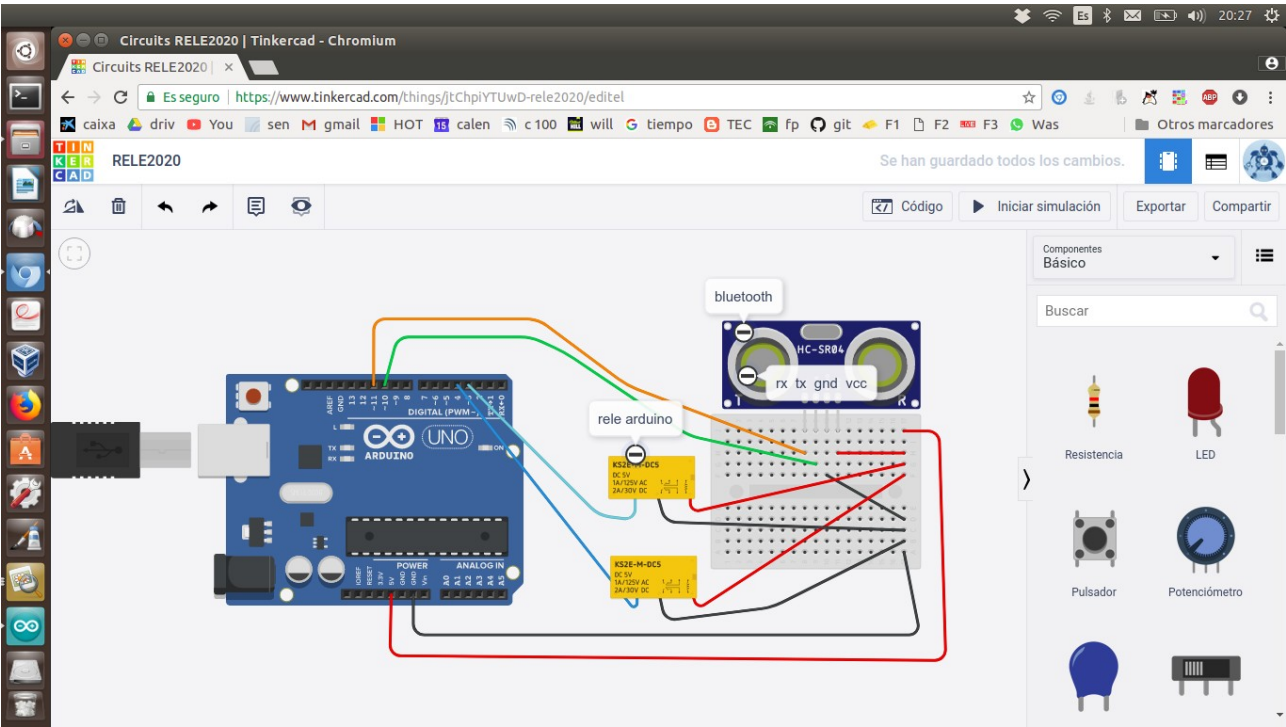
```
void loop() {  
  
  lcd.display();  
  delay(500);  
  lcd.noDisplay();  
  delay(500);  
}
```

UN PULSADOR IDE ANTIGUO (GUADALINEX 2013)



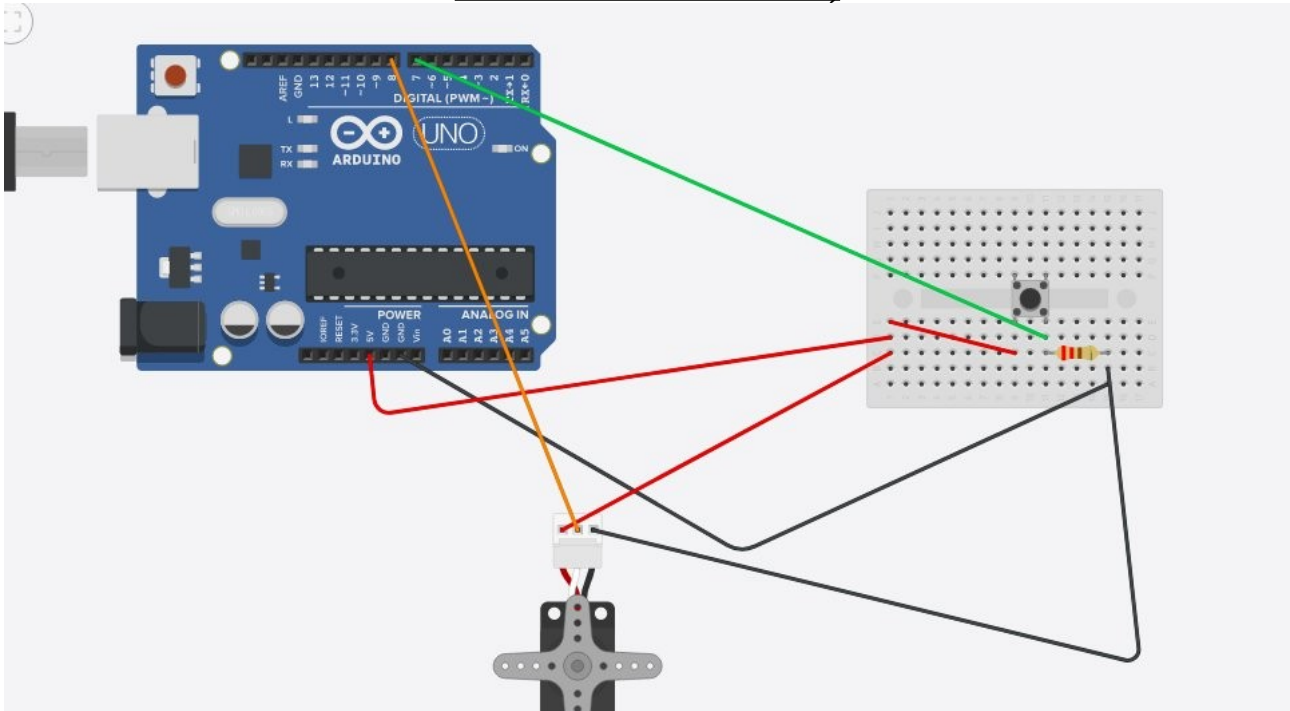
```
// Práctica encender y apagar un LED a través de botón pulsador
const int LED=13;
const int BOTON=7;
int val;
void setup(){
  pinMode(LED,OUTPUT);
  pinMode(BOTON,INPUT);
}
void loop(){
  val=digitalRead(BOTON);
  if (val==HIGH){
    digitalWrite(LED,HIGH);
  }
  else { digitalWrite(LED,LOW);
  }
}
```

RELE CONTROLADO POR MOVIL



```
#include <SoftwareSerial.h>
SoftwareSerial miPuertoSerie=SoftwareSerial(10,11);
int unChar;
void setup(){
miPuertoSerie.begin(9600);
pinMode(3, OUTPUT);
pinMode(4, OUTPUT);
pinMode(5, OUTPUT);
pinMode(6, OUTPUT);
}
void loop(){
char unChar=miPuertoSerie.read();
if (unChar>0){
switch (unChar){
case '2':
digitalWrite(3, HIGH);
digitalWrite(5,HIGH);
delay(100);
break;
case '1':
digitalWrite (4,HIGH);
digitalWrite(6,HIGH);
delay(100);
break;
case '3':
digitalWrite(3, LOW);
digitalWrite(5,LOW);
delay(100);
break;
case '0':
digitalWrite (4,LOW);
digitalWrite(6,LOW);
delay(100);
break;
}
}
}
```


SERVOS GRANDES (NEGROS, NO LOS PEQUEÑOS AZULES ,LOS ÁNGULOS DE PARO SON DIFERENTES)



```
// Práctica encender y apagar un LED a través de botón pulsador
#include <Servo.h> //incluye la librería servo
Servo servoY; //llamanos al servo 2 servo Y
const int LED=13;
const int BOTON=7;
int val;
void setup(){

pinMode(BOTON,INPUT);
servoY.attach(8); // le damos señal al servo Y con el pin 8

servoY.write(0);
}
void loop(){
val=digitalRead(BOTON);
if (val==HIGH){
servoY.write(150); //le decimos que el servo X vaya a la posición de 150 grados
delay(490);
}
else {
servoY.write(90); // al darle la posición 90 el servo se para
delay(500);
}
}
```