

Nombres y Apellidos: \_\_\_\_\_

Grado y Sección: \_\_\_\_\_

# Práctica calificada 10.1

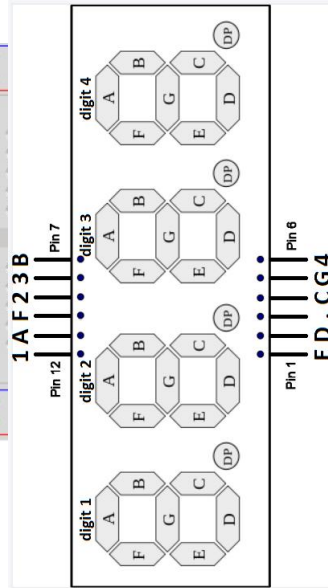
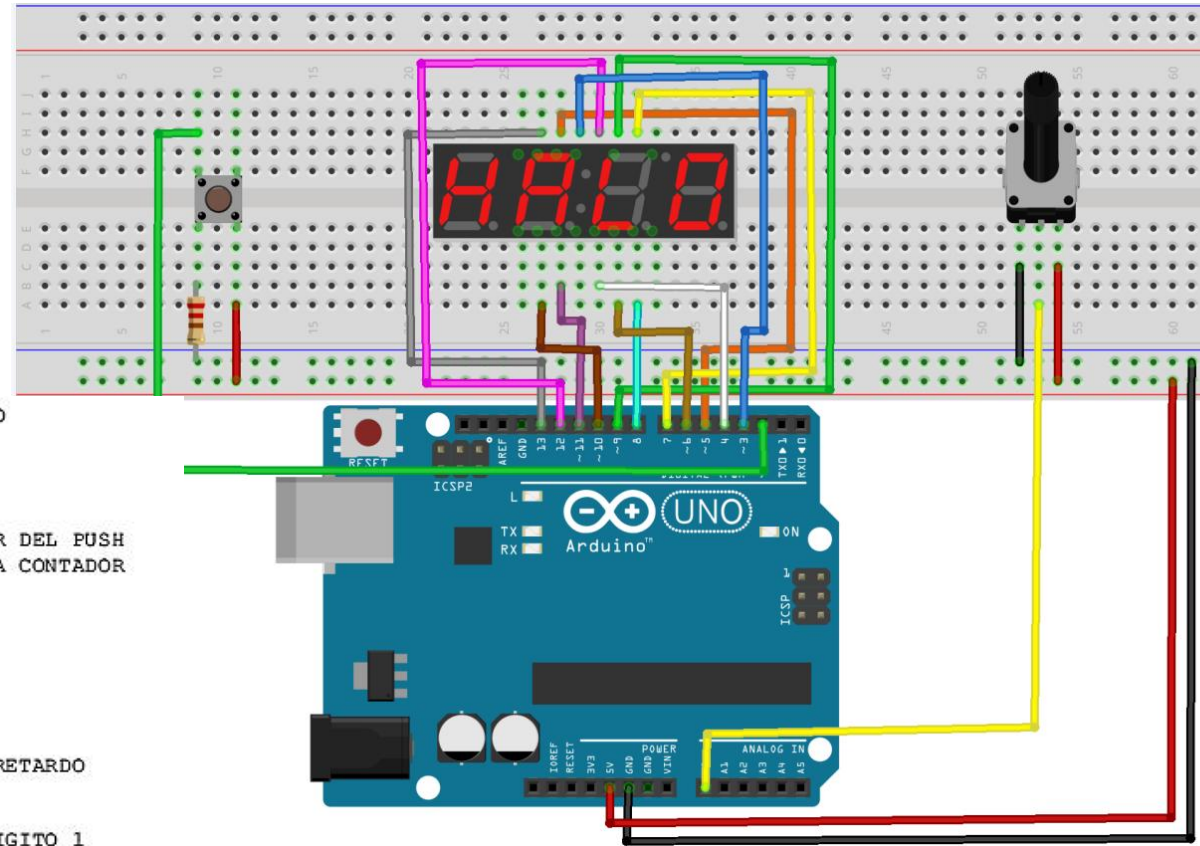
## CÓDIGO FUENTE

```
long n = 0;
int x = 100;
int del = 1000;
int a=5;
int b=7;
int c=4;
int d=11;
int e=10;
int f=3;
int g=6;
//SE CONFIGURA PINES COMO SALID.
void setup(){

pinMode(13, OUTPUT);
pinMode(12, OUTPUT);
pinMode(11, OUTPUT);
pinMode(10, OUTPUT);
pinMode(9, OUTPUT);
pinMode(8, OUTPUT);
pinMode(7, OUTPUT);
pinMode(6, OUTPUT);
pinMode(5, OUTPUT);
pinMode(4, OUTPUT);
pinMode(3, OUTPUT);
pinMode(2, INPUT); //PUSH BUTTON

pinMode(A0, INPUT); //POTENCIOMETRO
}

void loop(){
  int v=digitalRead(2); //LEER VALOR DEL PUSH
  //SI SE OPRIME EL PUSH SE REINICIA CONTADOR
  if(v==1){
    limpiar();
    n=0;
    delayMicroseconds(500);
  }
  //LEER VALOR DEL POTENCIOMETRO
  int t=analogRead(A0);
  //DE ACUERDO AL VALOR DE t ES EL RETARDO
  limpiar(); //LIMPIAR DISPLAY'S
  digito(1); //HABILITAR DIGITO 1
  numero((n/x/1000)%10); //OBTENER DIGITO 1
  delayMicroseconds(t); // RETARDO
  limpiar();
  digito(2); //HABILITAR DIGITO 2
  numero((n/x/100)%10); //OBTENER DIGITO 2
  delayMicroseconds(t);
  limpiar();
  digito(3); //HABILITAR DIGITO 3
  numero((n/x/10)%10); //OBTENER DIGITO 2
  delayMicroseconds(t);
  limpiar();
  digito(4); //HABILITAR DIGITO 4
  numero(n/x % 10); //OBTENER DIGITO 4
  delayMicroseconds(t);
  if(n/x > 9999){ //SI LLEGA A 9999 SE REINICIA
    n = 0;
    delay(5000);
  }
  else n++;
}
```



```
// FUNCION PARA HABILITAR DIGITO A IMPRIMIR
void digito(int x){
  digitalWrite(13, HIGH); // dig 1
  digitalWrite(12, HIGH); // dig 2
  digitalWrite( 9, HIGH); // dig 3
  digitalWrite( 8, HIGH); // dig 4
  //DE ACUERDO AL VALOR SE HABILITA DIGITO
  switch(x){
    case 1: digitalWrite(13, LOW); break;
    case 2: digitalWrite(12, LOW); break;
    case 3: digitalWrite(9, LOW); break;
    case 4: digitalWrite(8, LOW); break;
  }
}

//DE ACUERDO AL VALOR SE IMPRIME DIGITO EN DISPLAY
void numero(int x){
  switch(x){
    case 1: uno(); break;
    case 2: dos(); break;
    case 3: tres(); break;
    case 4: cuatro(); break;
    case 5: cinco(); break;
    case 6: seis(); break;
    case 7: siete(); break;
    case 8: ocho(); break;
    case 9: nueve(); break;
    default: cero(); break;
  }
}
```

... continuar el código

```

//FUNCION LIMPIAR DIGITO, TODOS
APAGADOS
void limpiar(){
  digitalWrite( 5, LOW); // A
  digitalWrite( 7, LOW); // B
  digitalWrite( 4, LOW); // C
  digitalWrite(11, LOW); // D
  digitalWrite(10, LOW); // E
  digitalWrite( 3, LOW); // F
  digitalWrite( 6, LOW); // G
}

//FUNCIONES DE NUMEROS PARA DISPLAY
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);
  digitalWrite(d,LOW);
  digitalWrite(e,LOW);
  digitalWrite(f,LOW);
  digitalWrite(g,LOW);
}

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

```

```

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

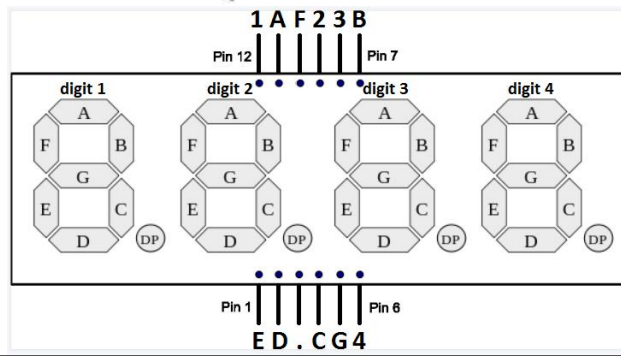
void aa(){
  digitalWrite(a,HIGH);
  digitalWrite(b,HIGH);
  digitalWrite(c,HIGH);
  digitalWrite(d,LOW);
  digitalWrite(e,HIGH);
  digitalWrite(f,HIGH);
  digitalWrite(g,HIGH);
}

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

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

```

... continuar el  
código



## DIAGRAMA DE FLUJO