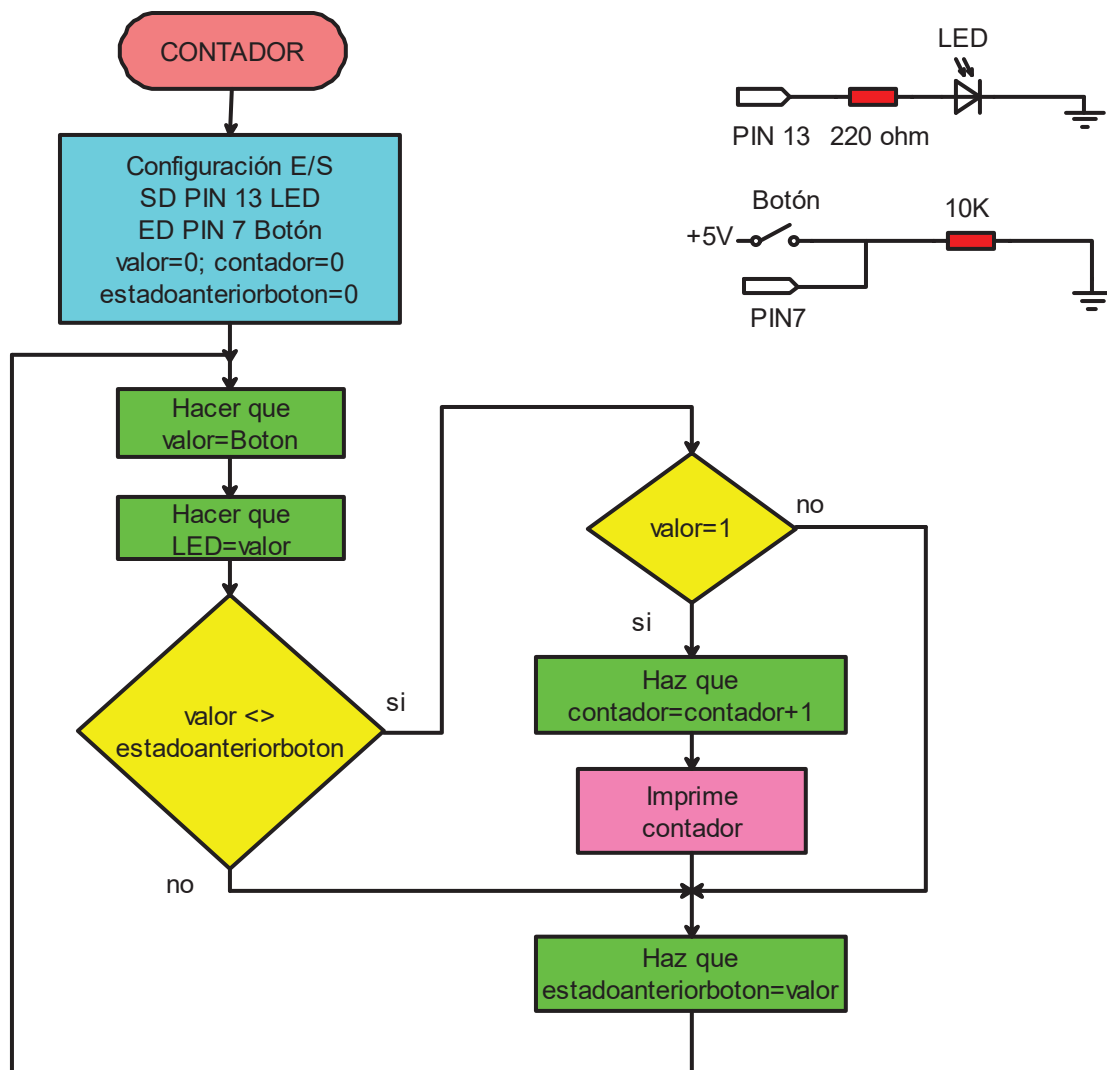
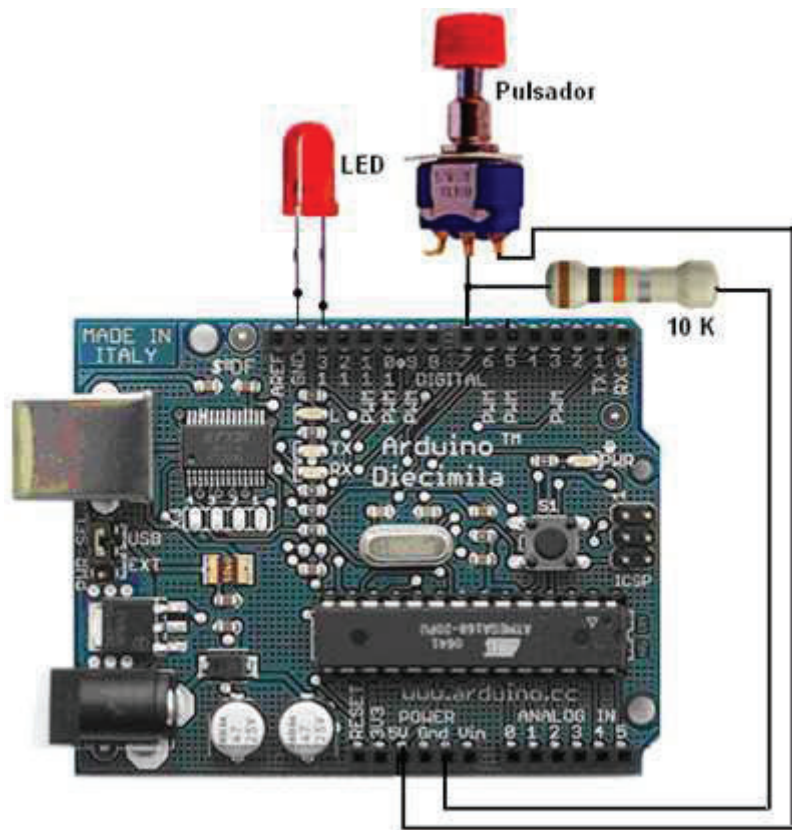


9. Contador

Se trata de contar las veces que se pulsa un botón conectado en la entrada 7 de Arduino a la vez que cada vez que contamos encendemos el led conectado en la salida 13. El valor de la variable que almacena el número de impulsos generados se envía al PC para que se visualice en la pantalla.





Esquema

```

/* Programa Contador
 * -----
 *
 * Detecta si el botón conectado a la entrada 7 ha sido presionado y enciende el LED
 * Envía al PC el valor de la variable de cuenta ""Contador" vía puerto serie.
 *
 * Christian Nold & Erica Calogero
 *
 */

int LED = 13;
int Boton = 7;
int valor = 0;
int contador = 0;
int estadoanteriorboton = 0;

void setup()
{
  beginSerial(9600);           // Configura velocidad de transmisión a 9600

```

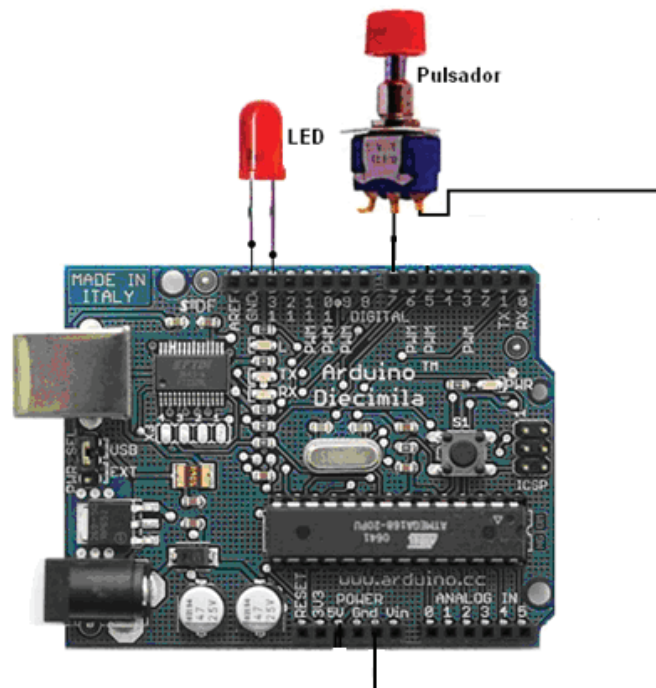
```

pinMode(LED, OUTPUT); // inicializa como salida digital el pin 13
pinMode(Boton, INPUT); // inicializa como entrada digital el 7
}

void loop()
{
  valor = digitalRead(Boton); // lee el valor de la entrada digital pin 7
  digitalWrite(LED, valor);
  if(valor != estadoanteriorboton){
    if(valor == 1){
      contador++;
      printInteger(contador);
      serialWrite(10);
      serialWrite(13);
    }
  }
  estadoanteriorboton = valor;
}

```

Podríamos prescindir de la resistencia colocada con el pulsador si habilitásemos la resistencia interna Pull-up de la entrada PIN7 en ese caso el circuito quedaría como el siguiente:



El programa en este caso seria muy parecido al anterior. Observese que ahora al pulsar el boton introducimos un “=” en el PIN7 por lo tanto si quiero que se encienda la salida PIN13 debo escribir en ella el valor leído del pulsador negado es decir “!valor”

```
/* Programa Contador Modificado
 * -----
 *
 * Detecta si el botón conectado a la entrada 7 ha sido presionado y enciende el LED
 * Envía al PC el valor de la variable de cuenta ""Contador" vía puerto serie.
 *
 * Christian Nold & Erica Calogero J.M. Ruiz
 *
 */

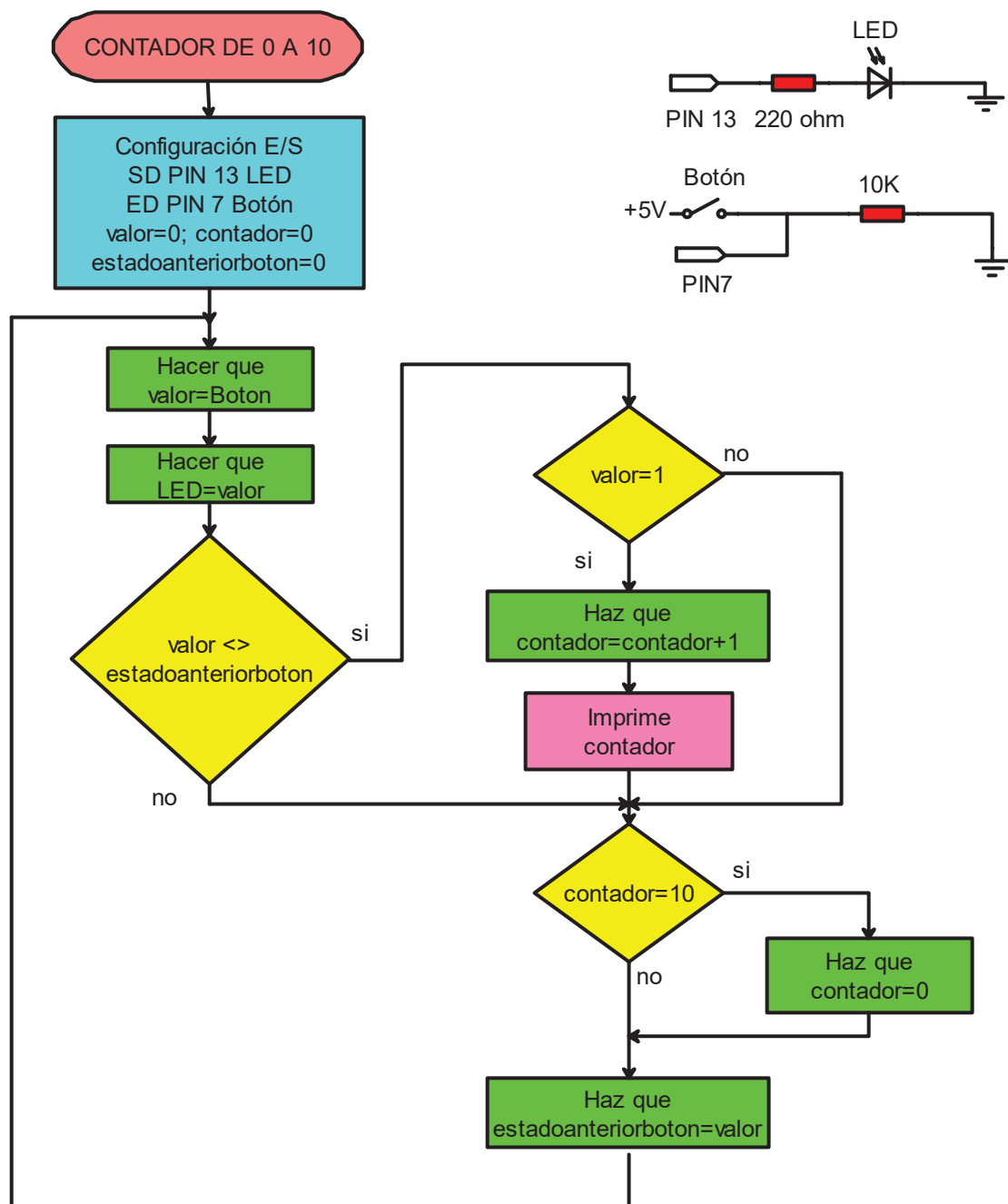
int LED = 13;
int Boton = 7;
int valor = 0;
int contador = 0;
int estadoanteriorboton = 0;

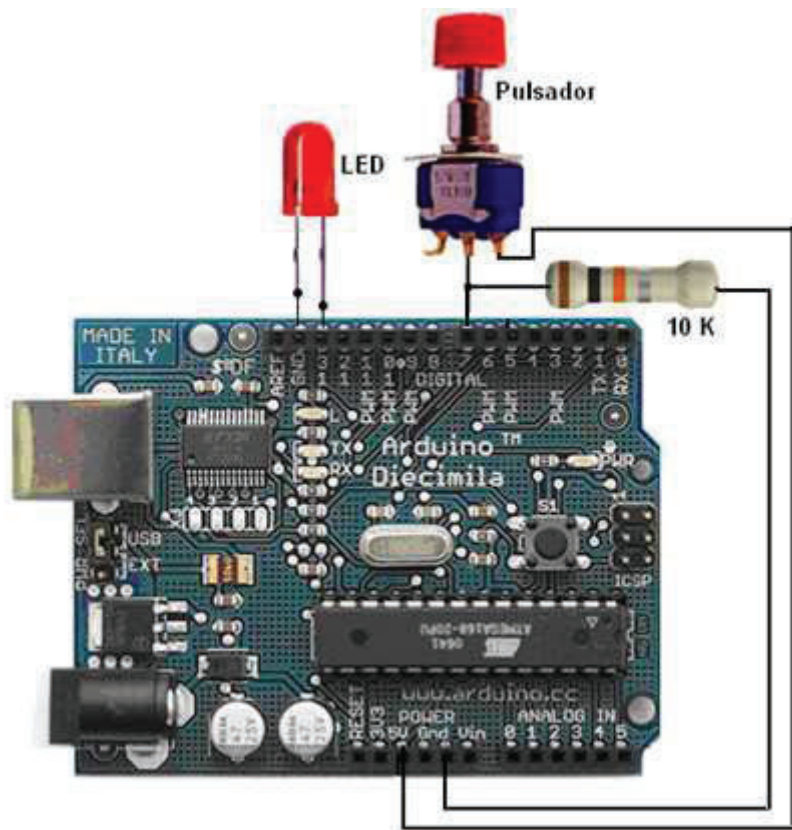
void setup()
{
  beginSerial(9600);          // Configura velocidad de transmisión a 9600
  pinMode(LED, OUTPUT);      // inicializa como salida digital el pin 13
  pinMode(Boton, INPUT);     // inicializa como entrada digital el 7
  digitalWrite(Boton,HIGH);  // Habilitamos la resistencia interna Pull-up del PIN7
}

void loop()
{
  valor = digitalRead(Boton); // lee el valor de la entrada digital pin 7
  digitalWrite(LED, !valor);  // Escribimos en la salida el bvlao leído negado
  if(valor != estadoanteriorboton){
    if(valor == 1){
      contador++;
      printInteger(contador);
      serialWrite(10);
      serialWrite(13);
    }
  }
  estadoanteriorboton = valor;
}
```

Contador de 0 a 10

Se trata de realizar una variación sobre el ejemplo anterior de tal manera que cuando el valor del contador llegue a 10 se ponga a cero y comience de nuevo.





Esquema

```

/* Programa Contador de 0 a 10
 * -----
 *
 * Detecta si el botón conectado a la entrada 7 ha sido presionado y enciende el LED
 * Envía al PC el valor de la variable de cuenta ""Contador" vía puerto serie.
 *
 * Christian Nold & Erica Calogero
 *
 */

int LED = 13;
int Boton = 7;
int valor = 0;
int contador = 0;
int estadoanteriorboton = 0;

void setup()
{
  beginSerial(9600);           // Configura velocidad de transmisión a 9600

```

```
pinMode(LED, OUTPUT); // inicializa como salida digital el pin 13
pinMode(Boton, INPUT); // inicializa como entrada digital el 7
}

void loop()
{
  valor = digitalRead(Boton); // lee el valor de la entrada digital pin 7
  digitalWrite(LED, valor);

  if(valor != estadoanteriorboton){
    if(valor == 1){

      contador++;
      printInteger(contador);
      serialWrite(10);
      serialWrite(13);
      if (contador==10) {          // Limita la cuenta al valor 10
        contador=0;
      }
    }
  }
  estadoanteriorboton = valor;
}
```