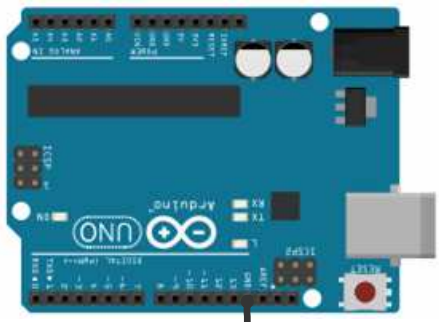
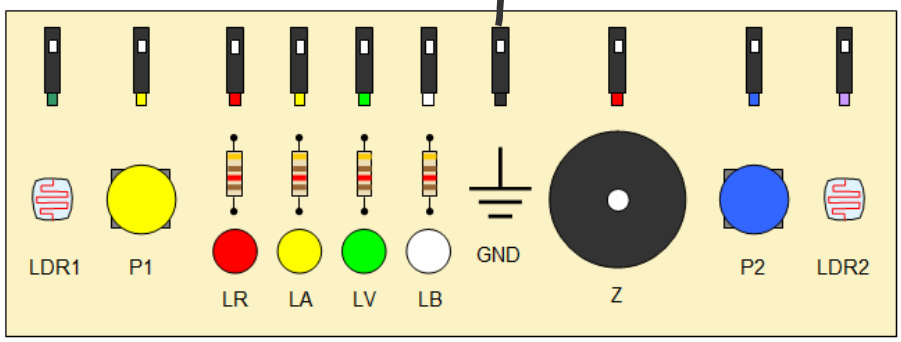
ACTIVIDADES ARDUINO



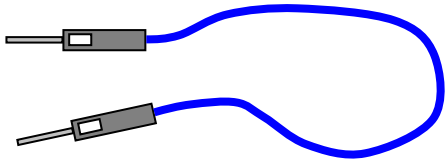
Para dar nuestros primeros pasos con Arduino, y para facilitar y agilizar las conexiones, hemos preparado un *"módulo de entrenamiento"* que contiene varios elementos actuadores (4 LEDs de diversos colores con sus resistencias correspondientes y un zumbador) y varios elementos sensores (2 pulsadores y 2 LDR). En cada ejercicio solo tendrás que utilizar algunos de los componentes.



Todos estos componentes se caracterizan porque uno de sus dos terminales va conectado a GND.

Lo que se ha hecho al construir el módulo es conectar el terminal correspondiente de cada elemento al terminal GND que aparece en el módulo.

De esta forma, para trabajar basta con conectar con un conector macho-macho únicamente dicho terminal GND del módulo con un pin GND de Arduino. Ya sólo tenemos que preocuparnos de conectar con conectores macho-macho el otro terminal de cada elemento al pin de Arduino que decidamos.

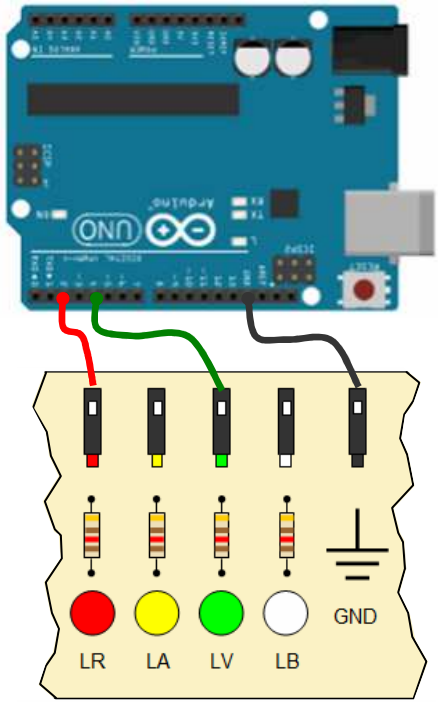


Conector macho-macho

BLOQUE A: LA FUNCIÓN setup(). EJECUTANDO LAS INSTRUCCIONES SÓLO UNA VEZ.

A.0.- Ejemplo resuelto. Elaborar un programa para que cada vez que se presione el botón de Reset de la placa se encienda el LED verde durante 5 segundos, a continuación de apague y se encienda el LED rojo durante otros 5 segundos y se apague.

```
void setup() {
   pinMode(2,OUTPUT);
   pinMode(4,OUTPUT);
   digitalWrite(2,HIGH);
   delay(5000);
   digitalWrite(2,LOW);
   digitalWrite(4,HIGH);
   delay(5000);
   digitalWrite(4,LOW);
}
void loop() {
}
```



A.1. Realiza los siguientes cambios al ejemplo anterior:

A.1.1.- Que los tiempos de encendido pasen a ser 2 segundos, que los pines de los LEDs sean el 3 y el 7.

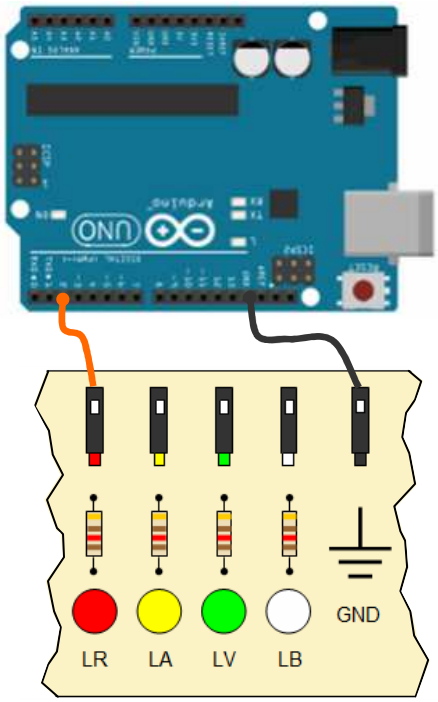
A.1.2.- Que tras apagarse el LED rojo se encienda el LED amarillo y se quede encendido. Se apagará al dar al botón de Reset para empezar de nuevo.

BLOQUE B: LA FUNCIÓN loop(). REPITIENDO LAS INSTRUCCIONES INDEFINIDAMENTE.

B.0.- Ejemplo Resuelto. Programa para que un LED realice de forma indefinida una secuencia de 1 segundo encendido y 0,5 segundos apagado.

```
void setup() {
  pinMode(2,OUTPUT);
}

void loop() {
  digitalWrite(2,HIGH);
  delay(1000);
  digitalWrite(2,LOW);
  delay(500);
}
```



B.1. Realiza el siguiente cambio al ejemplo anterior:

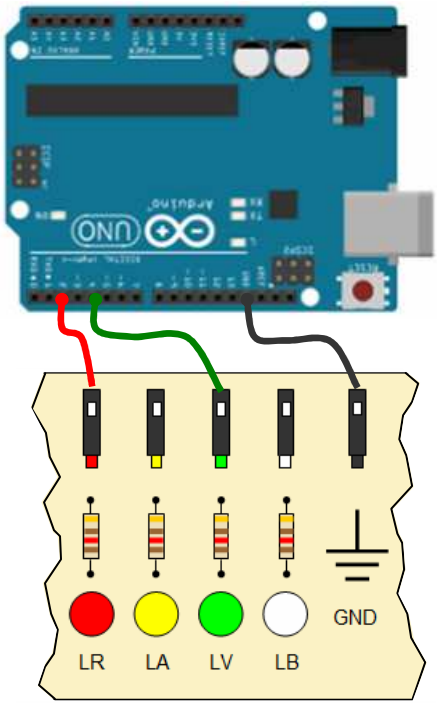
Que la secuencia sea 4 segundos encendidos y 2 apagado, a continuación 2 segundos encendido y 2 segundos apagado, después 1 segundo encendido y 2 segundos apagado, después 0,25 segundos encendido y 2 apagados y que este ciclo se vuelva a repetir indefinidamente.

B.2.- Realiza los siguientes programas:

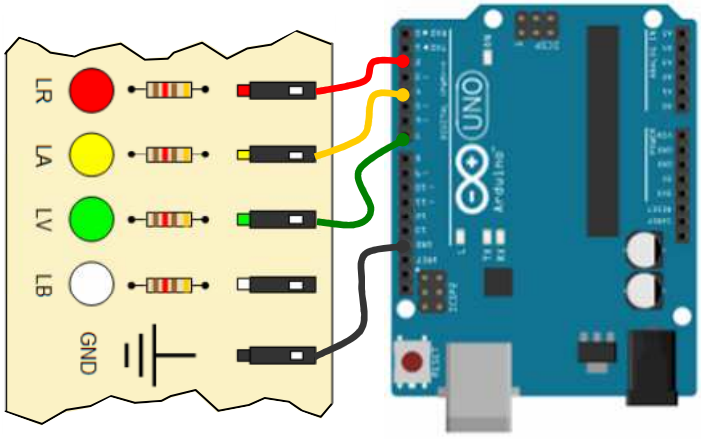
B.2.1.- Que se enciendan el LED rojo y el LED verde al mismo tiempo con la secuencia 2 segundos encendidos y 1 apagados, y así sucesivamente.

B.2.2.- Programa para que dos LED se enciendan de forma alterna (cuando el rojo está encendido el verde está apagado y viceversa) con una periodicidad de 1 segundo.

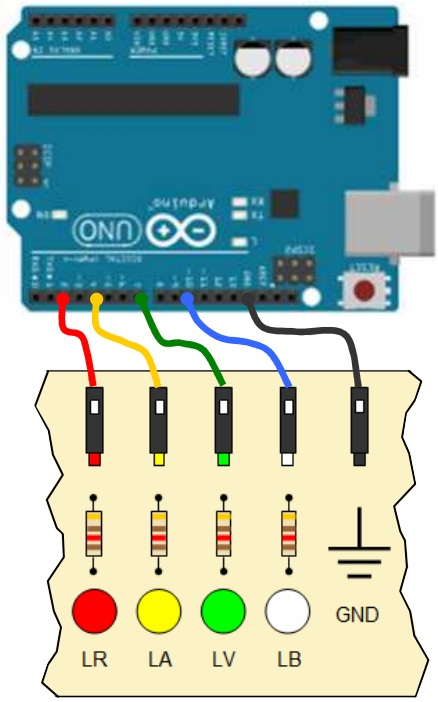
B.2.3.- Que se encienda el rojo solo 2 segundos, a continuación (sin que se apague el rojo) se encenderá el verde 2 segundos (estarán encendidos ambos), después se apagará el rojo y se quedará encendido el verde durante otros 2 segundos, a continuación se encenderá el rojo (sin que se apague el verde) durante otros 2 segundos, después se apagará el verde, con lo que habremos vuelto a la situación inicial. Que este ciclo se repita indefinidamente.



B.3.- Realiza un programa para que tres LED, rojo, amarillo y verde, simulen el funcionamiento de un semáforo, de forma que el verde esté encendido 4 segundos, el amarillo 1,5 segundos y el rojo 5 segundos, en dicho orden, y así sucesivamente.

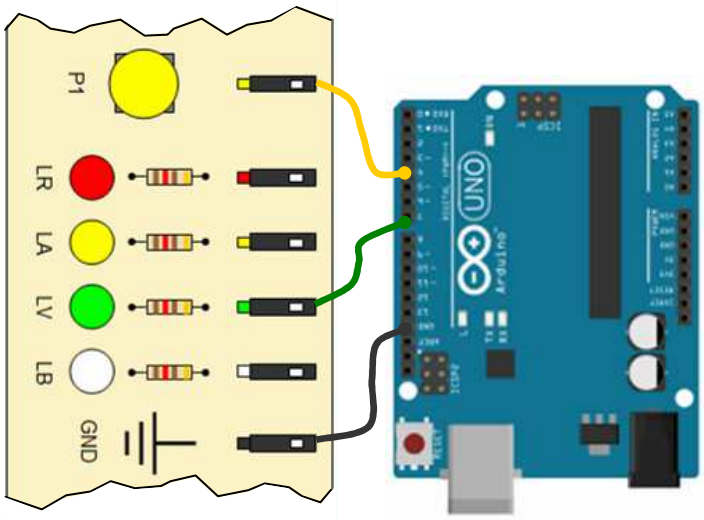


B.4.- Realiza un programa para que los cuatro LEDs del entrenador se vayan encendiendo de forma sucesiva, de izquierda a derecha y a continuación de derecha a izquierda (similar al "coche fantástico"), con una cadencia de 0,25 segundos aproximadamente cada LED. En cada momento sólo estará encendido un LED.



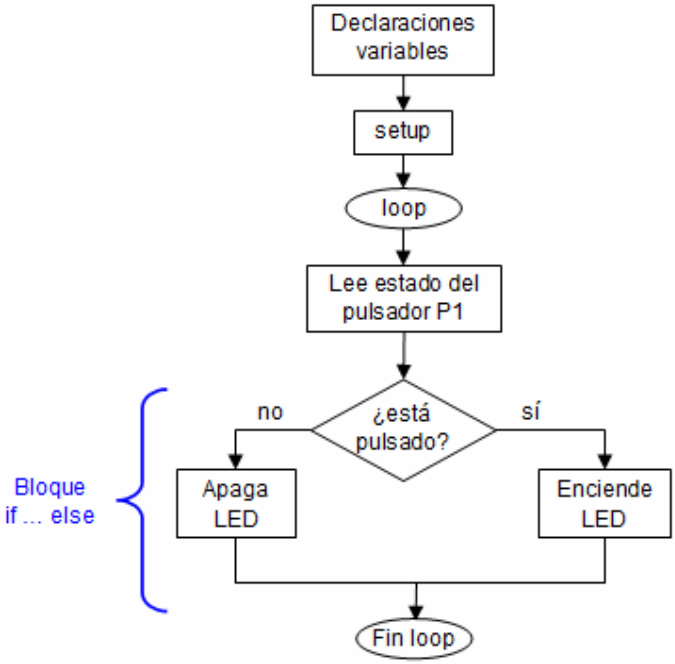
B.5.- Realiza un programa para que el encendido de los LEDs siga la siguiente secuencia, con un intervalo de tiempo fijo de 0,5 segundos: empiezan los cuatro apagados, se encenderá el rojo, tras 0,5 segundos el amarillo (el rojo sigue encendido), tras 0,5 segundos el verde (rojo y amarillo siguen encendidos), tras 0,5 segundos el blanco (los tres están encendidos), tras 0,5 segundos se apagará el rojo, tras otros 0,5 segundos se apagará el amarillo, después el verde y por último el blanco, con lo que volvemos a la situación de inicio. El ciclo se repetirá indefinidamente.

BLOQUE C: DECISIONES if … else



C.0.- Ejemplo resuelto que hace que un LED verde (LV) conectado en el pin 7 se encienda cuando un pulsador (P1) conectado al pin 4 está pulsado y el LED se apague cuando dicho interruptor está sin pulsar. Edítalo, cárgalo en la tarjeta y comprueba que funciona.

Observa el diagrama de flujo del programa e identifica cada bloque del diagrama con el código en el programa.




```
#define LV 7
#define P1 4
int estadopulsador;

void setup() {
  pinMode(LV, OUTPUT);
  pinMode(P1,  INPUT_PULLUP);
}

void loop() {
  estadopulsador=digitalRead(P1);
  if(estadopulsador==LOW){
    digitalWrite(LV, HIGH);
  }
  else{
    digitalWrite(LV, LOW);
  }
}
```

C.1.- Con el mismo montaje anterior realiza los siguientes programas:

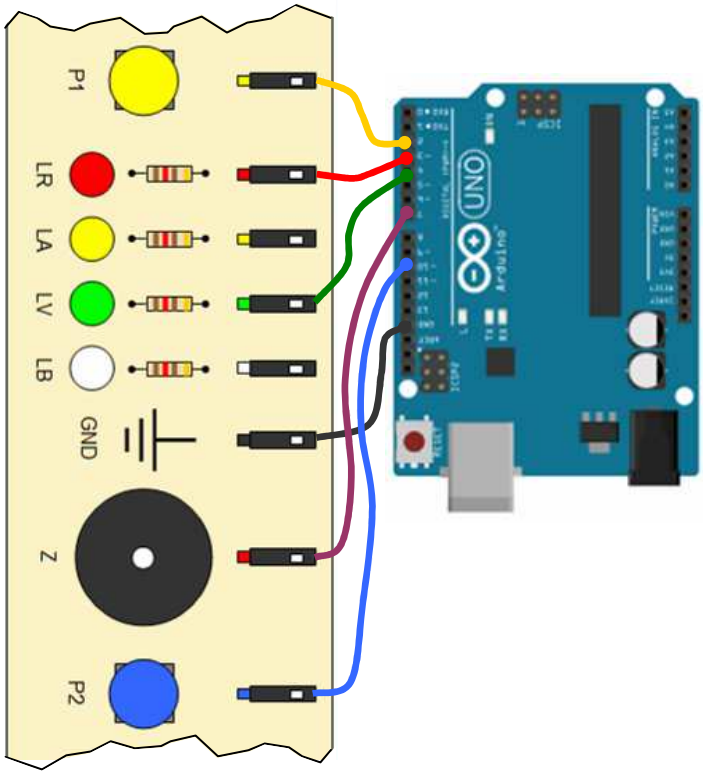
C.1.1.- Programa para que el LED esté parpadeando de forma intermitente a intervalos de 0,5 segundos siempre que no esté pulsado el pulsador P1. Cuando pulsemos el pulsador P1 el LED se apagará. Cuando volvamos a dejar de pulsar P1 el LED se pondrá a parpadear de nuevo. Realiza el diagrama de flujo.

C.1.2.- Programa similar al anterior, es decir, el LED parpadeará de forma intermitente a intervalos de 0,5 segundos cuando el pulsador P1 no esté pulsado. En el momento de pulsar P1, si el LED está encendido, se quedará encendido mientras que el pulsador permanezca pulsado y si en el momento de pulsar está apagado, permanecerá apagado mientras el pulsador permanezca pulsado. Al dejar de pulsar P1, el LED volverá a parpadear. Realiza el diagrama de flujo.

C.1.3.- Programa para que el LED se encienda al pulsar el pulsador P1 y se apague cuando se vuelva a pulsar el mismo pulsador P1. Si se mantiene pulsado el pulsador de forma permanente, el LED se quedará estable en el último estado adquirido. Realiza el diagrama de flujo. Nota: si este ejercicio no te sale usando la función if….else, mira más adelante cómo el bucle while() en el siguiente bloque de ejercicios.

C.2.- Programa para que el LED verde permanezca encendido y el LED rojo permanezca apagado mientras se pulsa el pulsador P1 y cambien su estado cuando se deje de pulsar. Igualmente, el zumbador pitará mientras se pulse el pulsador P2 y callará cuando no esté pulsado.

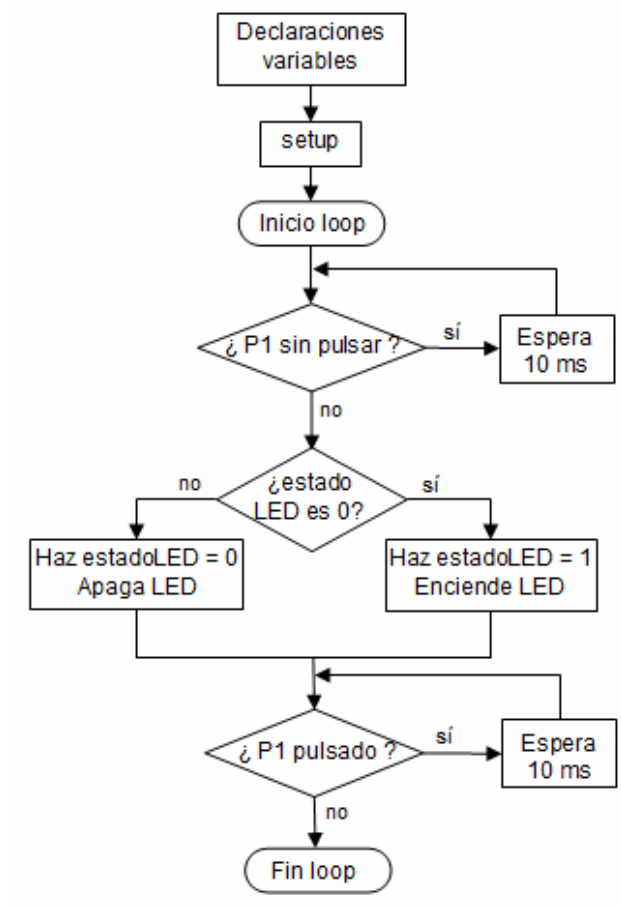
C.3.- Programa para que el LED rojo se encienda cuando se pulse el pulsador P1 y se apague cuando se pulse el pulsador P2. Si los dos pulsadores se pulsan simultáneamente el LED estará apagado.



C.4.- Programa para un juego de rapidez en responder a una pregunta. Se supone que juegan dos jugadores, uno maneja el pulsador P1 y el LED rojo y otro el pulsador P2 y el LED blanco. Tras realizar una pregunta, el jugador que cree saberla pulsa su pulsador, lo que encenderá su LED correspondiente (aunque ya deje de pulsar se quedará encendido). Una vez que un jugador haya pulsado y encendido su LED, el otro jugador no podrá encender su LED aunque pulse su pulsador. Para reiniciar el sistema para una nueva pregunta, se pulsará sobre el botón Reset (ya que no tenemos más pulsadores en el módulo entrenador), con lo que el LED que esté encendido se apagará y se empieza de nuevo.

BLOQUE D: BUCLE CONDICIONAL while().

D.0.- Ejemplo resuelto para ilustrar el uso de la función while(), volveremos a resolver el problema del ejercicio **C.1.3**, que era un programa para que el LED se encienda al pulsar el pulsador P1 y se apague cuando se vuelva a pulsar el mismo pulsador P1. Si se mantiene pulsado el pulsador de forma permanente, el LED se quedará estable en el último estado adquirido.




```
int LR=3;
int P1=2;
int estadoLED;

void setup() {
  pinMode(LR, OUTPUT);
  pinMode(P1,INPUT_PULLUP);
  estadoLED=0;
  digitalWrite(LR, LOW);
}

void loop() {
  while(digitalRead(P1)==HIGH){
    delay(10);
  }
  if(estadoLED==0){
    digitalWrite(LR,HIGH);
    estadoLED=1;
  }
  else{
    digitalWrite(LR,LOW);
    estadoLED=0;
  }
  while(digitalRead(P1)==LOW){
    delay(10);
  }
}
```

D.1.- Programa que realice el ciclo siguiente: 4 segundos encendido LED rojo, a continuación se apaga y enciende 4 segundos el LED verde, a continuación lo apaga y espera a que se pulse el pulsador P1 antes de empezar de nuevo.

D.2.- Programa para que cada vez que se pulsa el pulsador P1 el LED rojo cambie de estado (si está apagado que se encienda y se quede encendido y si está encendido que se apague y se quede apagado) y que cada vez que se pulsa el pulsador P2 el zumbador cambie de estado.

D.3.- Programa para que un LED se encienda cuando se pulse 3 veces un pulsador, y se apague cuando se pulse 2 veces el mismo pulsador, y así sucesivamente.

D.4.- Programa que controle los 4 LEDs con los dos pulsadores (P1 y P2). Empiezan los cuatro LEDs apagados. Por cada pulsación de P1 se enciende un nuevo LED (empezando por el blanco y terminando con el rojo) y por cada pulsación de P2 se apaga un LED (en dirección desde el rojo hacia el blanco). Como máximo puede haber los 4 LEDs encendidos y como mínimo todos apagados. Si se pulsa P1 cuando ya están los 4 LEDs encendidos o si se pulsa P2 cuando ya están todos los LEDs apagados, sonará el zumbador.

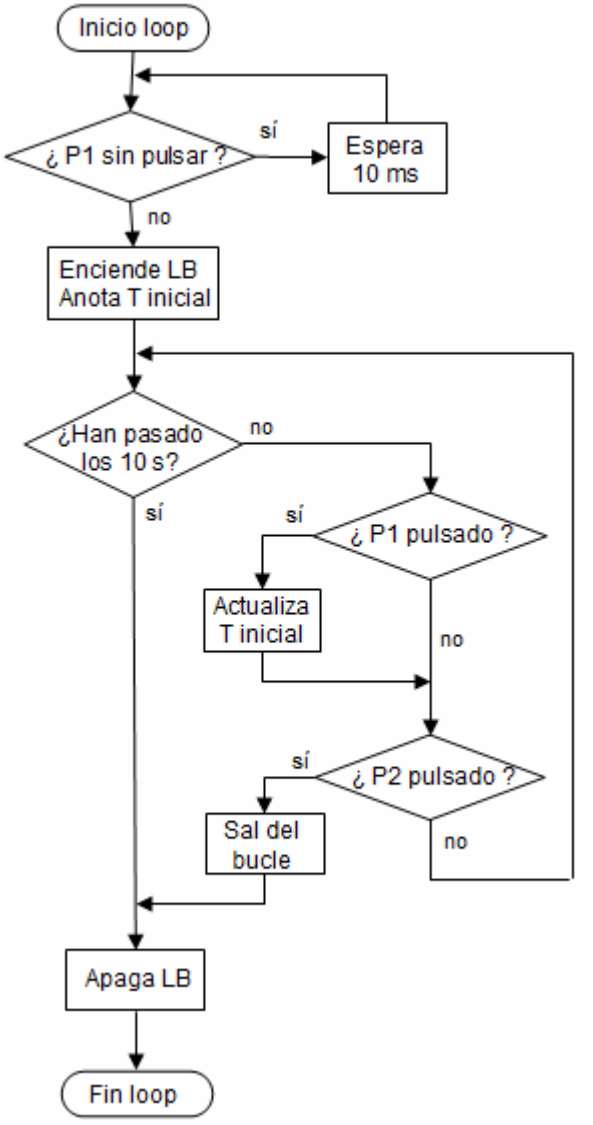
D.5.- Programa similar al anterior, pero para que se encienda un nuevo LED se tendrá que pulsar P1 tres veces y para apagar un LED habrá que pulsar P2 tres veces.

BLOQUE E: EL USO DE break Y DE millis()

E.0.- Ejemplo resuelto para ilustrar el uso de las funciones millis() y break. Elaborar un programa tal que al pulsar P1 el LED LB se enciende. Una vez encendido, al cabo de 10 s el LED se apaga solo. También se apaga el LED si antes de que transcurran los 10 segundos pulsamos P2. Si antes de pasar los 10 segundos se pulsa P1 vuelve a empezar a contar el tiempo de 10 segundos. Edítalo, carga el programa en la tarjeta, haz las conexiones y comprueba que funciona. Asegúrate de comprender lo que hace el programa antes de seguir.

```
int LB=3;
int P1=7;
int P2=11;
unsigned long tini; //variable para anotar
                    // el tiempo de inicio
                    // de la temporización

void setup() {
   pinMode(LB, OUTPUT);
   pinMode(P1, INPUT_PULLUP);
   pinMode(P2, INPUT_PULLUP);
   digitalWrite(LB,LOW); //Empezamos con
                         //el LED apagado
}
void loop() {
  while(digitalRead(P1)==HIGH) delay(10);
  digitalWrite(LB,HIGH);
  tini=millis();
  while(millis()-tini<10000){
    if(digitalRead(P1)==LOW) tini=millis();
    if(digitalRead(P2)==LOW) break;
        //Al ejecutarse break se sale del
        //bucle while aunque no se cumpla
        //su condición de haber pasado 10
        //segundos
  }
  digitalWrite(LB,LOW);
}
```



E.1.- Elabora un programa para un sistema formado por los dos pulsadores y un LED. El sistema estará a la espera de que se pulse P1. Tras esta pulsación, contará el tiempo transcurrido hasta que se pulse P2. Tras esta segunda pulsación, encenderá uno de los LEDs por el mismo tiempo transcurrido entre las pulsaciones.

E.2.- Elabora un programa para controlar un sistema formado por el LED verde, el LED blanco y los pulsadores P1 y P2. El funcionamiento será el siguiente: el LED verde estará permanentemente parpadeando a intervalos de 3 segundos. En cualquier momento, con pulsar brevemente el pulsador P1, el LED blanco cambiará de estado (no debe ser necesario mantener pulsado el pulsador un tiempo prolongado, sino que el cambio debe efectuarse de forma inmediata).

E.3.- Programa que controla un LED que se enciende con una cadencia inicial de 2 segundos. Cada vez que se pulse un pulsador P1 la cadencia se reduce en 0,25 segundos y así hasta llegar a un mínimo de 0,25 segundos. Cada vez que se pulse otro pulsador P2 la cadencia aumenta en 0,25 segundos y así hasta llegar a un máximo de 4 segundos.

Nota: en este problema no se podrán usar delay() para controlar los intervalos de encendido y apagado pues si la pulsación se produce durante la ejecución de delay() siendo más breve que la duración de dicha instrucción, el microprocesador no se entera de la pulsación.

BLOQUE F: LAS ENTRADAS ANALÓGICAS Y LAS SALIDAS PSEUDOANALÓGICAS

F.0.1.- Ejemplo resuelto. Cuando incide suficiente luz sobre la LDR1, el LED LB estará apagado. Cuando el nivel de luz incidente disminuye por debajo de un cierto valor umbral el LED se encenderá. Ten en cuenta que cuanta más luz incida sobre la LDR menos menor será el valor leído en la entrada analógica (A0 en este caso) por Arduino.

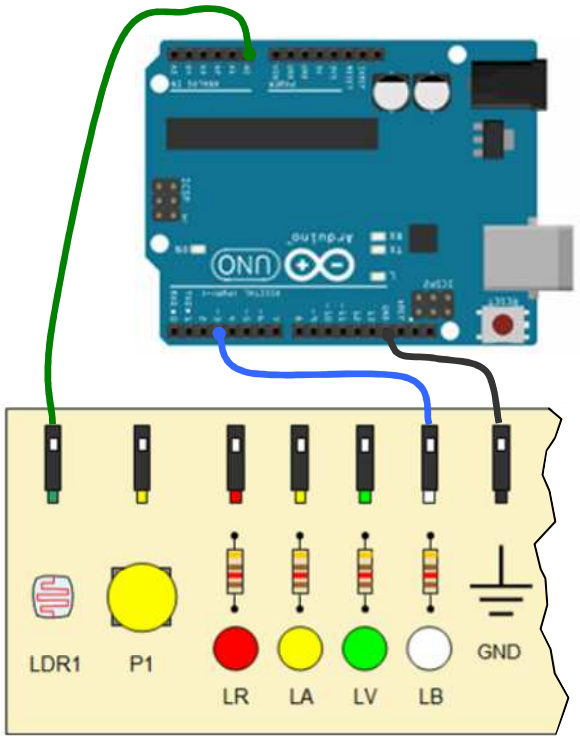
Prueba diferentes valores de umbral y observarás que varía la cantidad de oscuridad que hay que lograr sobre la LDR para que se encienda el LED.

¿Para qué crees que puede servir este circuito?

```
#define LDR1 A0
#define LB 3
int ValorLimite=300;

void setup() {
  pinMode(LDR1, INPUT_PULLUP);
  pinMode(LB, OUTPUT);
}

void loop() {
  if(analogRead(LDR1)>ValorLimite){
    digitalWrite(LB, HIGH);
  }
  else digitalWrite(LB,LOW);
}
```



F.0.2.- Ejemplo resuelto. Uso de salidas pseudoanalógicas (analogWrite) y del bucle for.

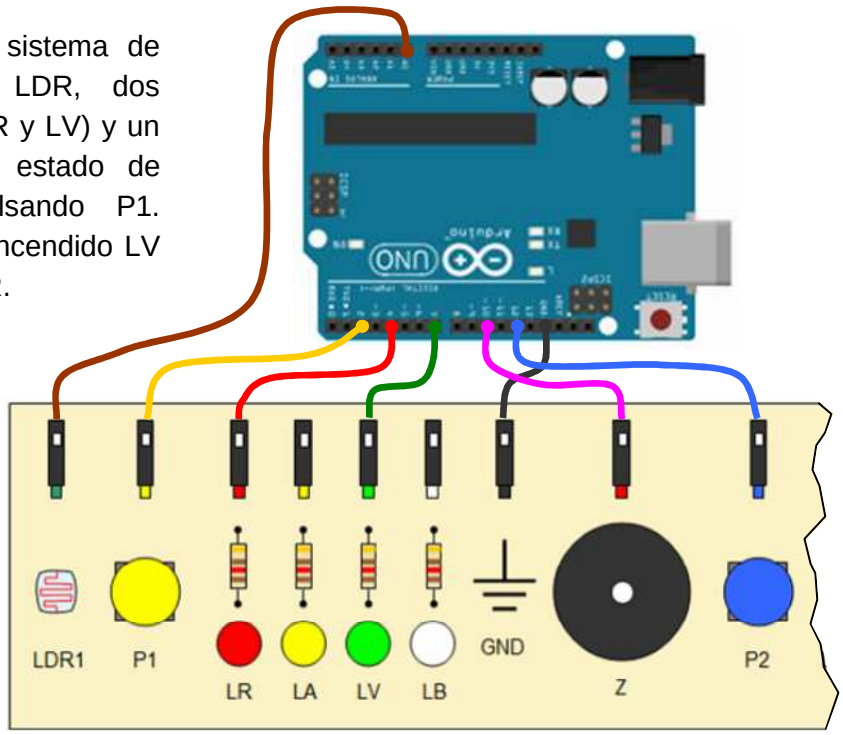
Con el programa que se adjunta se consigue la variación gradual de la intensidad de un LED conectado a una salida PWM (~). Nota: las salidas 3, 5, 6, 9, 10, y 11 son de tipo PWM en la tarjeta Arduino UNO.

El **bucle for** lo utilizamos para repetir un mismo código un número determinado de veces. Se sale del bucle bien cuando se ha llegado al número de veces previsto o bien cuando se ejecuta la instrucción **break**.

```
const int L1=3; //pin tipo PWM
void setup() {
  pinMode(L1, OUTPUT);
}
void loop() {
  for(int i=0;i<=255;i++){
    analogWrite(L1,i);
    delay(10);
  }
  for(int i=255;i>=0;i--){
    analogWrite(L1,i);
    delay(10);
  }
}
```

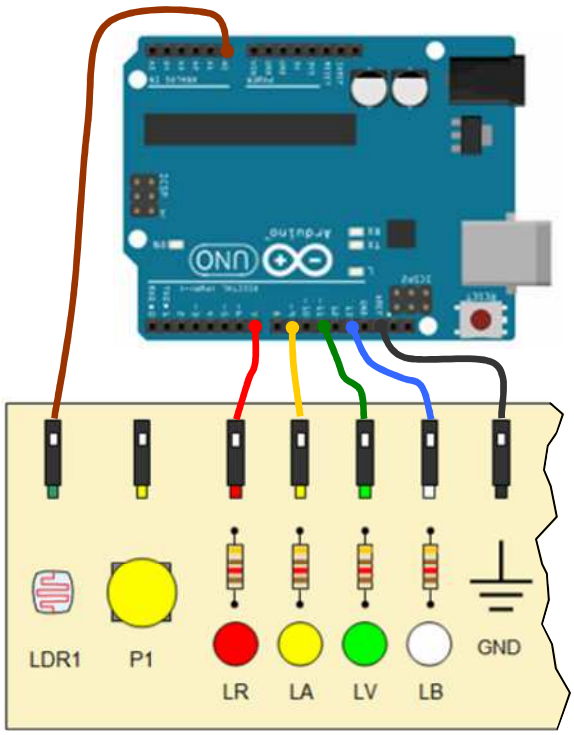
F.1.- Programa que controla un sistema de alarma que dispone de una LDR, dos pulsadores (P1 y P2), dos LED (LR y LV) y un zumbador. La alarma pasa del estado de desconectada a conectada pulsando P1. Cuando está desconectada está encendido LV y cuando está conectada lo está LR.

Si estando conectada se oscurece la LDR1 la alarma salta haciendo sonar el zumbador, el cual no se calla aunque se vuelva a iluminar la LDR1. Para desconectar la alarma o para desactivarla una vez ha saltado, se pulsará P2, con lo cual la alarma pasa a estado de desconectada.



F.2.- Elabora un programa para que en función del nivel de iluminación que incida sobre una LDR se ilumine un mayor o menor número de LEDs. Se dispondrá de 4 LED; con un nivel alto de iluminación no se encenderá ninguno. Con una oscuridad total se encenderán los cuatro. Para niveles intermedios se encenderán uno, dos o tres. Los niveles de luz los establecerás tú.

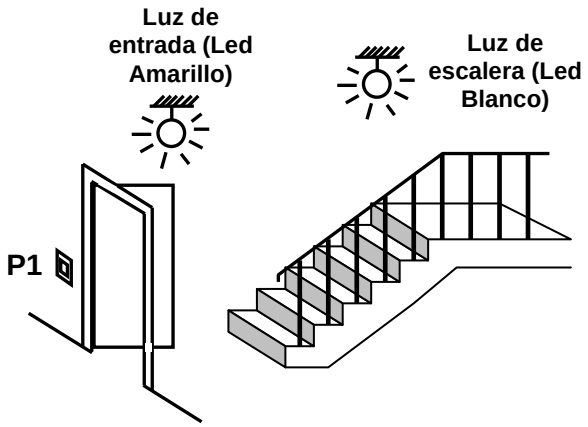
F.3.- Elabora un programa para que cuatro LEDs se enciendan sucesivamente (rojo, amarillo, verde, blanco,...) con una cadencia de 1 segundo cada uno cuando la LDR está totalmente destapada. Conforme se va aumentando la sombra sobre la LDR, la cadencia se hará más rápida (los intervalos de encendido y apagado de los LEDs serán más pequeños).

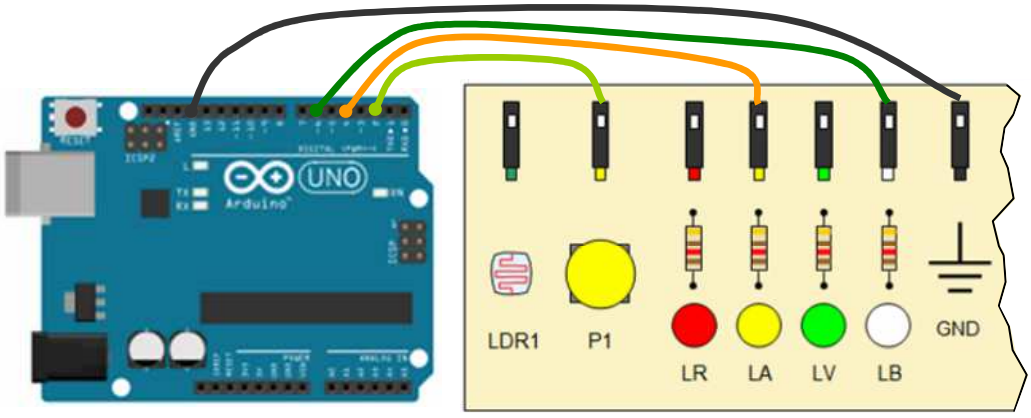


F.4.- Elabora un programa para controlar un sistema formado por los cuatro LEDs, el pulsador P1, la LDR1 y el zumbador. Inicialmente los cuatro LEDs estarán apagados. Cada vez que pasemos la mano por encima de la LDR oscureciéndola se encenderá un nuevo LED (empezará por el blanco hacia la izquierda) y cada vez que pulsemos P1 se apagará un LED. Si se llegaran a encender los cuatro LEDs (es decir, el LED rojo), pitará el zumbador como alarma, hasta que volvamos a rebajar el número de LEDs encendidos pulsando el pulsador.

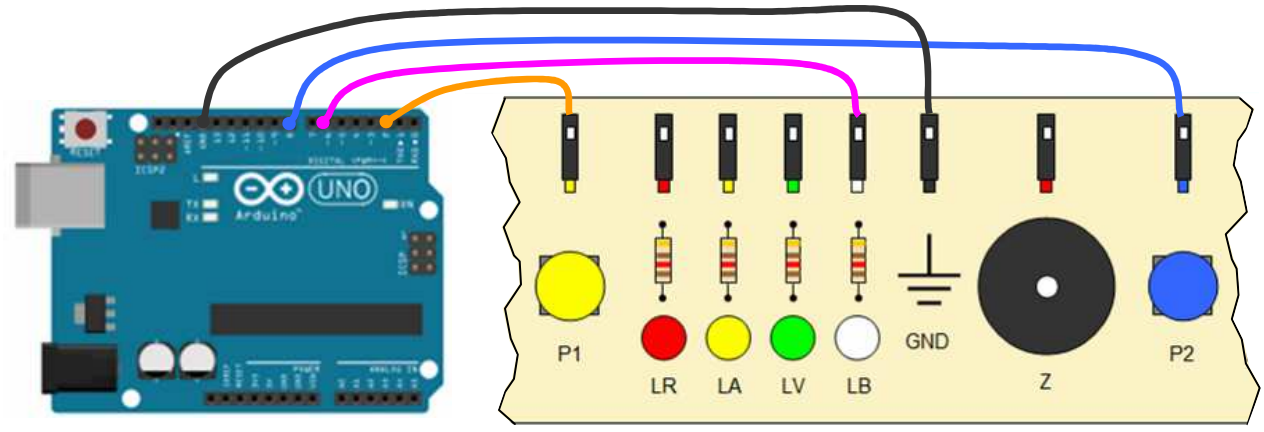
BLOQUE G: PROBLEMAS DIVERSOS

G.1.- Elabora el programa de control del apagado automático de las luces de escalera de un bloque de viviendas. Al pulsar el pulsador P1, se encenderán dos LEDs (simulan a las luces). El LED amarillo (que se supone que simula a la luz de la entrada del bloque) se apagará a los 5 s y el LED blanco (que se supone que simula a las luces de las escaleras) se apagará a los 15 s. Si antes de transcurrir estos tiempos se volviera a pulsar el pulsador P1 el cómputo de tiempo se reanudaría.

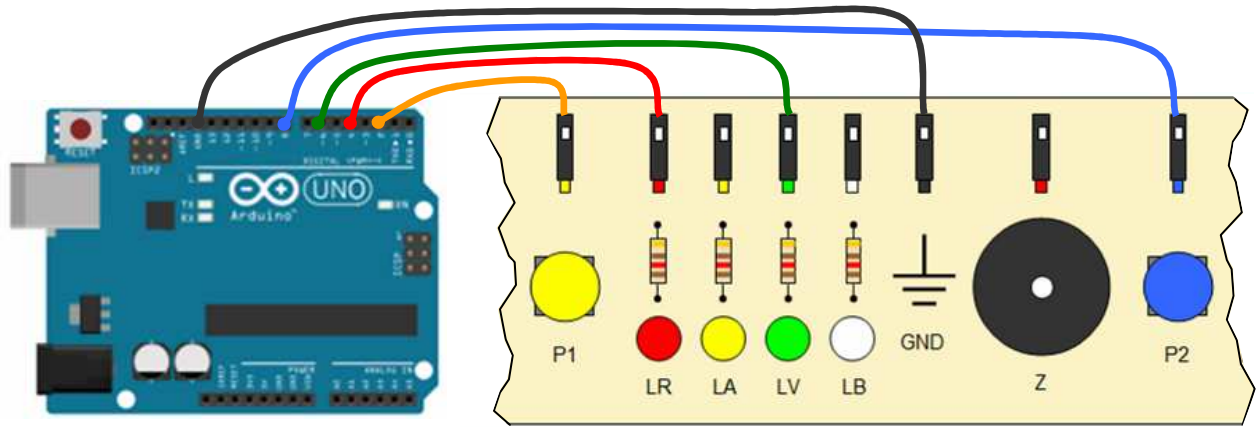




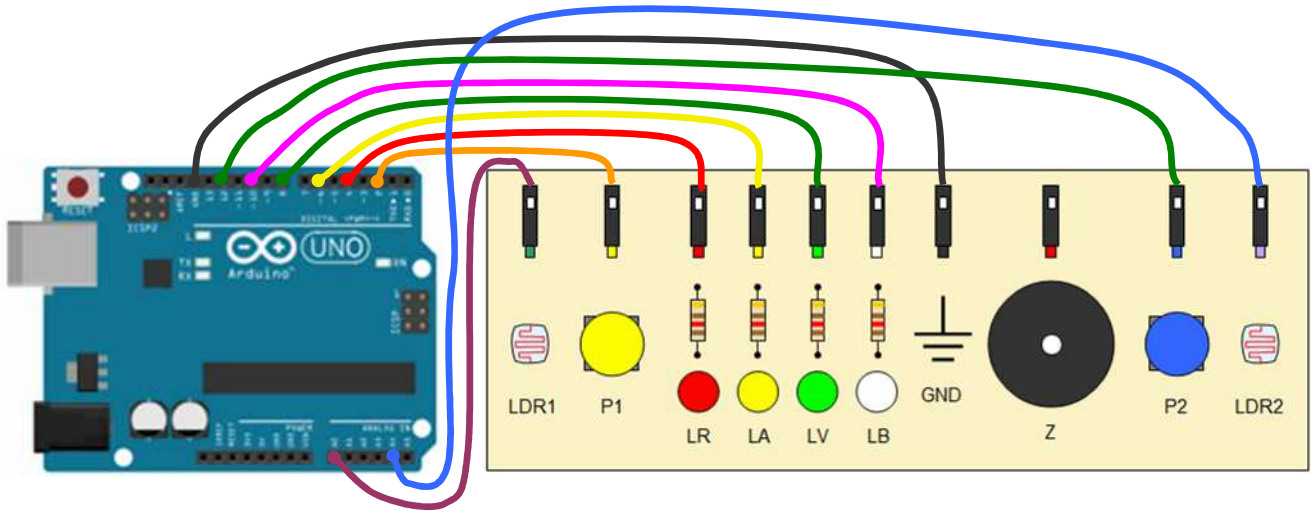
G.2.- Elabora un programa para el control del encendido y apagado automático del alumbrado de escalera de un bloque de viviendas que funcionará del siguiente modo: cada puerta de vivienda, mientras está cerrada, mantiene pulsado y cerrado un pulsador. Al abrir cualquiera de las puertas se deja de pulsar y se abre su pulsador correspondiente, haciendo que se encienda un LED que simula el alumbrado de la escalera. El LED se mantendrá encendido en tanto que alguna puerta esté abierta. Una vez cerradas todas las puertas, el LED se mantendrá encendido durante 10 segundos más. Si antes de terminar el tiempo de espera de 10 segundos se vuelve a abrir alguna puerta se reinicia de nuevo el ciclo. En nuestro caso, supondremos que sólo son dos viviendas ya que nuestro entrenador solo dispone de dos pulsadores.



G.3.- Elaborar el programa de control de un sistema formado por dos LEDs, LR y LV, y dos pulsadores, P1 y P2. Al pulsar P1 se enciende LR, al pulsar P2 se enciende LV y se apagan ambos al cabo de 6 segundos de la segunda pulsación. Si antes de pulsar el P1 se pulsa el P2 no ocurre nada. Para simplificar, lo haremos de forma que si se vuelve a pulsar el P1 antes de que se hayan apagado los LEDs no se le hace caso.



G.4.- Elabora un programa para el control del siguiente sistema: disponemos de dos LDR para contabilizar el número de personas que pasan por una puerta. Si se oscurece la LDR1 y a continuación las LDR2 la persona entra y si el orden de los oscurecimientos es el inverso, la persona sale. Cada vez que entra una persona se enciende un LED adicional, con un máximo de 4. Cada vez que sale una persona se apaga un LED, hasta un mínimo de 0. Se debe tener en cuenta que si se oscurece una LDR y tras su iluminación la siguiente en oscurecerse es ella misma, no se contará pues indicará que la persona, tras pasar por una LDR se ha vuelto hacia atrás antes de pasar por la otra.



G.5.- Elaborar el programa de control de un sistema con 4 LEDs, dos pulsadores (P1 y P2) y un zumbador (Z). Empiezan los 4 LEDs apagados. Pulso un número de veces el pulsador P1; cuando pulso P2 se encienden tantos LEDs como pulsaciones haya dado antes en el pulsador P1 (a partir de 4 pulsaciones se encienden sólo los 4 LEDs). Al pulsar de nuevo el pulsador P2 se apagan los LEDs, suena brevemente el zumbador (por ejemplo 0,5 segundos) y se empieza de nuevo.