



Proyectos

An aerial photograph of a landscape with a river, fields, and trees, tilted at an angle. The word "Proyectos" is overlaid in large, orange, 3D-style letters.

Daniel Gallardo García
Profesor de Tecnología
Jerez de la Frontera

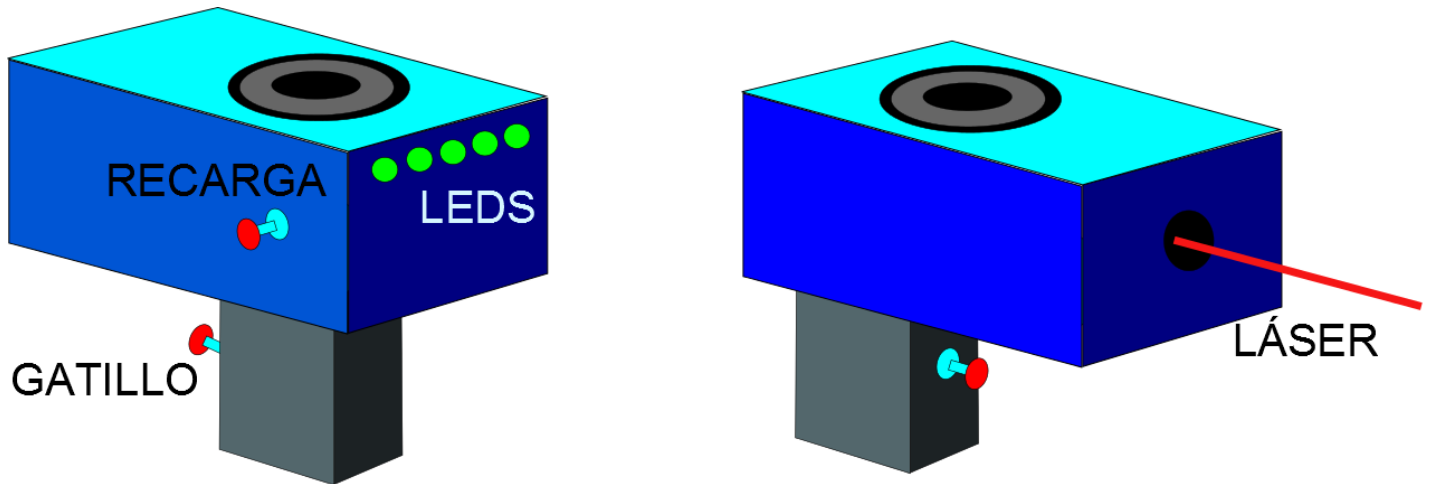


Índice de proyectos - 2016

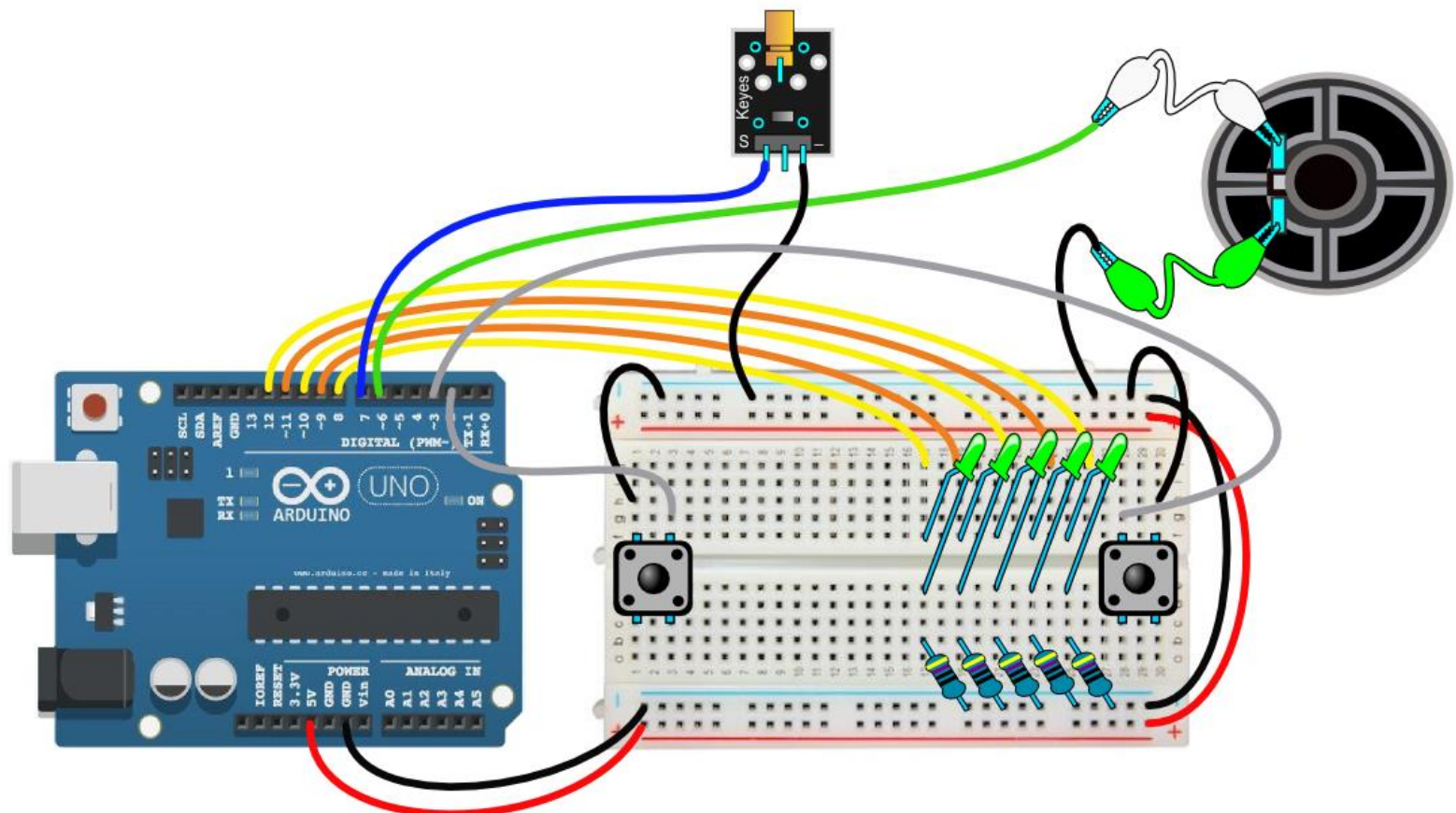
- 1.- Arduino Vs Zombies: Pistola de Rayos Láser
- 2.- Arduino Vs Zombies: Dianas Móviles
- 3.- ¿Cuánto sabes de Arduino? Verdadero/Falso
- 4.- Boom!
- 5.- Guitar Hero
- 6.- Habilidad digital
- 7.- Arduinio Car
- 8.- Arduino Plane
- 9.- Pompas de jabón machine
- 10.- Portero de fútbol

1.- Arduino Vs Zombies: Pistola de Rayos Láser

Es una pistola que disparará un rayo láser. Tendrá una recámara para 5 balas (o rayos), de tal manera que, al menos, cada cinco disparos habrá que hacer una recarga. Consta de un pulsador como gatillo y otro para recargar. Las balas disponibles se visualizarán a través de 5 LEDs, y también dispondrá de efectos sonoros a través de un altavoz.



El montaje y su correspondiente sketch podrían ser los siguientes:



```

/*Pistola Láser
 IES Laguna de Tollón (El Cuervo de Sevilla)
 Daniel Gallardo García */

int pinLed[] = {8, 9, 10, 11, 12};
int pinAltavoz = 6;
int pinLaser = 7;
int pinGatillo = 2;
int pinRecarga = 3;
int balas = 5;
int gatillo; //será el valor de la lectura de dicho pulsador
int recarga; //será el valor de la lectura de dicho pulsador
int melodiaRecarga[] = {349, 440, 523, 659, 698};
int ritmoRecarga[] = {1, 1, 1, 1, 2};

void setup() {
  for(int i=6; i<13; i++) pinMode(i, OUTPUT);
  for(int i=2; i<4; i++) pinMode(i, INPUT_PULLUP);
  for(int i=0; i<5; i++) digitalWrite(pinLed[i], HIGH);
}

void loop() {
  gatillo = digitalRead(pinGatillo);
  recarga = digitalRead(pinRecarga);
  if(!gatillo && balas==0) {
    tone(pinAltavoz, 196, 200);
    delay(250);
    tone(pinAltavoz, 147, 200);
    delay(250);
  }
  if(!gatillo && balas>0) {
    tone(pinAltavoz, 1397);
    digitalWrite(pinLaser, HIGH);
    delay(300);
    digitalWrite(pinLaser, LOW);
    balas--;
    digitalWrite(pinLed[balas], LOW);
    for(int i=1397; i>100; i--) {
      tone(pinAltavoz, i);
      delayMicroseconds(50);
    }
    noTone(pinAltavoz);
  }
  if(!recarga) cargarBalas();
}

void cargarBalas() {
  for(int i=0; i<5; i++) digitalWrite(pinLed[i], LOW);
  for(int i=0; i<5; i++) {
    tone(pinAltavoz, melodiaRecarga[i], ritmoRecarga[i]*200);
    digitalWrite(pinLed[i], HIGH);
    delay(ritmoRecarga[i]*200 + 50);
  }
  balas = 5;
}

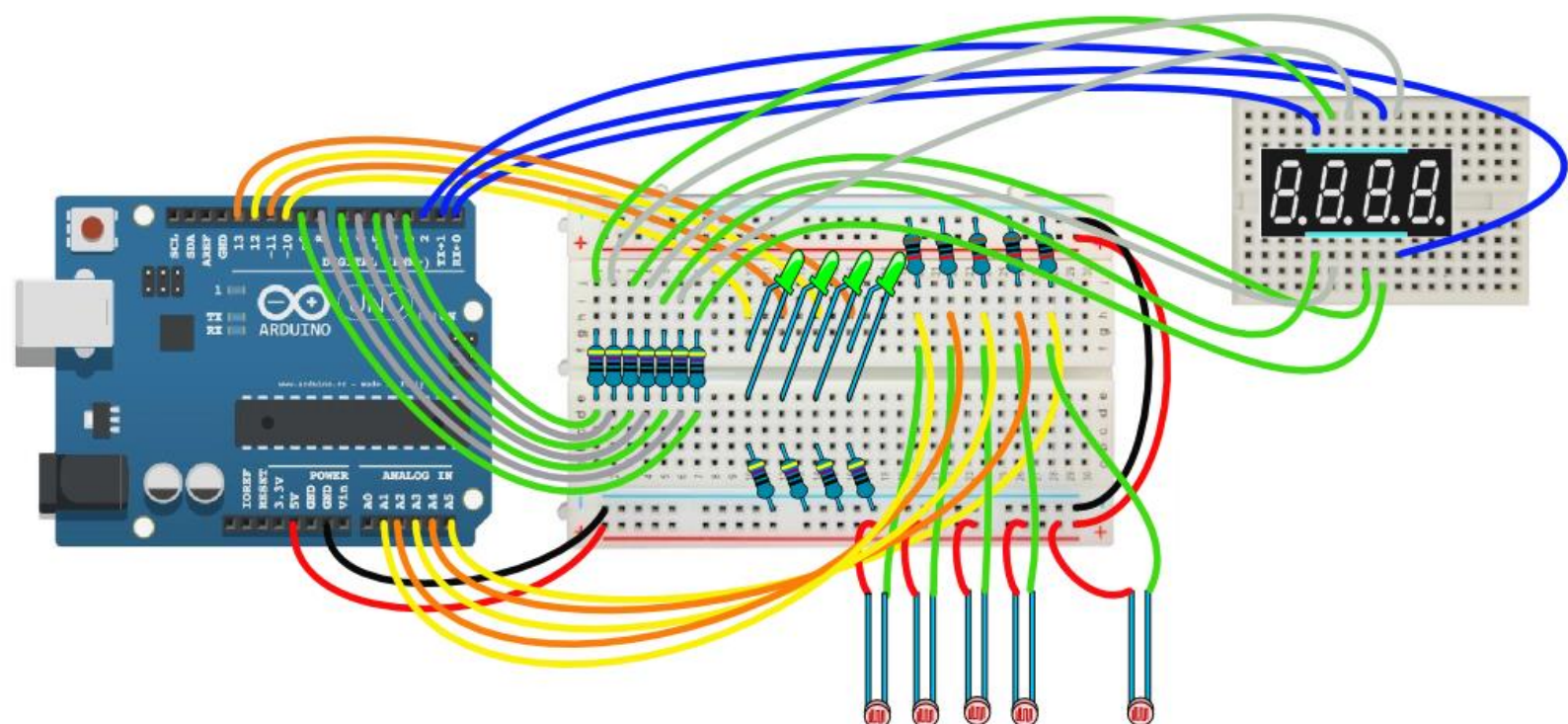
```

2.- Arduino Vs Zombies: Dianas Móviles

Este proyecto es complemento de la pistola láser. Consiste en un panel donde hay 4 posibles objetivos a los que disparar. Dispone de un display de 4 dígitos para indicar el nivel (del 1 al 7) y el número de aciertos. Solo se activará un objetivo, que se indicará a través de un LED. Además, hay opción de reiniciar la partida si se dispara a una LDR para que realice el reset.



El montaje y su correspondiente sketch podrían ser los siguientes:



```

/* Dianas móviles de Arduino Vs Zombies
   IES Laguna de Tollón (El Cuervo de Sevilla)
   Daniel Gallardo García */

int segmentosPines[] = {3, 4, 5, 6, 7, 8, 9}; // (a,b,c,d,e,f,g) no nos hará falta el puntoint
digitosPines[]={0, 1, 2}; // (digito1,digito3,digito4) no nos hará falta el segundo dígito
int pinReset = 0; //conecto el pin 0 con el pin Reset mediante un cable
byte segmentosNumeros[10][7]={ {1,1,1,1,1,1,0}, //número 0
                                {0,1,1,0,0,0,0}, //número 1
                                {1,1,0,1,1,0,1}, //número 2
                                {1,1,1,1,0,0,1}, //número 3
                                {0,1,1,0,0,1,1}, //número 4
                                {1,0,1,1,0,1,1}, //número 5
                                {1,0,1,1,1,1,1}, //número 6
                                {1,1,1,0,0,0,0}, //número 7
                                {1,1,1,1,1,1,1}, //número 8
                                {1,1,1,1,0,1,1} }; //número 9

int pinLed[] = {10, 11, 12, 13};
int pinLdr[] = {1, 2, 3, 4, 5}; //el LDR del pin 5 será para el Reset
int lecturaLdrCalibrado[5]; //lecturas iniciales de las LDRs, para calibrar
int lecturaLdr[5]; //lecturas posteriores de las LDRs, para saber si han sido disparadas o no
int unidadesMarcador = 0, decenasMarcador = 0;
int nivel = 1, objetivo = 1; //cada nivel se compone de 10 objetivos (menos tiempo a mayor nivel)
int marcador = 0; //almacenaré el número de aciertos en el marcador
int acierto = 0; //acierto será 1 si se ha disparado con acierto sobre el objetivo activo
int numeroDianaNuevo = 4, numeroDianaViejo; //nunca será 5, pero es para que no se repita la diana
unsigned long tiempo = 0, intervalo = 5000; //será el tiempo activo de cada objetivo (zombie diana)

void setup() {
    for(int i=0; i<15; i++) pinMode(i, OUTPUT);
    for(int i=0; i<5; i++) lecturaLdrCalibrado[i] = analogRead(pinLdr[i]);
    //guardo los valores iniciales de lectura de las LDRs
    randomSeed(analogRead(A0));
}

void loop() {
    for(nivel=1; nivel<8; nivel++) {
        luminoso();
        efectoLeds();
        intervalo = intervalo - nivel*100;
        for(int obj=0; obj<10; obj++) {
            numeroDianaViejo = numeroDianaNuevo; //voy a asegurarme de que el siguiente objetivo o diana...
            do {
                numeroDianaNuevo = random(4);
            } while(numeroDianaNuevo == numeroDianaViejo); //... no vuelva a ser el mismo que había antes.
            for(int i=0; i<4; i++) digitalWrite(pinLed[i], LOW);
            digitalWrite(pinLed[numeroDianaNuevo], HIGH); //enciendo el LED del objetivo activo
            tiempo = millis();
            acierto = 0;
            do {
                luminoso();
                lecturaLdr[numeroDianaNuevo] = analogRead(pinLdr[numeroDianaNuevo]);
                lecturaLdr[4] = analogRead(A5);
                if(lecturaLdr[numeroDianaNuevo] > lecturaLdrCalibrado[numeroDianaNuevo] + 200) {
                    acierto = 1;
                    marcador++;
                }
            }
            if(lecturaLdr[4] > lecturaLdrCalibrado[4] + 200) reset(); //resetearemos Arduino
        } while(tiempo+intervalo > millis() && acierto == 0);
    }
}

finDeFiesta();

void luminoso() {
    decenasMarcador = marcador / 10;
    unidadesMarcador = marcador - decenasMarcador*10;
    int valoresDigito[] = {nivel, decenasMarcador, unidadesMarcador};
    for(int digito=2; digito>=0; digito--) { /*haré todo dentro de un for: iré llamando a
        cada uno de los dígitos e iré colocando en el display su valor correspondiente*/
        for(int i=0; i<3; i++) digitalWrite(digitosPines[i], LOW);
        //pongo a LOW todos los pines de dígitos y solo activo uno
        digitalWrite(digitosPines[digito], HIGH);
        for(int j=0; j<7; j++) digitalWrite(segmentosPines[j], !segmentosNumeros[valoresDigito[digito]][j]);
        delay(5); //entre dígito y dígito debemos hacer una pequeña pausa
    }
}

void efectoLeds() { //los LEDs parpadearán 10 veces

```

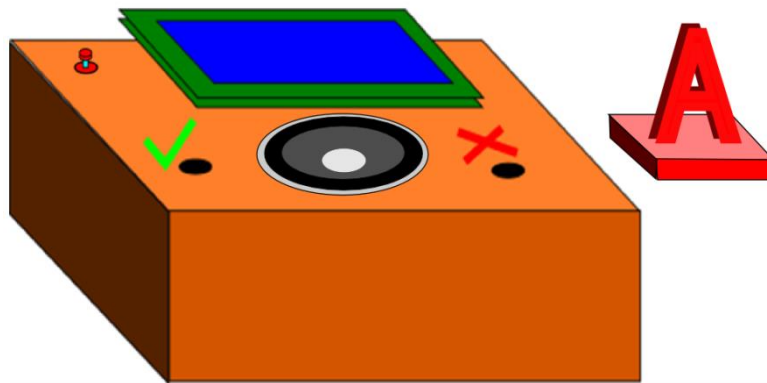
```
for(int j=0; j<10; j++) {
    delay(100);
    for(int i=0; i<4; i++) digitalWrite(pinLed[i], HIGH);
    delay(100);
    for(int i=0; i<4; i++) digitalWrite(pinLed[i], LOW);
}

void reset() {
    marcador = 0;
    nivel = 1;
    intervalo = 5000;
    for(int i=0; i<3; i++) efectoLeds();
    for(int i=0; i<5; i++) lecturaLdrCalibrado[i] = analogRead(pinLdr[i]);    //vuelvo a calibrar
}

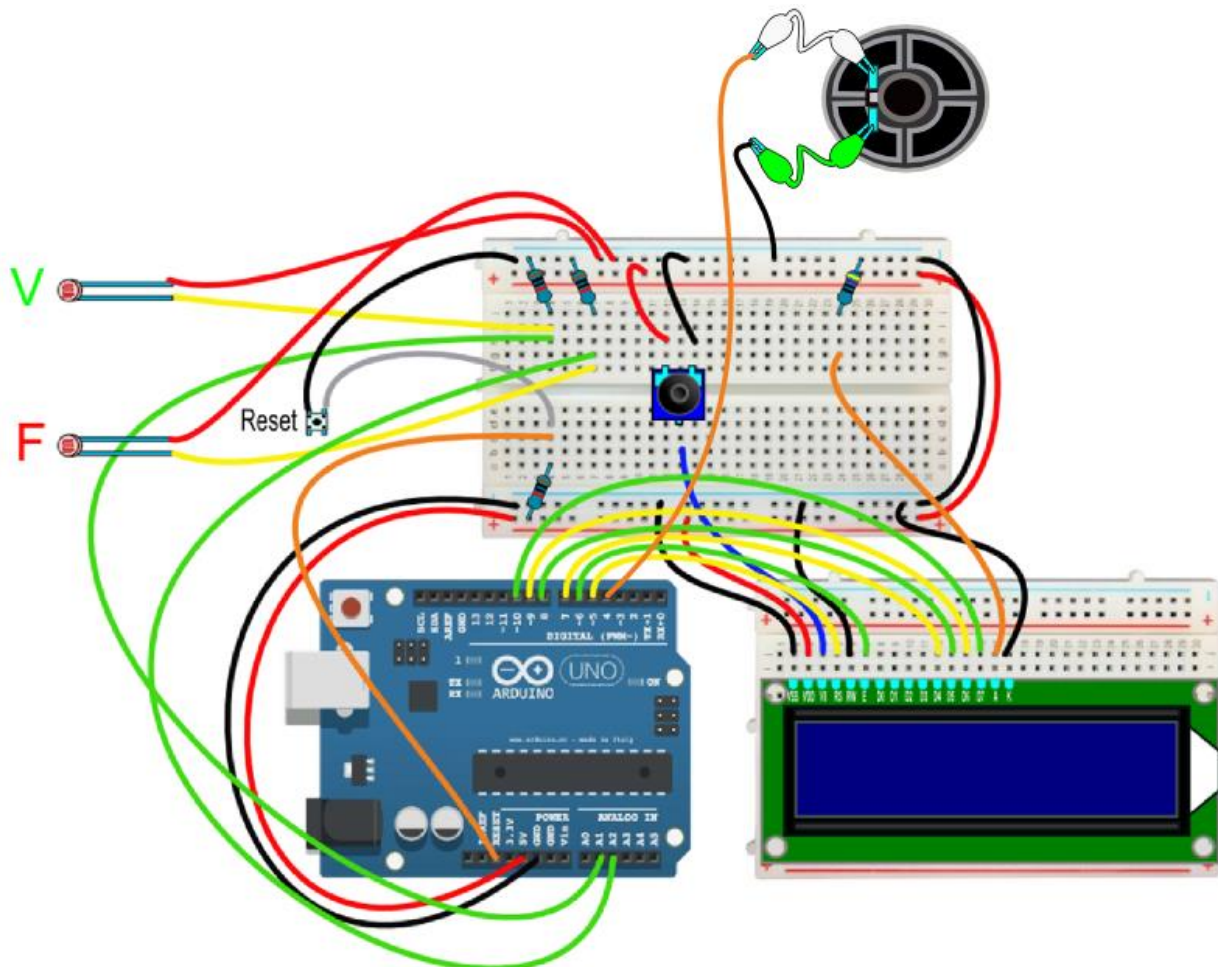
void finDeFiesta() {
    nivel = 7;    //al finalizar el bucle for de loop, "nivel" acaba con valor 8
    for(int i=0; i<1000; i++) {
        luminoso();
        for(int j=0; j<4; j++) digitalWrite(pinLed[j], LOW);    //apago todos los LEDs
        digitalWrite(pinLed[random(4)], HIGH);    //y enciendo uno al azar
    }
    reset();
}
```

3.- ¿Cuánto sabes de Arduino? Verdadero/Falso

Se trata de un juego donde aparecerán afirmaciones a través de una pantalla LCD, y que solo admiten dos respuestas posibles: Verdadero o Falso. Para responder se deberá tapar, con algún tipo de cajita, uno de las dos LDR de que dispone el juego. En la pantalla deberá aparecer el número de pregunta y el número de aciertos. Por último, se realizará un veredicto por parte de la máquina hacia el jugador, en función del número de aciertos. También dispondrá de un pulsador para resetear el juego para que comience desde el principio.



El montaje y su correspondiente sketch podrían ser los siguientes:




```

/*Juego de Preguntas tipo Verdadero/Falso: ¿Cuánto sabes de Arduino?
 IES Laguna de Tollón (El Cuervo de Sevilla)
 Daniel Gallardo García */

#include <LiquidCrystal.h>
LiquidCrystal lcd(5, 6, 7, 8, 9, 10); //configuración pantalla LCD (RS,E,D4,D5,D6,D7)
int pinAltavoz=4;
char* pregunta[]={
  "Para que Arduino","espere 2 s.,"debo poner:","      delay(2);",
  "Reservamos el","color negro","para conexiones","a tierra (GND)",
  "int x=5;","declara e inicia","una variable","llamada x",
  "Debo conectar en","serie a un LED","una resistencia","mayor de 10K",
  "La configuracion","de Arduino se","realiza dentro","de void pa lla",
  "La siguiente or-","den esta escrita","correctamente:","pinMode(3, LOW);",
  "Los cables rojos","se reservan para","conexiones a","5 V",
  "Los pines del 0","al 13 pueden","usarse como","salida digital",
  "Un pulsador se","puede utilizar","como entrada","analogica",
  "El bucle: for(","i=1; i<3; i++)","realiza el bucle","3 veces",
  "La LDR se","puede utilizar","como entrada","analogica",
  "Arduino solo","dispone de 6","pines de entrada","analogica",
  "Arduino es un","invento de","americanos y","japoneses",
  "Arduino es un","producto de","software y","hardware libre",
  "Jugar conmigo","ha sido lo mejor","de esta Feria","de Tecnologica" //pregunta broma
};
int respuesta[] = {0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1};
int numero; //es el número de pregunta, que irá del 1 al 20
int ldrVerdadero; //el la entrada analógica de la LDR del Verdadero
int ldrFalso; //es la entrada analógica de la LDR del Falso
int calibradoLdrVerdadero;
int calibradoLdrFalso;
int tiempoRespuesta;
int valorRespuesta;
int score = 0;
//constantes necesaria para la música
float do2=130.813, do_2=138.591, re2=146.832, re_2=155.563, mi2=164.814, fa2=174.614,
fa_2=184.997, sol2=195.998, sol_2=207.652, la2=220, la_2=233.082, si2=246.942,
do3=261.626, do_3=277.183, re3=293.665, re_3=311.127, mi3=329.628, fa3=349.228,
fa_3=369.994, sol3=391.995, sol_3=415.305, la3=440, la_3=466.164, si3=493.883,
do4=523.251, do_4=554.365, re4=587.330, re_4=622.254, mi4=659.255, fa4=698.456,
fa_4=739.989, sol4=783.991, sol_4=830.609, la4=880, la_4=932.328, si4=987.767;
float melodiaV[] = {mi4, re4, mi4, re4, do4, si3, la3, sol_3, la3, la3};
int ritmoV[] = {1, 1, 8, 1, 1, 1, 1, 8, 1, 8};
float melodiaF[] = {do3, re_3, si2, re3, la_2, do_3, la2, do3};
int ritmoF[] = {1,8,1,8,1,8,1,32};

void setup(){
  lcd.begin(16, 2);
  pinMode(pinAltavoz, OUTPUT); //será la salida del altavoz
  lcd.print(" BIENVENID@ AL");
  lcd.setCursor(0, 1);
  lcd.print(" Arduino-TEST");
  delay(2000);
  calibradoLdrVerdadero = analogRead(A1);
  calibradoLdrFalso = analogRead(A2);
}

void loop(){
  //El número de pregunta se desplaza de derecha a izquierda
  for(int p=0; p<15; p++) {
    lcd.clear();
    lcd.setCursor(16, 0);
    lcd.print("PREGUNTA No ");
    numero = p + 1;
    lcd.print(numero);
    for(int a=0; a<15; a++) {
      lcd.scrollDisplayLeft();
      delay(200);
    }
    //El número de pregunta parpadea tres veces
    for(int b=0;b<3;b++) {
      lcd.clear();
      delay(500);
      lcd.setCursor(1,0);
      lcd.print("PREGUNTA No ");
      lcd.print(numero);
      delay(500);
    }
    delay(500);
    //Aparecerá en la pantalla los cuatro fragmentos de la pregunta

```

```

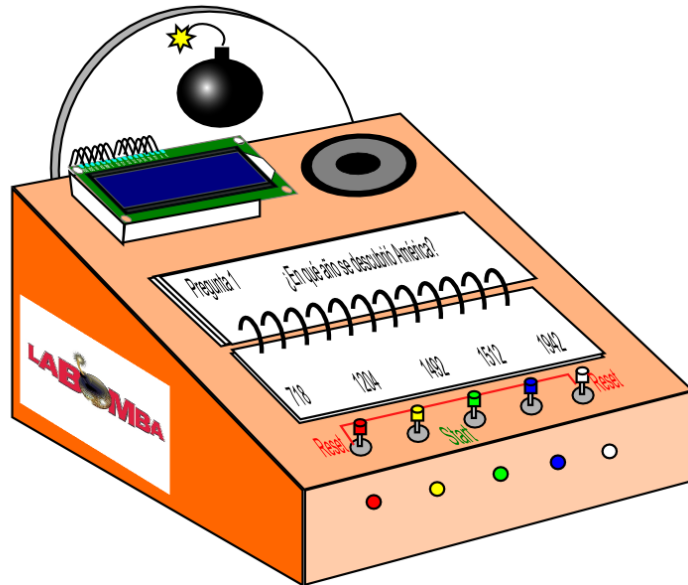
lcd.clear();
lcd.print(pregunta[p*4]);
lcd.setCursor(0,1);
lcd.print(pregunta[p*4+1]);
delay(3000);
lcd.clear();
lcd.print(pregunta[p*4+2]);
lcd.setCursor(0,1);
lcd.print(pregunta[p*4+3]);
delay(3000);
tiempoRespuesta=200;
    //Aparecerá un "?" girando esperando a que se responda
do {
    lcd.clear();
    lcd.setCursor(4,0);
    lcd.print("?");
    delay(tiempoRespuesta);
    for(int c=0;c<7;c++){
        lcd.scrollDisplayRight();
        delay(tiempoRespuesta);
    }
    lcd.clear();
    lcd.setCursor(11,1);
    lcd.print("?");
    delay(tiempoRespuesta);
    for(int d=0;d<7;d++) {
        lcd.scrollDisplayLeft();
        delay(tiempoRespuesta);
    }
    ldrVerdadero=analogRead(A1);
    ldrFalso=analogRead(A2);
    tiempoRespuesta=tiempoRespuesta-20;
    if(tiempoRespuesta<50) tiempoRespuesta=50;
} while (ldrVerdadero>calibradoLdrVerdadero-300 && ldrFalso>calibradoLdrFalso-300);
    //Comprobación de si la respuesta es correcta o falsa
if(ldrVerdadero<calibradoLdrVerdadero-299) valorRespuesta=1;
else valorRespuesta=0;
if(valorRespuesta==respuesta[p]) {
    score++;
    lcd.clear();
    if(respuesta[p]==0) lcd.print("F");
    else lcd.print("V");
    lcd.setCursor(4,0); lcd.print("Correcto !!!");
    lcd.setCursor(0,1); lcd.print("Puntos: "); lcd.print(score);
    lcd.setCursor(11,1); lcd.print("de "); lcd.print(numero);
    for(int i=0; i<sizeof(ritmoV)/2; i++) {
        tone(pinAltavoz,melodiaV[i],ritmoV[i]*100);
        delay(ritmoV[i]*100+10);
    }
    delay(1000);
}
else {
    lcd.clear();
    if(respuesta[p]==0) lcd.print("F");
    else lcd.print("V");
    lcd.setCursor(4,0); lcd.print("Fallaste !!!");
    lcd.setCursor(0,1); lcd.print("Puntos: "); lcd.print(score);
    lcd.setCursor(11,1); lcd.print("de "); lcd.print(numero);
    for(int j=0; j<sizeof(ritmoF)/2; j++) {
        tone(pinAltavoz,melodiaF[j],ritmoF[j]*50);
        delay(ritmoF[j]*50+1);
    }
    delay(1000);
}
}
    //Marcador total y calificación del jugador
lcd.clear();
lcd.setCursor(16,0); lcd.print(score); lcd.print(" puntos...");
for(int e=0;e<15;e++) {
    lcd.scrollDisplayLeft();
    delay(200);
}
delay(2000);
for(int f=0;f<12;f++) {
    lcd.scrollDisplayLeft();
    delay(200);
}
lcd.clear();
lcd.setCursor(16,1);
if(score<7) lcd.print("PESSSSSSIMO");
else if(score<12) lcd.print("MEDIOCRIDAD");

```

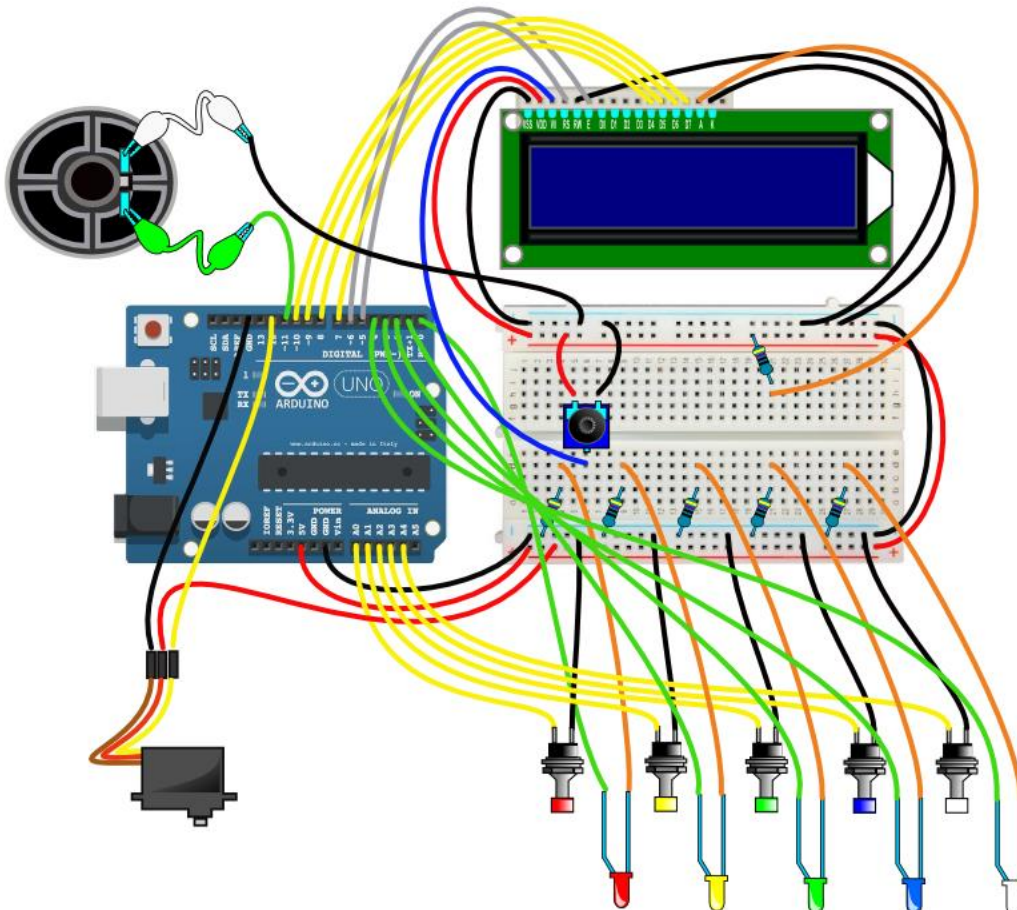
```
else lcd.print("ERES UN FRIKI!!");
for(int g=0;g<15;g++) {
  lcd.scrollDisplayLeft();
  delay(200);
}
delay(3000);
}
```

4.- Boom!

Es una versión Arduínica del programa de Antena3 Boom! Consiste en desactivar bombas cortando cables (en este caso, apretando pulsadores). En el caso de desactivar todas las respuestas incorrectas, el juego nos felicitará (con música, mostrando una carita sonriente y con un mensaje en la pantalla LCD) y pasaríamos a la siguiente pregunta. En caso de apretar la respuesta correcta, explotará la bomba (imagen de boom!, sonido de fracaso, mensaje en la pantalla, LEDs parpadeando).



El montaje y su correspondiente sketch podrían ser los siguientes:




```

        error = 1;
        for(int f=880; f>220; f--) {
            tone(pinAltavoz, f);
            delay(2);
        }
    }
    else {
        for(int f=220; f<880; f+=10) {
            tone(pinAltavoz, f);
            delay(2);
        }
    }
    noTone(pinAltavoz);
}
}
if(!digitalRead(pinPulsador[0]) && !digitalRead(pinPulsador[4])) resetea();
//si apretamos los pulsadores de los extremos, se reseteará

cablesSinCortar = 0;
for(int i=0; i<5; i++) cablesSinCortar += estadoCable[i];
} while (cablesSinCortar > 1 && tiempoRestante > 0 && error == 0);
if(tiempoRestante <= 0 || error == 1) perder(numeroPregunta);
else if(numeroPregunta < sizeof(cableCorrecto)/2 - 1) {
    siguientePregunta();
    numeroPregunta++;
}
else ganador();
}

void bienvenida() {
    jugar = 0;
    lcd.print(" BIENVENID@ A...");
    lcd.setCursor(0, 1);
    lcd.print("      BOOM!!!");
    do {
        for(int i=0; i<5; i++) {
            digitalWrite(pinLed[i], HIGH);
            digitalWrite(pinLed[i-1], LOW);
            delay(100);
            if(!digitalRead(pinPulsador[2])) jugar = 1;
        }
        digitalWrite(pinLed[4], LOW);
    } while(jugar == 0); //estará es estado de standby hasta que pulsemos el pulsador de enmedio
}

void resetea() {
    asm volatile("jmp 0");
}

void pantallaPregunta(int n) {
    lcd.clear();
    lcd.setCursor(16, 0);
    lcd.print("PREGUNTA No ");
    lcd.print(n);
    for(int i=0; i<15; i++) {
        lcd.scrollDisplayLeft();
        delay(200);
    }
    for(int i=0; i<3; i++) {
        lcd.clear();
        delay(200);
        lcd.setCursor(1, 0);
        lcd.print("PREGUNTA No ");
        lcd.print(n);
        delay(200);
    }
}

void perder(int n) {
    servomotor.write(0);
    lcd.clear();
    lcd.setCursor(5, 0);
    lcd.print(" BOOM! ");
    for(int j=0; j<10; j++) {
        for(int i=0; i<5; i++) digitalWrite(pinLed[i], HIGH);
        lcd.setCursor(random(-5, 14), random(1));
        lcd.print(" BOOM! ");
        sonidoBomba();
        for(int i=0; i<5; i++) digitalWrite(pinLed[i], LOW);
        lcd.setCursor(random(-5, 14), random(2));
    }
}

```

```

    lcd.print(" BOOM! ");
    sonidoBomba();
}
lcd.clear();
lcd.print("Has llegado a la");
lcd.setCursor(0, 1);
lcd.print("pregunta No ");
lcd.print(n);
delay(3000); //sustituir por canción
lcd.clear();
lcd.print("Para mi, eres un");
lcd.setCursor(0, 1);
if(n < 4) lcd.print(" PARDILLO");
else if(n < 7) lcd.print(" ENTERAILLO");
else lcd.print(" EMPOLLON");
delay(5000); //sustituir por canción
resetea();
}

void siguientePregunta() {
    servomotor.write(70);
    lcd.clear();
    lcd.print(" CORRECTO!!!");
    lcd.setCursor(0, 1);
    lcd.print("pasa la pagina");
    for(int i=0; i<5; i++) digitalWrite(pinLed[i], LOW);
    for(int i=0; i<5; i++) {
        digitalWrite(pinLed[i], HIGH);
        tone(pinAltavoz, melodiaCorrecto[i], ritmoCorrecto[i]*150);
        delay(ritmoCorrecto[i]*150+10);
    }
    delay(4000);
}

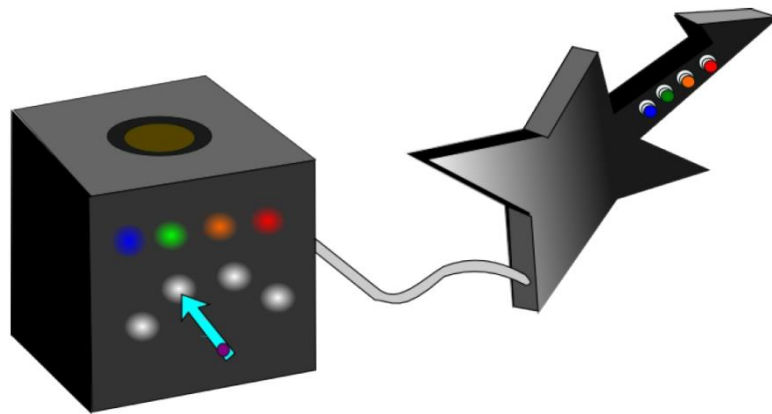
void ganador() {
    servomotor.write(140);
    lcd.clear();
    lcd.print(" ENHORABUENA!!!");
    lcd.setCursor(0, 1);
    lcd.print("Eres en CAMPEON!");
    for(int j=0; j<sizeof(ritmoGanador)/2; j++) {
        tone(pinAltavoz, melodiaGanador[j], ritmoGanador[j]*200);
        for(int i=0; i<5; i++) digitalWrite(pinLed[i], LOW);
        indiceLed = j % 5;
        digitalWrite(pinLed[indiceLed], HIGH);
        delay(ritmoGanador[j]*200 + 20);
    }
    delay(2000);
    jugar = 0;
    do {
        for(int i=0; i<5; i++) {
            digitalWrite(pinLed[i], HIGH);
            digitalWrite(pinLed[i-1], LOW);
            delay(100);
            if(!digitalRead(pinPulsador[0]) && !digitalRead(pinPulsador[4])) jugar = 1;
        }
        digitalWrite(pinLed[4], LOW);
    } while(jugar == 0); //estará es estado de standby hasta que pulsemos el pulsador de enmedio
    resetea();
}

void sonidoBomba() {
    for(int f=880; f>220; f-=5) {
        tone(pinAltavoz, f);
        delay(2);
    }
    noTone(pinAltavoz);
}

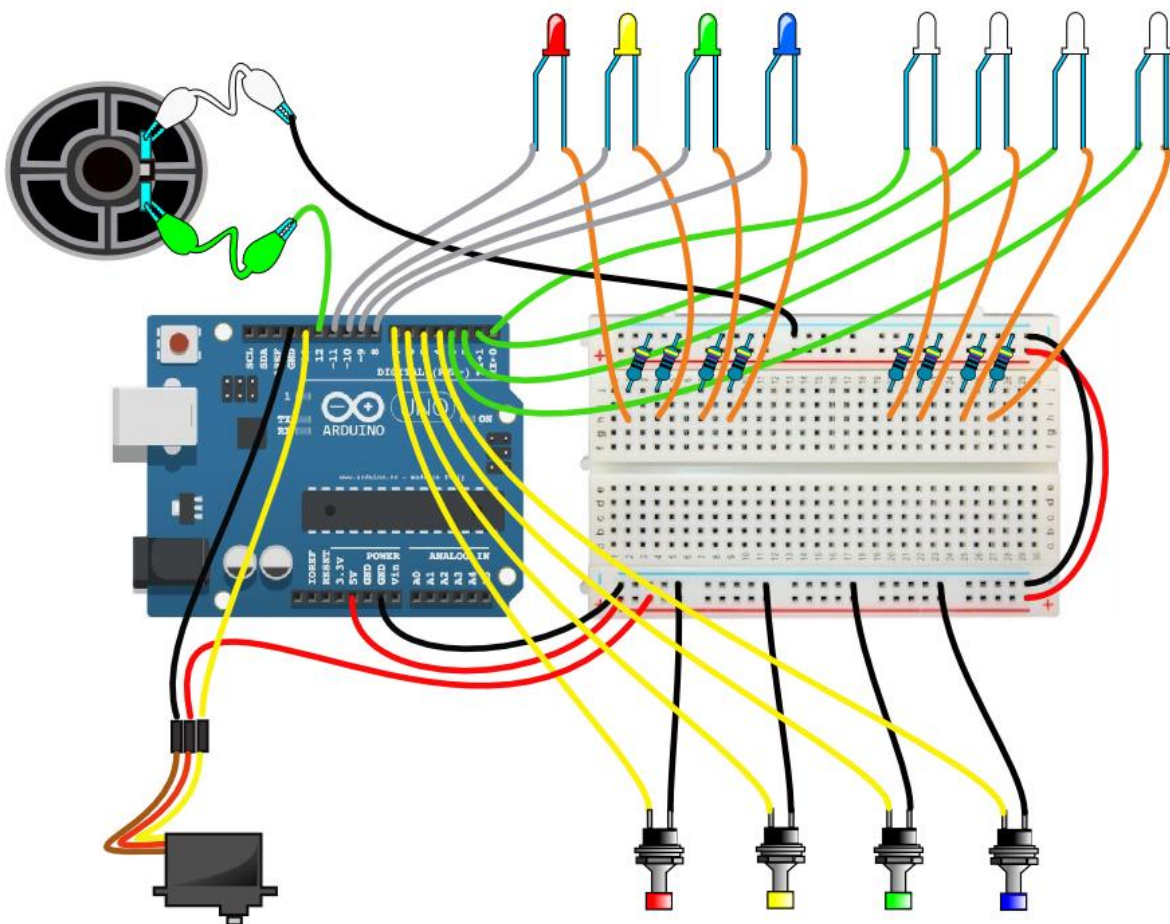
```

5.- Guitar Hero

Consiste en una versión del clásico juego de memoria Simon pero con forma de guitarra eléctrica. La guitarra llevaba cuatro pulsadores n.a. que corresponden a 4 colores y sus correspondientes notas. El “ampli” aloja todo el circuito electrónico, y lleva en su parte externa cuatro LEDs de colores (azul, verde, naranja y rojo), el altavoz, y un marcador de nivel que consiste en un servomotor que mueve una aguja indicadora y cuatro LEDs blancos para reforzar el nivel alcanzado. Cada vez que se consiga reproducir la secuencia que se escucha, la aguja gira un poco y la siguiente secuencia aumenta en uno el número de tonos. Al final, según el nivel alcanzado, sonará una melodía u otra.



El montaje y su correspondiente sketch podrían ser los siguientes:




```

/*Guitar Hero
 IES Laguna de Tollón (El Cuervo de Sevilla)
 Daniel Gallardo García */

#include <Servo.h>

Servo aguja;

int botonAzul, botonVerde, botonNaranja, botonRojo; //almacenará la lectura de los pulsadores, que estarán en
los pines 4, 5, 6 y 7
int led[] = {8, 9, 10, 11};
int ledNivel[] = {0, 1, 2, 3};
int secuenciaLuz[20]; //almacenará la secuencia para luego comprobar si es la correcta
int secuenciaBoton[20]; //almacenará la secuencia de los botones
int pinBoton; //almacenará qué botón se ha pulsado recientemente
int aleatorio; //número aleatorio del 0 al 3
int tiempoLuz; //tiempo de espera entre cada pulso de luz de una secuencia
boolean resultado; //si se hace correctamente la secuencia, valdrá 1. En caso contrario 0
float do2=130.813, do_2=138.591, re2=146.832, re_2=155.563, mi2=164.814, fa2=174.614,
fa_2=184.997, sol2=195.998, sol_2=207.652, la2=220, la_2=233.082, si2=246.942,
do3=261.626, do_3=277.183, re3=293.665, re_3=311.127, mi3=329.628, fa3=349.228,
fa_3=369.994, sol3=391.995, sol_3=415.305, la3=440, la_3=466.164, si3=493.883,
do4=523.251, do_4=554.365, re4=587.330, re_4=622.254, mi4=659.255, fa4=698.456,
fa_4=739.989, sol4=783.991, sol_4=830.609, la4=880, la_4=932.328, si4=987.767,pausa=0;
float tono[] = {do3, mi3, sol3, do4};
int ritmoCorrecto[] = {1, 1, 1, 4};
float melodiaFallo[] = {do3, re_3, si2, re3, la_2, do_3, la2, do3};
int ritmoFallo[] = {1, 8, 1, 8, 1, 8, 1, 32};
float melodiaMuerto[] = {la3, pausa, la3, pausa, la3, la3, do4, si3, si3, pausa, la3, la3, pausa, sol_3, la3,
pausa, do4, pausa, do4, pausa, do4, do4, sol4, fa_4, fa_4, pausa, mi4, mi4, pausa, re_4, mi4};
float melodiaMuertoAcorde[] = {la2, pausa, la2, pausa, la2, la2, pausa, pausa, pausa, pausa, pausa,
pausa, pausa, pausa, pausa, do3, pausa, do3, pausa, do3, do3, pausa, pausa, pausa, pausa,
pausa, pausa, pausa};
int ritmoMuerto[] = {4, 2, 4, 2, 1, 4, 4, 2, 2, 1, 2, 2, 1, 2, 4, 4, /*segunda escala*/ 4, 2, 4, 2, 1, 4, 4, 2,
2, 1, 2, 2, 1, 2, 4};
float melodiaBatman[]={do3, re_3, sol3, sol_3, sol3, pausa, sol_3, sol3, pausa, do3, re_3, fa3, sol3, pausa,
la3, pausa, la_3, fa3, fa3};
int ritmoBatman[] = {2, 2, 2, 10, 8, 4, 6, 2, /*segunda escala*/ 8, 2, 2, 2, 8, 2, 4, 2, 4, 2, 4};
float melodiaDracula[] = {mi4, re4, mi4, re4, do4, si3, la3, sol_3, la3, la3};
int ritmoDracula[] = {1, 1, 8, 1, 1, 1, 1, 8, 1, 8};
float melodiaAlienigena[] = {re4, mi4, do4, do3, sol3, pausa};
int ritmoAlienigena[] = {1, 1, 1, 1, 2, 1};
int velocidad; //me servirá para incrementar la velocidad en la melodía Alienígena
int par; //me servirá para repetir dos veces el tamaño de cada secuencia
int angulo; //es la posición del servo

void setup(){
  aguja.attach(13);
  for(int i=4; i<8; i++) pinMode(i, INPUT_PULLUP);
  for(int i=8; i<13; i++) pinMode(i, OUTPUT);
  for(int i=0; i<4; i++) pinMode(i, OUTPUT);
  randomSeed(analogRead(A0));
}

void loop(){
  angulo = 180;
  aguja.write(angulo);
  botonAzul = digitalRead(4);
  botonVerde = digitalRead(5);
  botonNaranja = digitalRead(6);
  botonRojo = digitalRead(7);
  tiempoLuz = 1050;
  for(int i=0; i<4; i++) digitalWrite(ledNivel[i], LOW);
  //luces de espera. Estarán hasta que se pulse algún botón
  do {
    for(int c=8; c<12; c++) {
      for(int d=8; d<12; d++) digitalWrite(d, LOW);
      digitalWrite(c, HIGH);
      delay(50);
      digitalWrite(c, LOW);
    }
    botonAzul = digitalRead(4);
    botonVerde = digitalRead(5);
    botonNaranja = digitalRead(6);
    botonRojo = digitalRead(7);
  } while(botonAzul == HIGH && botonVerde == HIGH && botonNaranja == HIGH && botonRojo == HIGH);
  //generación de las secuencias aleatorias
  delay(2000);
  resultado = 1;
}

```

```

tiempoLuz = 1050;
par = 0;
for(int n=3; n<20; n++) { //n será el número de luces en cada secuencia
    tiempoLuz = tiempoLuz - 10*n;
    par++;
    if(par % 2 == 0) n--; //solo aumentara el valor de n cada dos ciclos
    for(int i=0; i<n; i++) { //i será la i-ésima luz de la secuencia de tamaño n
        aleatorio = random(0,4);
        secuenciaLuz[i] = aleatorio;
        digitalWrite(led[aleatorio], HIGH);
        tone(12, tono[aleatorio]);
        delay(tiempoLuz);
        digitalWrite(led[aleatorio], LOW);
        noTone(12);
        delay(tiempoLuz/2);
    }
    //comparo la secuencia de botones con la guardada en secuenciaLuz
    for(int j=0; j<n; j++) {
        do {
            botonAzul = digitalRead(4);
            botonVerde = digitalRead(5);
            botonNaranja = digitalRead(6);
            botonRojo = digitalRead(7);
        } while(botonAzul == HIGH && botonVerde == HIGH && botonNaranja == HIGH && botonRojo == HIGH);
        delay(50);
        if(botonAzul == LOW) {
            secuenciaBoton[j] = 0;
            pinBoton = 4;
            digitalWrite(led[0], HIGH);
        }
        else if(botonVerde == LOW) {
            secuenciaBoton[j] = 1;
            pinBoton = 5;
            digitalWrite(led[1], HIGH);
        }
        else if(botonNaranja == LOW) {
            secuenciaBoton[j] = 2;
            pinBoton = 6;
            digitalWrite(led[2], HIGH);
        }
        else {
            secuenciaBoton[j] = 3;
            pinBoton = 7;
            digitalWrite(led[3], HIGH);
        }
        tone(12, tono[secuenciaBoton[j]]);
        do {
            delay(100);
        } while (digitalRead(pinBoton) == LOW); //espero hasta que se quite el dedo del pulsador
        delay(100);
        noTone(12); //apago la música y la luz
        for(int i=8; i<12; i++) digitalWrite(i, LOW);
        if(secuenciaBoton[j] != secuenciaLuz[j]) {
            resultado = 0;
            delay(500);
            for(int i=0; i<sizeof(ritmoFallo)/2; i++) { //música de fallo
                tone(12, melodiaFallo[i], ritmoFallo[i]*40);
                delay(ritmoFallo[i]*40 + 2);
            }
            delay(500);
            break;
        }
        if(resultado == 0) break;
    }
    if(resultado == 0) break;
    delay(500);
    if(angulo > 0) angulo = angulo - 15;
    aguja.write(angulo);
    if(par > 0) digitalWrite(ledNivel[0], HIGH);
    if(par > 2) digitalWrite(ledNivel[1], HIGH);
    if(par > 5) digitalWrite(ledNivel[2], HIGH);
    if(par > 8) digitalWrite(ledNivel[3], HIGH);
    for(int i=0; i<4; i++) { //música de correcto
        tone(12,tono[i], ritmoCorrecto[i]*50);
        delay(ritmoCorrecto[i]*50);
    }
    delay(2000);
}
if(par < 4) { //nivel Muerto. par va uno por encima para los leds
    for(int i=0; i<sizeof(ritmoMuerto)/2; i++) { //música de Muerto
        tone(12, melodiaMuertoAcorde[i], 30);
    }
}

```

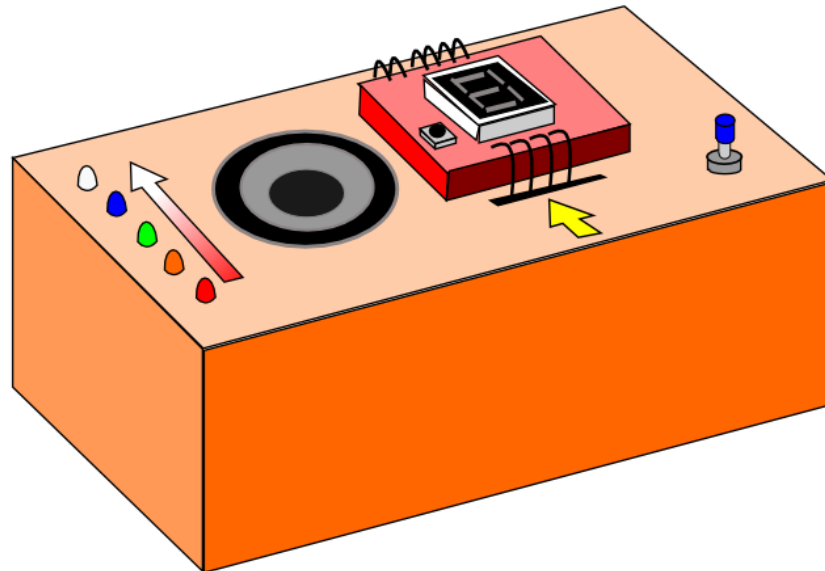
```

    delay(30);
    tone(12, melodiaMuerto[i], ritmoMuerto[i]*100);
    int iLed = (i+4) % 4;
    digitalWrite(led[iLed], HIGH);
    delay(ritmoMuerto[i]*100 + 2);
    digitalWrite(led[iLed], LOW);
}
delay(1000);
}
else if(par < 7) { //nivel Batman
for(int i=0; i<sizeof(ritmoBatman)/2; i++) { //música de Batman
tone(12, melodiaBatman[i], ritmoBatman[i]*50);
int iLed = (i+4) % 4;
digitalWrite(led[iLed], HIGH);
delay(ritmoBatman[i]*50 + 2);
digitalWrite(led[iLed], LOW);
}
delay(1000);
}
else if(par < 10) { //nivel Drácula
for(int i=0; i<sizeof(ritmoDracula)/2; i++) { //música de Drácula
tone(12, melodiaDracula[i], ritmoDracula[i]*100);
int iLed = (i+4) % 4;
digitalWrite(led[iLed], HIGH);
delay(ritmoDracula[i]*100 + 10);
digitalWrite(led[iLed], LOW);
}
delay(1000);
}
else if(par > 9) { //nivel Alienígena
velocidad = 1000;
for(int j=0; j<20; j++) {
for(int i=0; i<sizeof(ritmoAlienigena)/2; i++) { //música de Alienigena
tone(12, melodiaAlienigena[i], ritmoAlienigena[i]*velocidad);
int iLed = (i+4) % 4;
digitalWrite(led[iLed], HIGH);
delay(ritmoAlienigena[i]*velocidad);
digitalWrite(led[iLed], LOW);
}
velocidad = velocidad/1.5;
if(velocidad < 80) velocidad = 80;
}
delay(1000);
}
}
}

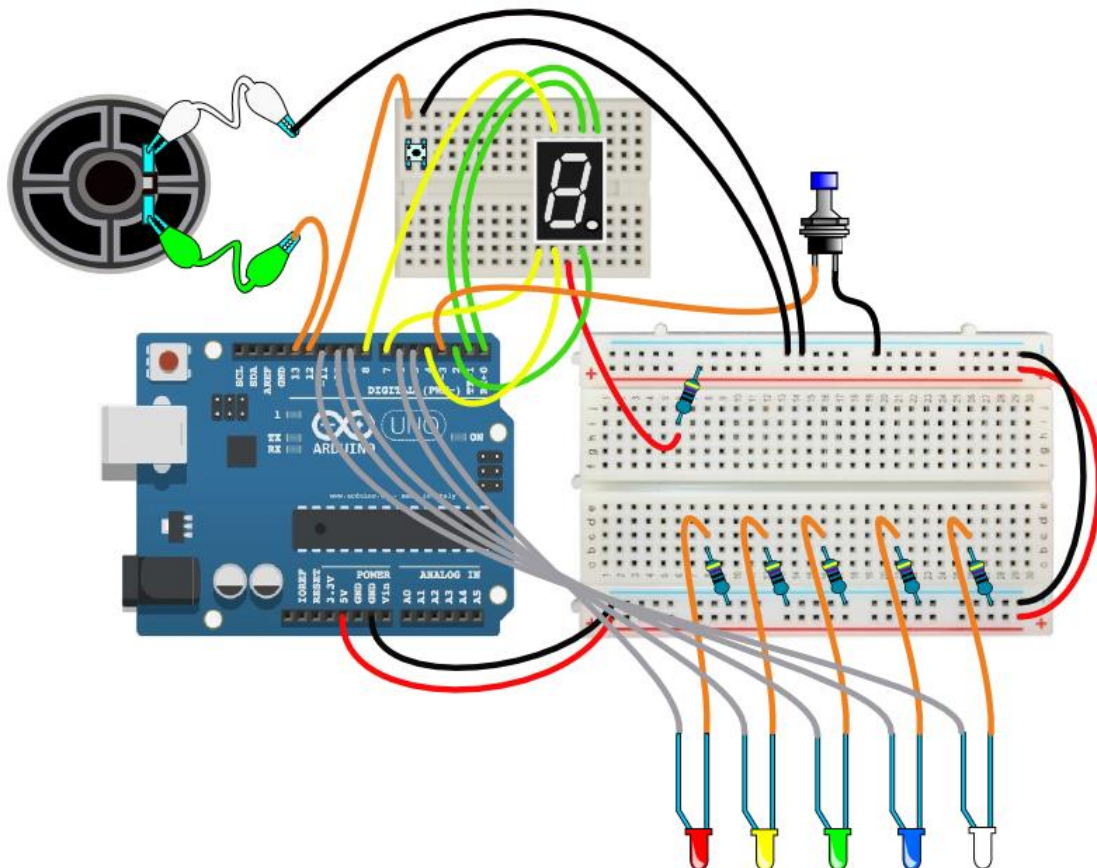
```

6.- Habilidad Digital

Es un juego de coordinación entre la vista y el dedo (de ahí “digital”). Una luz se mueve por los seis segmentos exteriores de un display 7-segmentos, y debemos apretar el pulsador justamente cuando pase por el segmento inferior. En caso de acertar, la velocidad aumentará en la siguiente partida; y en caso de fracasar, la velocidad disminuirá. Dispone de un botón de reset, para volver a la velocidad inicial.



El montaje y su correspondiente sketch podrían ser los siguientes:



```

/*Habilidad Digital
 IES Laguna de Tollón (El Cuervo de Sevilla)
 Daniel Gallardo García */

int pinSegmentos[] = {0, 1, 2, 4, 7, 8}; //(a,b,c,d,e,f) no necesito ni g ni dp
int pinLeds[] = {5, 6, 9, 10, 11}; //son pines para salidas analógicas
int pinAltavoz = 13;
int pinPulsador = 3;
int pinReset = 12;
//es importante respetar el orden de los segmentos del display
int estadoPulsador; //almacena el estado del pulsador
long tiempoComienzo; //tiempo de inicio de una vuelta
long tiempoIntervalo; //ayudará a saber cuando cambia el led
int numeroPin; //ayudará a saber el pin del led que debe estar encendido
int nivel = 1; //será el nivel por el que va el jugador
int velocidad = 100; //milisegundos que cada segmento estará encendido
float do2=130.813, do_2=138.591, re2=146.832, re_2=155.563, mi2=164.814, fa2=174.614,
fa_2=184.997, sol2=195.998, sol_2=207.652, la2=220, la_2=233.082, si2=246.942,
do3=261.626, do_3=277.183, re3=293.665, re_3=311.127, mi3=329.628, fa3=349.228,
fa_3=369.994, sol3=391.995, sol_3=415.305, la3=440, la_3=466.164, si3=493.883,
do4=523.251, do_4=554.365, re4=587.330, re_4=622.254, mi4=659.255, fa4=698.456,
fa_4=739.989, sol4=783.991, sol_4=830.609, la4=880, la_4=932.328, si4=987.767;
float melodiaError[]={do3, re_3, si2, re3, la_2, do_3, la2, do3};
int ritmoError[]={1,4,1,4,1,4,1,10};
float melodiaAcierto[]={sol2, do3, mi3, sol3};
int ritmoAcierto[]={2,2,2,8};

void setup() {
  for(int i=0; i<8; i++) pinMode(pinSegmentos[i], OUTPUT);
  for(int i=0; i<5; i++) pinMode(pinLeds[i], OUTPUT);
  pinMode(pinAltavoz, OUTPUT);
  pinMode(pinPulsador, INPUT_PULLUP);
  pinMode(pinReset, INPUT_PULLUP);
}

void loop() {
  for(int i=0; i<6; i++) digitalWrite(pinSegmentos[i], HIGH);
  //pongo todos los leds apagados
  tiempoComienzo = millis(); //es el origen de tiempo para un ciclo completo
  do {
    if(!digitalRead(pinReset)) resetea();
    tiempoIntervalo = millis() - tiempoComienzo; //calculo cuanto tiempo llevo
    //desde que comenzó el ciclo
    numeroPin = tiempoIntervalo / velocidad;
    //darán números enteros: 0, 1, 2, ... que aumentarán cada vez
    //que se sobrepasa el tiempo "velocidad" una, dos, ...n veces.
    //Es decir: calcula a qué pin le toca estar encendido
    digitalWrite(pinSegmentos[numeroPin-1], HIGH);
    digitalWrite(pinSegmentos[numeroPin], LOW);
    //enciencdo el pin correspondiente y apago el anterior
    estadoPulsador = digitalRead(pinPulsador); //leo el estado del pulsador
    if(numeroPin == 6) { //es el lado en el que he terminado el ciclo
      digitalWrite(pinSegmentos[6], HIGH);
      tiempoComienzo = millis();
    }
  } while (estadoPulsador == HIGH); //cuando apriete el pulsador, se detendrá el "do"
  if (numeroPin == 3) acierto(); //si pulso estando encendido el segmento "d"
  else fallo();
}

void acierto() {
  nivel++;
  for(int i=0; i<sizeof(ritmoAcierto)/2; i++) {
    tone(pinAltavoz, melodiaAcierto[i], ritmoError[i]*50);
    delay(ritmoError[i]*50 + 1);
  }
  for(int i=0; i<5; i++) digitalWrite(pinLeds[i], LOW); //apago los LEDs
  for(int j=0; j<3; j++) { //y harán efecto ascendente
    for(int i=0; i<5; i++) {
      digitalWrite(pinLeds[i], HIGH);
      delay(100);
    }
    for(int i=0; i<5; i++) {
      digitalWrite(pinLeds[i], LOW);
      delay(100);
    }
  }
  velocidad = velocidad - 10;
  luzNivel(nivel);
}

```

```

void fallo() {
  nivel--;
  for(int i=0; i<sizeof(ritmoError)/2; i++) {
    tone(pinAltavoz, melodiaError[i],ritmoError[i]*50);
    delay(ritmoError[i]*50 + 1);
  }
  for(int i=0; i<5; i++) digitalWrite(pinLeds[i], LOW); //apago los LEDs
  for(int j=0; j<3; j++) { //y harán efecto descendente
    for(int i=4; i>=0; i--) {
      digitalWrite(pinLeds[i], HIGH);
      delay(100);
    }
    for(int i=4; i>=0; i--) {
      digitalWrite(pinLeds[i], LOW);
      delay(100);
    }
  }
  velocidad = velocidad + 10;
  luzNivel(nivel);
}

void luzNivel(int n) {
  digitalWrite(pinLeds[1], HIGH); //el led naranja será el punto de partida, y siempre estará encendido.
  if(n < -1) digitalWrite(pinLeds[0], HIGH);
  else if(n == -1) analogWrite(pinLeds[0], 60);
  else if(n == 1) analogWrite(pinLeds[2], 60);
  else if(n == 2) digitalWrite(pinLeds[2], HIGH);
  else if(n == 3) {
    digitalWrite(pinLeds[2], HIGH);
    analogWrite(pinLeds[3], 60);
  }
  else if(n == 4) {
    digitalWrite(pinLeds[2], HIGH);
    digitalWrite(pinLeds[3], HIGH);
  }
  else if(n == 5) {
    digitalWrite(pinLeds[2], HIGH);
    digitalWrite(pinLeds[3], HIGH);
    analogWrite(pinLeds[4], 60);
  }
  else for(int i=2; i<5; i++) digitalWrite(pinLeds[i], HIGH);
}

void resetea() {
  asm volatile("jmp 0");
}

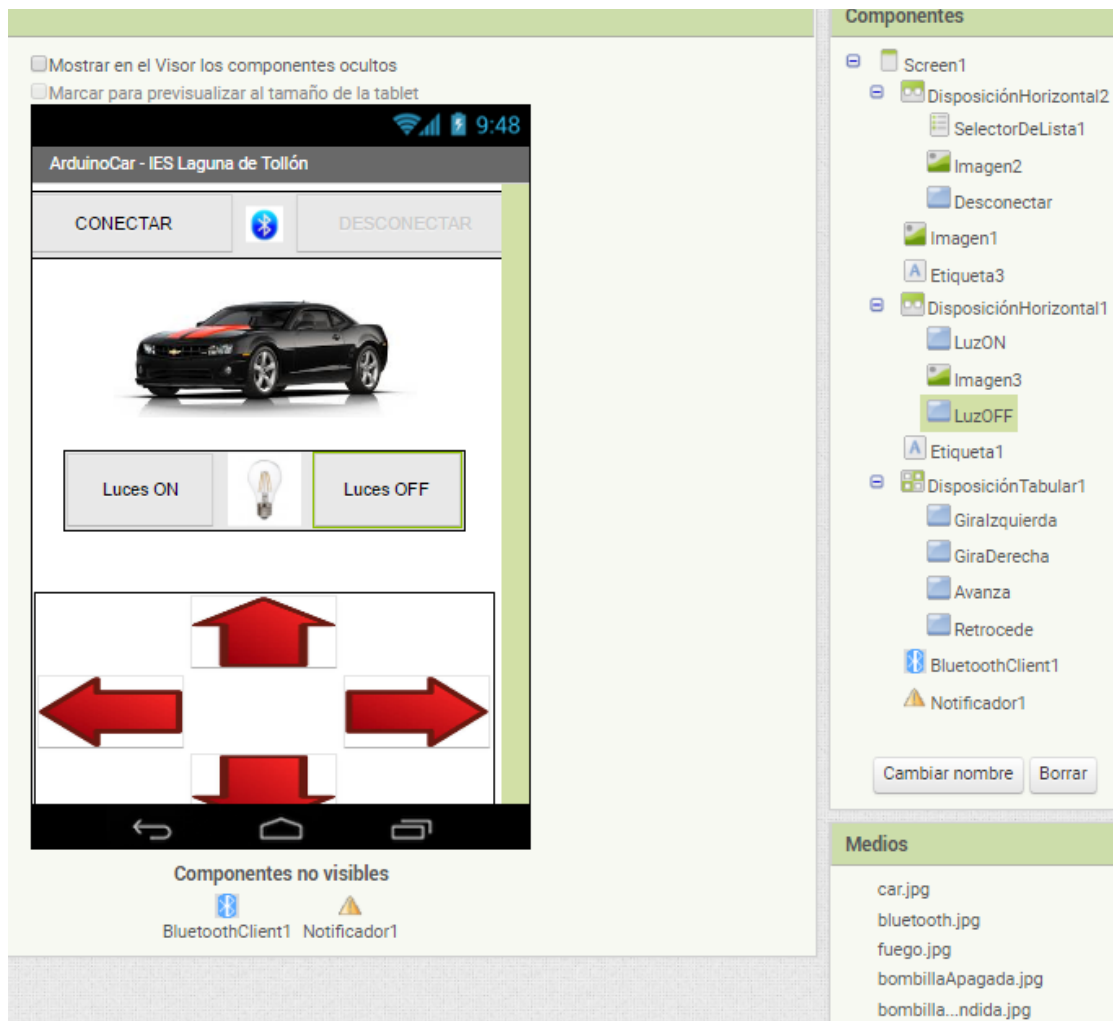
```

7.- ArduinoCar

Se trata de un coche gobernado desde un dispositivo móvil. Para ello necesitamos un módulo Bluetooth para nuestra Arduino, así como de instalar en nuestro móvil alguna aplicación para su control (la diseñaremos con AppInventor2). Además de las operaciones básicas de movimiento, tendrá la opción de encendido y apagado de las luces delanteras:



La aplicación para el móvil puede ser la siguiente:



Mostrar en el Visor los componentes ocultos
Marcar para previsualizar al tamaño de la tablet

ArduinoCar - IES Laguna de Tollón

CONECTAR [Bluetooth Icon] DESCONECTAR

[Car Image]

Luces ON [Lightbulb Icon] Luces OFF

[Left Arrow] [Up Arrow] [Right Arrow] [Down Arrow]

Componentes no visibles
BluetoothClient1 Notificador1

Componentes

- Screen1
 - DisposiciónHorizontal2
 - SelectorDeLista1
 - Imagen2
 - Desconectar
 - Imagen1
 - Etiqueta3
 - DisposiciónHorizontal1
 - LuzON
 - Imagen3
 - LuzOFF
 - Etiqueta1
 - DisposiciónTabular1
 - GirarIzquierda
 - GiraDerecha
 - Avanza
 - Retrocede
 - BluetoothClient1
 - Notificador1

Cambiar nombre Borrar

Medios

- car.jpg
- bluetooth.jpg
- fuego.jpg
- bombillaApagada.jpg
- bombilla...ndida.jpg

```
when SelectorDeLista1 .BeforePicking
do set SelectorDeLista1 . Elements to BluetoothClient1 . AddressesAndNames
```

```
when SelectorDeLista1 .AfterPicking
do if call BluetoothClient1 .Connect
    address SelectorDeLista1 . Selection
then set SelectorDeLista1 . Enabled to false
set Desconectar . Enabled to true
set LuzON . Enabled to true
set SelectorDeLista1 . TextColor to grey
set Desconectar . TextColor to red
call Notificador1 . ShowMessageDialog
    message " CONECTADO "
    title " Conexión "
    buttonText " Aceptar "
```

```
when Desconectar .Click
do set SelectorDeLista1 . Enabled to true
set Desconectar . Enabled to false
set LuzON . Enabled to false
set SelectorDeLista1 . TextColor to green
set Desconectar . TextColor to grey
call Notificador1 . ShowMessageDialog
    message " DESCONECTADO "
    title " Desconexión "
    buttonText " Aceptar "
```

```
when LuzON .Click
do set Imagen3 . Picture to " bombillaEncendida.jpg "
call BluetoothClient1 . SendText
    text " 1 "
```

```
when LuzOFF .Click
do set Imagen3 . Picture to " bombillaApagada.jpg "
call BluetoothClient1 . SendText
    text " 0 "
```

```
when Avanza .TouchDown
do call BluetoothClient1 . SendText
    text " 8 "
```

```
when Avanza .TouchUp
do call BluetoothClient1 . SendText
    text " 5 "
```

```
when Retrocede .TouchDown
do call BluetoothClient1 . SendText
    text " 2 "
```

```
when Retrocede .TouchUp
do call BluetoothClient1 . SendText
    text " 5 "
```

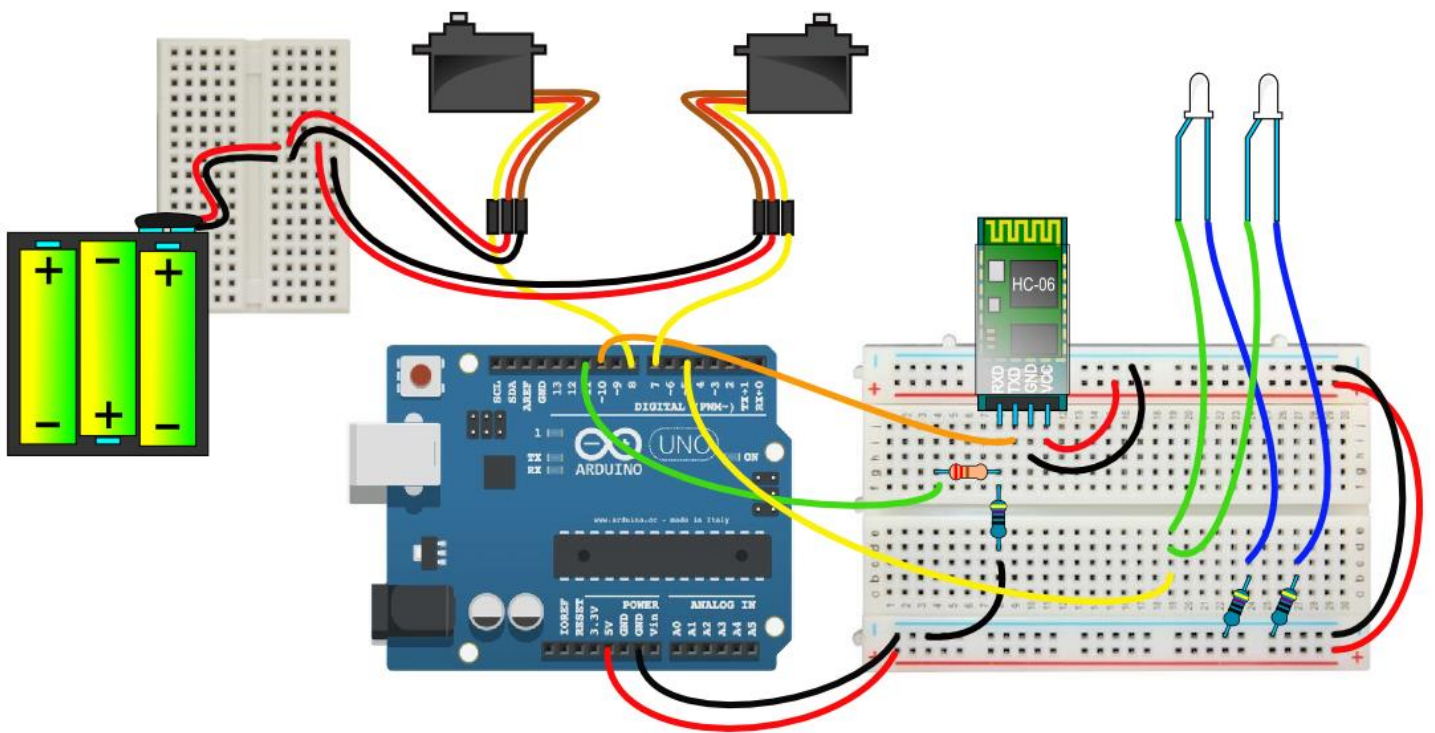
```
when Giralzquierda .TouchDown
do call BluetoothClient1 . SendText
    text " 4 "
```

```
when Giralzquierda .TouchUp
do call BluetoothClient1 . SendText
    text " 5 "
```

```
when GiraDerecha .TouchDown
do call BluetoothClient1 . SendText
    text " 6 "
```

```
when GiraDerecha .TouchUp
do call BluetoothClient1 . SendText
    text " 5 "
```


El montaje y su correspondiente sketch podrían ser los siguientes:



```
/*Arduino Car
IES Laguna de Tollón (El Cuervo de Sevilla)
Daniel Gallardo García */

#include <SoftwareSerial.h>
#include <Servo.h>

SoftwareSerial BT(10, 11);
Servo ruedaDerecha;
Servo ruedaIzquierda;

int val;
int pinLuz = 5; //a este pin único pin le conecto los dos LEDs

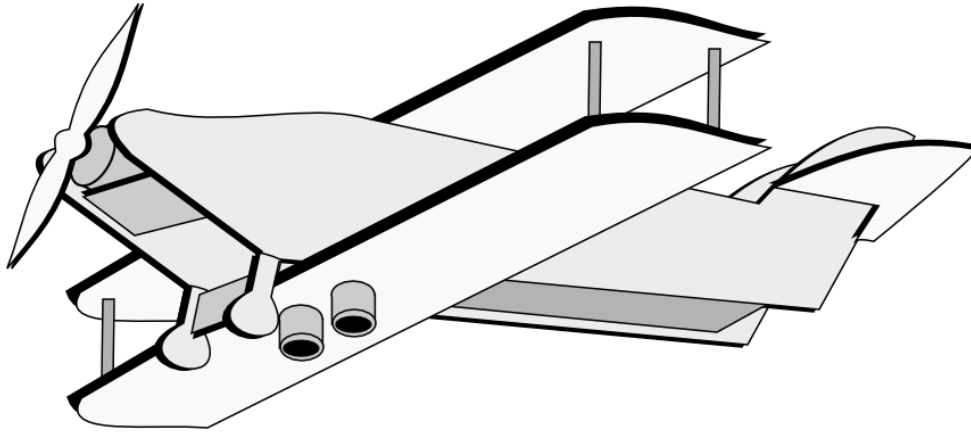
void setup() {
  BT.begin(9600);
  pinMode(pinLuz, OUTPUT);
  ruedaDerecha.attach(7);
  ruedaIzquierda.attach(8);
}

void loop() {
  if(BT.available()) val = BT.read();
  //val será el código de la señal que le llega por el puerto serie del Bluetooth,...
  //... es decir: nuestro móvil debe mandar dichas señales: 0,1,2,4,5,6 y 8 (basado en el teclado
  numérico)
  if(val == '0') digitalWrite(pinLuz, LOW);
  if(val == '1') digitalWrite(pinLuz, HIGH);
  if(val == '2') { //retrocede
    for(int i=90; i<181; i++) {
      ruedaDerecha.write(i);
      ruedaDerecha.write(180 - i);
      delay(15);
    }
  }
  if(val == '4') { //gira a la izquierda
    for(int i=90; i<181; i++) {
      ruedaDerecha.write(180 - i);
      delay(15);
    }
  }
}
```

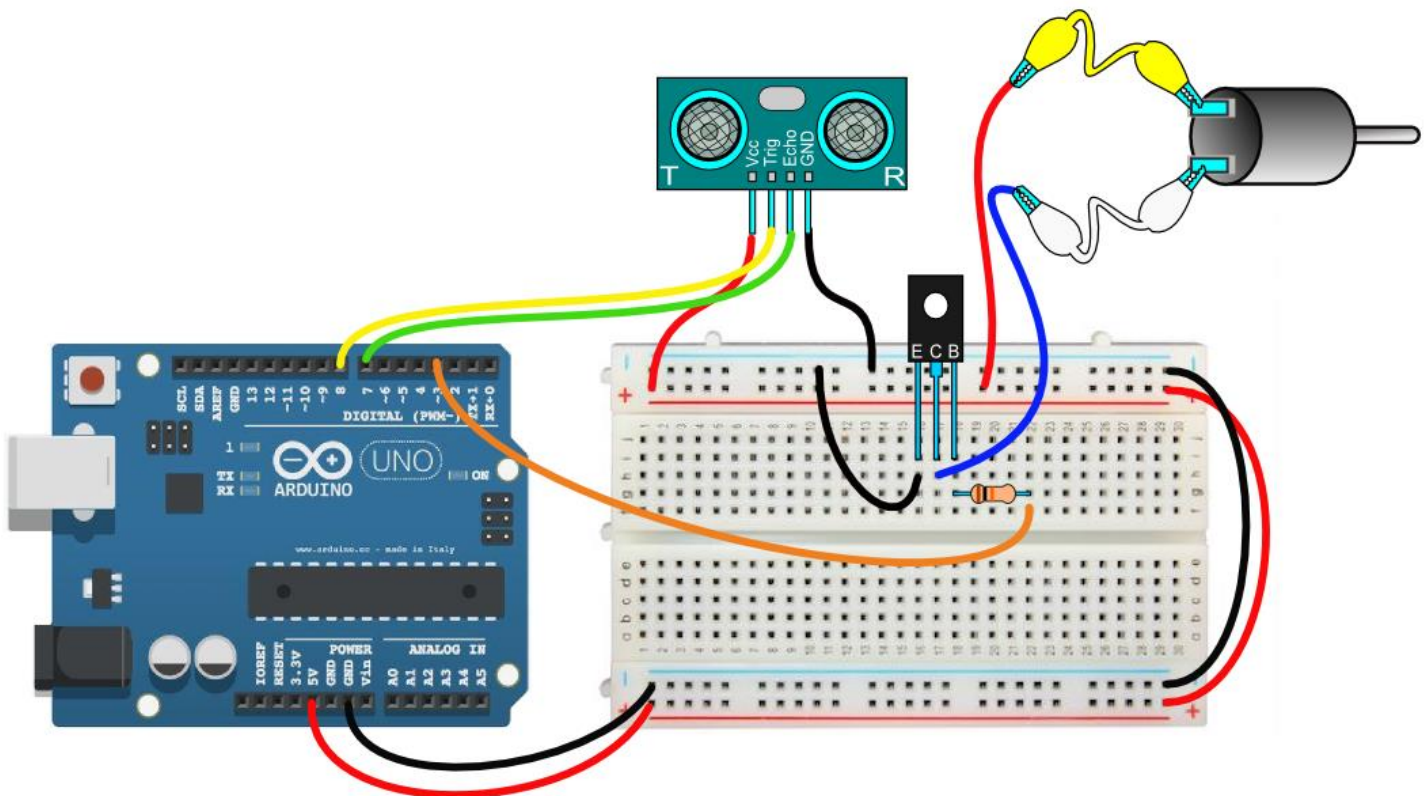
```
if(val == '5') { //se para
  ruedaDerecha.write(90);
  ruedaIzquierda.write(90);
}
if(val == '6') { //gira a la derecha
  for(int i=90; i<181; i++) {
    ruedaIzquierda.write(i);
    delay(15);
  }
}
if(val == '8') { //avanza
  for(int i=90; i<181; i++) {
    ruedaDerecha.write(180 - i);
    ruedaDerecha.write(i);
    delay(15);
  }
}
}
```

8.- ArduinoPlane

Es un avión que dispone de un sensor ultrasónico para detectar cuándo está a una cierta altura respecto a la superficie de apoyo (suelo, mesa...), a partir de la cual comenzará a girar su hélice.



El montaje y su correspondiente sketch podrían ser los siguientes:



```

/* Arduino Plane
   IES Laguna de Tollón (El Cuervo de Sevilla)
   Daniel Gallardo García */

int disparo = 8, eco = 7;    //pines para el sensor de distancia
int pinMotor = 3;
long tiempoInicial, tiempoFinal, duracion; /*variables que utilizaremos para
      obtener el tiempo hasta el rebote */
int cm, senalEco;    //variables para los centímetros y para detectar el rebote

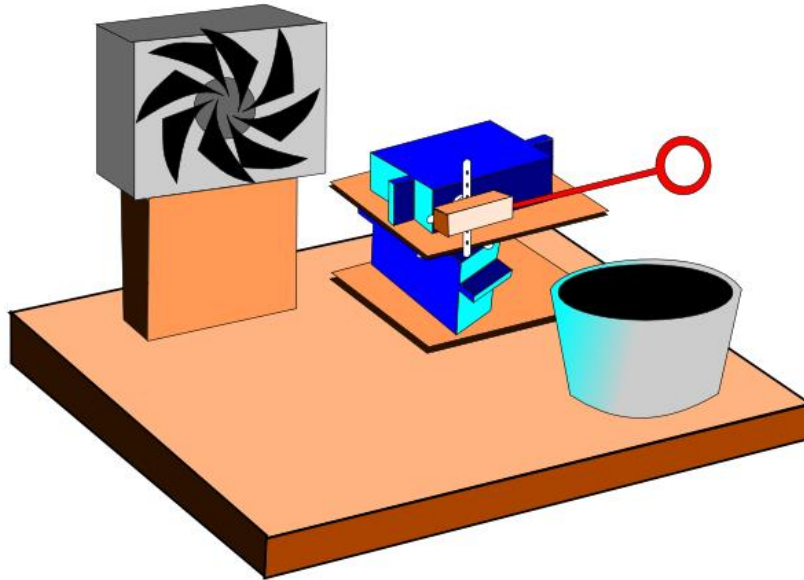
void setup(){
  Serial.begin(9600);
  pinMode(disparo, OUTPUT);
  pinMode(eco, INPUT);
  pinMode(pinMotor, OUTPUT);
}

void loop(){
  //Hago un disparo: lanzo un pulso de 5 us de duración
  digitalWrite(disparo,LOW);
  delayMicroseconds(2);
  digitalWrite(disparo,HIGH);
  delayMicroseconds(5);
  tiempoInicial = micros();    //pongo el "cronómetro" a cero
  digitalWrite(disparo, LOW);
  //Detectaré el tiempo que tarda en llegar el rebote
  senalEco = digitalRead(eco);
  while(senalEco == LOW){
    senalEco = digitalRead(eco); //para saber cuándo salgo del valor LOW
    if(tiempoInicial + 2000 < millis()) break;
  }
  while(senalEco == HIGH){
    senalEco = digitalRead(eco);
    tiempoFinal = micros();
  }
  //tiempoFinal es el instante en el que termina la señal de rebote en HIGH
  //Calculo la duración del recorrido sensor-obstáculo-sensor
  duracion = tiempoFinal - tiempoInicial;
  cm = int(duracion/58); /*el sonido se desplaza a 340m/s o 29ms/cm
      y como tiene que recorrer dos veces la distancia hasta el objeto,
      deberemos dividir entre 2*29 los microsegundos transcurridos */
  if(cm > 25) {
    digitalWrite(pinMotor, HIGH);
  }
  else {
    digitalWrite(pinMotor, LOW);
  }
}

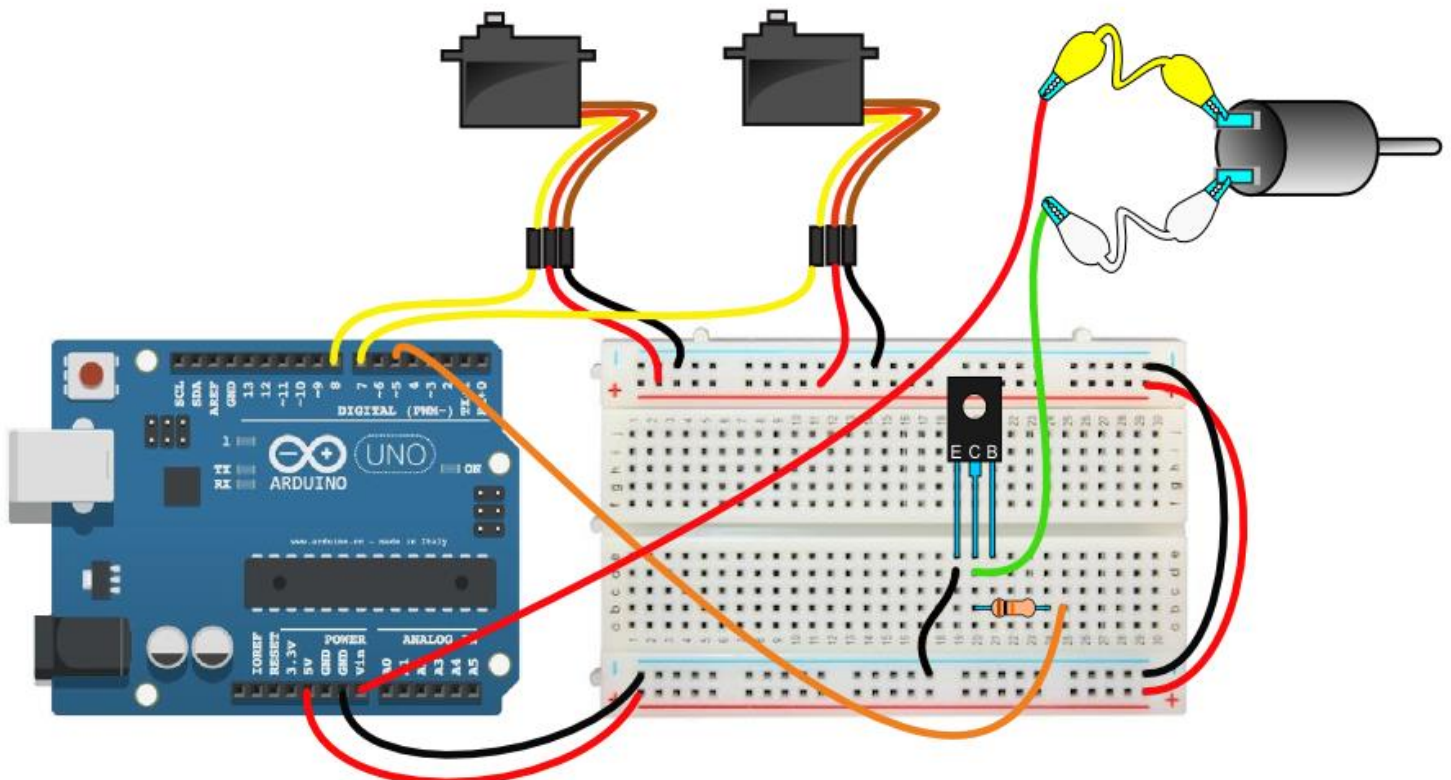
```

9.- Máquina de hacer pompas

Consiste en una máquina de hacer pompas de jabón: un brazo robótico formado por dos servomotores irá mojando el palito rojo en una disolución jabonosa y luego lo pondrá frente a un ventilador para que haga la pompa de jabón, y así indefinidamente:



El montaje y su correspondiente sketch podrían ser los siguientes:



```
/* Máquina de pompas de jabón  
IES Laguna de Tollón (El Cuervo de Sevilla)  
Daniel Gallardo García */
```

```
#include <Servo.h>
```

```
Servo horizontal;  
Servo vertical;
```

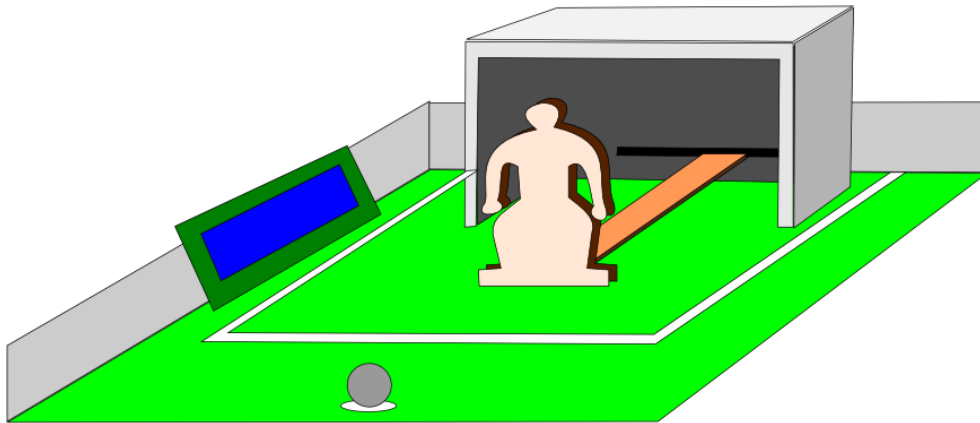
```
int pinMotor = 5;  
int hi = 0;  
int hf = 180;  
int vi = 0;  
int vf = 180;
```

```
void setup() {  
  horizontal.attach(7);  
  vertical.attach(8);  
  pinMode(pinMotor, OUTPUT);  
  horizontal.write(hi);  
  vertical.write(vi);  
}
```

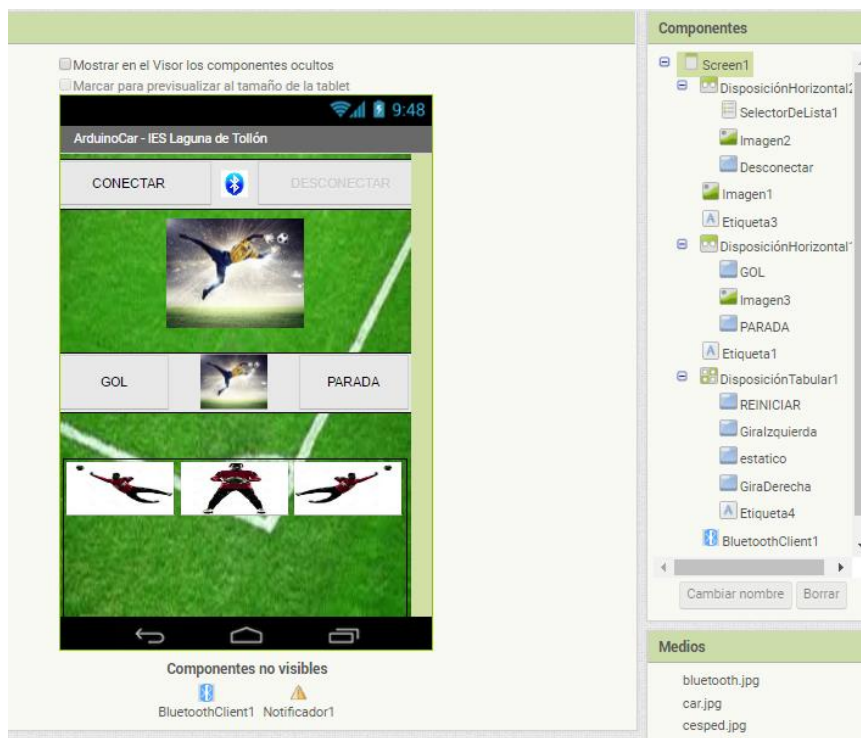
```
void loop() {  
  //movimiento para mojar en jabón:  
  for(int i=hi; i<hf; i++) {  
    horizontal.write(i);  
    delay(20);  
  }  
  for(int i=vi; i<vf; i++) {  
    vertical.write(i);  
    delay(20);  
  }  
  //movimiento para ventilador:  
  for(int i=hf; i>hi; i--) {  
    horizontal.write(i);  
    delay(20);  
  }  
  for(int i=vf; i>vi; i--) {  
    vertical.write(i);  
    delay(20);  
  }  
  //mover ventilador:  
  delay(1000);  
  digitalWrite(pinMotor, HIGH);  
  delay(3000);  
  digitalWrite(pinMotor, LOW);  
  delay(1000);  
}
```

10.- Portero de Fútbol

Consiste en un portero que se podrá mover a través de las órdenes mandadas desde un terminal móvil con bluetooth. Además, dispone de un videomarcador donde se podrá ver el recuento de goles y de paradas hasta el momento:



La aplicación para el móvil será:



```
cuando SelectorDeLista1 . AntesDeSelección
ejecutar poner SelectorDeLista1 . Elementos como BluetoothClient1 . DireccionesYNombres
```

```
cuando SelectorDeLista1 . DespuésDeSelección
ejecutar si
  llamar BluetoothClient1 . Conectar
  dirección SelectorDeLista1 . Selección
entonces
  poner SelectorDeLista1 . Habilitado como falso
  poner Desconectar . Habilitado como cierto
  poner SelectorDeLista1 . ColorDeTexto como [gris]
  poner Desconectar . ColorDeTexto como [rojo]
  llamar Notificador1 . MostrarDiálogoMensaje
  mensaje " CONECTADO "
  título " Conexión "
  textoEnBotón " Aceptar "
```

```
cuando Desconectar . Clic
ejecutar
  poner SelectorDeLista1 . Habilitado como cierto
  poner Desconectar . Habilitado como falso
  poner SelectorDeLista1 . ColorDeTexto como [verde]
  poner Desconectar . ColorDeTexto como [gris]
  llamar Notificador1 . MostrarDiálogoMensaje
  mensaje " DESCONECTADO "
  título " Desconexión "
  textoEnBotón " Aceptar "
```

```
cuando GOL . Presionar
ejecutar
  poner Imagen3 . Foto como " triste.jpg "
  llamar BluetoothClient1 . EnviarTexto
  texto " 1 "
```

```
cuando PARADA . Presionar
ejecutar
  poner Imagen3 . Foto como " contento.jpg "
  llamar BluetoothClient1 . EnviarTexto
  texto " 0 "
```

```
cuando GOL . Soltar
ejecutar
  llamar BluetoothClient1 . EnviarTexto
  texto " 5 "
```

```
cuando PARADA . Soltar
ejecutar
  llamar BluetoothClient1 . EnviarTexto
  texto " 5 "
```

```
cuando estatico . Presionar
ejecutar
  llamar BluetoothClient1 . EnviarTexto
  texto " 5 "
```

```
cuando estatico . Soltar
ejecutar
  llamar BluetoothClient1 . EnviarTexto
  texto " 5 "
```

```
cuando Giralzquierda . Presionar
ejecutar
  llamar BluetoothClient1 . EnviarTexto
  texto " 4 "
```

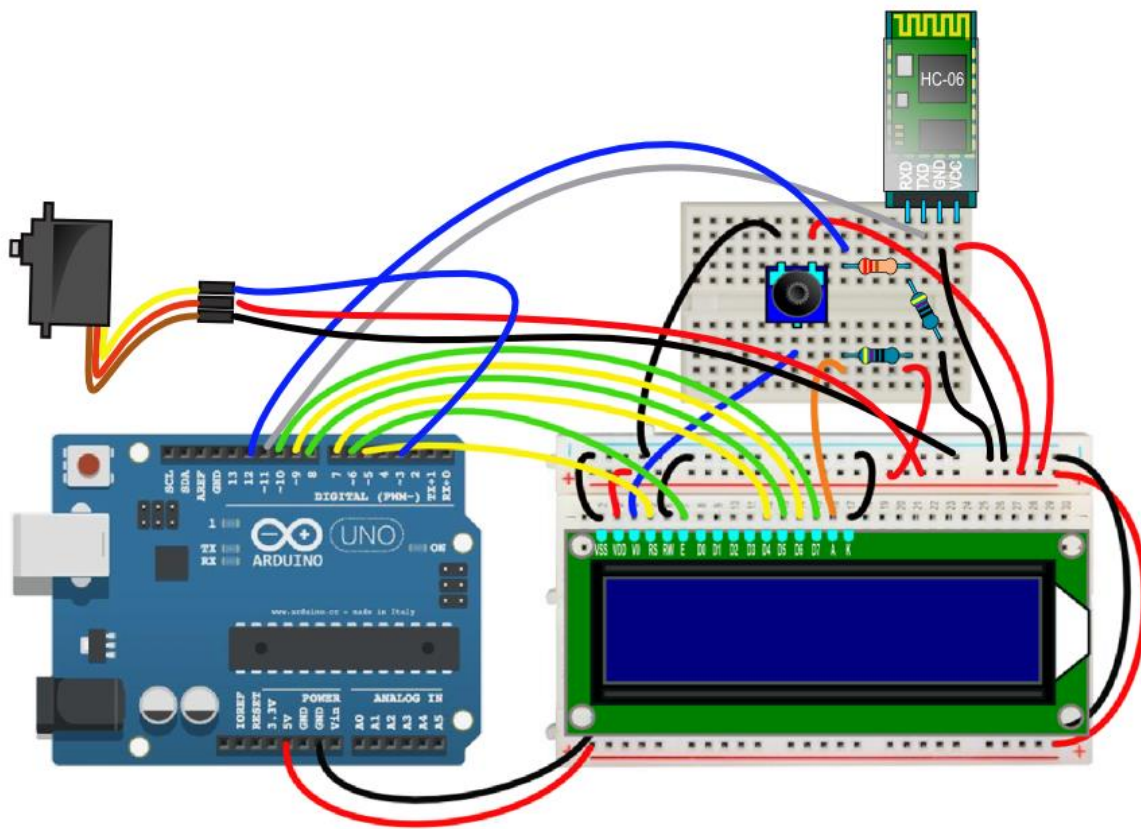
```
cuando Giralzquierda . Soltar
ejecutar
  llamar BluetoothClient1 . EnviarTexto
  texto " 5 "
```

```
cuando GiraDerecha . Presionar
ejecutar
  llamar BluetoothClient1 . EnviarTexto
  texto " 6 "
```

```
cuando GiraDerecha . Soltar
ejecutar
  llamar BluetoothClient1 . EnviarTexto
  texto " 5 "
```

```
cuando REINICIAR . Clic
ejecutar
  poner Imagen3 . Foto como " portero.jpg "
  llamar BluetoothClient1 . EnviarTexto
  texto " 2 "
```


El montaje y su correspondiente sketch podrían ser los siguientes:



```
/*Portero Fútbol
IES Laguna de Tollón (El Cuervo de Sevilla)
Daniel Gallardo García */

#include <SoftwareSerial.h>
#include <Servo.h>
#include <LiquidCrystal.h>

SoftwareSerial BT(11, 12);
Servo portero;
LiquidCrystal lcd(5, 6, 7, 8, 9, 10);

int val;
int goles = 0; //almacenaré los goles
int paradas = 0; //y las paradas

void setup() {
  BT.begin(9600);
  lcd.begin(16, 2);
  pinMode(pinLuz, OUTPUT);
  portero.attach(3);
  lcd.print("¿TE ATREVES A");
  lcd.setCursor(0, 1);
  lcd.print("MARCARME UN GOL?");
  delay(2000);
}

void loop() {
  if(BT.available()) val = BT.read();
  //val será el código de la señal que le llega por el puerto serie del Bluetooth,...
  //... es decir: nuestro móvil debe mandar dichas señales: 0,1,2,4,5,6 y 8 (basado en el teclado
  numérico)
  if(val == '0') {
    parada();
    paradas++;
    marcador(goles, paradas);
  }
}
```

```
    if(val == '1') {
        gol();
        goles++;
        marcador(goles, paradas);
    }
    if(val == '2') resetea();
    if(val == '4') portero.write(135);
    if(val == '5') portero.write(90);
    if(val == '6') portero.write(45);
}

void resetea() {
    asm volatile("jmp 0");
}

void gol() {
    for(int i=0; i<5; i++) {
        lcd.clear();
        lcd.print("  ;;GOOOOOOOOL!! ");
        delay(500);
        lcd.clear();
    }
}

void parada() {
    for(int i=0; i<5; i++) {
        lcd.clear();
        lcd.setCursor(0,1);
        lcd.print("  ;;PARADOOOON!! ");
        delay(500);
        lcd.clear();
    }
}

void marcador(int g, int p) {
    lcd.clear();
    lcd.print("GOLES:  ");
    lcd.print(g);
    lcd.setCursor(0, 1);
    lcd.print("PARADAS: ");
    lcd.print(p);
}
```

II Feria de la Tecnología y la Ingeniería – Málaga 2016 Programa CTC Genuino-Verkstad



Roboteando V – IES Fernando Savater 2016

