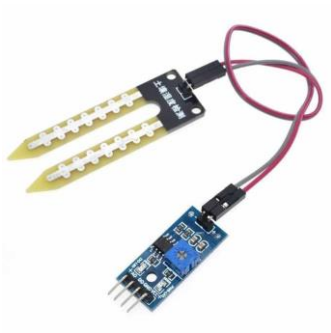
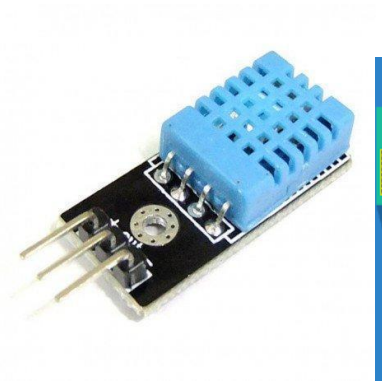
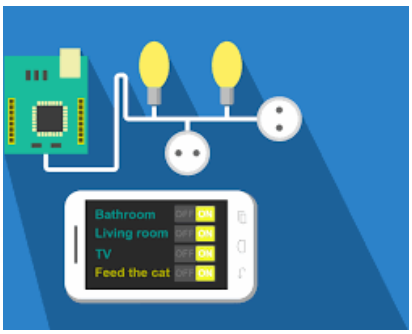
# PROYECTO DEL PROGRAMADOR DE RIEGOS PARA AGRICULTURA Y JARDINES. PROYECTO INNICIA








I.E.S. Mediterráneo

# ÍNDICE


1. **FASE DE ANÁLISIS**
    a. Detección del problema o necesidad.
    b. Búsqueda de información, investigación, planteamiento de la actividad.
    c. Búsqueda de soluciones posibles.
    d. Elección de las mejores de las soluciones, planteamiento de la actividad

2. **FASE DE CONSTRUCCIÓN**
    a. Diseño
    b. Preparación y planificación del trabajo.
    c. Construcción del producto.

3. **FASE DE EVALUACIÓN**
    a. Comprobación del resultado
    b. Presentación y evaluación.

## 1. FASE DE ANÁLISIS

### a. Detección del problema o necesidad.

El agua es el líquido más abundante de la tierra y el recurso natural más importante para los seres humanos, siendo la base de toda forma de vida.

El cambio climático provocado tanto por fenómenos naturales como por la acción del ser humano sobre el planeta, hoy en día es un hecho, que origina la desertización del planeta y el agotamiento de recursos hídricos.

Se pone de manifiesto la necesidad del buen aprovechamiento de los recursos hídricos y la implantación de las nuevas tecnologías, basadas en técnicas robóticas que permitan un desarrollo sostenible.

En los seres humanos existe una curiosidad natural por todo aquello que nos rodea. Esa curiosidad es el motor que nos impulsa a investigar, y por tanto a profundizar en el conocimiento de una determinada materia. Esta curiosidad puede potenciarse a través de la observación de fenómenos naturales, la cual nos lleva a la creación de inventos, y soluciones útiles para la humanidad.

El fenómeno natural de la lluvia, permite que se rieguen nuestros campos y jardines, observando el hecho de que la saturación de agua impide tanto el trabajo agrícola y el disfrute de jardines, durante un tiempo determinado, como el daño a los cultivos, no pudiendo ser controlado dicho fenómeno por el ser humano.

El principal objetivo es que el alumnado tome conciencia de la importancia del buen uso y aprovechamiento del agua, observando su entorno más cercano.

### b. Búsqueda de información, investigación, planteamiento de la actividad.

Se pide al alumnado que enumeren los efectos e inventos específicos que se han derivado de hechos o fenómenos naturales y/o artificiales tradicionales, relacionados con los siguientes aspectos, relacionados con la hidrografía:

- El cambio climático, la contaminación y la sostenibilidad.
- Recogida de aguas pluviales, desaladoras, agotamiento de acuíferos.
- Humedales y embalses naturales y artificiales de agua.
- Redes: Canales, galerías, tuberías, acequias,...
- Estaciones de tratamiento de agua potable.
- Estaciones depuradoras de aguas residuales
- Redes de abastecimiento y de riegos.

Estos aspectos son tratados en el centro por parte del profesorado participante del proyecto, en horario de clases, o bien en horario de tutorías. De esta manera destacamos las siguientes actuaciones, donde se han utilizado bien recursos propios o bien recursos facilitados en colabora INNICIA.

Biología y Geología: Actividades de análisis de la afección de la emisión de partículas a la atmósfera en los seres vivos y de fomento del empleo de las energías limpias renovables.

Filosofía y ciudadanía: Actividades de análisis sociológico de los hábitos de consumo y de fomento del desarrollo sostenible, tratando la obsolescencia.

Geografía e historia: Actividades de análisis de la evolución del consumo energético en España para evitar el cambio climático, a partir de datos históricos.

Física y química: Actividades de análisis de la reacción química de las partículas emitidas a la atmósfera con el agua de lluvia y de fomento de las energías limpias, que eviten la lluvia ácida.

Matemáticas y economía: Actividades de diseño de edificios bioclimáticos, teniendo en cuenta la orientación de los edificios y los ángulos de inclinación los rayos solares que inciden sobre los mismos, fomentando el ahorro energético calórico y lumínico.

Informática: Empleo de las herramientas de comunicación, y almacenamiento de datos, que fomentan el teletrabajo con el consiguiente ahorro de combustibles fósiles en el transporte.

Tecnología y economía: Empleo de sistemas robóticos y electrónicos para el ahorro de energía y reducción del consumo de agua.

Dibujo: Diseño de la carcasa y chasis del programador.

### c. Búsqueda de soluciones posibles.

Tras el análisis de la información se llega a las siguientes conclusiones:

- Se hace necesario un cambio de modelo social y económico sostenible, basado en tres pilares esenciales:
    - La protección medioambiental
    - El desarrollo social
    - Crecimiento económico
- La máxima captación y aprovechamiento de agua del planeta.
- La disminución de la contaminación de las aguas, la depuración de las mismas y disminución del consumo.
- La modernización de redes y conducciones de agua, con ayuda de nuevas tecnologías

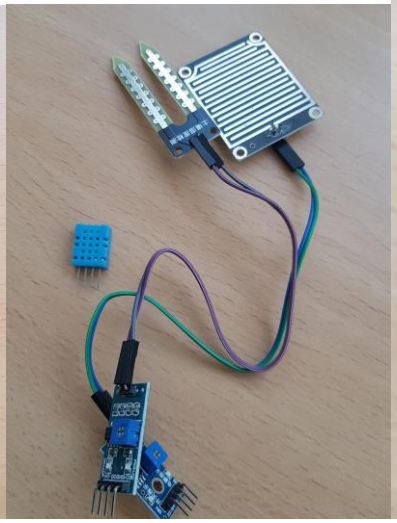
El presente proyecto pretende dar solución a uno de los aspectos analizados, correspondiente al aprovechamiento y reducción del consumo de agua en agricultura y jardines a través del empleo de nuevas tecnologías, basadas en la robótica.

Entre las soluciones obtenidas, se ha llegado a la conclusión, de que se puede reducir el consumo de agua, a través de un sistema de riego que analice tanto la humedad de la tierra, como la detección de la lluvia, teniendo en cuenta además la época de año y la temperatura.

Tras analizar los sistemas de riego actuales, se observa que no se controla, de forma automática, los factores que condicionan una humedad de la tierra óptima en el cultivo agrícola y de jardines.

### d. Elección de las mejores de las soluciones, planteamiento de la actividad.

Se establece como la mejor solución al planteamiento del problema un sistema robótico, con placa Arduino y sensores de humedad, lluvia y temperatura que tras el análisis mediante inteligencia artificial sea capaz de activar el sistema de riego cuando sea necesario, teniendo en cuenta los parámetros obtenidos.



El planteamiento de la actividad recoge la solución al problema, que permite una optimización del consumo de agua, a través de las siguientes fases:

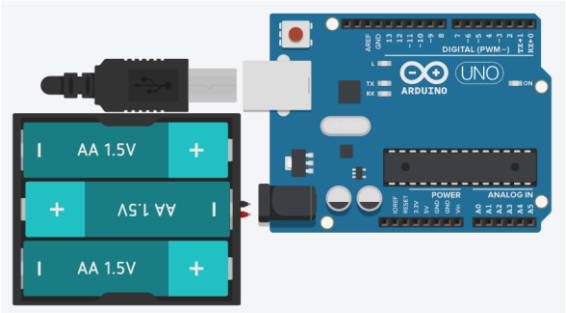
- Planificación y diseño
- Simulación
- Construcción
- Control del dispositivo
- Depuración y evaluación.

## 2. FASE DE CONSTRUCCIÓN

### a. Diseño

En el diseño y simulación, se ha utilizado la herramienta de diseño de Tinkercad (software libre on line), para diseñar el **chasis o carcasa del dispositivo**, esta actividad nos ha permitido emplear técnicas innovadoras de diseño asistido por ordenador 3D.

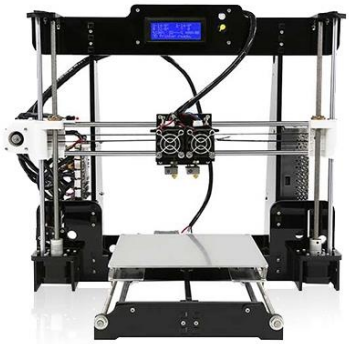
Para el diseño se ha tenido el cuenta el tamaño de la placa arduino uno para diseñar las dimensiones de la carcasa, así como las dimensiones del portapilas de alimentación de la placa.



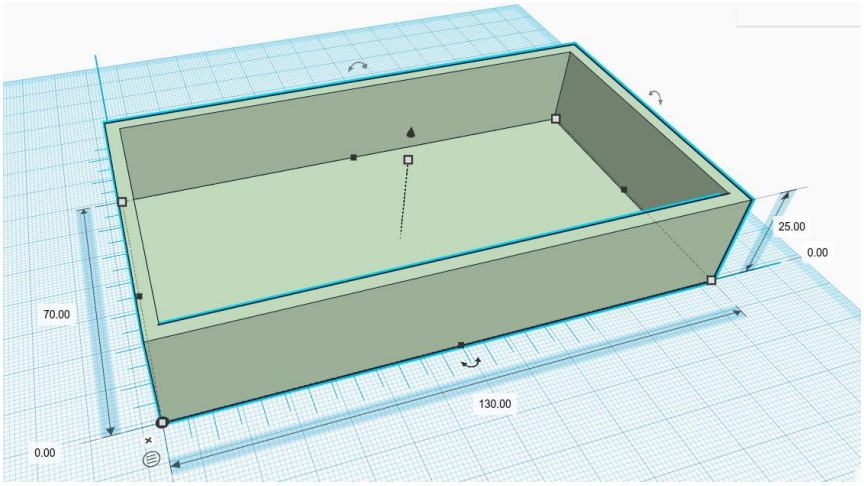

Tras el análisis de los dispositivos se observa que las dimensiones ideales sería un prisma rectangular de las siguientes medidas interiores:

Largo de 130 mm.
Ancho de 70 mm.
Alto de 25 mm.

El material elegido para la fabricación de la carcasa o chasis, es filamentos de **plástico de tipo "PLA"** (ácido poliláctico biodegradable, expuesto en la materia de química y biología) elaborado mediante impresión 3D.



Obteniendo como resultado final el siguiente producto.




El reto del dispositivo planteado es la activación y excitación del relé de corriente continua/alterna, para poner en marcha la electroválvula de riego, que permita la salida de agua en las siguientes condiciones:

1. No debe detectarse lluvia.
2. La humedad del suelo inferior al 25%
3. La frecuencia de medición de la humedad del suelo dependerá de la temperatura exterior y del detector de lluvia, estableciéndose las condiciones:

    a. Mientras se detecta lluvia, no se realizan mediciones de humedad del suelo.
    b. A partir de la última detección de lluvia, y si la temperatura ambiente es inferior a 15ºC, la frecuencia de medición de humedad del suelo será cada 72 horas.
    c. A partir de la última detección de lluvia, y si la temperatura ambiente se encuentra entre 15 y 20ºC, la frecuencia de medición de humedad del suelo será cada 48 horas.
    d. A partir de la última detección de lluvia, y si la temperatura ambiente se encuentra entre 20 y 30ºC, la frecuencia de medición de humedad del suelo será cada 24 horas.
    e. A partir de la última detección de lluvia, y si la temperatura ambiente se encuentra entre 30 y 35ºC, la frecuencia de medición de humedad del suelo será cada 12 horas.
    f. A partir de la última detección de lluvia, y si la temperatura ambiente es superior a 35ºC, la frecuencia de medición de humedad del suelo será cada 6 horas.

Para la simulación del dispositivo vamos a utilizar el simulador de Tinkercad. Para ello utilizaremos el sensor de temperatura y sustituiremos solo para la simulación los

sensores de lluvia y de humedad por interruptores, utilizando leds para la simulación de la activación de las electroválvulas.

## b. Simulación:

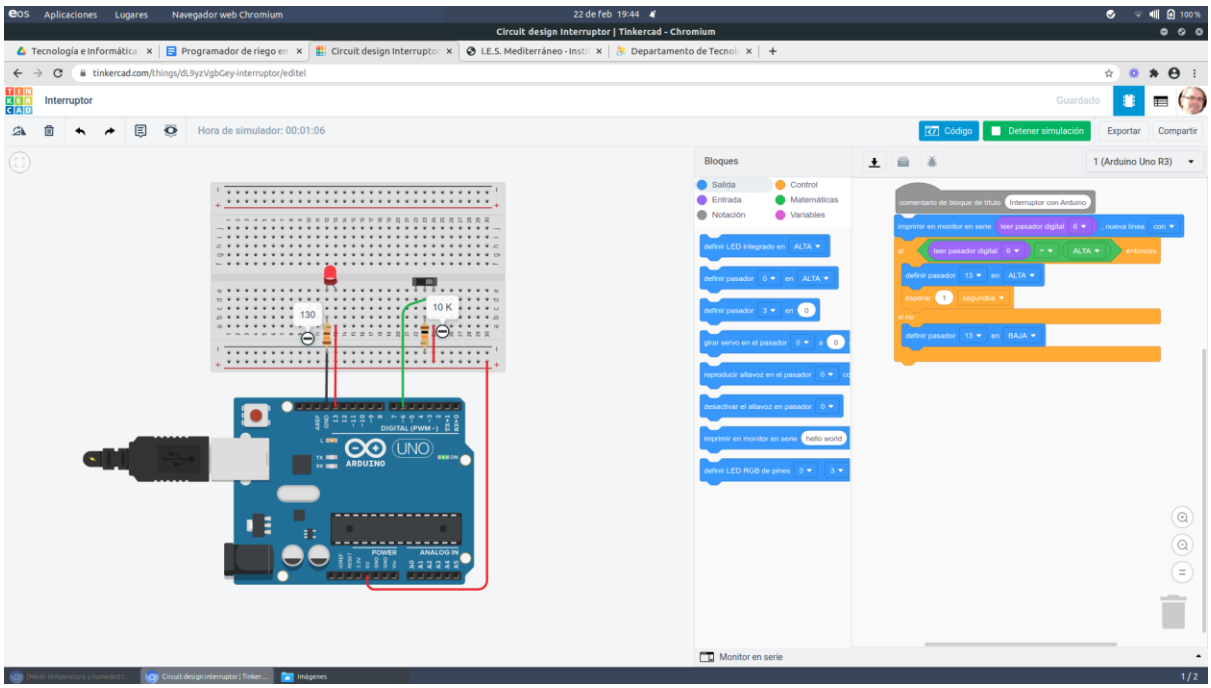
Para la simulación hemos utilizado la clase virtual que nos proporciona “Tinkercad” y elaborado imágenes y vídeos que se han subido al blog de Tecnología del IES. https://tecnologia.institutomediterraneo.es/. Estructurados en dos apartados:

1. Pequeños proyectos de robótica: Donde se establecen pequeños retos sencillos de robótica que sirven para la iniciación a la misma y donde se resuelve de forma gradual de menor a mayor dificultad.

2. Proyecto emprendedor: Donde se define y resuelve el reto principal del proyecto emprendedor.

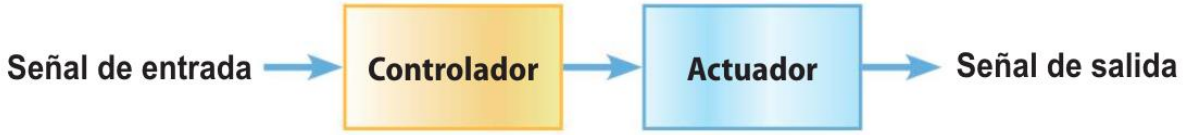
Distinguiremos los siguientes pequeños retos y proyectos robóticos:

*Proyecto de interruptor simple con placa Arduino:*

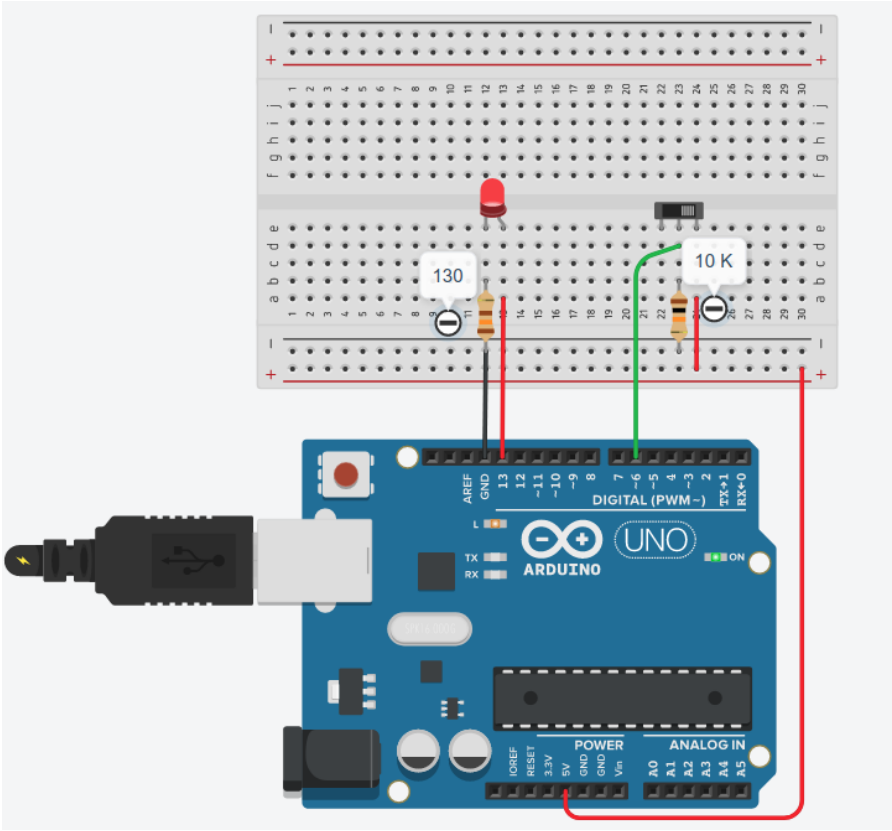



Este pequeño proyecto nos permite entender el funcionamiento de la robótica, donde siempre partimos de una señal de entrada, seguidamente se analiza a través de la placa arduino, mediante unas simples instrucciones o pequeño programa, para finalmente origina

una señal de salida que en este caso corresponde a la salida de energía que permite la iluminación del led.




El reto se establece de que se encienda un led cuando activamos el interruptor (posición de encendido), y que se apague cuando lo desactivamos (posición de apagado).




Analizando la figura vemos que la señal de entrada a la placa se realiza por el pin digital número 6, y la señal de salida se realiza por el pin digital número 13, disponiendo tanto a la entrada como a la salida resistencias de protección que nos permiten adaptar la tensión.

La idea es que este interruptor va a ser sustituido en los siguientes retos por un sensor que automáticamente transmita la señal del evento detectado, y el led va a ser sustituido por un relé electromagnético que nos permita activar con corriente alterna la electroválvula.

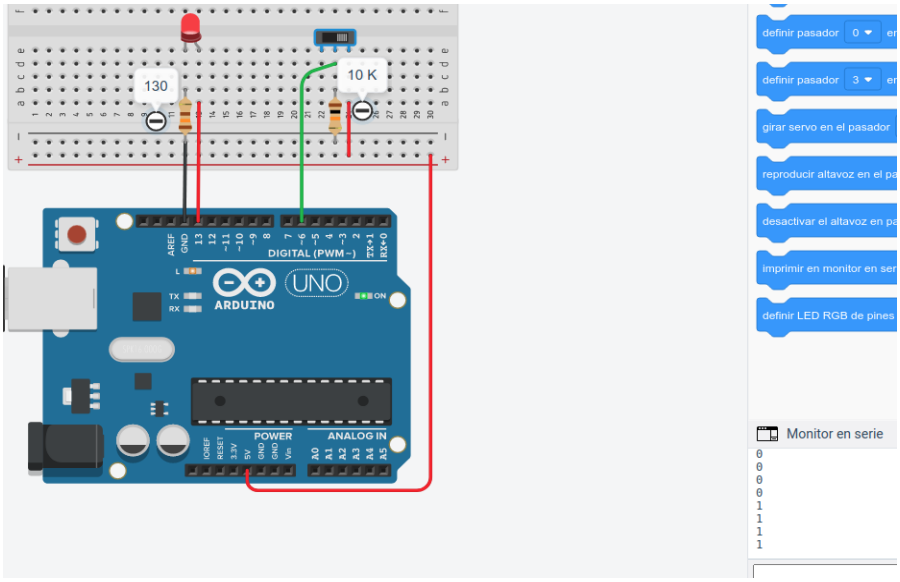
Vemos que Tinkercad, nos permite trabajar con bloques que nos va a facilitar la tarea de la programación, observando que los pines pueden adoptar dos estados, bien en “ALTA” o en “BAJA”, que corresponden a los valores numéricos 0 y 1 respectivamente.

El lenguaje de programación de Arduino, es similar a un lenguaje C++, y para la programación de bloques vista anteriormente arroja las siguientes sentencias:

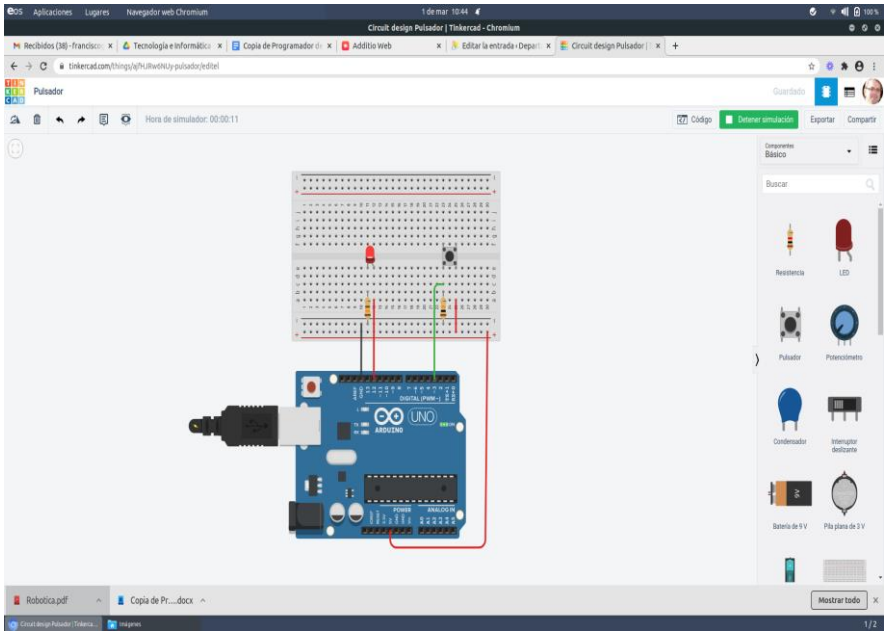
```
/*
  Interruptor con Arduino
*/

void setup()
{
  pinMode(6, INPUT);
  Serial.begin(9600);

  pinMode(13, OUTPUT);
}

void loop()
{
  Serial.println(digitalRead(6));
  if (digitalRead(6) == HIGH) {
    digitalWrite(13, HIGH);
    delay(1000); // Wait for 1000 millisecond(s)
  } else {
    digitalWrite(13, LOW);
  }
}
```

Como observamos, cuando por el pin digital 6 le entre energía o está activado (señal en ALTA o 1), sale energía por el pin digital 13 (señal en ALTA o 1), en caso contrario se desactiva (sela en BAJA o 0).

El **monitor en serie** nos permite comprender el funcionamiento del reto, cero apagado, 1 encendido.

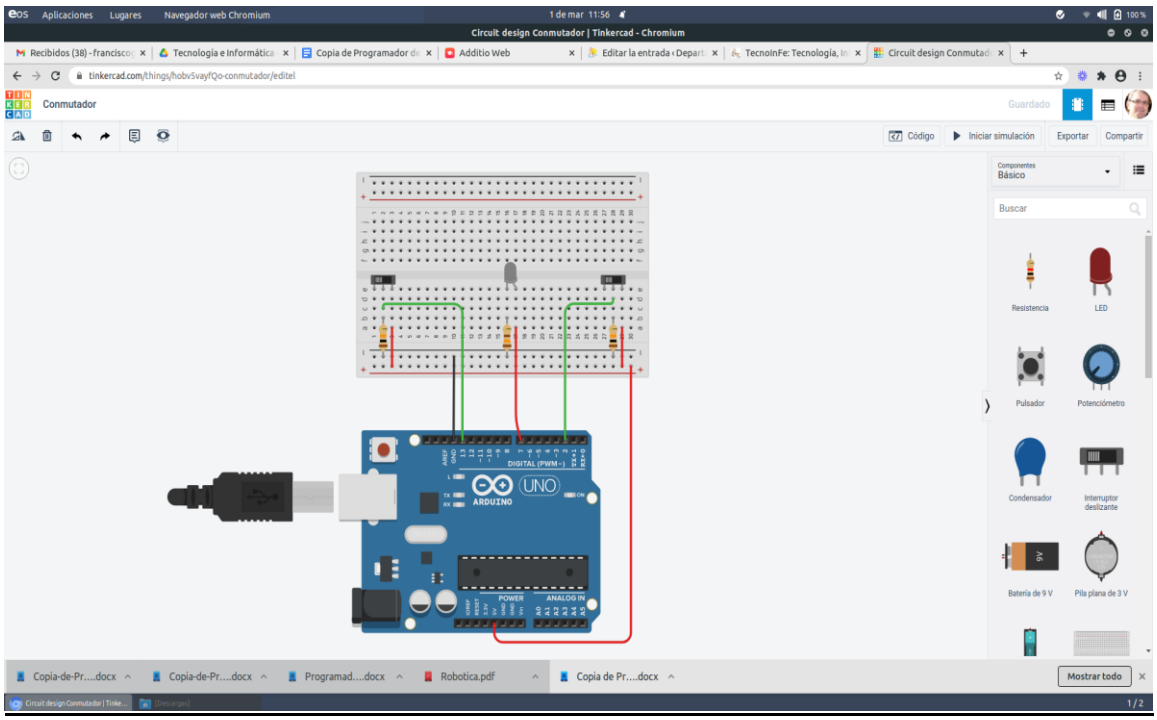



*Proyecto de pulsador simple con placa Arduino*



Este pequeño proyecto funciona igual que el interruptor simple, de hecho el código es el mismo, pero nos permite comprender el dispositivo pulsador, similar al de un pulsador del timbre de una vivienda, es decir, mientras pulsamos funciona, pero cuando dejamos de pulsar deja de funcionar.

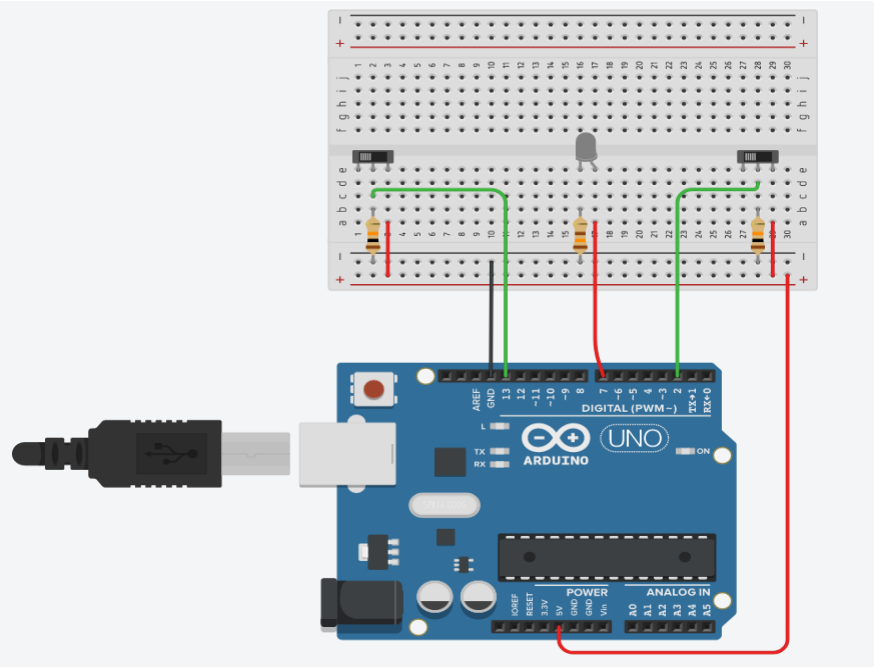
*Proyecto de conmutador con placa Arduino*




Este pequeño proyecto nos permite controlar el accionamiento de una luz desde dos interruptores colocados en distintos lugares, como por ejemplo los que existen en la habitación de tu casa.

Disponemos por tanto de dos señales de entrada, que se analizan a través de la placa Arduino, mediante unas simples instrucciones o pequeño programa, para finalmente origina una señal de salida que en este caso corresponde a la salida de energía que permite la iluminación del led.

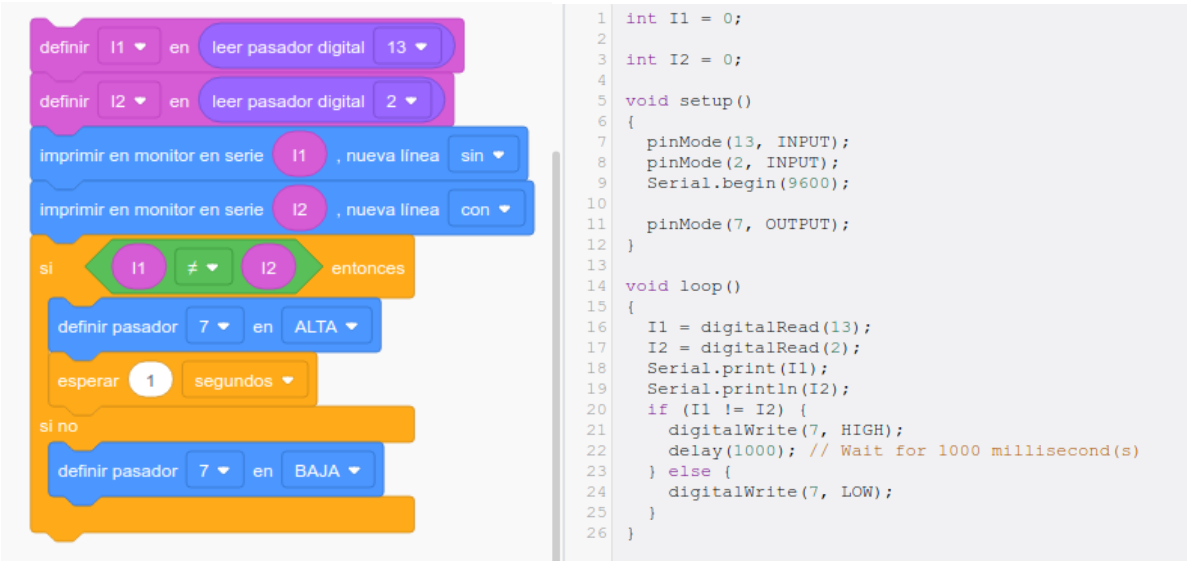
El reto se establece de que se encienda un led cuando activamos cualquiera de los dos interruptores (posiciones de encendido), y que se apague cuando lo desactivamos (posiciones de apagado).




Analizando la figura vemos que las señales de entrada a la placa se realizan por el pin digital número 2 y 13, y la señal de salida se realiza por el pin digital número 7, disponiendo tanto a la entrada como a la salida resistencias de protección que nos permiten adaptar la tensión.

Este proyecto permite la combinación de dos señales, y el análisis de las mismas, procedentes de dos sensores y el led va a ser sustituido, como siempre, por un relé electromagnético que nos permita activar con corriente alterna la electroválvula.

La programación en bloques sería la siguiente:

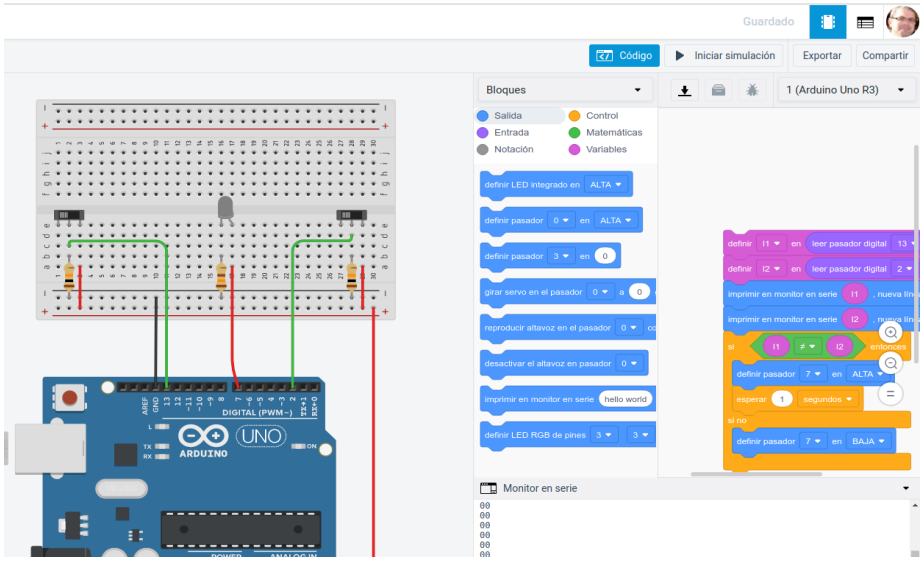


Definimos dos variables “I1 e I2” de tipo entero de valor inicial cero, donde vamos a pasar los valores que entre por los pines 13 y 12, en la simulación vamos a hacer visible estos valores a través de las líneas definidas en las líneas “imprimir en monitor en serie”.

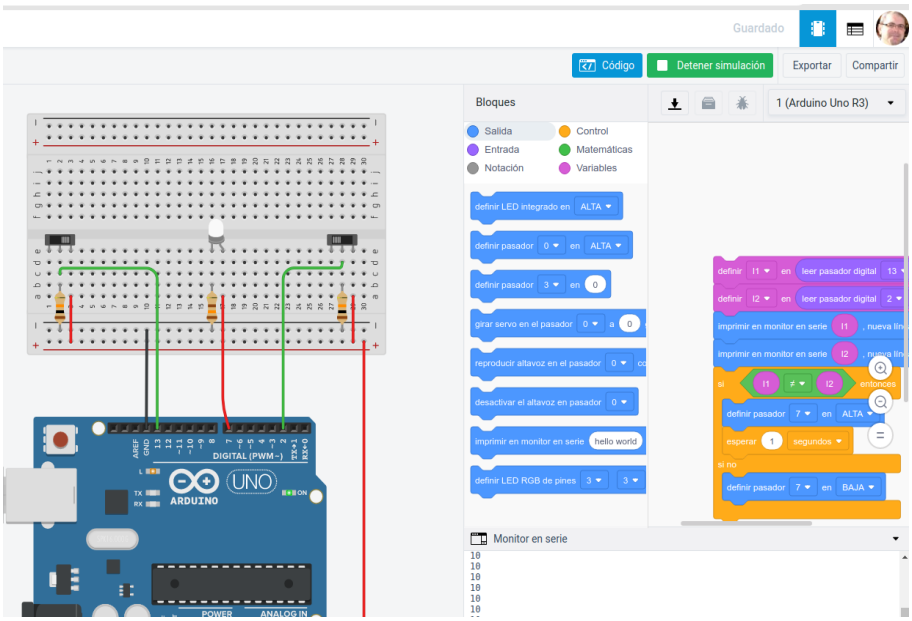
Analizando los valores de estas variables, que corresponden al estado de cada uno de los interruptores (desconectado 0; conectado 1) de manera que siempre que sean diferentes sale energía por el pin 7 (ALTA), y en caso contrario no sale energía (BAJA).

Veamos la simulación en las distintas posiciones de los interruptores:

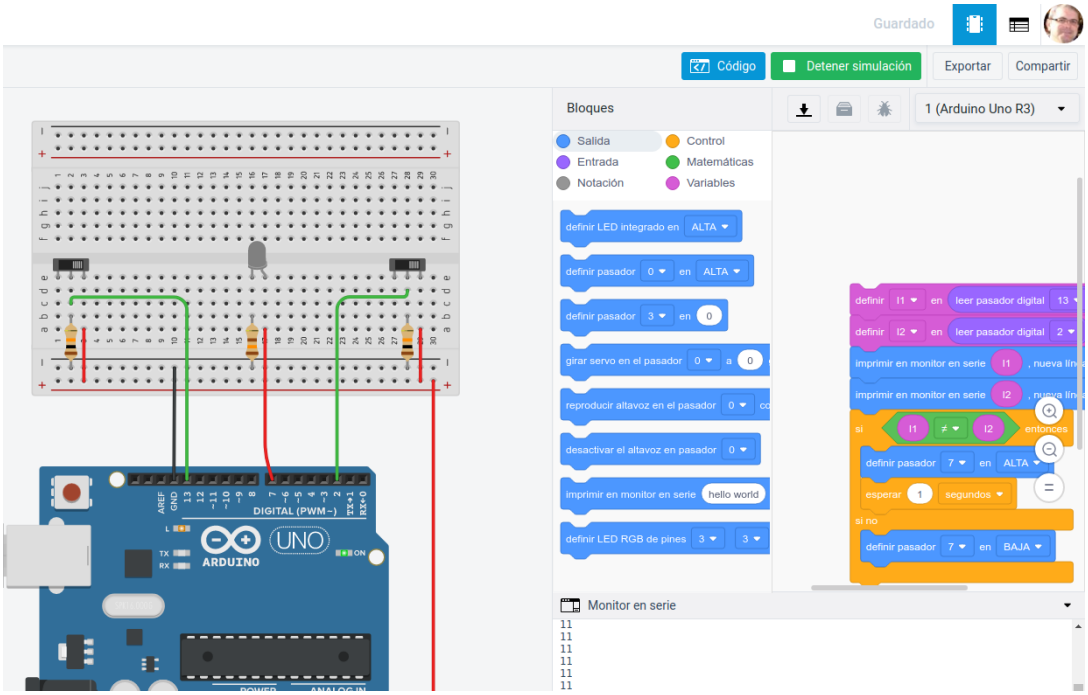
Posición primera de partida (apagado) I1=0 e I2=0



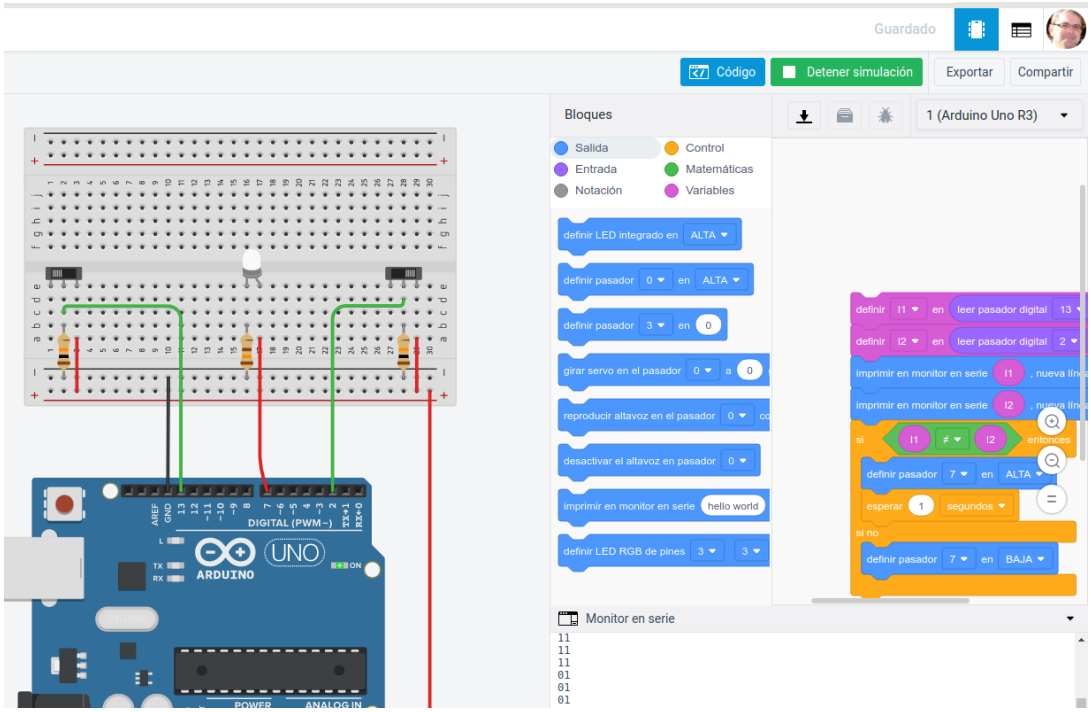

Posición segunda (encendido) I1=1 e I2=0



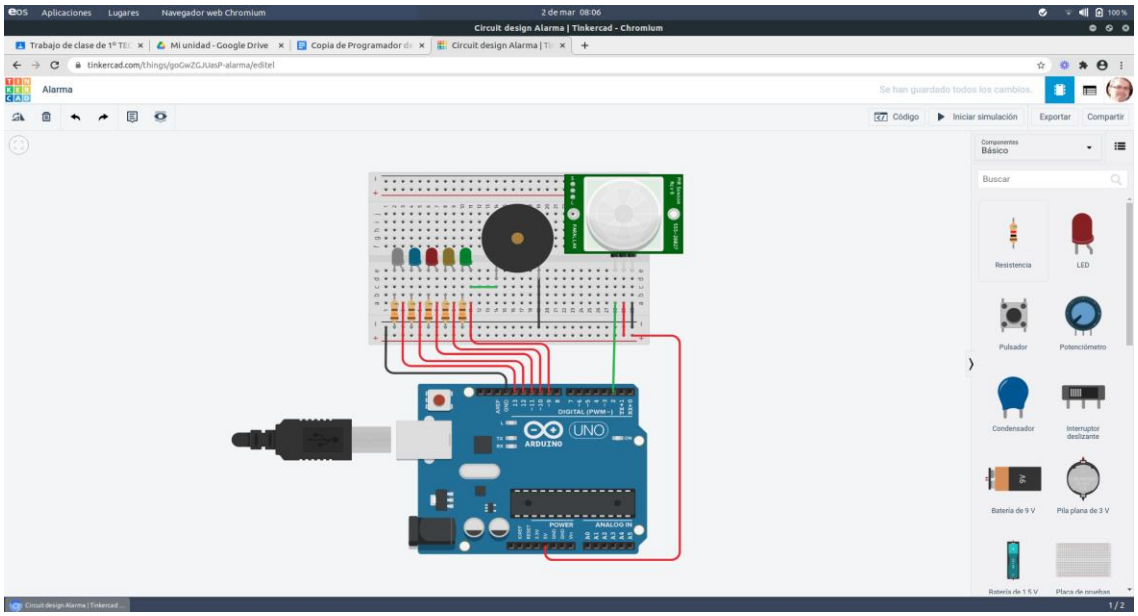
Posición tercera (apagado) I1=1 e I2=1




Posición cuarta (encendido) I1=0 e I2=1

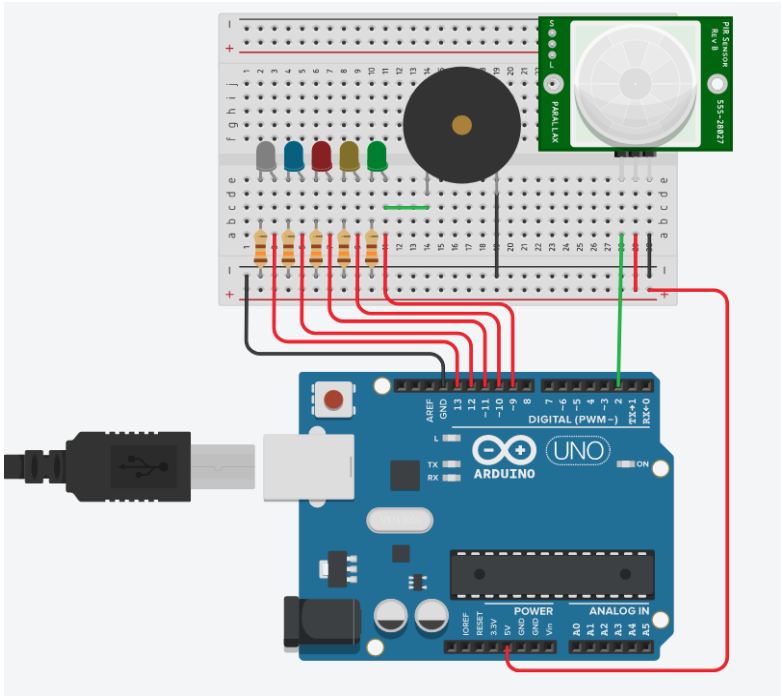


*Proyecto de alarma con placa Arduino, sensores digitales*

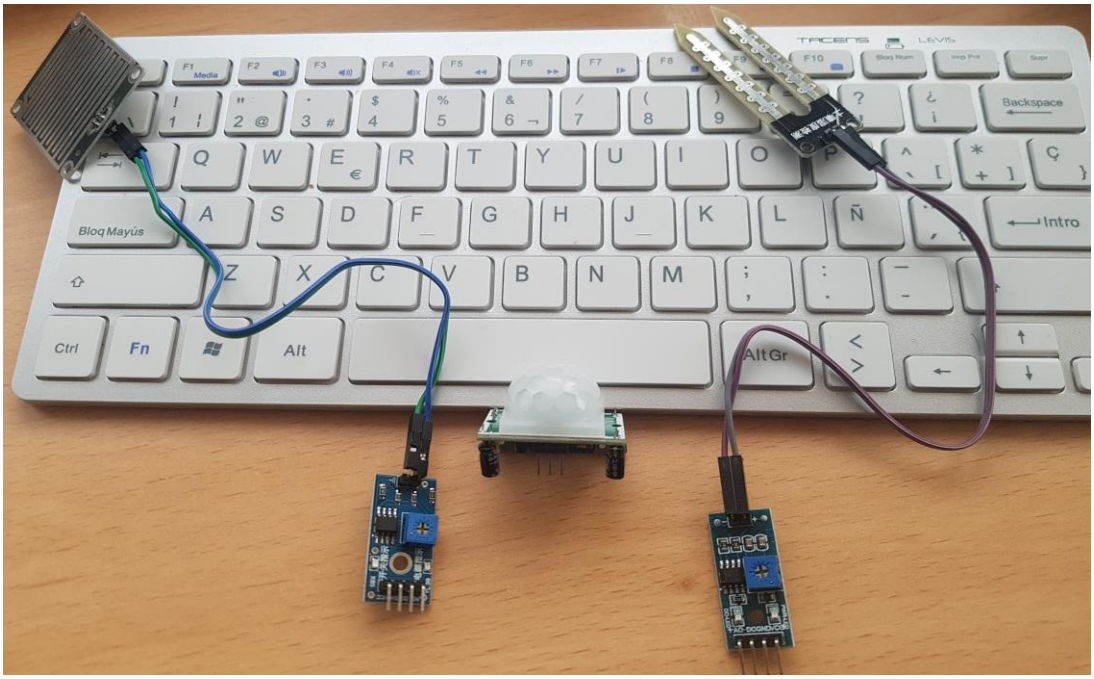



Este proyecto es similar al del interruptor simple, pero cambiando la acción manual del interruptor por un sensor de presencia, el cual hace funcionar la alarma cuando detecta presencia.

A diferencia del interruptor simple, el sensor de presencia debe ser alimentado con energía eléctrica, porque funciona de manera permanente.



De esta manera alimentamos la placa de pruebas Protoboard desde el pin de 5 voltios y el pin GND de la placa arduino a la fila del positivo y negativo de la placa Protoboard. Una vez alimentada la placa alimentaremos el sensor, tal y como se muestra en la figura, y conectaremos el cable de señal al pin digital Nº2, el cual pasará señal “1” o “ALTA” si detecta movimiento, conectando el sistema o en caso contrario pasará señal “0” o “BAJA” sino detecta movimiento.

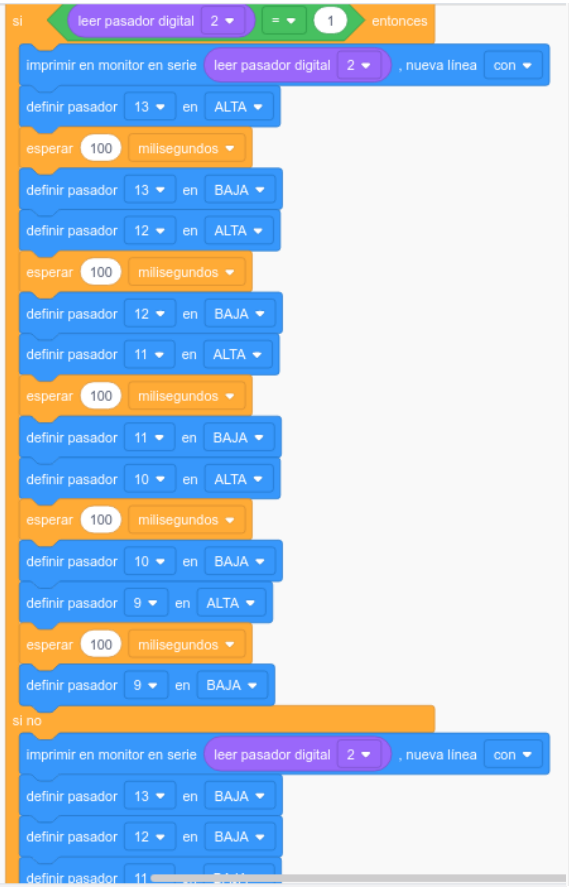



El sensor de presencia de color blanco de presencio observamos los tres pines de conexión indicados anteriormente, correspondientes dos pines a la alimentación y otro pin de tipo digital de señal.

Como observamos tanto en el sensor de detector de lluvia como en el sensor del detector de humedad terrestre, disponemos de cuatro pines, dos correspondientes a la alimentación y dos pines de señal uno digital y otro analógico, controlados ambos a su vez con potenciómetros.

Los pines digitales solo pasan dos valores, o “1” “ALTA” estado activado o “0” “BAJA” estado desactivado, observando que podríamos sustituir sin problemas el sensor de presencia por cualquier de estos detectores, bien el detector de lluvia (aplicación en el parabrisas de un coche) o el sensor de humedad de tierra (aplicación de riego en un invernadero), empleando su pin digital. Por tanto el motivo de emplear el sensor de presencia en el simulador de Tinkercad, es porque no han incorporado aún estos sensores a los componentes.

En cuanto a la programación en bloques y al código en lenguaje de programación, quedaría de la siguiente manera:



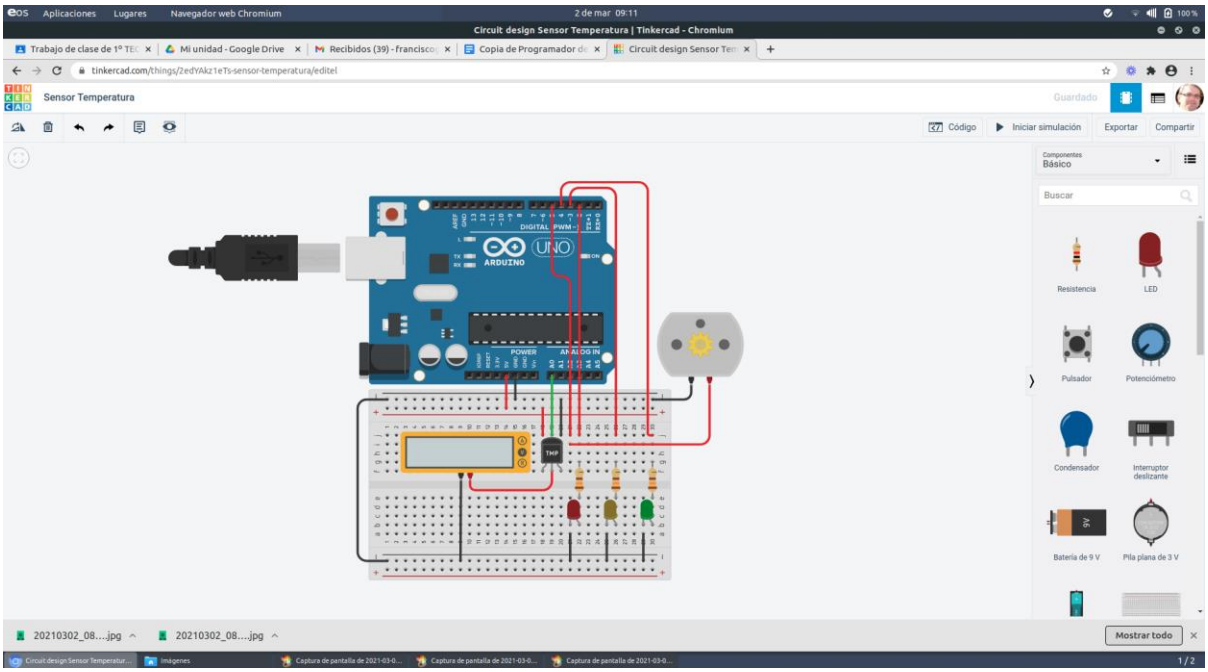

```
int sensor = 0;

void setup()
{
  pinMode(2, INPUT);
  Serial.begin(9600);

  pinMode(13, OUTPUT);
  pinMode(12, OUTPUT);
  pinMode(11, OUTPUT);
  pinMode(10, OUTPUT);
  pinMode(9, OUTPUT);
}

void loop()
{
  if (digitalRead(2) == 1) {
    Serial.println(digitalRead(2));
    digitalWrite(13, HIGH);
    delay(100); // Wait for 100 millisecond(s)
    digitalWrite(13, LOW);
    digitalWrite(12, HIGH);
    delay(100); // Wait for 100 millisecond(s)
    digitalWrite(12, LOW);
    digitalWrite(11, HIGH);
    delay(100); // Wait for 100 millisecond(s)
    digitalWrite(11, LOW);
    digitalWrite(10, HIGH);
    delay(100); // Wait for 100 millisecond(s)
    digitalWrite(10, LOW);
    digitalWrite(9, HIGH);
    delay(100); // Wait for 100 millisecond(s)
    digitalWrite(9, LOW);
  } else {
    Serial.println(digitalRead(2));
    digitalWrite(13, LOW);
    digitalWrite(12, LOW);
    digitalWrite(11, LOW);
    digitalWrite(10, LOW);
    digitalWrite(9, LOW);
  }
}
```

Como vemos permanentemente se está leyendo el pin de señal del sensor a través del pin digital Nº2 y en cuanto detecta presencia se activa (estado en 1 o ALTA), da salida de energía (estado en 1 o ALTA) por los pines digitales 13, 12, 11, 10 y 9, encendiendo los leds y sonando la sirena. En caso contrario no da salida de energía permaneciendo el sistema en reposo.

Al igual que los anteriores proyectos los leds pueden sustituirse por relés de activación de electroválvulas, como el que vemos en la imagen.

Donde observamos en la parte derecha, la entrada de corriente continua VCC (polo positivo) y GND (polo negativo) y N1 (pin de activación, correspondiente a la salida de cualquier pin digital de la placa, bien el 13, 12, 11, 10 o 9, y en el lado izquierdo la entrada de uno de los polos (común) de corriente alterna en la zona central y la salida de ese común con destino a la electroválvula, que puede ser activa (NO, normalmente abierta) o cerrada (NC, normalmente cerrada), señalando que el otro polo de alterna (fase) iría directamente a la electroválvula.

*Proyecto de termostato con placa Arduino, sensores analógicos*

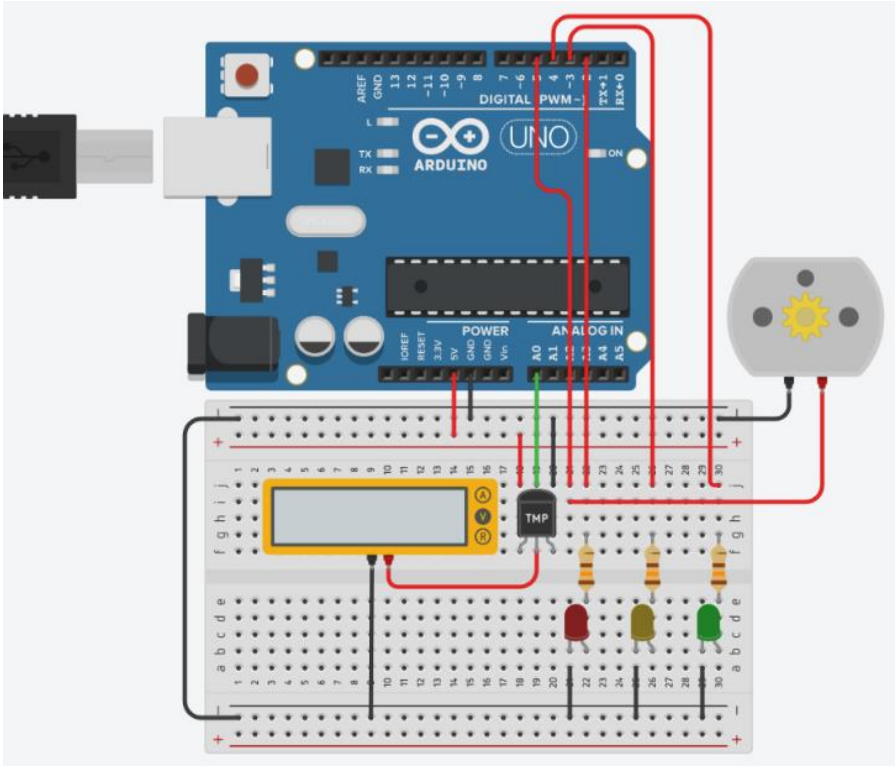



Este pequeño proyecto nos va a permitir conocer el funcionamiento de un sensor analógico, y el tratamiento de la información aportada a la placa Arduino.

El reto planteado es avisar mediante señal luminosa la temperatura a la que se encuentra el sensor, de manera que deben cumplirse las siguientes condiciones:

1ª.- Si la temperatura es menor de 20º C, no se encendería ninguna señal luminosa.
2ª.- Si la temperatura se encuentra entre 20º y 40ºC se enciende el led verde.
3ª.- Si la temperatura se encuentra entre 40º y 60ºC se enciende el led amarillo.
4ª.- Si la temperatura se encuentra entre 60º y 80ºC se pone en marcha el motor.
5ª.- Si la temperatura es mayor de 80ºC se enciende el led rojo.

Este dispositivo simula el funcionamiento del termostato de un vehículo, avisando de la temperatura, activando el ventilador o bomba hidráulica de refrigeración entre 60º y 80ºC, y avisando con el led rojo de la parada del vehículo, cuando la temperatura sobrepasa 80ºC




Analizando las conexiones de los dispositivos, alimentamos la placa Protoboard, tal y como vimos anteriormente desde la placa Arduino, porque necesitamos alimentar de manera continua al sensor de temperatura a través de las patillas de sus extremos.

La patilla central del sensor es la que manda información al pin analógico “A0” de la placa Arduino, del voltaje que varía en función de la temperatura del sensor y cuyo intervalo de valores debe oscilar entre 0 y 1023, es decir, un voltaje de cero voltios corresponde a 0 y un voltaje de 5 voltios corresponde a 1023, por lo que manejaremos un abanico de 1024 valores que son los que admite los pines analógicos de la placa Arduino.

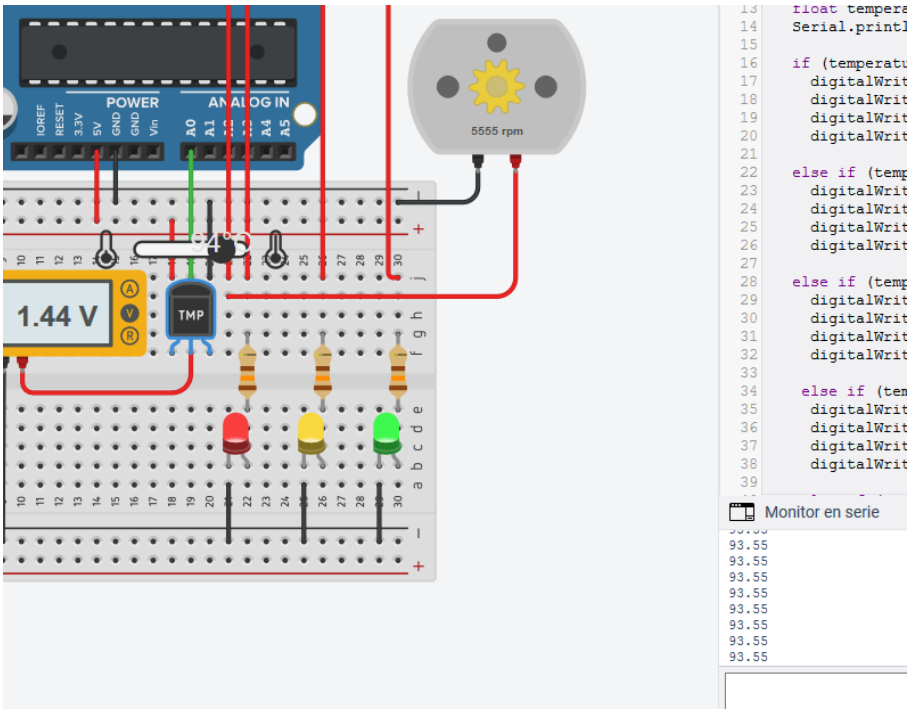
Sin embargo, a veces los sensores, dependiendo de los fabricantes, no dan valores dentro de este abanico y tendremos que calibrar el sensor en función de los valores máximos y mínimos. Esta calibración la obtendremos definiendo una función matemática, definida en el código de programación, que transforma lecturas del sensor de temperatura en ºC.

int sensorVal = analogRead(A0); // define la variable entera sensorVal y le pasa la lectura del sensor por el pin analógico A0

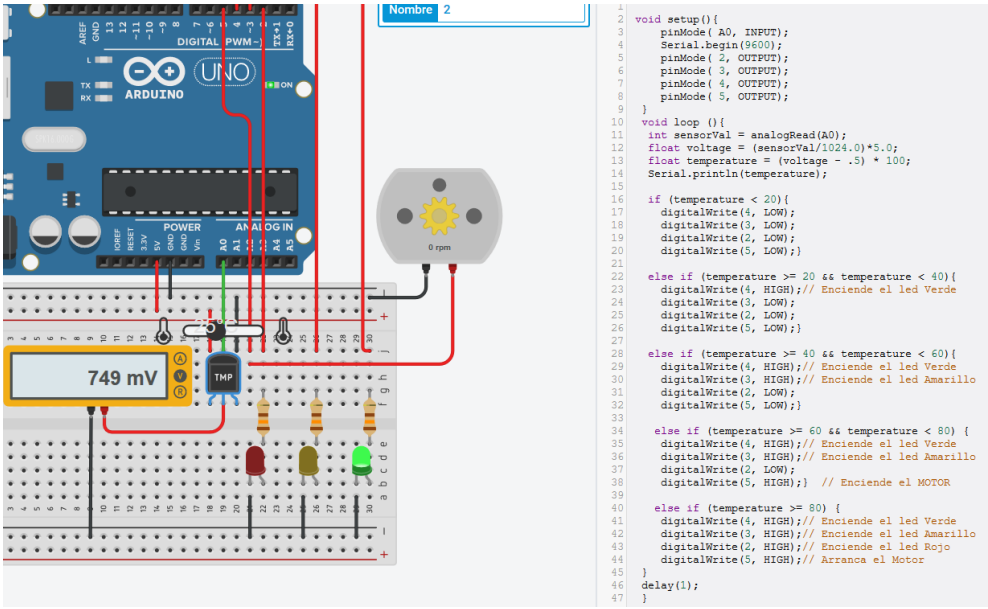
float voltage = (sensorVal/1024.0)*5.0; // define la variable decimal voltaje, para calcular los valores entre 0 y 1023, a partir del voltaje de 0 a 5 V.

float temperature = (voltage - .5) * 100;  // define la variable decimal temperatura y la calcula con la variable voltaje calculada anteriormente.

Para comprobar que la calibración es correcta debe aparecer en el monitor en serie la misma temperatura que el sensor.




Realizada la calibración y disponiendo del valor de la temperatura almacenado en la variable que lleva su mismo nombre en ºC, solo nos queda analizar la misma para dar salida de energía por los pines 2, 3, 4 y 5. Resultando el siguiente código de programación:

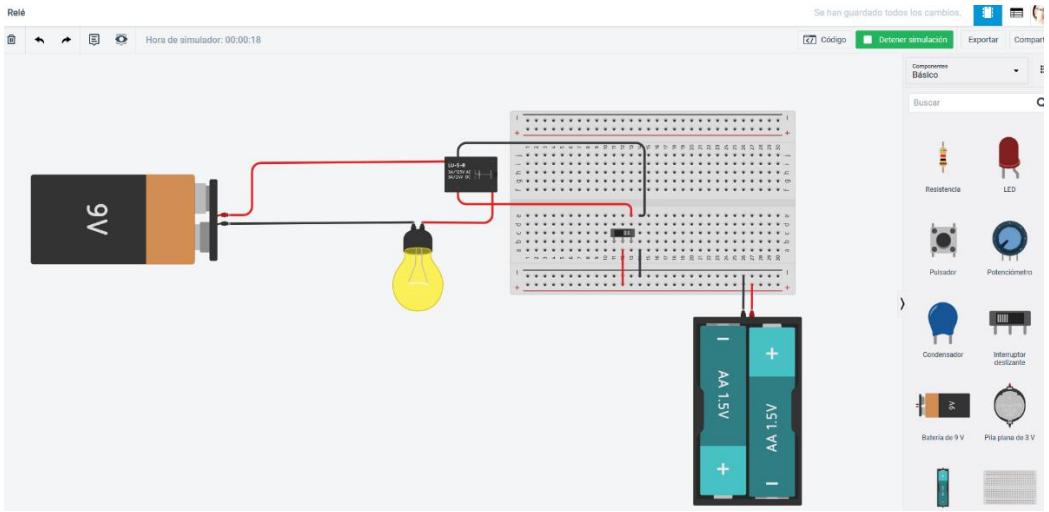


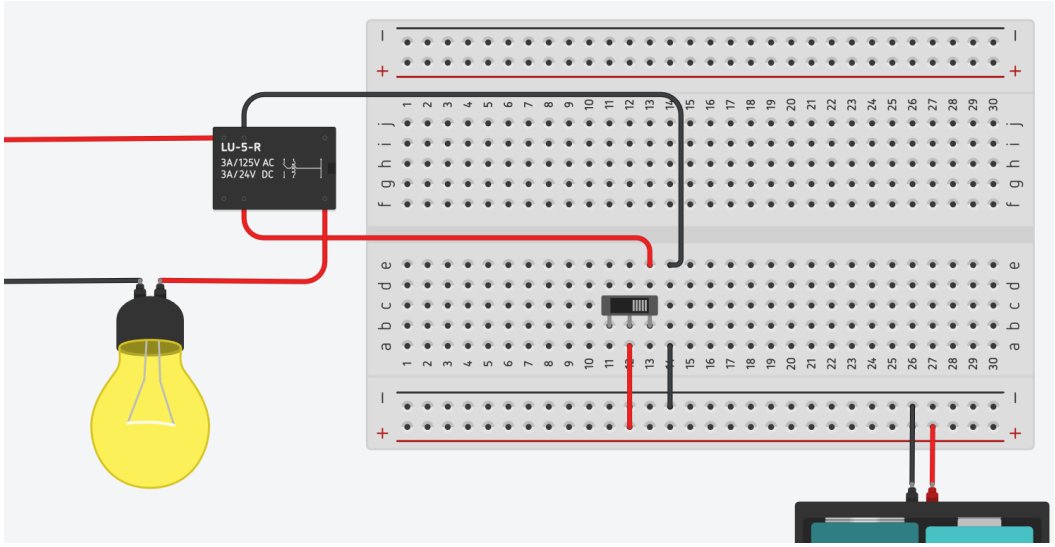
## *Proyecto de simulación de relé con Tinkercad*

Sin duda uno de los grandes descubrimientos es el relé, atribuido tanto al científico estadounidense Joseph Henry y al inventor inglés Edward Davy, sobre 1931 y 1935, que buscaban una mejora del telégrafo.

Un relé electromagnético se activa al pasar una corriente por una bobina o solenoide enrollado a un núcleo de hierro, generando un campo magnético que desplaza los contactos de los polos activando el paso de corriente por los mismos.

Al no disponer el simulador fuente de energía alterna, emplearemos para la simulación el siguiente relé de Tinkercad, empleando una fuente de alimentación de 3V, con el que activamos una bombilla con 9V de alimentación, para comprender el funcionamiento de activación del relé.

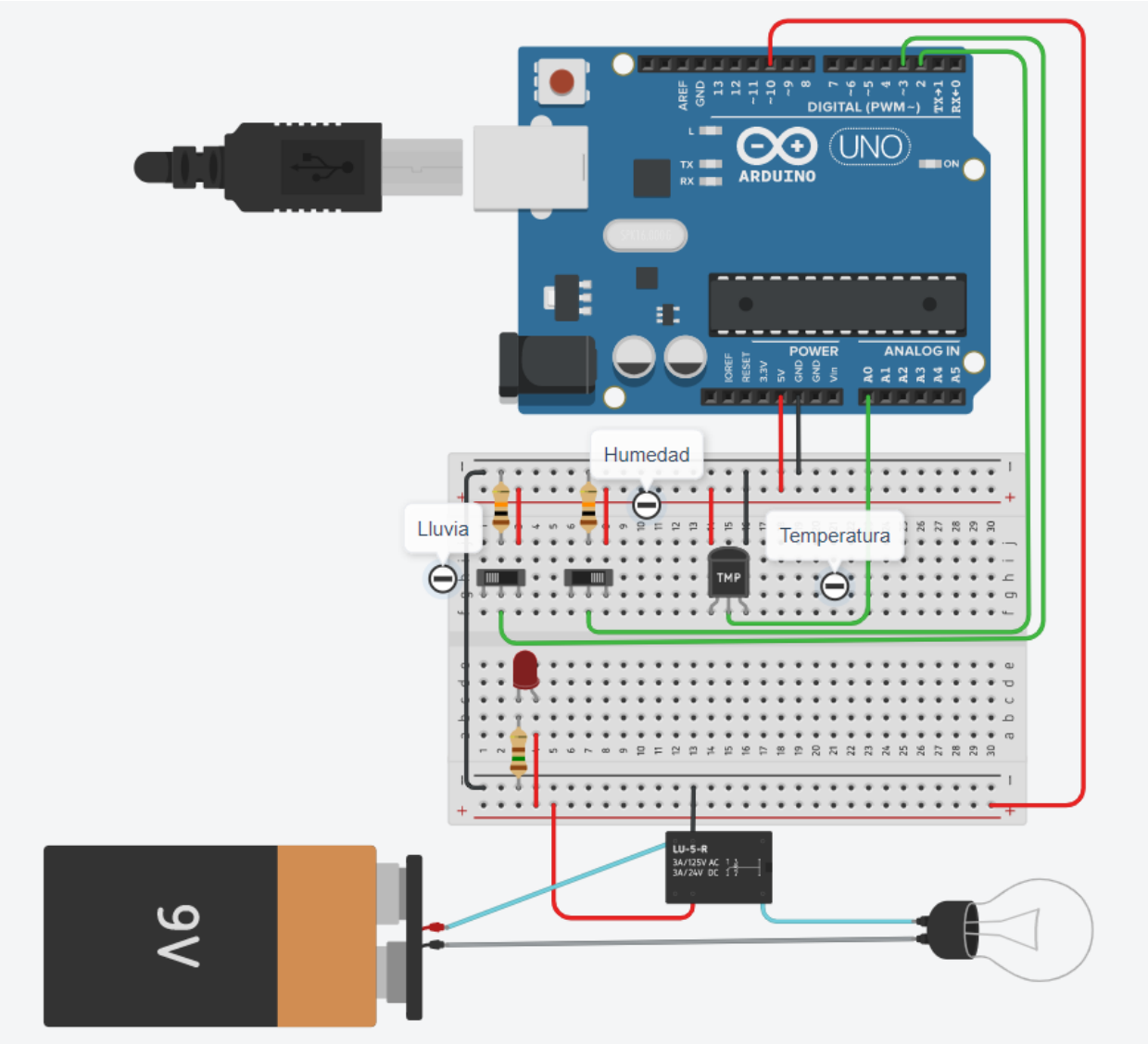


*Simulación del Proyecto de Programador de riegos para agricultura y jardines.*

Los anteriores proyectos vistos, no van a servir de base para la elaboración del proyecto final, ya que podremos considerarlo todos en un conjunto para obtener el resultado final, planteado en el reto.

De esta manera dispondremos del siguiente montaje en el simulador Tinkercad:

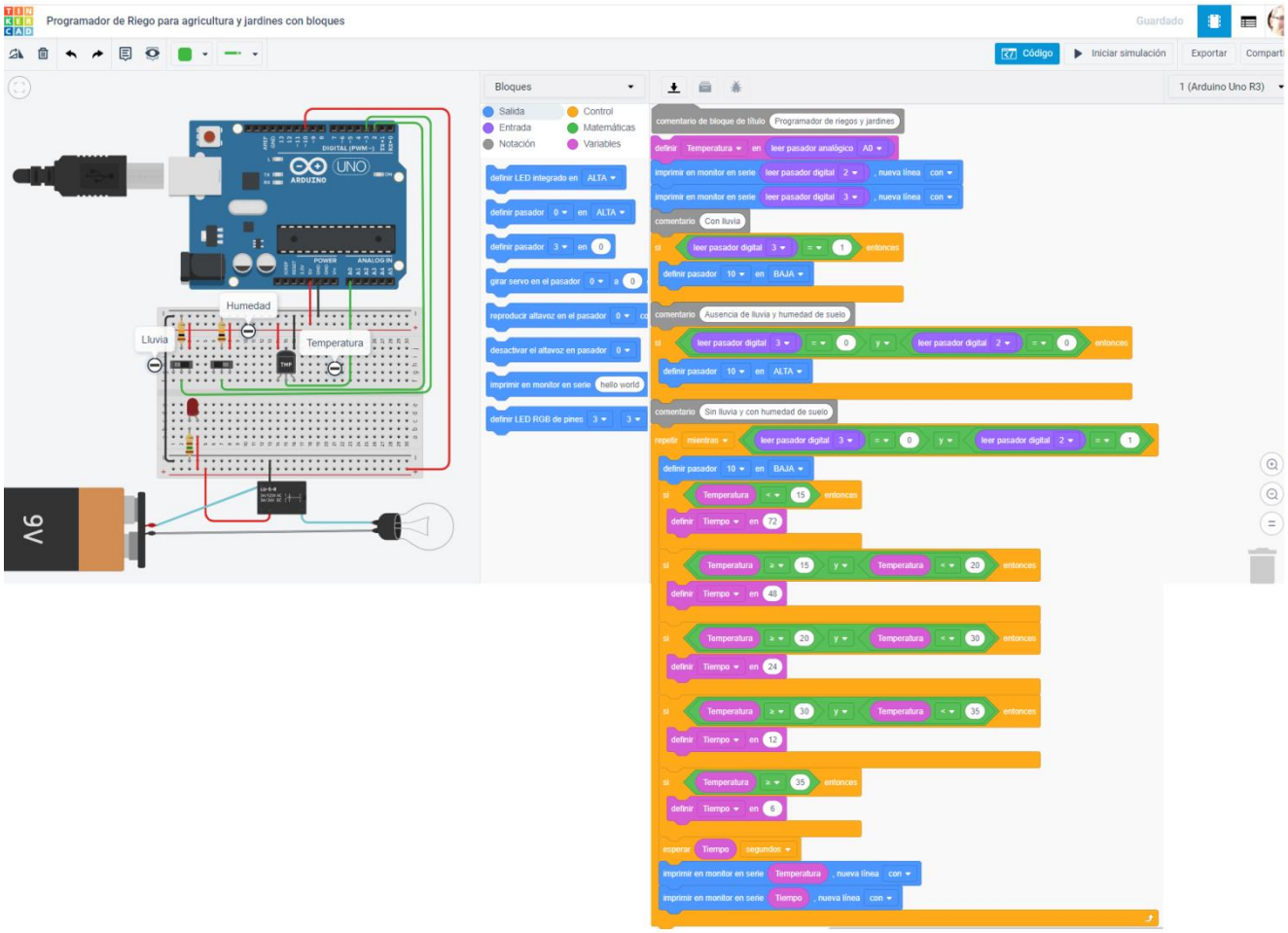



Analizando el montaje del simulador, y al no disponer el mismo de todos los dispositivos y sensores necesarios, realizaremos las siguientes consideraciones:

a. El primer interruptor de la izquierda, se va considerar un sensor de lluvia, con entrada por el pin digital D3
b. El segundo interruptor, se va considerar un sensor de humedad, con entrada por el pin digital D2
c. El sensor de temperatura va a tener entrada analógica por el pin analógico A0.
d. La salida de energía si se cumplen las condiciones de programación se realizará por el pin digital 10 y excitará el relé.
e. La pila de 9V es considerada como fuente de alimentación alterna al carecer el simulador de esta.
f. La bombilla simula la apertura y cierre de la electroválvula de riego.

Estas condiciones nos llevan a los siguientes códigos de programación, teniendo en cuenta la similitud de segundos a horas:

Por bloques:



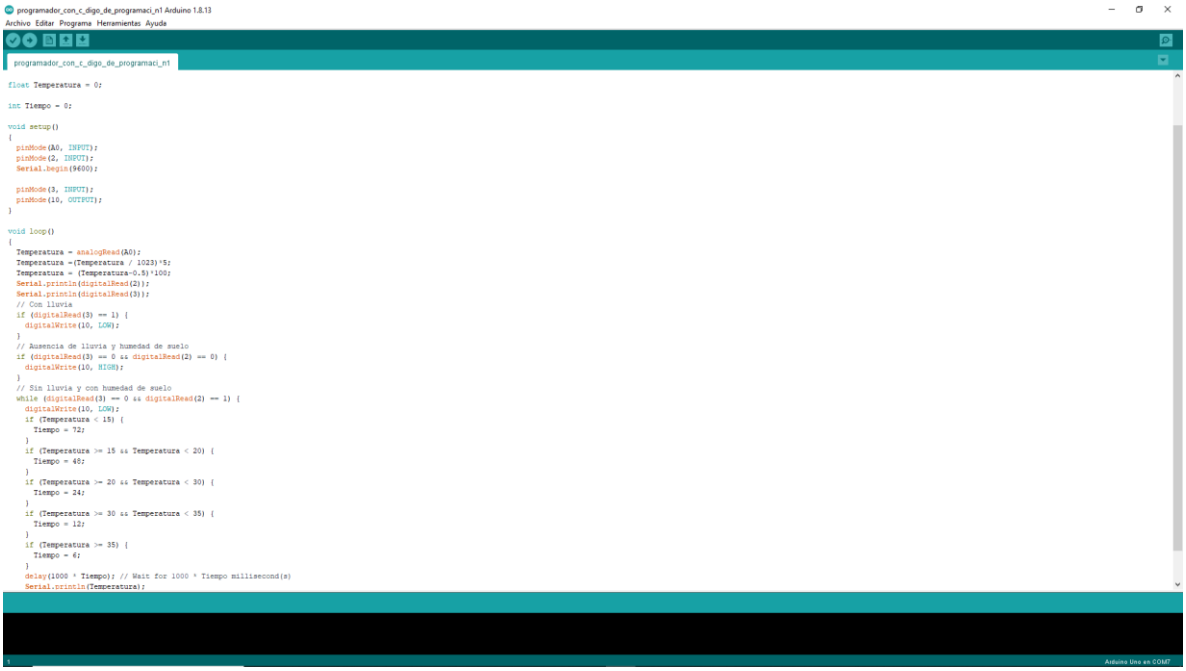
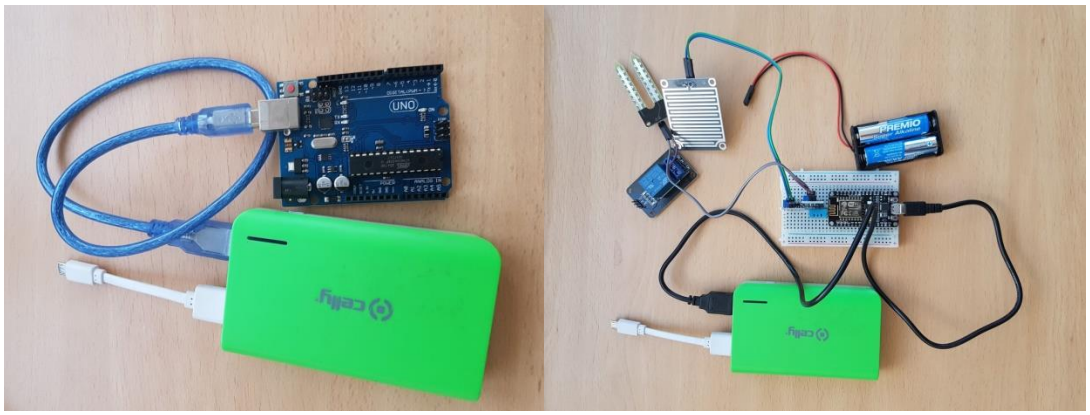
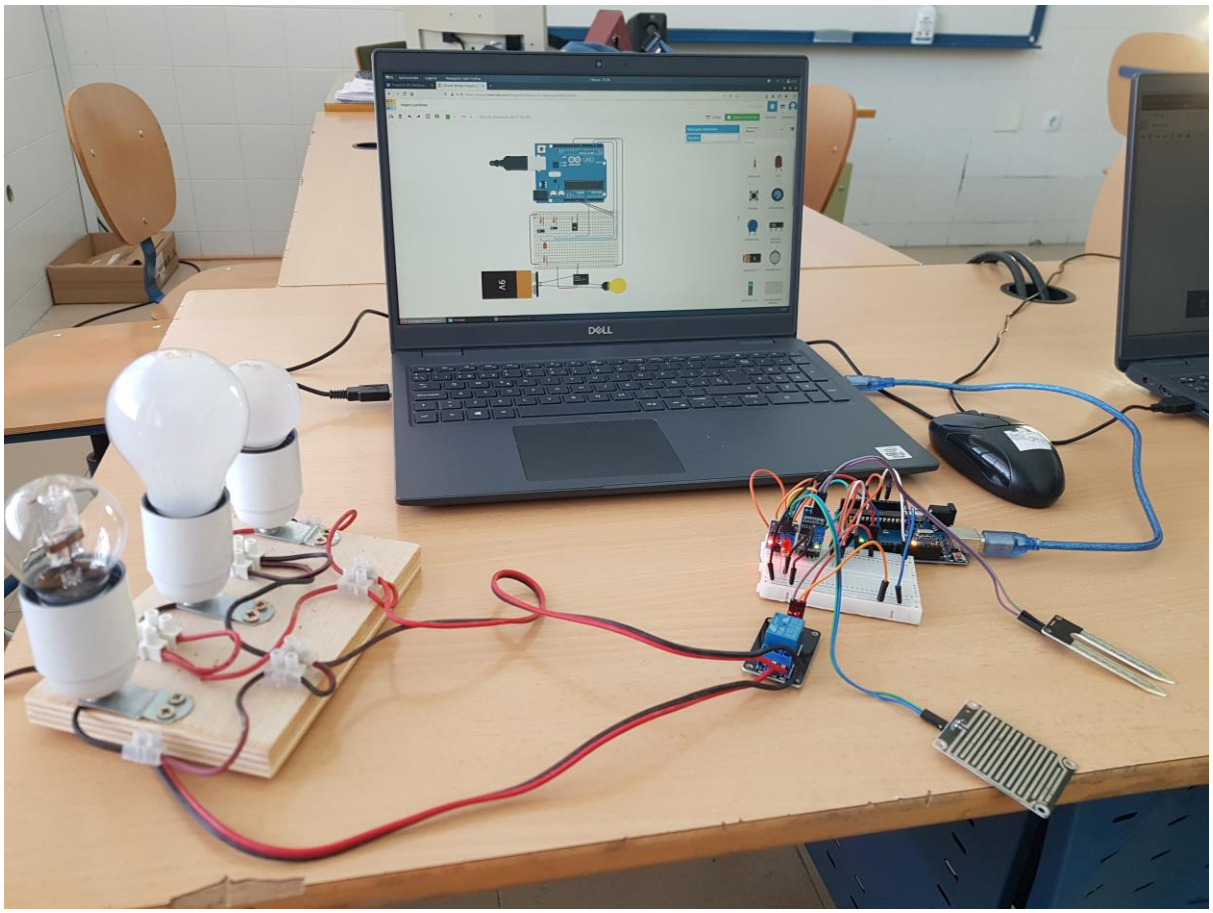
Con lenguaje de programación C++

```
// C++ code
//
/*
  Programador de riegos y jardines
*/

float Temperatura = 0;

int Tiempo = 0;

void setup()
{
  pinMode(A0, INPUT);
  pinMode(2, INPUT);
  Serial.begin(9600);

  pinMode(3, INPUT);
  pinMode(10, OUTPUT);
}

void loop()
{
  Temperatura = analogRead(A0);
  Temperatura =(Temperatura / 1023)*5;
  Temperatura = (Temperatura-0.5)*100;
  Serial.println(digitalRead(2));
  Serial.println(digitalRead(3));
  // Con lluvia
  if (digitalRead(3) == 1) {
    digitalWrite(10, LOW);
  }
  // Ausencia de lluvia y humedad de suelo
  if (digitalRead(3) == 0 && digitalRead(2) == 0) {
    digitalWrite(10, HIGH);
  }
  // Sin lluvia y con humedad de suelo
  while (digitalRead(3) == 0 && digitalRead(2) == 1) {
    digitalWrite(10, LOW);
    if (Temperatura < 15) {
      Tiempo = 72;
```

```
      }
      if (Temperatura >= 15 && Temperatura < 20) {
        Tiempo = 48;
      }
      if (Temperatura >= 20 && Temperatura < 30) {
        Tiempo = 24;
      }
      if (Temperatura >= 30 && Temperatura < 35) {
        Tiempo = 12;
      }
      if (Temperatura >= 35) {
        Tiempo = 6;
      }
      delay(1000 * Tiempo); // Wait for 1000 * Tiempo millisecond(s)
      Serial.println(Temperatura);
      Serial.println(Tiempo);
    }
  }
```

*Montaje del Proyecto de Programador de riegos para agricultura y jardines.*

Una vez simulado el proyecto, realizaremos el montaje del mismo en la placa de pruebas protoboard y cargaremos en la placa Arduino el código a través del software libre de arduino.



Finalizando con el montaje de componentes y sensores.