



Guía para el Profesor

Tecnologías de la información II / Information technologies II

BSTI1002/BTTI1002/BSTI1004/BTTI1005

Nivel Bachillerato

Índice

Información general del curso	3
Competencia del curso	3
Utiliza lenguajes de programación y uso de lógica computacional para la solución de problemas.....	3
Metodología.....	3
Evaluación	3
Bibliografía y recursos especiales	3
Contenido del curso (temas).....	4
Feria tecnológica.....	8
Uso de rúbricas	9
Tips importantes	9
Comprobación de lectura.	10
Solución a los retos	11

Información general del curso

- Nivel: Bachillerato
- Plan académico: 2013

Competencia del curso

Utiliza lenguajes de programación y uso de lógica computacional para la solución de problemas.

Metodología

Características del curso

- El curso está diseñado para desarrollar una competencia.
- Los contenidos están divididos en tres episodios, cada uno de los cuales se divide en cinco temas.
- Hay 13 retos a lo largo del curso, cada uno de los cuales incrementa en dificultad, conforme se avanza en el curso. Al terminar estos retos, se realiza un reto final, en el que se ponen en práctica todos los conocimientos adquiridos en el curso.
- La evaluación del curso está integrada por 13 retos, un reto final y la participación en la Feria Tecnológica.

Modelo didáctico

- Utiliza las técnicas de aula invertida y retos, por lo que el alumno revisa la teoría y los recursos antes de ir a clase; durante la clase, con la ayuda del profesor, los alumnos desarrollan los retos.
- Los retos se llevan a cabo de acuerdo a lo que se especifique, cada uno de ellos es realizado mediante el mismo software y una variedad de periféricos físicos, que acumulan dificultad conforme se avanza; al llegar al reto final, se busca el uso completo del software para la realización del mismo.

Evaluación

No. Actividad	Descripción del evaluable	Ponderación
1	Reto 1	5
2	Reto 2	5
3	Reto 3	5
4	Reto 4	5
5	Reto 5	5
6	Reto 6	5
7	Reto 7	5
8	Reto 8	5
9	Reto 9	6
10	Reto 10	6
11	Reto 11	6
12	Reto 12	6
13	Reto 13	6
14	Reto final	15
15	Feria tecnológica	15
Total		100

Bibliografía y recursos especiales

Libro de texto

GUÍA PARA EL PROFESOR

Moreno, A., y Córcoles, S. (2018). *Arduino. Edición 2018 Curso práctico*. España: RA-MA. ISBN: 978-84-9964-782-1

Disponible en formato eBook en https://www.ra-ma.es/libro/arduino-edicion-2018-curso-practico_83631/edicion/ebook-75718/

Libro de apoyo

Jayakumar, M. (2016). *Arduino and Android using MIT app inventor 2.0*. Estados Unidos: CreateSpace Independent Publishing Platform.

ISBN: 1523486805, 9781523486809

Requerimientos especiales

Laboratorio

Cómputo

Software

- Raptor.
- Arduino Software (IDE).
- Autodesk Tinkercad.

En la sección Bibliografía del curso hay un enlace que dirige a la lista completa de materiales. Para la adquisición del kit, favor de ponerse en contacto con el director de nivel de su campus.

Contenido del curso (temas)

Módulo 1. Introducción a la programación de periféricos.

Tema 1. Algoritmos, diagramas de flujo.

- En esta sección el estudiante aprenderá qué es un algoritmo y por qué es importante realizarlo antes de iniciar con la programación, comprenderá el lenguaje por medio de diagramas y prioridades en la ejecución de tareas al momento de que un ordenador o dispositivo de control ejecuta una tarea programada.
- Es importante que el estudiante comprenda los pasos para realizar un algoritmo, es uno de los temas más importantes ya que en esta primera sesión tendrá las bases para poder realizar programas con lógica exento de dudas.

Tema 2. Periféricos y primeros pasos con Arduino.

- Durante este tema el estudiante comprenderá qué es un periférico y cómo le pueden servir cuando necesite realizar dispositivos eléctricos. También iniciará con el uso de la tarjeta Arduino, para lo cual se explicará cómo está compuesta dicha tarjeta, dónde se descarga el software para la programación de la tarjeta y cuáles son los componentes del entorno del software.
- Es importante que se enseñe al estudiante cómo saber que en qué puerto se encuentra conectado la tarjeta Arduino cuando se conecte a la Computadora, ya que, al momento de descargar algún problema, la computadora informará de algún error. El siguiente video muestra cómo activar y cambiar de nombre al puerto COM <https://youtu.be/81wN4tojrMk>.
- Ver el documento “Anexos Soluciones de Reto” para encontrar el circuito que se tiene que implementar, así como el programa.

Tema 3. Manejo de datos digitales: escritura, lectura de puertos (Input & Output), y retardos.

- En esta sesión el estudiante aprenderá la diferencia entre entradas y salidas digitales en la tarjeta Arduino, las sentencias usadas para dar de alta salida o entrada cada pin de la tarjeta, así como la sentencia usada para la lectura de un pin de

GUÍA PARA EL PROFESOR

entrada o la escritura de un pin de salida. Otra enseñanza que el estudiante comprenderá es el uso de los retardos o de pausas de tiempo en un programa y cómo se pueden agregar.

- El instructor debe hacer énfasis en cuándo hay que usar una entrada y cuándo una salida, dando ejemplos prácticos de la vida cotidiana para un mayor entendimiento.
- Ver el documento “Anexos Soluciones de Reto” para encontrar el circuito que se tiene que implementar, así como el programa.

Tema 4. Tipos de datos, constantes y variables.

- Esta sesión es dedicada a que el estudiante comprenda los diferentes tipos de datos que se pueden tener en un programa para la tarjeta Arduino, cuándo se utiliza cada tipo de dato, en qué parte del programa se declara el tipo de dato y dónde se puede utilizar durante el programa.
- Es importante que el facilitador se asegure que los estudiantes entiendan cada tipo de dato y cuándo se deben de utilizar, así como la diferencia entre un dato int, float, char y string.
- Ver el documento “Anexos Soluciones de Reto” para encontrar el circuito que se tiene que implementar, así como el programa.

Tema 5. Manejo de sensores.

- Durante este tema se inicia con el uso de sensores, especialmente los digitales. Este tema funge como una introducción a dichos sensores. También se usa el condicional IF para evaluar condiciones y se retoma lo aprendido en el tema 3 sobre entradas y salidas digitales.
- El profesor debe poner especial atención cuando los estudiantes estén programando para que usen las sentencias adecuadas para dar de alta los pines usados del Arduino, ya sea para configurarlos como entrada o como salida. En el programa también se tiene que utilizar sentencias de lectura de pines y, finalmente, que comprende el primer condicional IF, ya que es uno de los más usados. Cuando se explique el uso del condicional es necesario darles ejemplos de uso cotidiano de condicionales.
- Ver el documento “Anexos Soluciones de Reto” para encontrar el circuito que se tiene que implementar, así como el programa.

Módulo 2. Muestreo de datos e introducción a sensores.

Tema 6. Algoritmos, diagramas de flujo.

- En esta sesión los estudiantes comprenderán las diferencias entre el uso de sensores digitales y sensores análogos, aprenderán algunos de los diferentes sensores análogos que se utilizan con la tarjeta Arduino y en qué aplicaciones se deben de usar. También deben de entender cómo se registra el valor de la lectura cuando se decide usar la sentencia de lectura análoga `analogRead()`; Es necesario el uso de condicionales para toma de decisiones, así que el estudiante tiene que recordar los conceptos aprendidos en la sesión anterior.
- Hay que poner especial atención en que los estudiantes puedan tomar lecturas de los sensores análogos utilizados, así como del correcto uso del condicional IF.
- Ver el documento “Anexos Soluciones de Reto” para encontrar el circuito que se tiene que implementar, así como el programa.

Tema 7. Estructuras de control.

- En el tema 5 los estudiantes ya aprendieron a usar el condicional IF, en este tema continuaran con el aprendizaje y aplicación del uso de condicionales o estructuras de control. Es necesario que se entienda cuándo es necesario utilizar cada tipo de estructura de control y cómo se deben de programar o usar la sintaxis correcta.
- Hay que asegurarse de que los estudiantes entiendan la sintaxis de cada estructura de control, de qué debe de ir en cada parte, para que puedan hacer un correcto uso de estos y tengan mayor claridad de cuándo se debe de utilizar cada uno.
- Ver el documento “Anexos Soluciones de Reto” para encontrar el circuito que se tiene que implementar, así como el programa.

Tema 8. Interfaz humano-máquina.

- En esta clase el estudiante conocerá el uso de pantallas LCD como interfaz de la tarjeta Arduino, cómo se realizan las conexiones con la tarjeta, así como también el uso de las librerías, específicamente la librería LiquidCrystal.h.
- El profesor tiene que poner atención de que los estudiantes conecten correctamente la tarjeta con la pantalla LCD. Posteriormente que se realice correctamente el uso de las sentencias, que se declare la librería, la variable global y se especifiquen las dimensiones de la pantalla a usar.
- Ver el documento “Anexos Soluciones de Reto” para encontrar el circuito que se tiene que implementar, así como el programa.

Tema 9. Manejo de motores.

- En temas anteriores los estudiantes han aprendido el uso de salidas digitales, por lo que en este tema se utilizarán para controlar el manejo de motor de DC mediante el Driver L298, para lo cual es necesario que entiendan el funcionamiento interno del motor y cómo es posible el cambio de dirección. Después, es necesario explicar el uso del Driver en cada uno de los pines.
- Asegúrate de que el estudiante entienda qué activa cada pin del Driver y por qué es posible el cambio de la dirección del motor en el uso del Driver, cómo sucede este cambio de dirección o qué es lo que lo hace posible.
- Ver el documento “Anexos Soluciones de Reto” para encontrar el circuito que se tiene que implementar, así como el programa.

Tema 10. Servomotores y Motores a Pasos.

- Continuando con lo visto en el tema 8 sobre el uso de las librerías en Arduino, en este tema el alumno aprenderá a utilizar las librerías para controlar servomotores y motores a pasos. El estudiante debe entender los principios por los que funciona un servo y un servomotor, así como las diferencias de cada uno de ellos y las posibles aplicaciones, y la importancia de la utilización de cada uno de ellos.
- En este tema el estudiante debe de utilizar correctamente las sentencias de la librería para poder controlar un servomotor o un motor a pasos.
- Ver el documento “Anexos Soluciones de Reto” para encontrar el circuito que se tiene que implementar, así como el programa.

Módulo 3. Robótica.

Tema 11. Control de motores de acuerdo con lectura de sensores.

- En este tema el estudiante se pone a prueba para que pueda aplicar lo aprendido en temas anteriores, ya que tiene que tomar lecturas de sensores, sean digitales o análogas y, con base en esas lecturas, utilizar estructuras de control para poder poner en diferente movimiento los motores de DC, dependiendo de las condiciones que se programen.
- Cuando el estudiante este programando, asegúrate de que declare los pines a usar.
- Cuando el estudiante utilice las estructuras de control, este haga una comparación correcta con la lectura, ya que es importante que cuando se están utilizando entradas análogas la lectura vaya de 0 a 1023.
- Ya que se tiene el programa listo y el estudiante esté realizando las conexiones con la tarjeta, es importante que estos se coloquen en los pines declarados correctamente, ya que en ocasiones se suelen tener algunos errores por cambio de pin.
- Ver el documento “Anexos Soluciones de Reto” para encontrar el circuito que se tiene que implementar, así como el programa.

Tema 12. Toma de decisiones con base en lectura de datos externos.

- En esta sesión se retoma lo visto en el tema 7 sobre las estructuras de control. Durante este tema es necesario explicar nuevamente el uso del SWITCH, ya que dicha sentencia será utilizada en el Reto Final.

- Poner atención en que el estudiante entienda el uso de la condición y la activación de cada CASE.
- Ver el documento “Anexos Soluciones de Reto” para encontrar el circuito que se tiene que implementar, así como el programa en la IDE como en APP Inventor.

Tema 13. Robótica autómata.

- En esta clase los estudiantes aprenderán las diferentes clasificaciones de la robótica. Es recomendable buscar aplicaciones de cada una de ellas para despertar el interés de la investigación del estudiante, algunos ejemplos se muestran en el siguiente video: <https://youtu.be/7wtBOZen5fo>.

Módulo 4. Comunicación.

Tema 14. Diferentes tipos de tecnologías de comunicación.

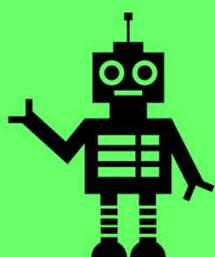
- En este tema el estudiante conocerá las diferentes tecnologías de comunicación que usamos comúnmente hoy en día, y el uso del bluetooth con la tarjeta Arduino. Es de vital importancia que el estudiante comprenda los pines de comunicación de la tarjeta (TX y RX) y qué sucede cuando se tienen conectados por medio de Bluetooth y se pretende descargar un programa de la PC a la tarjeta. Se debe de explicar al estudiante que esta comunicación mediante los 2 pines se da únicamente en una dirección con cada pin. El TX escribe, mientras que el RX se encarga de la lectura que dicta el dispositivo receptor, y al igual que la tarjeta Arduino cuenta con los pines TX y RX, también contará con éstos el dispositivo con el que se pretende entablar la comunicación.
- Ya que el estudiante tenga claro el uso del Bluetooth con Arduino, es necesario que aprenda a utilizarlo mediante el diseño de una App en MIT App Inventor. Se requerirá una App que apague y encienda las luces de 2 habitaciones.
- Ver el documento “Anexos Soluciones de Reto” para encontrar el circuito que se tiene que implementar, así como el programa en la IDE como en APP Inventor.

Tema 15. Uso de Apps y dispositivos móviles como control.

- En este último tema se explicará el entorno de la plataforma y cómo se realiza la programación, es por eso que el facilitador tiene que estar familiarizado con dicha plataforma y su forma de programación, para que así pueda resolver dudas al estudiante de manera clara y práctica.
- Se recomienda al facilitador realizar la lectura de “App Inventor Built-in Blocks” y “Designer and Blocks Editor” que encontrará en la plataforma de MIT App Inventor. El reto final consiste en una integración de conocimientos de temas anteriores, pero que sea controlado mediante una APP utilizando comunicación Bluetooth, es por eso que el profesor debe de conocer muy bien la forma de programación de bloques que utiliza App Inventor, ya que así podrá identificar con mayor facilidad errores comunes que están realizando los estudiantes.
- Ver el documento “Anexos Soluciones de Reto” para encontrar el circuito que se tiene que implementar, así como el programa en la IDE como en APP Inventor.

Torneo de Robótica de Preparatoria

Enero - Mayo 2020



Invitamos a todos los estudiantes de Tecnologías de Información II (TI2) a que participen, como parte de su evidencia final de la materia, en el Torneo de Robótica de Preparatoria. Tu curso de TI2 está diseñado para darte todas las herramientas necesarias, para que al final de este puedas realizar un robot competitivo, que tenga la capacidad de moverse en cualquier dirección, autónomamente o de forma teleoperada, y lograr todo tipo de retos.

MARIO KART BATTLE

Este año la temática del concurso de robótica es MARIO KART BATTLE:

- ✓ Un robot por cada tres estudiantes.
- ✓ Los robots harán alianzas con otros robots.
- ✓ Pasarán de ronda los que logren dejar sin globos al equipo contrario.



Habrán premios por:

- 1º, 2º y 3er lugar en la copa del torneo.
- Espíritu de equipo.
- Diseño innovador (se evaluará por robot en un stand).

Espera la convocatoria definitiva que se publicará el 27 de Enero de 2020

Uso de rúbricas

El reto final del curso tiene asignada una rúbrica con la cual es obligatorio que se califique, esto es muy importante para nuestro modelo de competencias ya que es la forma en la que medimos el desarrollo de las competencias en nuestros alumnos.

Es importante evaluar con la rúbrica que aparece en el apartado de la evidencia final ya que se les estará auditando constantemente su realización efectiva.

Video disponible para calificar con rúbricas en:

- ¿Cómo busco una rúbrica?: <https://youtu.be/QgDKeZv9tAI>
- ¿Cómo califico con una rúbrica?: <https://youtu.be/mAbIsLAgIp4>

Tips importantes

Los videotutoriales forman parte de la explicación de cada tema, es muy importante que todos los videos sean revisados por los profesores y alumnos.

Para visualizar correctamente los videotutoriales que se incluyen en cada tema hay que presionar la tecla Ctrl y hacer clic sobre el enlace.

Durante el periodo del curso habrá algunos alumnos que asistan al Study Tour en Seattle. Para dichos alumnos se ha elaborado un reto especial. Este reto va a sustituir a aquel que se realice durante la semana en que se lleve a cabo el viaje.

- **Material de capacitación en la plataforma tecnológica Canvas:**
 - Tutorial digital para profesores: <https://bit.ly/2SbMaNK>
 - Tutorial digital para alumnos: <https://bit.ly/35IBnPG>
- **¿En dónde o a quién reporto un error detectado en el contenido del curso?**

Lo puedes reportar a la cuenta atencioncursos@servicios.tecmilenio.mx pero te pedimos que también reportes sugerencias para el contenido y actividades del curso.

- **¿Quién me informa de la cantidad de sesiones y tiempo de cada sesión en las semanas?**

El coordinador docente te debe de proporcionar esta información.

- **¿En qué semanas se aplican los exámenes parciales y el examen final?**

En este curso no hay exámenes, la evaluación se basa en los retos y la feria tecnológica.

- **¿Tengo que capturar las calificaciones en Banner y en la plataforma educativa?**

Sí, es importante que captures calificaciones en la plataforma para que los alumnos estén informados de su avance y reciban retroalimentación de parte tuya de todo lo que realizan en el curso. En banner es el registro oficial de las calificaciones de los alumnos.

Comprobación de lectura.

Responde las siguientes preguntas en un documento y hazlo llegar con tus datos a través del siguiente enlace.

<https://utmedu.sharepoint.com/sites/pa/Lists/Capacitaciones/Item/newifs.aspx>

1. Menciona el nombre de la tarjeta que se utilizará para este curso.
2. ¿Qué recomendaciones hay que indicar a los alumnos para la visualización de los videotutoriales?
3. ¿Cuántos exámenes se incluyen en la evaluación del curso?
4. ¿En qué consiste el reto final?
5. ¿En qué consiste la feria tecnológica?

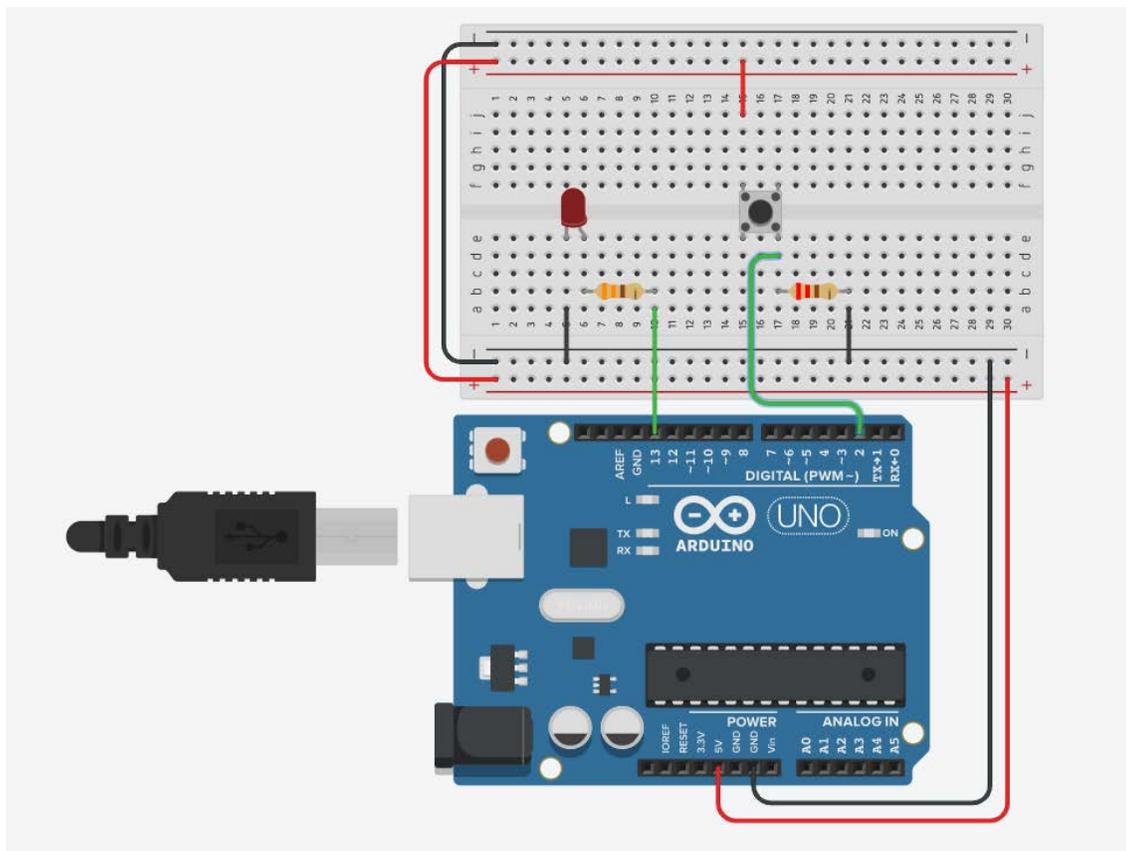
Solución a los retos

Reto 2

```
void setup()
{
  Serial.begin(9600);
  // Inicializa el puerto serie y lo prepara para comunicarse
}

void loop()
{
  Serial.println("Bienvenido a Tecnologías de la Información II");
  // Imprime mediante el Puerto Serie y baja una línea
  delay(10000);
  // Espera 10 segundos
}
```

Reto 3



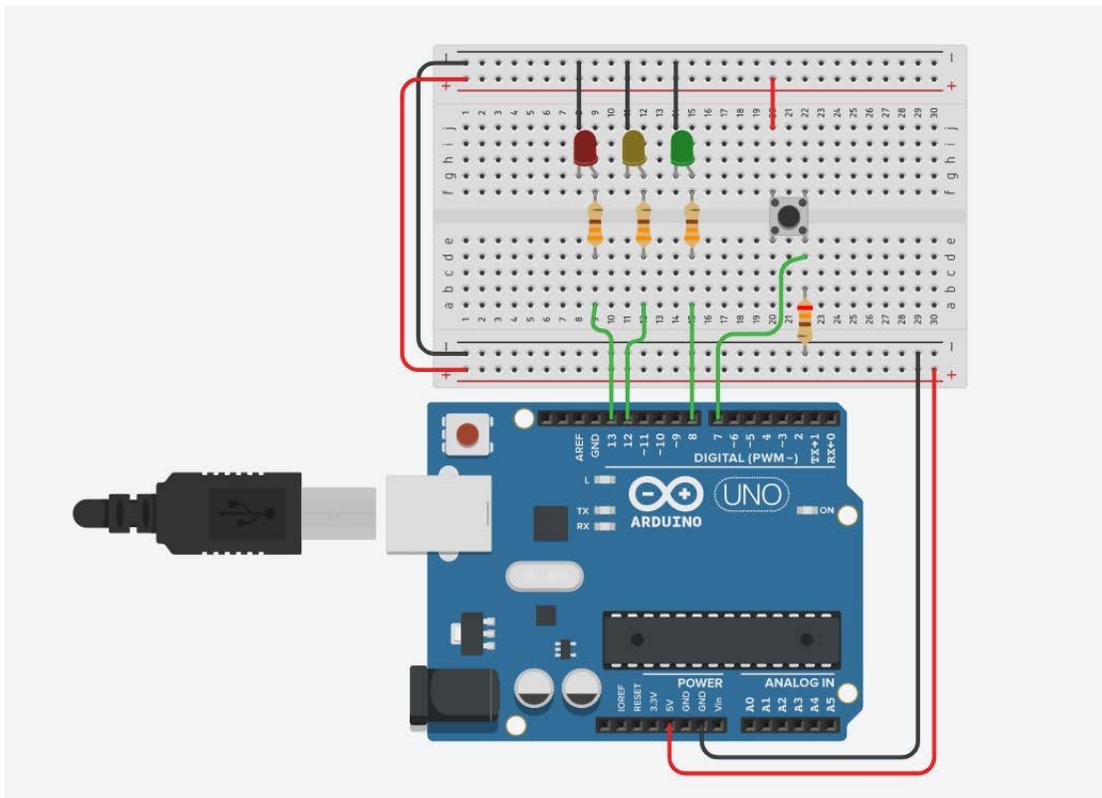
```

void setup() {
    // initialize the LED pin as an output:
    pinMode(13, OUTPUT);
    // initialize the pushbutton pin as an input:
    pinMode(2, INPUT);
}

void loop() {
    // check if the pushbutton is pressed. If it is, the buttonState is HIGH:
    if (digitalRead(2) == HIGH) {
        // turn LED on:
        digitalWrite(13, HIGH);
        delay (10000);
        digitalWrite(13, LOW);
    }
}

```

Reto 4



```

// declarando parametros
int led_verde = 8; //LED 1 al pin digital 8
int led_amarillo = 12; //LED2 al pin digital 12
int led_rojo = 13; //LED 3 al pin digital 13
int push = 7; // pin de entrada botón
int boton = 0; //valor se emplea para almacenar el estado
// del botón

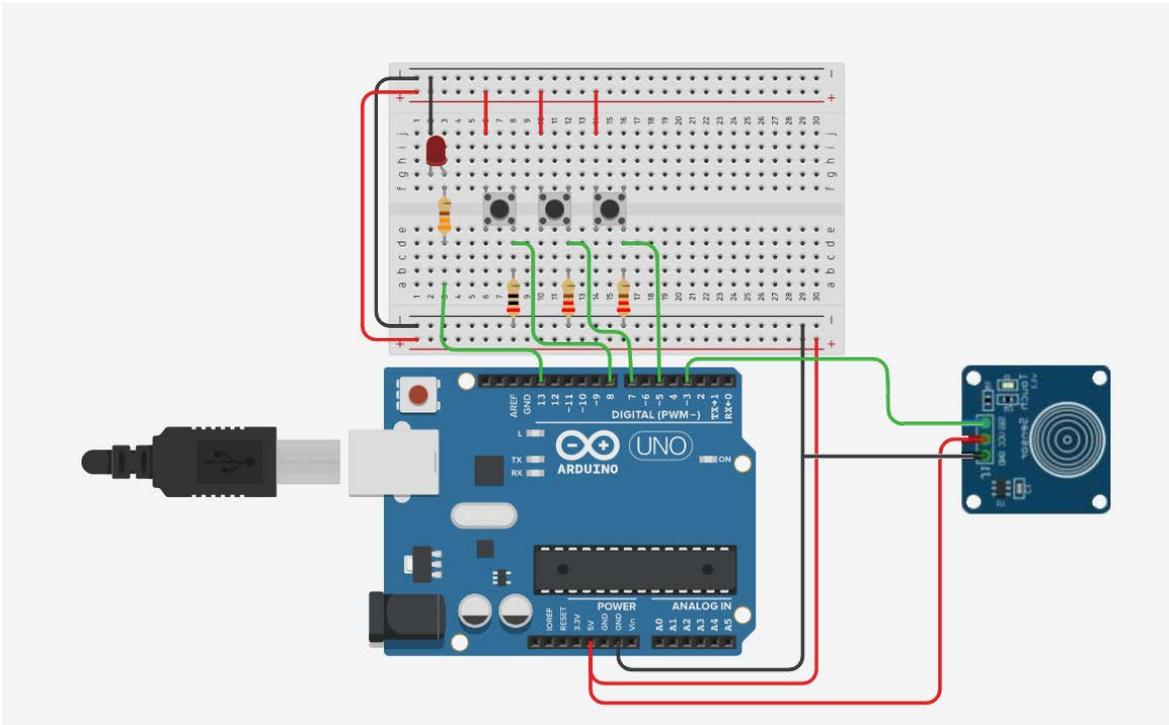
void setup() {
    // se indica que cada pin es de salida OUTPUT.
    pinMode(led_verde, OUTPUT);
    pinMode(led_amarillo, OUTPUT);
    pinMode(led_rojo, OUTPUT);
    pinMode(push, INPUT);
}

// lazo a ejecutar continuamente una vez cargado el código en el arduino

void loop(){
    boton = digitalRead(push); // lee el estado del botón
    // chequear si el boton esta presionado o no
    if (boton == HIGH) //Instrucción usada para que el //Arduino reconozca el pulso del botón
    en el momento justo, y evitar un mal //funcionamiento, debido a la rapidez con que trabaja
    el Arduino.
    {
        digitalWrite(led_verde,HIGH); // encender LED 1
        delay(5000); // mantener por 5 segundos
        digitalWrite(led_verde,LOW); // apagar LED 1
        digitalWrite(led_amarillo,HIGH); // encender LED 2
        delay(2000); // mantener por 2 segundos
        digitalWrite(led_amarillo,LOW); // apagar LED 2
        digitalWrite(led_rojo,HIGH); // encender LED 3
        delay(7000); // mantener por 7 segundos
        digitalWrite(led_rojo,LOW); // apagar LED 3
    }
}

```

Reto 5



```
// Contraseña Boton 1, 3 y Sensor
```

```
int Boton_1 = 8; // pin de entrada botón
int Boton_2 = 7; // pin de entrada botón
int Boton_3 = 5; // pin de entrada botón
int Sensor = 3; // pin de entrada Sensor
int led_rojo = 13; //pin digital 13 de salida para led
int vboton_1 = 0; //valor se emplea para almacenar el estado del botón
int vboton_2 = 0; //valor se emplea para almacenar el estado del botón
int vboton_3 = 0; //valor se emplea para almacenar el estado del botón
int vSensor = 0;
```

```
void setup() {
    // se indica que cada pin es de salidad OUTPUT o entrada INPUT.
    pinMode(Boton_1, INPUT);
    pinMode(Boton_2, INPUT);
    pinMode(Boton_3, INPUT);
```

```

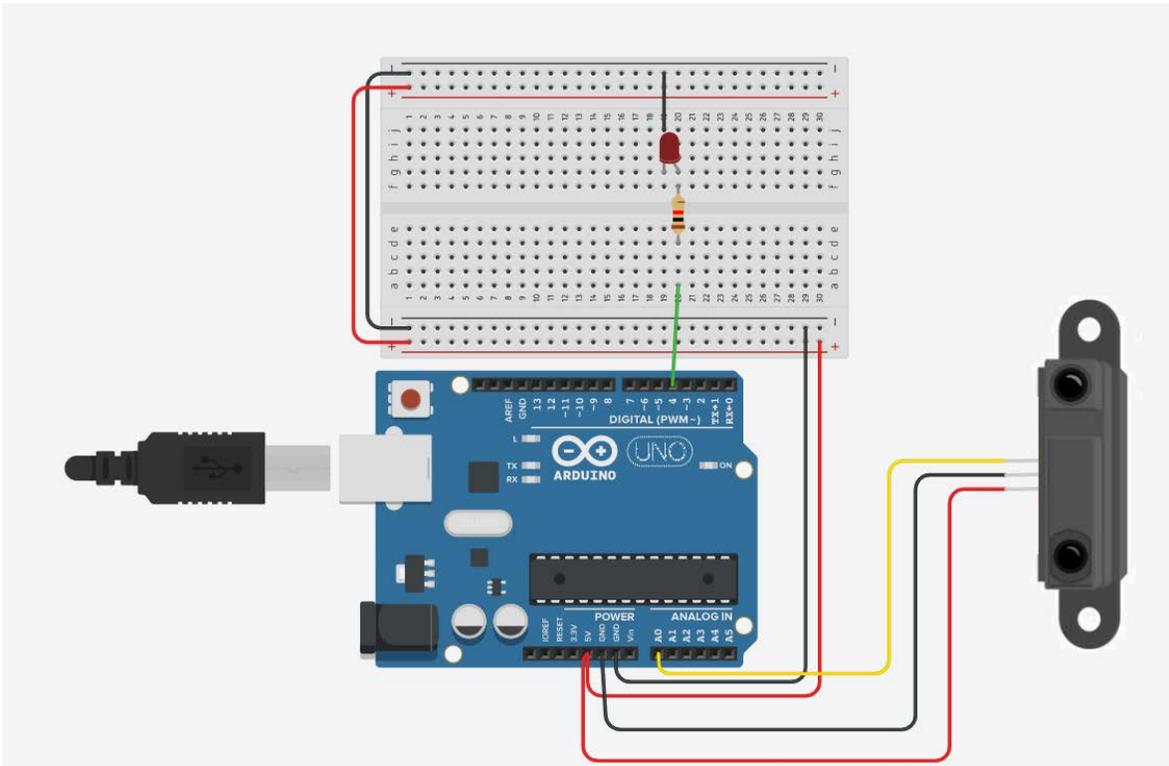
pinMode(Sensor, INPUT);
pinMode(led_rojo, OUTPUT);
Serial.begin(9600);
}

// lazo a ejecutar continuamente una vez cargado el código en el Arduino

void loop(){
    vboton_1 = digitalRead(Boton_1); // lee el estado del botón
    vboton_2 = digitalRead(Boton_2); // lee el estado del botón
    vboton_3 = digitalRead(Boton_3); // lee el estado del botón
    vSensor = digitalRead(Sensor); // lee el estado del sensor
    // chequear si el boton 1 esta presionado o no
    if (vboton_1 == HIGH) //Instrucción usada para que el //Arduino reconozca el pulso del
    botón en el momento justo, y evitar un mal //funcionamiento, debido a la rapidez con que
    trabaja el Arduino.
    {
        Serial.println("Boton1");
        if (vboton_3 == HIGH) //Instrucción usada para que el //Arduino reconozca el
        pulso del botón en el momento justo, y evitar un mal //funcionamiento, debido a la
        rapidez con que trabaja el Arduino.
        {
            Serial.println("Boton3");
            if (vSensor == HIGH) //Instrucción usada para que el //Arduino reconozca
            el pulso del Sensor en el momento justo, y evitar un mal //funcionamiento,
            debido a la rapidez con que trabaja el Arduino.
            {
                Serial.println("Sensor");
                digitalWrite(led_rojo,HIGH); // encender LED 3
                delay(7000); // mantener por 7 segundos
                digitalWrite(led_rojo,LOW); // apagar LED 3
            }
        }
    }
}
}

```

Reto 6



```
#define sensor A0 // Pin analogo del sensor Sharp
int Led_alto = 4;

void setup() {
    Serial.begin(9600);
    pinMode(Led_alto, OUTPUT);
}

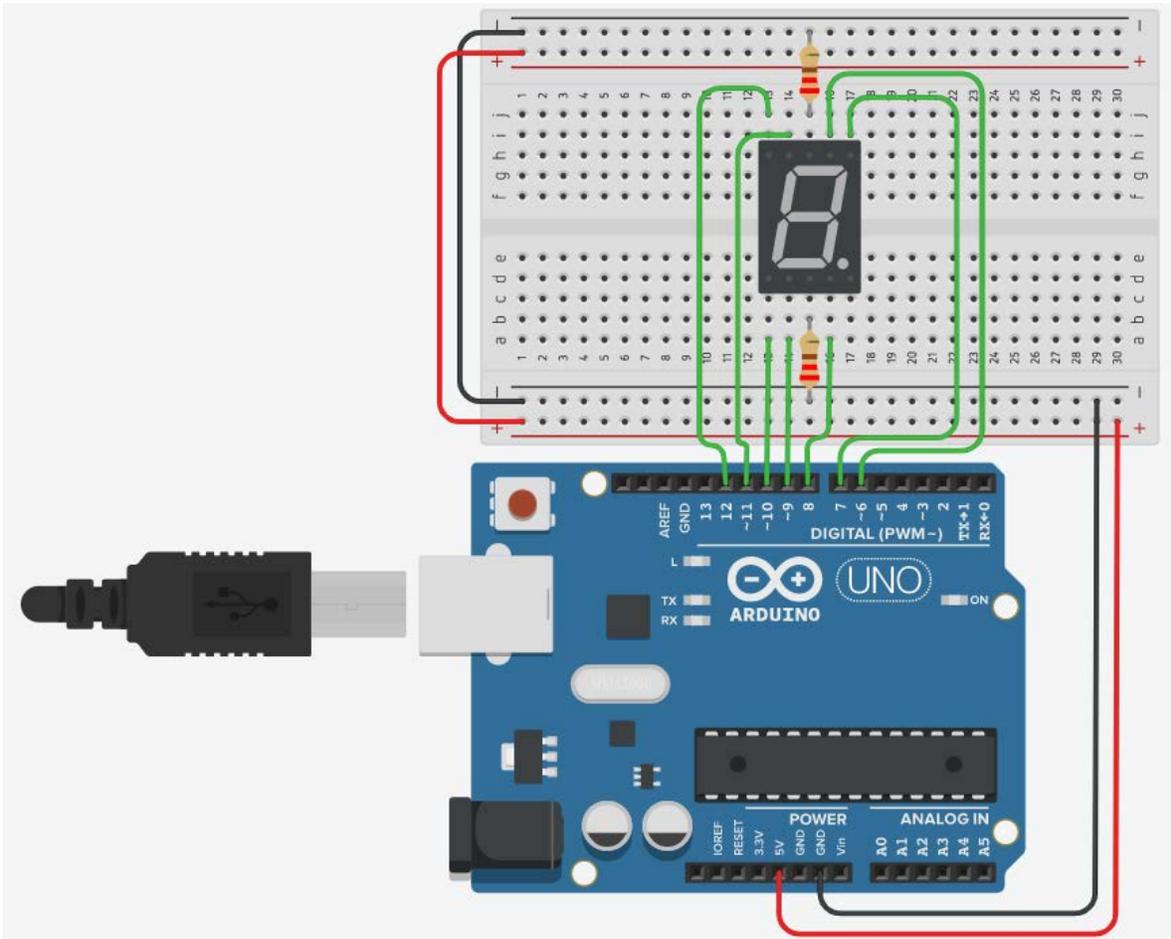
void loop() {
    // put your main code here, to run repeatedly:
    float volts = analogRead(sensor)*0.0048828125; //Valor del sensor por voltaje (5/1024)
    int distancia;
    distancia = 13*pow(volts,-1); // Datos de la hoja de especificación del sensor
    if (12 <= distancia >= 15) {
        Serial.println(distancia);
        digitalWrite(Led_alto, HIGH);
    }
}
```

```

else
{
    digitalWrite(Led_alto, LOW);
}
delay (500);
}

```

Reto 7



```

int a = 6; //display
int b = 7;
int c = 8;
int d = 9;
int e = 10;
int f = 11;

```

```
int g = 12;
```

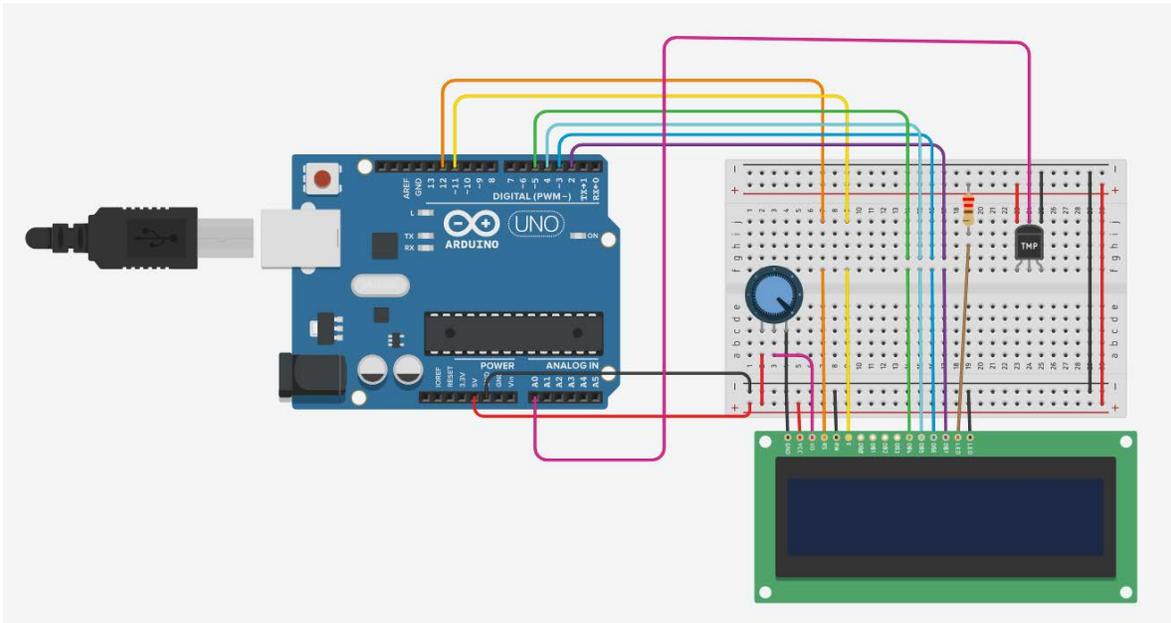
```
void setup() {  
    pinMode(a, OUTPUT); //display  
    pinMode(b, OUTPUT);  
    pinMode(c, OUTPUT);  
    pinMode(d, OUTPUT);  
    pinMode(e, OUTPUT);  
    pinMode(f, OUTPUT);  
    pinMode(g, OUTPUT);  
}
```

```
void loop() {  
    // Cero  
    digitalWrite(a,HIGH);  
    digitalWrite(b,HIGH);  
    digitalWrite(c,HIGH);  
    digitalWrite(d,HIGH);  
    digitalWrite(e,HIGH);  
    digitalWrite(f,HIGH);  
    digitalWrite(g,LOW);  
    delay(2000);  
    // UNO  
    digitalWrite(b,HIGH);  
    digitalWrite(c,HIGH);  
    digitalWrite(a,LOW);    digitalWrite(d,LOW);  
    digitalWrite(e,LOW);    digitalWrite(f,LOW);  
    digitalWrite(g,LOW);  
    delay(2000);  
    //DOS  
    digitalWrite(a,HIGH);  
    digitalWrite(b,HIGH);  
    digitalWrite(d,HIGH);  
    digitalWrite(e,HIGH);
```

```
digitalWrite(g,HIGH);
digitalWrite(c,LOW);
digitalWrite(f,LOW);
delay(2000);
//TRES
digitalWrite(a,HIGH);
digitalWrite(b,HIGH);
digitalWrite(c,HIGH);
digitalWrite(d,HIGH);
digitalWrite(f,LOW);
digitalWrite(e,LOW);
digitalWrite(g,HIGH);
delay(2000);
//CUATRO
digitalWrite(b,HIGH);
digitalWrite(c,HIGH);
digitalWrite(g,HIGH);
digitalWrite(f,HIGH);
digitalWrite(a,LOW);
digitalWrite(d,LOW);
digitalWrite(e,LOW);
delay (2000);
//CINCO
digitalWrite(a,HIGH);
digitalWrite(c,HIGH);
digitalWrite(d,HIGH);
digitalWrite(g,HIGH);
digitalWrite(f,HIGH);
digitalWrite(b,LOW);
digitalWrite(e,LOW);
delay(2000);
//SEIS
digitalWrite(a,HIGH);
digitalWrite(c,HIGH);
```

```
digitalWrite(d,HIGH);
digitalWrite(g,HIGH);
digitalWrite(b,LOW);
digitalWrite(e,HIGH);
digitalWrite(f,HIGH);
delay(2000);
//SIETE
digitalWrite(a,HIGH);
digitalWrite(c,HIGH);
digitalWrite(g,LOW);
digitalWrite(b,HIGH);
digitalWrite(d,LOW);
digitalWrite(e,LOW);
digitalWrite(f,LOW);
delay(2000);
//OCHO
digitalWrite(a,HIGH);
digitalWrite(c,HIGH);
digitalWrite(d,HIGH);
digitalWrite(e,HIGH);
digitalWrite(b,HIGH);
digitalWrite(f,HIGH);
digitalWrite(g,HIGH);
delay(2000);
//NUEVE
digitalWrite(a,HIGH);
digitalWrite(c,HIGH);
digitalWrite(f,HIGH);
digitalWrite(g,HIGH);
digitalWrite(b,HIGH);
digitalWrite(d,HIGH);
digitalWrite(e,LOW);
delay(2000);
}
```

Reto 8



```
#include <LiquidCrystal.h> //Libreria del LCD
float TempC;
const int Temp=A0;

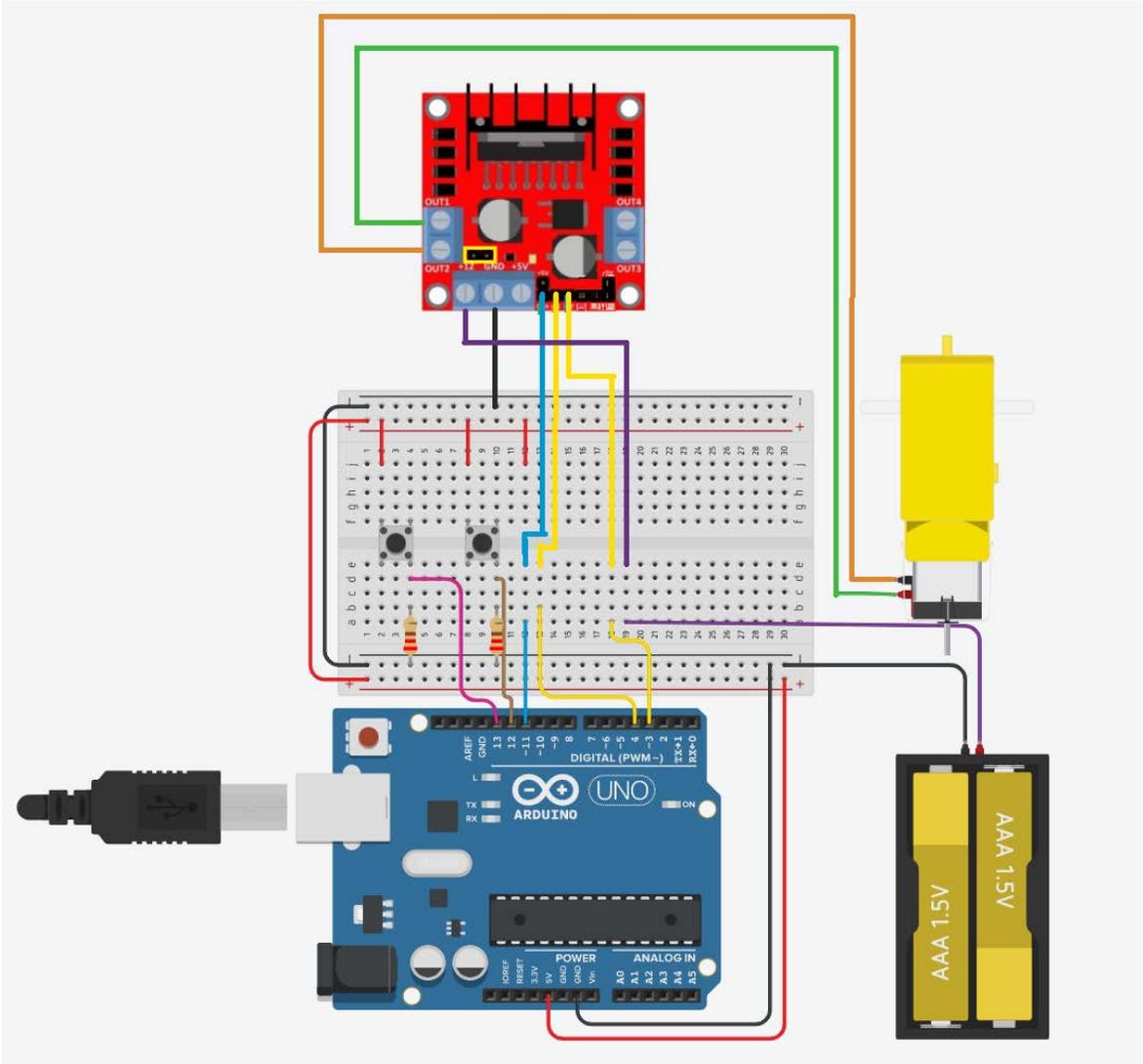
// Inicio de la librería con los pines de la pantalla
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() {
    // Número de columnas y renglones de la pantalla
    lcd.begin(16, 2);
    // Imprime mensaje de inicio
    lcd.print("hello, world!");
}

void loop() {
    TempC = analogRead(Temp); //Lee el pin análogo de temperatura
    TempC = (0.5*TempC*100.0)/1024.0; //Formula para calcular la temperatura
    // Inicia el cursor en la columna 0 y la línea 1
    lcd.setCursor(0, 1);
    // Imprime la temperatura del sensor
```

```
lcd.print("La T es:");  
lcd.print(TempC);  
lcd.print (" C");  
delay(5000); //Espera 5 segundos para calcular nuevamente la temperatura  
}
```

Reto 9



```
const int ENABLE = 11;  
const int INP1 = 4;  
const int INP2 = 3;  
const int BOTON_A = 13;  
const int BOTON_B = 12;
```

GUÍA PARA EL PROFESOR

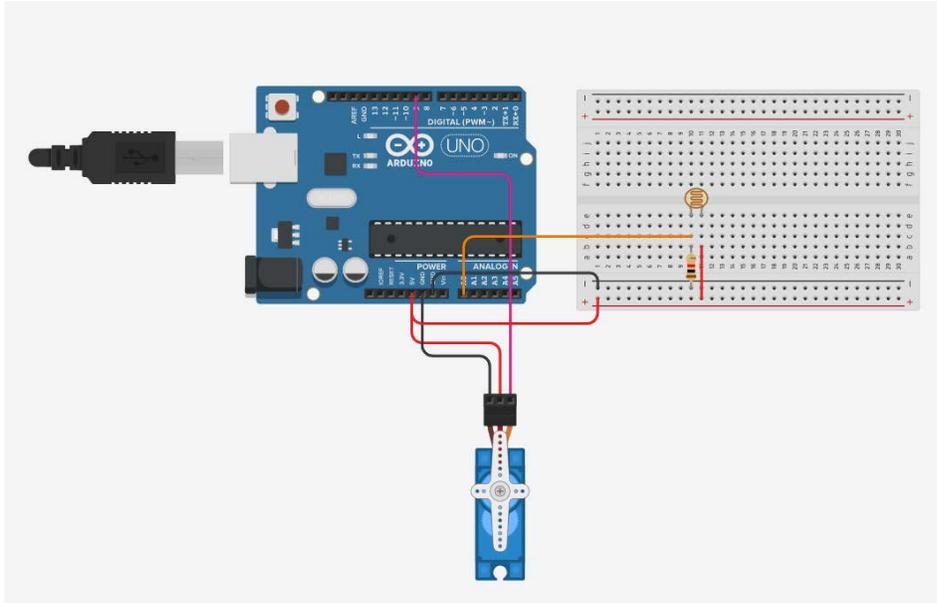
```

void setup () {
    pinMode(ENABLE, OUTPUT);
    pinMode(INP1, OUTPUT);
    pinMode(INP2, OUTPUT);
    pinMode(BOTON_A, INPUT);
    pinMode(BOTON_B, INPUT);
}

void loop (){
    if(digitalRead(BOTON_A) == HIGH) {
        digitalWrite(INP1, HIGH);
        digitalWrite(ENABLE, HIGH);
    }
    else if(digitalRead(BOTON_B) == HIGH) {
        digitalWrite(INP2, HIGH);
        digitalWrite(ENABLE, HIGH);
    }
    else {
        digitalWrite(INP1, LOW);
        digitalWrite(INP2, LOW);
        digitalWrite(ENABLE, LOW);
    }
}

```

Reto 10



```
#include <Servo.h>

Servo servo_9;

const int sensorPin = A0;    // pin del sensor

// variables:
int sensorValue = 0;        // variable de sensor

void setup() {
    servo_9.attach(9);
    Serial.begin(9600);
}

void loop() {
    // Lee el sensor
    sensorValue = analogRead(sensorPin);
    Serial.println(sensorValue);

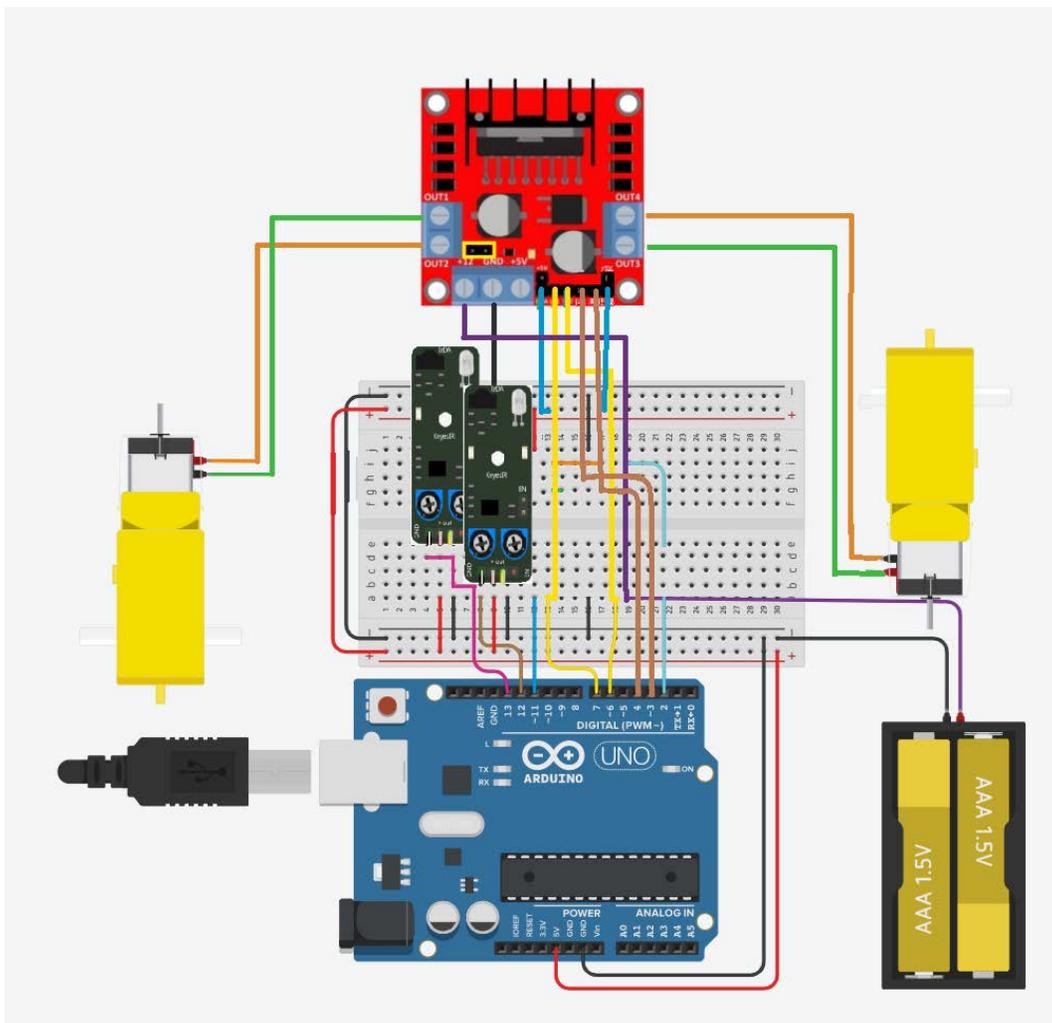
    if (sensorValue >= 620){
```

```

servo_9.write(90);
delay(2000); // Espera 2 segundos
}
if (540 >= sensorValue <= 619){
    servo_9.write(45);
    delay(2000);
}
if(0 >= sensorValue <= 530){
    servo_9.write(0);
    delay(2000);
}
}

```

Reto 11



```
int motorIzAd = 6;
int motorIzAt = 7;
int motorDeAd = 3;
int motorDeAt = 4;
int switchd = 13;
int switchi = 12;
int sensord= 0;
int sensori= 0;
```

```
void setup()
{
    pinMode(motorIzAd, OUTPUT);
    pinMode(motorIzAt, OUTPUT);
    pinMode(motorDeAd, OUTPUT);
    pinMode(motorDeAt, OUTPUT);
    pinMode(switchd, INPUT_PULLUP);
    pinMode(switchi, INPUT_PULLUP);
}
```

```
void loop()
{
    ADELANTE();
    sensord = digitalRead(switchd);
    sensori = digitalRead(switchi);
    if(sensord == HIGH)
    {
        ALTO();
        delay(500);
        ATRAS();
        delay(500);
        IZQUIERDA();
        delay(500);
        ADELANTE();
    }
}
```

```

    }
    if(sensori == HIGH)
    {
        ALTO();
        delay(500);
        ATRAS();
        delay(500);
        DERECHA();
        delay(500);
        ADELANTE();
    }
}

void ATRAS()
{
    digitalWrite(motorIzAd, HIGH);
    digitalWrite(motorIzAt, LOW);
    digitalWrite(motorDeAd, HIGH);
    digitalWrite(motorDeAt, LOW);
}

void IZQUIERDA()
{
    digitalWrite(motorIzAd, HIGH);
    digitalWrite(motorIzAt, LOW);
    digitalWrite(motorDeAd, LOW);
    digitalWrite(motorDeAt, HIGH);
}

void ADELANTE()
{
    digitalWrite(motorIzAd, LOW);
    digitalWrite(motorIzAt, HIGH);
    digitalWrite(motorDeAd, LOW);
}

```

```

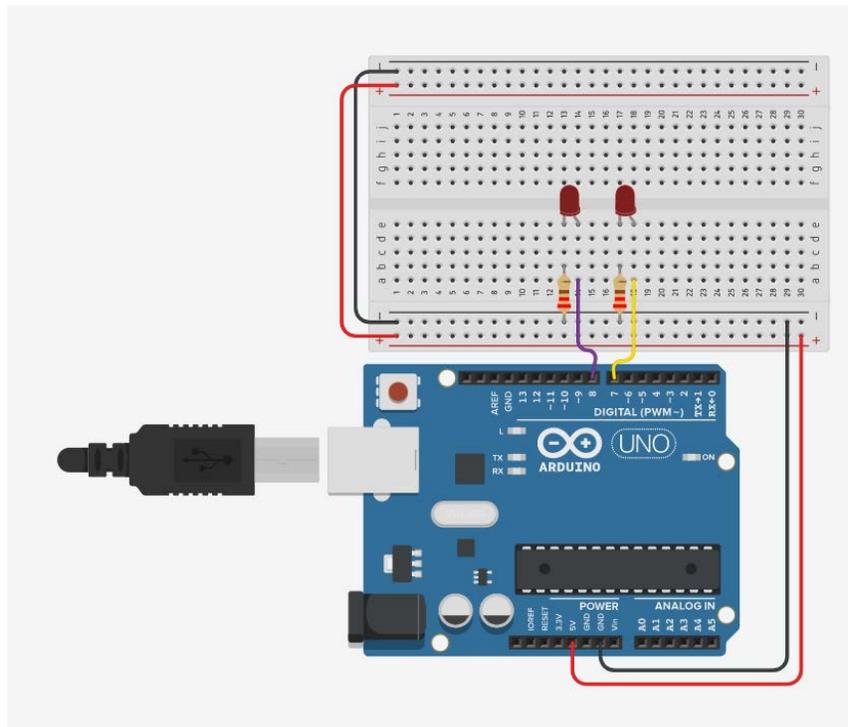
    digitalWrite(motorDeAt, HIGH);
}

void DERECHA()
{
    digitalWrite(motorIzAd, LOW);
    digitalWrite(motorIzAt, HIGH);
    digitalWrite(motorDeAd, HIGH);
    digitalWrite(motorDeAt, LOW);
}

void ALTO()
{
    digitalWrite(motorIzAd, LOW);
    digitalWrite(motorIzAt, LOW);
    digitalWrite(motorDeAd, LOW);
    digitalWrite(motorDeAt, LOW);
}

```

Reto 12



```

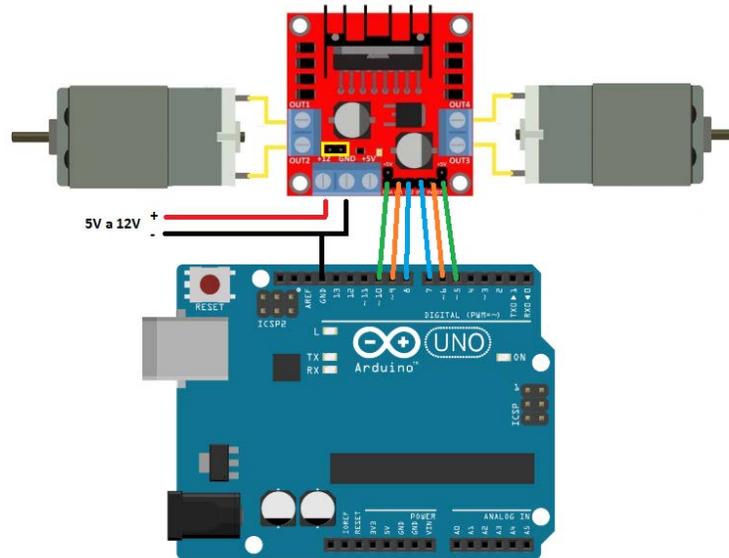
int led_1 = 8;
int led_2 = 7;
char valor;

void setup() {
    pinMode (led_1, OUTPUT);
    pinMode (led_2, OUTPUT);
    Serial.begin (9600);
}

void loop() {
    if (Serial.available ())
    {
        valor = Serial.read ();
        if (valor == 'A')
        {
            digitalWrite (led_1, HIGH);
        }
        if (valor == 'B')
        {
            digitalWrite (led_1, LOW);
        }
        if (valor == 'C')
        {
            digitalWrite(led_2, HIGH);
        }
        if (valor == 'D')
        {
            digitalWrite (led_2, LOW);
        }
    }
}

```

Reto Final



```
// Motor A
int ENA = 10;
int IN1 = 9;
int IN2 = 8;

// Motor B
int ENB = 5;
int IN3 = 7;
int IN4 = 6;

int state = 0;

void setup ()
{
    // Declaramos todos los pines como salidas
    pinMode (ENA, OUTPUT);
    pinMode (ENB, OUTPUT);
    pinMode (IN1, OUTPUT);
    pinMode (IN2, OUTPUT);
    pinMode (IN3, OUTPUT);
```

```

        pinMode (IN4, OUTPUT);
    }

void Adelante ()
{
    //Direccion motor A
    digitalWrite (IN1, HIGH);
    digitalWrite (IN2, LOW);
    analogWrite (ENA, 255); //Velocidad motor A
    //Direccion motor B
    digitalWrite (IN3, HIGH);
    digitalWrite (IN4, LOW);
    analogWrite (ENB, 255); //Velocidad motor B
}

void Atras ()
{
    //Direccion motor A
    digitalWrite (IN1, LOW);
    digitalWrite (IN2, HIGH);
    analogWrite (ENA, 128); //Velocidad motor A
    //Direccion motor B
    digitalWrite (IN3, LOW);
    digitalWrite (IN4, HIGH);
    analogWrite (ENB, 128); //Velocidad motor B
}

void Derecha ()
{
    //Direccion motor A
    digitalWrite (IN1, HIGH);
    digitalWrite (IN2, LOW);
    analogWrite (ENA, 200); //Velocidad motor A
    //Direccion motor B

```

```

    digitalWrite (IN3, LOW);
    digitalWrite (IN4, HIGH);
    analogWrite (ENB, 100); //Velocidad motor A
}

```

```

void Izquierda ()
{
    //Direccion motor A
    digitalWrite (IN1, LOW);
    digitalWrite (IN2, HIGH);
    analogWrite (ENA, 50); //Velocidad motor A
    //Direccion motor B
    digitalWrite (IN3, HIGH);
    digitalWrite (IN4, LOW);
    analogWrite (ENB, 150); //Velocidad motor A
}

```

```

void Parar ()
{
    //Direccion motor A
    digitalWrite (IN1, LOW);
    digitalWrite (IN2, LOW);
    analogWrite (ENA, 0); //Velocidad motor A
    //Direccion motor B
    digitalWrite (IN3, LOW);
    digitalWrite (IN4, LOW);
    analogWrite (ENB, 0); //Velocidad motor A
}

```

```

void loop ()
{
    if(Serial.available() > 0){
        state = Serial.read();
    }
}

```

```
if (state == 'A'){
    Adelante ();
    Serial.println("Adelante");
}
if(state == 'B'){
    Atras ();
    Serial.println("Atras");
}
if(state == 'E'){
    Parar ();
    Serial.println("Alto");
}
if(state == 'C'){
    Izquierda ();
    Serial.println("Giro: izquierda");
}
if(state == 'D'){
    Derecha ();
    Serial.println("Giro: derecha");
}
}
```