PRODELAB
Equipamiento de Educación

Curso de Programación con Arduino UNO

Muestra Comercial
Prohibida su Copia
Contenido Protegido via Educakit®

educakit

SuperKitArduino

Taller de iniciación para escuelas

Muestra Comercial
Prohibida su Copia
Contenido Protegido via Educakit®

Este curso fue diseñado y elaborado por la docente
Fernanda Aguayo

Prodelab Limitada.

Ventus 1484, Concepción, Chile.

Todos los derechos reservados. 2023

Contenido del Kit

Muestra Comercial
Prohibida su Copia
Contenido Protegido via Educakit®

Copyright © Todos los derechos reservados a Prodelab limitada.

📦**Contenido del Kit** **0**
Introducción **2**
¿Qué aprenderé al realizar este curso? 2
Desarrollo de Aptitudes **3**
Descargar el software Arduino IDE **4**
🔑**Tips iniciales** **7**
Taller 1: Set up y Loop **8**
I. Encendido y apagado del led incorporado del Arduino UNO 💡 8
II. ♾️ Ciclos infinitos (LOOPS) 10
Corriente Convencional v/s Corriente Real **12**
¿Qué es un LED? **13**
TALLER 2: LED Externo **14**
¿Qué es y cómo funciona un Protoboard? 16
I. Circuito con más de un led externo 18
Condicional if/else 20
TALLER 3: Pulsadores | Condicional 'if / else' **22**
I. Controlar el led interno mediante un pulsador usando if/else 23
Arme el circuito del montaje 23
II. Controlar el led interno mediante dos pulsadores 26
Taller 4: Bucle 'while' 29
III. Luces en espera - Bucle "while" con semáforos 30
I. El ciclo 'for' 34
TALLER 4: Buzzers | Ciclo 'for' **36**
II. Buzzers 37
TALLER 5: 🎶 Música con Arduino **42**
I. Teclado 43
TALLER 6: Ultrasonido **45**
I. Ultrasonido 46
TALLER 7: Servomotor **49**
I. Servomotor de giro restringido (FS90) 51
II. Servomotores de Rotación Continua (FS90R) 52
III. Controlando barridos de giro 54
📚 Aprendiendo a instalar librerías de distintos dispositivos. 57
TALLER 8: Pantalla LCD **59**
I. Mi primer mensaje 59
II. String 62
TALLER 9: Auto Robot I **67**
I. Auto Robot 70
TALLER 10: Auto Robot II **75**

Muestra Comercial
Prohibida su Copia
Contenido Protegido via Educakit®

Copyright © Todos los derechos reservados a Prodelab limitada.

Introducción

La robótica y la programación ya no se limitan a aplicaciones en industrias específicas, sino que se han expandido a casi todas las áreas profesionales. Su creciente influencia e importancia ha generado un aumento exponencial en la apertura de nuevos empleos y oportunidades laborales tanto en el presente como en el futuro.

En esta época de transformación tecnológica, es imprescindible **ser protagonistas de este cambio**. Es por esto, que es relevante acercar a los jóvenes a estas disciplinas, con el fin de potenciar el desarrollo de aptitudes y habilidades necesarias para enfrentar su futuro.

El ministerio de educación de Chile (Mineduc), cuenta con el Plan Nacional de Lenguajes Digitales, el cual busca promover la enseñanza del **pensamiento computacional** y la programación en el sistema educativo para potenciar las habilidades de resolución de problemas e innovación en ambientes tecnológicos.

El compromiso de **Prodelab** y nuestra docente **Fernanda Aguayo** de aportar de manera significativa, efectiva y vanguardista a la educación, nos ha llevado a desarrollar este proyecto de enseñanza didáctica: Una guía de 11 talleres y 15 actividades junto con el kit de artículos necesarios para llevarlo a cabo.

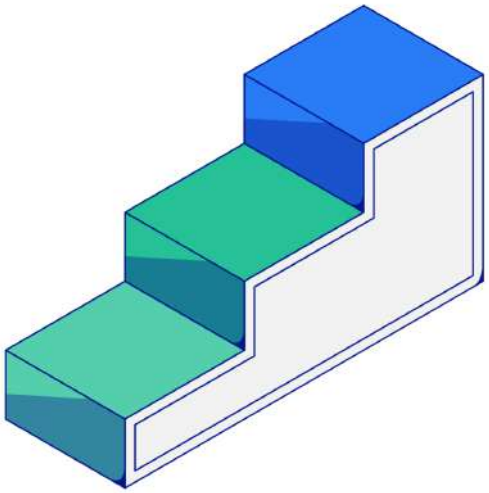
Muestra Comercial
Prohibida su Copia
Contenido Protegido via EducaKit®

¿Qué aprenderé al realizar este curso?

El alumno culminará el curso con una **base ideal** de programación: Manipulación de artículos de la electrónica, conexiones, programación, combinación de recursos para crear nuevas funciones/actividades.

Introducción interactiva al lenguaje de programación C++ y sus aplicaciones en la robótica.

El objetivo final de este curso es que el alumno pueda construir y programar dos robots; uno caminante y un vehículo a ruedas. Para lograr la construcción de estos dos robots, el curso trabaja desde lo más **simple** a lo más **avanzado,** paso a paso y **estructuradamente** para poder aprender a manipular, conectar y programar de tal forma que el aprendizaje se desarrolle de forma **progresiva y sin frustraciones.**



Copyright © Todos los derechos reservados a Prodelab limitada.

Desarrollo de Aptitudes

🎓 Te conviertes en el **protagonista** de tu propio aprendizaje

🧠 Mejoras tu **concentración**

🚀 Te acerca a nuevas **tecnologías** y te despierta el interés por las ciencias.

💡 Estimula la **creatividad** mediante la resolución de problemas

💪 Desarrollarás tu **resiliencia** y tolerancia a la frustración.

🔍 Mejoras en la **autocorrección** de errores

🔬 Estimula el Pensamiento **analítico**

🤝 Aprenderás a trabajar en equipo de manera efectiva

🤖 Desarrollarás el pensamiento lógico

Muestra Comercial
Prohibida su Copia
Contenido Protegido via Educakit®

Copyright © Todos los derechos reservados a Prodelab limitada.

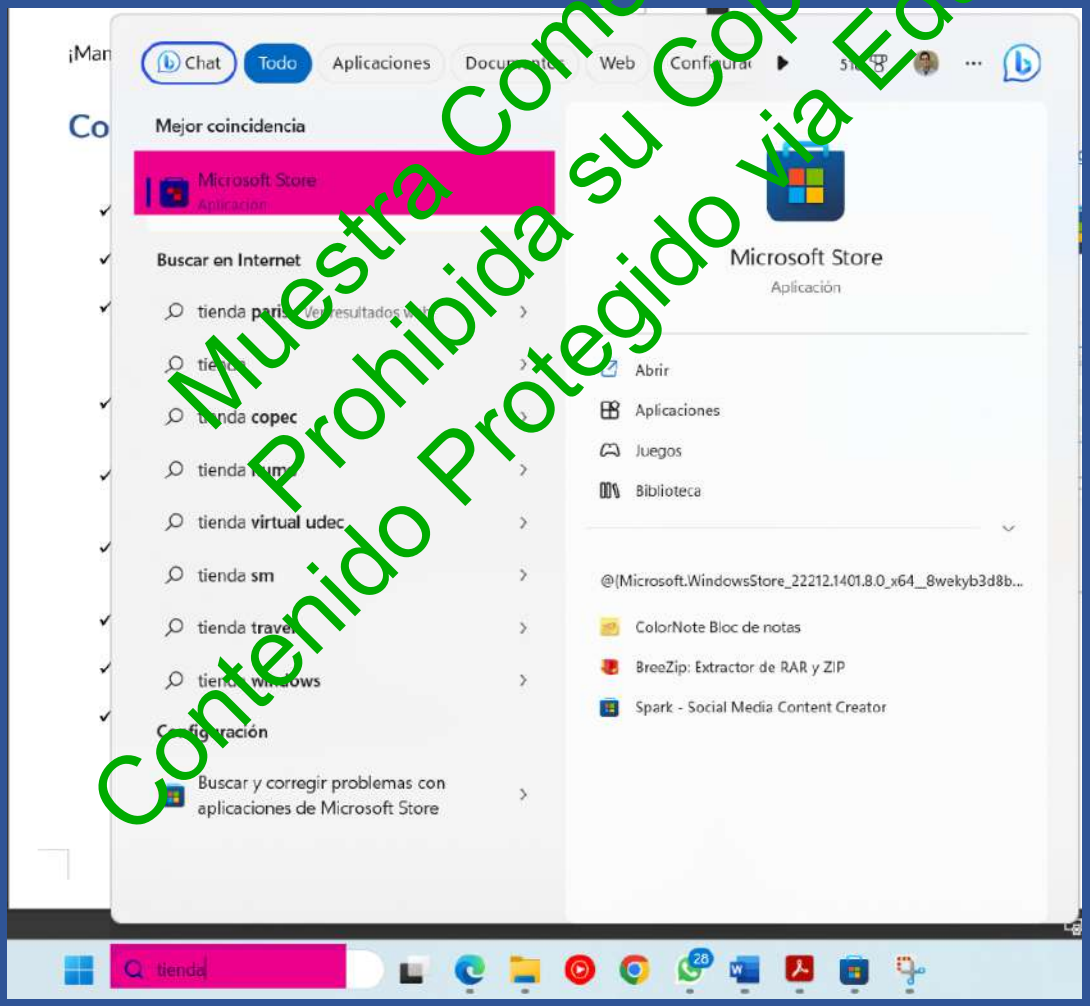
Descargar el software Arduino IDE

Opción A. Descarga de la página oficial de Arduino.

- Ingresa al siguiente link https://www.arduino.cc/en/software
- Arduino IDE 2 | Arduino Documentation ← Prueba el nuevo IDE de arduino!

Opción B. Sólo para Windows.

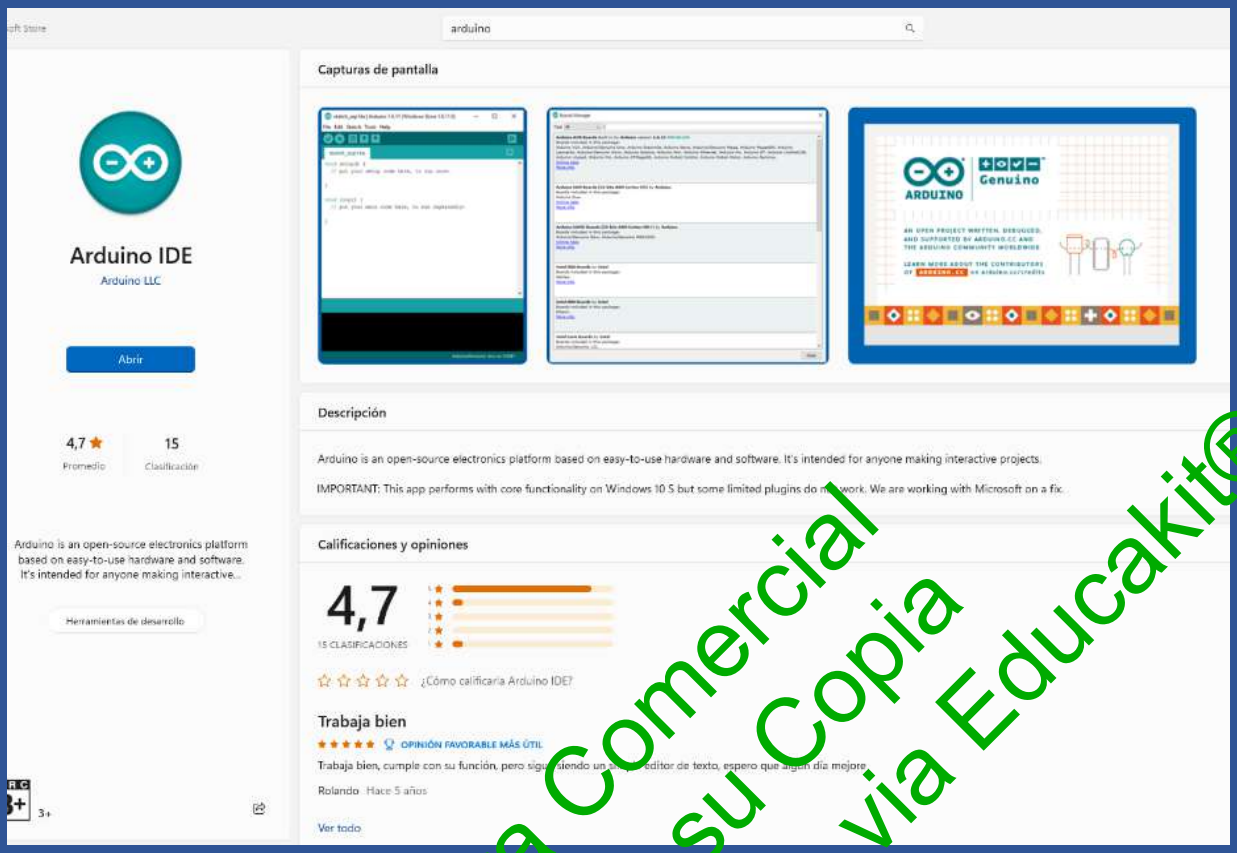
1. En la lupa de tu barra de tareas, busca **Microsoft Store** o **Tienda**.



Muestra Comercial
Prohibida su Copia
Contenido Protegido via Educakit®

Copyright © Todos los derechos reservados a Prodelab limitada.

2. En la barra de búsqueda: Busca "**Arduino IDE**"



3. Haz clic en **obtener** y luego **instalar**. Sigue los pasos.

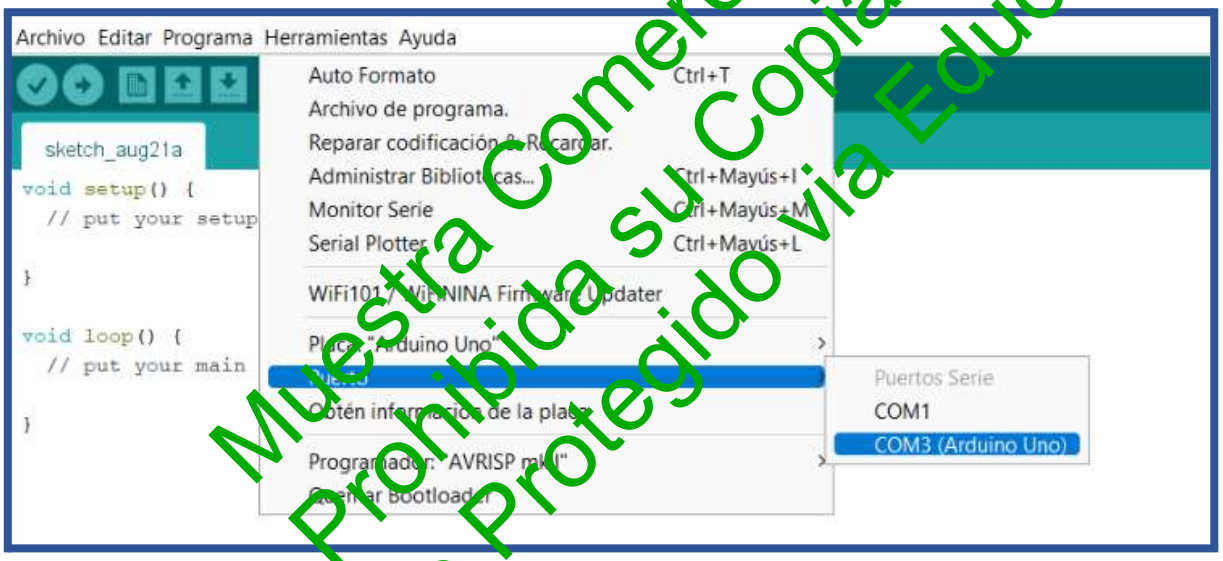
Muestra Comercial
Prohibida su Copia
Contenido Protegido via Educakit®

Copyright © Todos los derechos reservados a Prodelab limitada.

⚠️ Antes de ejecutar un código

1. Verifica que tu placa Arduino esté **conectada** a tu computadora.

2. Dirígete a ***Herramientas*** ➡️ ***Puerto*** y escoge el puerto que menciona la placa Arduino Uno.



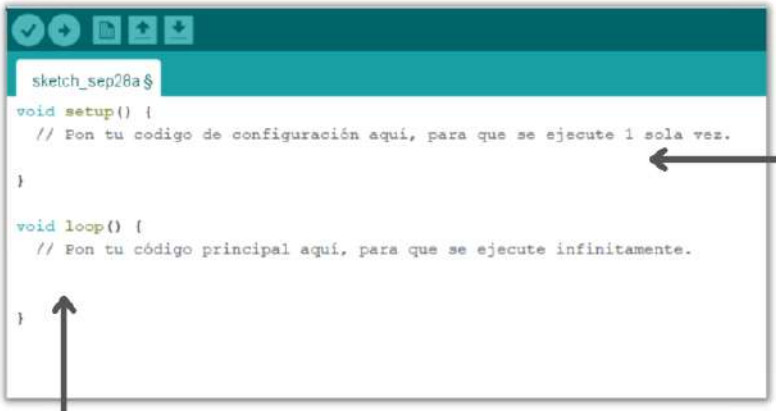
Muestra Comercial
Prohibida su Copia
Contenido Protegido Via Educakit®

Copyright © Todos los derechos reservados a Prodelab limitada.

🔑Tips iniciales

Hay dos **zonas** para escribir tu código dentro del software de Arduino.

Tu código debe ir **dentro** de los **corchetes** "{ }"para que sea leído.



En Set-up por lo general se puede:

- Crear variables y darle valores.
- Configurar las comunicaciones (por ejemplo, serial).
- Modos de configuración para pines digitales (entrada/salida).
- Inicializar cualquier componente de hardware (sensor/actuador) conectado al Arduino.
- **El código se ejecuta solo 1 vez.**

En loop ∞

- Se puede hacer todo lo demás
- Aquí, **el codigo se ejecuta infinitamente** (en loop)

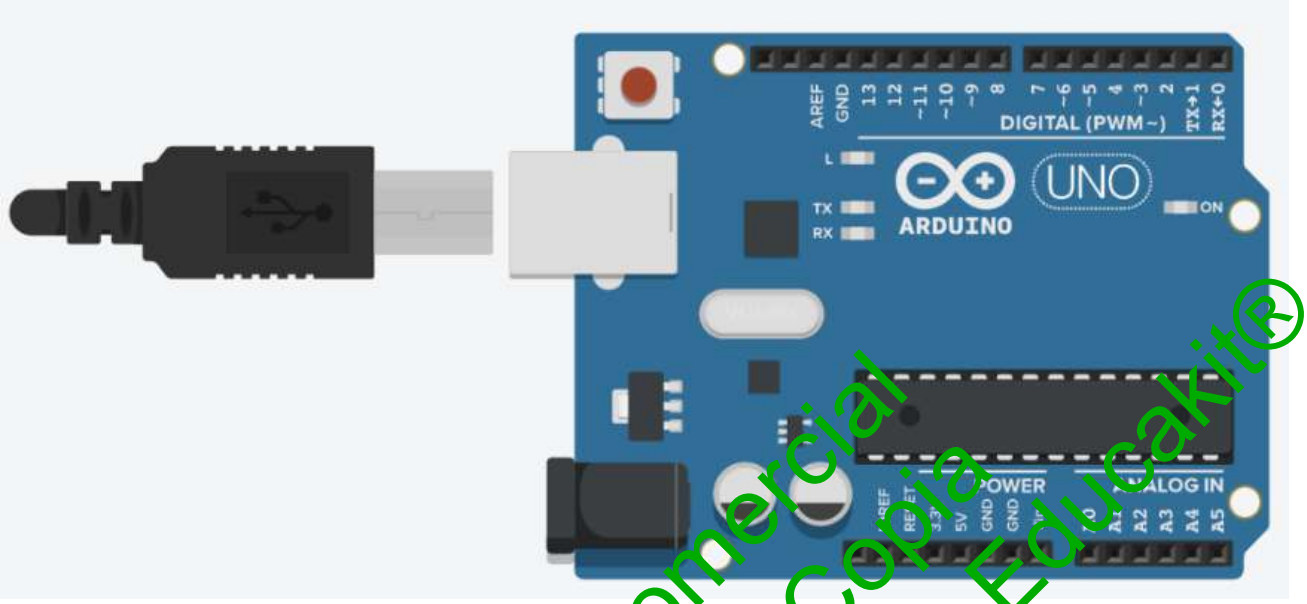
Usa '//' antes de escribir un comentario para que el programa no lo ejecute.
Esto sirve para guiarse, dejar comentarios, etc.

Muestra Comercial
Prohibida su Copia
Contenido Protegido via Educakit®

Copyright © Todos los derechos reservados a Prodelab limitada.

Taller 1: Set up y Loop

En este taller se trabajará para lograr que el led interno de la placa Arduino parpadee siguiendo patrones determinados.



Muestra Comercial
Prohibida su Copia
Contenido Protegido via Educakit®

📦 **Materiales**

- Placa Arduino
- Cable conector USB Arduino-PC

I. Encendido y apagado del led incorporado del Arduino UNO 💡

a. Introduce partes del siguiente código en el IDE de Arduino de tal forma que quede así:

```
void setup() {
pinMode(LED_BUILTIN, OUTPUT); //Establece el pin digital: LED_BUILTIN (led incorporado) como
salida.
digitalWrite(LED_BUILTIN, HIGH);  // Escribe un valor HIGH en un pin digital, prendiéndolo
delay(500);  //Espera 500 milisegundos para ejecutar la siguiente línea.
digitalWrite(LED_BUILTIN, LOW); //Apaga el pin escribiendo LOW
delay(500);
}
void loop() {
  // No se realiza ninguna acción en el bucle loop().
}
```

Copyright © Todos los derechos reservados a Prodelab limitada.

b. Observa lo que sucede con el led al ejecutar el programa y responde:

¿Cuántas veces se enciende el led?

¿Qué ocurre si en el programa entregado al inicio de la guía modificamos "delay(500)" por los siguientes valores?

delay(1000)	
delay(2000)	
delay(3000)	
delay(4000)	

c. Copia la sección del código que se presenta a continuación inmediatamente debajo del código en el cual estas trabajando (antes del ultimo "}") tantas veces como desees, carga el código, observa y responde:

```
digitalWrite(LED_BUILTIN, HIGH);
 delay(500);
digitalWrite(LED_BUILTIN, LOW);
delay(500);
```

¿Qué sucede si copiamos el código 4 veces? ¿Y si lo copiamos 7 veces?

Muestra Comercial
Prohibida su Copia
Contenido Protegido via Educakit®

Copyright © Todos los derechos reservados a Prodelab limitada.

II. ♾️ Ciclos infinitos (LOOPS)

a. Introduce el siguiente código Arduino IDE y cárgalo:

```
void setup( ) {
pinMode(LED_BUILTIN, OUTPUT);
}
void loop( ) { //Un código dentro de un loop se repetirá infinitamente.
digitalWrite(LED_BUILTIN, HIGH);
delay(500);
digitalWrite(LED_BUILTIN, LOW);
delay(500);
}
```

b. Observa qué sucede con el led y responde:

b.1 ¿Qué ha cambiado en el programa en comparación al programa anterior?

b.2 ¿Para qué sirve la sección void setup( ) y la sección void loop( ) ?

Muestra Comercial
Prohibida su Copia
Contenido Protegido via Educakit®

Copyright © Todos los derechos reservados a Prodelab limitada.

TAREA 📖

1. Edite el programa de tal forma que el LED interno parpadee 5 veces cada 250 ms.

2. Edite el programa para que el led encienda por 2 segundos y apague durante medio segundo indefinidamente.

3. Edite el programa para que el LED emita, solo una vez, la señal SOS en código morse.

4. Edite el programa para que el LED emita, solo una vez, su nombre en código morse.

A ·–	J ·–––	S ···	2 ··–––
B –···	K –·–	T –	3 ···–––
C –·–·	L ·–··	U ··–	4 ····–
D –··	M ––	V ···–	5 ·····
E ·	N –·	W ·––	6 –····
F ··–·	O –––	X –··–	7 ––···
G ––·	P ·––·	Y –·––	8 –––··
H ····	Q ––·–	Z ––··	9 ––––·
I ··	R ·–·	1 ·–––––	0 –––––

Muestra Comercial
Prohibida su Copia
Contenido Protegido via Educakit®

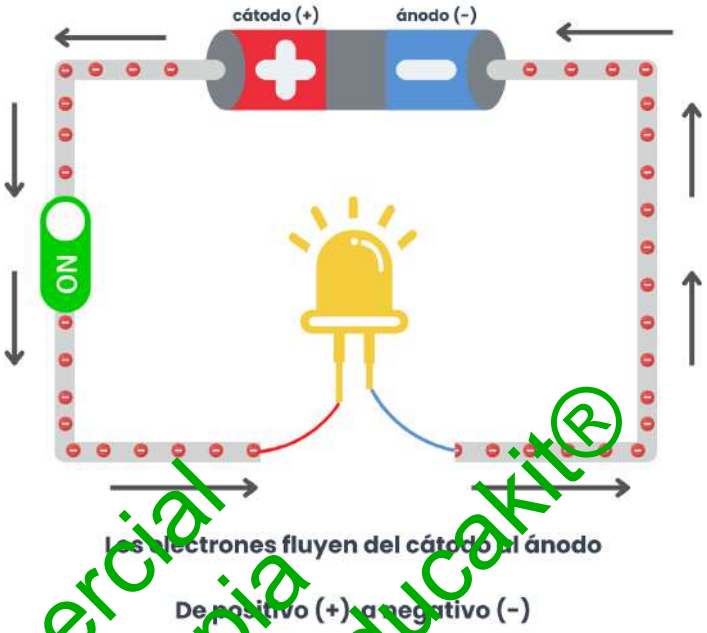
Copyright © Todos los derechos reservados a Prodelab limitada.

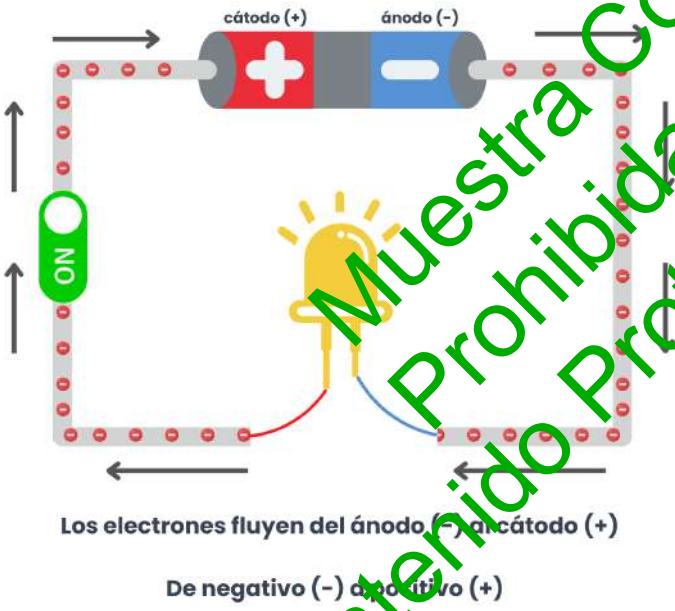
Corriente Convencional v/s Corriente Real

Convencional

La **corriente eléctrica convencional** se define como la dirección en la que se **supone** que fluye la corriente eléctrica, desde el polo positivo de la fuente de alimentación hacia el polo negativo.

(Esta es la que usamos)





Real

En realidad, los **electrones** son los que se mueven en el circuito, desde el polo negativo de la fuente de alimentación hacia el polo positivo. A esta dirección **real** de flujo de electrones se le llama **corriente eléctrica real**, que es opuesta a la dirección de la corriente convencional.

Muestra Comercial
Prohibida su Copia
Contenido Protegido via Educakit®

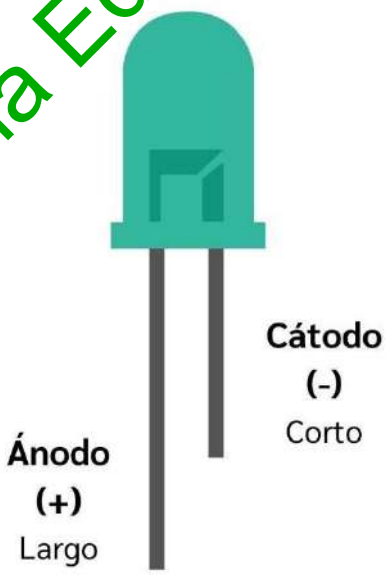
Copyright © Todos los derechos reservados a Prodelab limitada.

La historia detrás 📖

La corriente eléctrica convencional se **adoptó** originalmente porque se **creía** que fluía desde el polo positivo al negativo, pero luego se **descubrió** que en realidad fluye desde el polo negativo al positivo. A pesar de este descubrimiento, la convención de la corriente eléctrica convencional se mantuvo **porque ya era muy tarde para cambiar** y es más fácil de describir y analizar el comportamiento del circuito.

¿Qué es un LED?

Un **LED** (diodo emisor de luz) es un componente electrónico que se compone de un material **semiconductor** que permite el flujo de corriente eléctrica en una sola dirección. Cuando la electricidad fluye a través del LED en la **dirección correcta**, los electrones dentro del material emiten energía en forma de **fotones de luz**, lo que hace que el LED emita luz visible. Los LED pueden ser de diferentes colores, dependiendo del material semiconductor utilizado en su construcción.



Por otro lado, las **bombillas incandescentes** generan luz a través de la **generación de calor**, lo que resulta en una mayor **pérdida de energía** y una menor vida útil.

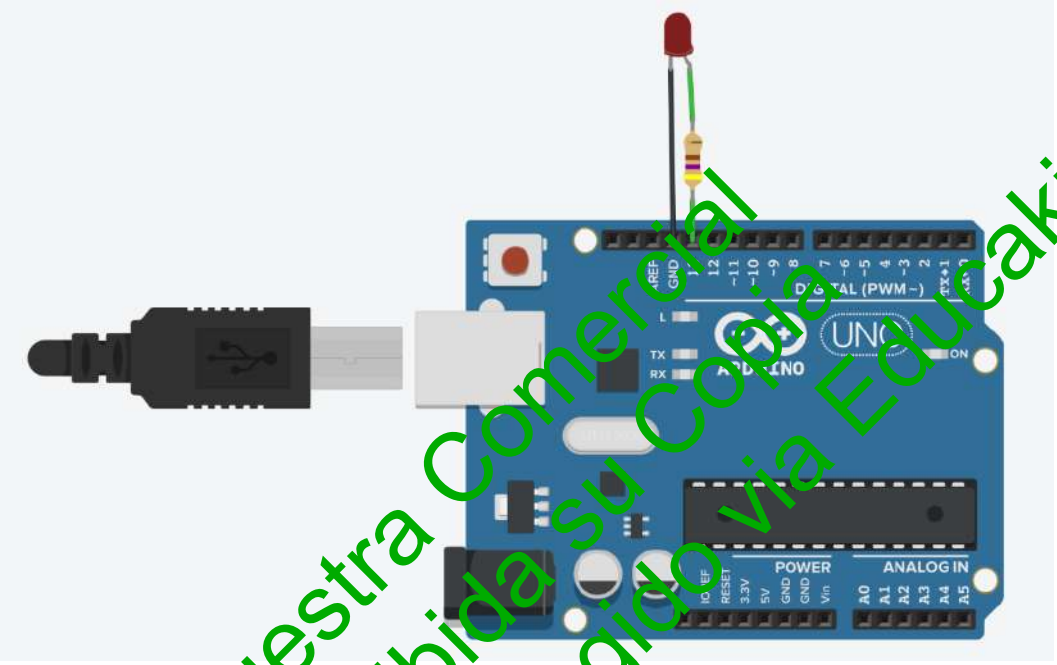
Muestra Comercial
Prohibida su Copia
Contenido Protegido via Educakit®

Copyright © Todos los derechos reservados a Prodelab limitada.

TALLER 2: LED Externo

En este taller se trabajará para lograr que los leds parpadeen siguiendo patrones determinados.

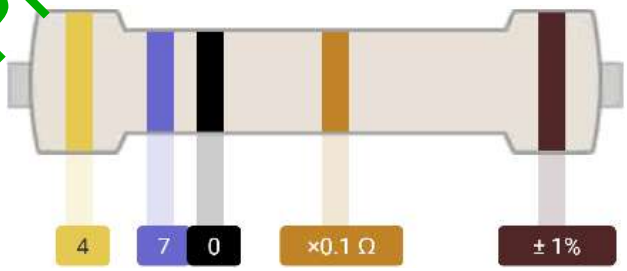
MONTAJE



Muestra Comercial
Prohibida su Copia
Contenido Protegido via Educakit®

Materiales

- Múltiples LED
- Resistencias 47 Ω 1%
- Placa Arduino
- Cables conectores
- Cable conector USB Arduino-PC



Calculadora de Resistencias

Copyright © Todos los derechos reservados a Prodelab limitada.

a. Arme el circuito del montaje, la resistencia va conectada al ánodo (+) (pata larga del LED).

b. Introduce el siguiente programa en el IDE de Arduino y cárgalo:

```
void setup() {
pinMode (13, OUTPUT);
}
void loop( ) {
digitalWrite (13, HIGH);
delay (1000);
digitalWrite (13, LOW);
delay (1000);
}
```

c. Observa que sucede y responde:

¿Qué sucede si modificamos el número que se encuentra en digitalWrite (13, HIGH); y en digitalWrite (13, LOW)?

Modifique uno, luego el otro y luego ambos. ¿Qué representa el valor "13" modificado anteriormente?

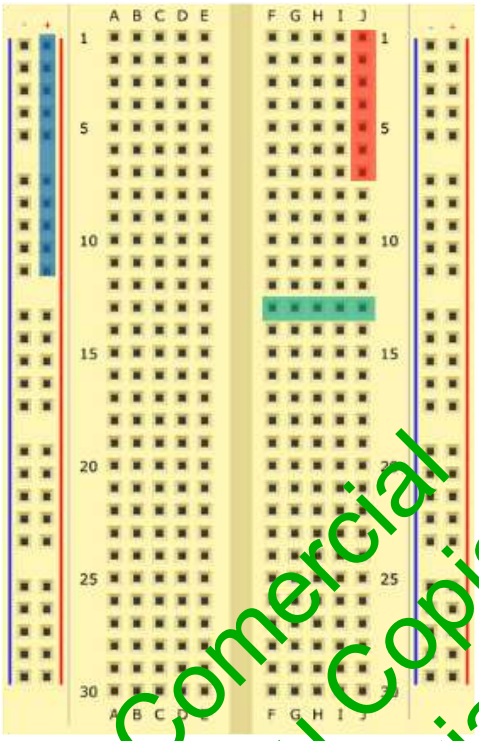
Muestra Comercial
Prohibida su Copia
Contenido Protegido via Educakit®

Copyright © Todos los derechos reservados a Prodelab limitada.

¿Qué es y cómo funciona un Protoboard?

Riel de Alimentación: Ubicados en los bordes, diferenciados por (+) o (-)

Están conectados verticalmente



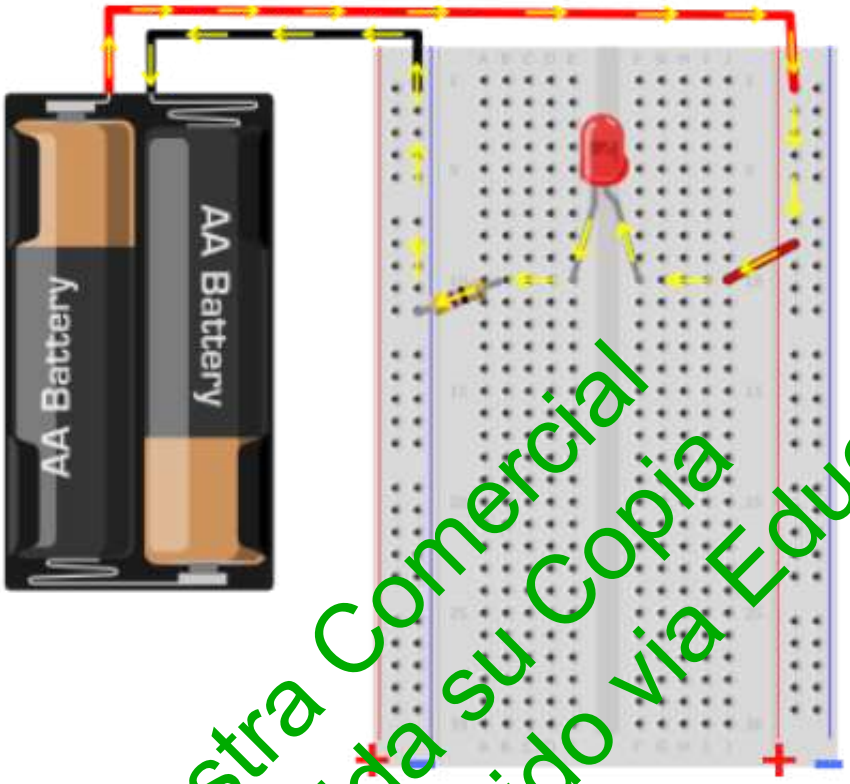
Los orificios a lo largo de las columnas NO están conectados.

Orificios a lo largo de cada fila están conectados

Muestra Comercial
Prohibida su Copia
Contenido Protegido via Educakit®

Copyright © Todos los derechos reservados a Prodelab limitada.

Ejemplo de uso de un Riel de Alimentación



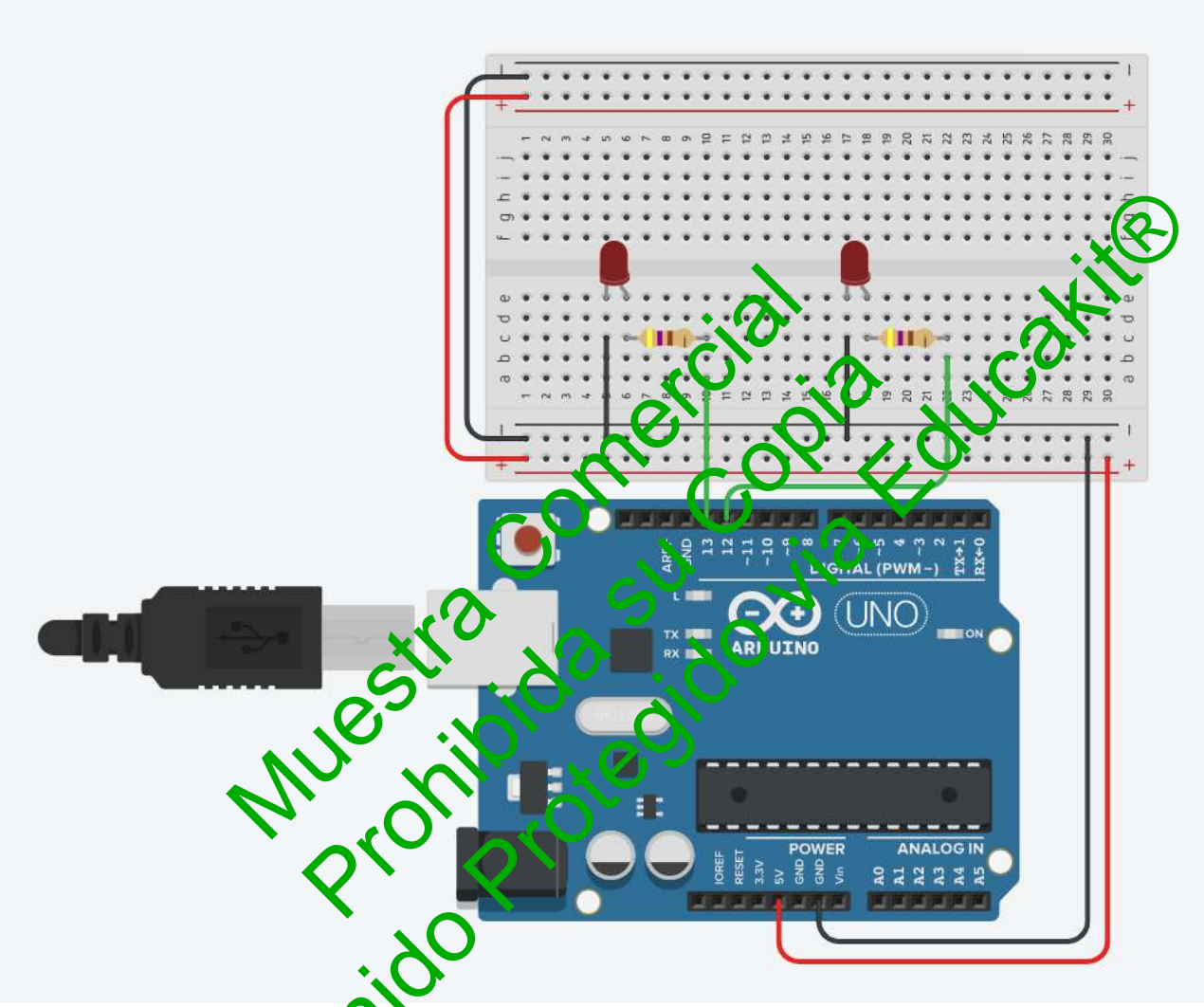
En una protoboard, **los rieles de alimentación** son las columnas marcadas con signos **positivos y negativos**. Estos rieles se utilizan para conectar la **fuente de energía** a la protoboard y alimentar los componentes electrónicos. Por ejemplo, si queremos conectar un **LED**, podemos conectar la pata larga al riel positivo y la pata corta a la columna donde queremos ubicarlo. Es importante **tener cuidado y respetar las polaridades** de los componentes para evitar dañarlos o poner en riesgo nuestra seguridad.

Muestra Comercial
Prohibida su Copia
Contenido Protegido via Educakit®

Copyright © Todos los derechos reservados a Prodelab limitada.

I. Circuito con más de un led externo

a. Arme el circuito como se muestra en el siguiente diagrama.



Muestra Comercial
Prohibida su Copia
Contenido Protegido vía Educakit®

Aunque no se utilice el riel de alimentación positivo, **se le da corriente de todas formas por costumbre.**

Copyright © Todos los derechos reservados a Prodelab limitada.

b. Introduce el siguiente programa en el IDE de Arduino y cárgalo:

```
void setup( ) {
pinMode (13, OUTPUT);
pinMode (12, OUTPUT); // Se establecen los pintes 13 y 12 como salida
}
void loop( ) {
digitalWrite (13, HIGH);
digitalWrite (12, HIGH);
delay (1000);
digitalWrite (13, LOW);
digitalWrite (12, LOW);
delay (1000);
}
```

c. Observa lo que ocurre y contesta:

¿Qué se podría hacer para que cuando un led está encendido el otro este apagado?

¿Qué se podría hacer para que un led parpadee cada 1 segundo y el otro cada medio segundo?

Muestra Comercial
Prohibida su Copia
Contenido Protegido via Educakit®

TAREA 📖

1. Confeccione un semáforo peatonal, el cual deberá parpadear 5 veces antes de cambiar de color verde a rojo.

Copyright © Todos los derechos reservados a Prodelab limitada.

2. Confeccione un semáforo de automóviles. Recuerde que el amarillo este prendido menos tiempo en comparación al rojo y al verde.

Condicional if/else

¿Qué es?

La declaración condicional **if/else** es una estructura de control fundamental en programación. Permite que tu programa tome decisiones basándose en ciertas condiciones.

¿Cómo funciona?

Imagina que estás frente a un **semáforo**. Si el semáforo está en **verde**, puedes cruzar la calle. Si está en **rojo**, tienes que esperar. Esta es una situación de la vida real que se puede representar con un **if/else.**

En términos de código, podría verse algo así:

```
if (semaforo == verde) {

   cruzarCalle();

}
 else {
   esperar();
}
```

Muestra Comercial
Prohibida su Copia
Contenido Protegido via Educakit®

Copyright © Todos los derechos reservados a Prodelab limitada.

```
if (semaforo==verde) {
    código 1
}
else {
    código 2
}
```

Si la condición **se cumple** (verdadera)

Se ejecuta

Si la condición **no se cumple** (falsa)

Se ejecuta

Muestra Comercial
Prohibida su Copia
Contenido Protegido via Educakit®

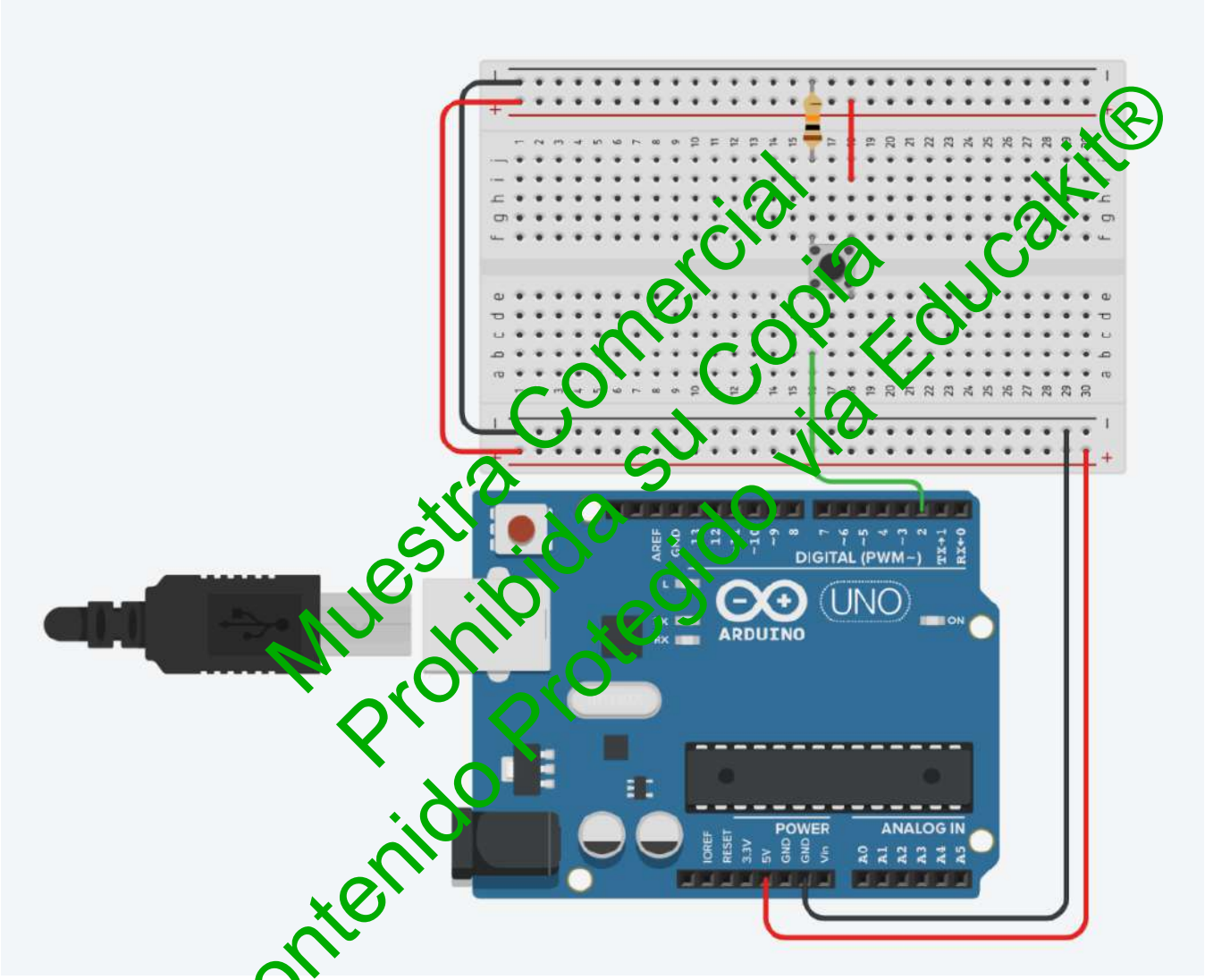
Copyright © Todos los derechos reservados a Prodelab limitada.

TALLER 3: Pulsadores | Condicional 'if / else'

En este taller se trabajará:

- Estructura if/else (si/entonces)
- **if/else** aplicado en un ejercicio con pulsadores.

Montaje



Muestra Comercial
Prohibida su Copia
Contenido Protegido via Educakit®

Materiales

- 1 Pulsador
- Resistencia 10 kΩ
- Cables conectores
- Placa Arduino
- Cable conector USB Arduino-PC
- Protoboard

Copyright © Todos los derechos reservados a Prodelab limitada.

I. Controlar el led interno mediante un pulsador usando if/else

Arme el circuito del montaje

a. Introduzca el siguiente programa en el IDE de Arduino y cárgalo:

```
void setup() {
pinMode(2, INPUT);
pinMode(13, OUTPUT); /// El led integrado está asociado al Pin 13, por lo que podemos escribirlo así o usar LED_BUILTIN como en el primer taller. Es exactamente lo mismo.
}
void loop() {
if (digitalRead(2) == HIGH) { // Si la lectura del pin 2 es HIGH.
digitalWrite(13, HIGH); // Entonces, enciende el LED incorporado.
}
else {
digitalWrite(13, LOW);  //Si no es así, apaga el LED.
}
}
```

c. Describe brevemente el funcionamiento del circuito.

d. ¿Qué parámetros es necesario cambiar para que el led esté encendido salvo cuando se aprieta el botón?

Muestra Comercial
Prohibida su Copia
Contenido Protegido via Educakit®

Copyright © Todos los derechos reservados a Prodelab limitada.

e. ¿Cuál es la función del código “if” y “else”?

f. Introduce el siguiente programa en el IDE de Arduino y cárgalo:

```
const int led = 13;
const int pulsador = 2;
int valorPulsador = 0;
void setup() {
pinMode(pulsador, INPUT);
pinMode(led, OUTPUT);
}
void loop() {
valorPulsador = digitalRead(pulsador);
if (valorPulsador == HIGH) {
 digitalWrite(led, HIGH);
}
 else {
digitalWrite(led, LOW);
}
}
```

Muestra Comercial
Prohibida su Copia
Contenido Protegido via Educakit®

g. Describe brevemente el funcionamiento del circuito.

Copyright © Todos los derechos reservados a Prodelab limitada.

h. ¿Qué diferencias encuentras entre el código anterior y el actual?, ¿cambia su funcionamiento?

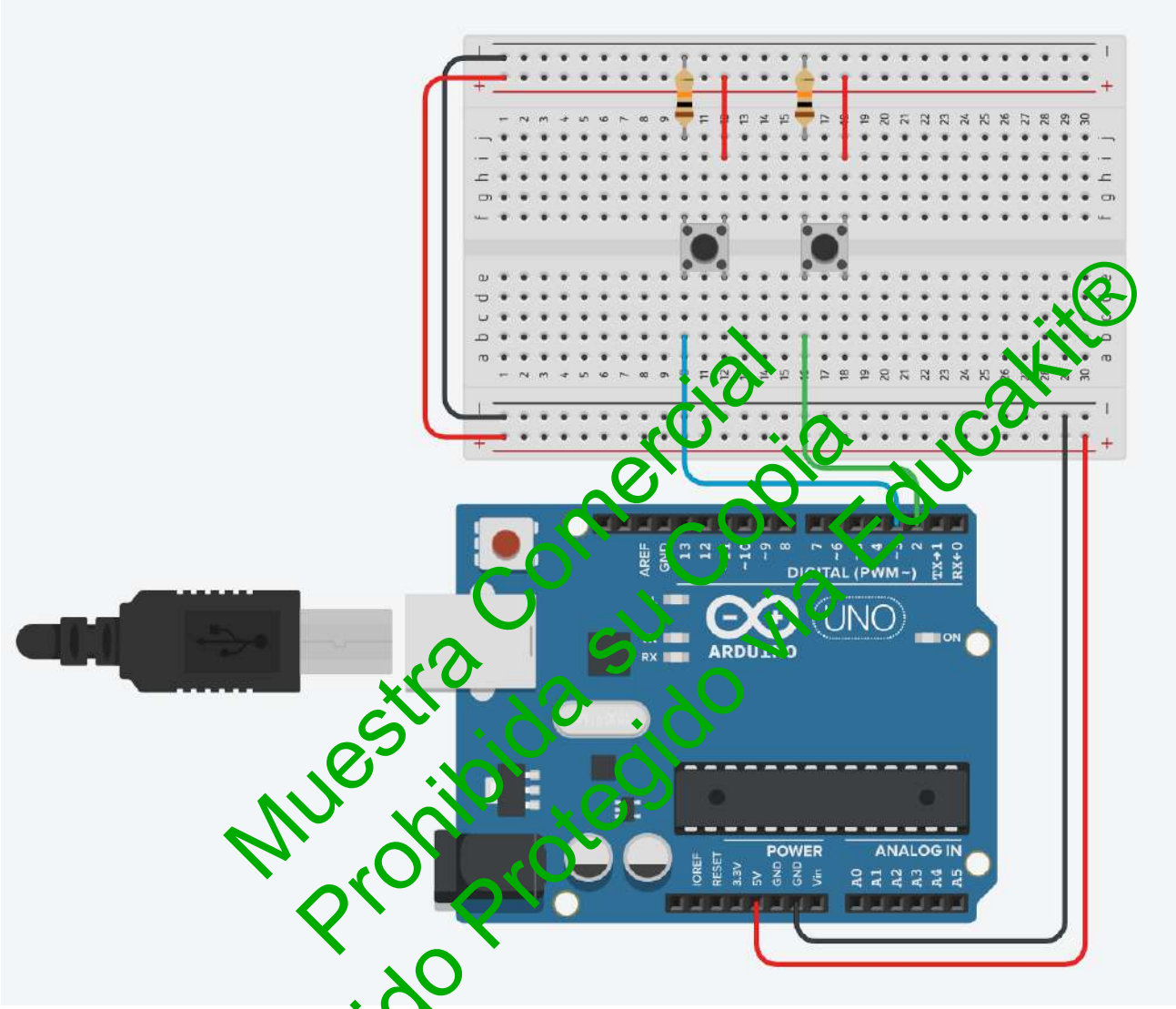
i. Respecto a la diferencia presentada en el código, ¿Cuándo crees que puede ser más útil ocupar el actual?

Muestra Comercial
Prohibida su Copia
Contenido Protegido via Educakit®

Copyright © Todos los derechos reservados a Prodelab limitada.

II. Controlar el led interno mediante dos pulsadores

a. Arme el circuito como se muestra en el siguiente diagrama.



Muestra Comercial
Prohibida su Copia
Contenido Protegido Via Educakit®

Materiales

• 2 Pulsadores • Resistencias 10 kΩ • Cables conectores	• Placa Arduino • Cable conector USB Arduino-PC • Protoboard

Copyright © Todos los derechos reservados a Prodelab limitada.

b. Introduce el siguiente programa en el IDE de Arduino y cárgalo:

```
const int led = 13;
const int pulsador1 = 3;
const int pulsador2 = 2;

// Se definen constantes: La usaremos para identificar el pin 13 como led, el pin 3 como pulsador
1 y el pin 2 como pulsador 2. Esto no es obligatorio.

int valorPulsador1 = 0;
int valorPulsador2 = 0;

// Definimos dos variables para almacenar el estado actual de los pulsadores.
// Al principio, asumimos que los pulsadores no están presionados (valor = 0 o LOW).

void setup() {
pinMode(led, OUTPUT);
pinMode(pulsador1, INPUT);
pinMode(pulsador2, INPUT);
}
void loop(){
valorPulsador1=digitalRead(pulsador1);
valorPulsador2=digitalRead(pulsador2);
if (valorPulsador1 == HIGH) {
digitalWrite(led, HIGH);
}
else if (valorPulsador2 == HIGH) { .
digitalWrite(led, LOW);
}
}
```

Muestra Comercial
Prohibida su Copia
Contenido Protegido via Educakit®

c. Describe brevemente el funcionamiento del circuito.

Copyright © Todos los derechos reservados a Prodelab limitada.

d. Describe con tus propias palabras el código descrito anteriormente y compáralo con el del comienzo de la actividad, mencionando la principal diferencia en términos del código.

TAREA 📖

1. Conecta un led externo y programalo para que al presionar el pulsador este se mantenga encendido.

2. Programa el circuito de tal manera que un botón haga parpadear el led y el otro botón logre que el mismo led se mantenga encendido.

3. Realiza un programa para el funcionamiento de un semáforo peatonal, donde se mantenga en rojo siempre hasta que se presione el botón y:

a) La luz roja se mantiene por 3 segundos
b) Se enciende la luz verde y se apaga la luz roja al mismo tiempo
c) La luz verde se mantuvo por 3 segundos y comienza a parpadear 5 veces
d) Vuelve a encender la luz roja

Muestra Comercial
Prohibida su Copia
Contenido Protegido Via Educakit®

Copyright © Todos los derechos reservados a Prodelab limitada.

Taller 4: Bucle *'while'*

En el **Taller 3**, exploramos cómo un semáforo nos permite tomar una decisión: cruzar la calle si está verde (if) o esperar si está rojo (else). En nuestro circuito, utilizamos un pulsador para representar el semáforo, y el LED es nuestra acción de cruzar la calle.

Para ejemplificar el Bucle While, analizaremos la siguiente idea:

Taller 3: Si el semáforo está en rojo, no puedo cruzarlo. Si no está en rojo, puedo cruzarlo.

```
if (semaforo == rojo) {

   esperarEnElLugar(); // SI (if) el semáforo esté en rojo, espera en el
lugar.

}
 else {
   esperar(); // Si No (else), puedes cruzar.
}
```

Ahora, en el **Taller 4,** llevaremos este concepto un paso más allá. Imagina que en lugar de un semáforo normal, tienes un semáforo especial que parpadea en rojo. Solo puedes cruzar cuando se apaga completamente.

Vamos a traducir esto a nuestro circuito: en lugar de tomar una acción instantánea con if/else, vamos a mantener una acción **mientras se cumpla una condición** con while.

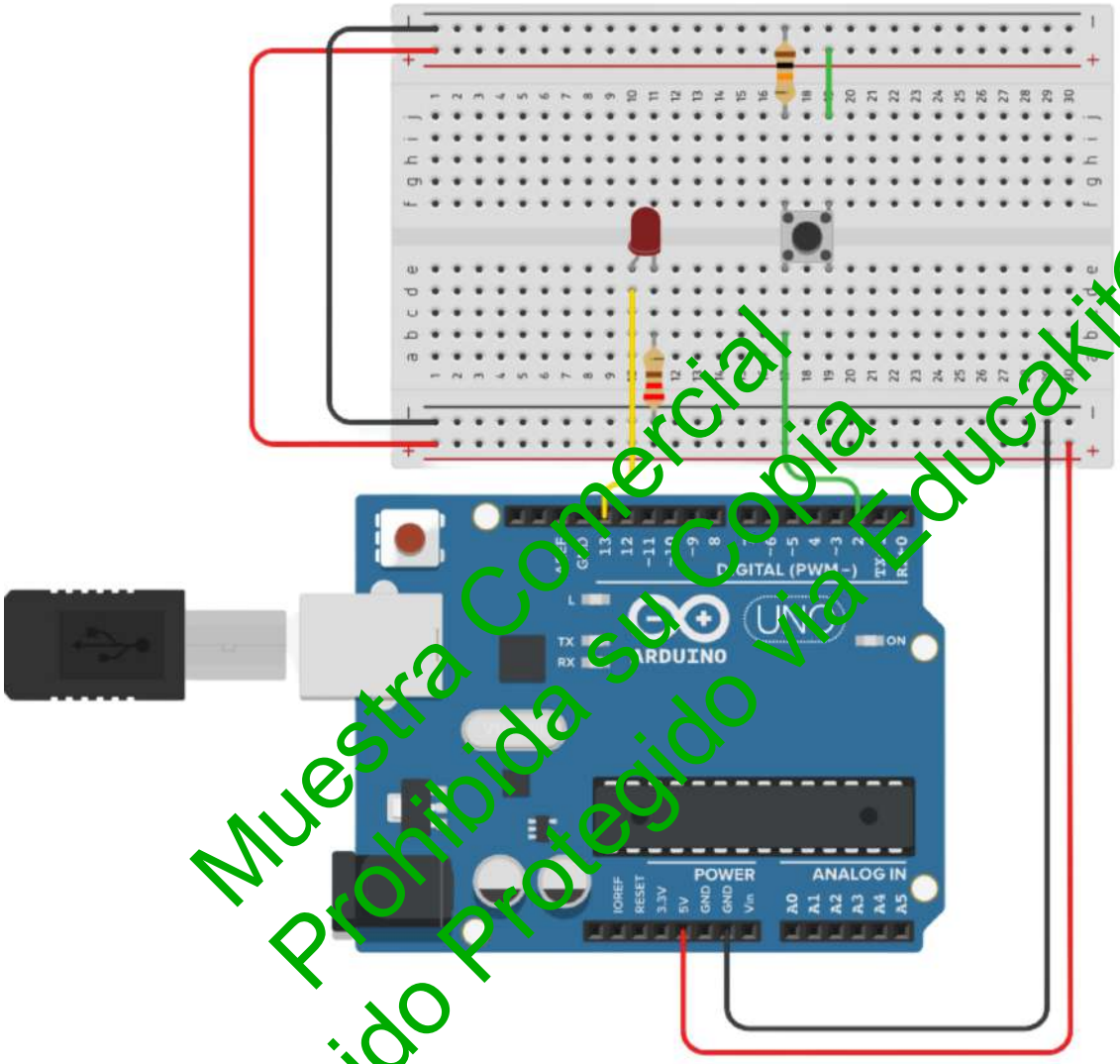
```
while (semaforo == rojo) {

   esperarEnElLugar(); // Mientras el semáforo esté en rojo, espera en el
lugar.

}
cruzarCalle(); // Cuando el semáforo deja de estar en rojo, puedes
cruzar.
```

Muestra Comercial
Prohibida su Copia
Contenido Protegido via Educakit®

Copyright © Todos los derechos reservados a Prodelab limitada.

III. Luces en espera - Bucle "while" con semáforos.

a. Arma el circuito como se muestra en el siguiente diagrama.



Muestra Comercial
Prohibida su Copia
Contenido Protegido Via Educakit®

Materiales

- 1 Pulsador y 1 LED rojo.
- Resistencias 10 kΩ para el pulsador y 220Ω para el LED rojo.
- Cables conectores
- Placa Arduino y Protoboard
- Cable conector USB

Copyright © Todos los derechos reservados a Prodelab limitada.

b. Escribe o copia el siguiente código a tu Arduino IDE.

```
void setup() {
  pinMode(2, INPUT); // Configura el pin del pulsador como entrada
  pinMode(13, OUTPUT); // Configura el pin del LED como salida
}

void loop() {
  // Comienza el bucle while
  while(digitalRead(2) == HIGH) { // Mientras el pulsador esté presionado...
    digitalWrite(13, HIGH); // ...enciende el LED
    delay(500); // Espera medio segundo
    digitalWrite(13, LOW); // Apaga el LED
    delay(500); // Espera otro medio segundo
  }
  // El LED se apagará cuando el pulsador se suelte
}
```

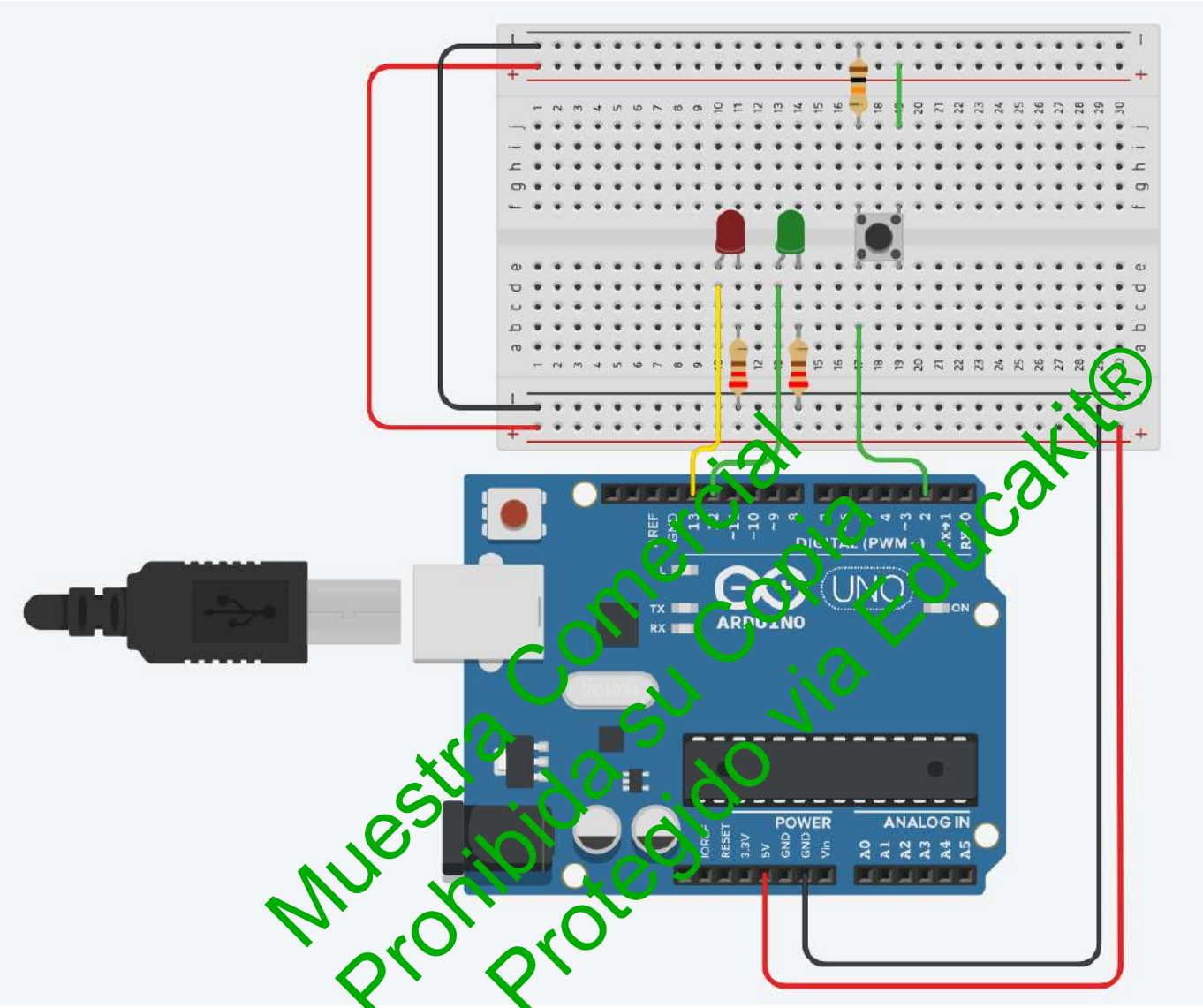
Muestra Comercial
Prohibida su Copia
Contenido Protegido via Educakit®

c. Describe brevemente el funcionamiento del circuito.

d. ¿Cómo lo asocias a los ejemplos de los semáforos?

Copyright © Todos los derechos reservados a Prodelab limitada.

e. Modifiquemos el circuito agregando un “semáforo verde”



Muestra Comercial
Prohibida su Copia
Contenido Protegido via Educakit®

f. Escribe lo que necesitas agregar al código original para realizar lo siguiente: "El semáforo estará en rojo mientras mantengas el botón pulsado, cuando no esté pulsado se pondrá verde y podrás cruzar."

Te dejamos algunas *[pistas]* para modificar el código original.

```
void setup() {
  pinMode(2, INPUT); // Configura el pin del pulsador como entrada
  pinMode(13, OUTPUT); // Configura el pin del LED como salida
  [Configura un pin extra aquí]
}

void loop() {
  // Comienza el bucle while
  while(digitalRead(2) == HIGH) { // Mientras el pulsador esté presionado...
    digitalWrite(13, HIGH); // ...enciende el LED
    delay(500); // Espera medio segundo
    digitalWrite(13, LOW); // Apaga el LED
    delay(500); // Espera otro medio segundo
  }

  [Escribe aquí el codigo para encender el LED verde]
}
```

Muestra Comercial
Prohibida su Copia
Contenido Protegido via Educakit®

Copyright © Todos los derechos reservados a Prodelab limitada.

I. El ciclo 'for'

¿Qué es?

El bucle for es una estructura de control en programación que permite repetir un bloque de código un número determinado de veces.

¿Cómo funciona?

Imagina que estás practicando tu lanzamiento de baloncesto. Decides que vas a lanzar el balón 10 veces. Este es un ejemplo de una situación en la que podrías usar un bucle for.

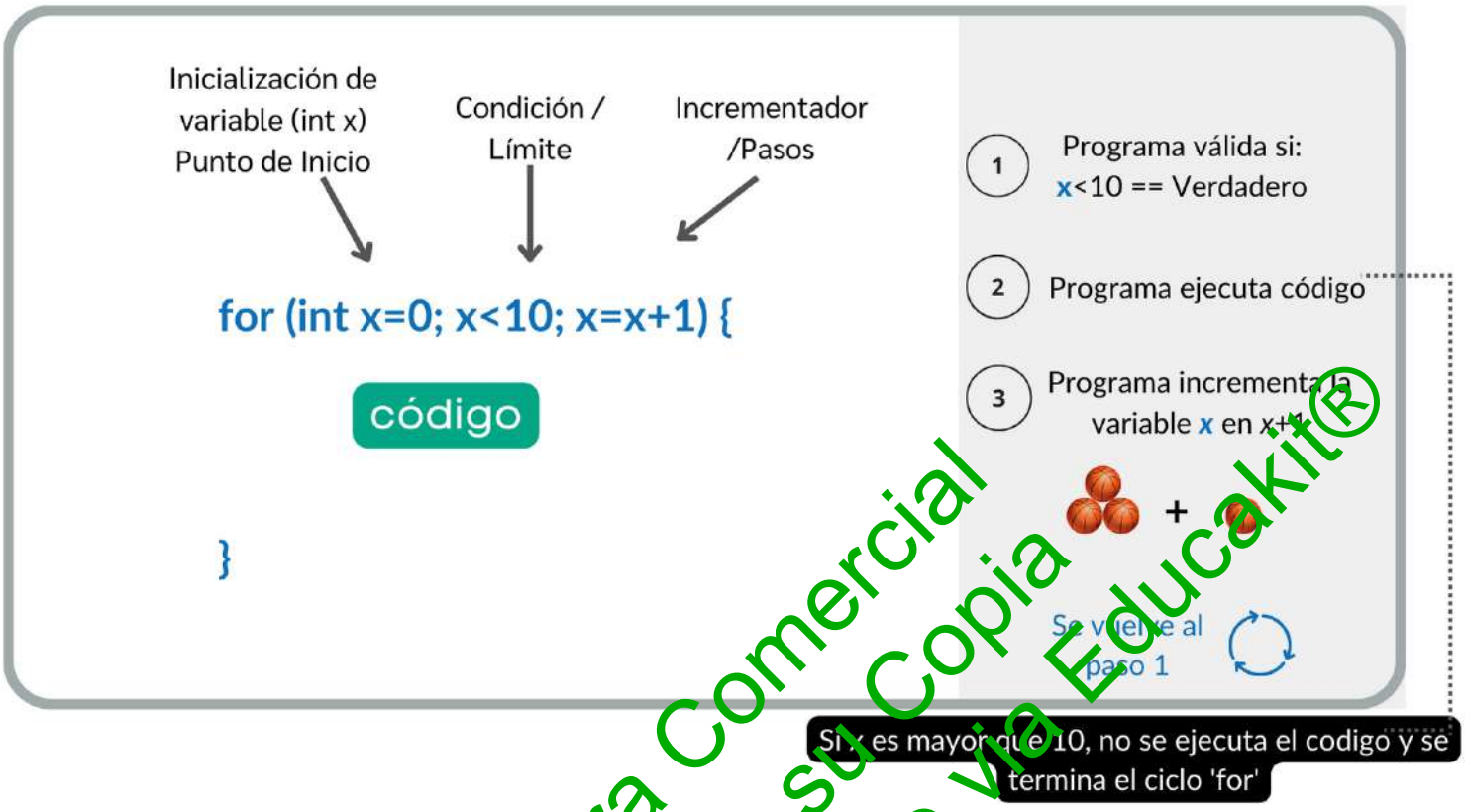
En términos de código, podría verse algo así:

```
for (int x = 0; x < 10; x++) {
  lanzarBalon();
}
```

Muestra Comercial
Prohibida su Copia
Contenido Protegido via Educakit®

Copyright © Todos los derechos reservados a Prodelab limitada.

a. **Ejemplo: ¿Cómo hago que un código se ejecute 10 veces?**



Muestra Comercial
Prohibida su Copia
Contenido Protegido Via Educakit®

- Recuerda que la variable x también se puede crear antes de establecer el ciclo for. *Int x*

- *x<3* El código dentro de 'for' se ejecutará siempre y cuando el valor de x sea menor a 3. **Si no es así**, el ciclo 'for' **termina** y se sigue con el código que hay a continuación.

- X = x+1 Después de ejecutar el código dentro de 'for', el valor de x ahora será x+1.

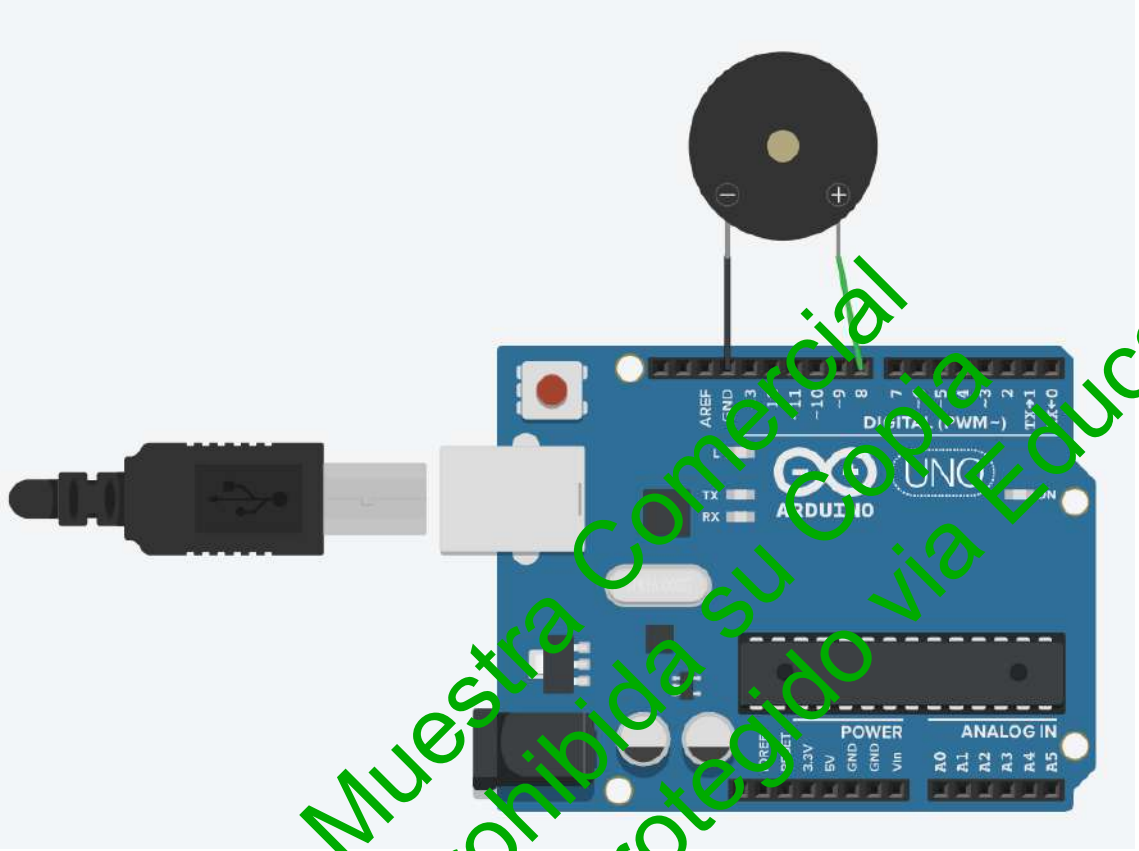
★ El valor de incrementador puede ser la ecuación que nosotros queramos, en este caso es x+1.
★ En C++, hay distintas formas de hacer este aumento: x+=1 o **x++**

Copyright © Todos los derechos reservados a Prodelab limitada.

TALLER 4: Buzzers | Ciclo 'for'

En este taller se trabajará para lograr programar un buzzers con alguna canción o melodía.

Montaje



Muestra Comercial
Prohibida su Copia
Contenido Protegido via Educakit®

Materiales

- 3 buzzers
- Placa Arduino
- Cable conector USB Arduino-PC

Copyright © Todos los derechos reservados a Prodelab limitada.

II. Buzzers

a. Introduce el siguiente programa en el IDE de Arduino:

```
void setup()
{
pinMode(8, OUTPUT);

tone(8, 230);
delay(200);
noTone(8);

tone(8, 523);
delay(300);
noTone(8);
}
void loop()
{

}
```

Muestra Comercial
Prohibida su Copia
Contenido Protegido via Educakit®

b. ¿Qué ocurre si cambiamos el 230 del código tone?

Copyright © Todos los derechos reservados a Prodelab limitada.

c. A continuación, encontrarás información con respecto a los tonos y las figuras musicales.

Figura	Tiempo (ms)
Redonda	4000
Blanca	2000
Negra	1000
Corchea	500
Semicorchea	250
Fusa	125
Semifusa	61

Nota	Frecuencia
Do	261 Hz
Re	294 Hz
Mi	330 Hz
Fa	349 Hz
Sol	392 Hz
La	440 Hz
Si	494 Hz
Do	523 Hz

Muestra Comercial
Prohibida su Copia
Contenido Protegido via Educakit®

Copyright © Todos los derechos reservados a Prodelab limitada.

d. Ahora, nos ayudaremos del ciclo 'for' para repetir 3 veces la melodía de "Feliz Cumpleaños".

```
void setup()
{

pinMode(8, OUTPUT);

for (int x=0; x<3; x=x+1){

tone(8, 261);
delay(500);
noTone(8);

tone(8, 261);
delay(500);
noTone(8);

tone(8, 294);
delay(500);
noTone(8);

tone(8, 261);
delay(500);
noTone(8);

tone(8, 349);
delay(500);
noTone(8);

tone(8, 330);
delay(500);
noTone(8);
delay(500);

tone(8, 261);
delay(500);
noTone(8);

tone(8, 261);
delay(500);
noTone(8);
```

Muestra Comercial
Prohibida su Copia
Contenido Protegido via Educakit®

Copyright © Todos los derechos reservados a Prodelab limitada.

```
tone(8, 294);
delay(500);
noTone(8);

tone(8, 261);
delay(500);
noTone(8);

tone(8, 392);
delay(500);
noTone(8);

tone(8, 349);
delay(500);
noTone(8);
delay(500);

tone(8, 261);
delay(500);
noTone(8);

tone(8, 261);
delay(500);
noTone(8);

tone(8, 520);
delay(500);
noTone(8);

tone(8, 440);
delay(500);
noTone(8);

tone(8, 349);
delay(500);
noTone(8);

tone(8, 330);
delay(500);
noTone(8);

tone(8, 294);
delay(500);
noTone(8);
delay(500);
```

Muestra Comercial
Prohibida su Copia
Contenido Protegido via Educakit®

Copyright © Todos los derechos reservados a Prodelab limitada.

```
tone(8, 494);
delay(500);
noTone(8);

tone(8, 494);
delay(500);
noTone(8);

tone(8, 440);
delay(500);
noTone(8);

tone(8, 349);
delay(500);
noTone(8);

tone(8, 392);
delay(500);
noTone(8);

tone(8, 349);
delay(500);
noTone(8);
delay(1000);
}
}
void loop()
{

}
```

Muestra Comercial
Prohibida su Copia
Contenido Protegido via Educakit®

TAREA 📖

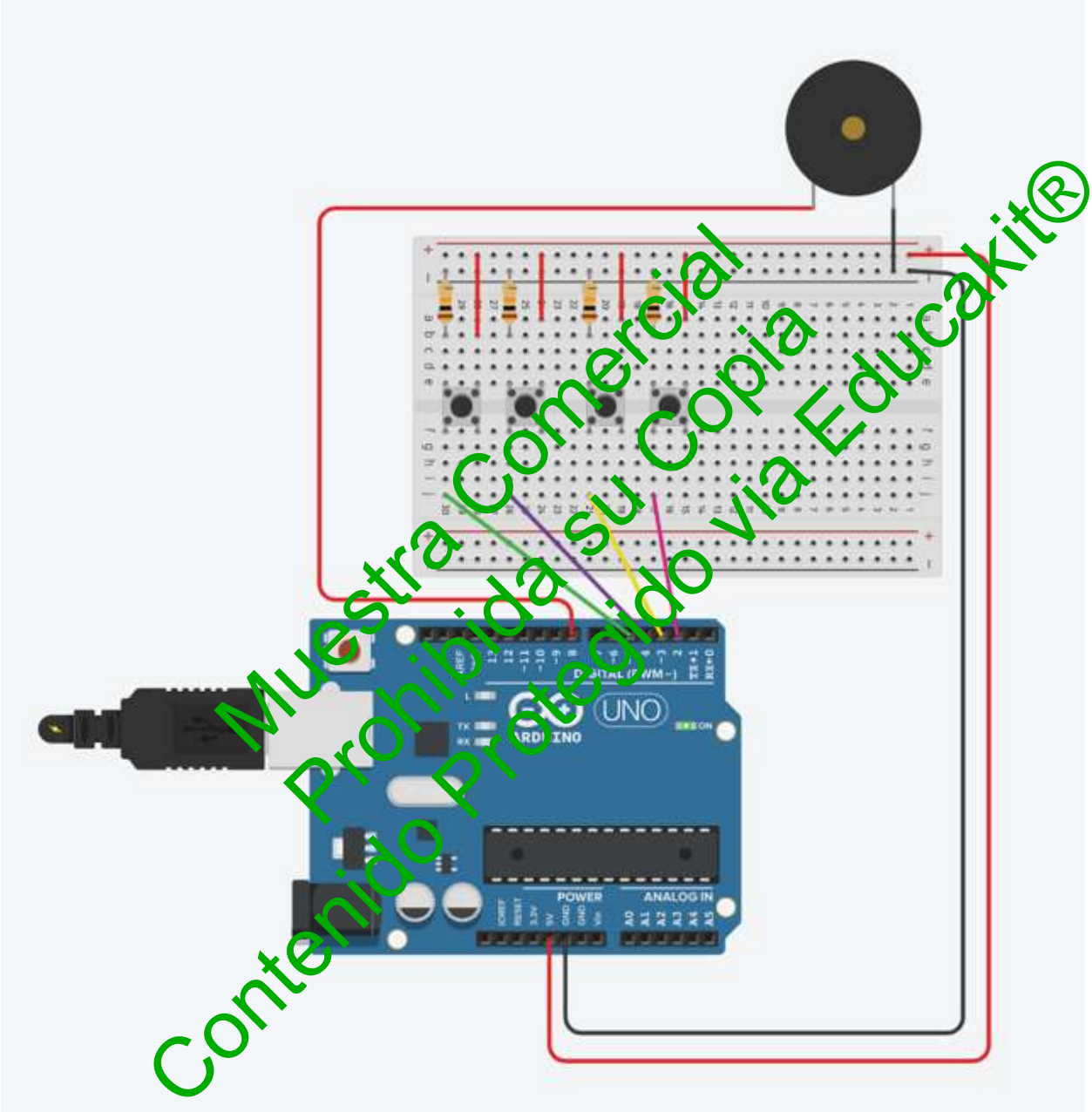
1. Recrea una melodía que tenga un buen sonido en el Arduino, por ejemplo, la melodía de Mario Bros, Star Wars o algún clásico que se pueda reproducir de la mejor forma posible, considerando las limitaciones del timbre del Arduino.

Copyright © Todos los derechos reservados a Prodelab limitada.

TALLER 5: 🎶 Música con Arduino

En este taller se trabajará para lograr programar un buzzer con sonidos diferentes desde varios pulsadores.

Montaje



Muestra Comercial
Prohibida su Copia
Contenido Protegido via Educakit®

Copyright © Todos los derechos reservados a Prodelab limitada.

Materiales

• 4 botones • Resistencias 10 kΩ • Cables conectores • Buzzer	• Placa Arduino • Cable conector USB Arduino-PC • Protoboard

I. Teclado

a. Introduce el siguiente programa en el IDE de Arduino:

```
int valorPulsadorA = 0;
int valorPulsadorB = 0;
int valorPulsadorC = 0;
int valorPulsadorD = 0;

void setup() {
pinMode(2, INPUT);
pinMode(3, INPUT);
pinMode(4, INPUT);
pinMode(5, INPUT);
pinMode(8, OUTPUT);
}
void loop() {
valorPulsadorA = digitalRead(2);
valorPulsadorB = digitalRead(3);
valorPulsadorC = digitalRead(4);
valorPulsadorD = digitalRead(5);
if (valorPulsadorA == HIGH){

tone(8,262);
}
else if (valorPulsadorB == HIGH){tone(8,294);}
else if (valorPulsadorC == HIGH){tone(8,330);}
else if (valorPulsadorD == HIGH){tone(8,349);}

else {
noTone(8);
}
}
```

Muestra Comercial
Prohibida su Copia
Contenido Protegido via Educakit®

Copyright © Todos los derechos reservados a Prodelab limitada.

b. Describe brevemente el funcionamiento del circuito.

¿Qué valores son necesarios modificar para cambiar las notas entregadas por el buzzer?

c. ¿Qué es necesario modificar para agregar una nueva nota?

Muestra Comercial
Prohibida su Copia
Contenido Protegido via Educakit®

TAREA

Modifica el código para que cada pulsador corresponda a una nota definida; como Sol, La, Do, Mi, entre otras. Utiliza notas con las cuales puedas crear una melodía sencilla.

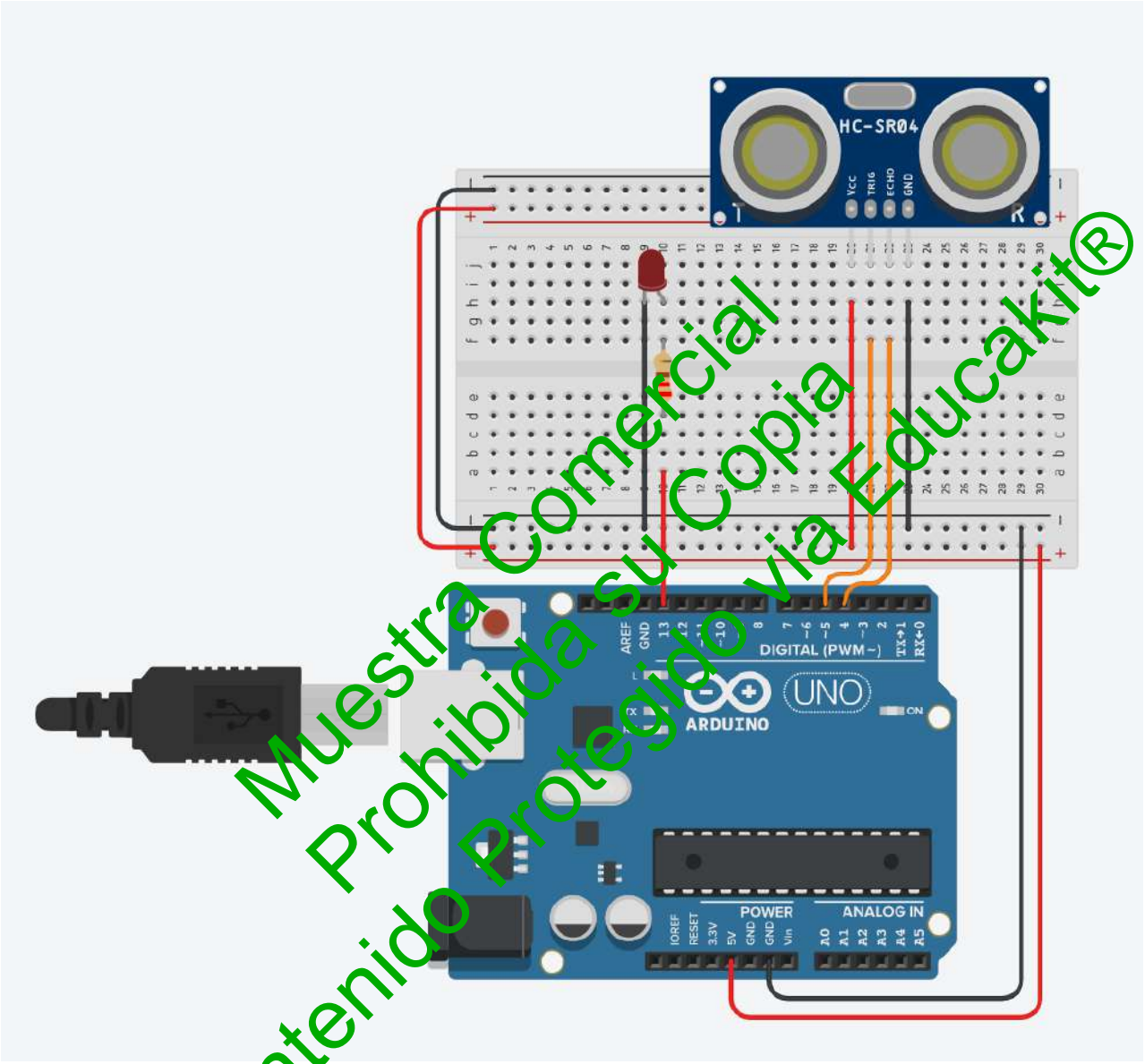
Mejora el código dándole el nombre de la nota al pulsador: En vez de utilizar "Pulsador A", utiliza el nombre de la nota musical que ocupaste.

Copyright © Todos los derechos reservados a Prodelab limitada.

TALLER 6: Ultrasonido

En este taller se trabajará para lograr programar un sensor de ultrasonido.

🔧**Montaje**



Muestra Comercial
Prohibida su Copia
Contenido Protegido via Educakit®

Copyright © Todos los derechos reservados a Prodelab limitada.

Materiales

• Sensor ultrasonido HC-SR04 • Resistencias 220 Ω • Cables conectores • LED	• Placa Arduino • Cable conector USB Arduino-PC • Protoboard

I. Ultrasonido

a. Introduce el siguiente programa en el IDE de Arduino :

```
const int Trigger = 5;   //Pin digital 5 para el Trigger del sensor
const int Echo = 4;   //Pin digital 4 para el Echo del sensor

void setup() {
  pinMode (13, OUTPUT);
  Serial.begin(9600); // iniciamos la comunicación
  pinMode(Trigger, OUTPUT);
  pinMode(Echo, INPUT);
  digitalWrite(Trigger, LOW); //Inicializamos el pin con 0
}

void loop()
{
  long t; //tiempo que demora en llegar eco
  long d; //distancia en centimetros
  digitalWrite(Trigger, HIGH);
  delayMicroseconds(10);          //Enviamos un pulso de 10 microsegundos
  digitalWrite(Trigger, LOW);
  tiempo = pulseIn(Echo, HIGH); //obtenemos el ancho del pulso
  distancia = (tiempo/2)*0.034; //escalamos el tiempo a una distancia en cm
 if (d >=20){   //si la distancia es mayor a 20cm
 digitalWrite (13, LOW);
 }
  else {
  digitalWrite(13,HIGH);
  }
  }
```

Muestra Comercial
Prohibida su Copia
Contenido Protegido via Educakit®

Copyright © Todos los derechos reservados a Prodelab limitada.

b. Observa lo que ocurre al cargar el código junto los comentarios de éste y responde las preguntas.

b.1 ¿Qué ocurre si cambio el 20 ocupado en " if (d >=20)"?

b.2 ¿Y si cambiamos el ">" por un "<" del " if (d >=20)"?

b.3 Si en el código original cambiamos " if (d >=20)" por "if (d >= 20 || d <= 10)" ¿Qué ocurre con el led?

b.4 ¿Para qué es útil el código de la pregunta anterior?

Muestra Comercial
Prohibida su Copia
Contenido Protegido via Educakit®

Copyright © Todos los derechos reservados a Prodelab limitada.

TAREA 📖

1. Realice un código donde una luz led se encienda cuando un objeto se encuentre en un rango de 50 cm y 250 cm de distancia.
2. Conecta un led más en el circuito y realice la siguiente actividad:

   a. Un led verde que encienda la luz cuando la distancia sea mayor de 200 cm
   b. Un led amarillo que parpadee cuando un objeto llegue a una distancia menor a 200 cm

3. Conecte un parlante donde encienda cuando la distancia sea menor a 200 cm y que funcione como una alarma, junto con el parlante también debe encender un led de color rojo.

Muestra Comercial
Prohibida su Copia
Contenido Protegido via Educakit®

Copyright © Todos los derechos reservados a Prodelab limitada.

TALLER 7: Servomotor

Un **servomotor** es un dispositivo robotizado que puede moverse a posiciones específicas con alta precisión. Funciona recibiendo señales de control que dictan a qué ángulo moverse. Esta característica los hace ideales para tareas que requieren precisión, como mover un brazo robotizado o ajustar la dirección de un sensor.

Ejemplo: Imagina un brazo mecánico en una línea de ensamblaje que necesita girar exactamente 90 grados para agarrar una pieza. Ese movimiento controlado es realizado por un servomotor.

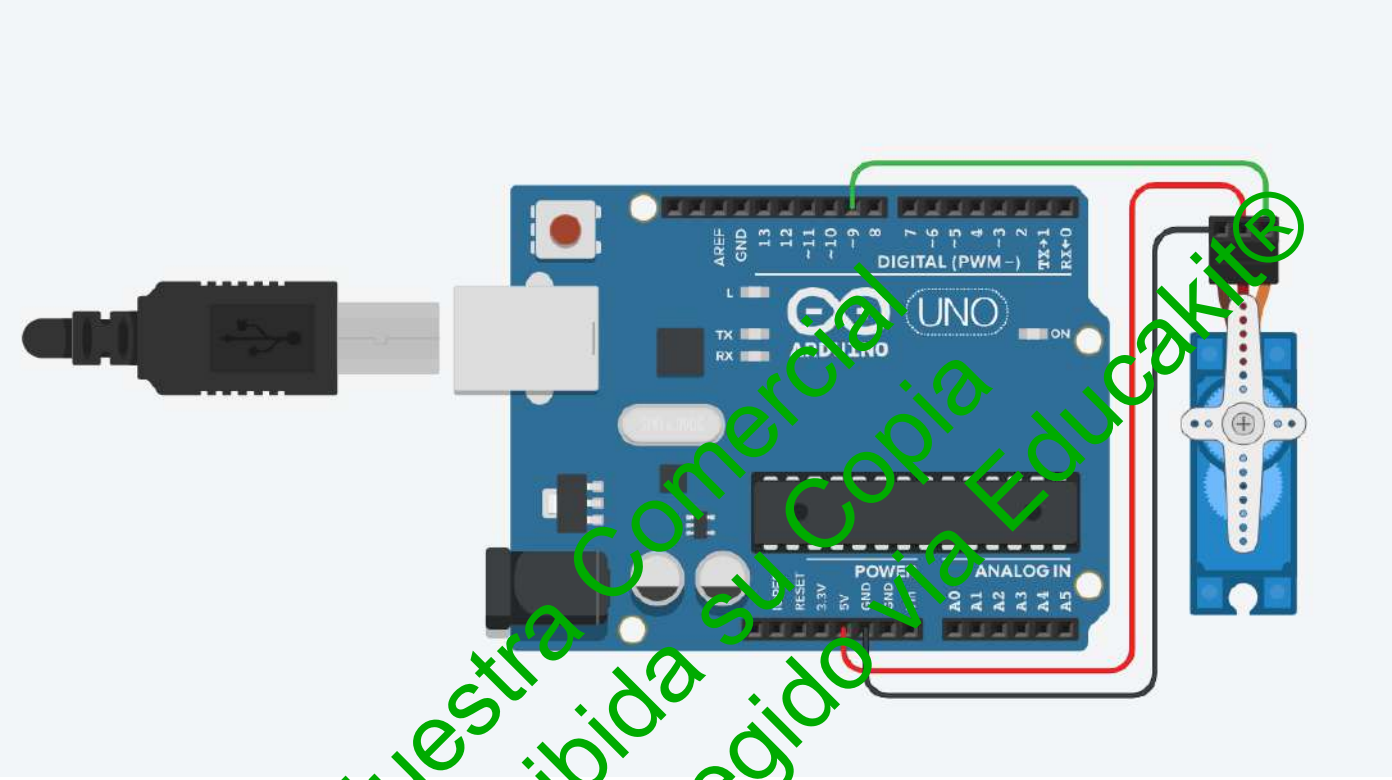
Muestra Comercial
Prohibida su Copia
Contenido Protegido via Educakit®

Copyright © Todos los derechos reservados a Prodelab limitada.

📦 Materiales

Servomotor de giro restringido (FS90) | Arduino | Cables Conectores M-H

🔧 Montaje



Muestra Comercial
Prohibida su Copia
Contenido Protegido Via Educakit®

Copyright © Todos los derechos reservados a Prodelab limitada.

I. Servomotor de giro restringido (FS90)

a. Introduce el siguiente programa en el IDE de Arduino:

```
#include <Servo.h>
Servo servoMotor; //crea un objeto Servo para controlar al servomotor.
void setup() {
  servoMotor.attach(9); // vincula el servo en el pin 9 de la placa al objeto creado myservo the servo.
}
void loop() {
servoMotor.write(0);  // Le dice al servo que vaya a la posición "0°
  delay(3000);
servoMotor.write(90);
  delay(3000);
servoMotor.write(180);
  delay(3000);
}
```

b. Describe brevemente el funcionamiento del circuito.

c. Modifique los valores dentro de las funciones servoMotor.write(0); por valores mayores a 180. ¿Qué ocurre en cada caso?

Muestra Comercial
Prohibida su Copia
Contenido Protegido via Educakit®

Copyright © Todos los derechos reservados a Prodelab limitada.

II. Servomotores de Rotación Continua (FS90R)

Después de explorar el servomotor FS90, vamos a sumergirnos en el mundo de los **servomotores de rotación continua**, como el FS90R. Estos servomotores son una variante especial, diseñados no para detenerse en un ángulo específico, sino para girar de manera **continua**. Esta habilidad los convierte en herramientas ideales para aplicaciones donde el **movimiento constante** es esencial, como en las ruedas de un robot.

A **diferencia** de los servomotores estándar, los servomotores de rotación continua como el FS90R utilizan los **valores de programación** de manera distinta. En ellos, los valores típicamente usados para definir ángulos en servomotores convencionales ahora controlan la **dirección y la velocidad del giro**. Es crucial comprender que aunque los números sean los mismos (0, 95, 180), **su interpretación y efecto** en un servomotor modificado como el FS90R son completamente diferentes. Esta comprensión es clave para adaptarse y aprovechar al máximo las capacidades únicas de los servomotores de rotación continua.

a. Con el mismo código utilizado anteriormente, observa y describe brevemente el funcionamiento del circuito.

b. Modifique los valores dentro de las funciones servoMotor.write(0); servoMotor.write(90); servoMotor.write(180);

¿Qué ocurre al modificar los valores?

Muestra Comercial
Prohibida su Copia
Contenido Protegido via Educakit®

Copyright © Todos los derechos reservados a Prodelab limitada.

c. ¿A qué corresponden los valores reemplazados?

d. ¿Cuál es la diferencia entre ambos servomotores utilizados?

Muestra Comercial
Prohibida su Copia
Contenido Protegido via Educakit®

Funciones de un Servomotor de Giro Continuo (FS90R)

🚫 **Detener Servomotor 360**

Para detener un servomotor de 360 grados con Arduino, se utiliza la siguiente instrucción:

```
servo.write(95);
```

Esto significa que detener un servomotor de 360 grados es equivalente a programar un servomotor estándar en 95 grados.

🔄 **Sentido Anti-Horario**

Para mover un servomotor de 360 grados en sentido antihorario, se utiliza la siguiente instrucción:

```
servo.write(180);
```

Esto equivale a programar un servomotor estándar en 180 grados.

🔃 **Sentido Horario**

Para mover un servomotor de 360 grados en sentido horario, se utiliza la siguiente instrucción:

```
servo.write(0);
```

Esto equivale a programar un servomotor estándar en 0 grados.

Copyright © Todos los derechos reservados a Prodelab limitada.

III. Controlando barridos de giro

a. Con un motor de **giro restringido (FS90)** ingresa el siguiente código en el IDE de Arduino:

```
#include <Servo.h>
Servo myservo;
int pos = 0;

void setup() {
 myservo.attach(9);
}

void loop() {
 for (pos = 0; pos <= 180; pos += 1) {
  myservo.write(pos);
  delay(15);
 }

 for (pos = 180; pos >= 0; pos -= 1) {
  myservo.write(pos);
  delay(15);
 }
}
```

Muestra Comercial
Prohibida su Copia
Contenido Protegido via Educakit®

Copyright © Todos los derechos reservados a Prodelab limitada.

En caso de que lo necesites… Mismo código, con comentarios:

```
#include <Servo.h>
Servo myservo; // Objeto para manejar el servo.

int pos = 0; // Posición actual del servo en grados.

void setup() {
  myservo.attach(9); // Asigna el servo al pin 9.
}

void loop() {

 // Mueve el servo de 0 a 180 grados.
  for (pos = 0; pos <= 180; pos += 1) {
    myservo.write(pos); // Establece la nueva posición.
    delay(15); // Breve pausa para permitir el movimiento.
  }

  // Regresa el servo de 180 a 0 grados.
  for (pos = 180; pos >= 0; pos -= 1) {
    myservo.write(pos); // Establece la nueva posición.
    delay(15); // Breve pausa para permitir el movimiento.
  }
}
```

Muestra Comercial
Prohibida su Copia
Contenido Protegido via Educakit®

b. ¿Cuál es la diferencia entre los códigos anteriores y el actual?

Copyright © Todos los derechos reservados a Prodelab limitada.

c. **TAREA** 📖

1. Controlando el ángulo de giro, construya un cronómetro que mida al menos 15 segundos.

2. Controlando el barrido de giro, construya un cronómetro que mida al menos 15 segundos y luego se devuelva midiendo el mismo tiempo.

3. Incluye dos pulsadores en el montaje, donde:

   a. Con un pulsador el servomotor se mueva a 90°

   b. Con un segundo pulsador el servomotor se mueva a 0°

4. Realice un montaje donde se conecte un sensor de ultrasonido y:

   a. Si la distancia al sensor es menor de 100 cm el servomotor se mueva en un ángulo de 90°

   b. Si la distancia al sensor es mayor que 100 cm el servomotor mantenga en un ángulo de 0°

Muestra Comercial
Prohibida su Copia
Contenido Protegido via Educakit®

Copyright © Todos los derechos reservados a Prodelab limitada.

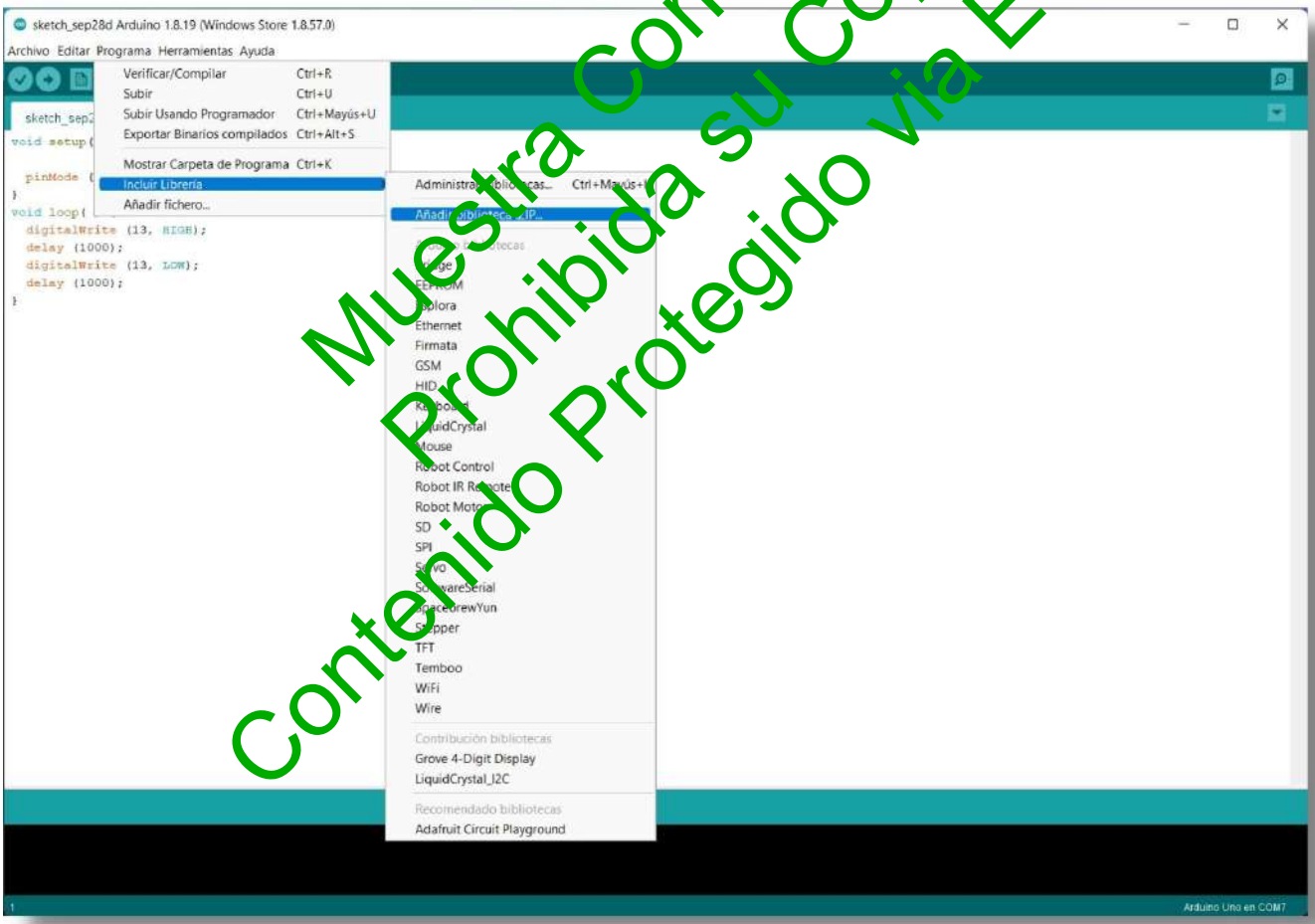
📚 Aprendiendo a instalar librerías de distintos dispositivos.

Algunos componentes electrónicos y dispositivos requieren de una **librería** para ser usados con Arduino. Una librería es un conjunto de código que se utiliza para controlar y **programar** ese **componente** específico. La librería es necesaria porque el componente tiene una función o características **especiales que no están incluidas** en la librería de Arduino básica. Por lo tanto, para usar el componente de manera efectiva y controlar todas sus funciones, se debe agregar su librería correspondiente al programa de Arduino.

Pasos

- Descarga la librería de Arduino para para este dispositivo: Prodelab.cl/Arduino, o buscala en internet como "LCD 1602 I2C Library".

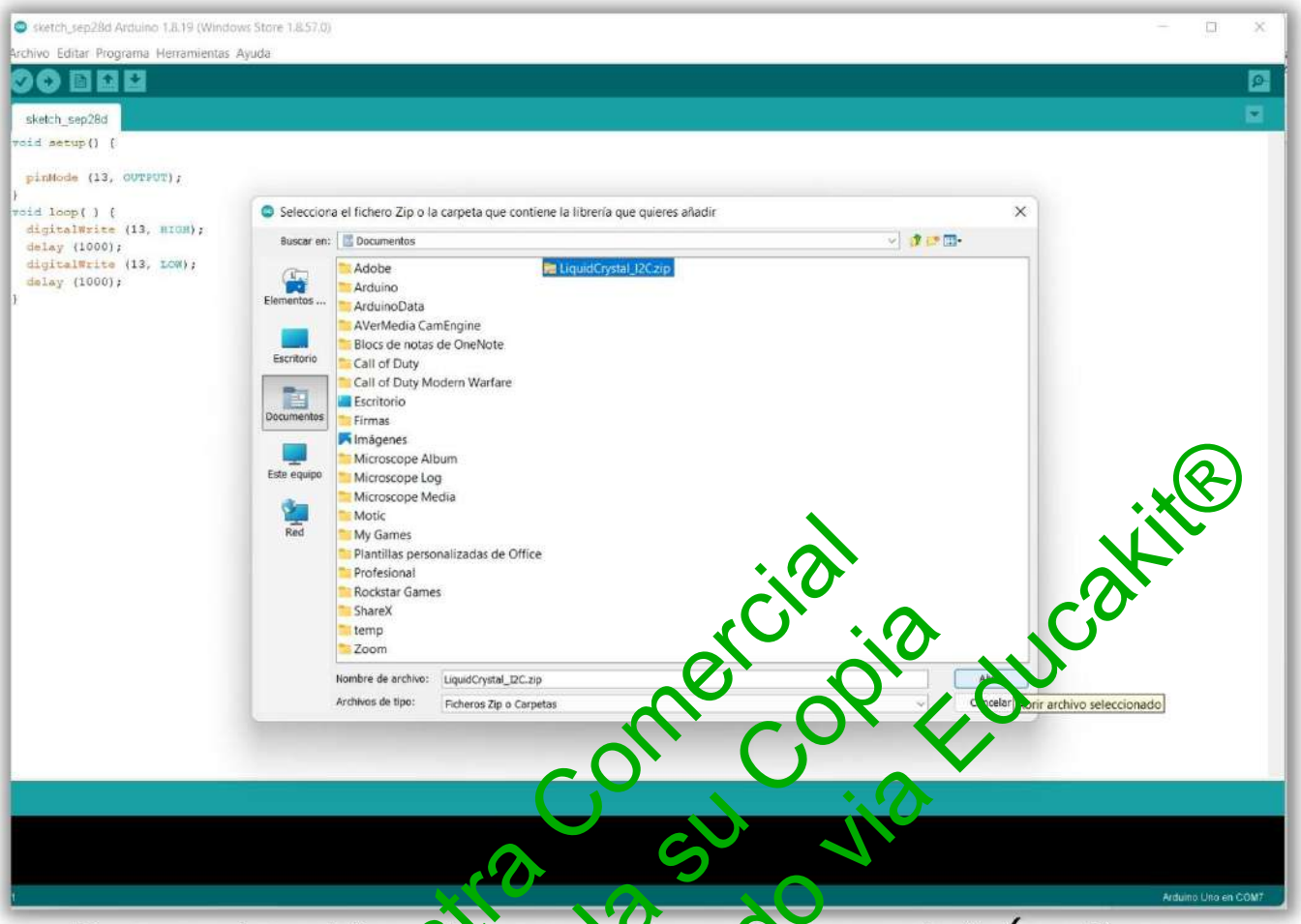
- Instala de esta forma:



Programa > Incluir Librería > Añadir biblioteca en .ZIP

Muestra Comercial
Prohibida su Copia
Contenido Protegido via Educakit®

Copyright © Todos los derechos reservados a Prodelab limitada.



Busca el archivo descargado y click en abrir (no tienes que descomprimirlo)

Muestra Comercial
Prohibida su Copia
Contenido Protegido via Educakit®

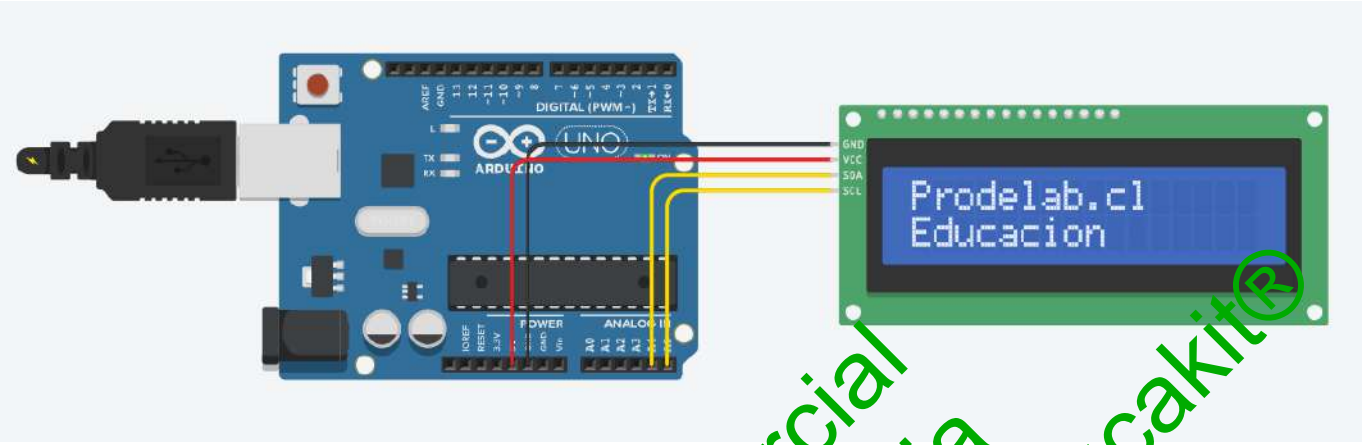
En prodelab.cl/arduino podrás descargar las librerías necesarias para todos los componentes extras del kit.

Copyright © Todos los derechos reservados a Prodelab limitada.

TALLER 8: Pantalla LCD

📦**Materiales:** Pantalla LCD I2C | Cable Jumper Macho-Hembra

🔧**Montaje**



I. Mi primer mensaje

a. Realiza el montaje e Introduce el siguiente programa en el IDE de Arduino:

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 16, 2); // Configura el LCD con la dirección I2C y el tamaño
                                    // hasta a 0x20 si es necesario

void setup() {
  lcd.init(); // Inicializa la pantalla LCD
  lcd.backlight(); // Enciende la luz de fondo
}

void loop() {
  lcd.setCursor(0,0); // Establece el cursor en la esquina superior izquierda
  lcd.print("Hola!");
  delay(1000);
  lcd.clear();
  delay(1000);
}
```

Muestra Comercial
Prohibida su Copia
Contenido Protegido via Educakit®

Copyright © Todos los derechos reservados a Prodelab limitada.

Problemas Comunes (Troubleshooting)

- Si no ves el mensaje y la pantalla si prende:

  - Cambia (0x27, 16, 2) por (0x20 16, 2).
  - Con un destornillador, gira levemente la tuerca de atrás hacia un lado u otro. Se minucioso al regular esto.

- Se interrumpe la conexión usb y el arduino se apaga:

  - Haz las conexiones y luego enchufa el arduino a tu PC. No al revés

b. Experimenta con la Posición del Texto:

- Cambia lcd.setCursor(0,0); por lcd.setCursor(2,0); y observa la nueva posición del texto.
- Luego cambia a lcd.setCursor(4,1); para ver cómo el texto se mueve a otra línea.

b.1 ¿Para qué sirve el primer número del código modificado anteriormente?

b.2 ¿Para qué sirve el segundo número del código modificado anteriormente?

Muestra Comercial
Prohibida su Copia
Contenido Protegido via Educakit®

Copyright © Todos los derechos reservados a Prodelab limitada.

c. ¿Para qué sirve el código " lcd.clear();"?

d. ¿Para qué sirve el código "lcd.print("Hola!");"?

Muestra Comercial
Prohibida su Copia
Contenido Protegido via Educakit®

Copyright © Todos los derechos reservados a Prodelab limitada.

II. String

En Arduino, un String es una cadena de texto compuesta por caracteres individuales, cada uno con una posición específica. Imagina un String como un tren de letras y espacios, donde cada vagón es un carácter. Por ejemplo, en "Hola Mundo", **'H'** es el primer vagón (posición 0), **'o'** es el segundo (posición 1), y así sucesivamente. Esto significa que podemos acceder o modificar cada carácter por separado, usando su posición en el String, lo cual es clave en tareas como mostrar mensajes en pantallas o almacenar y manipular información textual.

Muestra Comercial
Prohibida su Copia
Contenido Protegido Via Educakit®

Copyright © Todos los derechos reservados a Prodelab limitada.

a. Ingresa el siguiente código a tu IDE.

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

// Inicializa la pantalla LCD con módulo I2C
LiquidCrystal_I2C lcd(0x27, 16, 2); // Ajusta a 0x20 si es necesario

String texto_fila= "Programar texto en movimiento"; // Declaración de un String

void setup() {
 lcd.init();            // Inicializa la pantalla LCD con I2C
 lcd.backlight();       // Enciende la luz de fondo
}

void loop() {

 for(int i=0; i<=29; i++)
 {
  String texto = texto_fila.substring(i-1); // Utiliza substring para manipular el String
  lcd.clear();
  lcd.setCursor(0,0);          // Establece el cursor en la esquina superior izquierda
  lcd.print(texto);
  delay(300);
 }
}
```

Muestra Comercial
Prohibida su Copia
Contenido Protegido via Educakit®

Importante 💡

- Longitud del String: En el ejemplo del movimiento del texto, el String "Programar texto en movimiento" tiene 29 caracteres (incluyendo espacios).
- Funcionamiento del Bucle for: El bucle for de 0 a 29 permite recorrer cada carácter del String. En cada iteración del bucle, se muestra una parte del String en la pantalla LCD.
- Relación entre Numeros y Letras: Cada número en el bucle for representa un índice en el String. El método substring(i-1) se utiliza para obtener una parte del String desde el índice i-1 hasta el final. Esto hace que el texto se desplace en la pantalla, mostrando diferentes secciones del String en cada iteración.

Copyright © Todos los derechos reservados a Prodelab limitada.

b. Modifica el código anterior el " for(int i=0; i<=29; i++)" por " for(int i=29; i>=0; i--)" ¿Qué ocurre con el texto de la pantalla?

c. Observa el código y con lo aprendido hasta ahora, ¿qué debes modificar para que el texto se mueva más rápido?

d. Modifica el código para que el texto se mueva a la derecha pero partiendo de la columna inferior.

e. Modifica el código para que el texto se mueva a la izquierda, pero sin ocupar la primera fila de la pantalla.

Muestra Comercial
Prohibida su Copia
Contenido Protegido via Educakit®

f. Introduce el siguiente programa en el IDE de Arduino →

Copyright © Todos los derechos reservados a Prodelab limitada.

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);
unsigned char seg = 0, min = 0, hora = 0;

void setup() {
  lcd.init();
  lcd.backlight();
}

void loop() {
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("RELOJ");

  lcd.setCursor(0, 1);
  if (hora < 10) lcd.print('0');
  lcd.print(hora);
  lcd.print(':');

  if (min < 10) lcd.print('0');
  lcd.print(min);
  lcd.print(':');

  if (seg < 10) lcd.print('0');
  lcd.print(seg);

  delay(1000);
  seg++;

  if (seg > 59) {
    seg = 0;
    min++;
  }

  if (min > 59) {
    min = 0;
    hora++;
  }

  if (hora > 23) {
    hora = 0;
  }
}
```

Muestra Comercial
Prohibida su Copia
Contenido Protegido via Educakit®

Copyright © Todos los derechos reservados a Prodelab limitada.

g. ¿Qué observas en la pantalla LCD cuando cargas y ejecutas este código? Describe cómo se comportan los números en la pantalla.

h. ¿Cómo modificarías el código para agregar una función de horas al reloj? Piensa en los límites de tiempo que debe tener un reloj.

i. El código actual muestra 'RELOJ' en la pantalla. ¿Cómo lo logra? ¿En qué parte del código se hace esto?

Muestra Comercial
Prohibida su Copia
Contenido Protegido via Educakit®

TAREA 📖

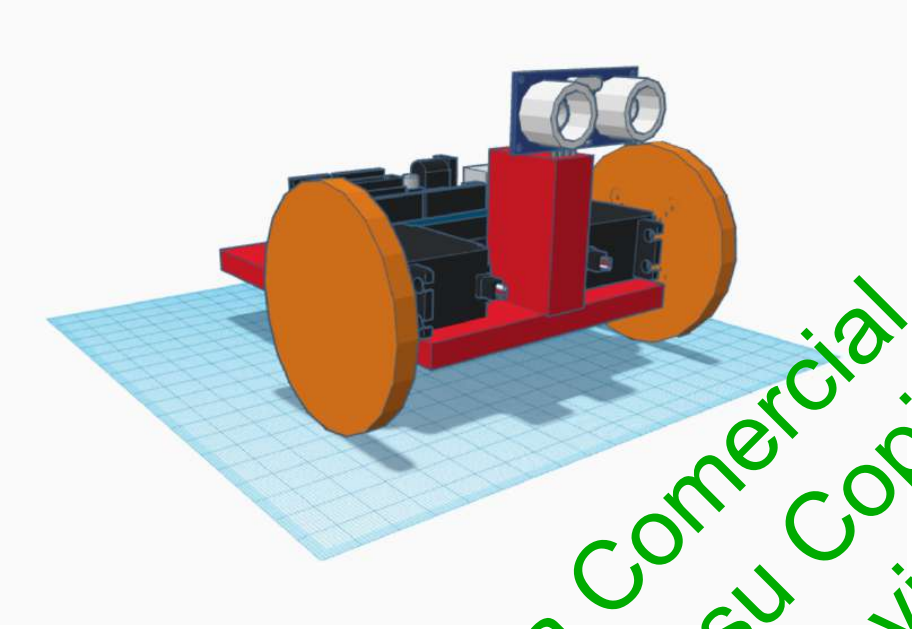
1. Conecta dos botones junto a la pantalla LCD e investiga en internet como poder realizar un reloj donde con un botón se modifique la hora y con otro botón se modifiquen los minutos.
2. Conecta el sensor de ultrasonido y realiza un código donde la distancia detectada por el ultrasonido se vea reflejada en la pantalla.

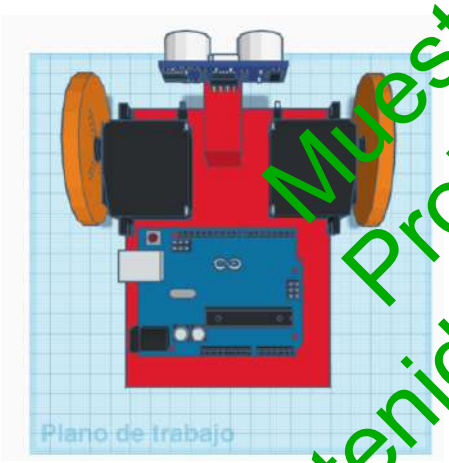
Copyright © Todos los derechos reservados a Prodelab limitada.

TALLER 9: Auto Robot I

En este taller trabajaremos la realización de un auto robot.

🔧 **Montaje sugerido**





Muestra Comercial
Prohibida su Copia
Contenido Protegido via Educakit®

Copyright © Todos los derechos reservados a Prodelab limitada.

🛠️ Construye tu Vehículo Arduino de Dos Ruedas

¡Ha llegado el momento de poner a prueba tu ingenio!

Ya has trabajado con el 'cerebro', que es Arduino, y has experimentado con su 'sistema nervioso' y su 'musculatura' a través de diversos componentes. Ahora, te enfrentas al desafío final: construir un vehículo de dos ruedas utilizando servomotores FS90R. ¿Estás listo para dar vida a tu propio vehículo Arduino casero?"

Te mostramos algunos ejemplos:

Muestra Comercial
Prohibida su Copia
Contenido Protegido via Educakit®

Aunque los ejemplos proporcionados incluyen varias conexiones, es crucial comenzar con lo básico. Asegurémonos primero de que el robot se mueva correctamente. Una vez logrado esto, podremos avanzar agregando elementos adicionales como el sensor HC-SR04, LEDs, fotoresistores y resistencias.

📦 Materiales

• Pantalla LCD 16x2 • Potenciómetro • Resistencias • Cables conectores • Sensor de ultrasonido • 2 servomotores de giro continuo • 2 ruedas	• Placa Arduino • Cable conector USB Arduino-PC • Protoboard • Porta Pilas • Pilas • 2 Pulsadores

Copyright © Todos los derechos reservados a Prodelab limitada.

Consideraciones para el Montaje

- Con Materiales reutilizables realiza una base como muestra la imagen, esta debe ser del tamaño necesario para que pueda sostener un porta pilas, la protoboard, los servomotores, el ultrasonido y en un futuro una pantalla LCD.

- La base debe ser un material resistente, cartón piedra, plástico, como por ejemplo un pote viejo de comida, parte de una caja de zapatos, entre otros Materiales.

- Debes considerar que los servomotores solo manejan las ruedas de adelante del robot, la rueda de atrás puede rodar, pero no posee programación independiente, solo debe tener la posibilidad de girar.

- Para la rueda trasera puedes ocupar esfera que poseen los envases de desodorantes roll on, ya que esta esfera es mejor porque posee una mayor resistencia una pelota de pin pon, o mejor giro que una pelota de goma.

- Es importante que los servomotores queden en la parte delantera del robot, debido a que así lograra un movimiento con menor error, ya que, si colocamos la rueda no programable adelante, esta si se desliza por la superficie hará que el movimiento programado no se realice de la mejor forma.

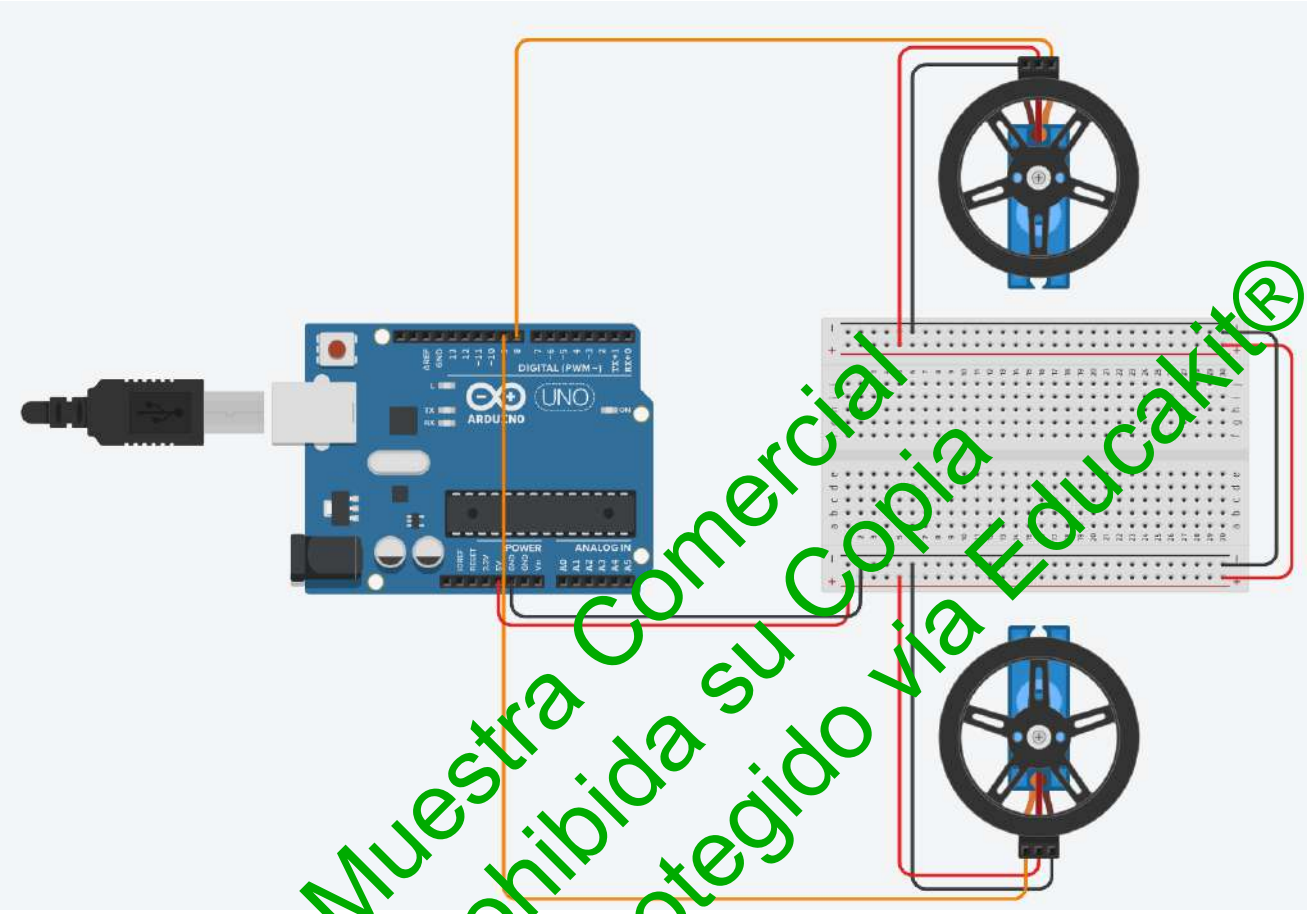
- Una vez construido el montaje, comenzamos a programar.

Muestra Comercial
Prohibida su Copia
Contenido Protegido Via Educakit®

Copyright © Todos los derechos reservados a Prodelab limitada.

I. Auto Robot

Te ofrecemos una ayudita y, al mismo tiempo, te desafiamos a crear tu propio circuito. La orientación de los servomotores dependerá de ti, deberás analizar cómo se comporta el código.



Muestra Comercial
Prohibida su Copia
Contenido Protegido via Educakit®

Copyright © Todos los derechos reservados a Prodelab limitada.

a. Conecta los dos servomotores al Arduino en el pin 8 y 9.
b. Copia el siguiente código en el IDE de Arduino.

```
#include <Servo.h>
Servo servoMotor1;
Servo servoMotor2;

void setup() {
servoMotor1.attach(8);
servoMotor2.attach(9);
}
void loop() {
servoMotor1.write(0);
servoMotor2.write(0);
delay(3000);
}
```

c. Modifica el código anterior según lo aprendido en los talleres anteriores para que los servomotores muevan el robot para adelante, atrás, derecha e izquierda.

Código: Adelante

Muestra Comercial
Prohibida su Copia
Contenido Protegido via Educakit®

Copyright © Todos los derechos reservados a Prodelab limitada.

Código: Atrás
Código: Derecha

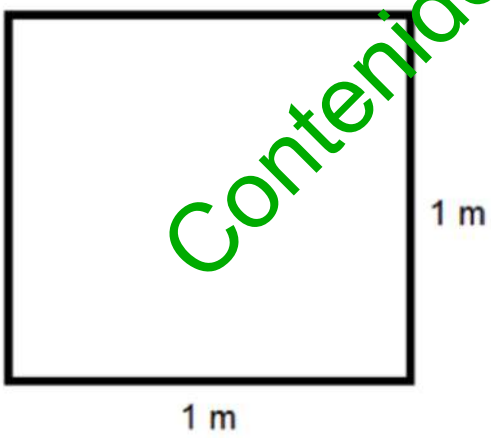
Muestra Comercial
Prohibida su Copia
Contenido Protegido via Educakit®

Copyright © Todos los derechos reservados a Prodelab limitada.

Código: Izquierda

Muestra Comercial
Prohibida su Copia
Contenido Protegido via Educakit®

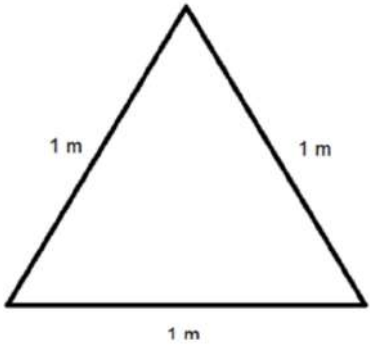
TAREA

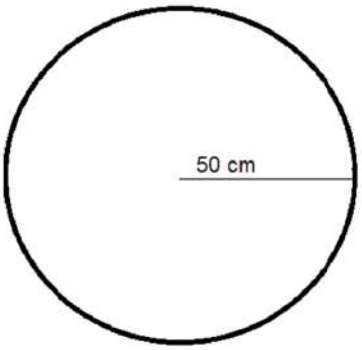


1. Programa el robot para que recorra una trayectoria cuadrada de 1m de lado.

Copyright © Todos los derechos reservados a Prodelab limitada.

2. Programa el robot para que recorra una trayectoria triangular de 1m de lado.





3. Programa el robot para que su trayectoria sea una circunferencia de 50 cm de radio, deteniéndose en el punto inicial.

4. Repita las tareas anteriores, esta vez haciendo que recorra las trayectorias hacia atrás.

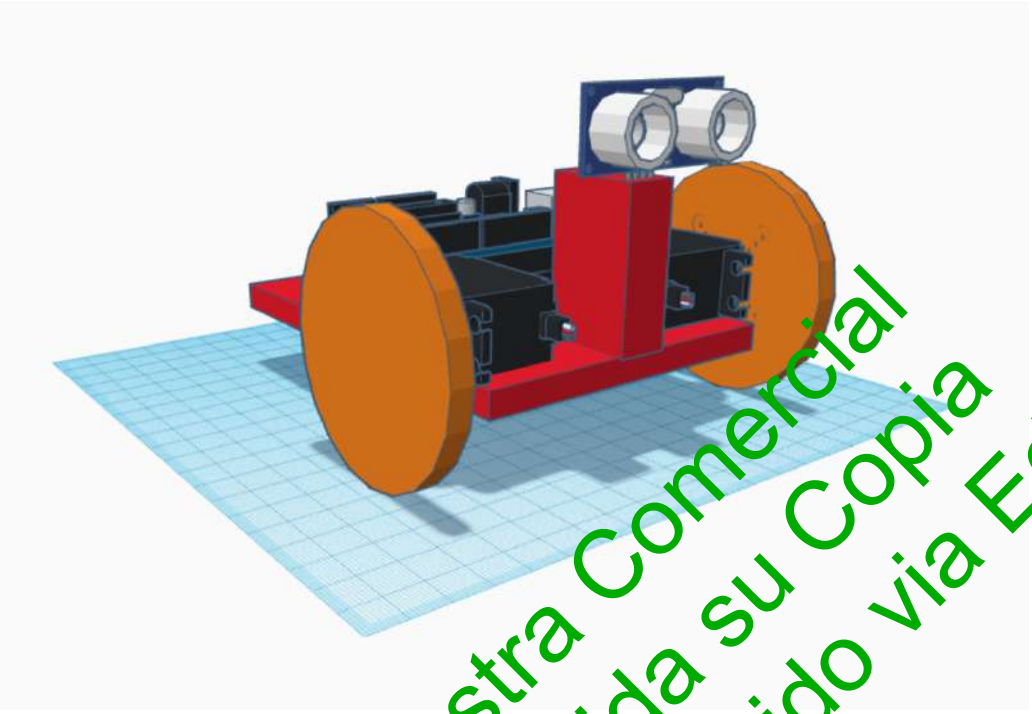
Muestra Comercial
Prohibida su Copia
Contenido Protegido via Educakit®

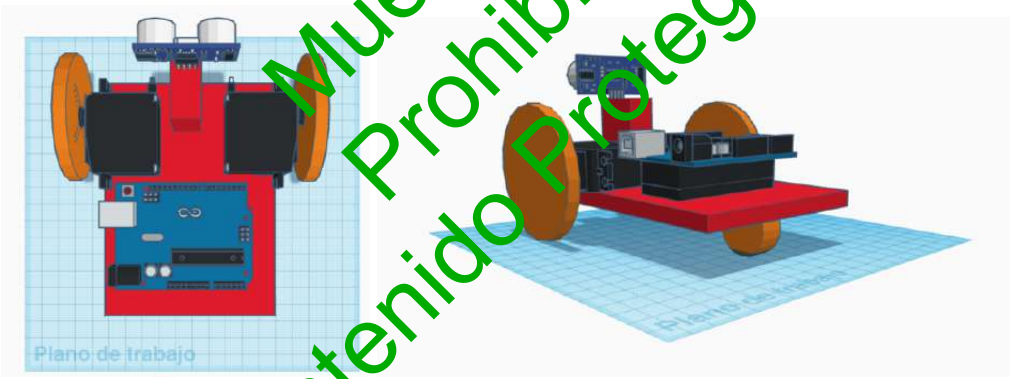
Copyright © Todos los derechos reservados a Prodelab limitada.

TALLER 10: Auto Robot II

En este taller trabajaremos la realización de un auto robot que sortee obstáculos.

Montaje





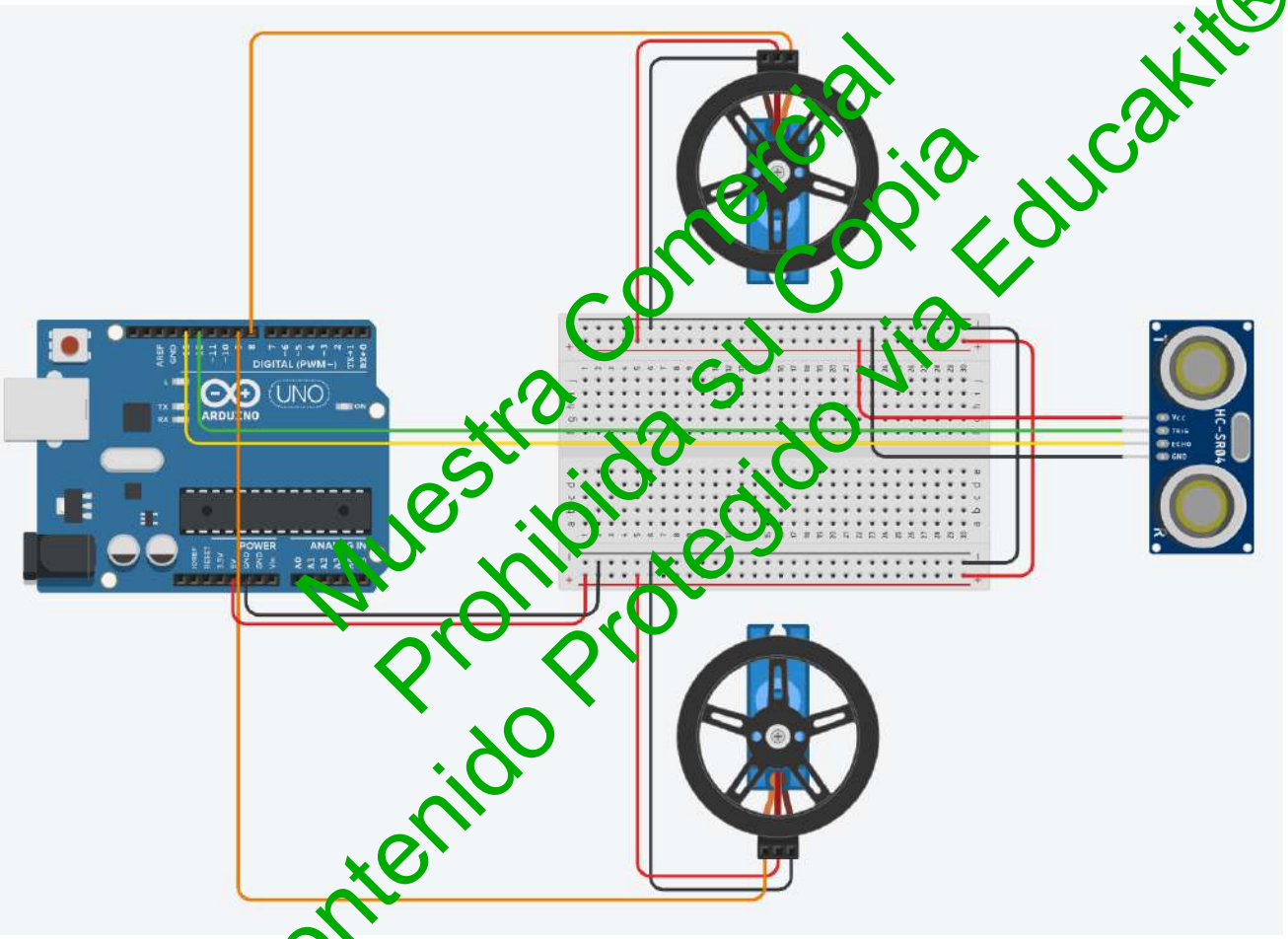
Muestra Comercial
Prohibida su Copia
Contenido Protegido via Educakit®

Copyright © Todos los derechos reservados a Prodelab limitada.

Materiales

• Pantalla LCD 16x2 • Potenciómetro • Resistencias • Sensor de ultrasonido • 2 servomotores de giro continuo • 2 ruedas	• Placa Arduino • Cable conector USB Arduino-PC • Protoboard • Porta Pilas • Pilas • Cables conectores

a. Conecta el sensor de ultrasonido en el montaje ya construido en el taller anterior.



Muestra Comercial
Prohibida su Copia
Contenido Protegido via Educakit®

b. Con los conocimientos adquiridos en talleres anteriores programa el sensor de ultrasonido para que detecte obstáculos y realice las siguientes acciones:

1. El auto se mueve hacia adelante mientras no tenga un obstáculo adelante.
2. Será considerado obstáculo cuando el auto esté frente a un objeto a 20 o menos cm de distancia.
3. Cuando el obstáculo se presente el auto deberá ir hacia atrás por 10 cm y luego girar a la derecha.

Copyright © Todos los derechos reservados a Prodelab limitada.

```
#include <Servo.h>
Servo servoMotor1;
Servo servoMotor2;

const int pinTrig = 10;
const int pinEcho = 11;

long duracion;
int distancia;

void setup() {
  servoMotor1.attach(8);
  servoMotor2.attach(9);
  pinMode(pinTrig, OUTPUT);
  pinMode(pinEcho, INPUT);
  Serial.begin(9600);
}

void loop() {
  digitalWrite(pinTrig, HIGH);
  delayMicroseconds(10);
  digitalWrite(pinTrig, LOW);

  duracion = pulseIn(pinEcho, HIGH);
  distancia = duracion * 0.034 / 2;

  if (distancia <= 20) {
    // Retroceder
    servoMotor1.write(180);
    servoMotor2.write(180);
    delay(1000); // Ajusta este tiempo para que el auto retroceda 10 cm

    // Girar a la derecha
    servoMotor1.write(0);
    servoMotor2.write(180);
    delay(500); // Ajusta este tiempo para que el giro sea adecuado
  } else {
    // Moverse hacia adelante
    servoMotor1.write(0);
    servoMotor2.write(0);
  }

  delay(100);
}
```

Muestra Comercial
Prohibida su Copia
Contenido Protegido via Educakit®

Copyright © Todos los derechos reservados a Prodelab limitada.

I. Conecte la pantalla LCD al robot y programe para que pueda dar la bienvenida al encender el auto y mostrar la distancia del primer obstáculo que él pueda percibir.

Troubleshooting Comunes 💡

- **Ajuste de Velocidad:** Si un motor gira más rápido que otro, prueba ajustar los valores en el código. Por ejemplo, usar 5 en lugar de 0 puede ayudar a equilibrar las velocidades.
- **Calibración de FS90R**: Los FS90R a veces requieren calibración. Puedes encontrar guías detalladas en línea.
- **Uso de Potenciómetros:** Para ajustes más finos, considera usar potenciómetros para controlar la velocidad de los motores.

Es importante recordar que, en la práctica, las condiciones ideales de la teoría no siempre se aplican al 100%. Por ejemplo, los cables no son conductores perfectos y pueden generar cierta pérdida de potencial eléctrico, especialmente cuando se usan varios en un mismo circuito. Esto podría resultar en pequeñas diferencias de voltaje entre los componentes, como los servomotores, causando que uno reciba levemente más voltaje que el otro. Este fenómeno es un recordatorio de la importancia de la experimentación y ajuste en proyectos de electrónica y robótica

Muestra Comercial
Prohibida su Copia
Contenido Protegido via Educakit®

Copyright © Todos los derechos reservados a Prodelab limitada.

PRODELAB
Equipamiento de Educación

Aprendizaje A través de la Experiencia

Muestra Comercial
Prohibida su Copia
Contenido Protegido vía Educakit®

Prodelab Limitada.

Ventus 1484, Concepción, Chile.

Todos los derechos reservados. 2023