

Guía de trabajo en laboratorio N° 9:

Fundamentos de programación - Arduino

Secciones:	Apellidos :
Docente: Miguel Tupac Yupanqui Alanya	Nombres :
	Fecha :/...../2020

Instrucciones: Desarrollar las actividades que indica el docente en base a la guía de trabajo que se presenta.

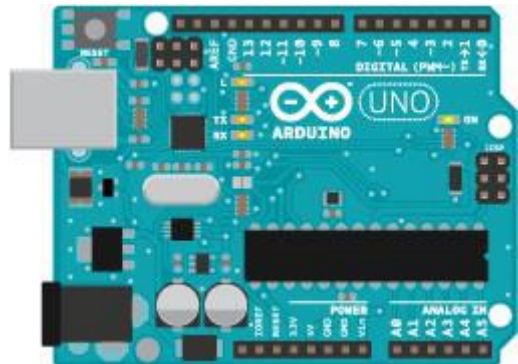
1. **Propósito:** Comprender los tipos de variables, sentencias básicas, monitor serial empleando el simulador tinkercad para Arduino.

2. Equipos, herramientas o materiales

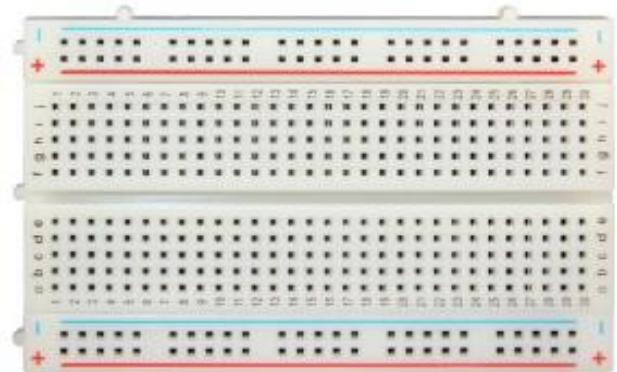
- Simulador Tinkercad
- Computador con acceso a Internet

3. Fundamento Teórico

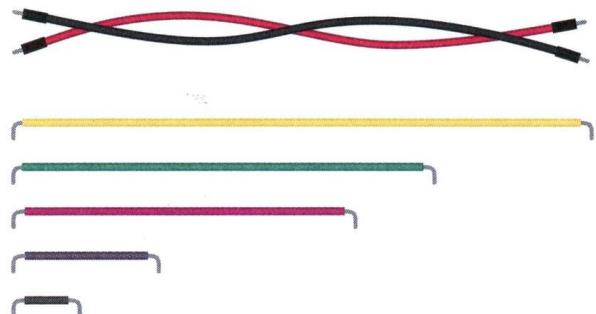
- **Arduino Uno** – La tarjeta de desarrollo del microcontrolador la cual será el corazón de tus proyectos. Descrito en clase.



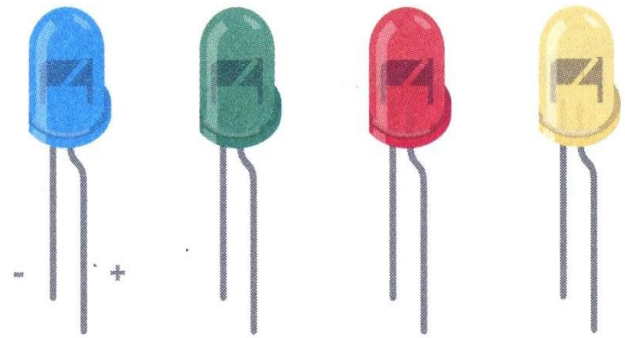
- **Protoboard (Placa de pruebas)** – Placa con filas de agujeros sobre la cual se puede conectar cables y componentes sin necesidad de usar un soldador.



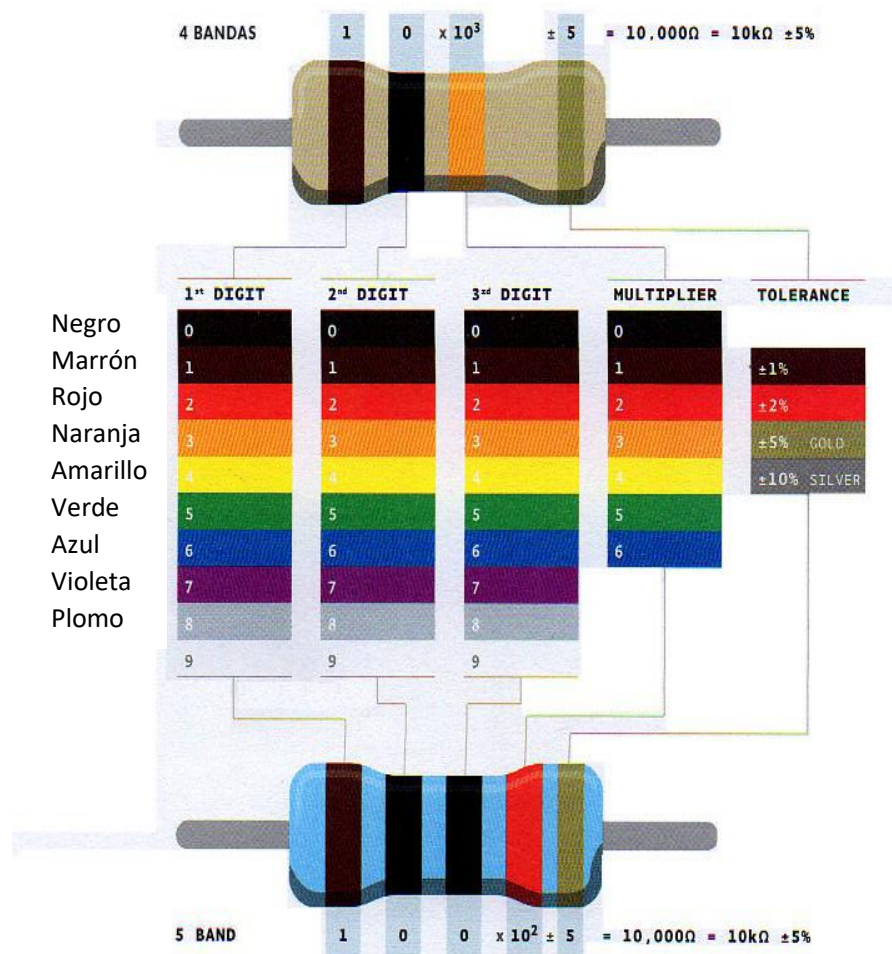
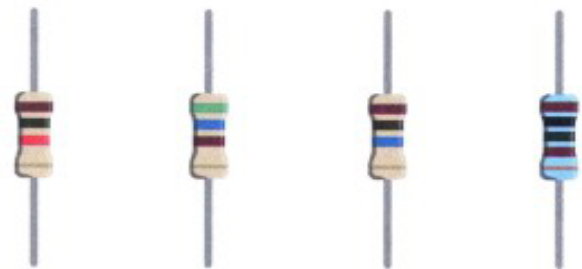
- **Cables puente** – Utilizarlos para conectar unos componentes con otros sobre la placa de prueba, y la tarjeta de Arduino.



- **Diodos Emisores de Luz (LEDs)** – Diodo que emite luz cuando la corriente lo atraviesa. Como en todos los diodos, la corriente solo fluye en un sentido a través de estos componentes. El ánodo, que normalmente se conecta al positivo de la alimentación, es generalmente el terminal más largo y el cátodo el terminal más corto.

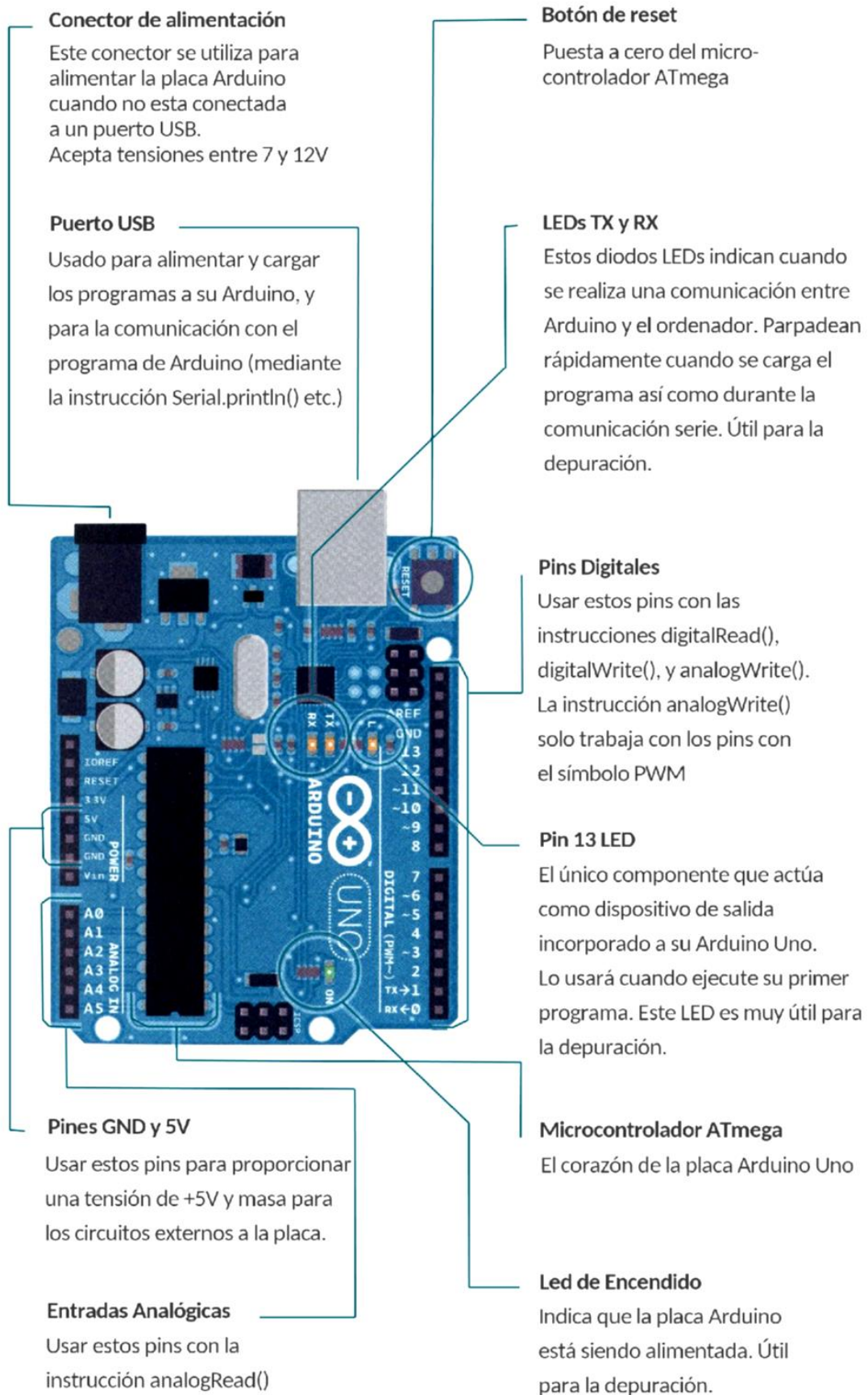


- **Resistencias** – Se opone al paso de la corriente eléctrica en un circuito, dando como resultado a un cambio en la tensión y en dicha corriente. El valor de las resistencias se mide en ohmios (Ω). Las bandas de colores en un lado de la resistencia indica su valor.



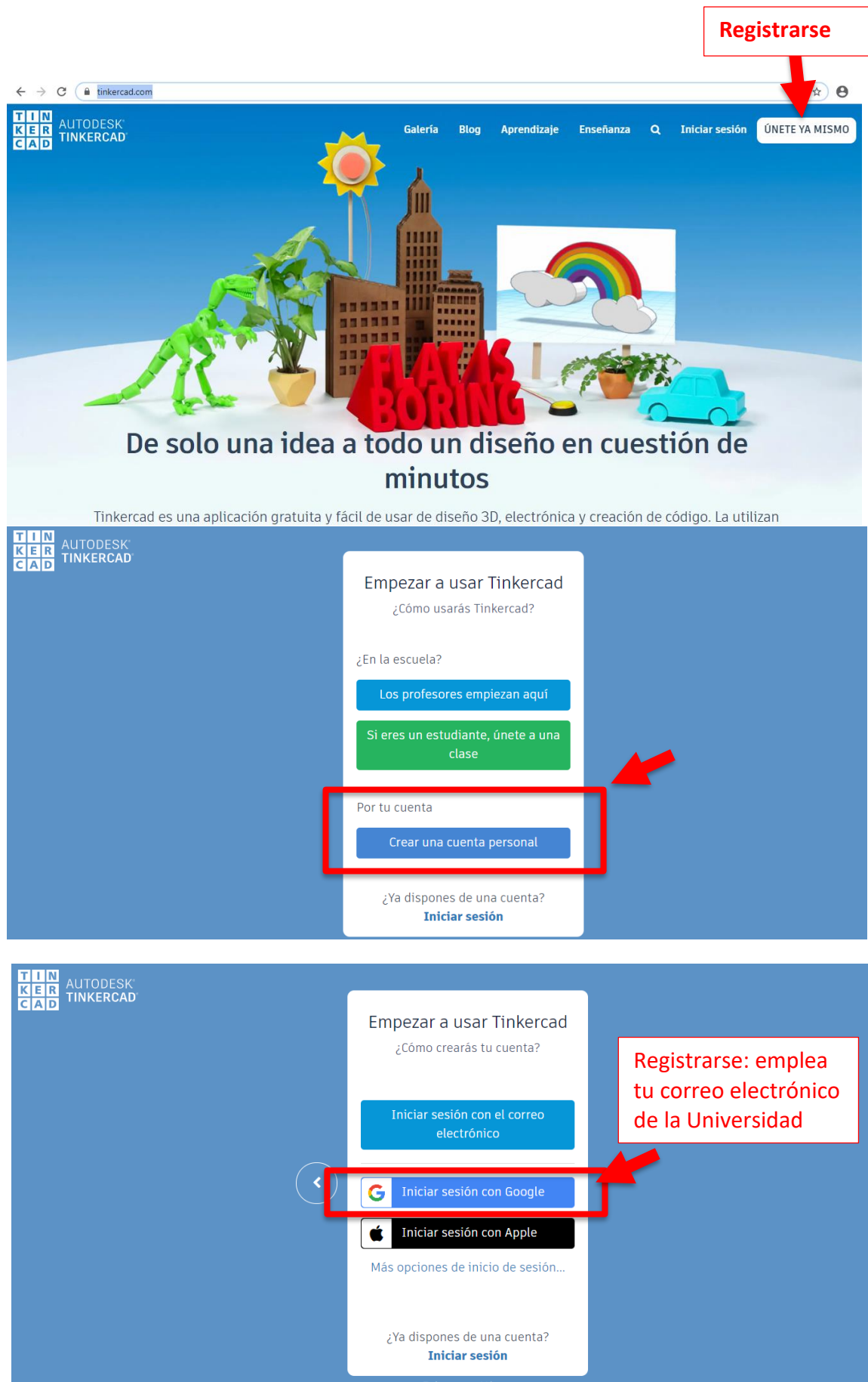
<https://www.digikey.com/es/resources/conversion-calculators/conversion-calculator-resistor-color-code-4-band>

<https://www.inventable.eu/paginas/ResCalculatorSp/ResCalculatorSp.html>



4. Procedimiento

Ingresa a <https://www.tinkercad.com/>



Registrarse

TINKERCAD AUTODESK TINKERCAD

Galería Blog Aprendizaje Enseñanza Iniciar sesión ÚNETE YA MISMO

De solo una idea a todo un diseño en cuestión de minutos

Tinkercad es una aplicación gratuita y fácil de usar de diseño 3D, electrónica y creación de código. La utilizan

Empezar a usar Tinkercad

¿Cómo usarás Tinkercad?

¿En la escuela?

Los profesores empiezan aquí

Si eres un estudiante, únete a una clase

Por tu cuenta

Crear una cuenta personal

¿Ya dispones de una cuenta?

Iniciar sesión

Empezar a usar Tinkercad

¿Cómo crearás tu cuenta?

Iniciar sesión con el correo electrónico

Iniciar sesión con Google

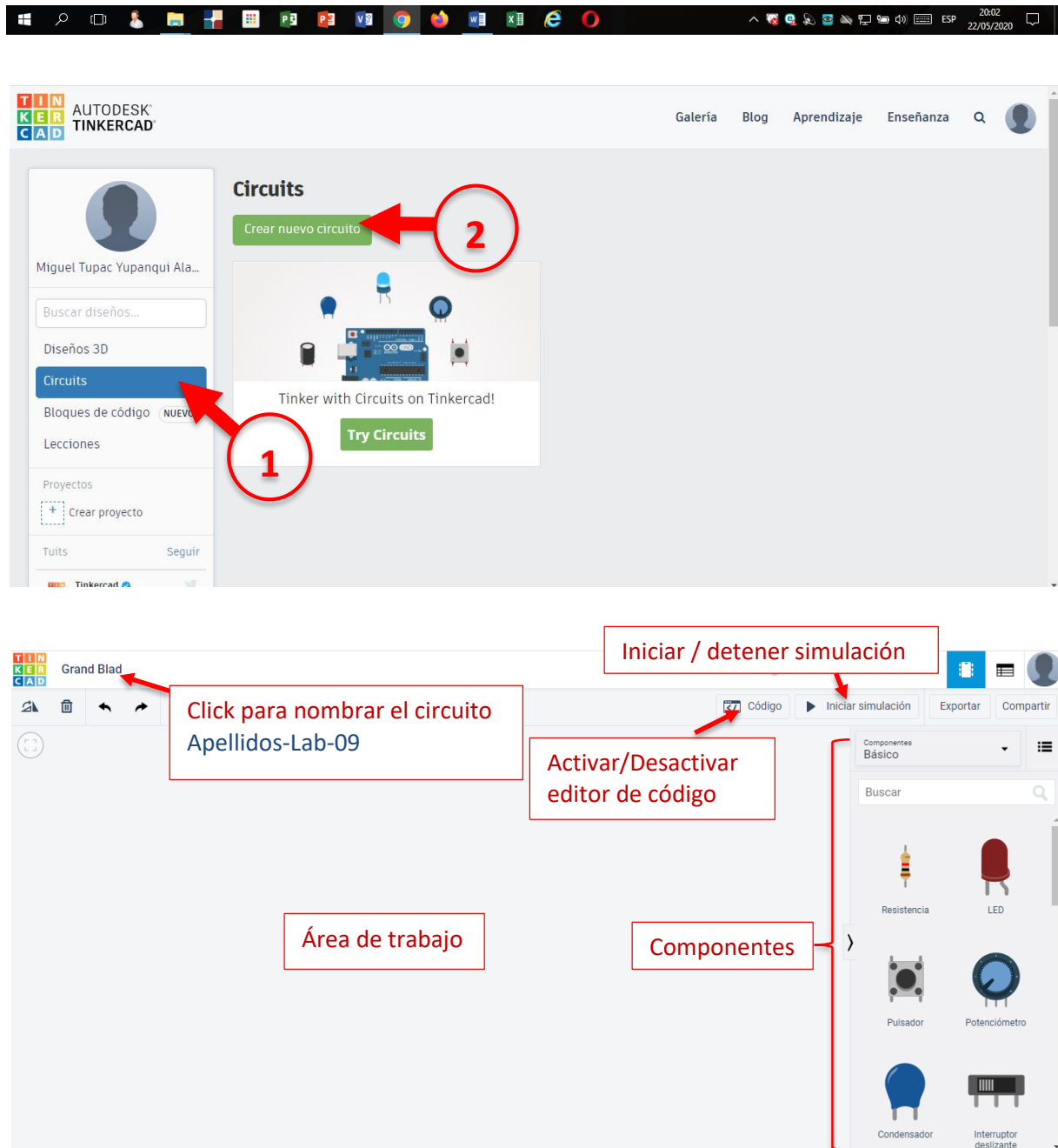
Iniciar sesión con Apple

Más opciones de inicio de sesión...

¿Ya dispones de una cuenta?

Iniciar sesión

Registrarse: emplea tu correo electrónico de la Universidad



“Monitor serial” del entorno de programación de Arduino.

“**Monitor serial**”: esta herramienta abrirá una consola con la que podemos comunicarnos con Arduino, tanto para mandarle comandos como para recibir información. Por ejemplo, es posible enviar comandos desde el ordenador para encender LEDs.

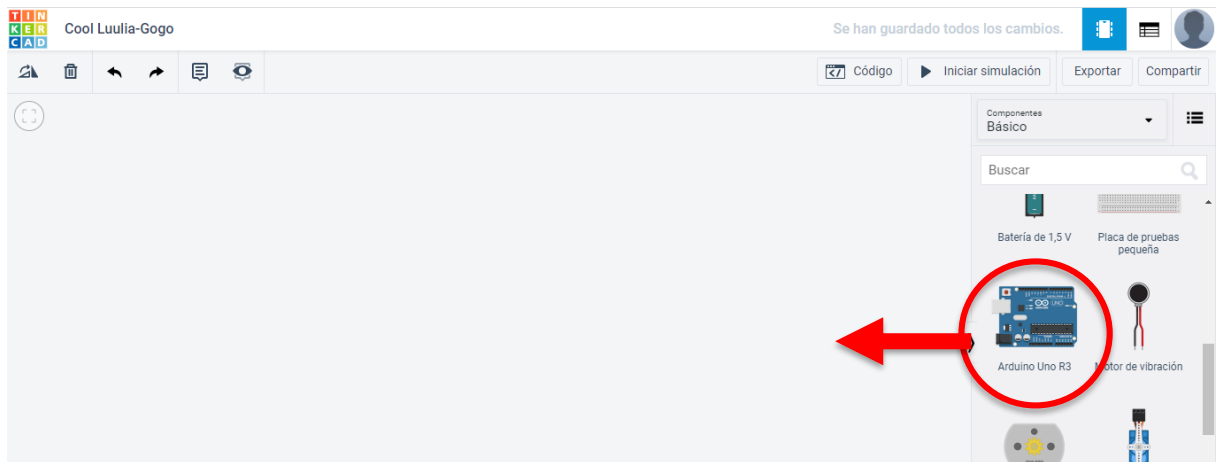
Para aperturarla, después de haber escrito el código en el editor correspondiente, hacer clic en el botón inferior derecho.



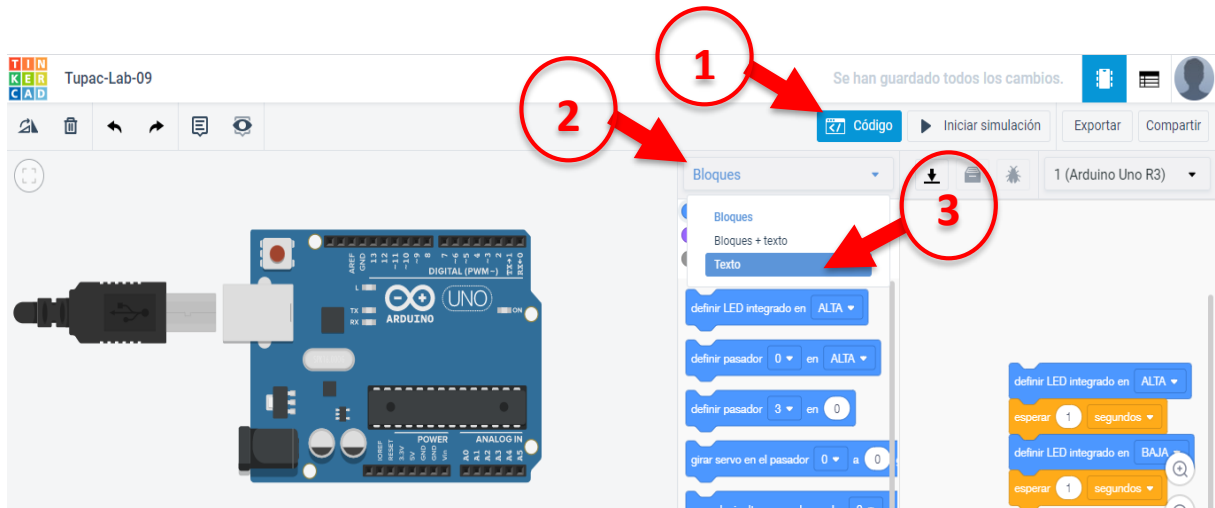
Monitor en serie

Primer programa para nuestro Arduino

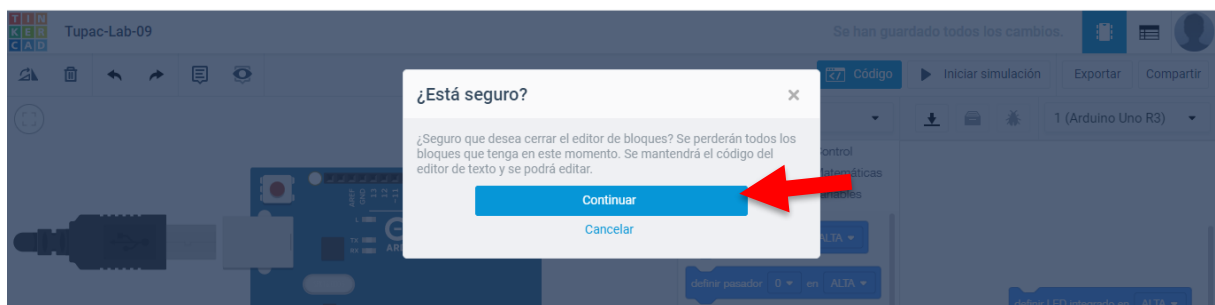
En el área de componentes buscamos la placa del Arduino Uno, y lo llevamos a la zona de trabajo.

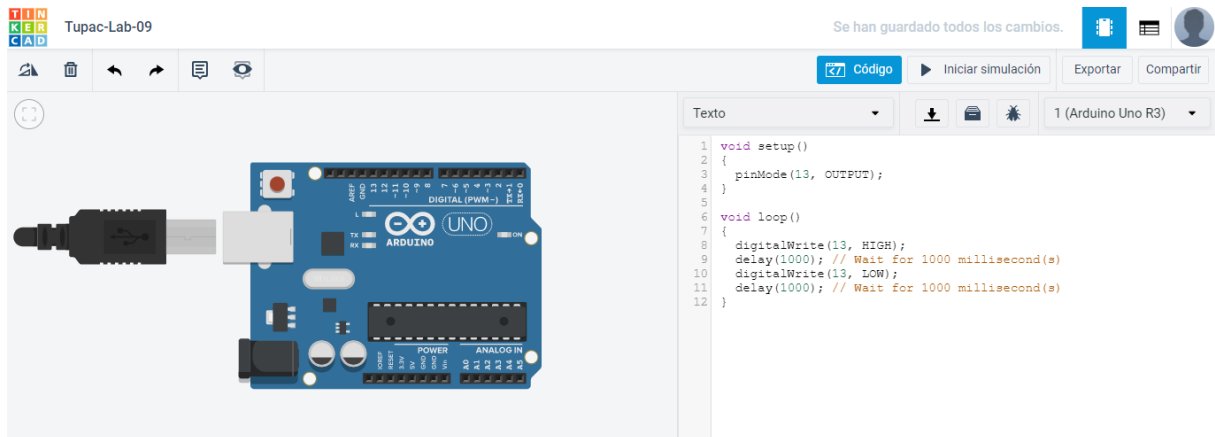


Hacemos clic en el editor de código (1), luego un clic en bloque (2), de las opciones que brinda seleccionamos Texto (3).

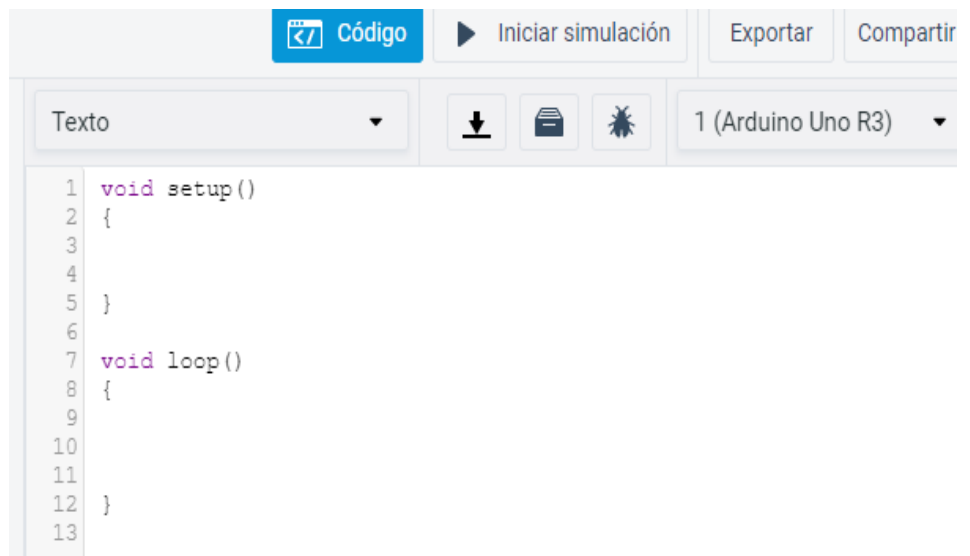


Seguidamente hacemos clic en **continuar**.

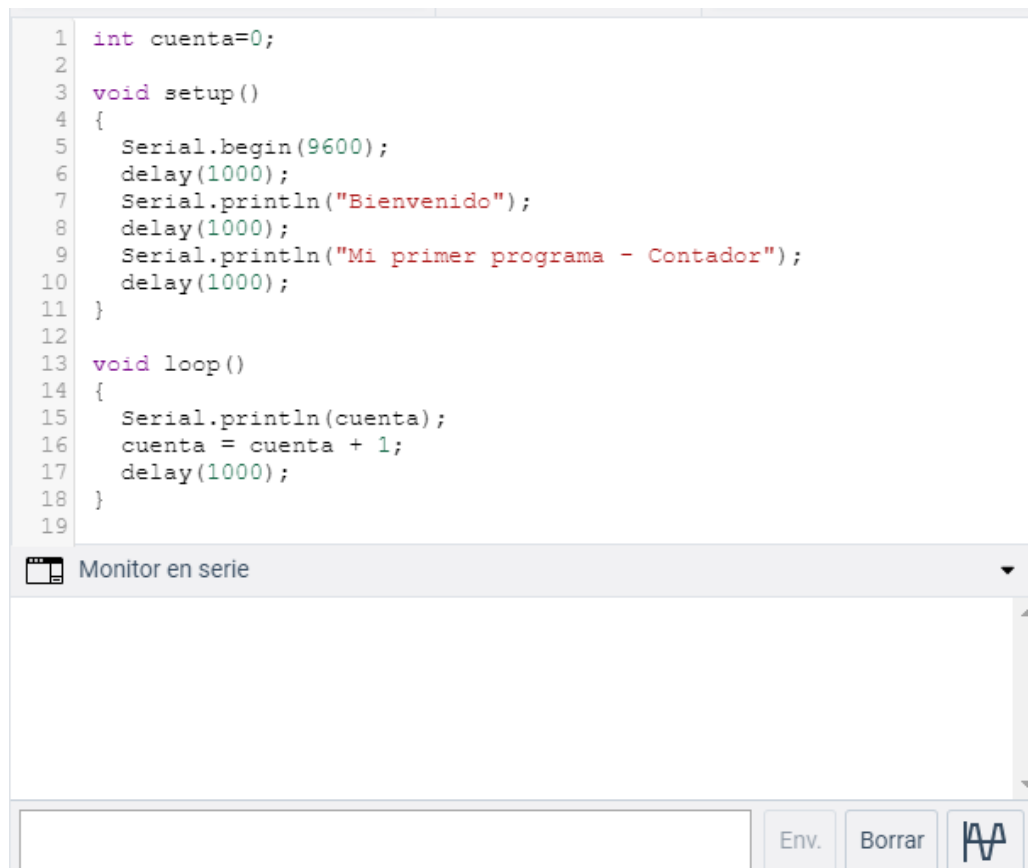




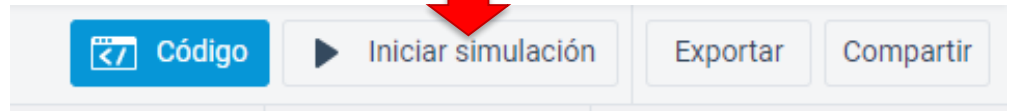
Para escribir nuestro programa, borramos el programa que viene previamente establecido.



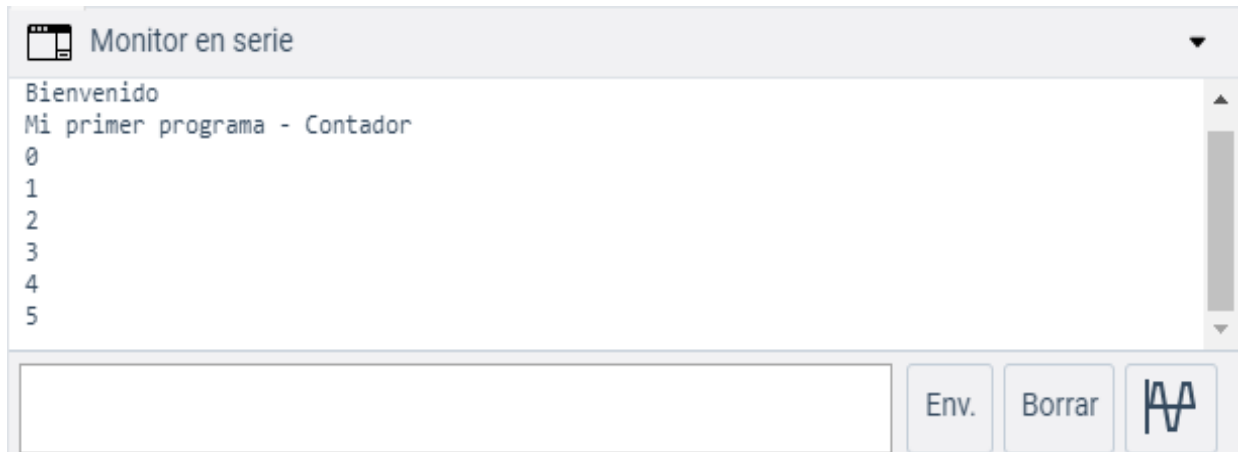
Escribimos el siguiente código y abrimos el Monitor serial.



Le damos clic en el botón: **iniciar simulación**



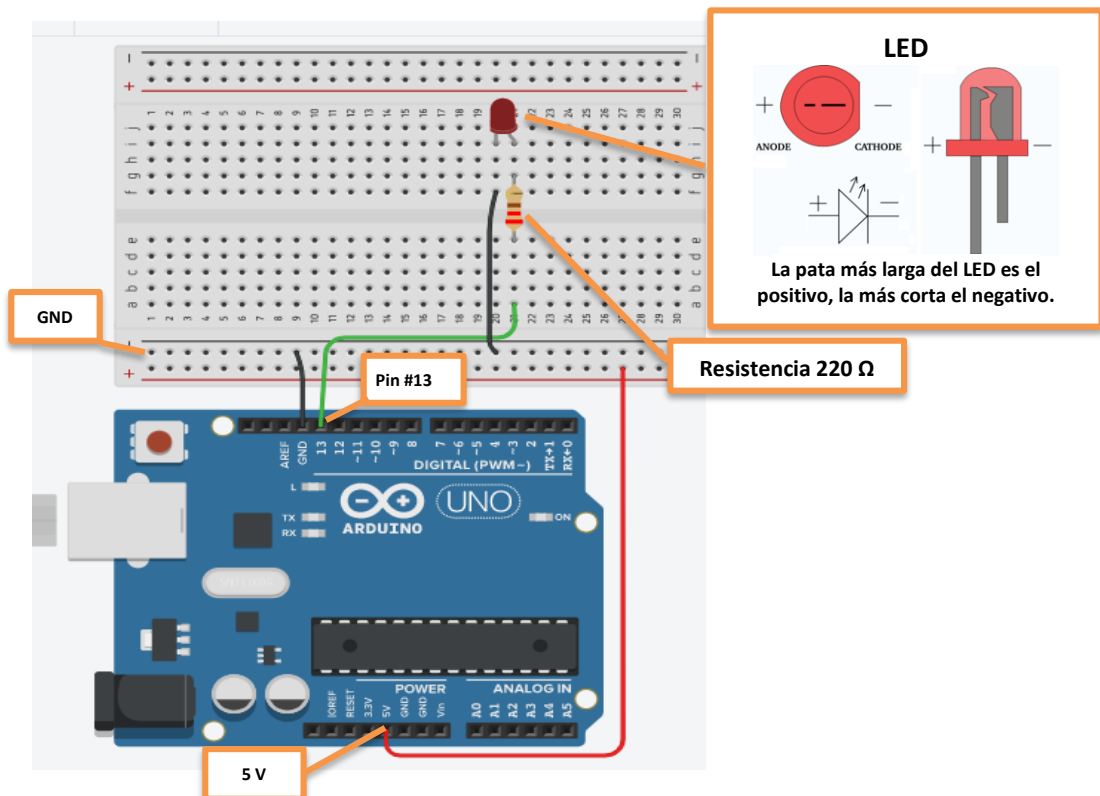
Y en el monitor serial podemos apreciar la ejecución del programa (estamos recibiendo la información), se puede observar que la función **void setup()** se ejecuta una sola vez, mientras que la función **void loop ()** se ejecuta repetidas veces.



Proyecto: encendido y apagado de un LED

Ahora implementaremos un **nuevo proyecto**, que permita **encender y apagar un LED**. Tener presente que en Arduino, las **señales digitales** tienen **2 valores: 0V que representa un nivel bajo (apagado) y 5V que representa un nivel alto (encendido)**.

Conectamos los componentes de forma similar al esquema siguiente:



Abrimos el **editor de código** para realizar la programación.

Recordar:

- El código que escribimos en **void setup()** se va a ejecutar una única vez al inicio del programa. Por tanto, en esta función vamos a **configurar** nuestro Arduino.
- El código que escribamos en **void loop()** se va a ir ejecutando línea a línea y cuando llegue al final va a volver a empezar, haciendo esto una y otra vez de forma indefinida. Por tanto, en esta función vamos a escribir **cómo queremos que** Arduino **actúe**.

Escribimos el siguiente programa:

```
1 void setup() {
2   pinMode(13,OUTPUT);    //Configuramos el pin 13 como SALIDA
3 }
4
5 void loop() {
6   digitalWrite(13,HIGH); //Establece en nivel ALTO en pin 13, ENCIENDE el Led
7   delay(1000);           // Espera 1000 milisegundos = 1 segundo
8   digitalWrite(13, LOW); //Establece en nivel BAJO en pin 13, APAGA el Led
9   delay(1000);           // Espera 1000 milisegundos = 1 segundo
10 }
11
```

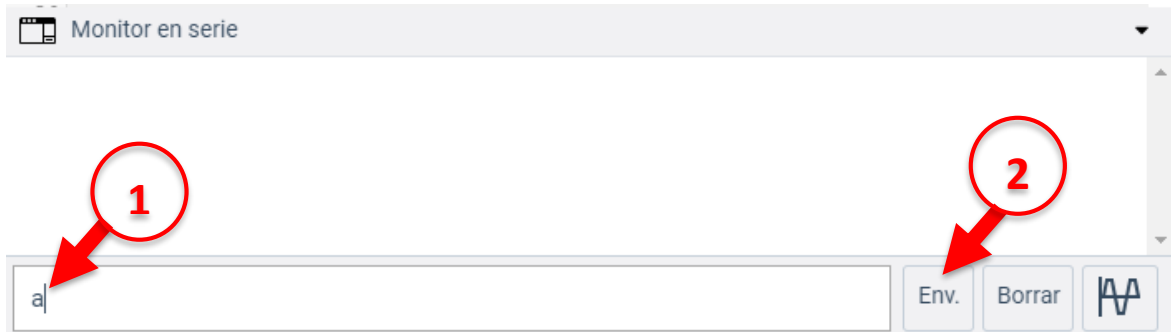
Le damos clic en el botón **iniciar simulación**, y podemos observar que **el LED** en la protoboard **estará parpadeando**.

Ingreso de datos por el Monitor Serial

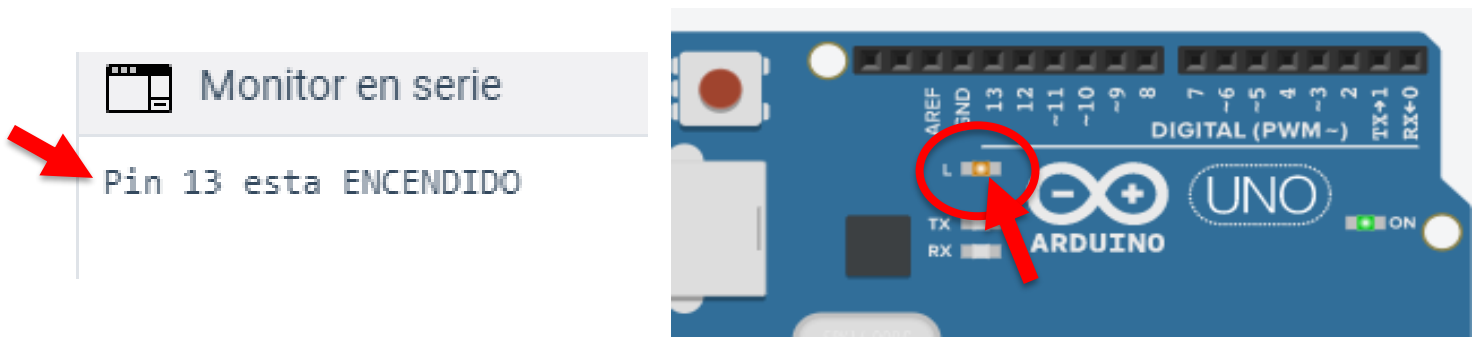
Para comprobar el ingreso de datos por el monitor serial, escribimos el siguiente programa:

```
1 char opcion=' ';
2
3 void setup() {
4   Serial.begin(9600); //se inicia la comunicación serial a 9600 Baudios
5   pinMode(13,OUTPUT); //se establece que el pin 13 es de salida
6 }
7
8 void loop() {
9   if (Serial.available()>0) { //si existe datos disponibles los leemos
10    opcion=Serial.read();      //leemos la opcion enviada y guardamos en opcion
11    if(opcion=='a') {
12      digitalWrite(13,HIGH);
13      Serial.println("Pin 13 esta ENCENDIDO");
14    }
15    if(opcion=='b') {
16      digitalWrite(13,LOW);
17      Serial.println("Pin 13 esta APAGADO");
18    }
19  }
20 }
```

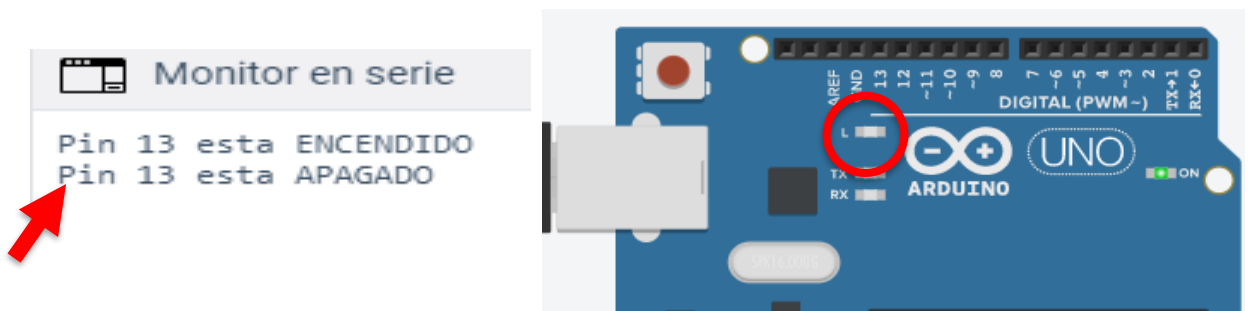
Aperturamos el monitor serial y hacemos clic en el botón **iniciar simulación**. En el campo de texto inferior escribimos la letra “a” (1) y hacemos clic en **Enviar** (2).



Podemos observar en el monitor serial el mensaje “Pin 13 está ENCENDIDO”, y verificamos esto en el Arduino que efectivamente está encendido.



Ahora en el campo de texto escribimos la letra “b” y hacemos clic en **Enviar**. Podemos observar en el monitor serial el mensaje “Pin 13 está APAGADO”, y verificamos en el Arduino que efectivamente está apagado.



5. Actividades

- Implemente un nuevo circuito usando 2 LEDs (Verde y Rojo – en ese orden de izquierda a derecha), elabora un programa para que parpadeen alternadamente 0,8 segundos (cuando esté encendido el LED Verde, el LED Rojo debe estar apagado y viceversa).
- Implemente un nuevo circuito usando 4 LEDs (Verde, Rojo, Azul y Amarillo – en ese orden de izquierda a derecha), elabora un programa para que parpadeen alternadamente 0,8 segundos, de la siguiente manera: cuando estén encendidos los LEDs Verde y Azul, deben estar apagados los LEDs Rojo y Amarillo, viceversa).