



## **Actividad |1| Semáforo LED.**

**Internet de las Cosas.**

---

**Ingeniería en Desarrollo de Software.**



TUTOR: Marco Alonso Rodríguez Tapia.

ALUMNO: Ramón Ernesto Valdez Felix.

FECHA: 18/02/2026.

<b>Introducción.....</b>	<b>3</b>
<b>Descripción.....</b>	<b>3</b>
<b>Justificación.....</b>	<b>4</b>
<b>Desarrollo.....</b>	<b>4</b>
<b>Armado del Circuito.....</b>	<b>5</b>
<b>Codificación.....</b>	<b>6</b>
<b>Emulación del Circuito.....</b>	<b>9</b>
<b>Conclusion.....</b>	<b>11</b>
<b>Referencias.....</b>	<b>11</b>

## Introducción.

En esta primera actividad de la materia de Internet de las Cosas. Realizaremos la práctica donde tenemos como objetivo consolidar los conocimientos básicos de electrónica y programación mediante la creación de un semáforo LED interactivo. Utilizando el simulador Tinkercad, pasarás de la teoría visualizada en tus recursos de estudio a la implementación técnica, conectando componentes esenciales como la placa Arduino Uno, resistencias y diodos LED.

El desarrollo de este proyecto no solo requiere un ensamblaje físico virtual preciso, sino también la aplicación de una lógica de programación estructurada y coherente. Durante el proceso, deberás gestionar el flujo de energía mediante la creación de variables de tipo entero, configurar correctamente los pines de salida en el bloque void setup y diseñar una secuencia cíclica lógica en el void loop. El objetivo final es replicar fielmente el comportamiento de un semáforo real (Rojo, Amarillo, Verde y retorno), permitiéndote comprender cómo el software controla el hardware para automatizar procesos cotidianos.

## Descripción.

En esta práctica inicial de la asignatura de Internet de las Cosas representa tu primer acercamiento al diseño de sistemas embebidos mediante la simulación de un semáforo inteligente, un paso fundamental para dominar la interacción entre el software y el hardware. El propósito principal es transformar los conceptos teóricos e introductorios en una solución técnica tangible, utilizando la plataforma Tinkercad para integrar hardware virtual como la placa Arduino Uno, resistencias de precisión y diodos emisores de luz.

Más allá del simple ensamblaje físico, este reto te exige aplicar una lógica de programación rigurosa. Deberás estructurar el código definiendo variables enteras, inicializando los pines de salida en el bloque void setup y programando una secuencia cíclica interactiva en el void loop. Al replicar el comportamiento real de los colores rojo, amarillo y verde, lograrás comprender profundamente cómo el

software interactúa con componentes electrónicos para automatizar señales de control, sentando las bases fundamentales para proyectos de IoT mucho más complejos en el futuro.

## Justificación.

La realización de esta práctica se justifica por la necesidad de cerrar la brecha entre la teoría de los sistemas embebidos y la implementación técnica en el Internet de las Cosas. Al desarrollar un semáforo LED en la plataforma Tinkercad, el estudiante no sólo replica un circuito básico, sino que adquiere habilidades críticas en el manejo de hardware virtual, como la placa Arduino y el uso de resistencias de protección. Este ejercicio es fundamental para comprender cómo pequeñas líneas de código pueden controlar elementos físicos de manera precisa y segura.

Asimismo, la importancia de este proyecto radica en el fortalecimiento del pensamiento lógico y algorítmico. El uso de variables de tipo entero y la configuración de funciones esenciales como void setup y void loop son pilares de la programación en C++. Al programar una secuencia cíclica que emula un sistema real, el alumno desarrolla la capacidad de diseñar soluciones automatizadas, sentando las bases para proyectos de mayor complejidad dentro del ecosistema IoT.

## Desarrollo.

El desarrollo de esta actividad se centra en la transición del aprendizaje teórico a la implementación práctica de un sistema de control básico dentro del ecosistema del Internet de las Cosas. Al simular el circuito, se obtiene un sistema funcional que replica el comportamiento de un semáforo real, demostrando la capacidad de automatizar procesos cotidianos mediante la programación de microcontroladores.

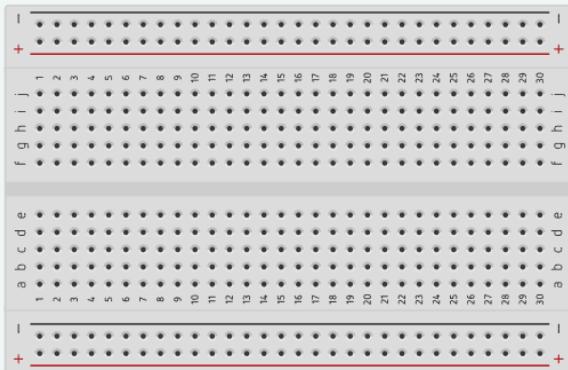
[Link: GitHub.](#)

[Link: Actividad.](#)

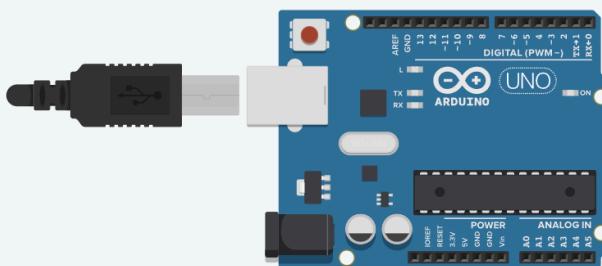
# Armado del Circuito.

Utilizamos los componentes básicos de la herramienta de trabajo tinkerCAD para realizar la actividad de la creación de un semáforo LED.

## Circuitos basicos a utilizar: Semaforo LED.



**Placa de pruebas (Breadboard):** Es la base blanca con agujeros que permite conectar los componentes entre sí sin necesidad de soldar. Las líneas horizontales y verticales están conectadas internamente para compartir electricidad.



**Placa Arduino Uno:** Es el "cerebro" programable. Recibe el código que escribes y envía señales eléctricas a través de sus pines para controlar los LEDs.

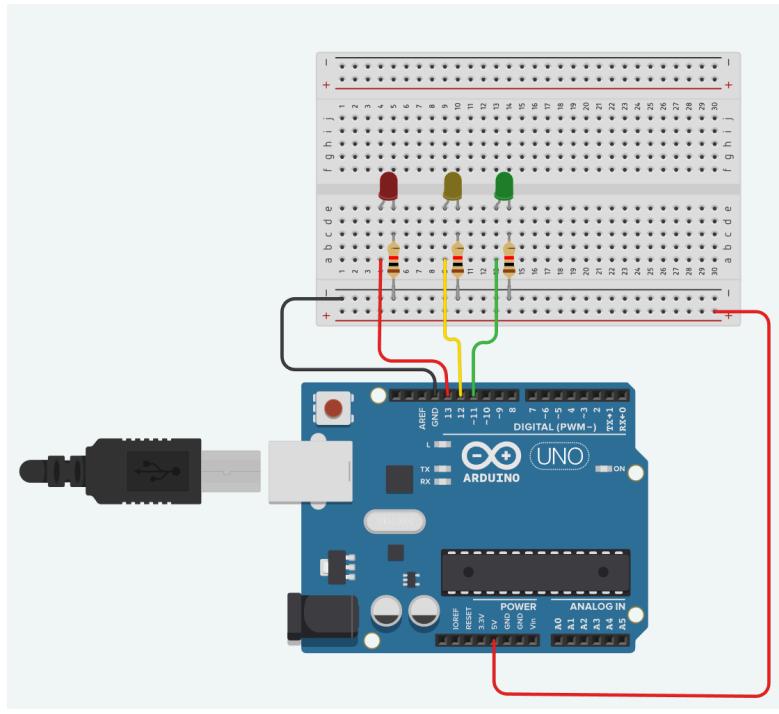


**Diodos LED (Rojo, Amarillo, Verde):** Son los indicadores luminosos. Solo permiten el paso de la corriente en un sentido y se encienden cuando reciben energía desde el Arduino.



**Resistencias (220 ohms):** Son esos pequeños cilindros con bandas de colores. Su función es limitar la cantidad de corriente que llega a los LEDs para evitar que se quemen.

**Cables de conexión (Jumpers):** Los hilos de colores que transportan la energía y los datos entre la placa Arduino y la placa de pruebas. Con todos estos circuitos básicos realizamos el armado de actividad (Semáforo LED).



## Codificación.

En este punto de la actividad realizaremos la documentación de la codificación del semáforo LED que nos pide realizar la materia de internet de las cosas como primer actividad y daremos un explicación breve de lo que realiza el código.

### Codificación de Semaforo LED.

```
int ledRojo = 13;
```

```
int ledAmarillo = 12;
```

```
int ledVerde = 11;
```

#### 1. Preparación del Terreno (Variables y Setup)

Antes de que el semáforo empiece a funcionar, el código realiza dos tareas

```
void setup() {
    pinMode(ledRojo, OUTPUT);
    pinMode(ledAmarillo, OUTPUT);
    pinMode(ledVerde, OUTPUT);
```

fundamentales:

- **Asignación de nombres:** Al usar int ledRojo = 13, le estás diciendo al Arduino que cada vez que lea la palabra "ledRojo", debe dirigir su atención al pin físico número 13. Esto hace que el código sea humano y fácil de leer.
- **Definición de roles:** En el void setup(), la instrucción pinMode(..., OUTPUT) prepara los pines para enviar energía hacia afuera, en lugar de recibirla de sensores.

```
void loop()
{
    digitalWrite(ledRojo, HIGH);
    delay(3000);
    digitalWrite(ledRojo, LOW);
    digitalWrite(ledAmarillo, HIGH);
```

## 2. El Ciclo de Vida (Void Loop)

Esta función es un bucle infinito. Una vez que llega al final, vuelve a empezar inmediatamente desde la primera línea.

## 3. Ejecución Paso a Paso de la Secuencia

Cuando inicias la simulación, esto es lo que ocurre eléctricamente:

1. **Fase Roja:** digitalWrite(ledRojo, HIGH)

```
delay(1000);

digitalWrite(ledAmarillo, LOW);

digitalWrite(ledVerde, HIGH);

delay(3000);

digitalWrite(ledVerde, LOW);

digitalWrite(ledAmarillo, HIGH);

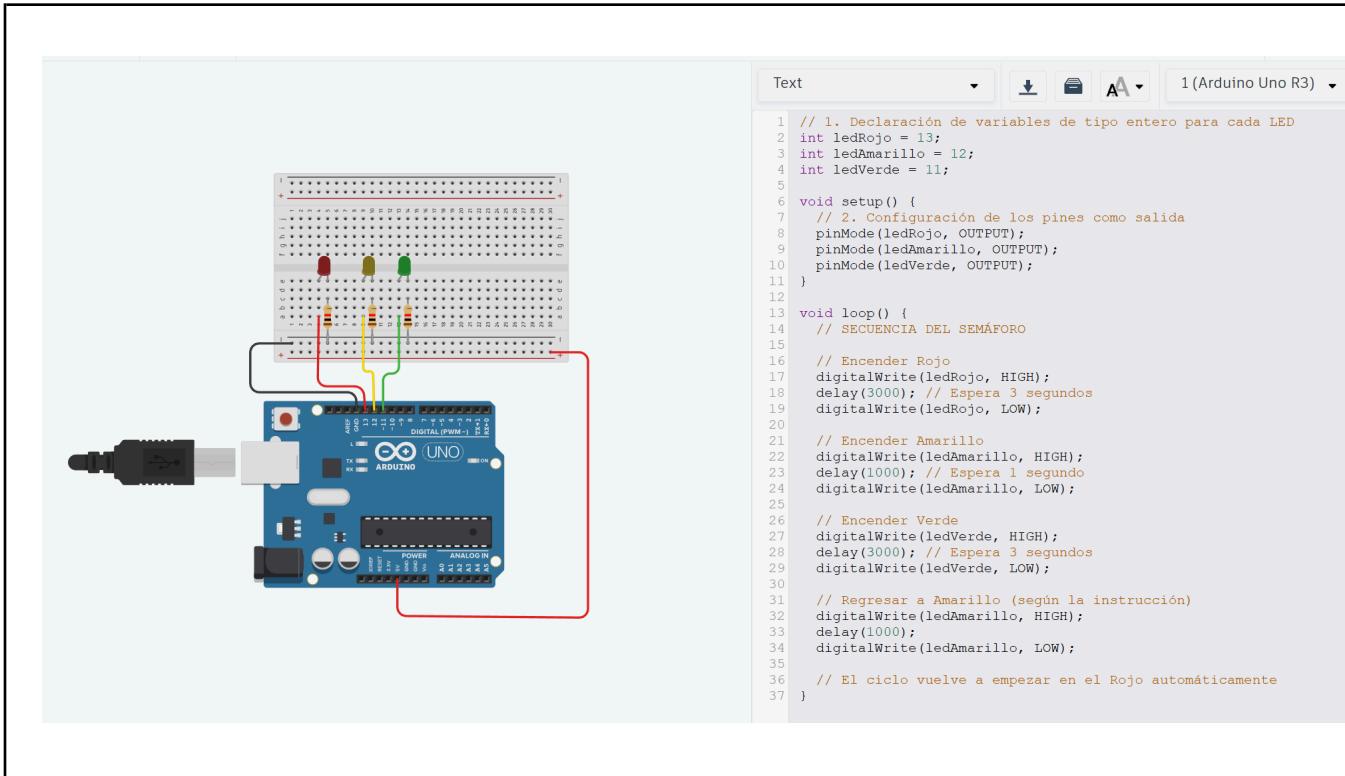
delay(1000);

digitalWrite(ledAmarillo, LOW);

}
```

envía 5 voltios al pin 13. La corriente fluye a través de la resistencia, enciende el LED rojo y el código se "detiene" por 3000 milisegundos (3 segundos) gracias al delay. Luego, el LOW corta la energía y el LED se apaga.

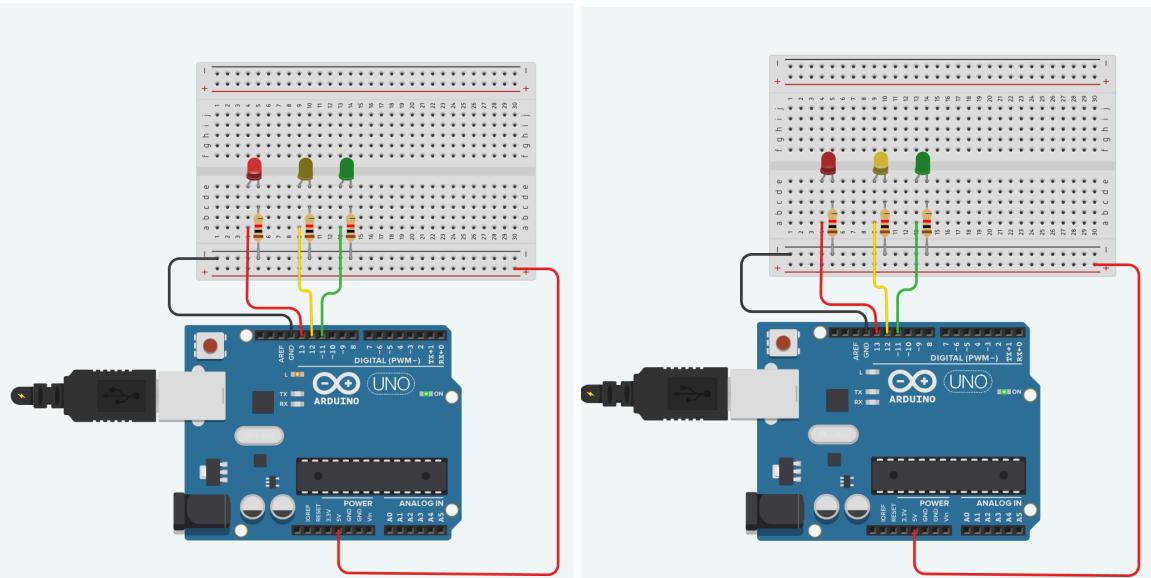
2. **Transición Amarilla:** Inmediatamente después, el pin 12 recibe energía por 1 segundo. Esta es una señal de advertencia corta, tal como en un semáforo real.
3. **Fase Verde:** El pin 11 se activa por otros 3 segundos, permitiendo el "paso" virtual.
4. **Retorno Seguro:** Cumpliendo con tu requerimiento específico, el código no salta directamente al rojo, sino que vuelve a encender el amarillo por 1 segundo. Esto es clave para una simulación realista de tráfico donde el cambio de verde a rojo requiere una transición.

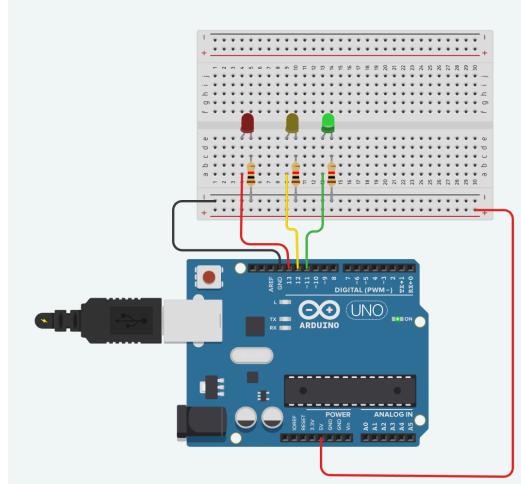


## Emulación del Circuito.

Al emular este código en una plataforma como Tinkercad, estás ejecutando un programa que convierte instrucciones lógicas en señales eléctricas reales. Aquí se tiene la emulación de los circuitos en su ejecución encendiendo y apagando cada uno de los LED aplicado en la lógica de la codificación del semáforo LED con los colores rojo, amarillo y verde.

### Prueba de emulación de actividad del semáforo LED. (Rojo, Amarillo y Verde)





Private

### Stunning Elzing-Esboo

TINKER CAD Stunning Elzing-Esboo

Circuit by Ramón Valdez

Design is visible only to you. [Change visibility to share](#)

Edited February 18, 2026  
Created February 18, 2026

<https://www.tinkercad.com/things/dS5XPnDLSmY-stunning-elzing-esboo>

## Conclusion.

En conclusión: La importancia de esta actividad dentro del campo del Internet de las Cosas (IoT) radica en la comprensión de cómo la programación de sistemas embebidos transforma nuestro entorno.

En el ámbito laboral, el dominio de microcontroladores como Arduino permite a ingenieros y técnicos diseñar soluciones de automatización industrial, desde semáforos inteligentes que optimizan el tráfico en tiempo real hasta sistemas de monitoreo de seguridad. Esta práctica enseña a estructurar procesos lógicos que, mediante actuadores y sensores, pueden gestionar recursos de manera eficiente, reduciendo costos y errores humanos.

En la vida cotidiana, el IoT se manifiesta en el hogar inteligente, donde la lógica aplicada en este semáforo se replica en luces automáticas o electrodomésticos programables. Entender el flujo de datos entre el software y el hardware empodera al usuario para no solo consumir tecnología, sino también para crearla y personalizarla. En conclusión, lo realizado en esta actividad es la base fundamental para desarrollar ciudades más conectadas y hogares más funcionales, mejorando significativamente la calidad de vida y la competitividad profesional.

## Referencias.

*Google. (n.d.). Gemini. Retrieved February 19, 2026, from*

*<https://gemini.google.com/>*

*Tinkercad. (n.d.). Tinkercad. Retrieved February 19, 2026, from*

*<https://www.tinkercad.com/dashboard/designs/all>*