

## Práctica de laboratorio: activación de un LED con Blockly (Versión para el instructor)

**Nota para el instructor:** El color de fuente rojo o las partes resaltadas en gris indican texto que solamente aparece en la copia del instructor.

### Objetivos

**Parte 1:** abra Packet Tracer y examine el programa Blockly para hacer parpadear el LED

**Parte 2:** controle un LED RGB mediante Blockly

### Aspectos básicos

Blockly es un lenguaje de programación visual que permite a los usuarios crear programas conectando bloques que representan diversas estructuras de lenguaje lógico en lugar de escribir el código real. Blockly se ejecuta dentro de un navegador web y puede traducir el programa visualmente creado, como JavaScript, PHP o Python. En esta práctica de laboratorio, utilizará Blockly para examinar la programación de Blockly y controlar un LED.

### Situación

Uso de la programación de Blockly para controlar un objeto LED de IoT. En esta práctica de laboratorio, se usa Cisco Packet Tracer, ya que proporciona soporte para Blockly con objetos de IoT.

### Recursos necesarios

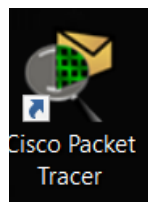
- Cisco Packet Tracer 7.1.1 y versiones superiores están instaladas y disponibles.

## Parte 1: Inicie Cisco Packet Tracer (PT) y utilice Blockly

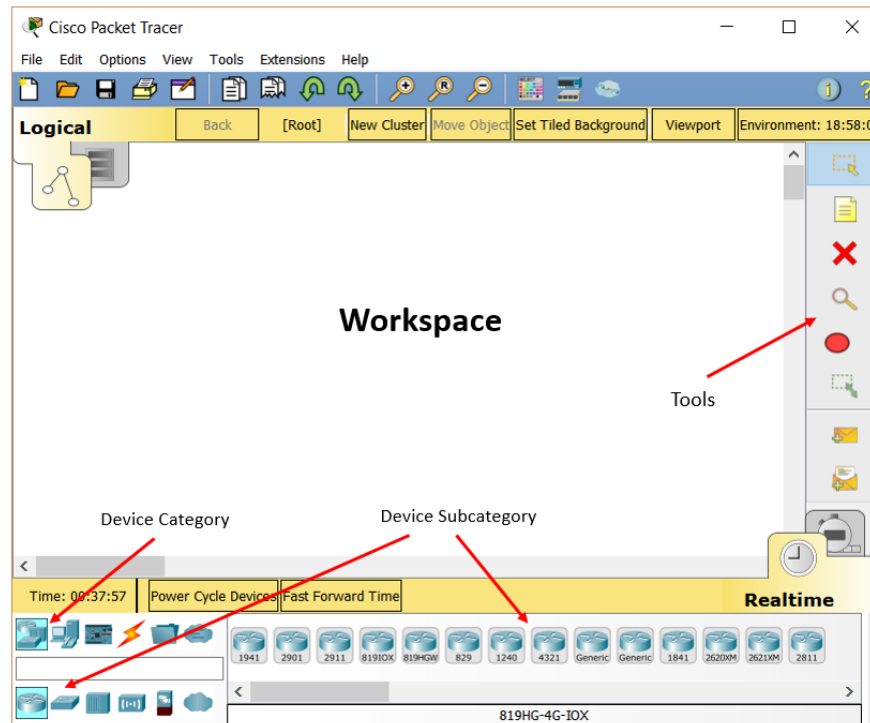
En la parte 1, accederá al programa Cisco Packet Tracer y examinará el control de LED mediante la programación de Blockly.

### Paso 1: Inicie Packet Tracer.

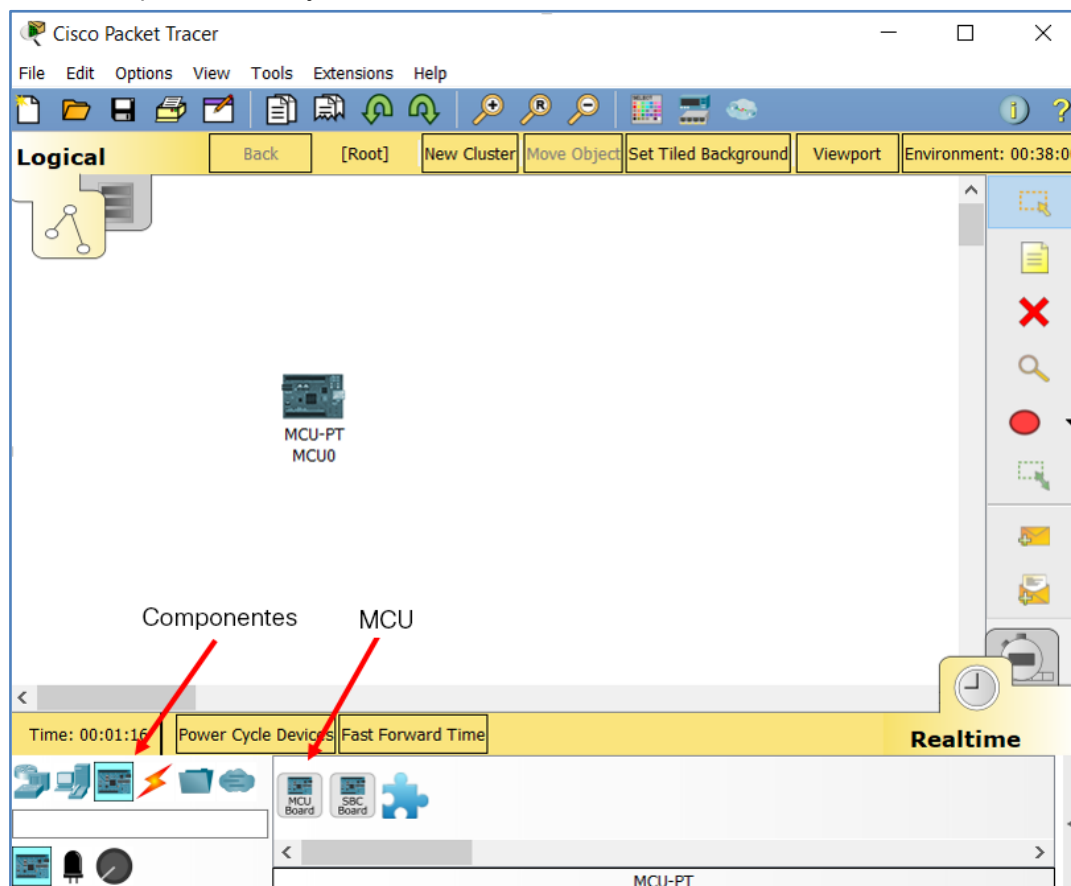
- a. Haga doble clic en el icono de Cisco Packet Tracer para abrir el programa PT.



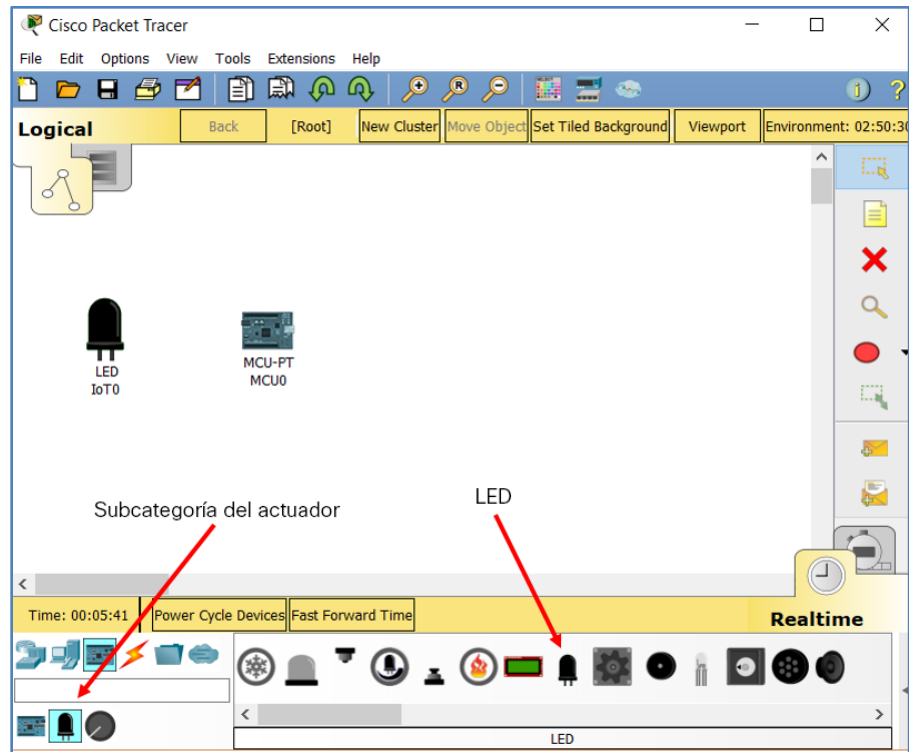
- b. Se muestra la interfaz de usuario.



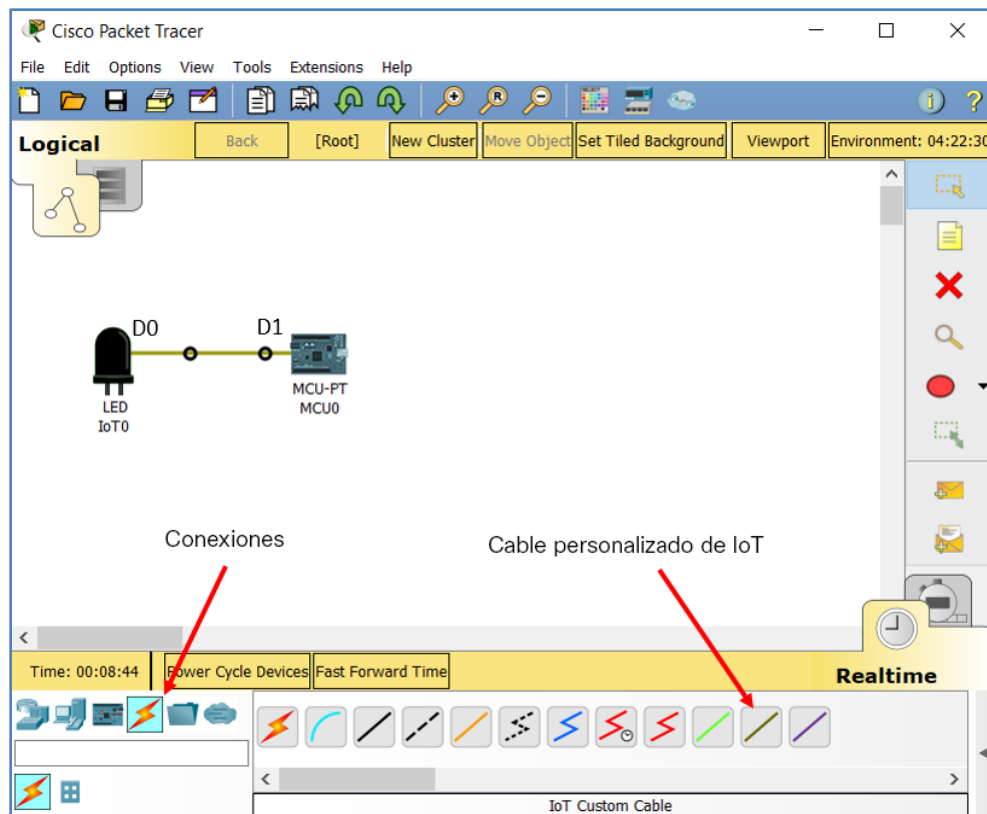
- c. Haga clic en la categoría **Components** (Componentes), y luego haga clic en **MCU Board** (Placa de MCU) y arrástrela al espacio de trabajo.



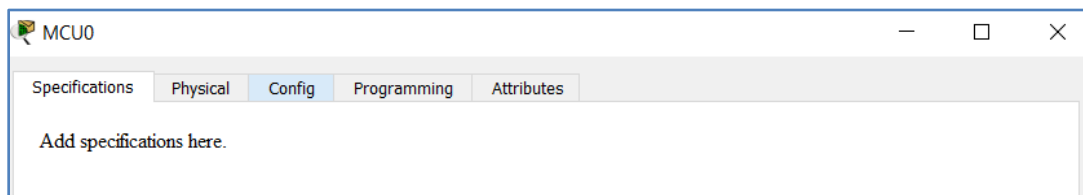
- d. Haga clic en la subcategoría **Actuators** (Actuadores), seleccione **LED** y arrástrelo al espacio de trabajo.



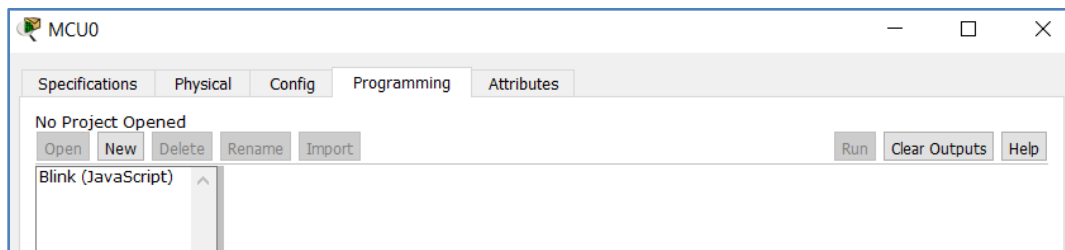
- e. Haga clic en la categoría **Connections** (Conexiones), seleccione **IoT Custom Cable** (Cable personalizado de IoT) para vincular la MCU en el puerto **D1** y el LED en el puerto **D0**.



- f. Haga doble clic en la **MCU**. Se muestra la ventana de configuración.

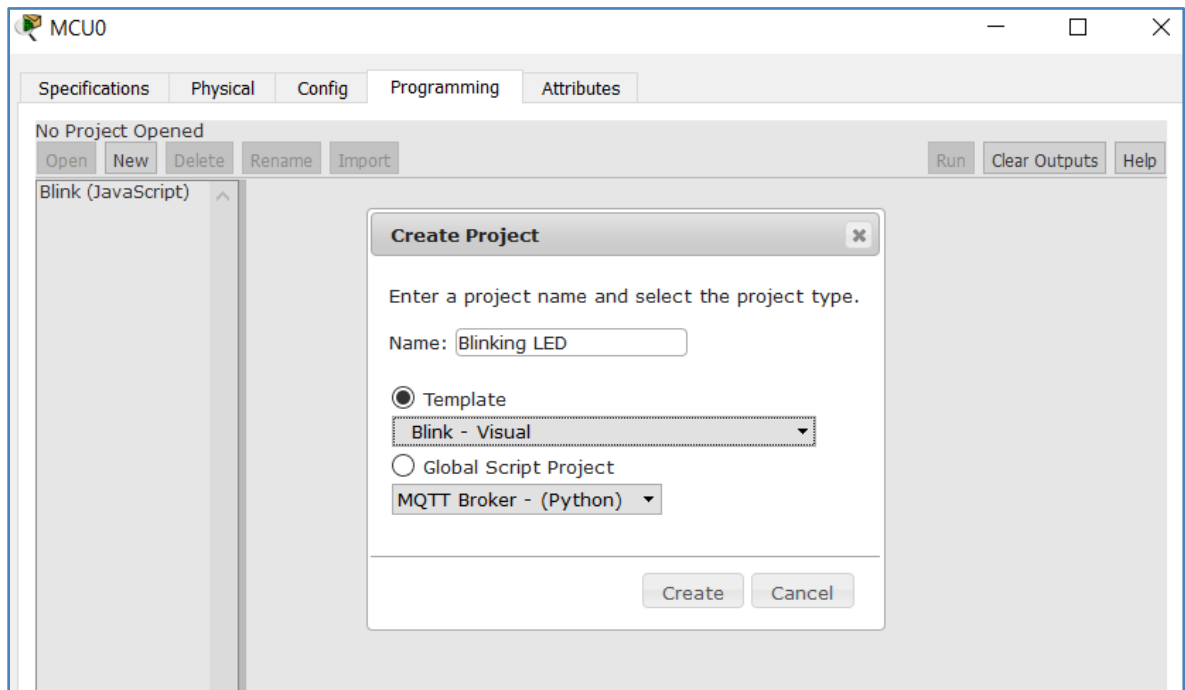


- g. Haga clic en la pestaña **Programming** (Programación). (Si no ve la pestaña Programming [Programación], haga clic en el botón **Advanced** [Avanzado], que se encuentra en la esquina inferior derecha).

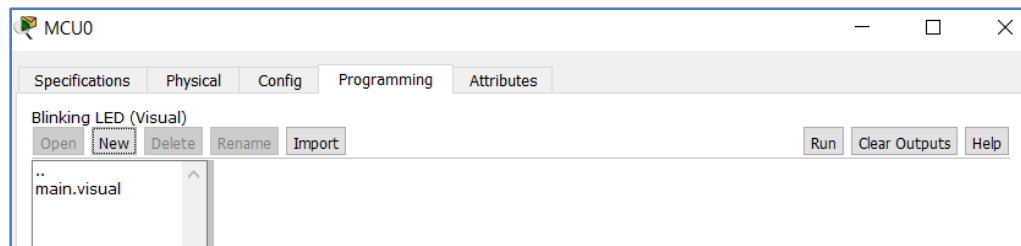


### Paso 2: Examine un programa Blockly creado previamente

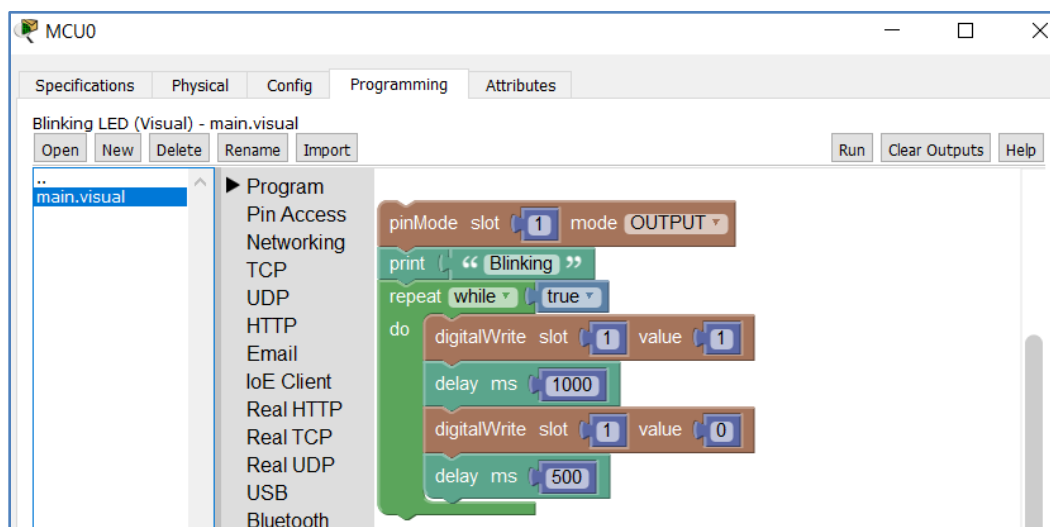
- Debajo de la nota **No Project Opened** (No hay proyectos abiertos), haga clic en **New** (Nuevo). En la ventana **Create Project** (Crear proyecto), ingrese **Blinking LED** (LED parpadeante) como nombre del proyecto. En el menú desplegable **Template** (Plantilla), seleccione **Blink - Visual** (Parpadeo: visual).



- Haga clic en Create (Crear).



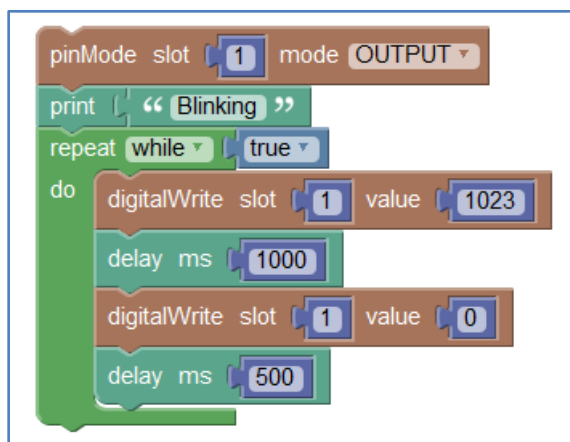
- c. Haga doble clic en **main.visual**. Se muestra el programa Blockly creado previamente.



- d. Haga clic en Run (Ejecutar). ¿El LED parpadea?

No

- e. Haga clic en **Stop** (Detener), y cambie el campo **Value** (Valor) del primer bloque **digitalWrite** a 1023.



- f. Haga clic en Run (Ejecutar). ¿El LED parpadea?

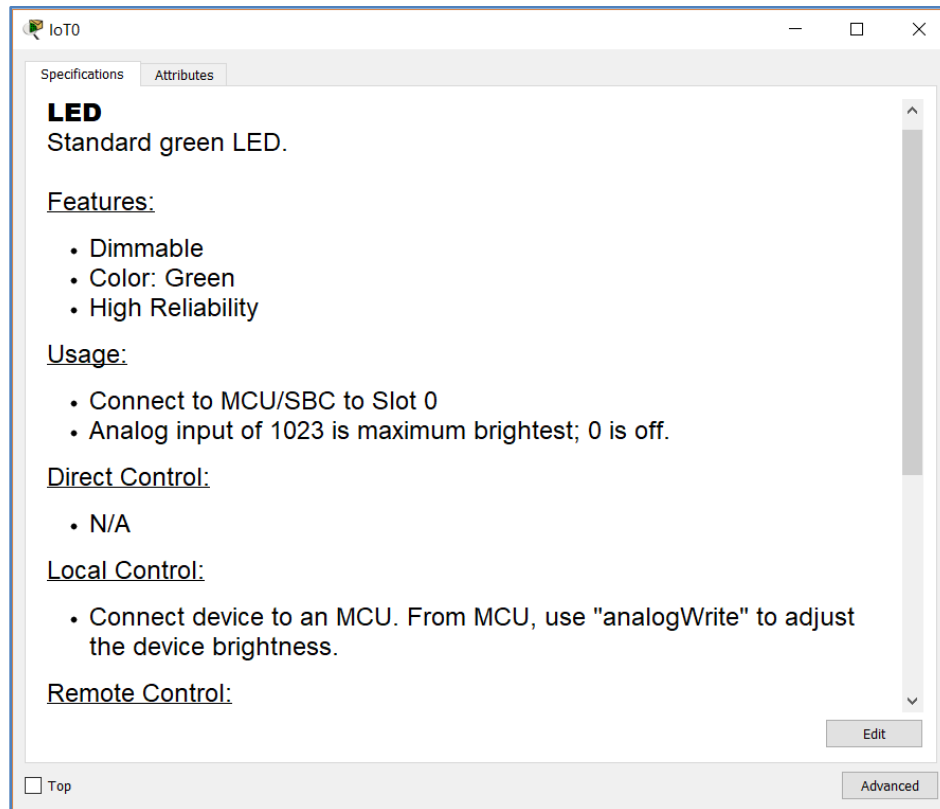
Sí

¿Por qué no parpadeó cuando el valor se estableció en 1023?

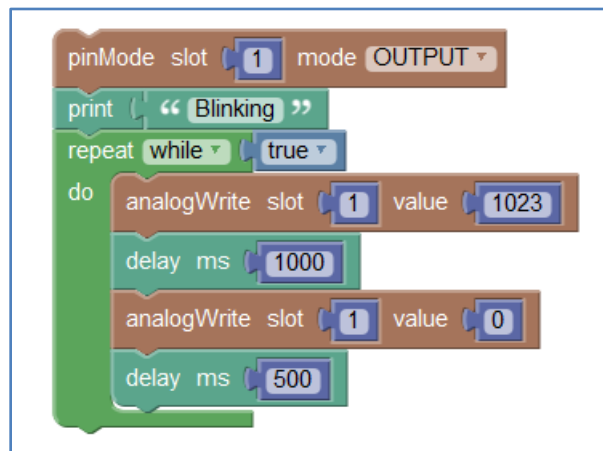
---

La salida digital toma el valor 1023 como "ALTO" y activa la corriente de salida.

- g. Haga clic en el LED y analice su especificación.



- h. Indica que podemos utilizar "analogWrite" para ajustar el brillo del dispositivo. Expanda el grupo **Pin Access** (Pin Access (Acceso a Pin)) y utilice el bloque **analogWrite** para reemplazar el bloque **digitalWrite**.



- i. Ahora, cambie los valores del primero y segundo bloque **analogWrite**, y observe los diferentes niveles de brillo de LED.

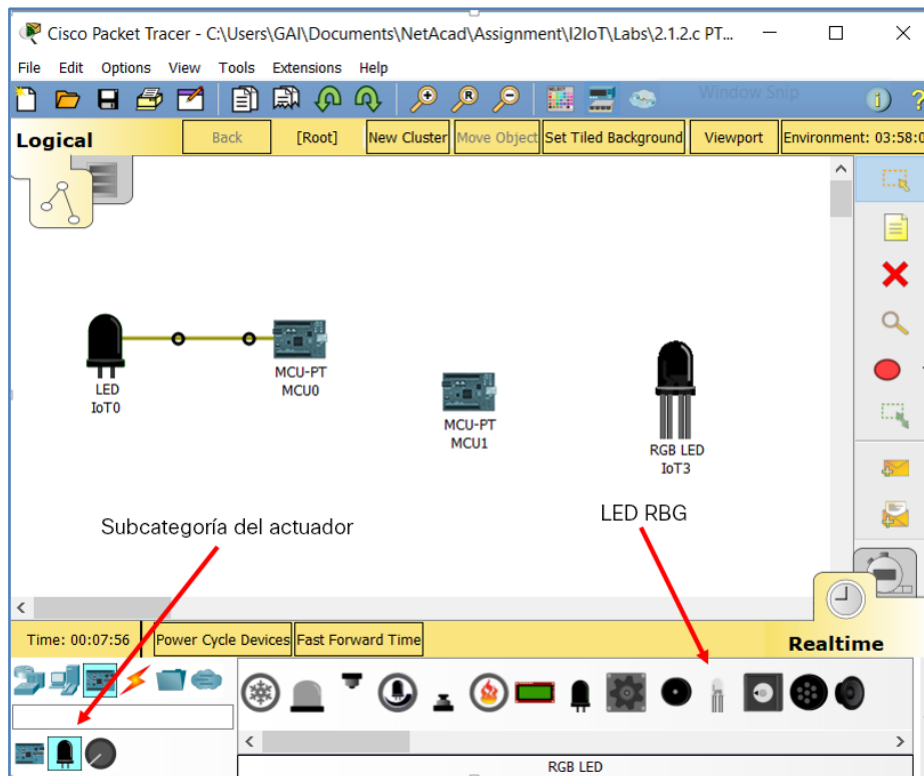
## Parte 2: Controle un LED RGB con Blockly

En la parte 2, utilizó Blockly para controlar un LED rojo, verde, azul (RGB, red, green, blue). Un indicador RGB puede mostrar diferentes colores con la combinación de color rojo, verde y azul.

### Paso 1: Agregue una MCU y un LED RGB.

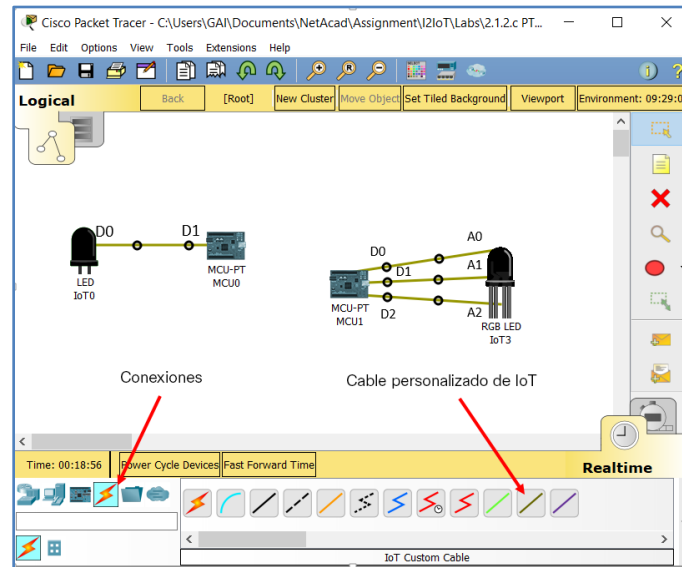
En el paso 1, agregó otra placa de MCU y un LED RGB en el espacio de trabajo.

- Haga clic la subcategoría **Actuators** (Actuadores), seleccione **RGB LED** (LED RGB) y arrástrelo al espacio de trabajo. Agregue otra placa de MCU.

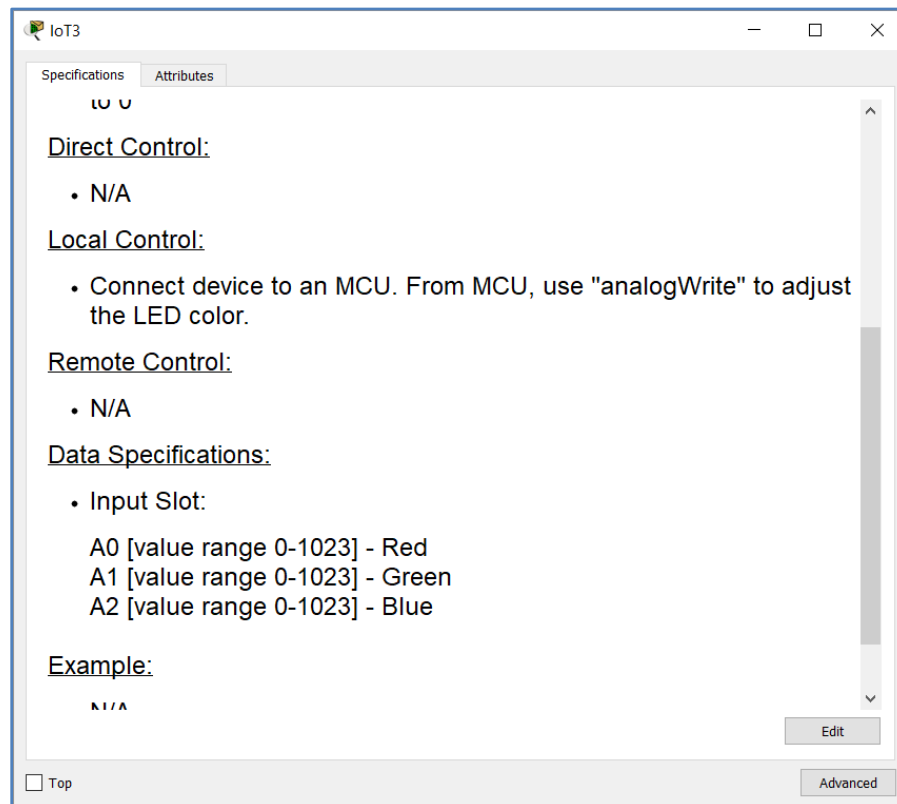




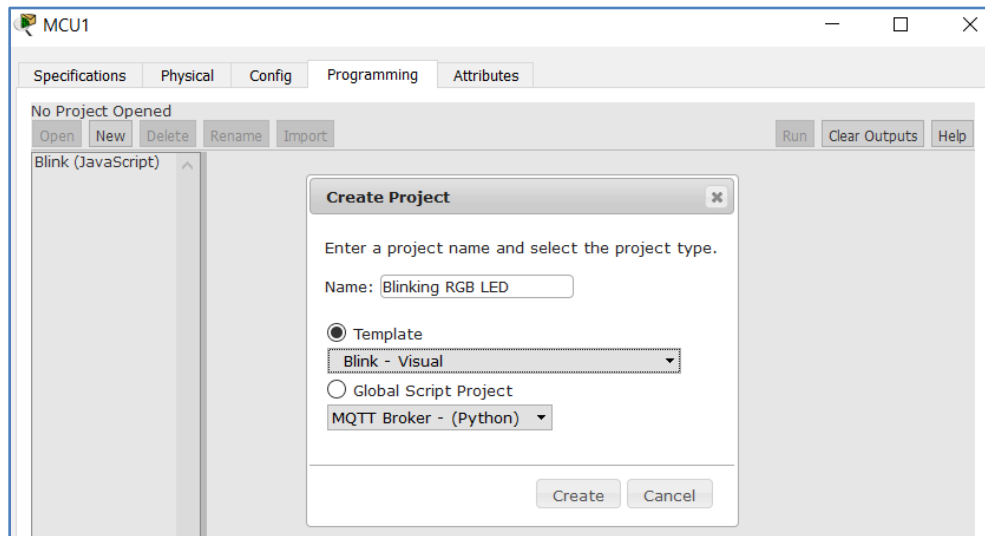
- b. Haga clic en la categoría **Connections** (Conexiones) y seleccione tres elementos en **IoT Custom Cable** (Cable personalizado de IoT) para vincular la MCU y el LED RGB.



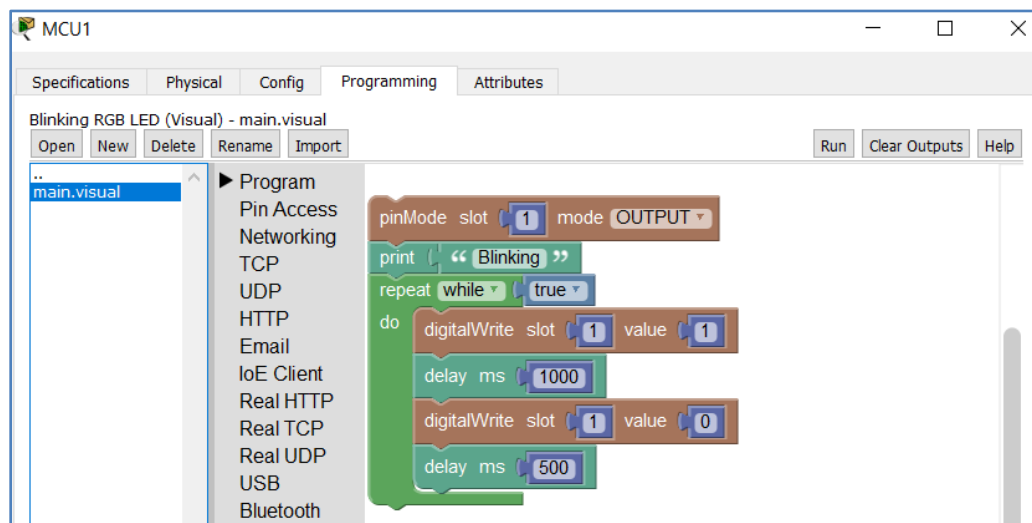
- c. Haga clic en **RGB LED** (LED RGB) y revise su especificación. Tenga en cuenta que diferentes entradas de pin representan distintos colores.



- d. Abra el programa Blockly creado previamente. Haga clic en MCU > Programming (MCU > Programación). Debajo de la nota **No Project Opened** (No hay proyectos abiertos), haga clic en **New** (Nuevo). En la ventana **Create Project** (Crear proyecto), ingrese **Blinking RGB LED** (LED RGB parpadeante) como nombre del proyecto. En el menú desplegable Template (Plantilla), seleccione **Blink - Visual** (Parpadeo: visual).

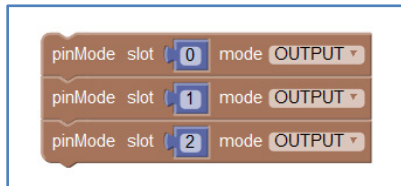


- e. Haga clic en Create (Crear). Haga doble clic en main.visual. Se muestra el programa Blockly creado previamente.

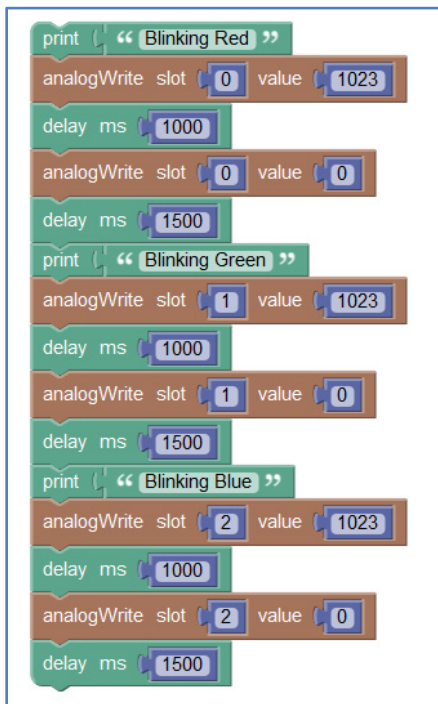


### Paso 2: Modifique el programa Blockly

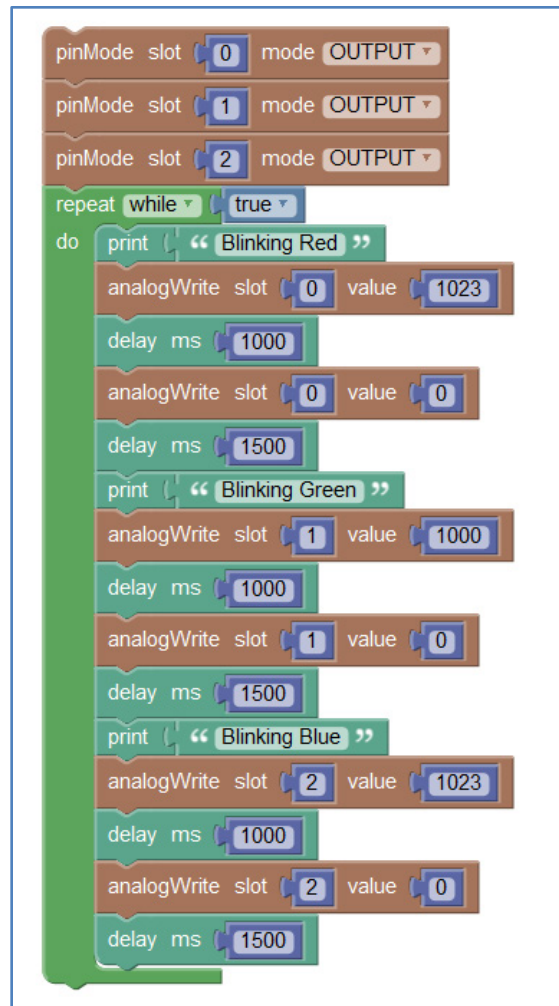
- Expanda el grupo **Pin Access** (Pin Access (Acceso a Pin)) y agregue dos bloques más **pinMode** para establecer tres casillas como **OUTPUT** (SALIDA) (en MCB para enviar una señal al LED RGB).



- En **Pin Access group** (Grupo de Pin Access (Acceso a Pin)), seleccione bloques **analogWrite** para reemplazar bloques **digitalWrite**. Además, agregue algunos bloques **print** (imprimir).



c. El programa final es el siguiente:



d. Ejecute el programa. El LED debe aparecer de color ROJO, VERDE y AZUL en secuencia.

### Desafío

Modifique el programa para mostrar una combinación de colores de las tres entradas con valores diferentes, generados de manera aleatoria, para cada casilla.

**A continuación, se proporciona un ejemplo:**

