PRACTICA USO BÁSICO DEL ESP3-S3

**Integrantes:**

Bedoya Montealegre Brayan Smith

Molina Fierro Jhon Sebastián

CORPORACIÓN UNIVERSITARIA DEL HUILA – CORHUILA

04/09/2025 – 06/09/2025

# **RESUMEN**

En esta práctica de laboratorio se exploraron las capacidades del microcontrolador ESP3-S3, donde mediante el uso del entorno de desarrollo Thonny y el lenguaje MicroPython, se buscó comprender el funcionamiento de los pines GPIO, su configuración como entradas y salidas, y su aplicación en el control de dispositivos externo. Este tipo de práctica, se desarrolló una mayor familiaridad con la computación del ESP3-S3 y los pines GPIO, tales como el hardware que es principalmente lo que puede evidenciar en él, y como punto de enfoque poder manipular los elementos correspondientes a la actividad para una mejor compresión sobre cómo funciona la electrónica y la programación juntas. Se uso una metodología independiente, en el que cada grupo podía ir al ritmo que consideraba pertinente, junto con la supervisión y apoyo del profesor, dándonos consejos, cuidado y guía con lo que tuviéramos duda o problemas. El punto para resaltar fue el gran conjunto de ESP3-S3 cumpliendo con su propósito gracias a las instrucciones dadas en la guía de laboratorio y el profesor, leds iluminándose, botones presionados, contadores en los display, retos en la generación de códigos donde la creatividad y el conocimiento sobresalían, al final, en esta practica de laboratorio el ESP3-S3 fue una herramienta para el aprendizaje de electrónica y programación básica a la mano permitiendo comprender el uso de pines GPIO y la lógica de entradas y salidas digitales.

*In this laboratory practice, the capabilities of the ESP32-S3 microcontroller were explored. Using the Thonny development environment and the MicroPython language, the aim was to understand the functioning of the GPIO pins, their configuration as inputs and outputs, and their application in controlling external devices. Through this type of practice, greater familiarity with the ESP32-S3 and its GPIO pins was developed, highlighting the hardware as the main component to be studied, and focusing on manipulating the elements involved in the activity to achieve a better understanding of how electronics and programming work together.*

*An independent methodology was used, in which each group could progress at its own pace, with supervision and support from the instructor, who provided advice, guidance, and assistance whenever doubts or problems arose. A key aspect to highlight was the wide functionality of the ESP32-S3 in fulfilling its purpose, thanks to the instructions provided in the laboratory guide and by the professor. LEDs lit up, buttons were pressed, counters appeared on the displays, and challenges in code generation allowed creativity and knowledge to stand out.*

*In the end, in this laboratory practice, the ESP32-S3 proved to be an effective tool for learning basic electronics and programming, enabling students to understand the use of GPIO pins and the logic of digital inputs and output.*

# **INTRODUCCIÓN**

En presencia del ESP3-S3 y demás elementos como los leds, interruptores DIP, pulsadores y displays de 7 segmentos, Jumpers, permitiéndonos adquirir una comprensión práctica de lo básico entre la electrónica digital y su relación directa con la programación. Se nos provee una serie de ejercicios automatizados y electrónicos con el uso de estos materiales. Se busca la manipulación de estos para la instalación de su IDE, en este caso Jhonny, para la programación en MicroPython, siguiendo paso a paso las indicaciones que se dan en la guía de la práctica de laboratorio para saber como funciona el microcontrolador ESP3-S3, comprender el manejo de los pine GPIO y la configuración como entradas y salidas, así como el aprendizaje de herramientas del software como del hardware para la resolución de problemas simples y complejos.

# **MÉTODO**

Participantes

* Participaron: **Brayan Smith Bedoya Montealegre y Jhon Sebastián Molina Fierro** estudiantes del curso *microcontroladores,* cursando el **6.º semestre** de Ingeniería en Sistemas (edad: 18–19, M = 18.5; SD = 0.71). Se formaron grupos de **5** integrantes como máximo. Ambos trabajaron en un mismo grupo de forma colaborativa y contaban con conocimientos básicos en electrónica y programación en Python.

Materiales:

* 5 resistencias de 220 Ohmios
* 5 resistencias de 1K Ohmios
* 5 resistencias de 10KOhmios
* 5 leds de colores surtidos
* 3 pulsadores
* 2 Display de 7 segmentos
* Jumpers
* Cable TIPO-C – TIPO-C
* 1 computador portátil
* 1 protoboard

Procedimiento paso a paso:

1. Primero se descarga e instala la última versión de Python a poder ser, para luego instalarla, para confirmar su instalación ingresamos al CMD y escribimos el comando *python*, se deberá ver como en la Figura 1, salimos en la consola de Python con el comando *exit* e instalamos la librería ESPTOOL con el comando *pip install esptool* o *Python -m pip install esptool*, en la Figura 2 se muestra como debe verse. Finalmente se descargan los drivers del ESP3-S3, se actualizan y se descarga e instala el IDE Thonny para la programación de los pines y demás capacidades del microcontrolador.

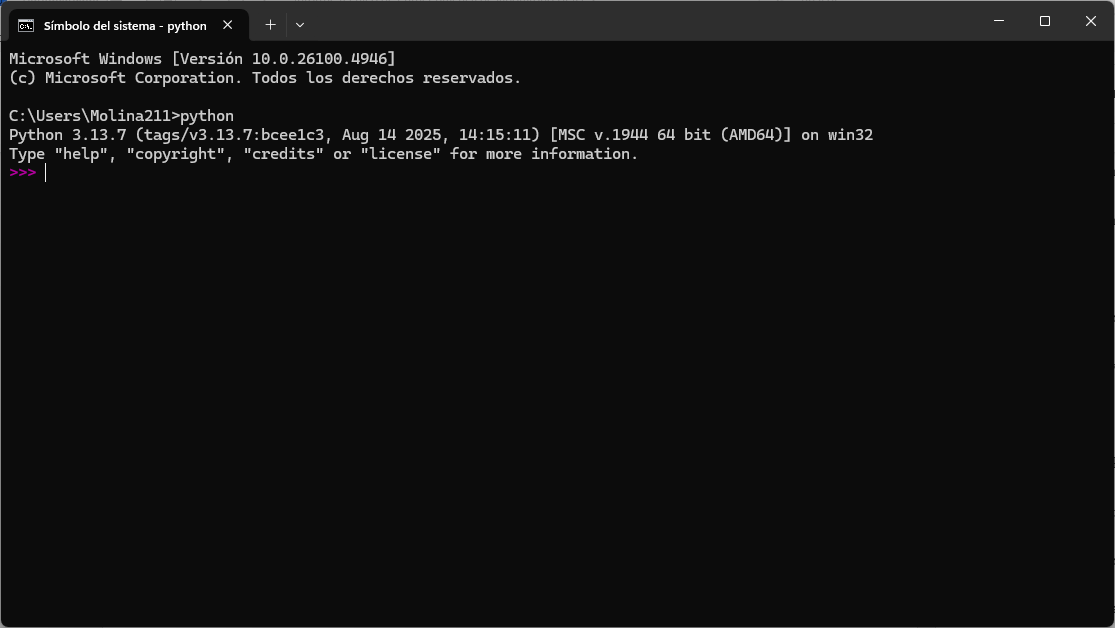


Figura 1 - Revisión de Python descargado

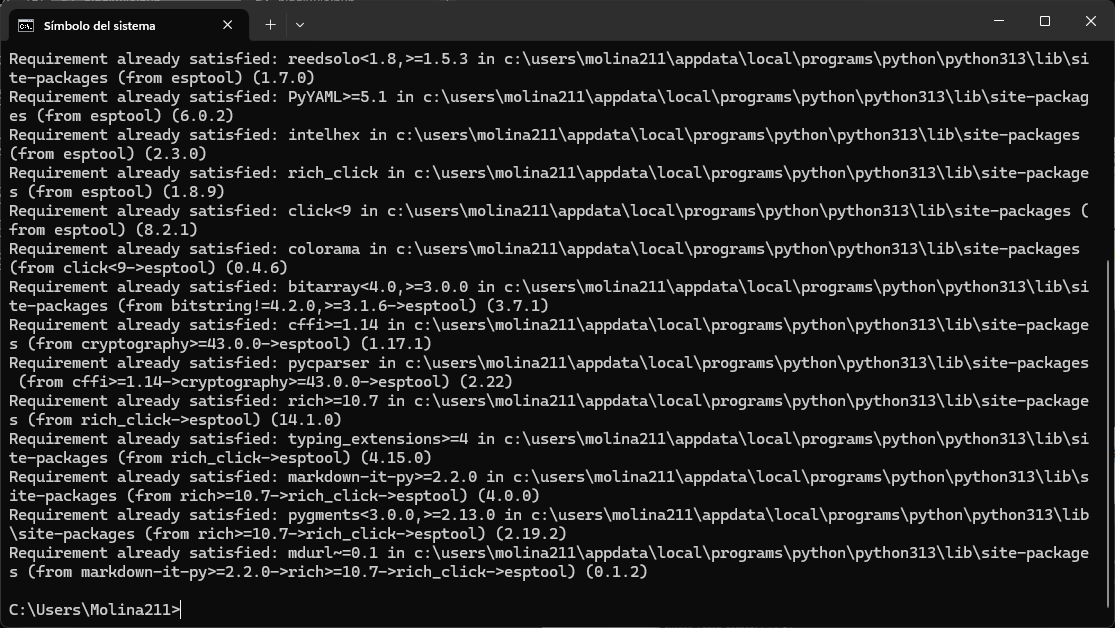


Figura 2 - Descarga de la libreria ESPTOOL

1. Se abre el IDE Thonny y lo configuramos para que tuviera conectividad con el ESP3-S3, nos dirigimos a “Ejecutar” ubicado en la parte de arriba en la barra de opciones y seleccionamos “Configurar interprete” como se muestra en la Figura 3. Después, dirigiéndonos a la Figura 4, en “Opciones de Thonny” se selecciona “MicroPython (ESP32)” como el lenguaje con el que vamos a ejecutar y programar el microcontrolador, y en “Puerto o WebREPL” seleccionamos el único disponible con la terminación de (COM2) o (COM3) o (COM5) o cualquiera digito visible. Ahora, seleccionamos “Instala o actualiza MicroPython (esptool)”, se nos abrirá otra ventana, como se ve en la Figura 5, donde en el apartado de “MicroPython family” seleccionamos “ESP32-S3”, en “variant” seleccionamos “Espressief · ESP32-S3”, y en “versión” seleccionamos “1.26.0”, presionamos el botón de instalar y listo, esta todo listo para que mi equipo y yo empecemos con la manipulación del microcontrolador y el desarrollo de las actividades.

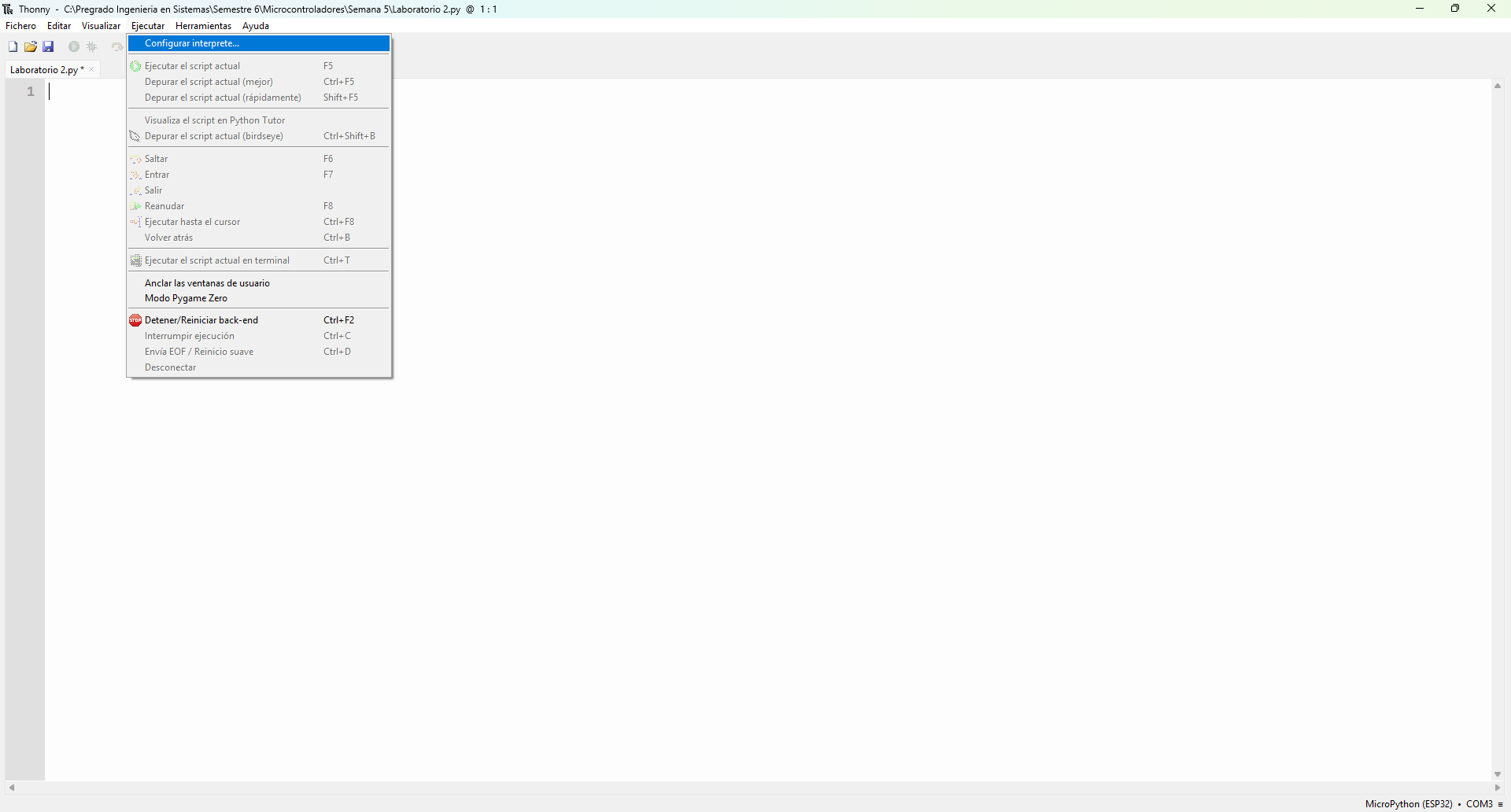


Figura 3 - Configuración de Thonny 1

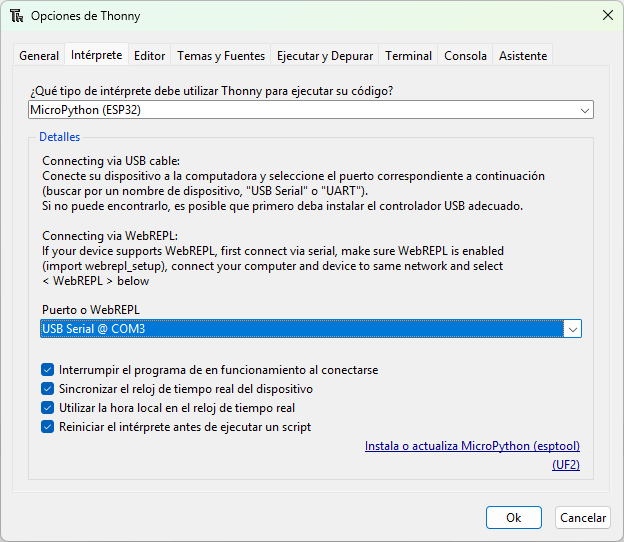


Figura 4 - Configuración de Thonny 2

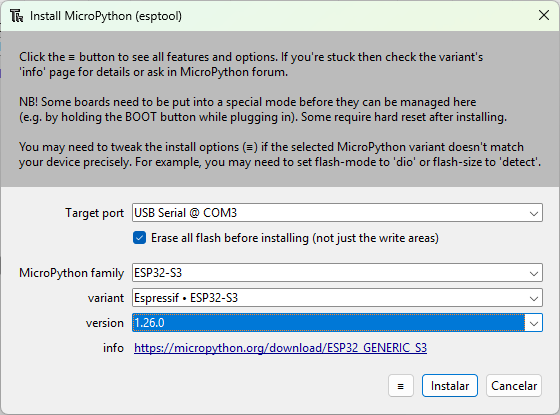


Figura 5 - Configuración de Thonny 3

1. En este primer ejercicio se nos pidió algo básico, encender un led, para ello, ubicamos el led, específicamente su palo largo (ánodo) en la posición del Pin 13 y el otro palo corto (cátodo) con la resistencia de 220 Ohmios que tiene el otro extremo a tierra (-), junto con un Jumper Father-Father desde GND a tierra (-), con el circuito hecho en la Figura 6, se nos pidió ejecutar este código en el Thonny visto en la Figura 7, donde en resumen lo que hace es encender y apagar el led cada 0.2 segundos, dando un mensaje de “Len ON” cuando se encienda y “Len OFF” cuando se apague.

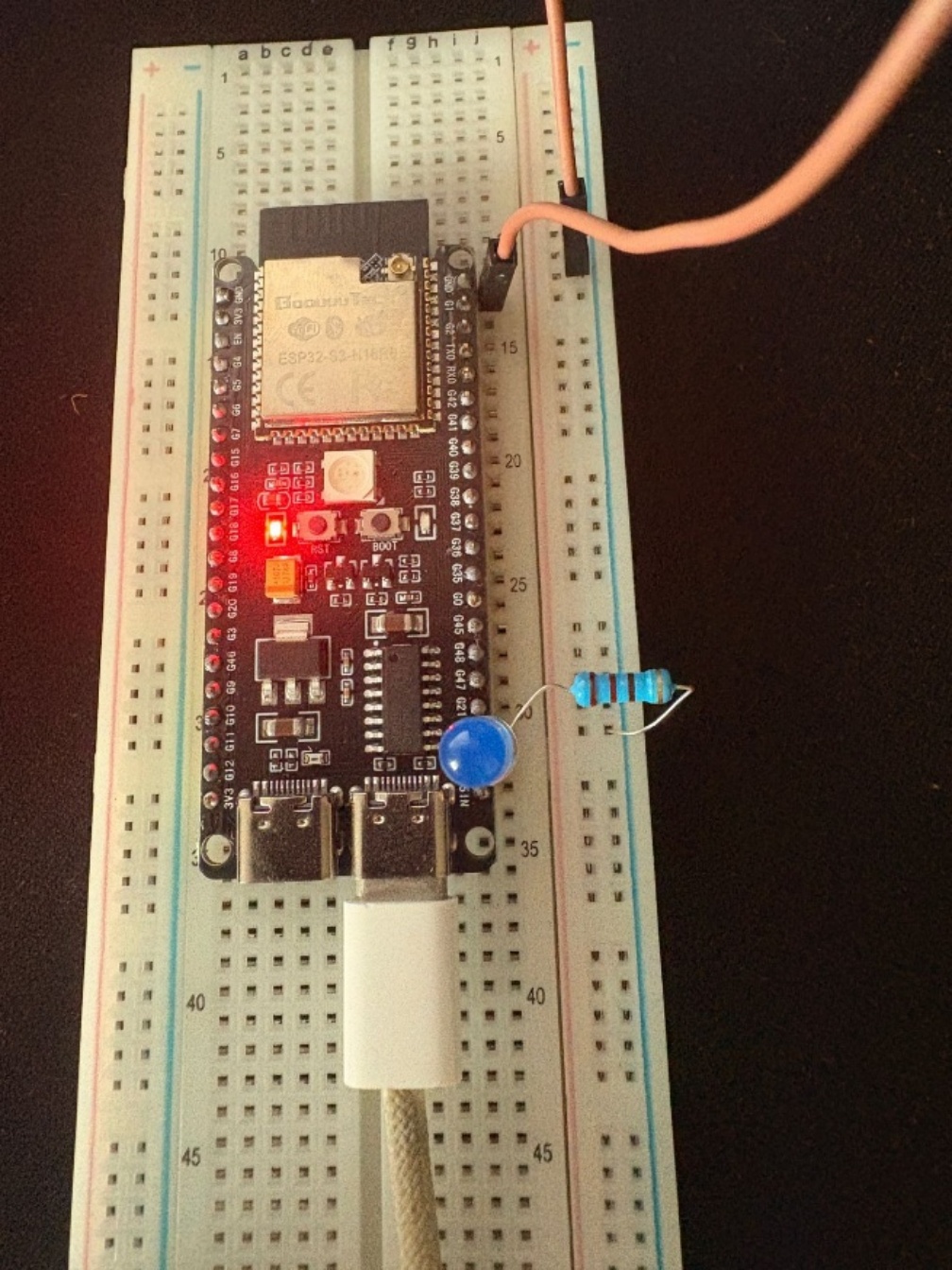


Figura 6 - Primer circuito - Encender Led

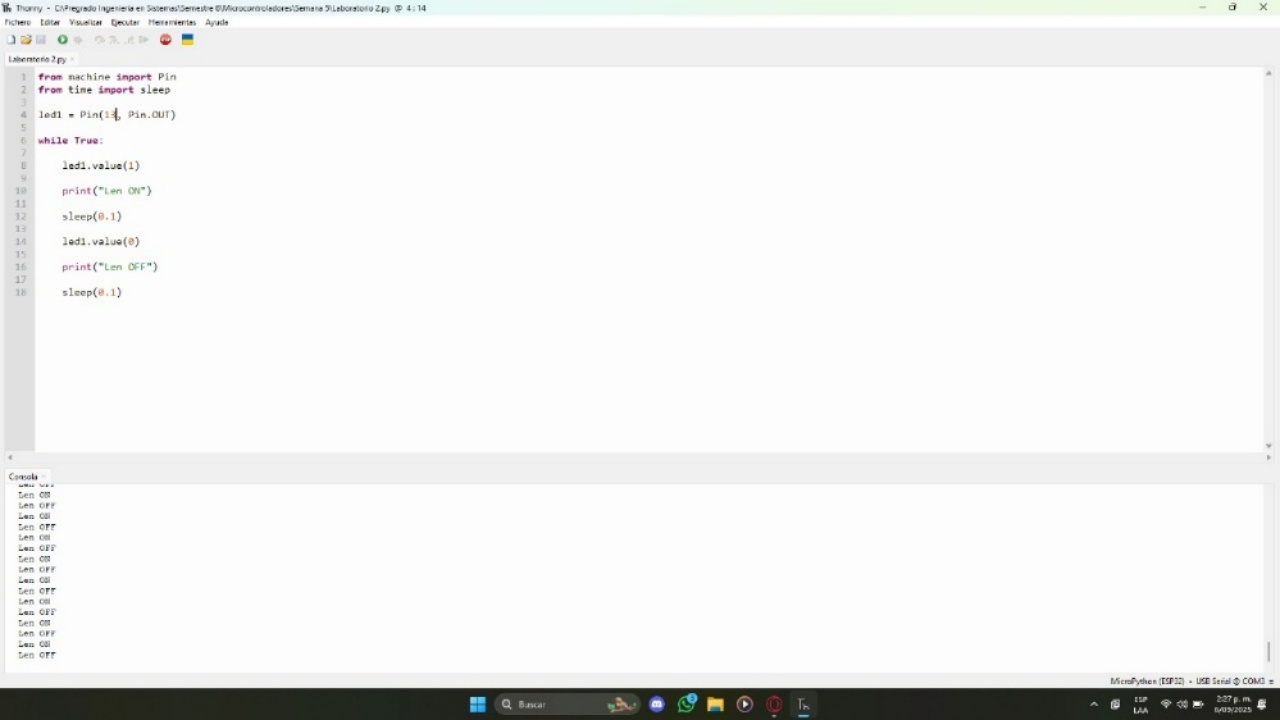


Figura 7 - Primer código - Encender Led

1. Con el circuito de la Figura 6, nos piden una nueva adición, hacer que un pulsador apague el led cuando se presione (Figura 9) y se encienda cuando deje de estar presionado (Figura 8), para completar el circuito usamos un pulsador de 2 puntos, uno en el Pin 37 y el otro en tierra (-), nada más, ahora, en el apartado del código (Figura 10), definimos el pulsador como entrada y que al presionar de “0” y al no presionarlo “1”, dando como resultado que se encienda y apague el led a voluntad, con el añadido de ver el estado del pulsador y el mensaje de “Len ON” cuando este encendido el led.

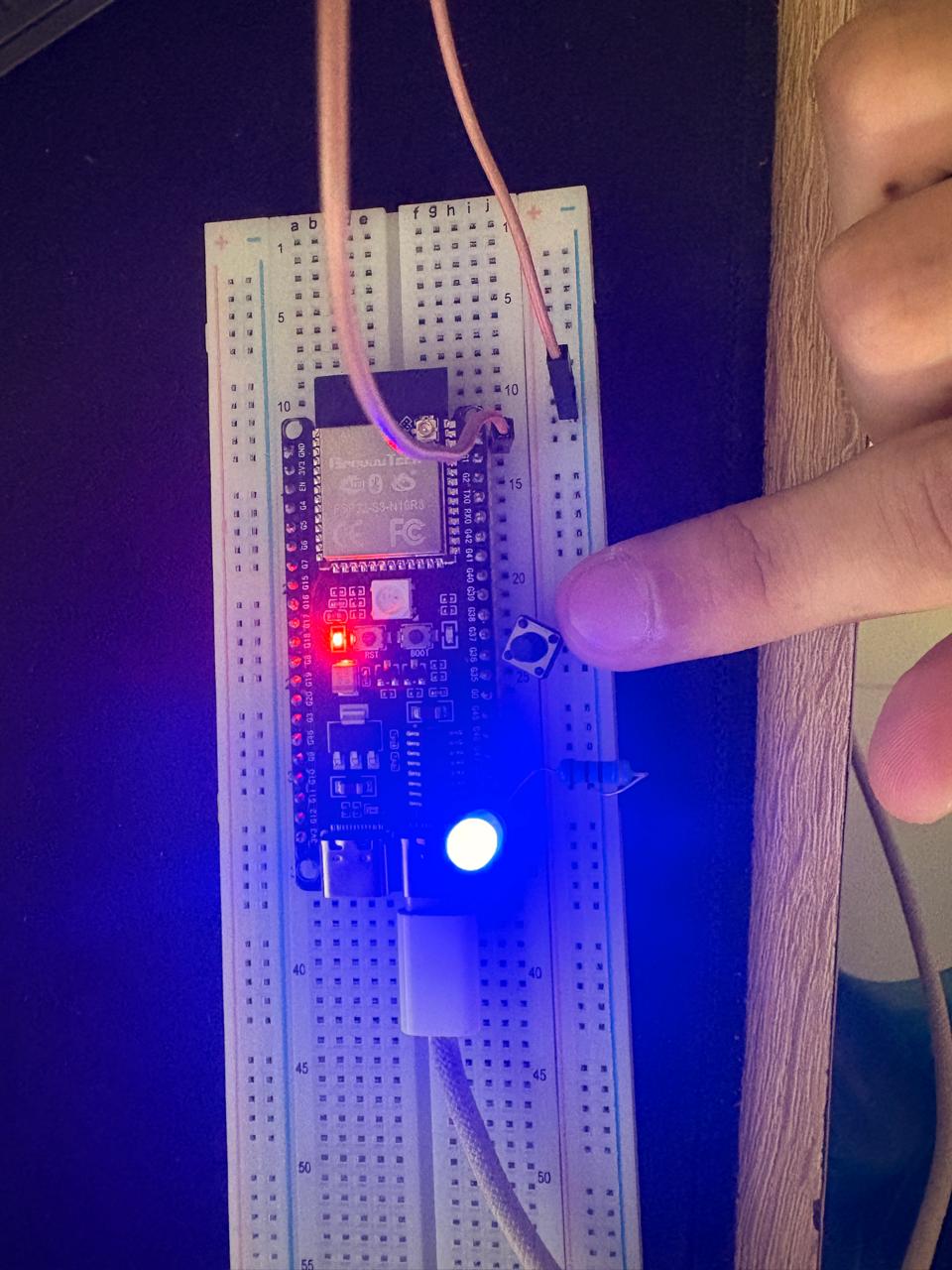


Figura 8 - Segundo circuito - Encender y apagar led a voluntad

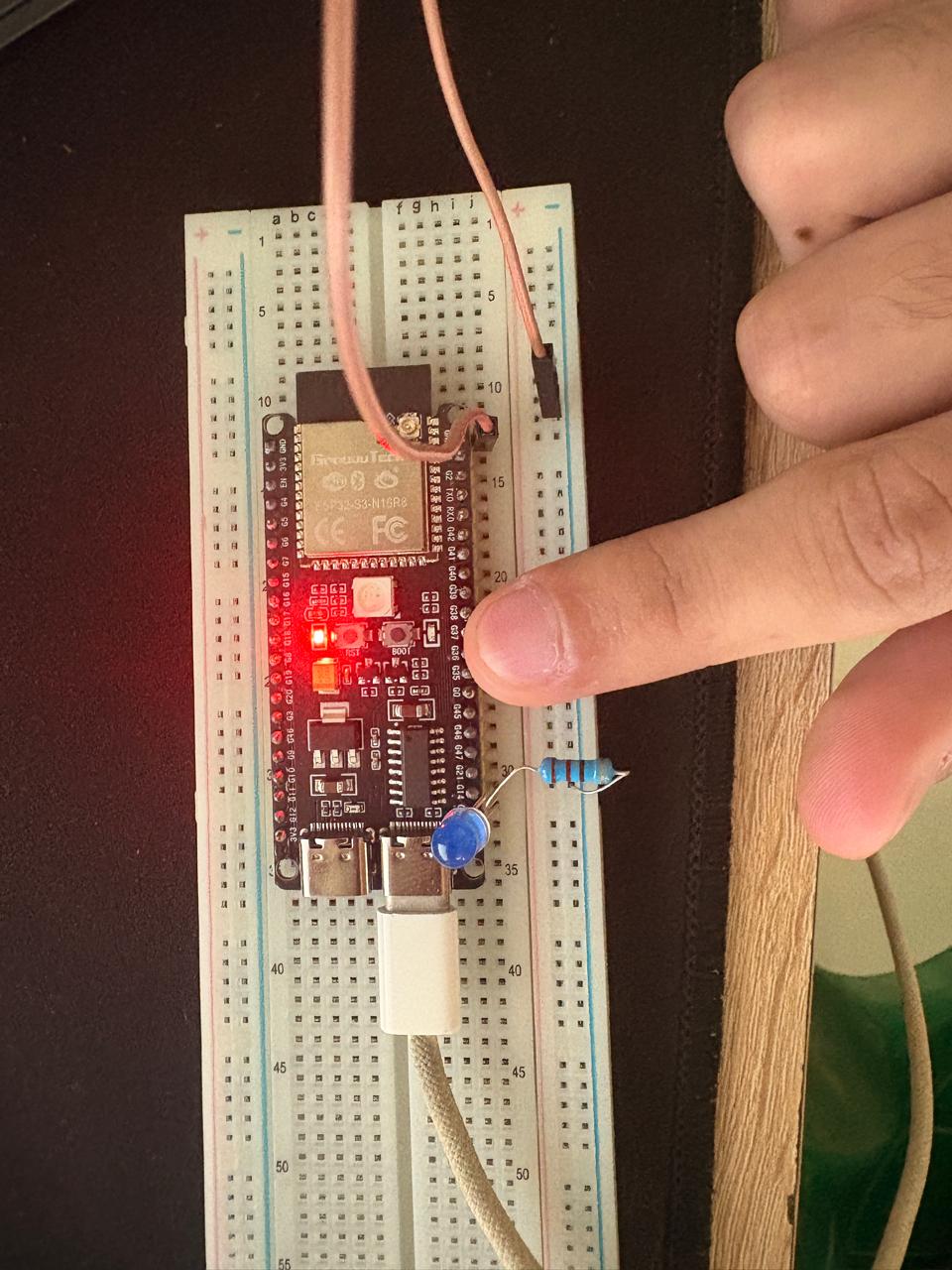


Figura 9 - Segundo circuito - Encender y apagar led a voluntad

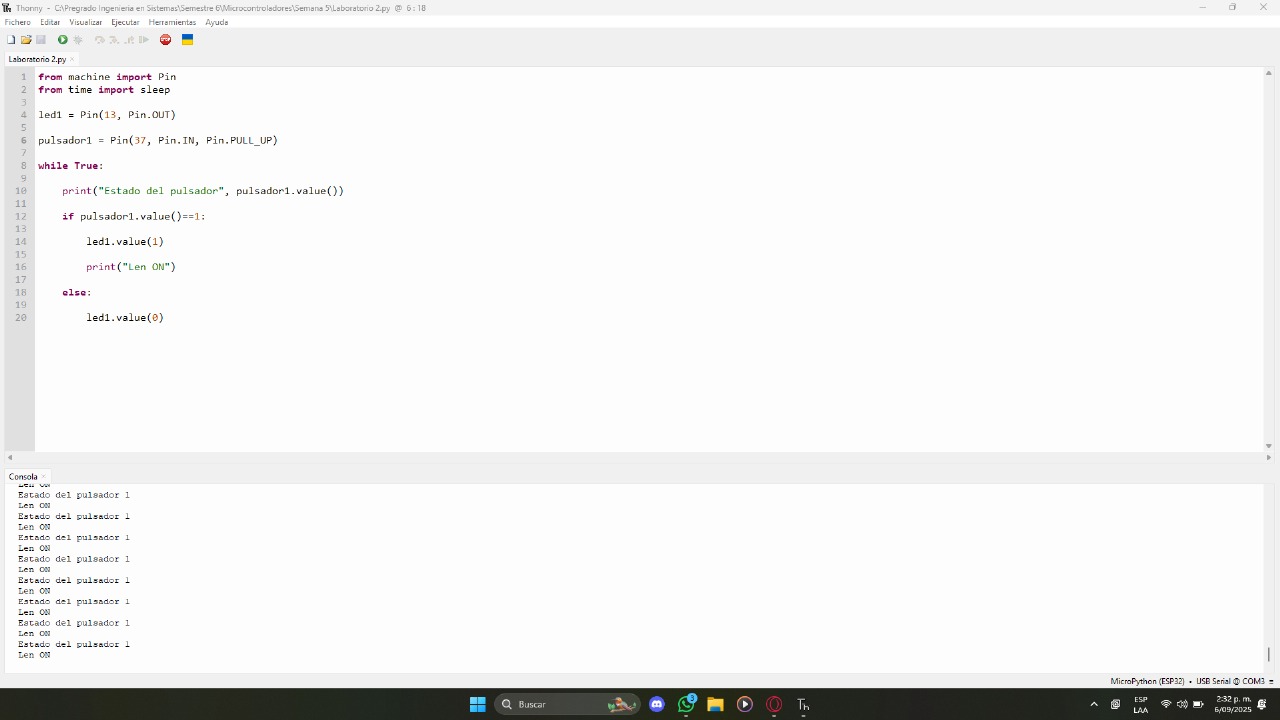


Figura 10 - Segundo código - Encender y apagar led a voluntad

1. En el siguiente ejercicio, se nos pidió hacer una secuencia de led que van desde la izquierda hasta la derecha de manera infinita, para ello, ubicamos los leds en los pine 13, 21 y 48, cada uno de ellos linealmente puestos con su respectiva resistencia de 220 Ohmios junto con el Jumper Father-Father de GND a tierra (-) como se ve en la Figura 11, 12 y 13, ahora, una vez terminado el apartado del circuito seguimos con la programación, donde en la Figura 14 podemos ver que definimos los pines en una lista, luego generamos un “for” para que muestre los pines y los defina como salidas, generamos posiciones de salida en lista para hacer el efecto secuencial, definimos el método “actualizarSalida” para aplicar le efecto secuencial a los pines recorriendo la lista de las salidas 1, 2 y 3, y aplicamos este método a cada efecto secuencial. Como producto final logramos encender 3 leds de manera secuencial exitosamente (Figura 12) (Figura 13) (Figura 13).

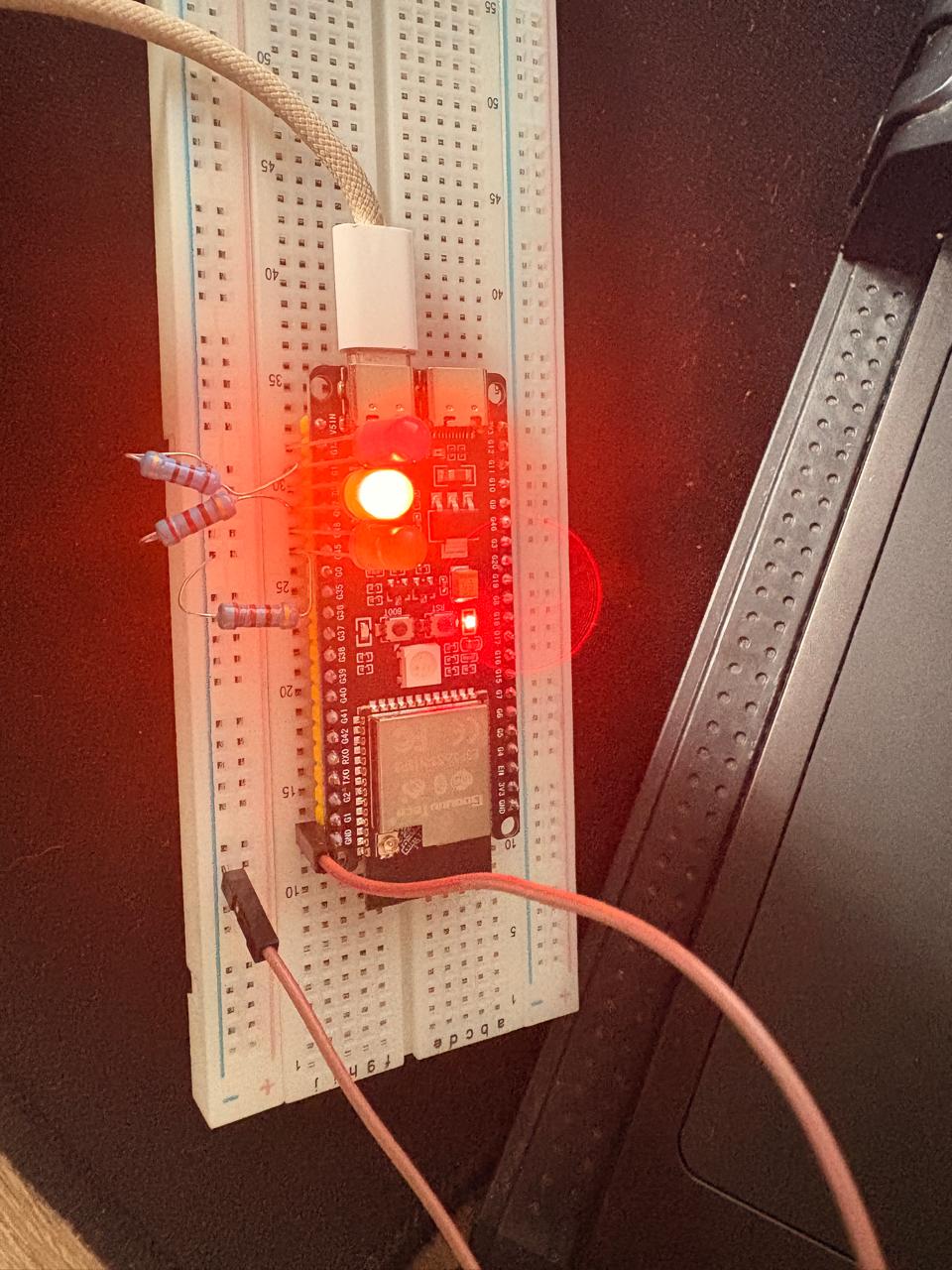


Figura 11 - Tercer circuito - Secuencia de leds

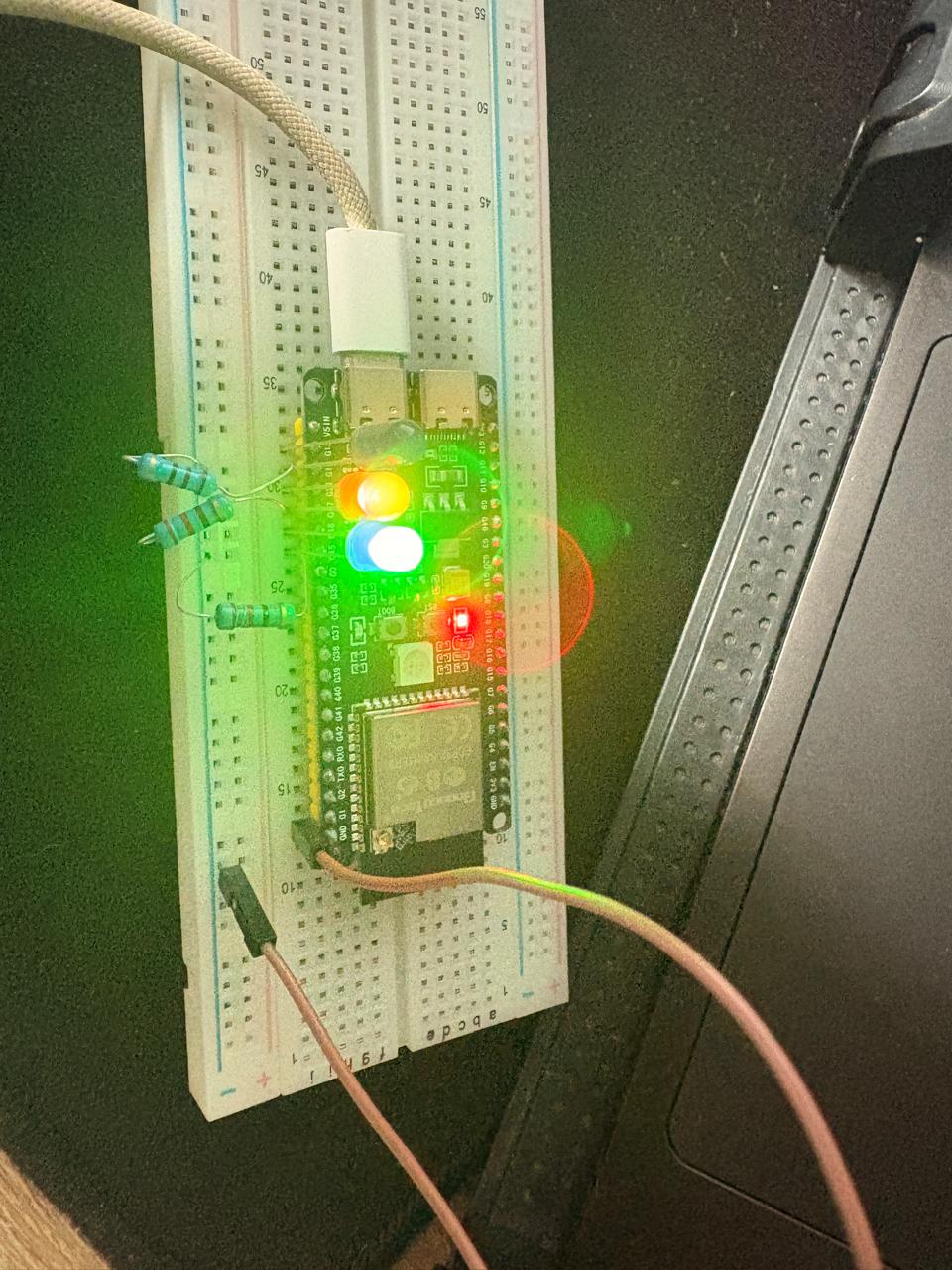


Figura 12 - Tercer circuito - Secuencia de leds

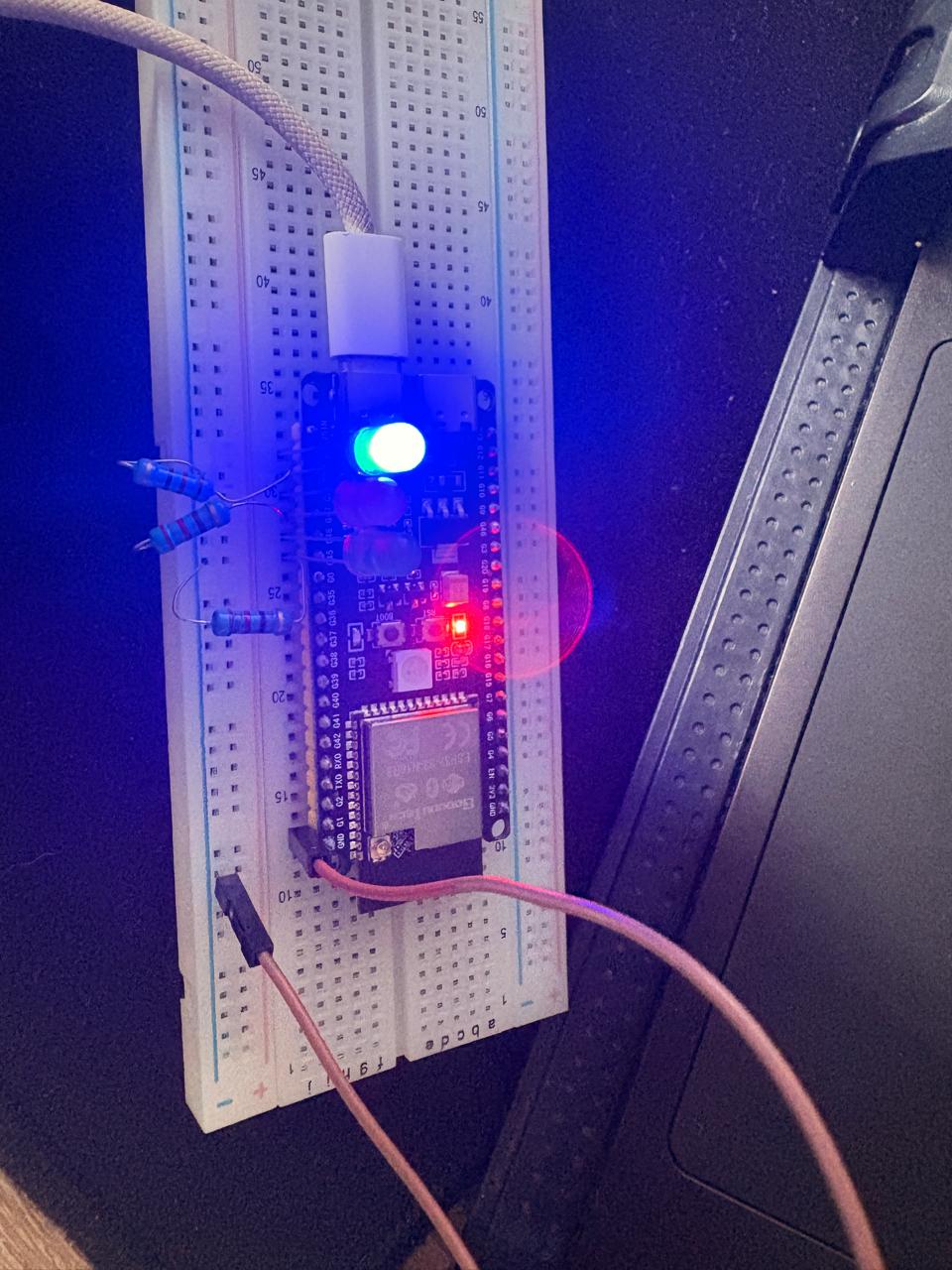


Figura 13 - Tercer circuito - Secuencia de leds

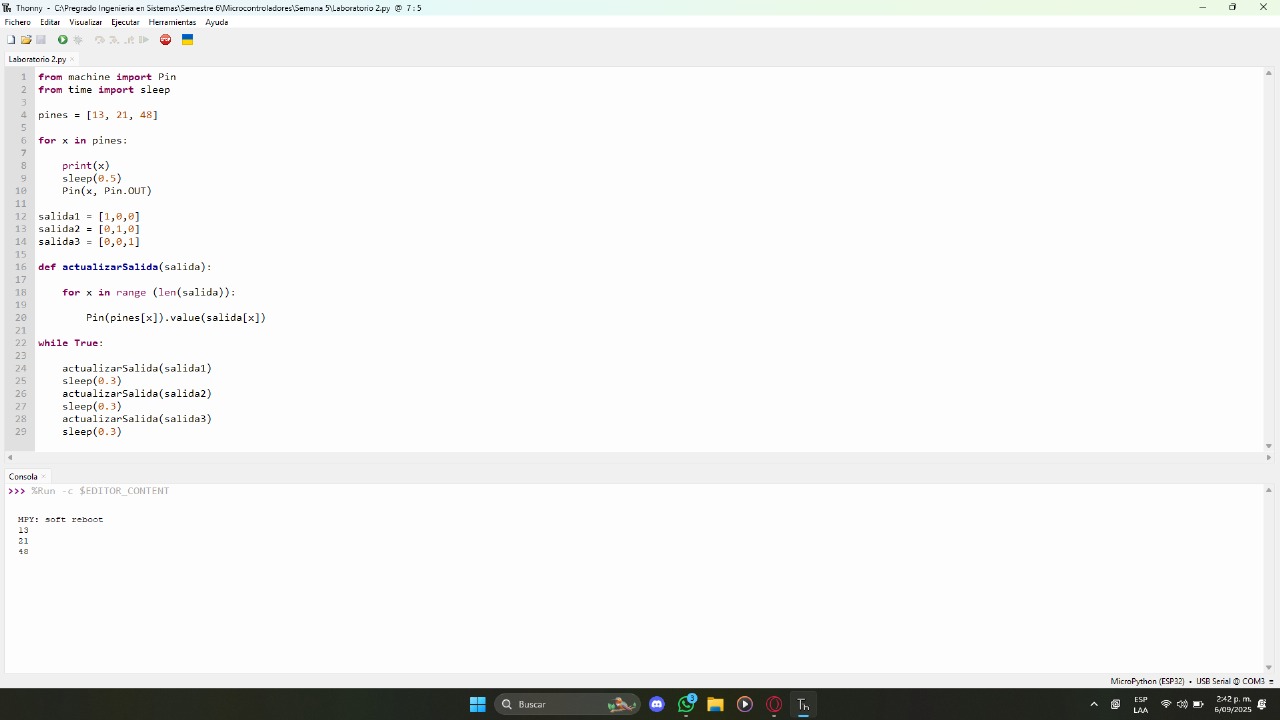


Figura 14 - Tercer código - Secuencia de leds

1. Como reto, se nos pidió diseñar un sistema que cuando este presionado el pulsador, la secuencia de leds se desplace hacia la derecha y cuando no este presionado el pulsador se desplace a la izquierda, para esto en le circuito simplemente añadimos un pulsador en el Pin 38 como se ve en la Figura 15 y 16, después, en el código si fue un reto, tuvimos que definir el pulsador como entrada y que al presionar de “0” y al no presionarlo “1”, agregamos unos nuevos efectos de secuencial pero en la otra dirección para hacer el efecto de izquierda a derecha y de derecha a izquierda, finalmente aplicamos en el “while True” un condicional con el mismo método usado en el ejercicio base aplicándole también al nuevo efecto secuencial añadido, en otras palabras, si el pulsador no esta presionado se mueve de derecha a izquierda y manda un mensaje “Derecha”, y si esta presionado se mueve de izquierda a derecha mandando otro mensaje “Izquierda”, se evidencia en la Figura 17.

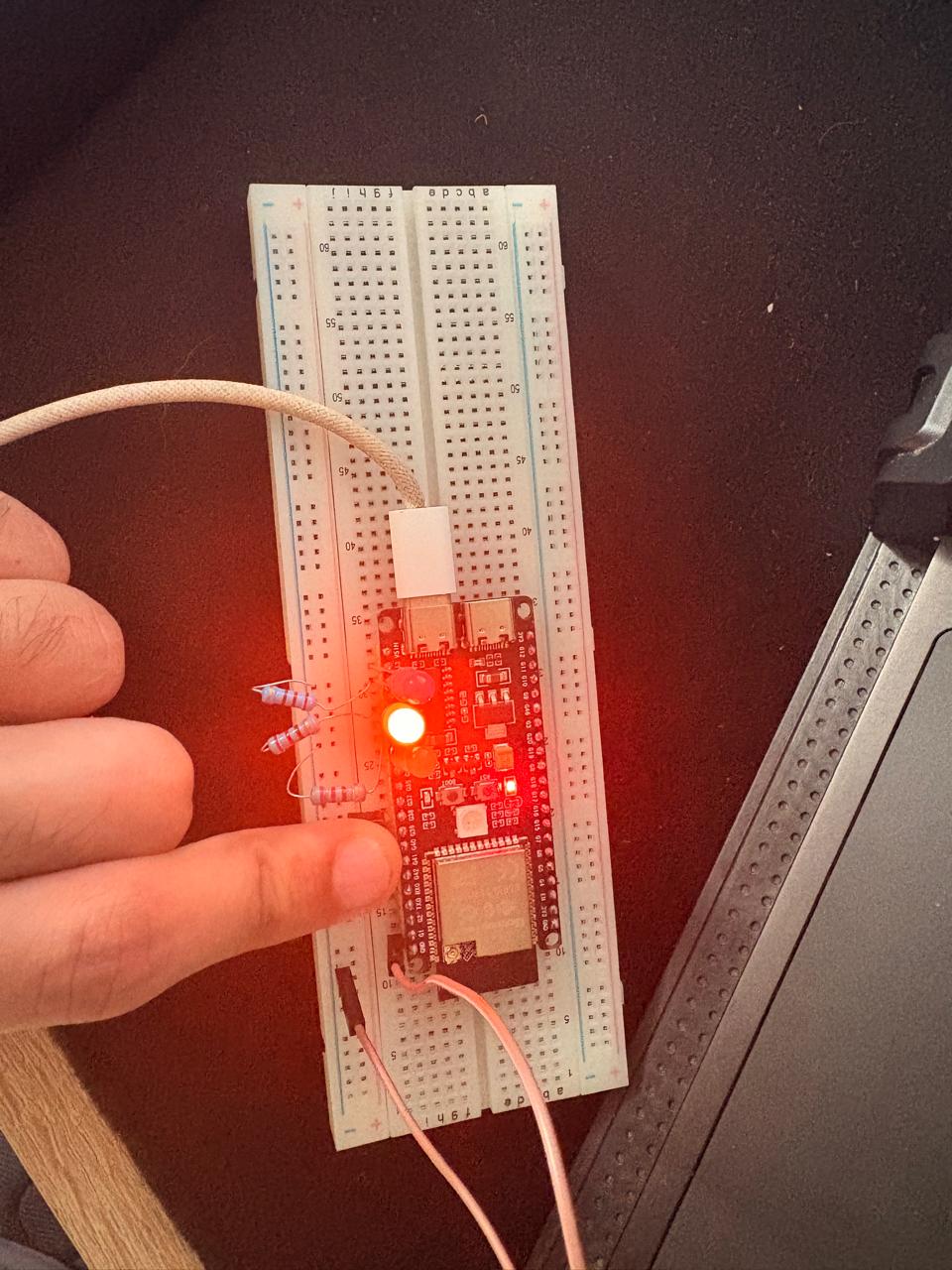


Figura 15 - Cuarto circuito - Secuencia led R-L y L-R

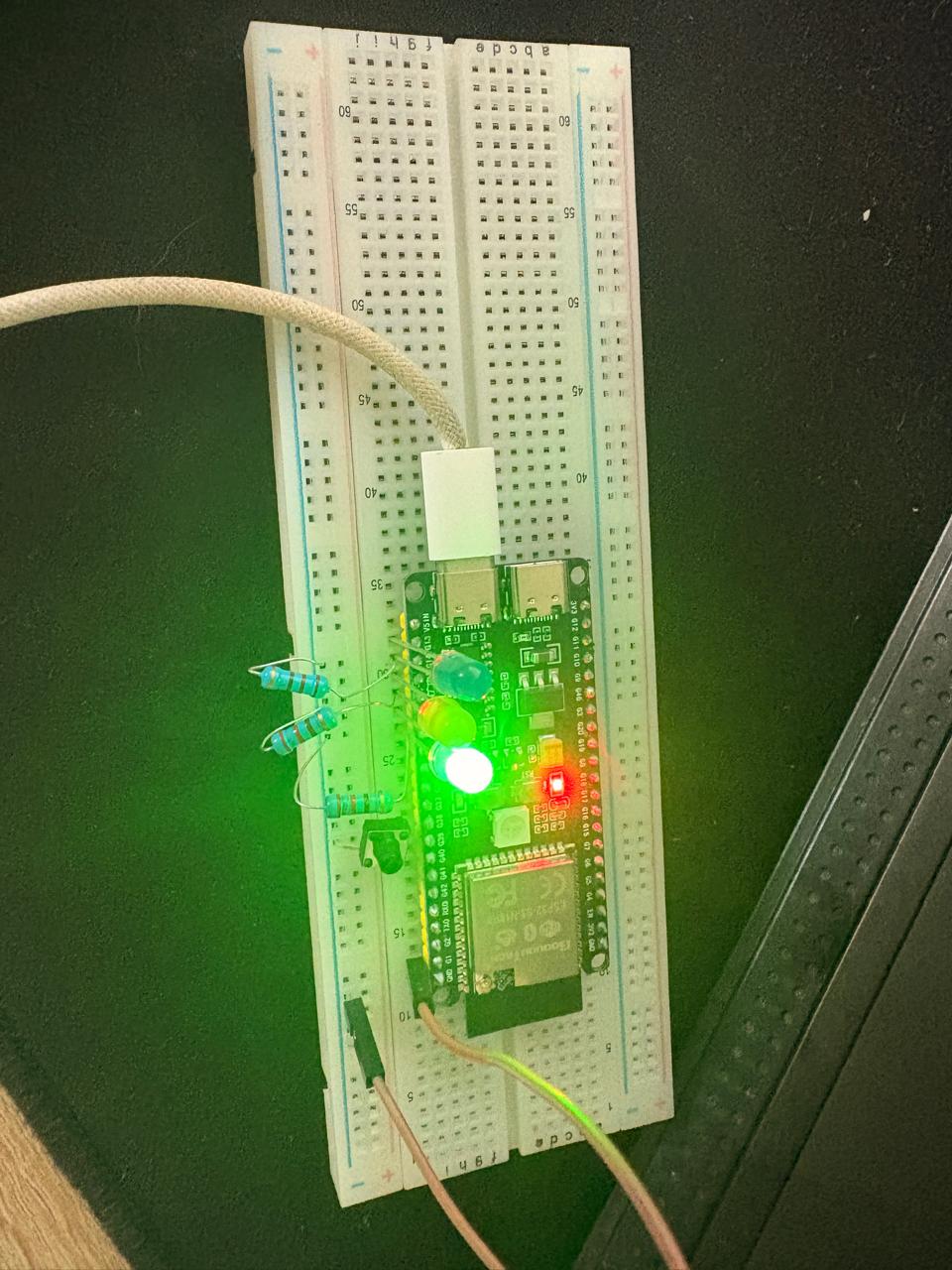


Figura 16 - Cuarto circuito - Secuencia led R-L y L-R

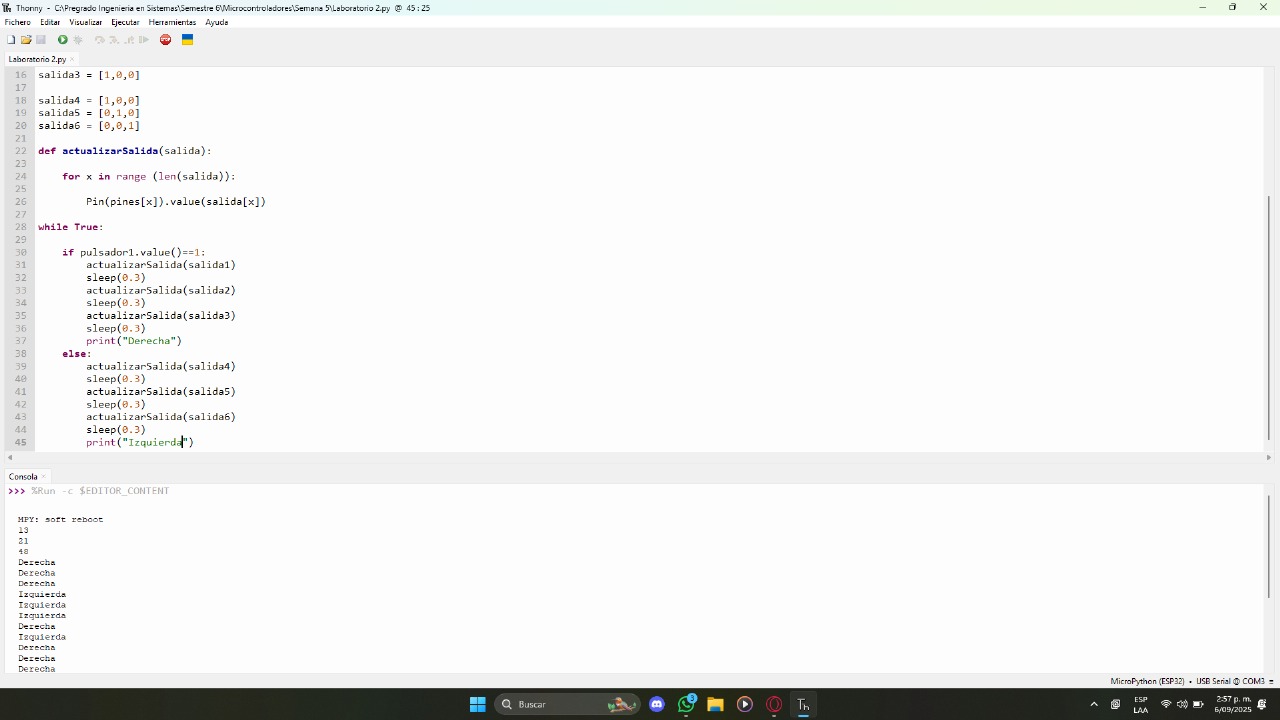


Figura 17 - Cuarto código - Secuencia led R-L y L-R

1. En el siguiente reto, se nos pidió diseñar un sistema de seguridad con pulsadores y leds donde cada uno de los pulsadores se usaron como sensor de una puerta delantera, una puerta trasera y un patio y los leds como alarma silenciosa, sonora y un notificador de SMS.

* **Alarma silenciosa:** Si se abre solo una de las puertas (delantera o trasera), se activará una alarma silenciosa (indicada por un LED amarillo).
* **Alarma sonora:** Si se abre una puerta y se detecta movimiento en el patio, se activará una alarma sonora (indicada por un LED rojo).

Si se activan las tres alarmas (ambas puertas y el sensor de movimiento), se activará la alarma sonora. (indicada por un LED rojo).

* **Notificación SMS:** En caso de que se activen las tres alarmas, se enviará un SMS a un número de teléfono preconfigurado y se mostrara en un led azul.

De esta forma en la Figura 18 podemos ver el circuito, donde los leds mantienen su ubicación con las resistencias del ejercicio de secuencialidad, lo diferente es la agregación de uno de color rojo para seguir los requerimientos del ejercicio, al lado, se evidencia una serie de 3 pulsadores en los pine 36, 38 y 41, donde serán los que hagan alusión a la activación de los sensores para las alarmas (leds). En el código visto en la Figura 24 podemos ver que se definieron los pines de led como salida y los pulsadores como entradas, luego se definieron con un “0” los leds para que se inicien apagados y en “0” los pulsadores para hacer uso de ellos en True y False, en este caso están en True, luego se usaron los valores booleanos tales como *^* y condicionales como *and* y *not* para ejecutar las condiciones necesarias para el sistema de seguridad, encendiendo o apagando su led correspondiente (Figura 19) (Figura 20) (Figura 21) (Figura 22) (Figura 23).

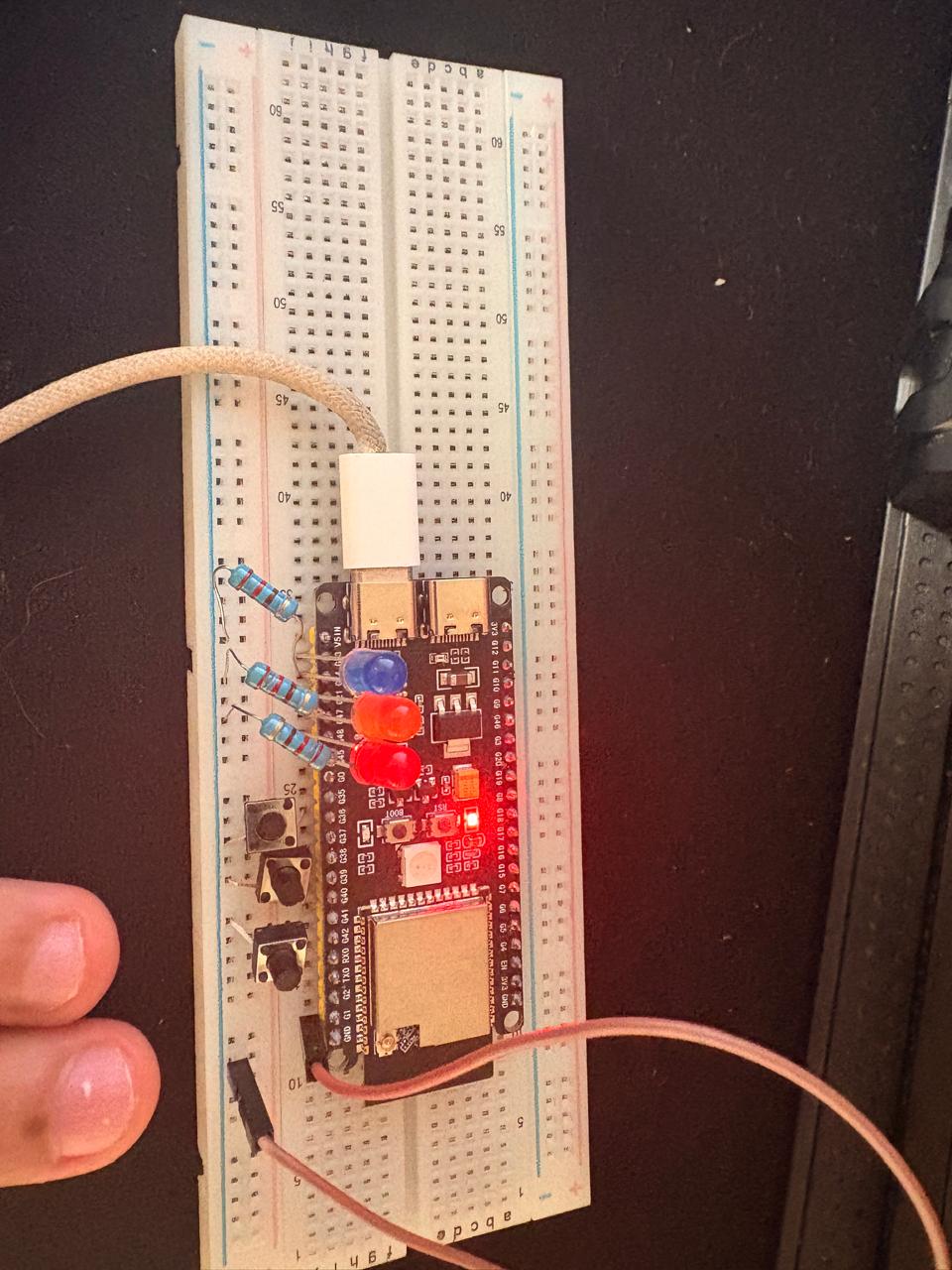


Figura 18 - Quinto circuito - Sistema de seguridad

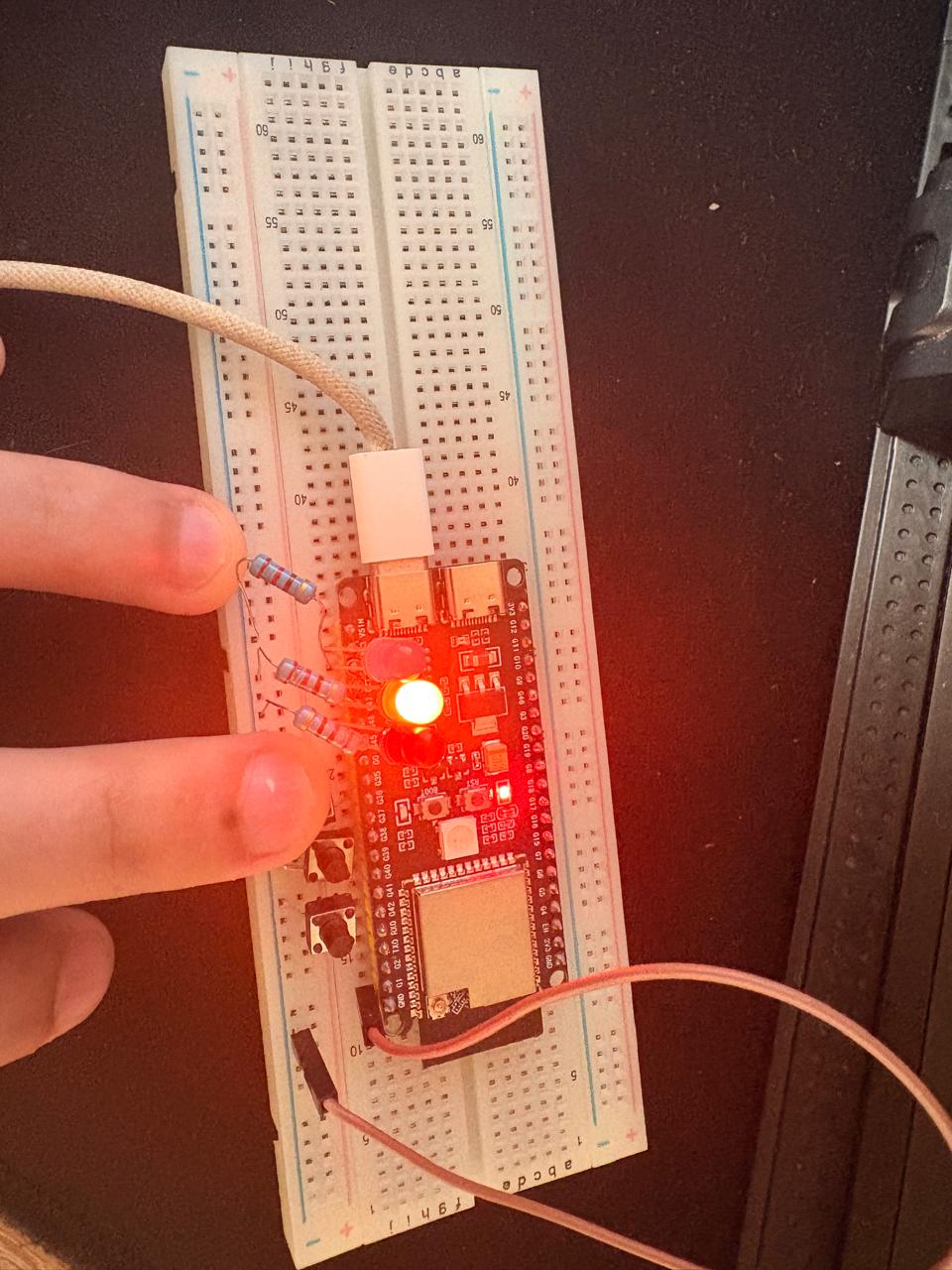


Figura 19 - Quinto circuito - Sistema de seguridad

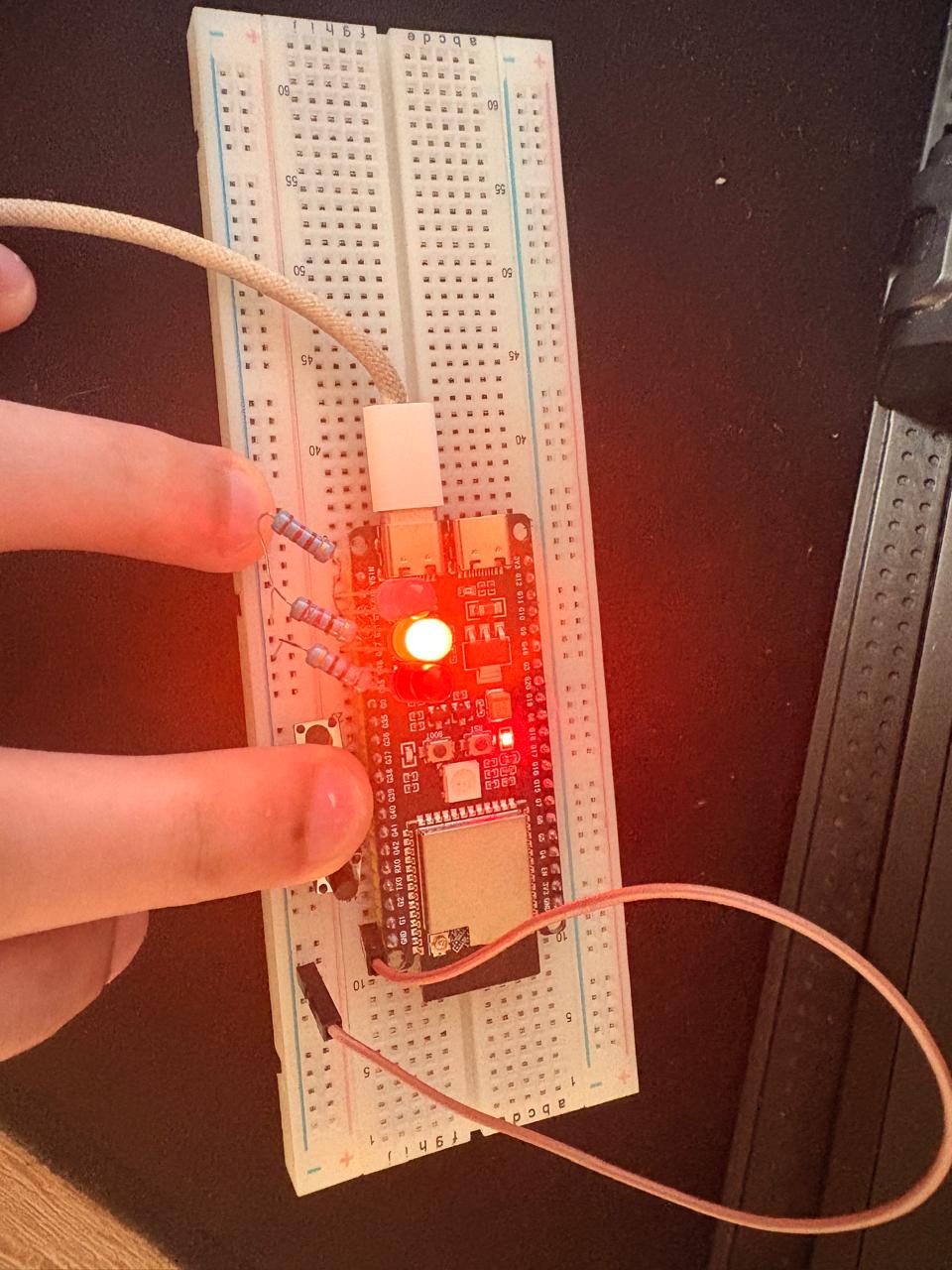


Figura 20 - Quinto circuito - Sistema de seguridad

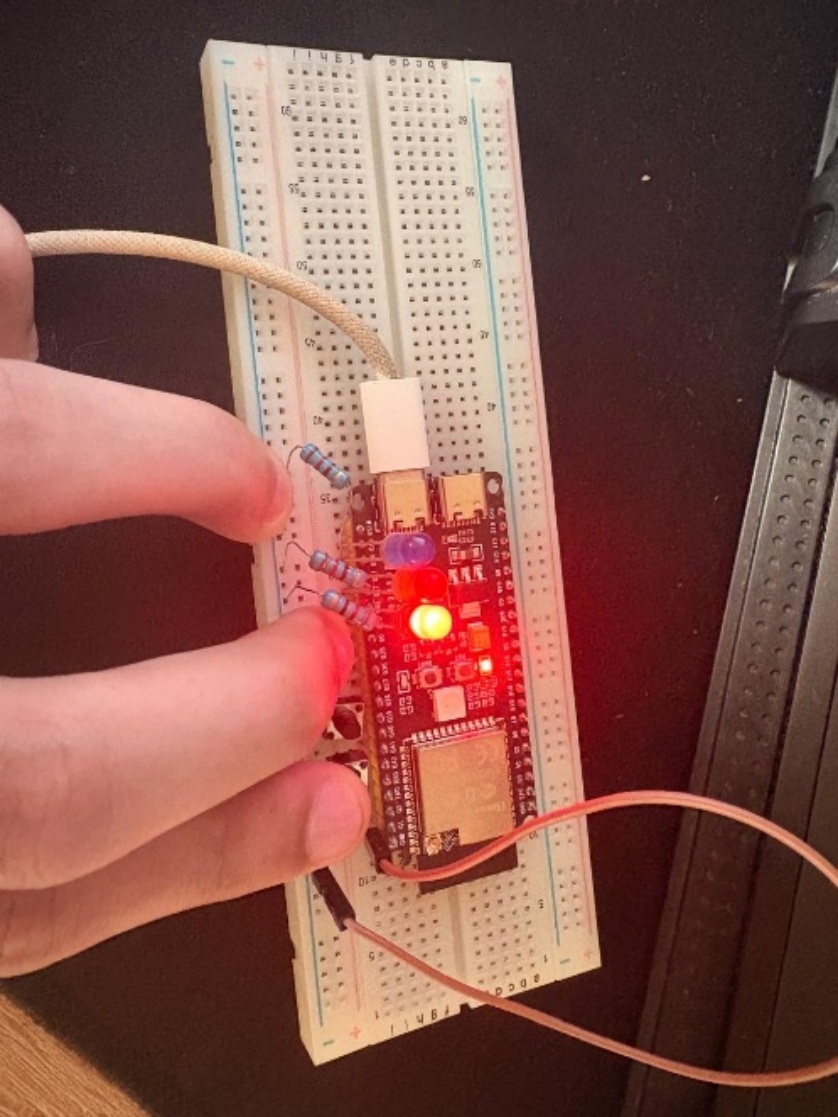


Figura 21 - Quinto circuito - Sistema de seguridad

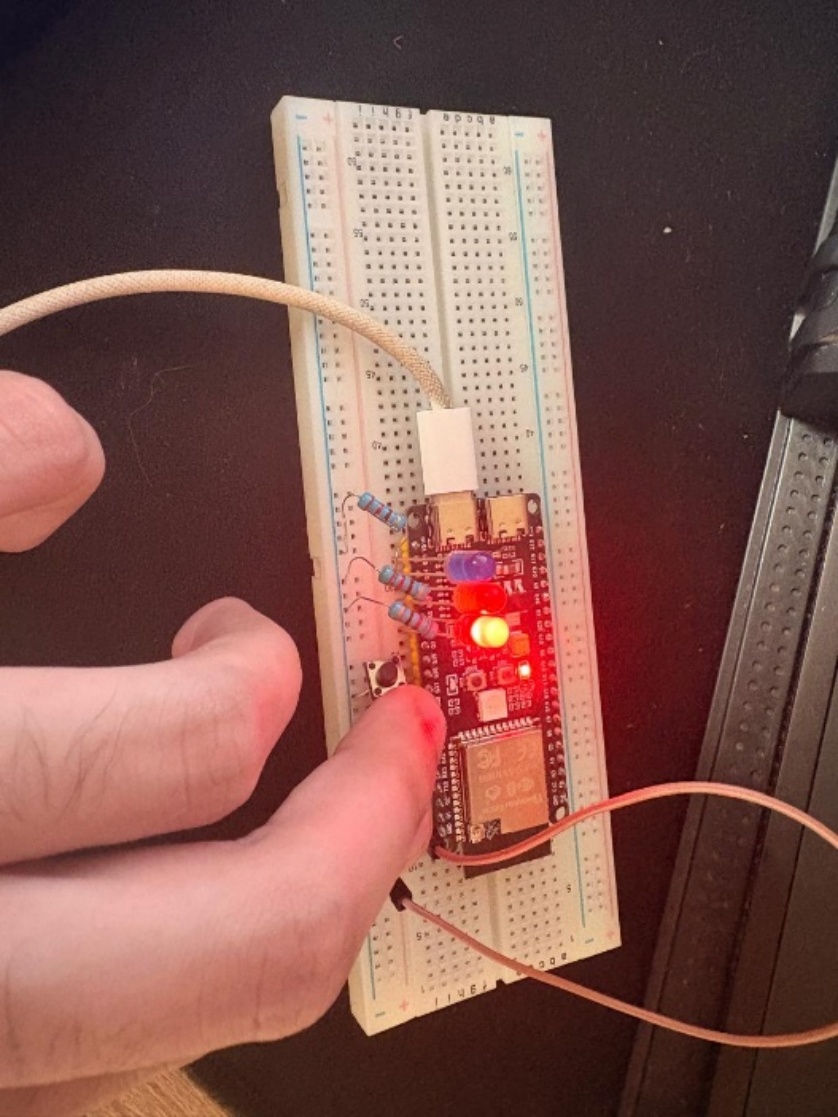


Figura 22 - Quinto circuito - Sistema de seguridad

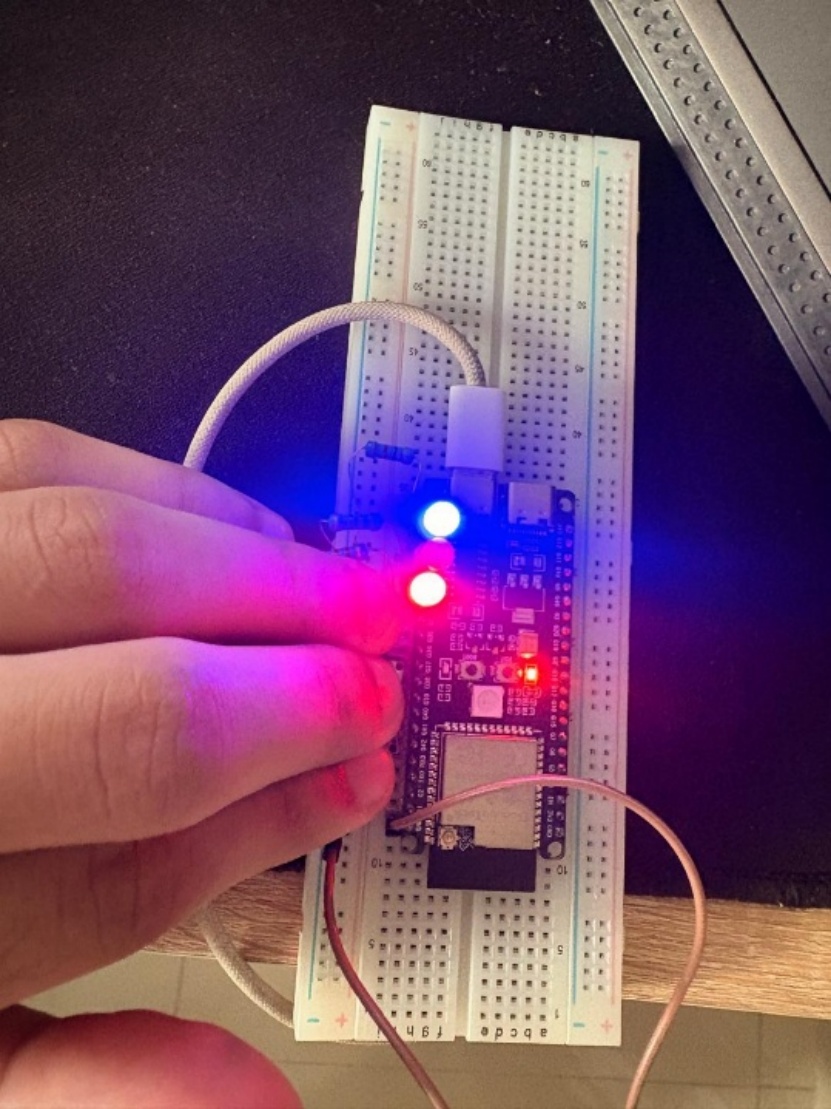


Figura 23 - Quinto circuito - Sistema de seguridad

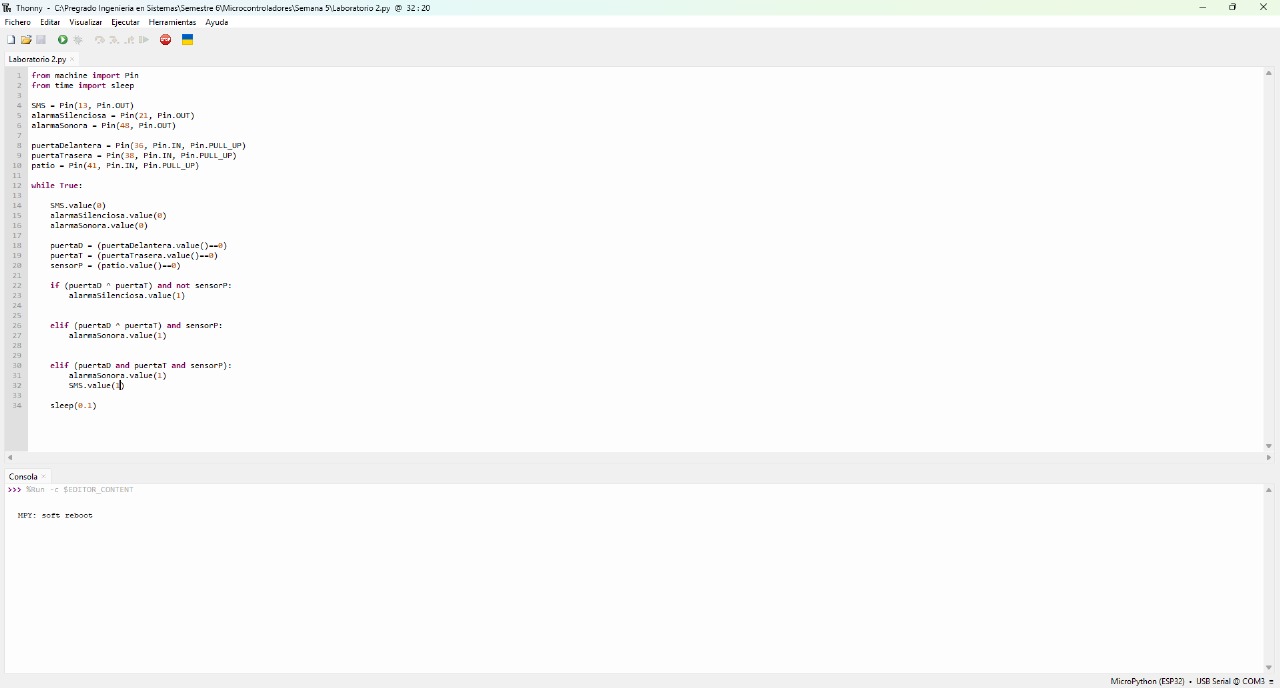


Figura 24 - Quinto código - Sistema de seguridad

1. Para el ultimo ejercicio, se nos pidió montar un sistema contador que muestre los números del 1 al 0 en un display de 7 segmentos en intervalos de 0 a 1 segundo, para esto, en la Figura 25 vemos que