## **Decodificación protocolos con Python (cliente/servidor)**

La siguiente experiencia trata de entender cómo funcionan los protocolos de comunicación, para este caso se dispone de la siguiente trama.



Ilustración 56 Protocolo comunicación

Un cliente “X” envía la trama codificada, para luego el servidor decodificarla y mostrar los datos, además de realizar las acciones correspondientes.

**START BYTE:** es un valor que indica el comienzo de la trama, en este caso será “0xc5”.

**LEN:** es el largo total de la trama en este caso siempre es el mismo largo, por ende, su valor será “0x06””.

**SRC ADDR:** dirección de origen (cliente), como una dirección ip es formada por 4 bytes y en esta trama solo tenemos uno, se utilizará el ultimo octeto.

**DST ADDR:** dirección de destino (servidor), mismo caso que en el “SRCADD”.

**CMD:** comando para activar o desactivar GPIO.

**DATA:** indica que GPIO se debe realizar la acción.

**CHECK:** es la suma de control, indica que no hay perdida de datos en el trayecto de envió de datos.

### **Cliente**

Cliente, es la persona que decide que GPIO activar o desactivar, a simples rasgos, este cliente envía una trama encriptada en hexadecimal, con un checksum para comprobar que los datos son los correctos y no hay perdidas de datos durante el trayecto.

.

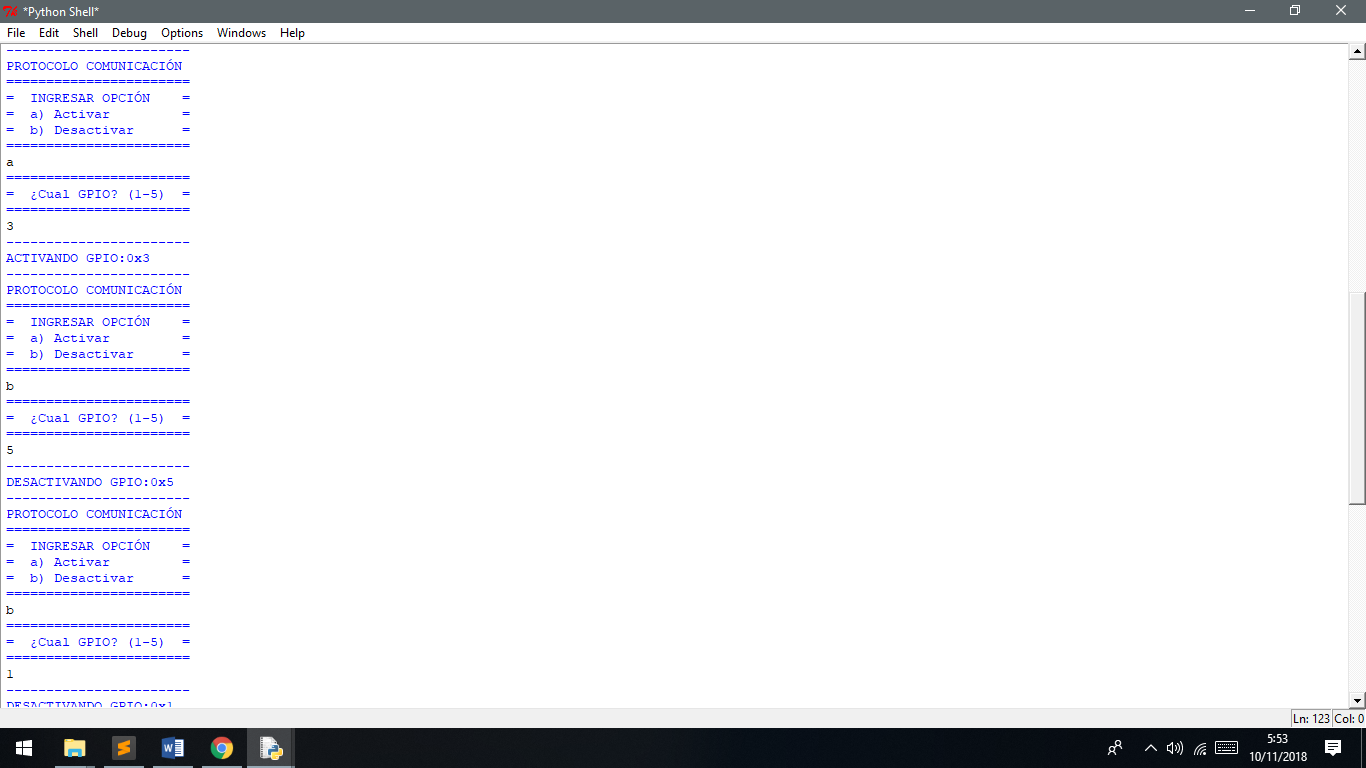


Ilustración 57 Mando Cliente.

### **Servidor**

Este será el encargado de recibir la trama, realizar la verificación del checksum, en caso de que este correcto energizara o des energizara las salidas GPIO de la Raspberry.

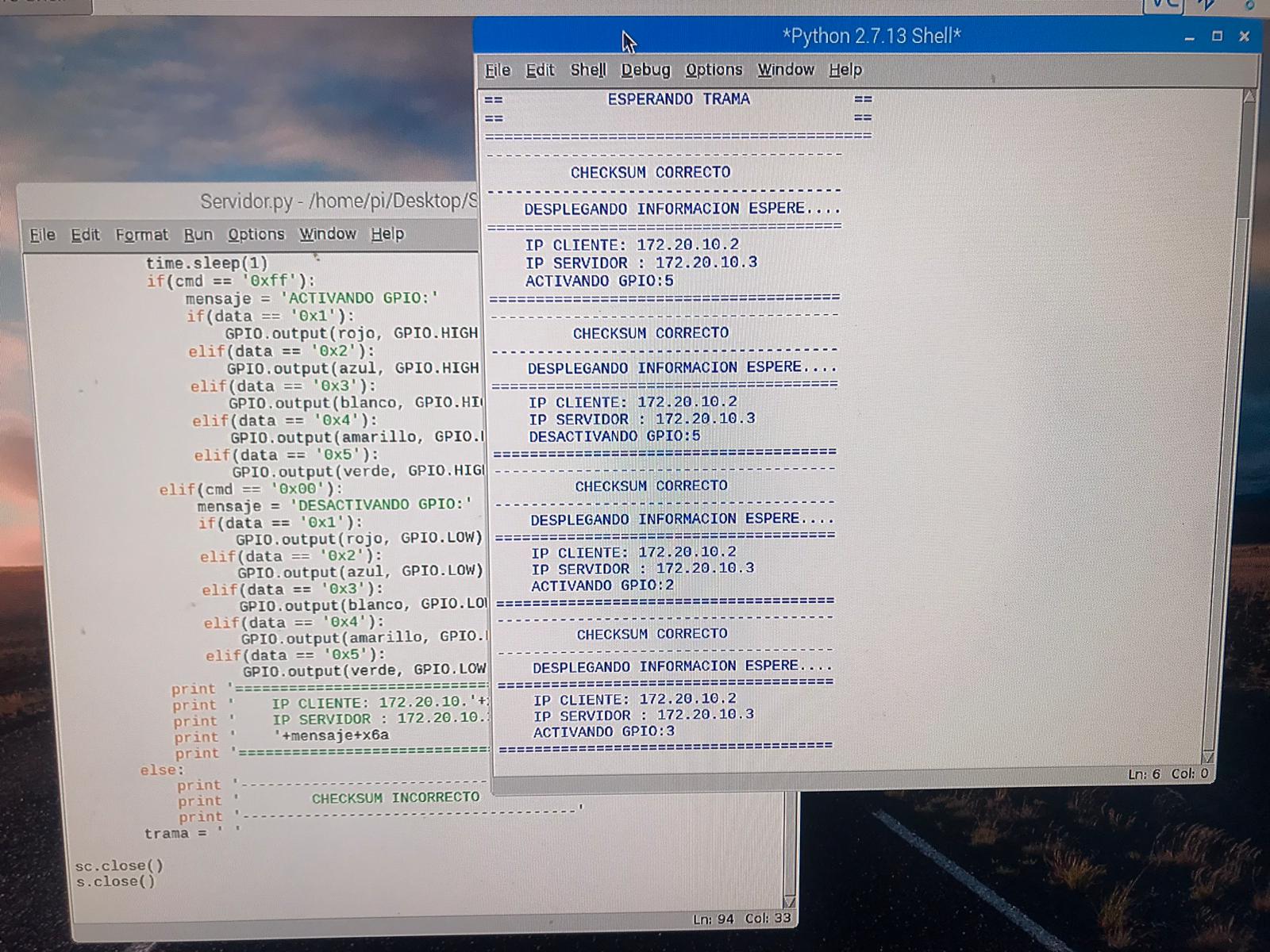


Ilustración 58 Recibo de datos(Servidor)

Los leds se conectan a los GPIO 5, 6, 13, 19, 26 y GND, siendo el siguiente un ejemplo de su conexionado.

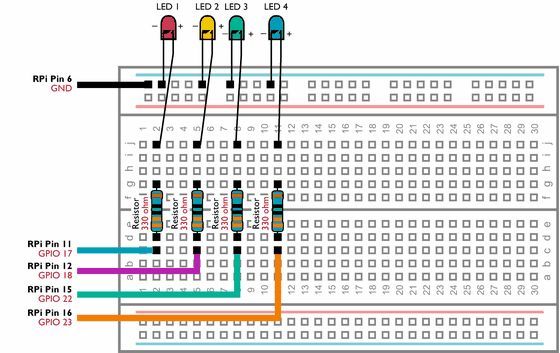


Ilustración 59 Conexionado Leds.

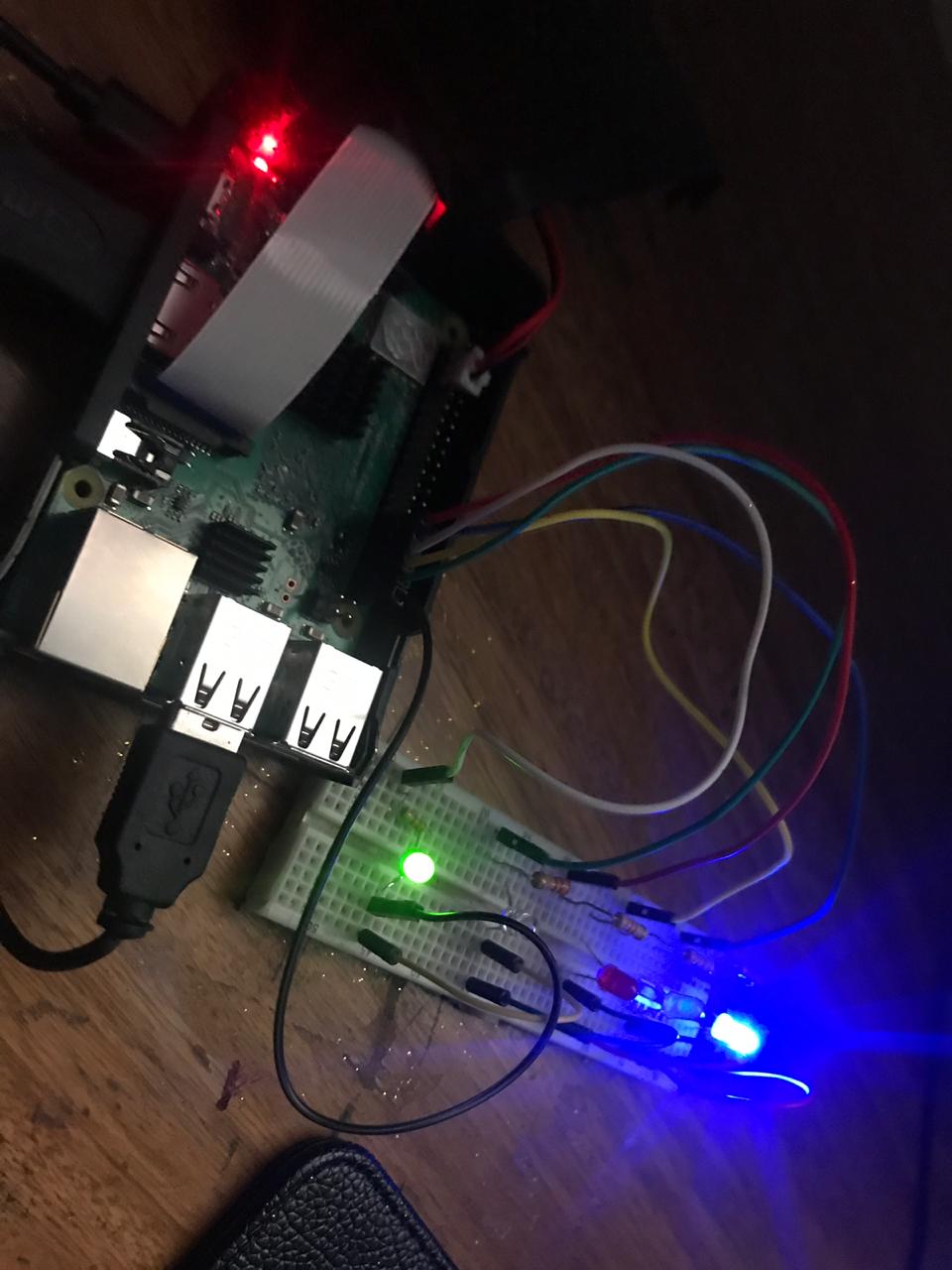


Ilustración 60 Leds Encendidos mediante protocolo TCP/IP

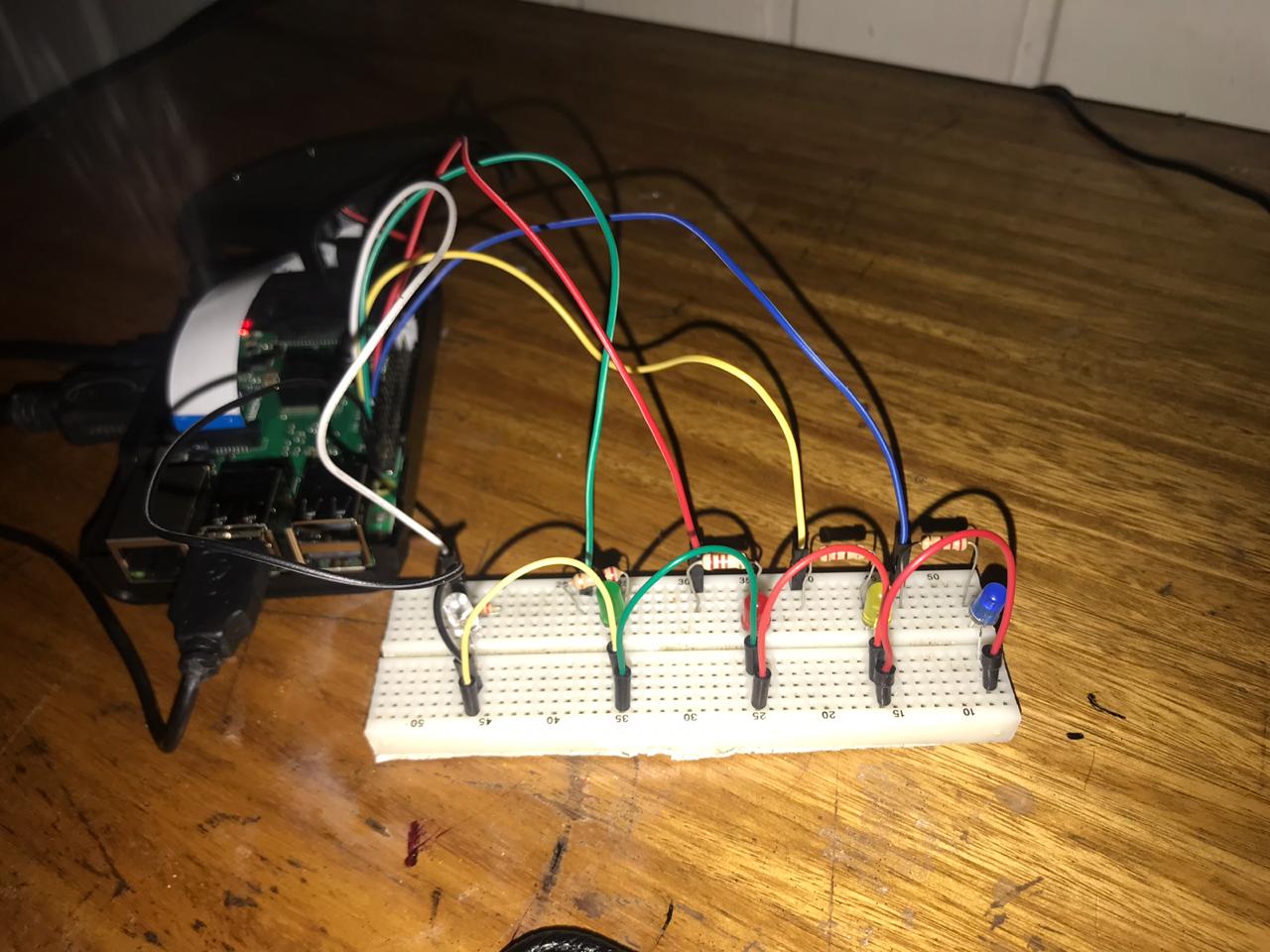


Ilustración 61 Conexionado Leds Raspberry.