

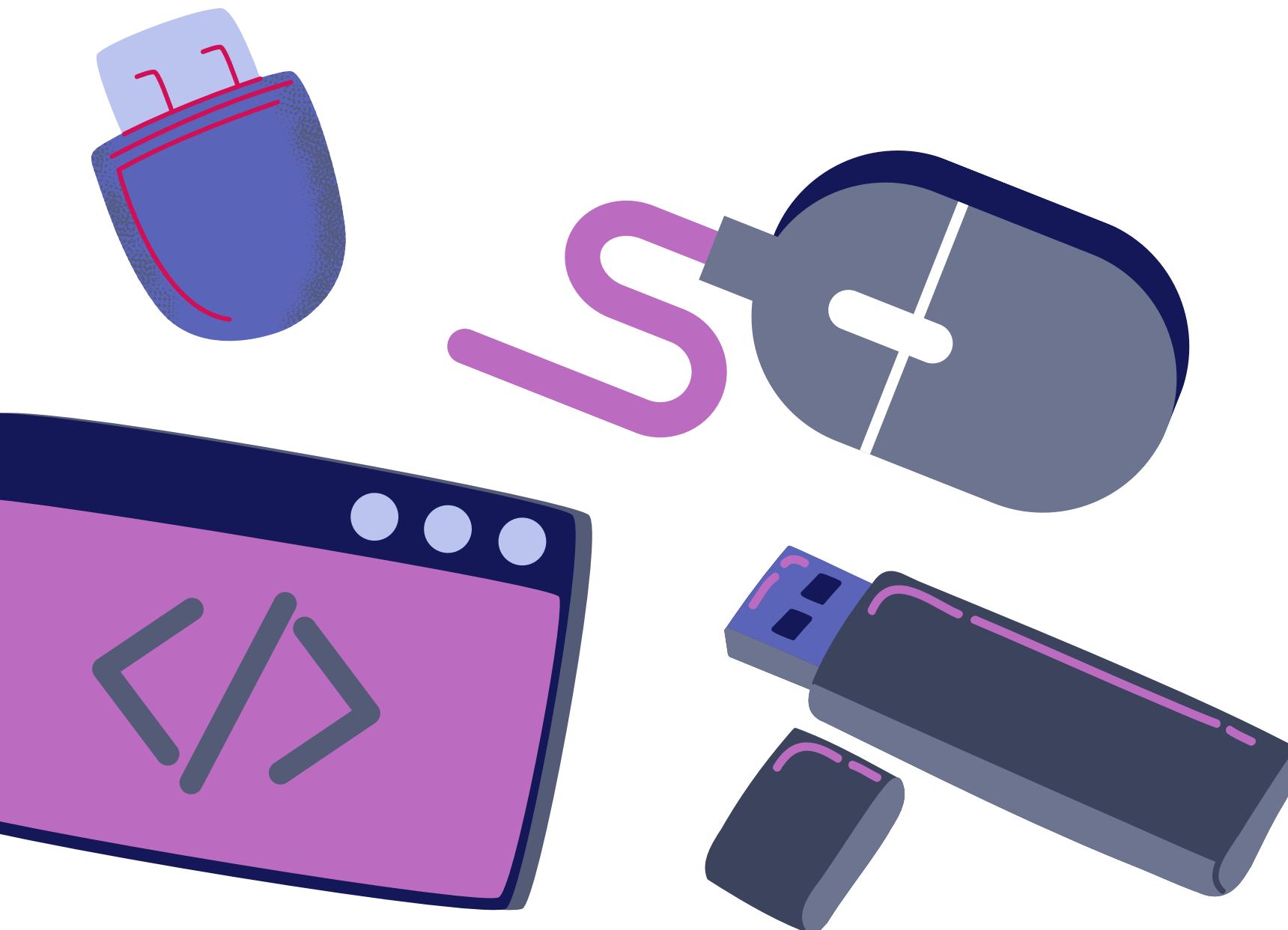
Proyecto Realizado por Jimenez Martinez Hector Yael

CONCURRENCIA

APLICADA USANDO ESP32

Basado en el la metodología de cascada.

ÍNDICE



-
- 01. Análisis y definición de requisitos.**

 - 02. Diseño del sistema y el software**

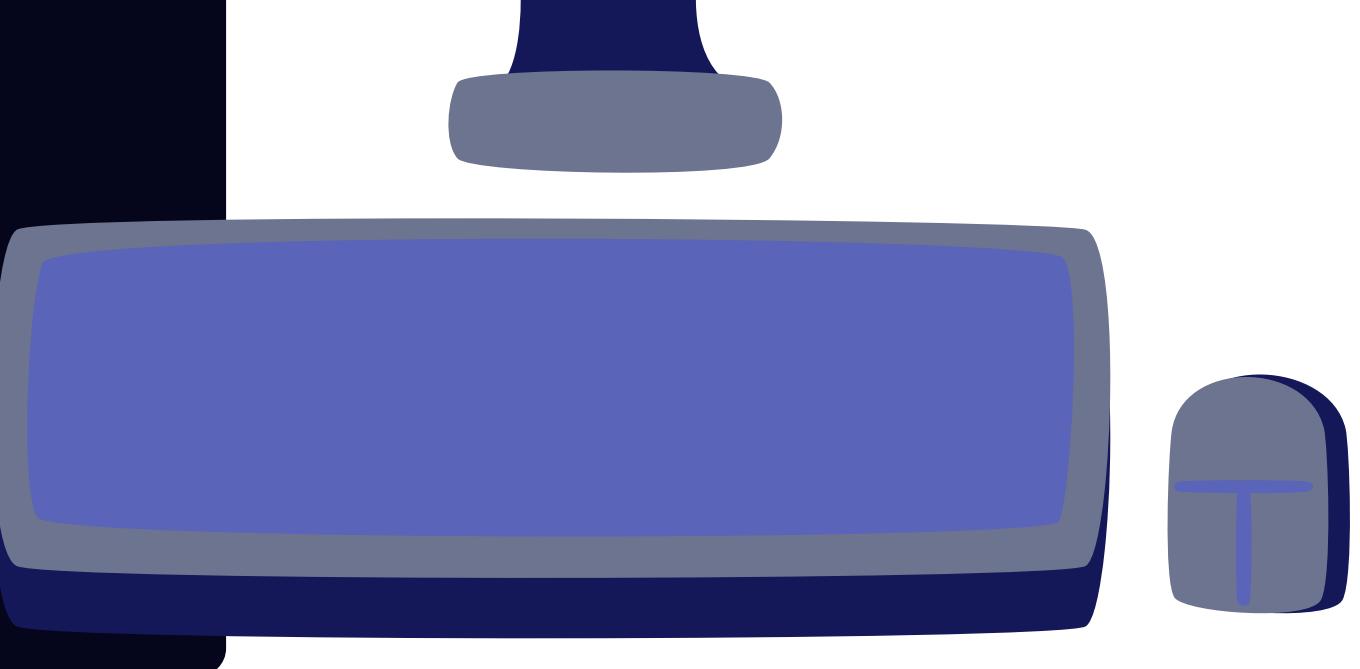
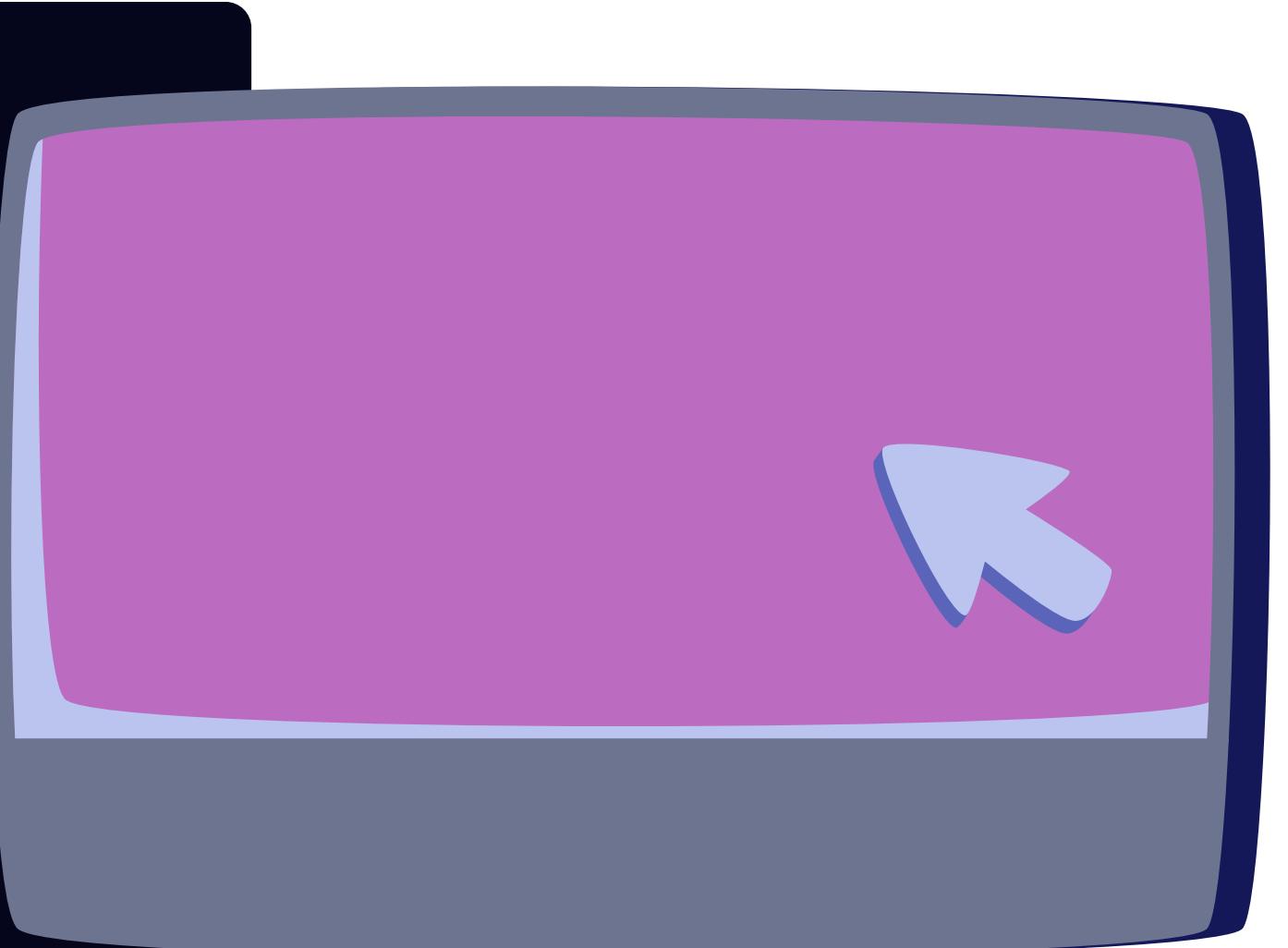
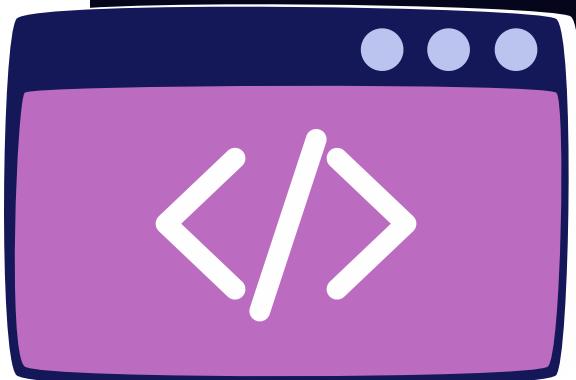
 - 03. Implementación y prueba de unidad**

 - 05. Integración y prueba de sistema**

 - 06. Operación y mantenimiento.**

Análisis de requisitos

identificación del problema:
Dificultad de conocer el estado
de la atención de las cajas en un
supermercado en tiempo real.



SERVICIOS Y METAS.

- Crear una simulación de la atención de personas en las cajas de un supermercado.
- Utilizar ESP32, concurrencia y la comunicación serial
- Lograr un sistema eficiente e intuitivo.

RESTRICCIONES:

- Limite a una cierta cantidad de cajas.
- Necesidad de una conexión serial en todo momento.

DISEÑO DEL SISTEMA Y SOFTWARE

DIAGRAMAS

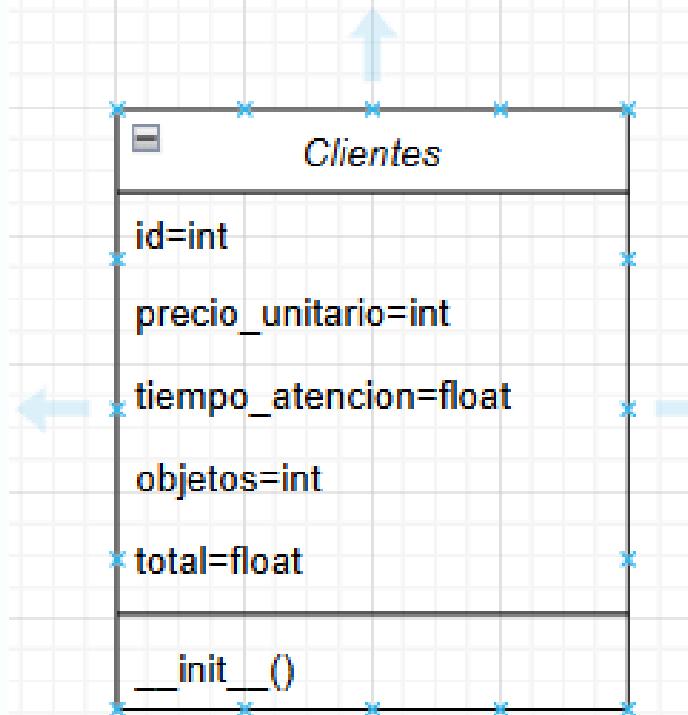
- antes de ponernos a programar, es necesario hacer algunos diagramas previos, el mas util de estos, es el uml o diagrama de clases donde se ve de una forma sencilla la organización del programa.



PROGRAMACIÓN

- el programa estará hecho en python con programación orientada a objetos ya que es la forma mas optima de modelar el tipo de programa que se requiere.

Diagrama UML GUI cajas



Clientes

- `id=int`
- `precio_unitario=int`
- `tiempo_atencion=float`
- `objetos=int`
- `total=float`

`__init__()`

Supermercado

- `ventana=tkinter`
- `total_clientes=int`
- `max_cajas=int`
- `atendidos=int`
- `ingreso_total=float`
- `registro=[]`
- `lock=threading`
- `hilos_cajas=[]`
- `estado_cajas=False`
- `sem=None`

`crear_interfaz()`

`log()`

`actualizar_resumen()`

`enviar_a_esp32()`

`iniciar_simulacion()`

`generar_clientes()`

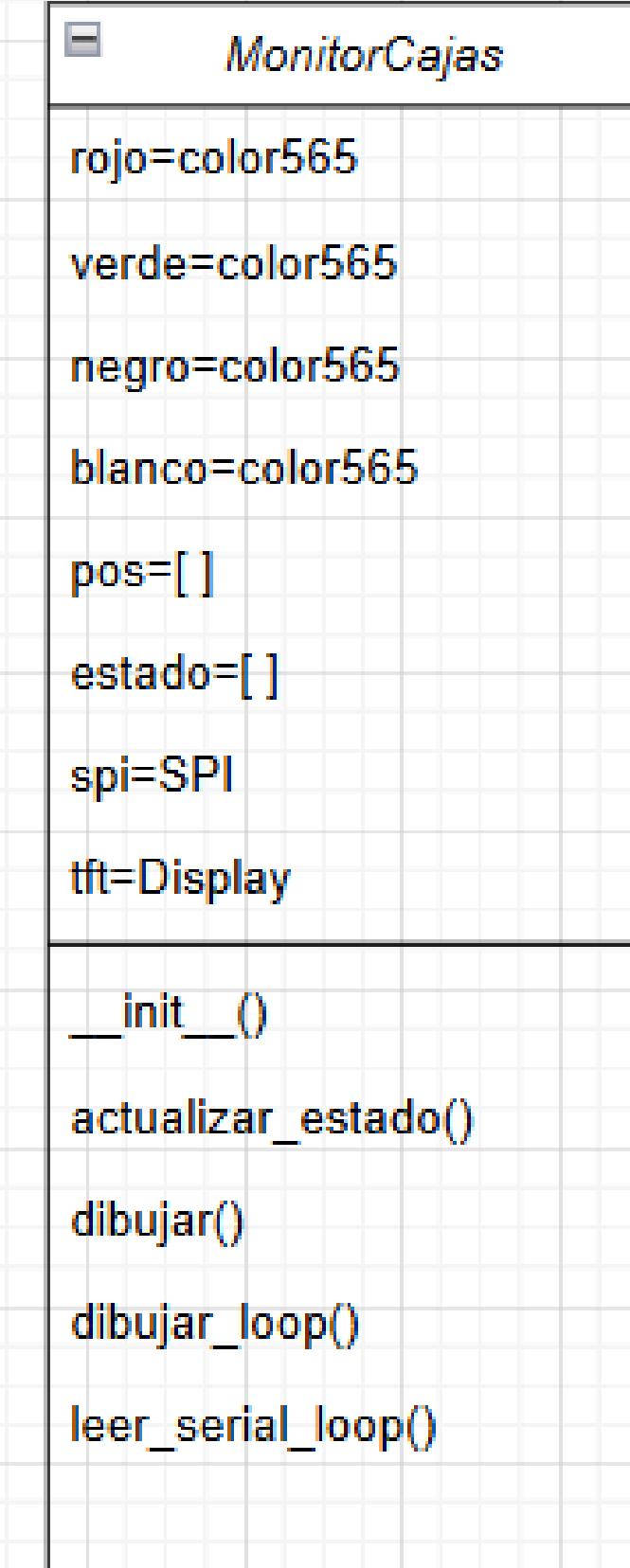
`caja()`

`detener_simulacion()`

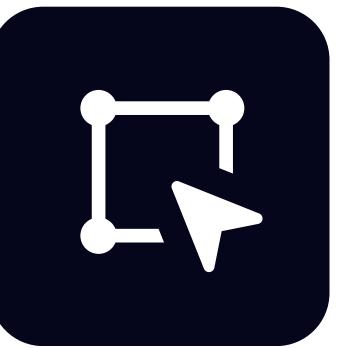
`finalizar_simulacion()`

`guardar_csv()`

Diagrama UML programa ESP32

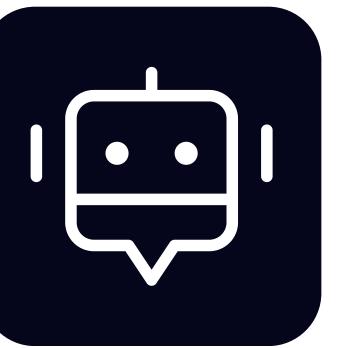


PRUEBAS DE UNIDAD



PROGRAMACIÓN

Ya que tenemos el diagrama uml de la sección pasada, requerimos pasarlo a un lenguaje de programación.



COMPROBACIÓN

Verificar que cada unidad cumpla con su especificación.

INTEGRACIÓN Y PRUEBA DE SISTEMA

La fase de integración y prueba de sistema es una de las etapas finales de la metodología de desarrollo en cascada, y tiene como propósito principal asegurar que todos los componentes desarrollados del sistema funcionen de manera conjunta y satisfagan los requerimientos definidos al inicio del proyecto.

```
puerto_esp32 = "COM7"
baudrate = 115200

try:
    ser = serial.Serial(puerto_esp32, baudrate, timeout=1)
    print(f"Conectado al ESP32 por {puerto_esp32}")
except:
    ser = None
    print("No se pudo conectar al ESP32")
ventana = tk.Tk()
supermercado = Supermercado(ventana)
ventana.mainloop()
```

Programa principal donde se especifica el puerto y la velocidad

Metodo para enviar las instrucciones al ESP32

```
def enviar_a_esp32(self):
    if ser and ser.is_open:
        try:
            cajas_str = ",".join(['1' if ocupado else '0' for ocupado in self.estado_cajas])
            ser.write(f"CAJAS:{cajas_str}\n".encode())
        except Exception as e:
            self.log(f"Error al enviar al ESP32: {e}")
```

```
# Inicializa pantalla
spi = SPI(1, baudrate=10000000, sck=Pin(18), mosi=Pin(23))
self.tft = Display(spi, dc=Pin(4), cs=Pin(5), rst=Pin(17),
| | | | width=240, height=320, rotation=0)
self.tft.clear(self.blanco)
```

Inicialización de la pantalla
y determinación de sus pines.

```
def leer_serial_loop(monitor):
    while True:
        if sys.stdin in select.select([sys.stdin], [], [], 0)[0]:
            try:
                linea = sys.stdin.readline().strip()
                if linea.startswith("CAJAS:"):
                    partes = linea[6:].split(",")
                    if len(partes) == 4:
                        estados = [bool(int(p)) for p in partes]
                        monitor.actualizar_estado(estados)
            except Exception as e:
                print("Error al leer:", e)
            time.sleep(0.05)
```

Método para leer las instrucciones
que entran al esp32

MUCHAS GRACIAS