

PROYECTO SISTEMAS DE COMUNICACIONES

RECONOCIMIENTO  
FACIAL

# OBJETIVO

---

- ★ Desarrollar un sistema que permita identificar personas mediante reconocimiento facial para controlar el acceso a un espacio.

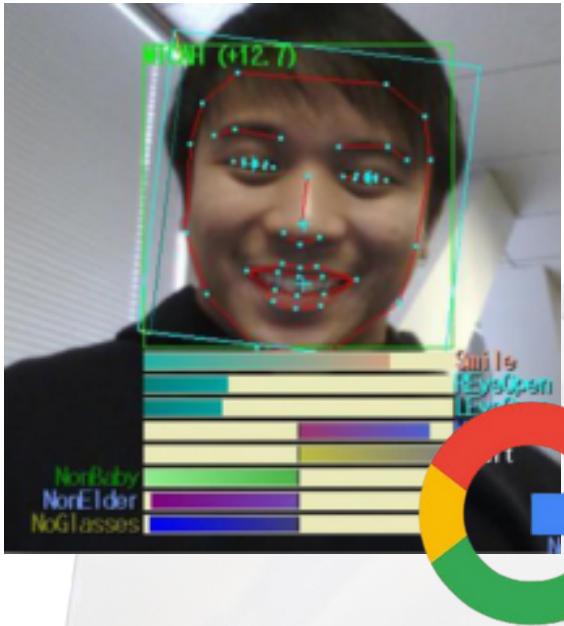
# METAS

- ★ Implementar la comunicación serial entre una computadora y un microcontrolador Arduino para ejecutar acciones físicas.
- ★ Validar el funcionamiento del sistema para apertura de puerta y encendido de luz en caso de acceso autorizado.

# INTRODUCCIÓN

Este proyecto desarrolla un sistema de control de acceso mediante reconocimiento facial, usando la cámara de un celular y procesamiento con OpenCV en Python. Integra visión por computadora, comunicación serial y Arduino para identificar usuarios y activar respuestas físicas como luces o apertura de puerta. Es una solución segura, automatizada y adaptable a distintos entornos.

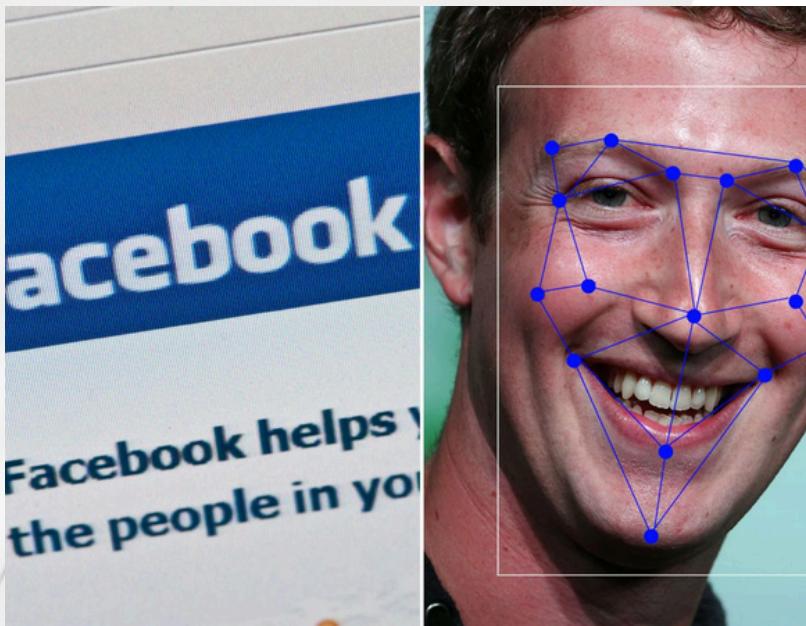
# MARCO HISTÓRICO



1890 – Alphonse Bertillon: Clasificación de personas por rasgos físicos con fines policiales.

1991 – Turk y Pentland: Uso de computadoras y técnicas como PCA y Eigenfaces para comparar rostros automáticamente.

Desde 2012: Avances con deep learning y redes neuronales → mayor precisión y velocidad en tiempo real.

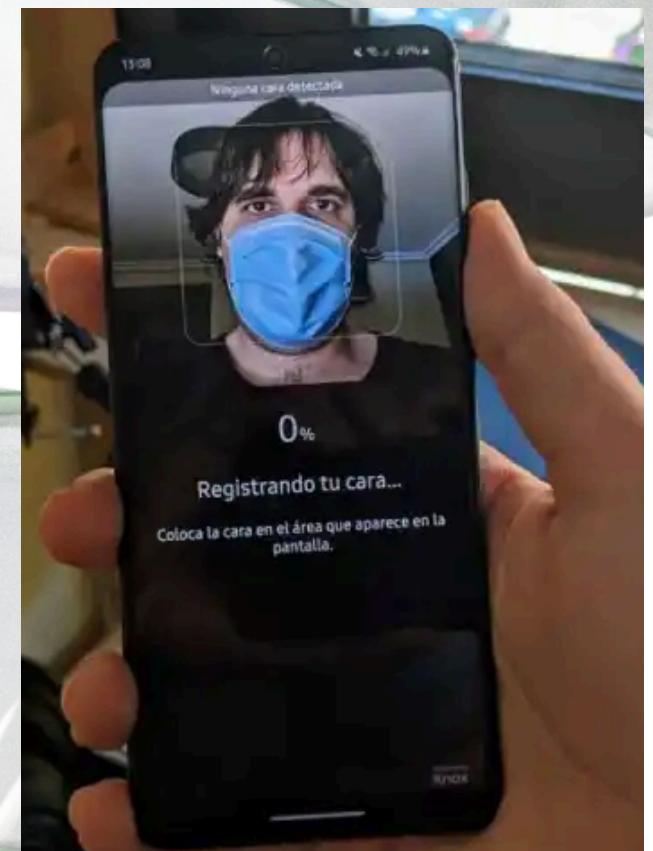


Tecnologías actuales:  
FaceNet (Google)  
DeepFace (Facebook)  
ArcFace (Microsoft) → Precisión >99%

# MARCO TEÓRICO

## Reconocimiento Facial – Principios y Funcionamiento

- 📸 Proceso automatizado que compara dos rostros para verificar identidad.
- 🔎 Detección facial: localizar el rostro en la imagen (con cascadas, redes neuronales o deep learning).
- ⚡ Normalización: ajustar tamaño, orientación y posición del rostro.
- 🧠 Extracción de características: ojos, nariz, boca, contorno → convertidos a vectores numéricos.
- ⚡ Comparación probabilística con base de datos → asigna una similitud.
- ⚙️ Etapas clave: captura, preprocesamiento, detección, extracción, comparación, decisión.



# MARCO TEÓRICO

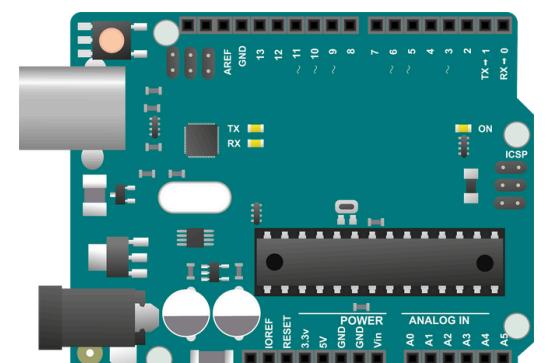
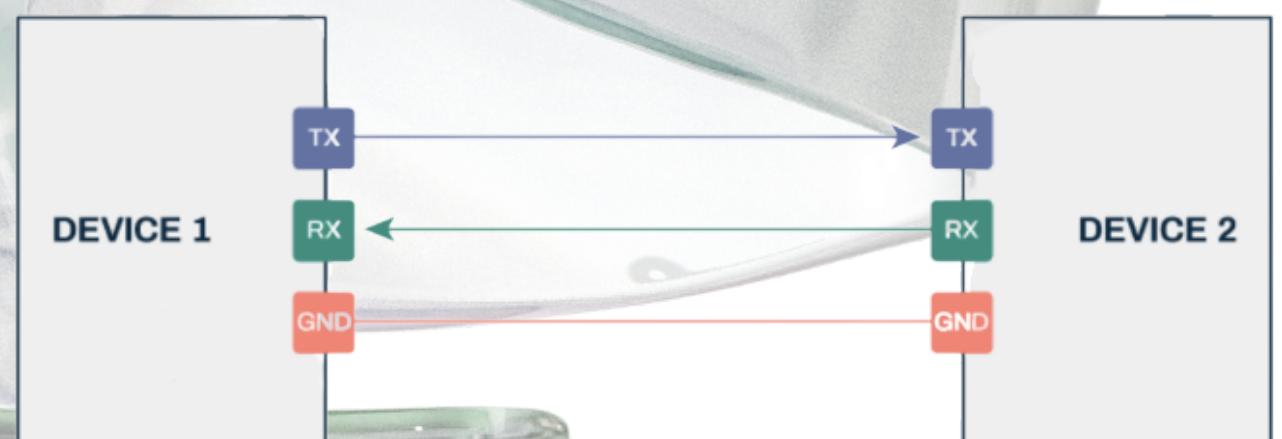
## Comunicación serial (UART):

Es un protocolo de comunicación serial que permite el envío de datos entre dos dispositivos, bit por bit, sin necesidad de una señal de reloj compartida, que es usada entre la PC y Arduino.

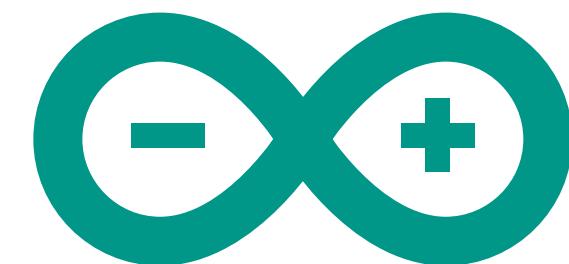
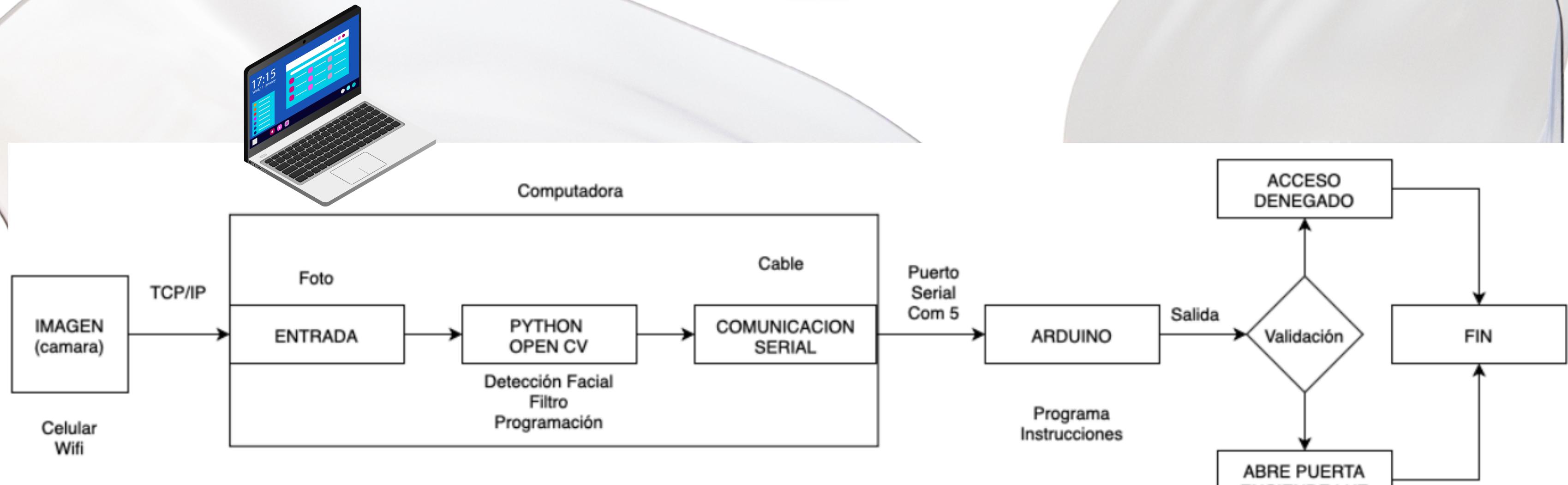
- Protocolo sencillo, ideal para sistemas embebidos.
- Paquetes: start bit + datos + paridad (opcional) + stop bits.
- Velocidad típica: 9600 baudios.

## Protocolo WiFi:

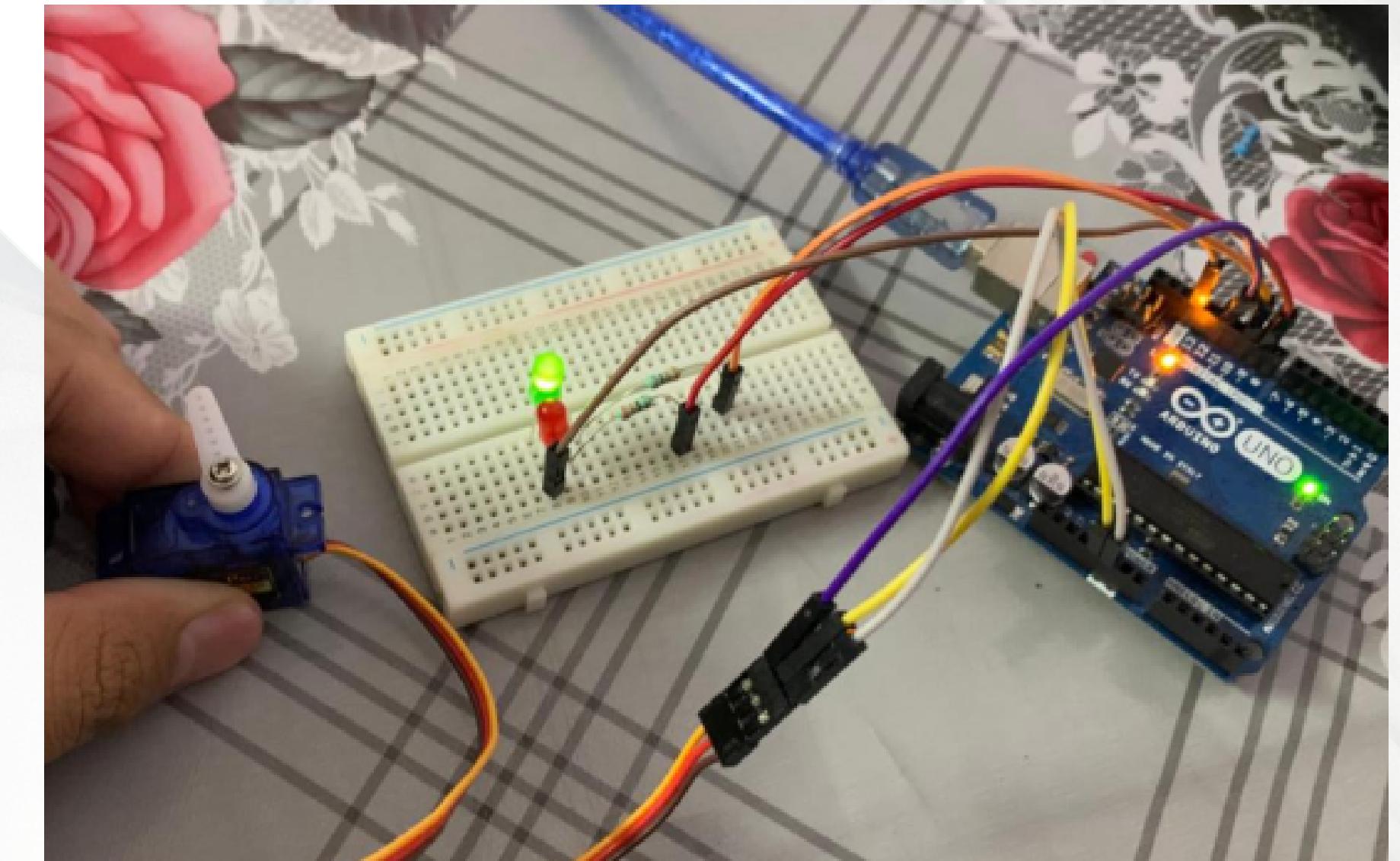
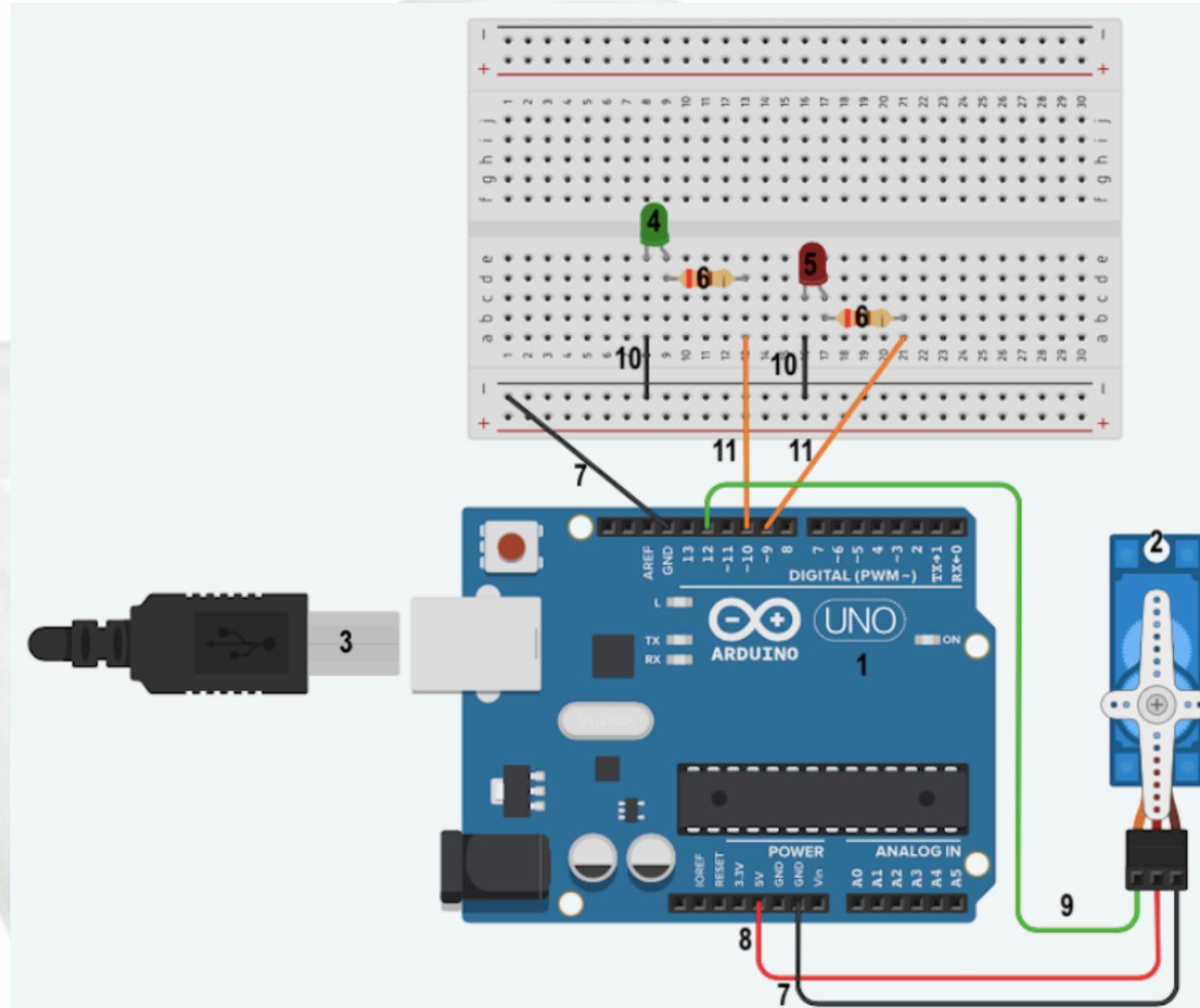
- Estándares IEEE 802.11.
- Comunicación en 2.4 GHz y 5 GHz.



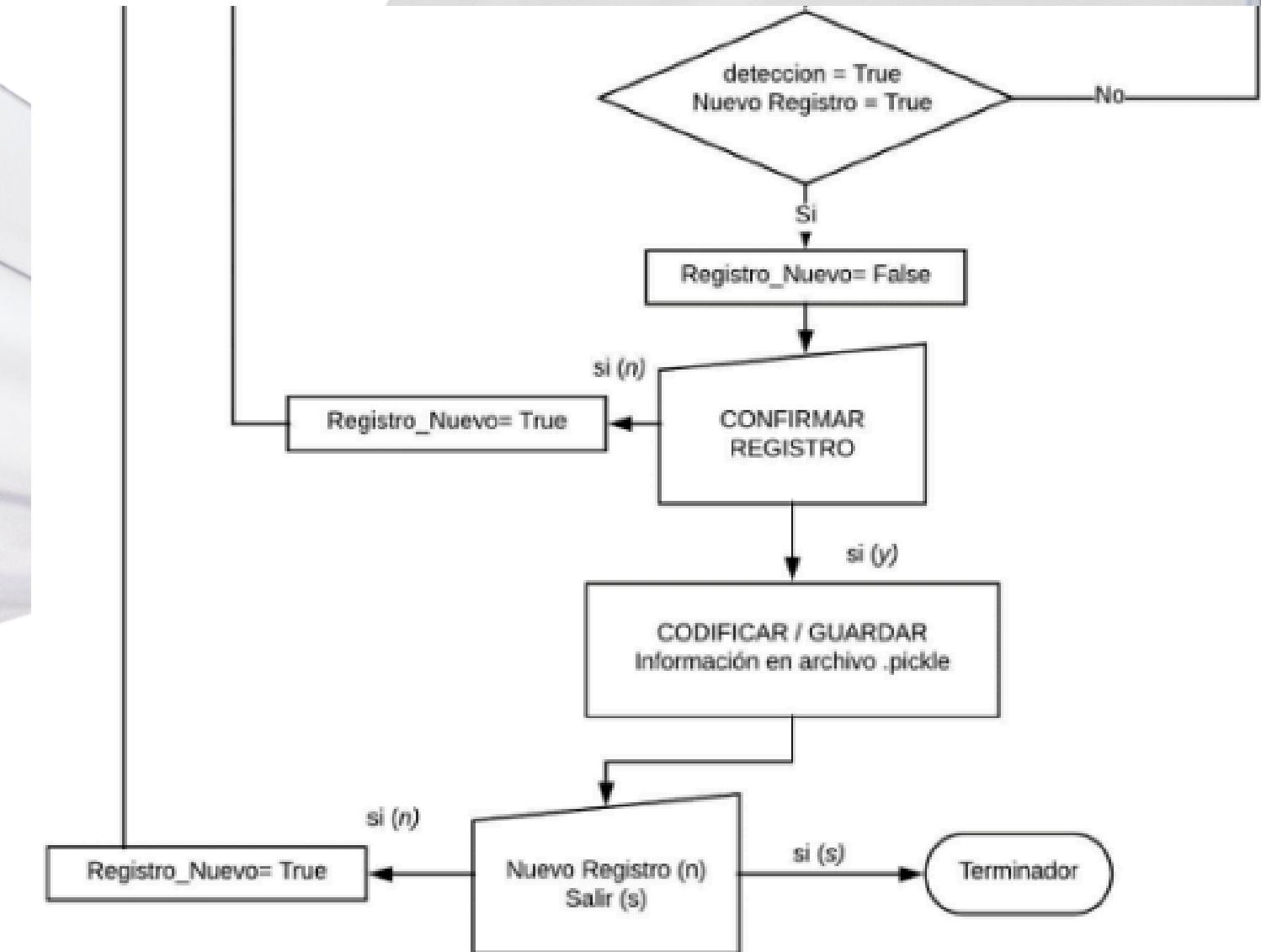
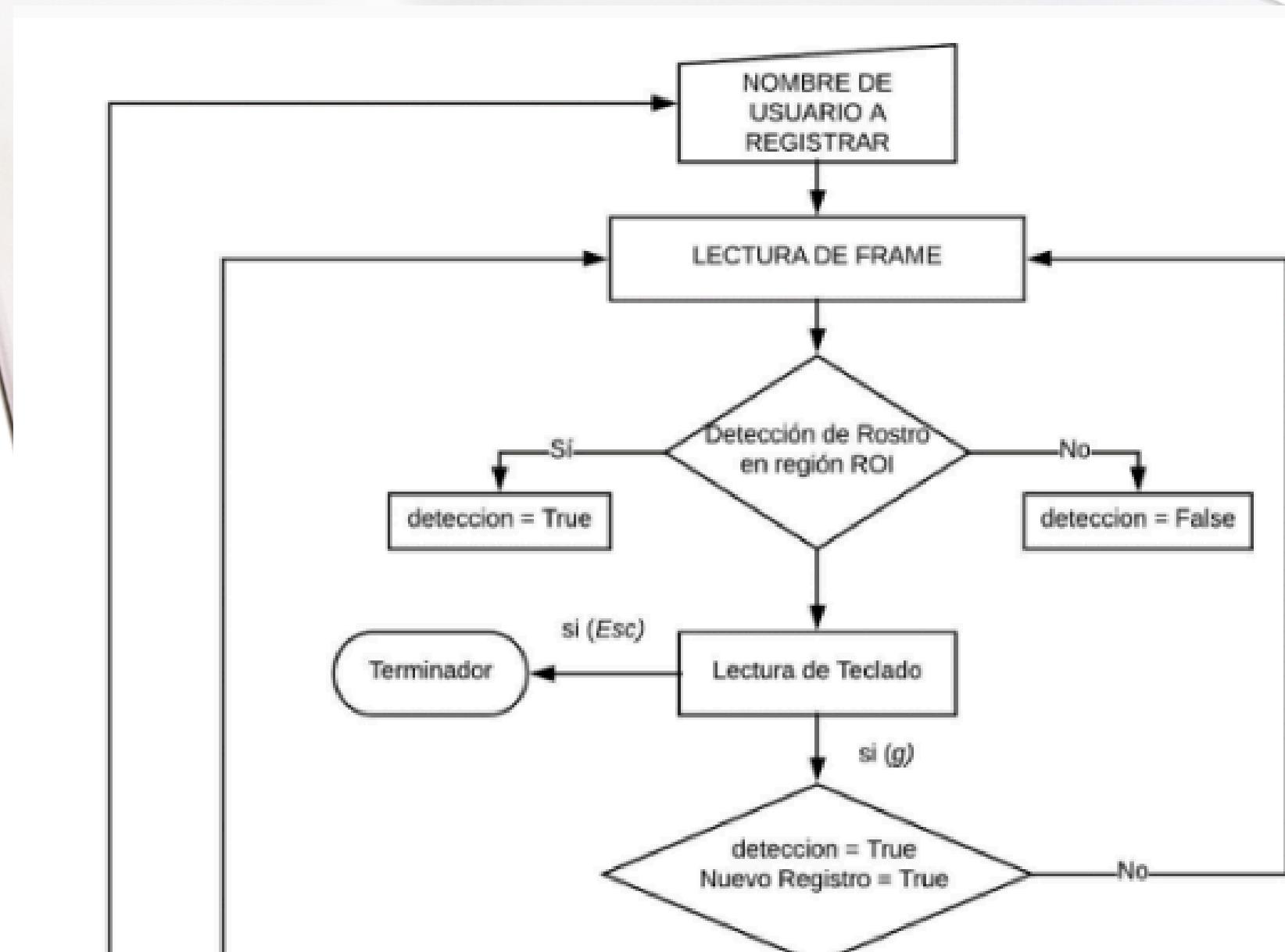
# DIAGRAMA DE BLOQUES



# DIAGRAMA ELECTRÓNICO



# DIAGRAMA DE FLUJO



# ANALISIS MATEMÁTICO

## PROTOCOLO: UART (USB-SERIAL)

VELOCIDAD: 9600 BPS

MENSAJE DE CONTROL: 1 BYTE (POR EJEMPLO: 0X01 PARA "ABRIR PUERTA")

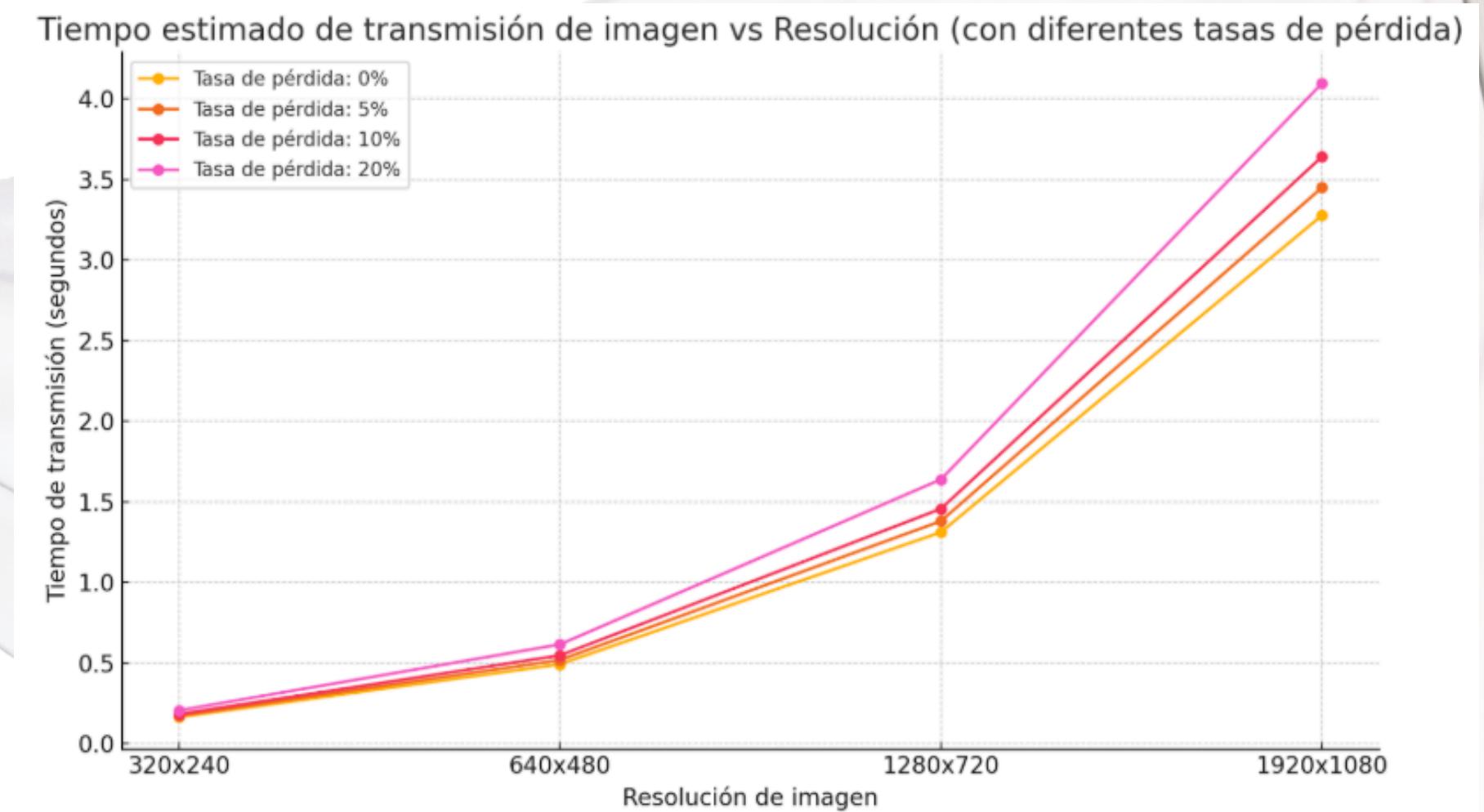
## CÁLCULO DEL TIEMPO DE TRANSMISIÓN DE COMANDO

$$T_{SERIAL} = 8 \text{ BITS}/9600 \text{ BPS} = 0.00083 \text{ S}$$

TIEMPO TOTAL ESTIMADO DE LA REACCIÓN

$$T_{TOTAL} = 0.5 + 0.2 + 0.00083 = 0.7008 \text{ S}$$

PROBABILIDAD DE PÉRDIDA DEL WIFI  $P = 0.1$

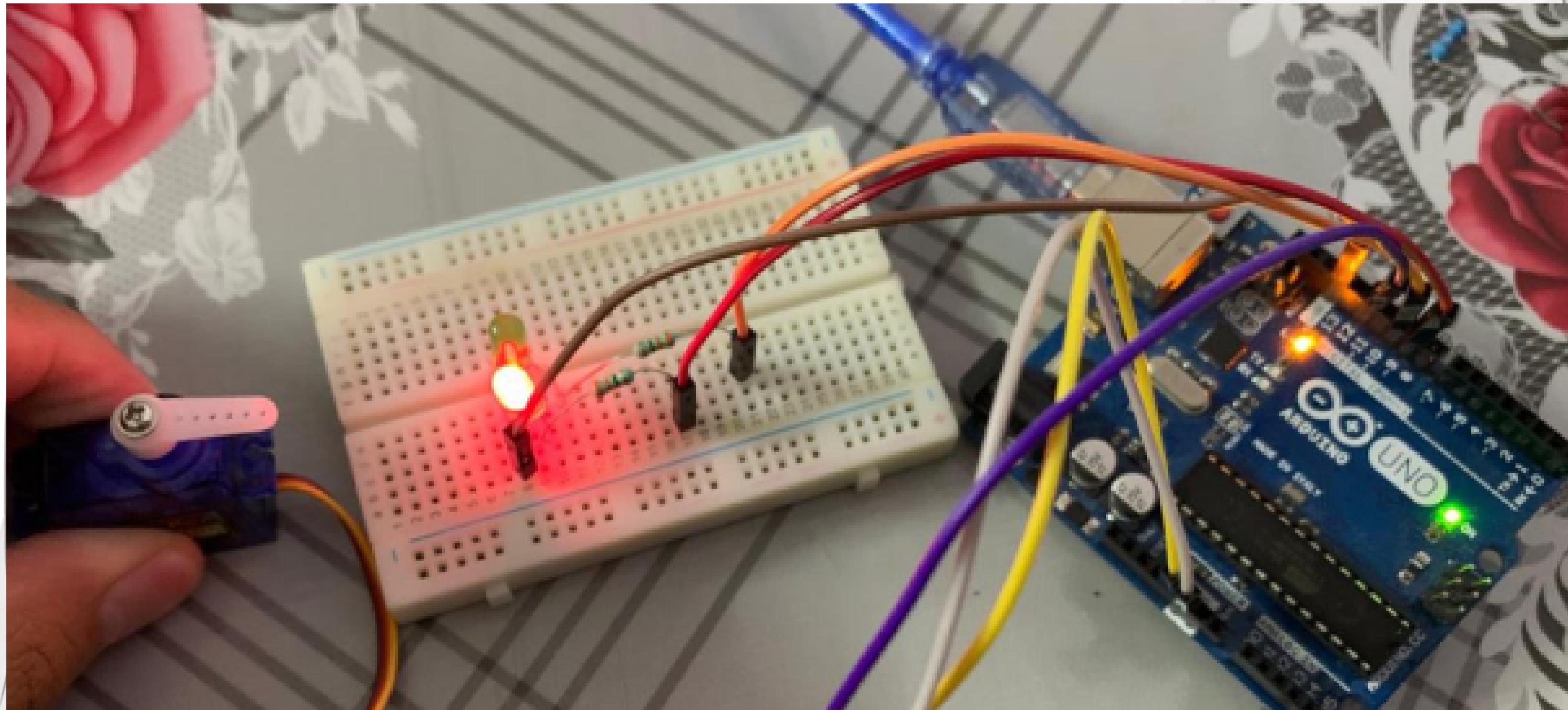


## LÚMENES

NECESITAMOS, AL MENOS 300-500 LEX O UNA LÁMPARA DE 500-100 LÚMENES SI ES A CORTA DISTANCIA.

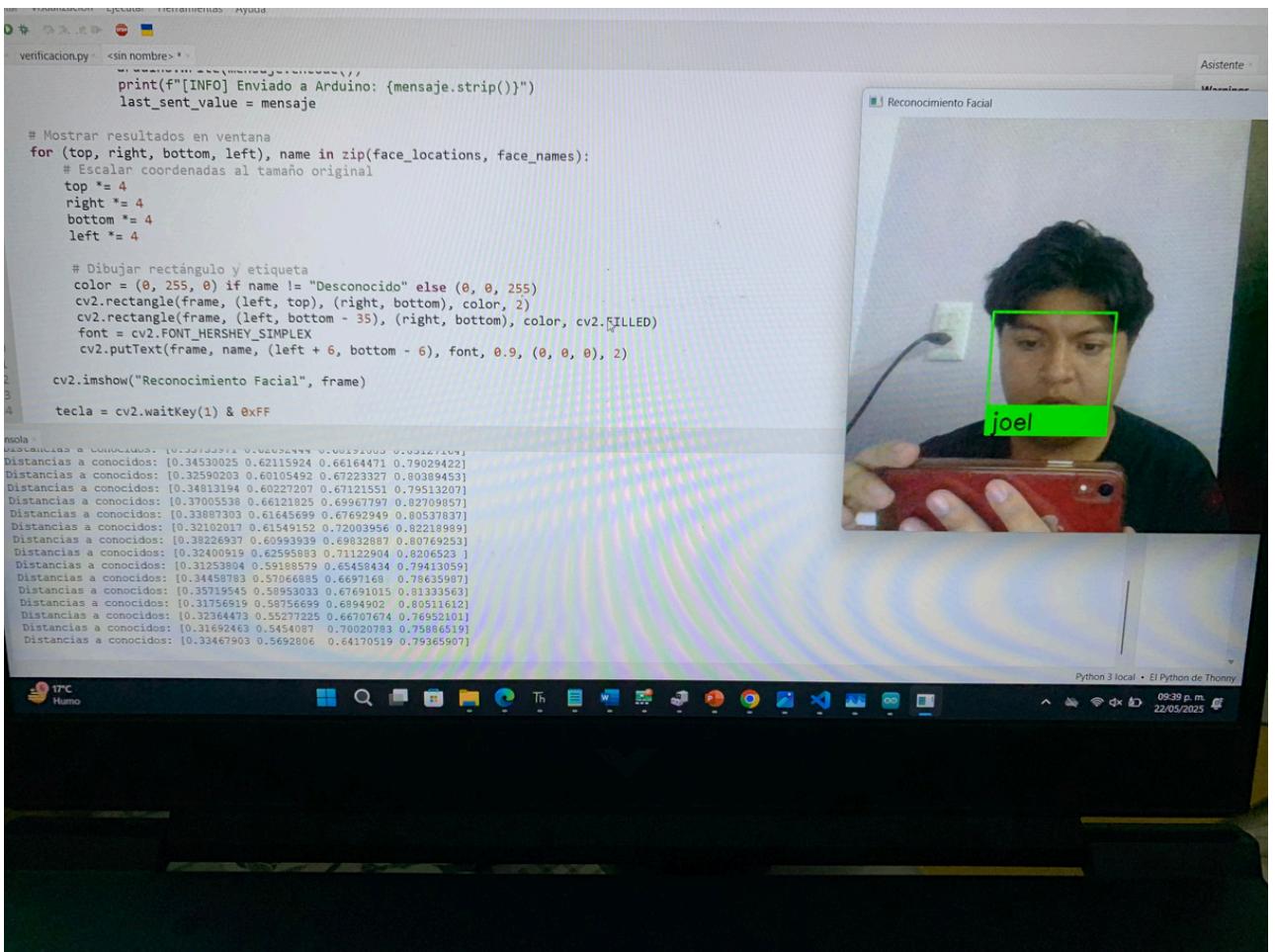
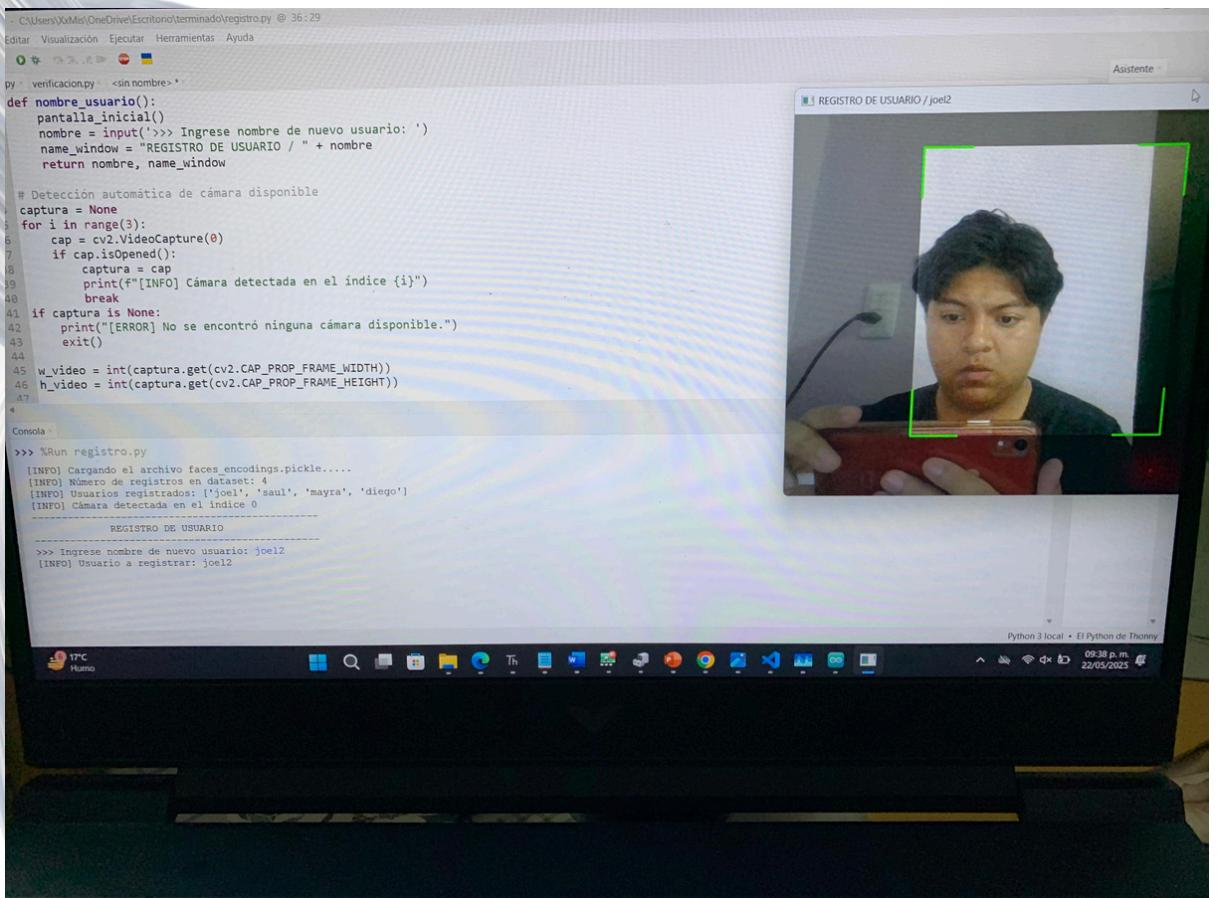
# ARMADO DEL CIRCUITO

---



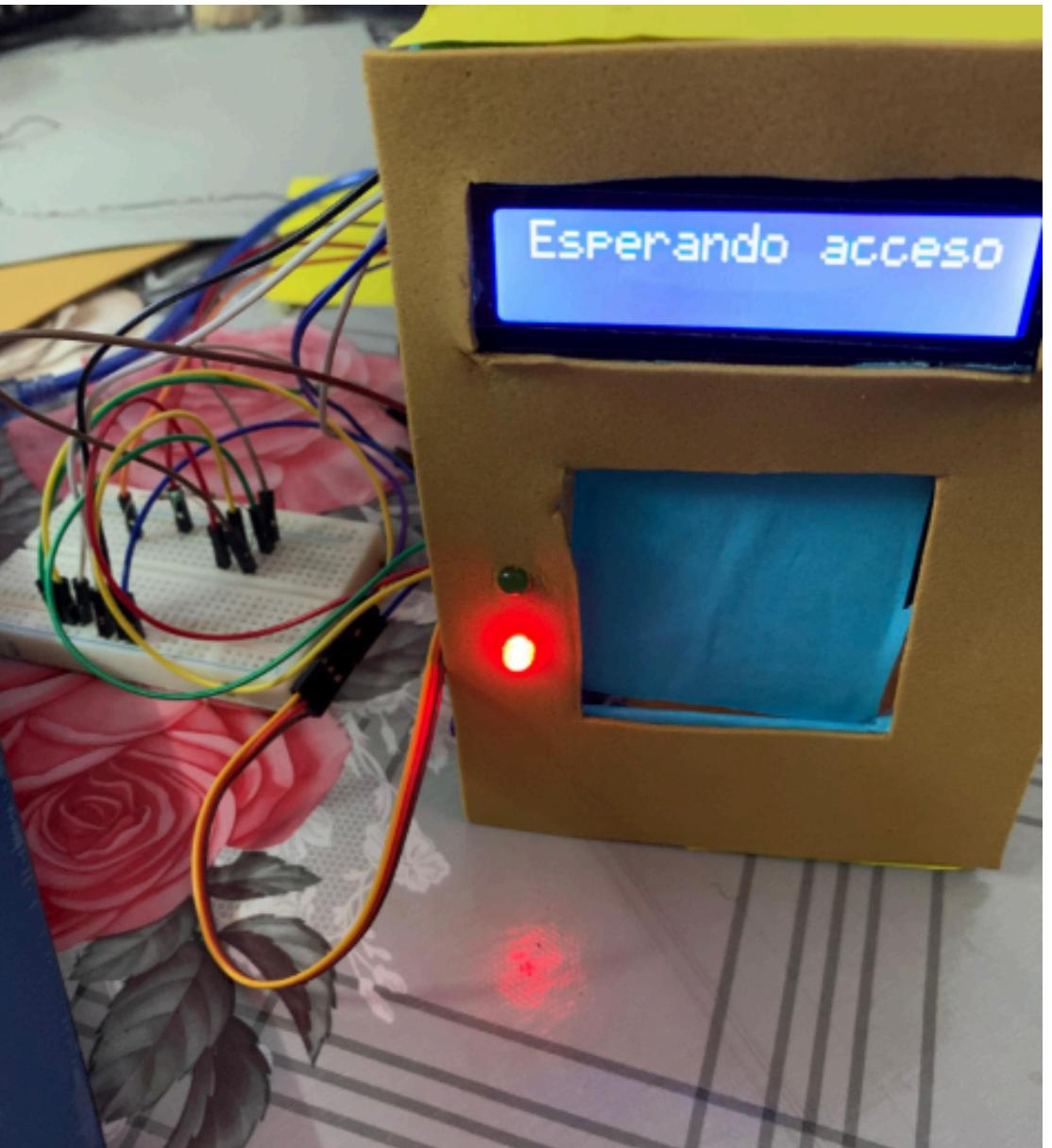
# PRUEBAS REGISTRO

# VERIFICACION



# PRUEBAS

---



SIN ROSTRO VERIFICADO



CON ROSTRO  
REGISTRADO Y VERIFICADO