

PROYECTO: SISTEMA DE ASISTENCIA PARA ADULTOS MAYORES QUE VIVEN SOLOS

Integrantes:

Justmary Anderson

Luis Boniche

Cesar Gonzalez

Jenitze Pierre

Luisa Zuluaga

INTRODUCCIÓN

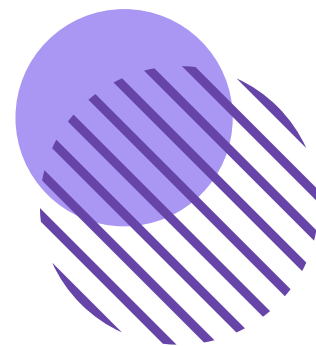
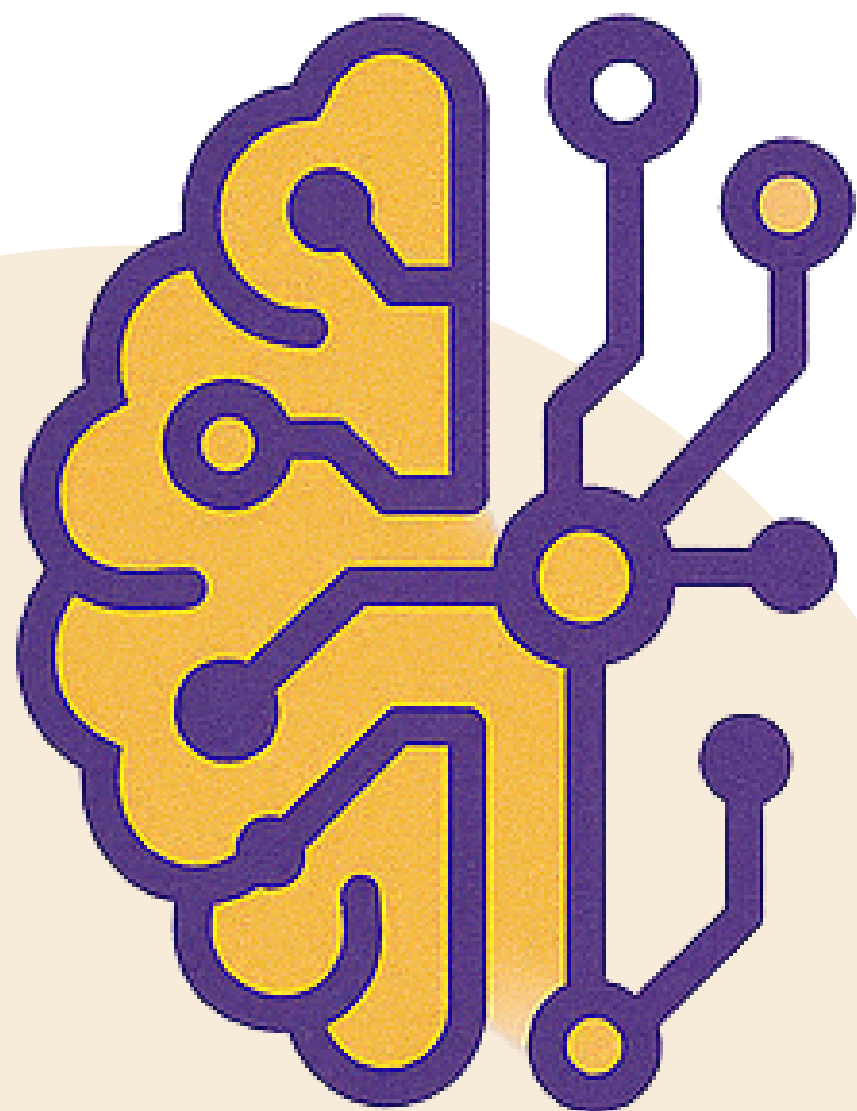
Cada vez más adultos mayores viven solos, lo que representa un riesgo frente a emergencias domésticas como caídas, accidentes o episodios de salud súbitos.



PROBLEMA

Muchas personas adultas mayores viven solas, lo que representa un alto riesgo ante situaciones de emergencia como caídas, desmayos, accidentes domésticos o problemas de salud repentinos. En estos casos, la imposibilidad de pedir ayuda a tiempo puede agravar las consecuencias. Esto evidencia la necesidad de un sistema accesible, inteligente y no invasivo que combine monitoreo ambiental con opciones inclusivas de comunicación para garantizar una respuesta rápida ante emergencias










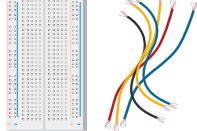
OBJETIVOS

Diseñar e implementar un sistema de asistencia para adultos mayores que permita el monitoreo del entorno y la detección de emergencias mediante sensores y lenguaje de señas, con alertas automáticas.

ESPECIFICOS

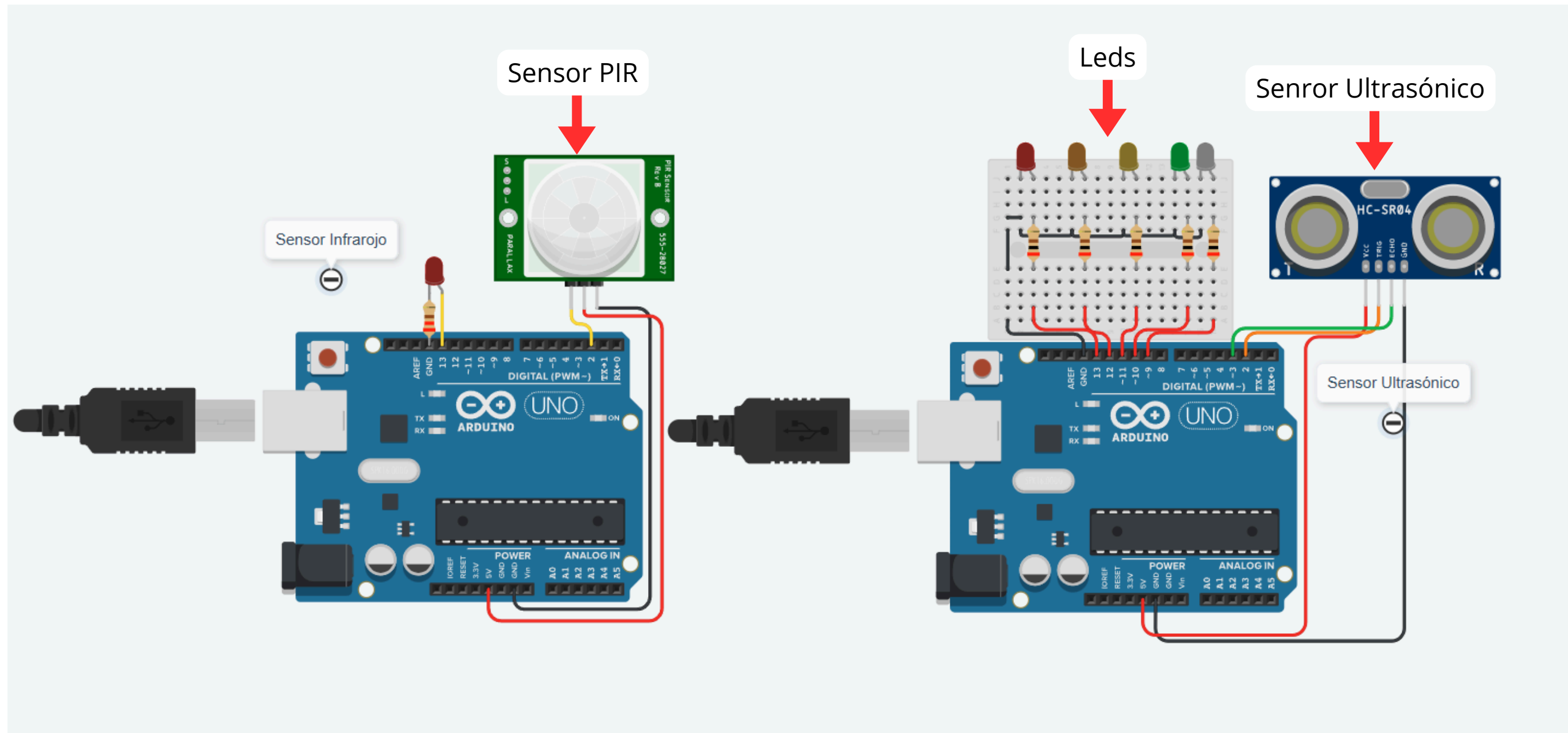
- Detectar señales de emergencia mediante lenguaje de señas usando visión artificial.
- Monitorear la actividad del adulto en distintas áreas del hogar con sensores conectados a Arduino.
- Analizar los datos recibidos para identificar comportamientos anómalos.
- Notificar automáticamente a los contactos de emergencia.

COMPONENTES

Componente	Cantidad	Descripción	Ilustración
Arduino UNO	1	Microcontrolador que recibe datos de sensores	
Sensor PIR	2-3	Detecta movimiento en habitaciones (ej. baño, sala)	
Sensor ultrasónico HC-SR04	1	Alternativa para detectar presencia prolongada o ausencia	
Cámara web	1	Puede ser integrada o externa (captura de señas)	
PC con Python instalado	1	Procesa imágenes de señas y recibe datos de sensores	
Protoboard y cables	varios	Para el montaje del circuito	



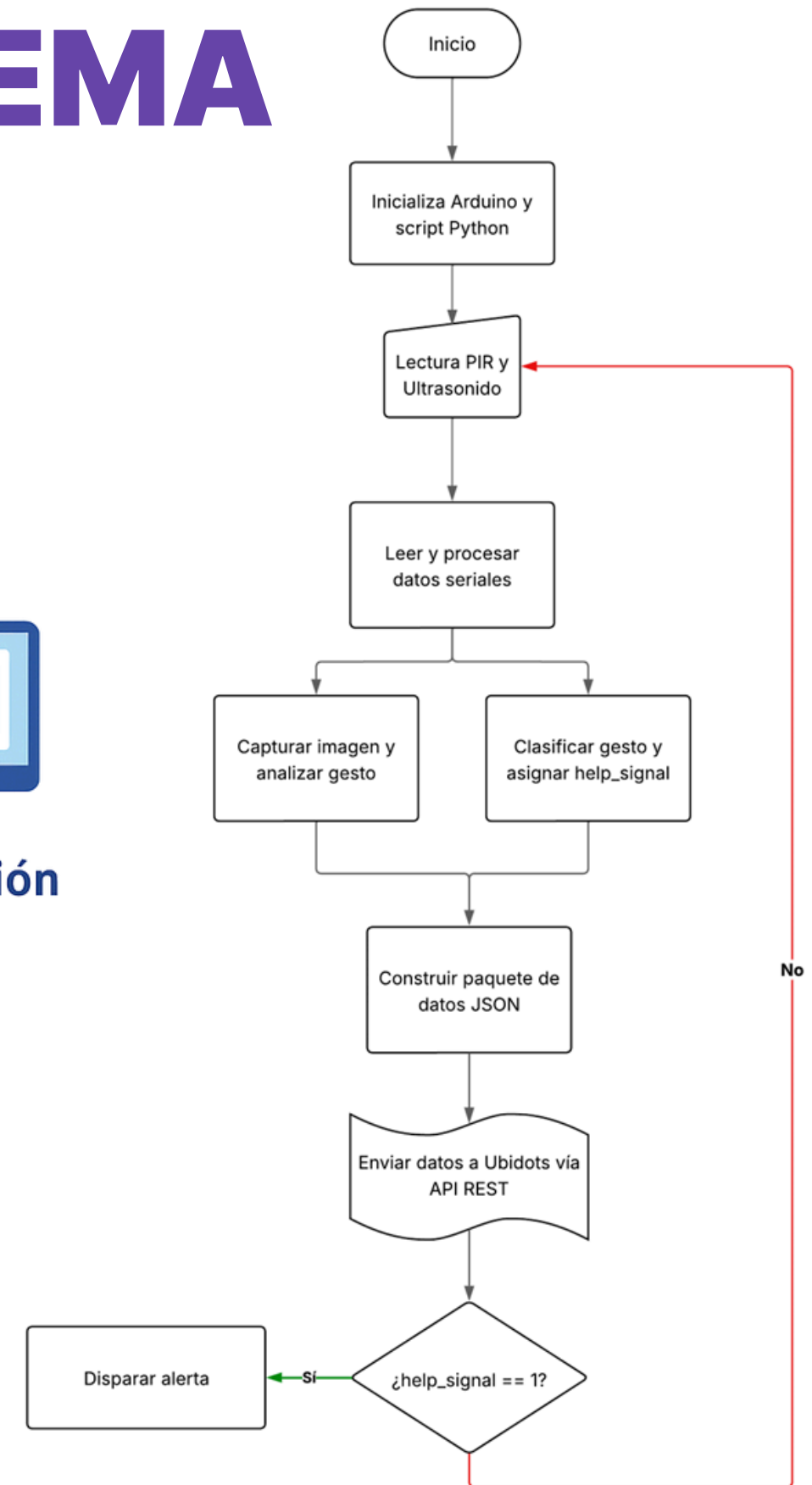
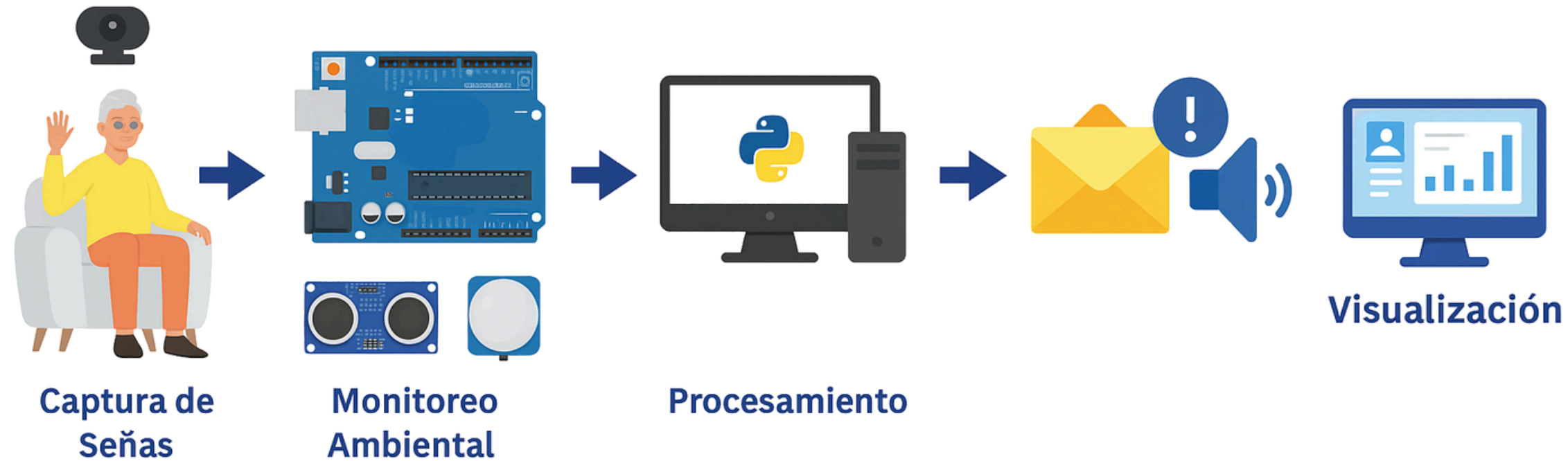
PROTOTIPO EN TINKERCAD



INTELECTO

FUNCIONAMIENTO DEL SISTEMA

- DIAGRAMA DE FLUJO



TOPOLOGÍA DE RED

- La topología del sistema implementado responde a una arquitectura de tipo “Dispositivo físico – Computador intermedio – Plataforma en la nube”, siguiendo un modelo secuencial simple, robusto y eficiente.

1

1. Dispositivo IoT: Arduino Uno

Arduino es el nodo terminal (edge device) → capta datos de sensores.

2

2. Computadora local (PC)

PC con Python actúa como un gateway local que:

- Recibe datos por serial.
- Procesa localmente (IA).
- Envía datos a la nube.

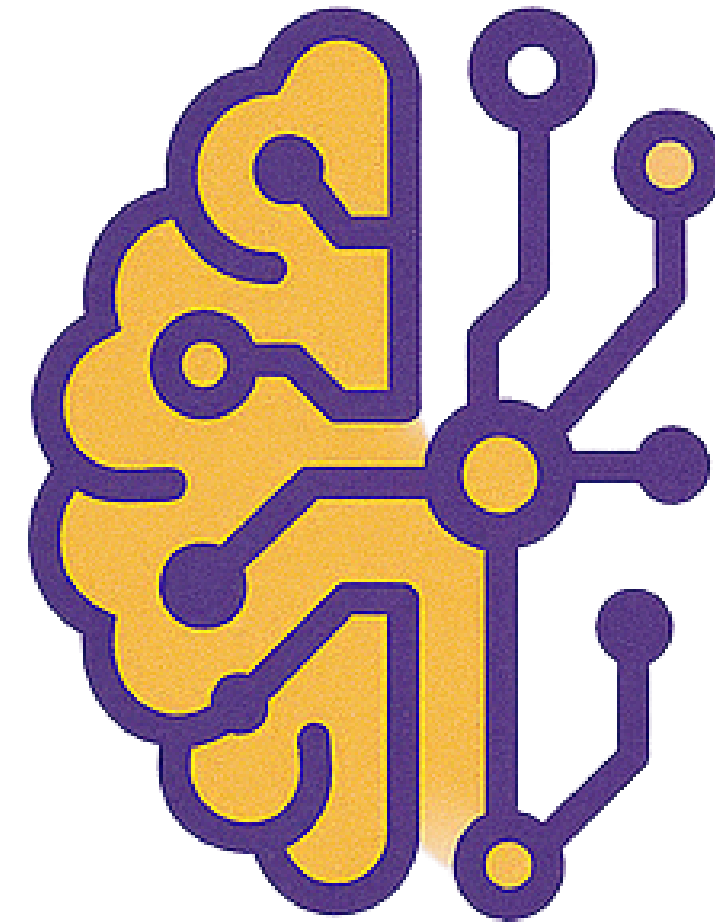
Ubidots funciona como el servidor central en la nube

3

3. UBIDOTS: NUBE

La conexión a Ubidots se realiza de dos maneras:

- Datos de sensores: se envían por comunicación serial a un script en Python y, desde allí, se publican en Ubidots utilizando el token correspondiente.
- Datos de reconocimiento de gestos: al detectar un gesto de peligro se envía el valor 1 (y 0 en caso contrario), y para el gesto de "todo OK" se procede exactamente igual.



Conexion al Ubidots

```
UBIDOTS_TOKEN=BBUS-6EPi0ilCWjHDuwZY0bUPYAojGFdwN6  
UBIDOTS_DEVICE_LABEL=Arduino_01
```


Indicator widget



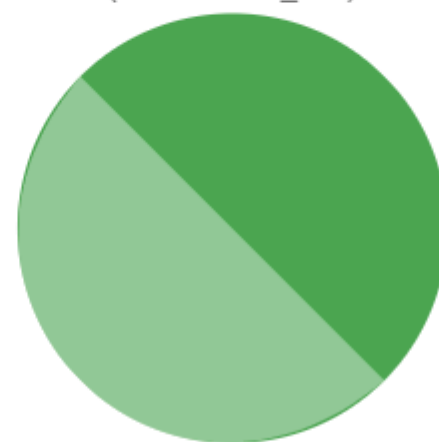
DISTANCIA

Last value



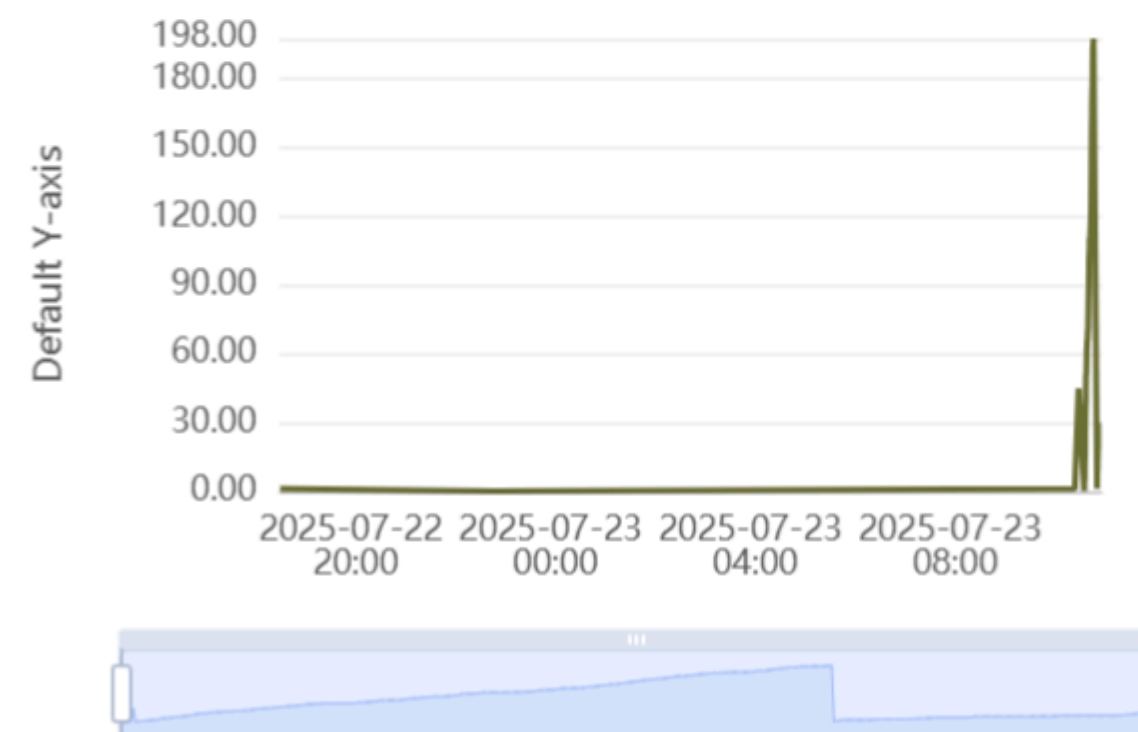
Indicator widget

peligro-total
(intelecto_01)



On

Line chart



DASHBOARD EN UBIDOTS

CREDITOS DE PARTICIPCION



- César González – Encargado del diseño y montaje del circuito físico con Arduino
- Luis Boniche – Responsable del entrenamiento del modelo de Inteligencia Artificial.
- Justmary Anderson – Se encargó de la configuración de la plataforma Ubidots Participó también en la documentación técnica, incluyendo la elaboración de diagramas y redacción de secciones clave del informe
- Luisa Zuluaga y Jenitzé Pierre – Responsable de la investigación teórica y soporte documental. Recolectó información técnica sobre sensores, arquitectura IoT y plataformas compatibles, y colaboró en la validación conceptual del sistema. Redactó partes del marco teórico y participó en la revisión del informe final.



DEMOSTRACIÓN DEL SISTEMA

LIVE 

REFERENCIAS

Indriani, I., Harris, M., & Suryaperdana, A. (2020). Applying hand gesture recognition for user guide application using MediaPipe. Atlantis Press. <https://doi.org/10.2991/icoim-19.2019.12>

Lugaresi, C., Tang, J., Nash, H., McClanahan, C., Uboweja, E., Hays, M., ... & Grundmann, M. (2019). MediaPipe: A framework for building perception pipelines. arXiv. <https://arxiv.org/abs/1906.08172>

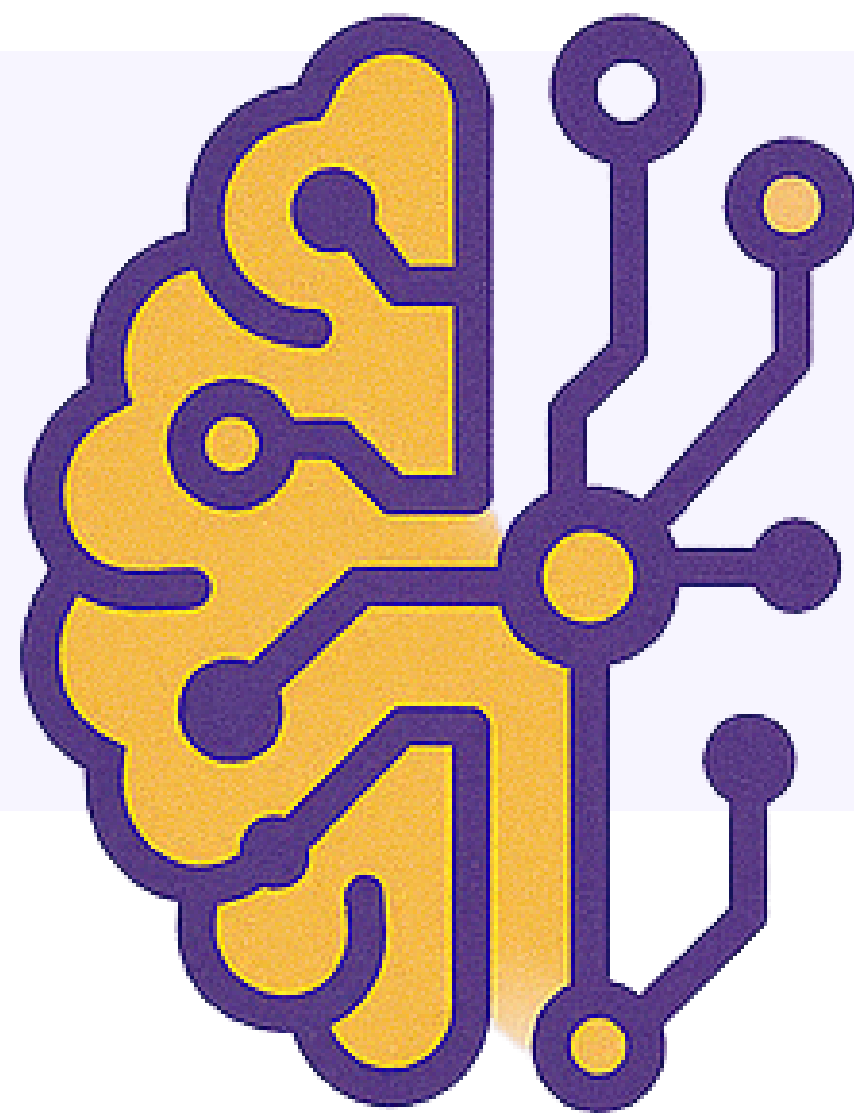
Kumar, R., Bajpai, A., & Sinha, A. (2023). MediaPipe and CNNs for real-time ASL gesture recognition. arXiv. <https://arxiv.org/abs/2305.05296>
Sung, G., Sokal, K., Uboweja, E., Bazarevsky, V., Baccash, J., Bazavan, E. G., ... & Grundmann, M. (2021). On-device real-time hand gesture recognition. arXiv. <https://arxiv.org/abs/2111.00038>

Warchocki, J., Vlasenko, M., & Eisma, Y. B. (2023). GRLib: An open-source hand gesture detection and recognition Python library. arXiv. <https://arxiv.org/abs/2310.14919>

Queen, C. (2022). A study on real-time hand gesture recognition technology by MediaPipe. Journal of System and Management Sciences, 12(2), 462–476. <https://ejournals.aasmr.org/jsms/Vol12/JSMS%20April%202022/Vol.12No.02.25.pdf>

TechExplorations. (2023). Arduino sensors tutorial: detect motion with PIR sensor. <https://techexplorations.com/guides/arduino/sensors/pir>

Instructables. (2023). Arduino Uno and PIR sensor: building a smart motion detection. <https://www.instructables.com/Arduino-Uno-and-PIR-Sensor-Building-a-Smart-Motion/>



GRACIAS

