



OmniSens

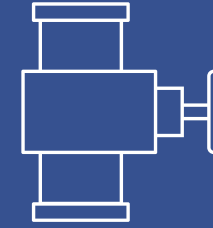
SISTEMA DE MONITOREO Y CONTROL AMBIENTAL CON IOT

PROYECTO FINAL – TRAYECTO FORMATIVO IOT – TST
2024

Equipo de trabajo



Fernando Gimenez Coria (@FerCbr)



Diego Ares (@diegote7)



Paola Pantoja (@PaolaaPantoja)



Romina Vanesa Huk (@RoHu17)



Macarena Aylèn Carballo (@MacarenaAC)



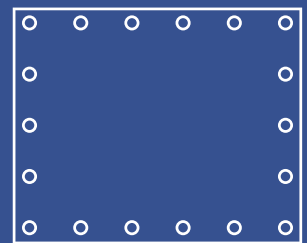
Leandro Roldan (@pleroldan)



Emma Vilma Gutierrez (@emygut)



Raul Jara (@r-j28)



Juan Diego Gonzalez Antoniazzi (@JDGA1997)



INTRODUCCION

- ¿QUÉ ES OMNISENS?

UN SISTEMA IOT DESARROLLADO PARA MONITOREAR VARIABLES AMBIENTALES EN ENTORNOS INDUSTRIALES O SENSIBLES COMO LABORATORIOS, GALPONES, VIVEROS O CÁMARAS DE CULTIVO.

Objetivo principal:

- Detectar condiciones fuera de rango y actuar automáticamente para restablecerlas, además de notificar a los usuarios a través de una interfaz.

Motivación y Alcance

Motivaciones del proyecto

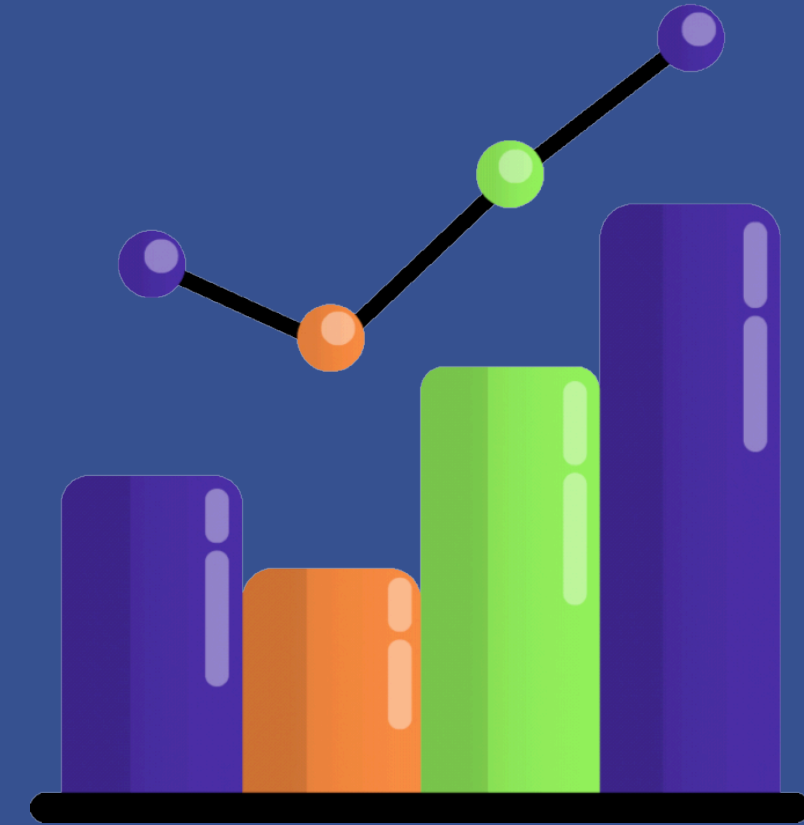


- Aumentar la seguridad ambiental en espacios cerrados.
- Automatizar procesos de ventilación, iluminación y alerta.
- Registrar y visualizar información histórica para auditorías o mantenimientos.

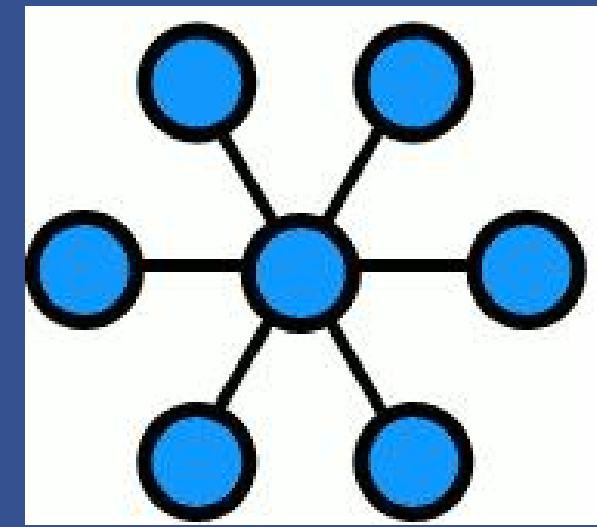
Alcance



- Entorno cerrado (pequeña industria/laboratorio).
- Capacidad de expansión modular para nuevos sensores o zonas.



Arquitectura General del Sistema



- **Sensores y Controladores Edge**

Los sensores miden variables críticas y envían los datos a controladores edge, que pueden accionar dispositivos como extractores de aire, luces, alarmas sonoras y balizas luminosas.

- **Comunicación LoRa y Nodo Concentrador**

Los controladores edge se comunican mediante un enlace LoRa hacia un nodo concentrador, formando una red de tipo estrella. Este nodo envía los datos recolectados a un broker MQTT remoto para su posterior análisis y almacenamiento.

- **Servidor en la Nube y Dashboard**

Toda la información recolectada se analiza en un servidor remoto y se almacena en una base de datos. Luego, se muestra en un dashboard que permite visualizar la información procesada y enviar órdenes automáticas o comandos indicados por el usuario.

SENSORES DISTRIBUIDOS (NODOS) → NODO CONCENTRADOR → SERVIDOR REMOTO.

Componentes principales

Sensores:

- Temperatura
- humedad
- CO₂
- presión
- luminosidad.

Actuadores:

- Extractores
- Luces
- Alarmas
- Balizas

Red de comunicación:

- Lora entre nodos y el gateway

Backend & Dashboard:

- MongoDB
- Node.js
- React



Hardware Utilizado

Componente	Función
ESP32	Lectura de sensores + control local
Raspberry Pi	Gateway concentrador + servidor MQTT
Sensores DHT22	Temperatura y humedad
MQ-135	Calidad del aire
LDR	Luminosidad
Relés/SSR	Activación de actuadores

Comunicaciones



Red LoRa:



- Bajo consumo, largo alcance.
- Ideal para transmitir pequeñas cargas de datos.

MQTT (Message Queuing Telemetry Transport):



- Protocolo ligero para IoT.
- Publicador/suscriptor entre sensores, gateway y dashboard.

Backend y Base de Datos

Servidor Node.js:

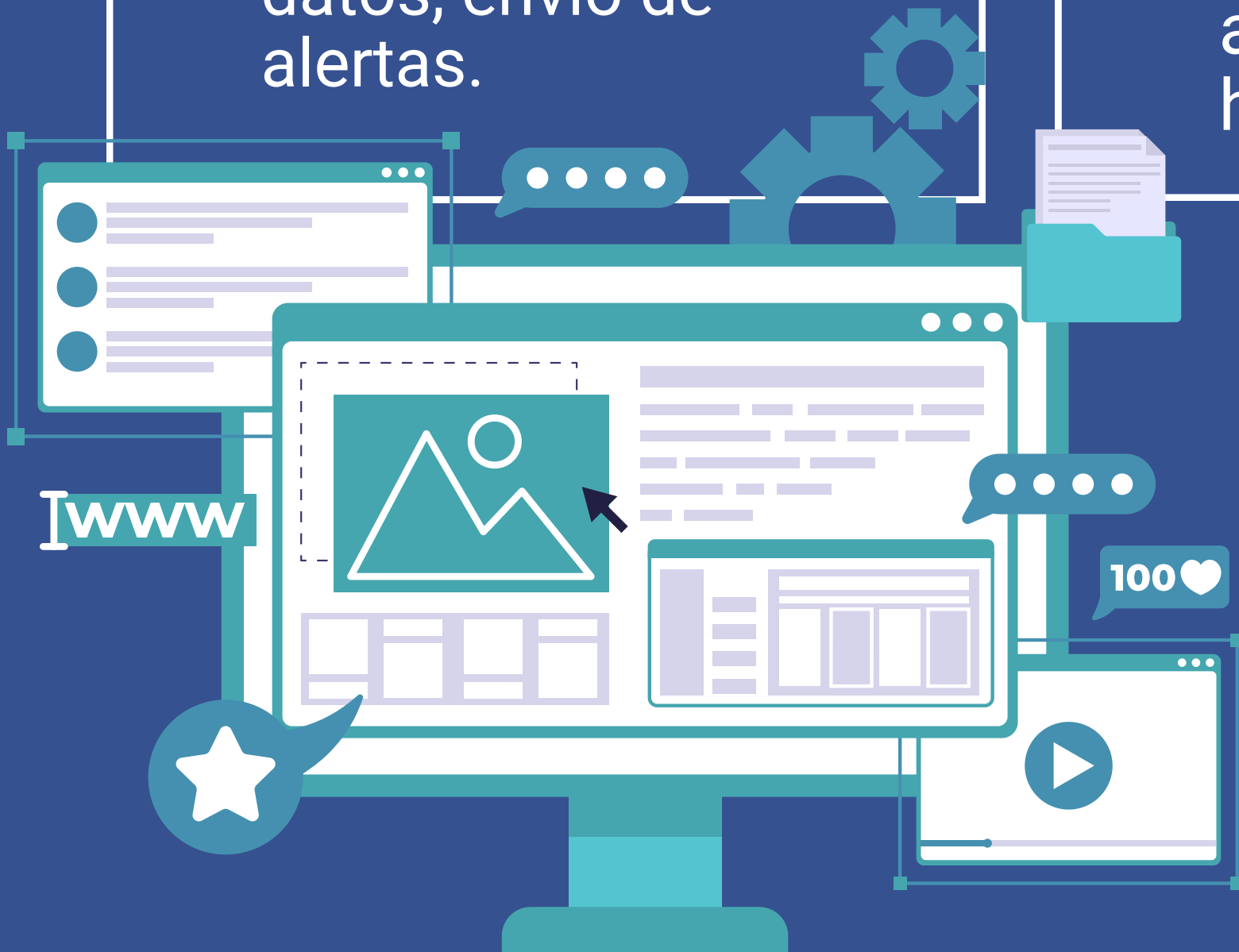
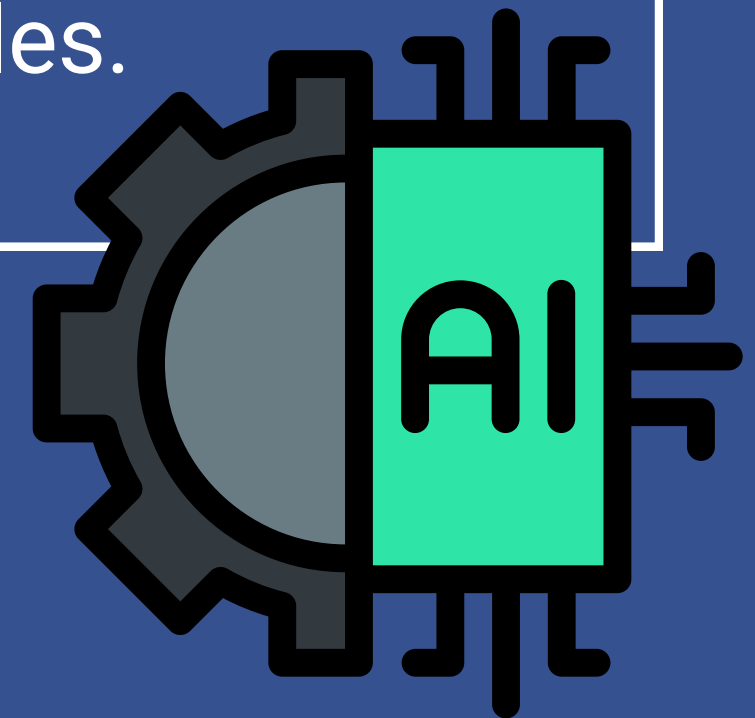
- Manejo de eventos, procesamiento de datos, envío de alertas.

MongoDB:

- Base de datos NoSQL para almacenar lecturas históricas.

APIs REST:

- Permiten acceder a los datos desde el frontend o apps móviles.



Interfaz de Usuario

Dashboard en React:

- Visualización en tiempo real de variables.
- Gráficos históricos por día, semana o mes.
- Panel de control para envío manual de comandos.

Automatización y Alertas



Acciones automáticas:

- Encendido de extractores si el CO₂ supera cierto umbral.
- Activación de alarmas si se detecta un sensor desconectado.



Notificaciones:

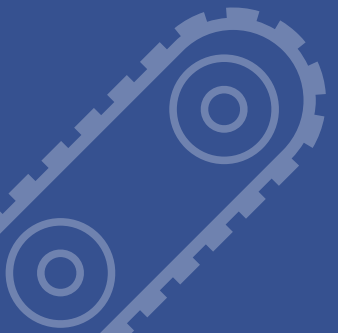
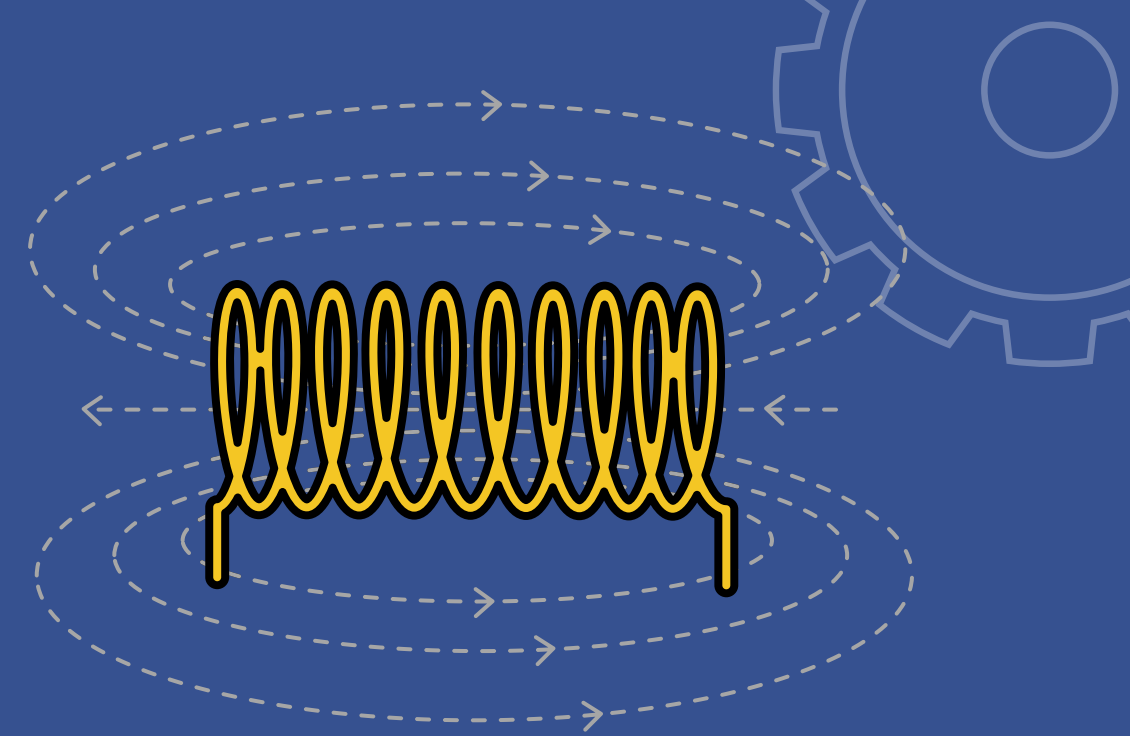
- Vía correo electrónico o mensajería instantánea.
- Logs de alertas con timestamp.



Pruebas y Validación

Pruebas de campo:

- Validación de comunicación LoRa en distintos entornos.
- Comparación de sensores con instrumentos calibrados.
- Latencia en la activación de dispositivos ante eventos.



Fases del Proyecto

Fase 1 - Iniciación del Proyecto: Establecimiento de la documentación inicial y definición de la arquitectura general.

Fase 2 - Prototipo Sensado y Control:

Fase 3 - Integración y Pruebas: Verificación de la interoperabilidad entre los diferentes módulos del sistema.

Fase 4 - Documentación y Escalado: Completar la documentación técnica y planificar siguientes iteraciones.

Posibilidades de Escalado

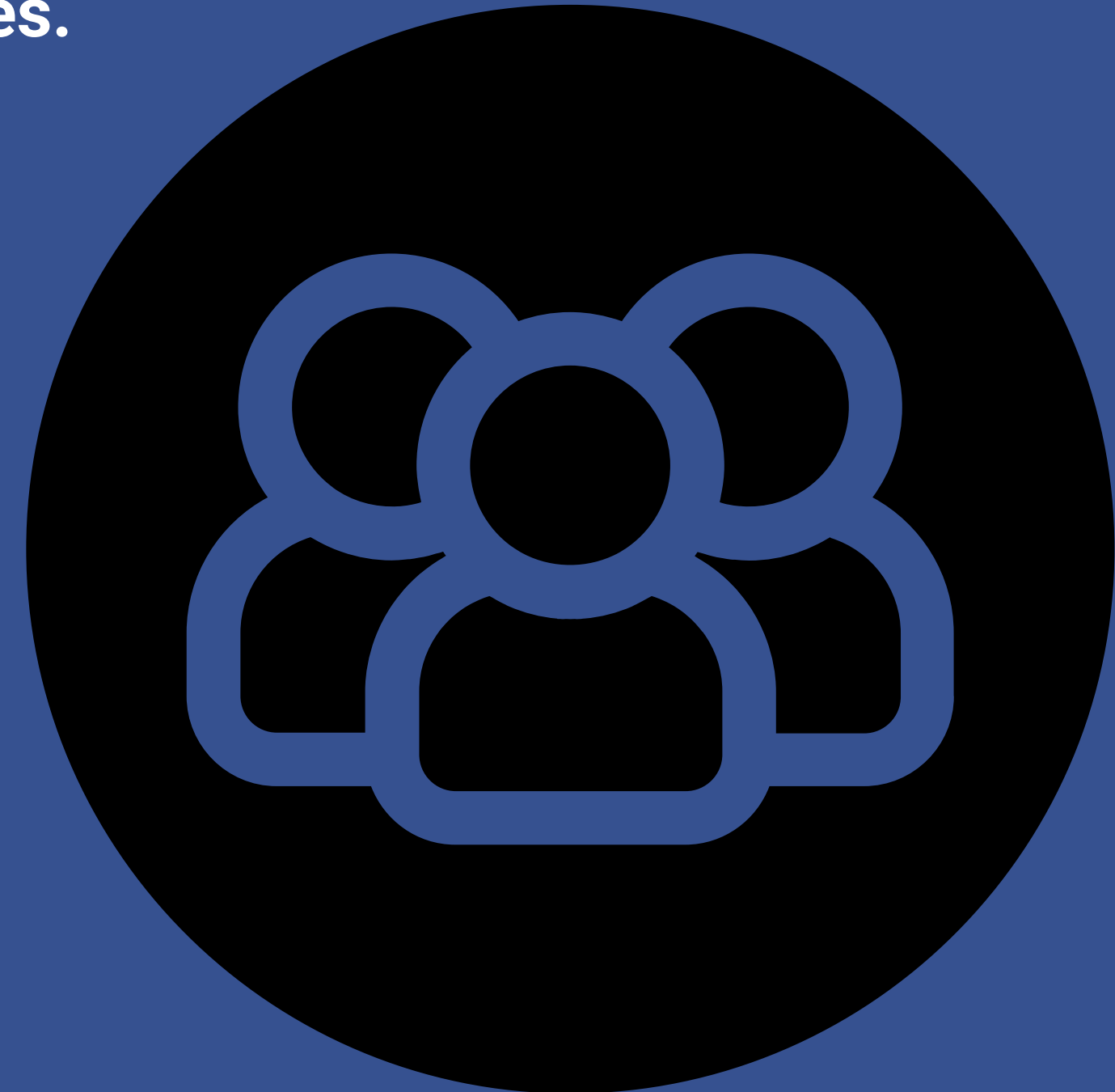


- Agregar sensores industriales con protocolos como Modbus.
- Incorporar AI para predicción de patrones ambientales.
- Expandir a múltiples ambientes con un sistema distribuido.
- Migración a servicios cloud como AWS IoT o Azure.



Aprendizajes del Equipo

- Trabajo colaborativo en entornos distribuidos.
- Integración de múltiples tecnologías (Hardware + Software).
- Buenas prácticas en desarrollo ágil y control de versiones.



Enlaces de Interés

Programador-FullStack-IoT/**Modulo-...**



Repositorio dedicado al proyecto correspondiente al modulo programador FullStack IoT perteneciente a la Tecnicatura Superior en Telecomunicaciones de ISPC

7

Contributors

27

Issues

1

Star

0

Forks



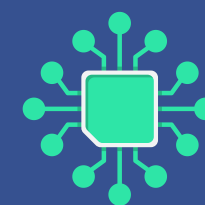
Programador-FullStack-IoT/Modulo-Programador-FullStack-IoT-TST-2024: Repositorio dedicado al proyect...

Repositorio dedicado al proyecto correspondiente al modulo programador FullStack IoT perteneciente a la Tecnicatura Superior en Telecomunicaciones de ISPC - Programador-FullStack-IoT/Modulo-...



GitHub

MUCHAS GRACIAS



Grupo Plata.