

SISTEMA DE MONITOREO IOT WEB-INTEGRADO

GRUPO 5 - INTEGRANTES:

- Caballero Peching, Carlos Arturo
- Isasi Melendez, Abraham
- Morales Buitron, Katherin Janet
- Portuguez Lozano, Carlos Enrique
- Samanez Espinoza, Aaron Jose Miller



**Universidad
Norbert Wiener**
Powered by Arizona State University

CURSO: Internet de las Cosas (IOT)
DOCENTE: SEGUNDO GERARDO GAMARRA QUISPE

DESCRIPCIÓN DEL PROYECTO

En muchos entornos laborales no se necesita controlar asistencia, sino verificar si hay actividad durante ciertos turnos. Este proyecto permite detectar movimiento y relacionarlo con los horarios del personal, para inferir quiénes podrían estar presentes. Es útil en casos como supervisión nocturna, detección de actividad fuera de horario o monitoreo general del uso de espacios, sin requerir identificación directa.



OBJETIVO GENERAL

Desarrollar un sistema de monitoreo basado en IoT que permita detectar actividad en un entorno físico mediante sensores de movimiento y relacionar dicha actividad con los turnos laborales registrados, con el fin de supervisar el uso del espacio y generar información contextual sobre la posible presencia de empleados, sin requerir identificación directa.



OBJETIVOS ESPECÍFICOS

- Implementar un sistema de detección de movimiento mediante un sensor PIR conectado a un microcontrolador ESP32.
- Transmitir los datos de detección a una plataforma web en tiempo real utilizando un formato estructurado.
- Diseñar una base de datos que almacene registros de actividad junto con los horarios laborales del personal.
- Desarrollar una interfaz web que permita visualizar las detecciones e interpretarlas en función de los turnos asignados.
- Facilitar el análisis contextual de la actividad registrada para identificar posibles presencias o anomalías horarias.

MARCO TEÓRICO

1. Internet de las Cosas (IoT)



El IoT consiste en la conexión de dispositivos físicos que recopilan y envían datos sin intervención humana. En este proyecto, permite que un sensor PIR detecte actividad y envíe la información a una plataforma web para su análisis. Su aplicación en entornos laborales mejora el monitoreo en tiempo real y la supervisión remota, aportando eficiencia y trazabilidad.

MARCO TEÓRICO

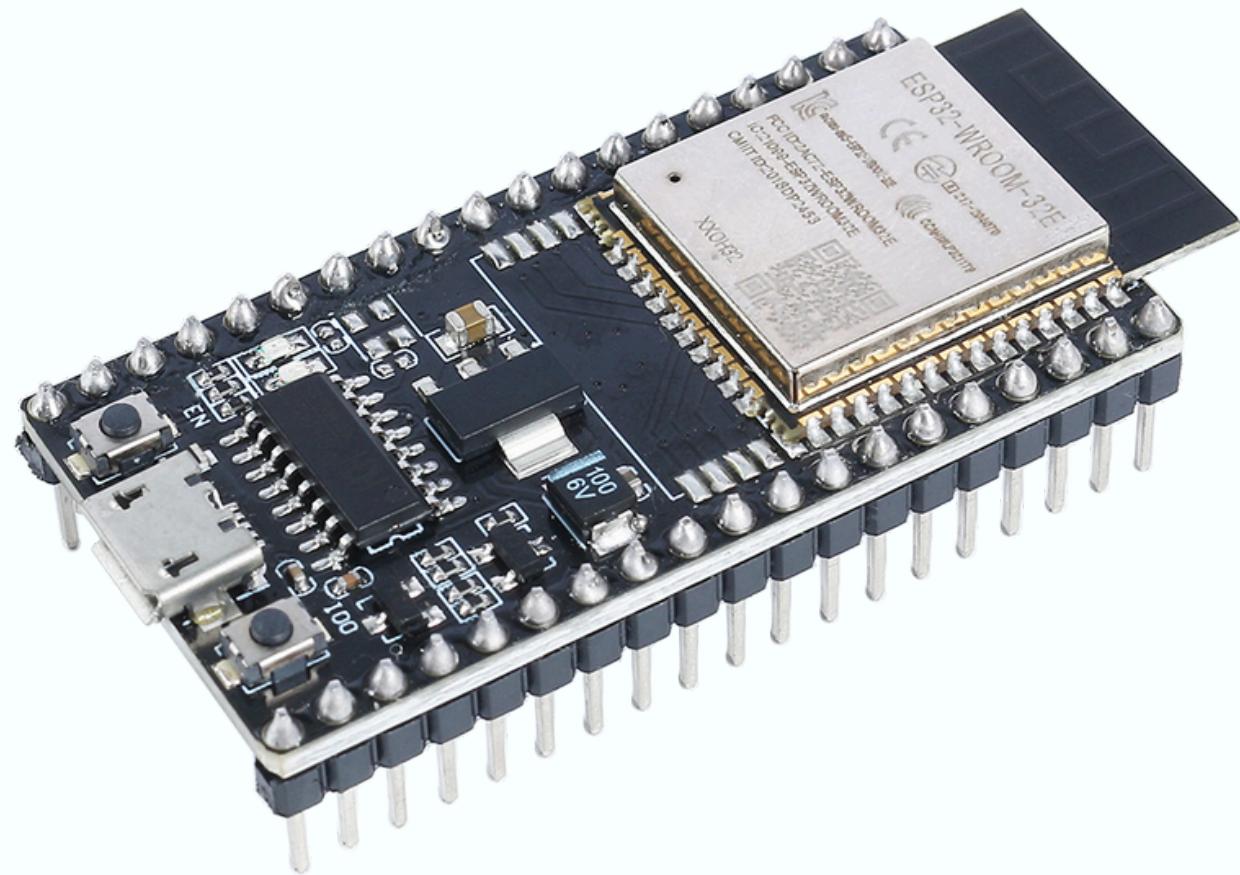
2. Sensores PIR (Passive Infrared Sensor)

Los sensores PIR detectan movimiento mediante cambios en la radiación infrarroja, permitiendo identificar actividad sin reconocer personas. Son económicos, de bajo consumo y se usan en seguridad y automatización. En este proyecto, permiten registrar cuándo hay movimiento en un entorno, para luego interpretarlo según los horarios laborales.



MARCO TEÓRICO

3. ESP32



El ESP32 es un microcontrolador con Wi-Fi y Bluetooth integrado, ideal para proyectos IoT por su bajo costo y alto rendimiento. En este proyecto, recibe señales del sensor PIR y las envía a una plataforma web, actuando como nodo central para lograr una comunicación directa, autónoma y en tiempo real.

MARCO TEÓRICO

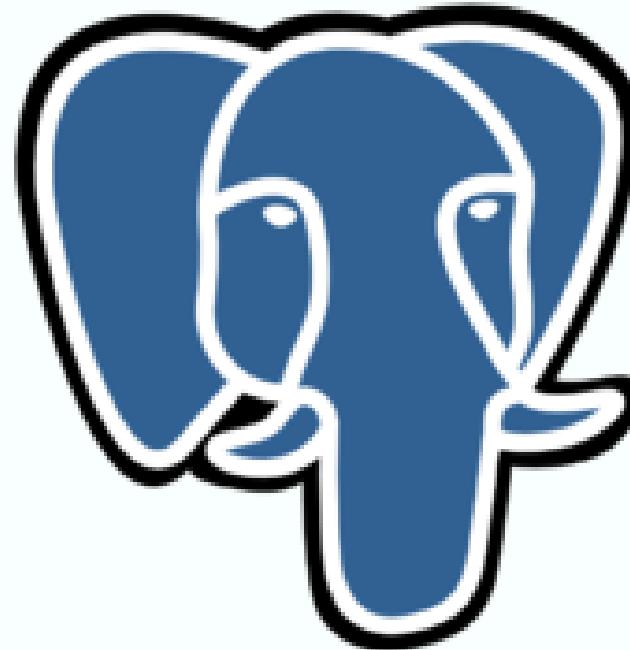
4. Transmisión de Datos y Formato JSON

En sistemas IoT, los datos se transmiten comúnmente mediante HTTP usando el formato JSON, que es ligero y fácil de interpretar. En este proyecto, el ESP32 envía la información del sensor PIR en JSON mediante una solicitud HTTP POST, permitiendo que el servidor la procese y almacene de forma estructurada y eficiente.



MARCO TEÓRICO

5. Bases de Datos Relacionales (PostgreSQL)



PostgreSQL es un sistema de bases de datos relacional, y en este proyecto se utilizó mediante la plataforma Neon, que lo ofrece como servicio en la nube. Su estructura permite almacenar registros, horarios y empleados de forma organizada, facilitando consultas para analizar la actividad detectada según los turnos laborales.

MARCO TEÓRICO

6. Interpretación de Datos

El sistema interpreta las detecciones de movimiento según la hora en que ocurren, comparándolas con los horarios laborales registrados. Así, infiere posibles presencias sin identificar personas directamente, lo que permite supervisar la actividad del entorno y detectar eventos fuera de horario de forma contextual.



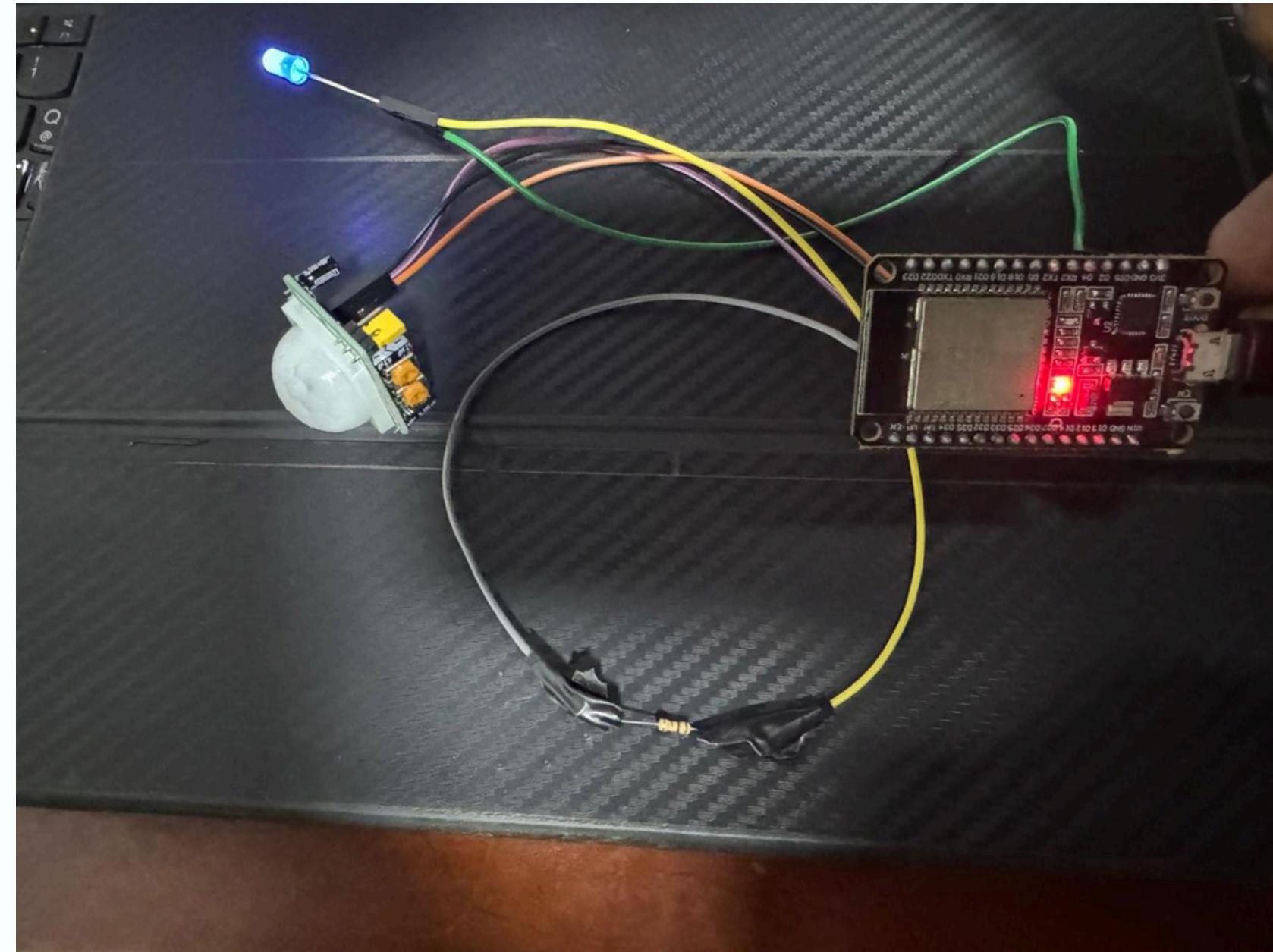
MARCO TEÓRICO

7. Aplicaciones similares



Los sensores PIR se usan ampliamente en oficinas y entornos industriales para detectar ocupación sin identificar personas. Son útiles para optimizar el uso de espacios, monitorear zonas en tiempo real y mejorar la seguridad en áreas restringidas. Combinados con plataformas IoT, ayudan a reducir costos y mejorar la gestión operativa.

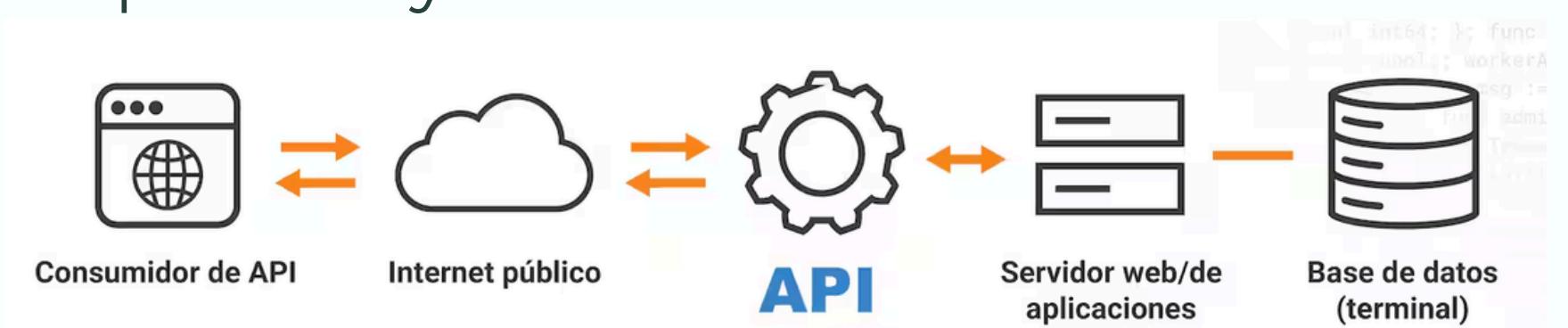
CIRCUITO



HERRAMIENTAS VIRTUALES DESARROLLADAS

1. API

Se creó una API en Node.js para recibir los datos enviados por el ESP32 en formato JSON. Esta API valida y procesa cada detección del sensor PIR, y luego almacena los registros en la base de datos PostgreSQL alojada en Neon, permitiendo una comunicación estable y en tiempo real entre el dispositivo y el sistema web.



HERRAMIENTAS VIRTUALES DESARROLLADAS

2. Base de datos (Neon Postgres)

La base de datos se implementó en Neon, utilizando PostgreSQL en la nube. Se definieron tablas para detecciones, empleados y horarios laborales, lo que permite organizar y consultar la información de forma eficiente. Las relaciones entre tablas permiten vincular la actividad detectada con los posibles empleados presentes en ese momento.



HERRAMIENTAS VIRTUALES DESARROLLADAS

3. Interfaz web (Next.js)

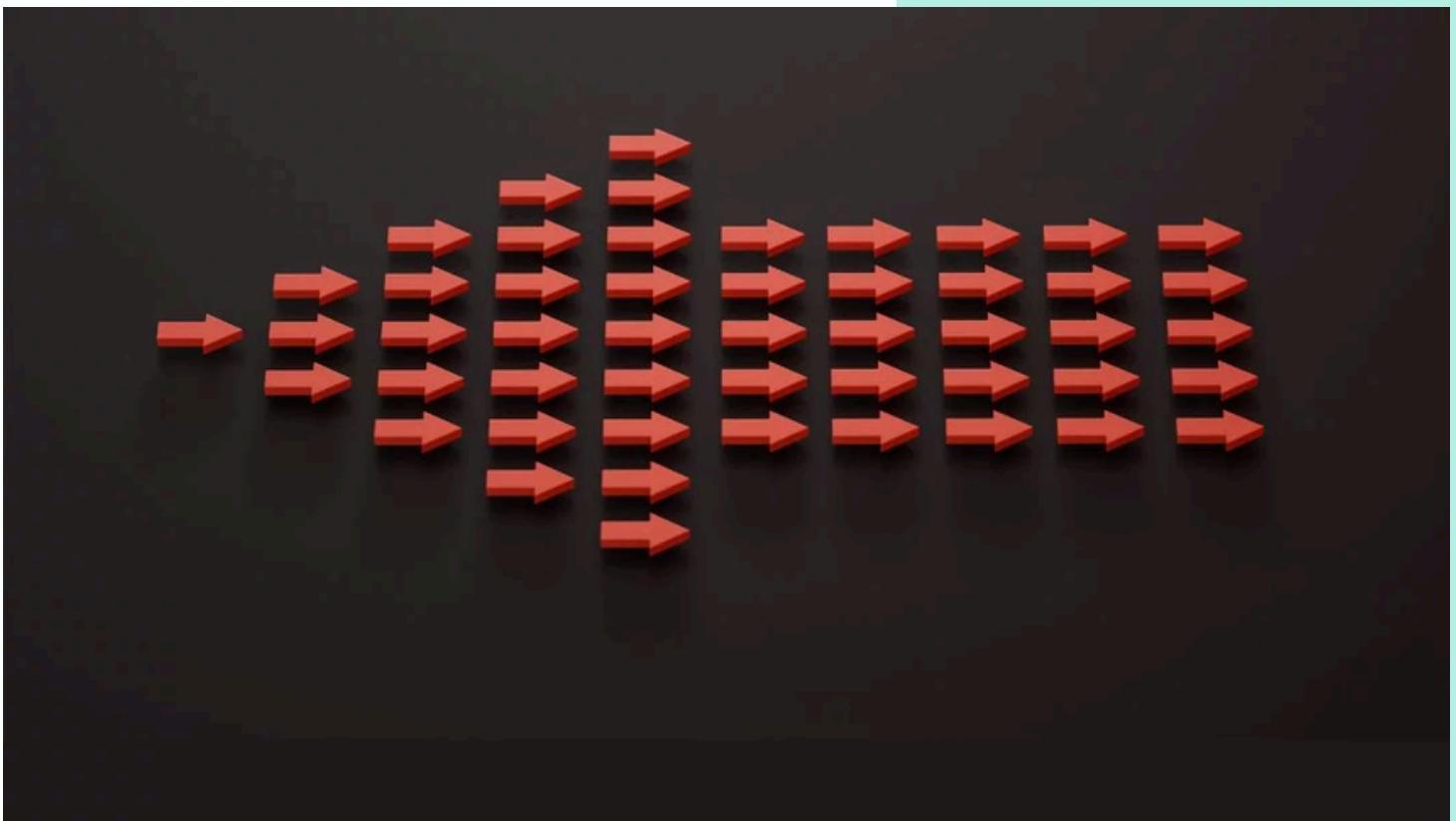
Se creó una interfaz web con Next.js para mostrar los datos recolectados por el sistema. Permite visualizar los registros de actividad por fecha y hora, vinculados a los turnos laborales, facilitando la identificación de patrones y eventos fuera de horario. La información se actualiza en tiempo real desde la base de datos.



HERRAMIENTAS VIRTUALES DESARROLLADAS

4. Lógica de correlación horaria

El sistema incluye una lógica que compara cada detección con los horarios laborales registrados para inferir qué empleados podrían haber estado presentes. Aunque no identifica personas directamente, permite contextualizar la actividad detectada y generar reportes sin necesidad de sistemas de acceso específicos.



RESULTADOS

Gestión de Empleados

[Registrar empleado](#) [Ver empleados registrados](#) [Volver al inicio](#)

Registrar Empleado

Nombres:

Apellidos:

DNI:

[Registrar](#)

RESULTADOS

Gestión de Empleados

[Registrar empleado](#) [Ver empleados registrados](#) [Volver al inicio](#)

Gestión de Empleados

Asignar horario (UTC-5)

Empleado:

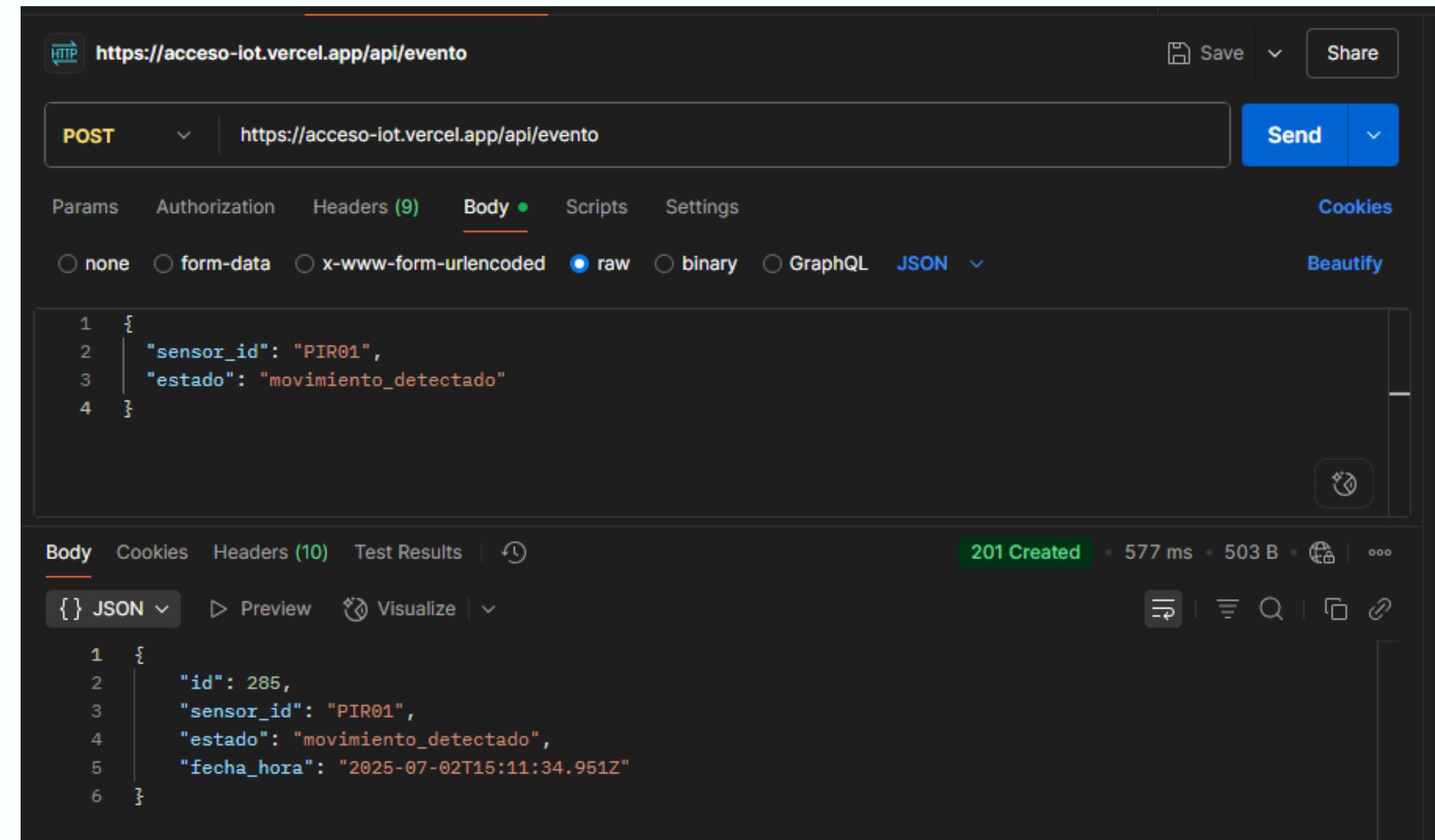
hora_inicio: 🕒

hora_fin: 🕒

Asignar horario

El contrato debe tener entre 4 y 8 horas de duración.

RESULTADOS



HTTP <https://acceso-iot.vercel.app/api/evento>

POST <https://acceso-iot.vercel.app/api/evento> Send

Params Authorization Headers (9) Body Scripts Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify

```
1 {
2   "sensor_id": "PIR01",
3   "estado": "movimiento_detectado"
4 }
```

Body Cookies Headers (10) Test Results 201 Created 577 ms 503 B

{ } JSON Preview Visualize

```
1 {
2   "id": 285,
3   "sensor_id": "PIR01",
4   "estado": "movimiento_detectado",
5   "fecha_hora": "2025-07-02T15:11:34.951Z"
6 }
```

RESULTADOS

Panel de Monitoreo

Estado Actual

Movimiento Detectado

Última detección: 2/7/2025, 10:11:34 a. m.

Empleados en turno

Coco Lany

Volver al inicio

RESULTADOS

ESP-32 | Arduino IDE 2.3.6

Archivo Editar Sketch Herramientas Ayuda

GESTOR DE BIBLIOTECAS

Filtre su búsqueda...

Tipo: Todo Tema: Todo

AIPIc_Opta de Arduino

Arduino IDE PLC runtime library for Arduino Opta This is the runtime library and plugins fo...
Más información

1.2.0 INSTALAR

AIPIc_PMC de Arduino

Arduino IDE PLC runtime library for Arduino Portenta Machine Control This is the runtime...
Más información

1.0.6 INSTALAR

Arduino Cloud Provider Examples de...

Examples of how to connect various Arduino boards to cloud providers
Más información

1.2.1 INSTALAR

ESP-32.ino

```
17 // Variables de detección
18 int anterior = LOW;
19 > unsigned long ultimoMovimiento = 0; ...
38   Serial.println("\nTiempo agotado");
39   return;
```

Mensaje (Intro para mandar el mensaje de 'ESP32-WROOM-DA Module' a 'COM5')

Movimiento detectado

HTTP: 200
Respuesta: Movimiento registrado con éxito en 'dbo.pc_sp_inserta_eventos'
Movimiento detectado

HTTP: 200
Respuesta: Movimiento registrado con éxito en 'dbo.pc_sp_inserta_eventos'
Movimiento detectado

HTTP: 200
Respuesta: Movimiento registrado con éxito en 'dbo.pc_sp_inserta_eventos'
Movimiento detectado

HTTP: 200
Respuesta: Movimiento registrado con éxito en 'dbo.pc_sp_inserta_eventos'
Movimiento detectado

HTTP: 200
Respuesta: Movimiento registrado con éxito en 'dbo.pc_sp_inserta_eventos'
Movimiento detectado

HTTP: 200
Respuesta: Movimiento registrado con éxito en 'dbo.pc_sp_inserta_eventos'

Conectado a WiFi
IP: 192.168.1.18
Sin movimiento prolongado

HTTP: 201
Respuesta: {"id":6,"sensor_id":"PIR01","estado":"sin_movimiento","fecha_hora":"2025-06-26T01:32:05.489Z"}
Sin movimiento prolongado

HTTP: 201
Respuesta: {"id":7,"sensor_id":"PIR01","estado":"sin_movimiento","fecha_hora":"2025-06-26T01:32:15.725Z"}
Sin movimiento prolongado

HTTP: 201
Respuesta: {"id":8,"sensor_id":"PIR01","estado":"sin_movimiento","fecha_hora":"2025-06-26T01:32:25.875Z"}
Sin movimiento prolongado

Descargando el índice: package_esp8266com_index.json

Lín. 61, col. 27 ESP32-WROOM-DA Module en COM5 4 20:32 25/06/2025 31

RESULTADOS

Histórico de Actividades

[← Volver al inicio](#)

[Consultar](#)

Contratos del empleado:

9:00:00 a. m. - 1:00:00 p. m.

#	Estado	Fecha y hora
1	movimiento_detectado	2/7/2025, 10:11:34 a. m.

Total de eventos: 1

Página 1 de 1 [Ir](#) [Anterior](#) [Siguiente](#)

CONCLUSIONES

El proyecto permitió implementar un sistema de monitoreo IoT funcional que detecta actividad y la interpreta según horarios laborales, sin identificar personas. Se concluye que esta solución es útil para supervisar entornos de trabajo de forma básica, con bajo costo y posibilidad de ampliación futura.

BIBLIOGRAFÍA

- Código Villafañe. (2022). Subir repositorio a GitHub luego deploy a Vercel 🖥 💪 [Video]. YouTube. <https://youtu.be/MZEMjyGG0NY?si=1-MvRiqMEIQRgyck>
- HackMamba. (2024, marzo 12). Build a simple e-commerce PIM with Next.js, Prisma, and Neon. DEV Community. <https://dev.to/hackmamba/build-a-simple-e-commerce-pim-with-nextjs-prisma-and-neon-4ei2>
- Neon: Serverless Postgres Database. (2024). Neon: Serverless Postgres Database [Video]. YouTube. https://www.youtube.com/watch?v=EB0Nu_e9wCs

LINKS DEL PROYECTO

- **Código fuente:** <https://github.com/CaballeroPCarlos/acceso-iot>
- **Página alojada en:** <https://acceso-iot.vercel.app>
- **Demostración de funcionalidad:** <https://youtu.be/hU5ptVF-bi4>

MUCHAS
GRACIAS