AI & SUSTAINABILITY IN VET EDUCATION

Nombre de los/las autores/as: Politeknika Txorierri

Título del WP: WP3 – AI-SAVES@ProjectHub

Referencia de la Actividad: Sistema de gestión de aparcamiento

Logo:

POLITEKNIKA TXORIERRI

Cofinanciado por la Unión Europea

Este proyecto ha sido financiado con el apoyo de la Comisión Europea. Esta comunicación refleja únicamente las opiniones del autor, y la Comisión no puede ser considerada responsable del uso que pueda hacerse de la información contenida en la misma.

TABLA DE CONTENIDOS

1. Título de Proyecto: 5
2. Categoría del Proyecto: 5
3. Descripción del Proyecto: 5
4. Palabras Clave: 6
5. Público Objetivo: 6
6. Objetivos del Proyecto: 6
7. Cronograma del Proyecto: 6
8. Materiales Necesarios 7
9. Recursos Proporcionados: 8
10. Herramientas y Tecnologías Utilizadas: 8
11. Plan de implementación paso a paso: 8
12. Criterios de Evaluación: 9
13. Idiomas Disponibles: 10
14. Contribuidores/as: 10
15. Retroalimentación y mejora Continua: 10

Cofinanciado por la Unión Europea

Este proyecto ha sido financiado con el apoyo de la Comisión Europea. Esta comunicación refleja únicamente las opiniones del autor, y la Comisión no puede ser considerada responsable del uso que pueda hacerse de la información contenida en la misma.

1. **Título de Proyecto:**

Eficiencia energética mediante IoT: Aparcamiento inteligente

2. **Categoría del Proyecto:**

IoT.

3. **Descripción del Proyecto:**

Este proyecto tiene como objetivo diseñar e implementar un sistema de aparcamiento inteligente que monitorice automáticamente el acceso de vehículos, detecte la ocupación de plazas de aparcamiento y proporcione información en tiempo real sobre la disponibilidad a través de una infraestructura IoT. El sistema mejora la gestión del tráfico y la experiencia del usuario al ofrecer datos actualizados sobre la capacidad del aparcamiento, reduciendo la congestión y la circulación innecesaria de vehículos.

El sistema utiliza múltiples microcontroladores ESP32 equipados con sensores de distancia para monitorizar el estado de ocupación de plazas individuales. Cada ESP32 es responsable de una plaza de aparcamiento y reporta continuamente si la plaza está ocupada o libre en función de las mediciones de distancia. Además, un ESP32 independiente situado en la entrada del aparcamiento monitoriza la entrada y salida de vehículos, permitiendo al sistema mantener un recuento preciso del número de vehículos presentes en el interior. Todas las actualizaciones de estado se transmiten mediante MQTT al controlador central.

Para proporcionar una visión global y coherente del aparcamiento, una Raspberry Pi actúa como controlador principal y broker MQTT. Agrega los datos recibidos de todos los nodos ESP32, calcula el número de plazas ocupadas y disponibles, y determina si se ha alcanzado la capacidad máxima. Esta información se publica en un panel de visualización utilizando Node-RED, permitiendo a usuarios y administradores evaluar rápidamente la disponibilidad. Si el aparcamiento se llena, el sistema puede activar alertas o indicadores visuales para informar a los conductores y evitar nuevos accesos.

Desde una perspectiva de sostenibilidad, el sistema de aparcamiento inteligente ayuda a reducir el consumo de combustible y las emisiones al minimizar el tiempo que los vehículos pasan buscando plazas libres. Esto contribuye a una movilidad urbana más eficiente y apoya los principios de smart city centrados en la optimización de recursos y la responsabilidad ambiental.

- La arquitectura de funcionamiento del sistema es la siguiente:

Cofinanciado por la Unión Europea

Este proyecto ha sido financiado con el apoyo de la Comisión Europea. Esta comunicación refleja únicamente las opiniones del autor, y la Comisión no puede ser considerada responsable del uso que pueda hacerse de la información contenida en la misma.

- Detectar la entrada y salida de vehículos en el acceso al aparcamiento.
- Monitorizar el estado de ocupación de cada plaza.
- Actualizar el número total de vehículos dentro del aparcamiento.
- Agregar los datos de las plazas en el controlador central.
- Calcular plazas disponibles y ocupadas.
- Publicar información en tiempo real en el panel de Node-RED.

4. **Palabras Clave:**

IoT, Formación Profesional, Automatización, Sostenibilidad, Cuidado ambiental, Aprendizaje práctico

5. **Público Objetivo:**

Tanto docentes como estudiantes.

6. **Objetivos del Proyecto:**

- Desarrollar conocimientos prácticos en IoT.
- Comprender cómo diseñar e implementar un sistema inteligente de gestión de aparcamiento.
- Adquirir experiencia práctica en la programación de microcomputadores y microcontroladores.
- Comprender el diseño a nivel de sistema: conexión de sensores, control de actuadores y gestión del flujo de datos.
- Mejorar las habilidades de trabajo en equipo, documentación y presentación.
- Explorar la sostenibilidad mediante decisiones de diseño energéticamente eficientes.

7. **Cronograma del Proyecto:**

Tarea	**Horas estimadas**
Hardware y programación de Raspberry y ESP32 + instalación y configuración de Node-RED	13 horas
Práctica y validación	4 horas
Preparación de documentación y presentación	3 horas
Total	**20 horas**

Cofinanciado por la Unión Europea

Este proyecto ha sido financiado con el apoyo de la Comisión Europea. Esta comunicación refleja únicamente las opiniones del autor, y la Comisión no puede ser considerada responsable del uso que pueda hacerse de la información contenida en la misma.

8. Recursos Proporcionados:

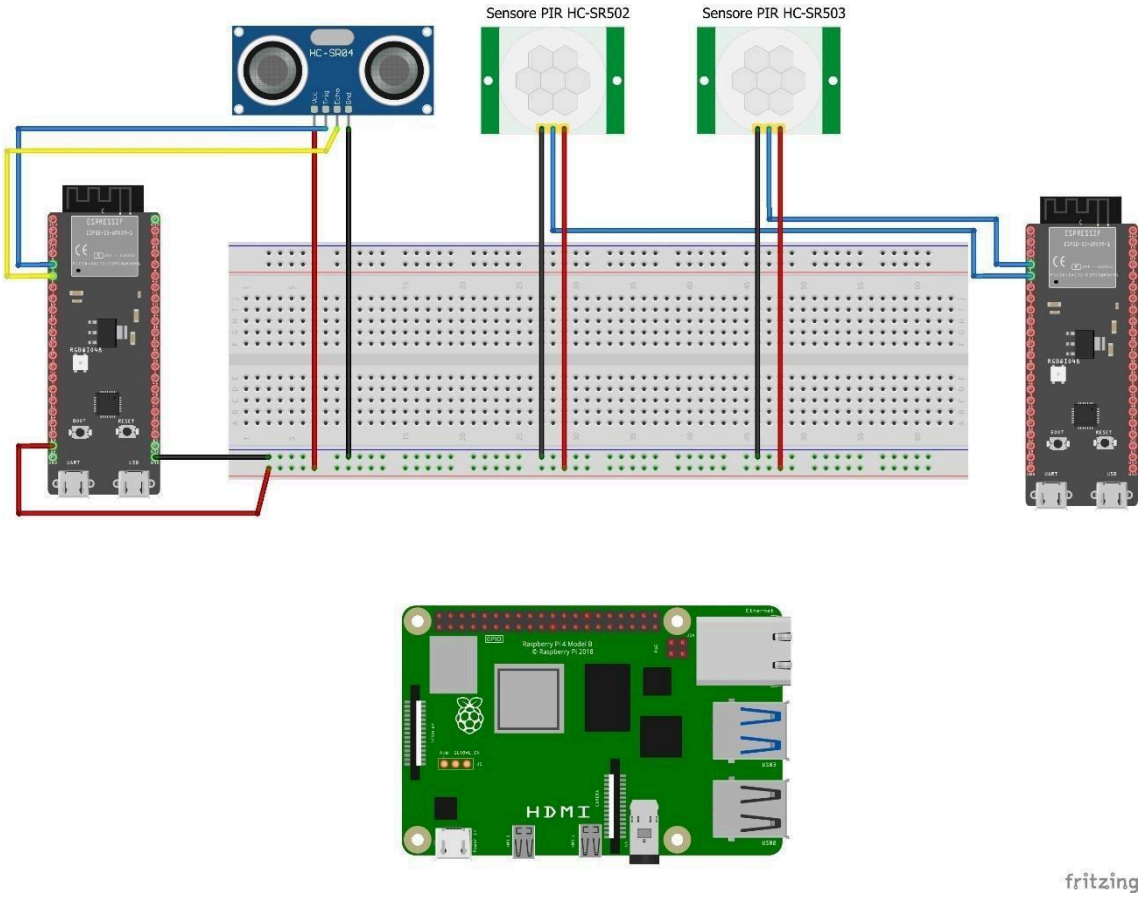
- Raspberry Pi, ESP32 y kits de sensores (AI-SAVES Kit)
- Guías de programación
- Software y hardware necesarios

9. Herramientas y Tecnologías Utilizadas:

- Raspberry Pi 4 y ESP32
- Hardware: sensor ultrasónico, sensor PIR
- Lenguajes de programación Python y C/C++, utilizando Geany e IDE
- Aplicación Node-RED y protocolo MQTT
- GitHub para control de versiones y colaboración
- Fritzing para diseño y prueba de circuitos

10. Plan de implementación paso a paso:

1) Configurar la Raspberry Pi con el software necesario (Raspbian OS, Python).
2) Realizar sesiones prácticas donde el alumnado construya y programe el sistema domótico basado en IoT.



Cofinanciado por la Unión Europea

Este proyecto ha sido financiado con el apoyo de la Comisión Europea. Esta comunicación refleja únicamente las opiniones del autor, y la Comisión no puede ser considerada responsable del uso que pueda hacerse de la información contenida en la misma.

a) Montar el circuito según el mapeo de pines del ESP32.
b) Cargar el código necesario (pueden utilizarse los dos archivos.ino proporcionados para trabajar localmente) en el ESP32 y comprobar que funciona correctamente; deben verificarse la medición de distancia (para decidir si hay un coche en la plaza) y la detección de presencia (para determinar si un coche entra o sale del aparcamiento).
c) Activar el entorno virtual de Python en la Raspberry Pi:

```
cd project_folder_path

python -m venv .venv

source .venv/bin/activate
```

d) Instalar las librerías necesarias:

```
pip install paho-mqtt
```

e) Ejecutar el script necesario/proporcionado usando Geany:

```
python parking-control.py
```

f) Cargar los archivos necesarios/proporcionados en el ESP32 usando el IDE:

```
in-out.ino

occupied.ino
```

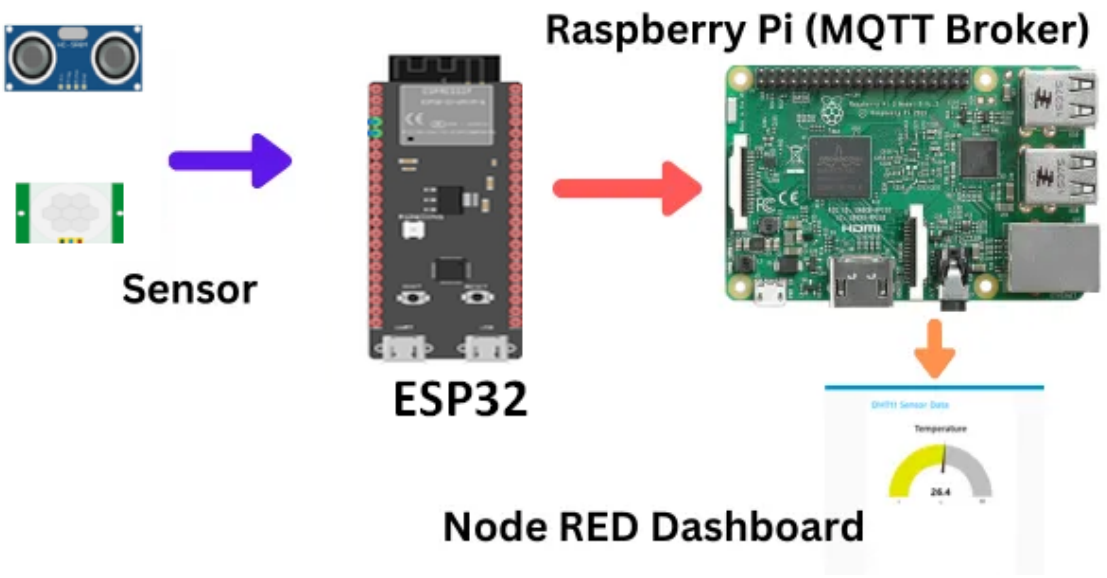
g) Asegurarse de que la Raspberry Pi y el ESP32 están conectados a la misma red WiFi (SSID).
h) Instalar Node-RED en la Raspberry e importar el archivo JSON proporcionado para crear el panel:

```
dashboard.json
```

i) Utilizar las diferentes entradas y salidas para comprobar que el sistema funciona correctamente y verificar que los datos se muestran adecuadamente en el panel de Node-RED.

Cofinanciado por la Unión Europea

Este proyecto ha sido financiado con el apoyo de la Comisión Europea. Esta comunicación refleja únicamente las opiniones del autor, y la Comisión no puede ser considerada responsable del uso que pueda hacerse de la información contenida en la misma.



3) Evaluar la comprensión del alumnado mediante demostraciones prácticas del proyecto.

11. Criterios de Evaluación:

- **Implementación del hardware (25%):** Circuito correctamente conectado, sin cables sueltos y con sensores correctamente colocados.
- **Funcionamiento del código (30%):** El script se ejecuta sin errores, detecta correctamente la ocupación de plazas y la entrada/salida de vehículos, incluyendo funcionalidad MQTT.
- **Funcionamiento de Node-RED (15%):** El panel muestra correctamente los datos proporcionados por la Raspberry.
- **Comprensión y explicación (20%):** Capacidad para explicar el funcionamiento del sistema y sus aplicaciones reales.
- **Creatividad y optimización (10%):** Propuestas de mejora, como añadir funcionalidades adicionales de automatización.

12. Idiomas Disponibles:

- inglés y español

13. Contribuidores/as:

- **Aitor Rey (Experto técnico) –** Aportó experiencia en IoT.
- **Jokin Goioaga (Experto técnico)** – Aportó experiencia en IoT.

Cofinanciado por la Unión Europea

Este proyecto ha sido financiado con el apoyo de la Comisión Europea. Esta comunicación refleja únicamente las opiniones del autor, y la Comisión no puede ser considerada responsable del uso que pueda hacerse de la información contenida en la misma.

14. Retroalimentación y mejora Continua:

- **Encuestas**: Al final de cada hito, alumnado y profesorado completarán encuestas para evaluar la eficacia del proyecto.
- **Revisión del proyecto:** Tras la finalización, se realizará una reunión de revisión para debatir posibles mejoras en futuros proyectos basados en IA.
- **Análisis de datos:** Los datos recogidos se analizarán para identificar fortalezas y áreas de mejora, orientando futuros desarrollos.

Cofinanciado por la Unión Europea

Este proyecto ha sido financiado con el apoyo de la Comisión Europea. Esta comunicación refleja únicamente las opiniones del autor, y la Comisión no puede ser considerada responsable del uso que pueda hacerse de la información contenida en la misma.