



UNIVERSIDAD EUROPEA DE MADRID

ESCUELA DE ARQUITECTURA, INGENIERÍA Y DISEÑO

GRADO EN INGENIERÍA INFORMÁTICA

PROYECTO DE INGENIERÍA INFORMÁTICA

ANTEPROYECTO GRUPO 3

Fernando Fernández Rabanal

Daniel Vicente Moldovan

Daniel Nicolás Ramírez Ocampos

Álvaro López López

CURSO 2025-2026

Índice

Índice.....	2
Capítulo 1. RESUMEN DEL PROYECTO.....	3
1.1 Planteamiento del problema.....	3
1.2 Objetivos del proyecto.....	3
1.3 Resultados obtenidos.....	4
1.4 Estructura del proyecto.....	4
Capítulo 2. INTRODUCCIÓN.....	6
2.1 Contexto.....	6
2.2 Estado del arte.....	6
2.3 Planteamiento del problema.....	7
Capítulo 3. OBJETIVOS.....	8
3.1 Objetivos generales.....	8
3.2 Objetivos específicos.....	8
3.3 Beneficios del proyecto.....	9
Capítulo 4. DESARROLLO DEL PROYECTO.....	10
4.1 Metodología y herramientas empleadas.....	10
Capítulo 5. Planificación del proyecto.....	11
Capítulo 6. PRESUPUESTO.....	13
Estimación de horas.....	13
Equipo técnico utilizado.....	13
Software utilizado.....	13
Materiales empleados.....	14
Resumen Final.....	15
Capítulo 7. Equipo de trabajo.....	16
REFERENCIAS Y BIBLIOGRAFÍA.....	17

Capítulo 1. RESUMEN DEL PROYECTO

1.1 Planteamiento del problema

Nuestro proyecto busca resolver la necesidad de contar con un sistema que facilite la gestión de una comisaría de policía de manera más sencilla, segura y eficiente. Actualmente, muchas instalaciones policiales carecen de una solución centralizada que integre el monitoreo de sensores como temperatura, humo o luz con el control de dispositivos como luces, puertas o ventiladores, todo en una sola plataforma fácil de usar. Esta ausencia genera ineficiencias, incrementa los riesgos de seguridad y complica la toma de decisiones en tiempo real para el personal, ya que los procesos no están suficientemente automatizados ni adaptados a los distintos roles de los usuarios. El problema principal que abordamos es la falta de una plataforma unificada de gestión inteligente que permita a los responsables supervisar y controlar de manera centralizada aspectos clave como la temperatura, la calidad del aire, la detección de humo, la iluminación, la apertura de puertas y la vigilancia mediante cámaras. Queremos crear un programa que una todos estos elementos en una interfaz clara y adaptable, permitiendo a administradores gestionar el sistema, a policías visualizar datos y a otros usuarios, como presos, interactuar según sus roles, todo funcionando sin problemas en cualquier dispositivo, desde computadoras hasta tablets.

1.2 Objetivos del proyecto

Objetivos generales

El objetivo general del presente trabajo, desarrollado por Qualifast Buildings, consiste en diseñar e implementar un sistema de monitorización y control basado en el Internet de las Cosas (IoT) en Python, que gestione de manera eficiente una comisaría de policía integrando sensores y actuadores en una interfaz sencilla y adaptable.

Objetivos específicos

- Integrar sensores y actuadores en una aplicación.
- Ofrecer vistas personalizadas según el rol del usuario.
- Mostrar datos en tiempo real de forma clara, incluyendo históricos.
- Implementar alertas para situaciones críticas.
- Garantizar portabilidad con ficheros JSON y arquitectura MVC.

1.3 Resultados obtenidos

Hasta el momento, se ha definido un Product Backlog inicial adaptado a las necesidades de la comisaría, incluyendo historias de usuario para monitoreo, control de actuadores y alertas básicas. Además, se ha establecido una estructura preliminar del proyecto y se han identificado las herramientas necesarias, como Flet y PyCharm, para su desarrollo.

Aunque el proyecto se encuentra en fase inicial, se prevé que los resultados alcanzados permitan:

- Disponer de una aplicación funcional que centralice la gestión de sensores y actuadores en la comisaría.
- Mejorar la seguridad y la eficiencia operativa de las instalaciones mediante la automatización de procesos.
- Facilitar la toma de decisiones en tiempo real gracias a la visualización de datos y alertas.
- Ofrecer una interfaz gráfica coherente, adaptable a diferentes resoluciones y accesible para todos los roles definidos

1.4 Estructura del proyecto

La memoria se organiza en capítulos que abordan: un resumen ejecutivo (Capítulo 1), una introducción con contexto y estado del arte (Capítulo 2), los objetivos detallados (Capítulo 3), el desarrollo con metodología y planificación (Capítulo 4), el presupuesto (Capítulo 5), el equipo de trabajo (Capítulo 6) y las referencias (Capítulo 7), facilitando una lectura estructurada y metodológica.

Capítulo 2. INTRODUCCIÓN

El presente proyecto aborda el desarrollo de un sistema de monitorización basado en el Internet de las Cosas (IoT) para gestionar eficientemente una comisaría de policía mediante sensores y actuadores, integrando tecnología avanzada en un entorno de seguridad.

2.1 Contexto

El problema surge de la necesidad de optimizar la gestión de condiciones ambientales y la seguridad en comisarías, donde la falta de sistemas integrados puede comprometer la operatividad y el bienestar del personal. La solución propuesta busca automatizar procesos y proporcionar datos en tiempo real, respondiendo a una demanda creciente de eficiencia en entornos policiales.

2.2 Estado del arte

- **Sensor de temperatura (DHT11/DHT22):** Dispositivos digitales que miden temperatura y humedad, utilizados en monitoreo ambiental. El DHT11 ofrece una solución básica, mientras que el DHT22 proporciona mayor precisión, ideal para áreas críticas como salas de detención.
- **Fotorresistor (LDR):** Sensor de luz que ajusta su resistencia según la intensidad lumínica, empleado para activar iluminación automática en pasillos o estacionamientos, mejorando la seguridad nocturna.
- **Sensor de humo (MQ-2):** Detecta gases combustibles mediante cambios en su resistencia, útil para sistemas de alarma contra incendios en oficinas o garajes.
- **Sensor de calidad de aire (MQ-135):** Mide contaminantes como CO₂, aplicable en espacios cerrados para garantizar la salud del personal y detectar sustancias sospechosas.
- **Diodos LED:** Emiten luz para iluminación de emergencia o indicadores de estado, controlables mediante PWM para adaptarse a necesidades específicas.
- **Motor DC 12V:** Automatiza puertas o barreras, integrándose con sensores de distancia para accesos seguros.
- **Ventilador 5V:** Regula la temperatura en salas de servidores, con control variable basado en lecturas térmicas.
- **Placa ESP32:** Microcontrolador con WiFi y Bluetooth, actúa como núcleo para integrar sensores en un sistema IoT.
- **Cámara OV2640:** Captura imágenes para vigilancia, con capacidad de streaming y detección de movimiento.

2.3 Planteamiento del problema

El desafío principal es la ausencia de una solución integrada que combine monitoreo ambiental, control de actuadores y gestión de usuarios en una comisaría. Basado en el estado del arte, se identifica una carencia de sistemas que unifiquen estos elementos en una interfaz adaptable, lo que este proyecto busca resolver mediante un software desarrollado con FLET y metodologías ágiles, adaptado a roles específicos como administradores, policías y presos.

Capítulo 3. OBJETIVOS

3.1 Objetivos generales

El objetivo general del presente trabajo consiste en desarrollar un sistema de monitorización basado en el Internet de las Cosas (IoT) para gestionar de forma eficiente una comisaría de policía, integrando sensores y actuadores en una interfaz adaptable y fácil de usar. Este sistema busca transformar la manera en que se supervisan las condiciones ambientales y se controla la seguridad dentro de la comisaría, ofreciendo una solución práctica que ahorre tiempo y recursos al personal. La idea es crear una herramienta que no sólo reúna datos de diferentes sensores, como temperatura o humo, sino que también permita activar actuadores como luces o puertas de forma automática o manual, todo desde una plataforma intuitiva que se ajuste a las necesidades de cada usuario. Además, al ser un proyecto desarrollado por Qualifast Buildings, se enfoca en cumplir con los requisitos de portabilidad y escalabilidad, asegurando que pueda adaptarse a otras instalaciones en el futuro. Este enfoque innovador responde a la creciente demanda de tecnologías inteligentes en entornos de seguridad, donde la integración de dispositivos IoT puede marcar una diferencia significativa en la operatividad diaria.

3.2 Objetivos específicos

- **Diseñar una interfaz gráfica funcional:** Crear una interfaz usando FLET en Python que muestre de manera clara y organizada los datos captados por los sensores, como temperatura (DHT11/DHT22), humo (MQ-2) y calidad del aire (MQ-135), cumpliendo con el requisito de coherencia estética y visualización intuitiva.
- **Implementar el patrón Modelo-Vista-Controlador:** Asegurarse de que el software siga esta estructura para garantizar una separación clara entre la lógica, la presentación y el control, facilitando su mantenimiento y escalabilidad.
- **Desarrollar un sistema de login adaptable:** Configurar un módulo de registro y acceso que distinga entre roles (administrador, jefe de policía y policía), permitiendo vistas personalizadas según las necesidades de cada usuario, como exige el requisito de una única aplicación para todos los roles.
- **Integrar al menos tres tipos de sensores por usuario principal:** Asignar a un rol como el administrador sensores como fotorresistores (LDR) para luz, cámaras (OV2640) para vigilancia y sensores de humo (MQ-2), cumpliendo con la necesidad de diversidad en la monitorización.
- **Crear ficheros JSON con datos realistas:** Desarrollar archivos JSON que contengan perfiles realistas de usuarios principales como administradores, jefes de policía y policías, incluyendo supervisores (como jefes o técnicos), usuarios clínicos (médicos o analistas), y también presos, con información veraz como DNI, número de teléfono y direcciones.
- **Habilitar el control de actuadores:** Permitir la gestión remota de dispositivos como LEDs para iluminación, motores DC 12V para puertas y ventiladores 5V para ventilación, integrándolos con los datos de los sensores para respuestas automáticas.

- **Planificar y ejecutar sprints con SCRUM:** Organizar el desarrollo en sprints definidos (como los cuatro listados en el Product Backlog), asignando tareas específicas a cada etapa y monitorizando el progreso con tableros Trello y actas de reunión.

3.3 Beneficios del proyecto

Este proyecto aportará una mejora significativa en la gestión de la comisaría al proporcionar un sistema que automatiza el monitoreo ambiental y la seguridad, reduciendo el tiempo de respuesta ante emergencias como incendios o intrusiones. Ofrecerá a los administradores un control total sobre los dispositivos, a los policías una vista clara de los datos y a los presos un registro básico, todo en una plataforma intuitiva. Además, al usar ficheros JSON portátiles y una interfaz adaptable, facilitará la escalabilidad y el mantenimiento, beneficiando tanto al personal como a la eficiencia operativa de Qualifast Buildings en el desarrollo de soluciones IoT.

Capítulo 4. DESARROLLO DEL PROYECTO

4.1 Metodología y herramientas empleadas

Para llevar a cabo este proyecto, hemos elegido una metodología ágil basada en SCRUM, que nos permite trabajar en ciclos cortos (sprints) de desarrollo, adaptándonos a los cambios y mejorando continuamente el producto. Este enfoque nos ayuda a dividir el trabajo en tareas manejables, como las historias de usuario definidas en el Product Backlog, y a realizar revisiones regulares con el equipo para ajustar el rumbo según sea necesario. Además, utilizaremos tableros Trello para organizar las tareas, actas de reunión para documentar avances y rúbricas de evaluación inter pares para asegurar la colaboración equitativa entre los miembros del equipo de Qualifast Buildings.

En cuanto a las herramientas tecnológicas, el desarrollo se realizará en Python utilizando el framework FLET, que nos permite crear una interfaz gráfica moderna y adaptable, cumpliendo con el requisito imprescindible de implementación. Este framework se integrará con el patrón Modelo-Vista-Controlador (MVC) para estructurar el código de manera clara, separando la lógica de negocio, la presentación y el control, lo que facilitará su mantenimiento y escalabilidad. Para el procesamiento de datos, emplearemos librerías como json para manejar los ficheros en formato JSON, asegurando que las rutas sean relativas al proyecto para garantizar su portabilidad, otro requisito clave.

Los dispositivos que utilizaremos incluyen la placa ESP32 como microcontrolador central, que conectará sensores como el DHT11/DHT22 (temperatura y humedad), MQ-2 (humo), MQ-135 (calidad de aire) y fotorresistores (LDR) para luz, junto con actuadores como LEDs, motores DC 12V y ventiladores 5V. También incorporaremos la cámara OV2640 para vigilancia. Estas herramientas se programarán con el entorno Arduino IDE para su integración con el software. Para el diseño y pruebas, usaremos PyCharm como IDE principal, que nos permitirá compilar y depurar el código de forma eficiente. Además, contaremos con seminarios y recursos digitales de la Universidad Europea de Madrid para reforzar nuestros conocimientos prácticos y teóricos.

Capítulo 5. Planificación del proyecto

La planificación se estructurará en cuatro sprints, siguiendo un enfoque SCRUM y alineándose con las historias de usuario del Product Backlog, adaptadas a los requisitos del proyecto para Qualifast Buildings. A continuación, se detalla un diagrama de Gantt simplificado y una descripción de las actividades, esfuerzos y fechas de entrega aproximadas.

- **Sprint 1 (27 de octubre - 9 de noviembre de 2025) - Effort: 160 horas**
 - **Enfoque principal:** Análisis, arquitectura, entorno, prototipos iniciales.
 - **Tarea:** Diseño de la interfaz gráfica inicial, módulo de login y vistas básicas.
 - **Hito:** Entrega del anteproyecto y diseño inicial de ventanas.
 - **Descripción:** Configuraremos la interfaz con FLET, implementando un login seguro adaptable a roles (comisario, policía, preso) y diseñando un plano básico con íconos para sensores (HU10) y una tabla de datos (HU12). Cada miembro dedicará 40 horas (10 a investigación del entorno IoT, 20 a desarrollo de la interfaz y login, 10 a revisión del diseño), totalizando 160 horas, cubriendo "Vista en modo plano de la comisaría" (HU10), "Interacción con el plano" (HU11), "Vista en modo tabla de datos" (HU12), y "Actualización automática de datos" (HU13).

- **Sprint 2 (10 de noviembre - 23 de noviembre de 2025) - Effort: 160 horas**
 - **Enfoque principal:** Desarrollo core en Python, integración base IoT.
 - **Tarea:** Integración de sensores y gestión de usuarios.
 - **Hito:** Prototipo funcional con lectura de datos y usuarios.
 - **Descripción:** Conectaremos sensores DHT11/DHT22, LDR, MQ-2, y MQ-135 a la ESP32 vía WiFi para lectura y visualización (HU10, HU1-5), y crearemos ficheros JSON con al menos 9 usuarios, 4 supervisores y 2 clínicos (HU15). Cada miembro aportará 40 horas (15 a integración de hardware, 15 a coding de usuarios y JSON, 10 a pruebas), sumando 160 horas, abarcando "Gestión de usuarios" (HU15), "Control de permisos por rol" (HU16), "Inicio de sesión con autenticación segura" (HU17), y "Registro de actividad de usuarios" (HU18).

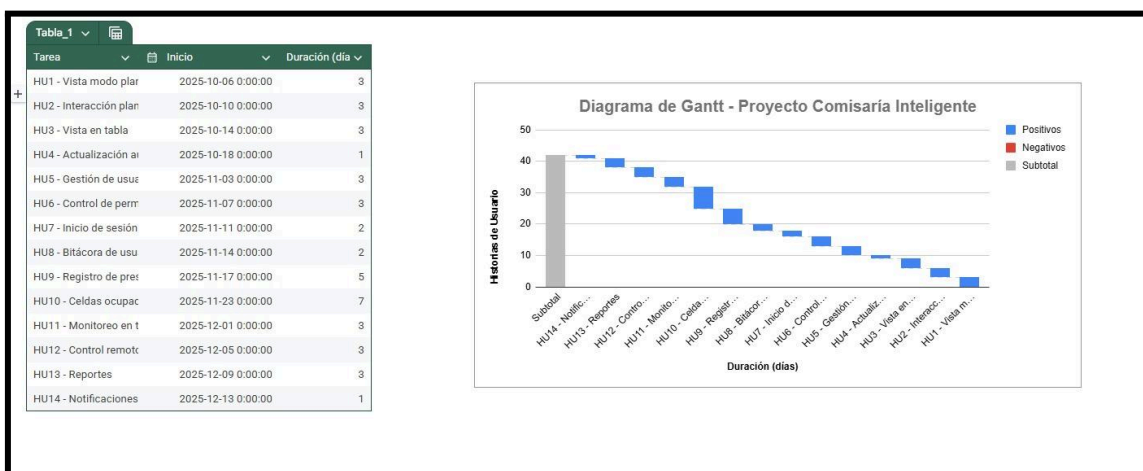
- **Sprint 3 (24 de noviembre - 30 de noviembre de 2025) - Effort: 80 horas**
 - **Enfoque principal:** UI/UX inicial, pruebas unitarias, ajustes rápidos.

- **Tarea:** Implementación de actuadores, responsividad y registro de presos.
 - **Hito:** Interfaz adaptable a diferentes resoluciones.
 - **Descripción:** Integraremos actuadores (LEDs, ventiladores, motores) vía PWM (HU10, HU11) y ajustaremos la interfaz para tablets y PCs. Además, implementaremos el módulo de registro de presos (HU20) y visualización de celdas (HU21). Con 2 miembros trabajando (20 horas cada uno: 10 desarrollo de actuadores y UI, 5 testing, 5 ajustes), totalizando 80 horas.
- **Sprint 4 (1 de diciembre - 14 de diciembre de 2025) - Effort: 160 horas**
 - **Enfoque principal:** Integración completa, pruebas de sistema, documentación.
 - **Tarea:** Finalización con panel de control, alertas y reportes.
 - **Hito:** Entrega final del proyecto.
 - **Descripción:** Implementaremos el panel de control (HU23), control remoto (HU24), generación de reportes (HU25), y notificaciones (HU26) con alertas visuales y sonoras. Cada miembro dedicará 40 horas (15 desarrollo de funcionalidades, 15 documentación, 10 presentación), sumando 160 horas.

Alineación con historias de usuario:

- **Sprint 1** cubre las bases de la interfaz y login (HU10, HU11, HU12, HU13).
- **Sprint 2** se enfoca en sensores y gestión de usuarios (HU1-5, HU15-18).
- **Sprint 3** integra actuadores y funcionalidades específicas (HU10, HU11, HU20, HU21).
- **Sprint 4** finaliza con control avanzado y documentación (HU23-26).

DIAGRAMA INCLUIDO EN EL PRODUCT BACKLOG



Capítulo 6. PRESUPUESTO

Estimación de horas

Sprint	Duración	Enfoque principal	Horas equipo
Sprint 1	2 semanas	Análisis, arquitectura, entorno, prototipos iniciales	160 h
Sprint 2	2 semanas	Desarrollo core en Python, integración base IoT	160 h
Sprint 3	1 semana	UI/UX inicial, pruebas unitarias, ajustes rápidos	80 h
Sprint 4	2 semanas	Integración completa, pruebas de sistema, documentación	160 h

Equipo técnico utilizado

Tipo de equipo	Valor estimado	Comentarios
Portátiles de desarrollo (4 uds.)	3.000 €	Valor de mercado (750 € c/u). No se adquieren, pero se valora su uso.
Router/WiFi para pruebas IoT	80 €	Red local para dispositivos ESP32 y pruebas de conectividad.

Software utilizado

Software	Coste	Comentarios
----------	-------	-------------

PyCharm (licencia gratuita)	0 €	Licencia gratuita para el equipo.
Python + librerías (DHT, requests, Flask/FastAPI, etc.)	0 €	Software libre.
Arduino IDE	0 €	Programación de ESP32/ESP32-CAM.
GitHub	0 €	Repositorio y control de versiones.

Material es empleados

Material	Cantidad	Valor estimado	Comentarios
Sensor de temperatura DHT22	2 uds.	20 €	Monitorización térmica en áreas críticas.
Fotorresistor (LDR) + resistencias	Pack	8 €	Control de iluminación y detección de oscuridad.
Sensor de humo MQ-2	2 uds.	24 €	Alarma contra incendios en zonas de riesgo.
Sensor calidad de aire MQ-135	2 uds.	30 €	Vigilancia de contaminantes en salas concurridas.
LEDs (pack 100 uds.)	1 pack	6 €	Indicadores de estado e iluminación auxiliar.
Motores DC 12V	5 uds.	50 €	Automatización de mecanismos y cierres.
Ventilador 5V (pequeño)	2 uds.	20 €	Disipación de calor en armarios de equipo.

Placa ESP32	2 uds.	40 €	Nodo IoT principal para integración de sensores.
ESP32-CAM (OV2640)	2 uds.	36 €	Vigilancia e imágenes en entradas/pasillos.
Drivers L298N	2 uds.	24 €	Control de motores DC vía PWM.
Fuentes de alimentación y convertidores	—	30 €	Adaptadores 5V/12V y step-down.
Cables, conectores, soportes y protoboard	—	50 €	Montaje, pruebas y fijación de sensores.

Resumen Final

Concepto	Detalle	Coste estimado (€)
Horas de trabajo	560 h × 15 €/h (planificación, desarrollo, integración IoT, pruebas, doc.)	8.400
Equipo técnico	4 portátiles (3.000 €), router WiFi (80 €), switch Ethernet (50 €), multímetro (50 €)	3.180
Software	PyCharm (licencia gratuita), Python, Arduino IDE, GitHub, Figma (gratuitos)	0
Materiales empleados	Sensores (DHT22, LDR, MQ-2, MQ-135), LEDs, motores DC, ventiladores, ESP32, ESP32-CAM, drivers, fuentes de alimentación, conectores y soportes	389
Subtotal		11.969

Capítulo 7. Equipo de trabajo

Álvaro López (Product Owner): Álvaro lidera el proyecto como Product Owner, definiendo las prioridades y asegurándose de que las historias de usuario del Product Backlog se alineen con los objetivos de Qualifast Buildings. Con conocimientos básicos en desarrollo Python y experiencia en el curso de Ingeniería Biomédica, destaca por su capacidad para coordinar al equipo y tomar decisiones estratégicas que guían el desarrollo del sistema IoT para la comisaría.

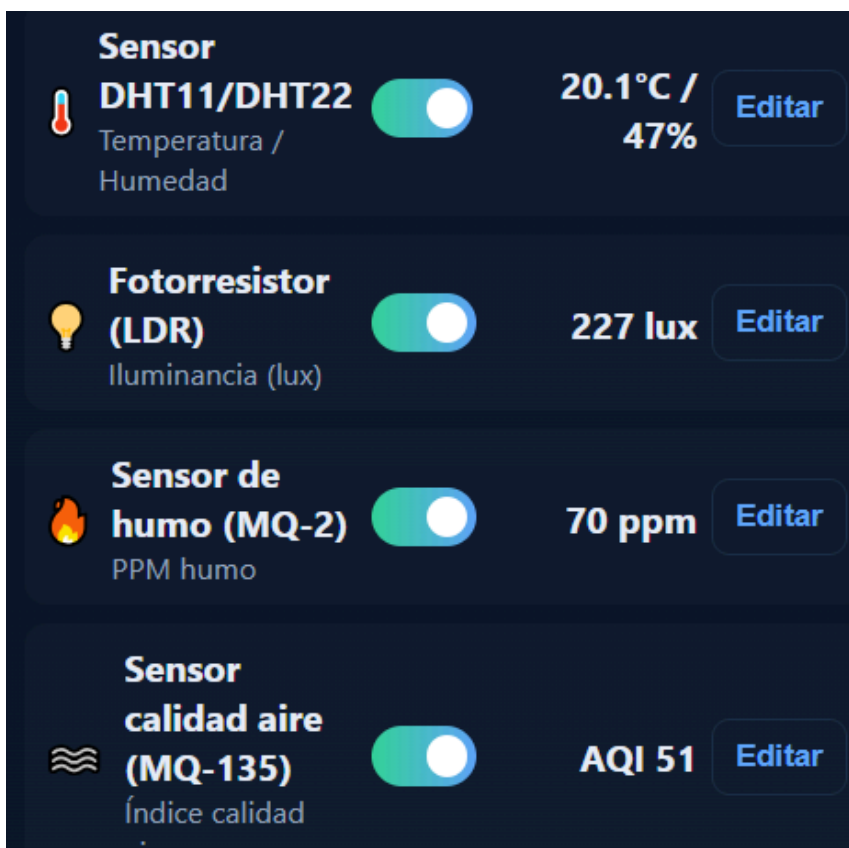
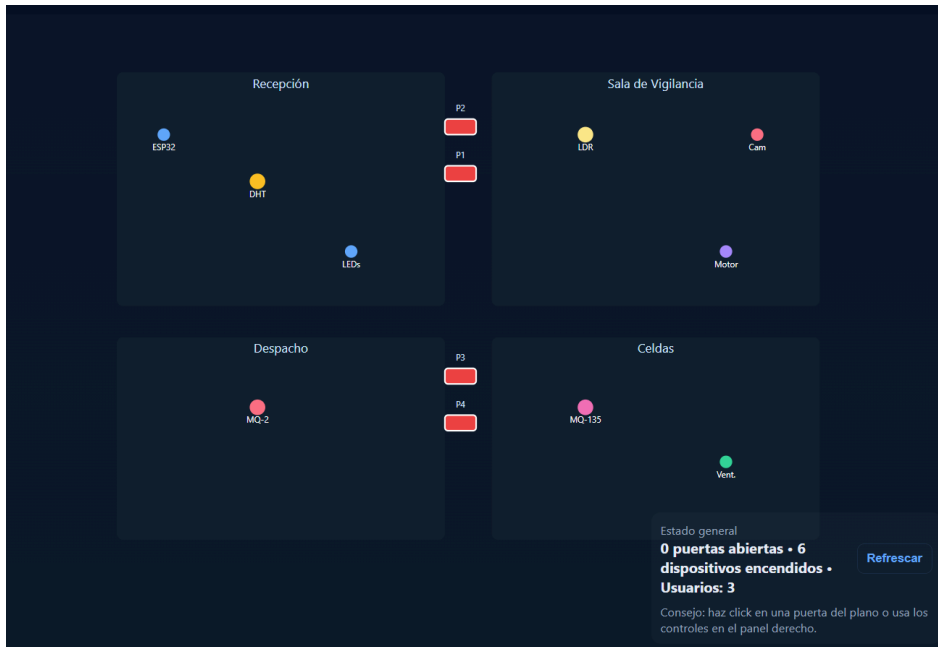
Nicolás Ramírez (Scrum Master): Nicolás actúa como Scrum Master, facilitando las reuniones y asegurando que el equipo siga la metodología SCRUM de forma efectiva. Con una base similar en programación Python y habilidades organizativas adquiridas en el curso, sobresale por su habilidad de planificación y conocimientos de GitHub debido a su experiencia profesional.

Fernando Fernández (Equipo Técnico): Fernando forma parte del equipo técnico, aportando sus conocimientos en desarrollo Python y manejo de hardware como la ESP32, adquiridos durante el curso. Su destreza para trabajar en equipo y su interés por la integración de sensores y actuadores lo hacen un pilar clave en la implementación del prototipo.

Daniel Vicente (Equipo Técnico): Daniel también pertenece al equipo técnico, con una base sólida en programación Python y experiencia en el curso que lo prepara para tareas como la creación de ficheros JSON. Destaca por su atención al detalle y su entusiasmo por probar nuevas soluciones, contribuyendo activamente al desarrollo del software.

Capítulo 8. Mockups de la aplicación

Link a la página interactiva: <https://qualifastbuildings.neocities.org/>



Registros de presos
Ver / crear según rol

Nuevo preso

Juan Pérez
Robo menor

27/10/2025, 18:51:04

Ver

Usuarios
Creación y gestión (solo comisario)

usuario

contraseña

comisario

comisario
comisario

inspector
inspector

policia
policia

Puertas
Puedes abrir/cerrar haciendo click en el plano o aquí

DOOR-1
Estado: Cerrada

Abrir

DOOR-2
Estado: Cerrada

Abrir

DOOR-3
Estado: Cerrada

Abrir

DOOR-4
Estado: Cerrada

Abrir

Acciones rápidas

Abrir todas

Cerrar todas

Reset
(configuración)

Los cambios se guardan en el navegador (localStorage).

REFERENCIAS Y BIBLIOGRAFÍA

En este apartado se recopilan las fuentes consultadas para el desarrollo del proyecto, incluyendo documentación técnica sobre sensores y actuadores, herramientas de desarrollo y metodologías ágiles. Se ha utilizado el estilo de citación ISO 690:2010, común en ingeniería para referencias en línea, priorizando accesibilidad y precisión en la descripción de recursos web y electrónicos.

[1] RANDOM NERD TUTORIALS. ESP32 with DHT11/DHT22 Temperature and Humidity Sensor using Arduino IDE [En línea]. 2019 [Citado: 26/10/2025]. Disponible en: <https://randomnerdtutorials.com/esp32-dht11-dht22-temperature-humidity-sensor-arduino-ide/>.

[2] INSTRUCTABLES. ESP32 Smoke Detector Project With MQ-2 Sensor [En línea]. 2025 [Citado: 26/10/2025]. Disponible en: <https://www.instructables.com/ESP32-Smoke-Detector-Project-With-MQ-2-Sensor/>.

[3] TEACH ME MICRO. MQ-135 Air Quality Sensor Tutorial [En línea]. 2023 [Citado: 26/10/2025]. Disponible en: <https://www.teachmemicro.com/mq-135-air-quality-sensor-tutorial/>.

[4] ESP32IO. ESP32 - Light Sensor [En línea]. 2023 [Citado: 26/10/2025]. Disponible en: <https://esp32io.com/tutorials/esp32-light-sensor>.

[5] FLASHGAMER. ESP32 and the OV2640 camera module [En línea]. 2023 [Citado: 26/10/2025]. Disponible en: <https://flashgamer.com/blog/comments/esp32-and-the-ov2640-camera-module>.

[6] RANDOM NERD TUTORIALS. ESP32 with DC Motor - Control Speed and Direction [En línea]. 2024 [Citado: 26/10/2025]. Disponible en: <https://randomnerdtutorials.com/esp32-dc-motor-l298n-motor-driver-control-speed-direction/>.

[7] DRONEBOT WORKSHOP. ESP32 PWM Fan Controller [En línea]. 2025 [Citado: 26/10/2025]. Disponible en: <https://dronebotworkshop.com/esp32-pwm-fan/>.

[8] ESPRESSIF SYSTEMS. LED Control (LEDC) - ESP32 [En línea]. 2023 [Citado: 26/10/2025]. Disponible en: <https://docs.espressif.com/projects/esp-idf/en/stable/esp32/api-reference/peripherals/ledc.html>.

[9] FLET. Introduction | Flet [En línea]. 2023 [Citado: 26/10/2025]. Disponible en: <https://flet.dev/docs/>.

[10] TRELL O. How to use Trello for scrum (and better teamwork) [En línea]. 2024 [Citado: 26/10/2025]. Disponible en:

<https://blog.trello.com/how-to-scrum-and-trello-for-teams-at-work>.

[11] GITHUB. Topics: python-projects [En línea]. 2023 [Citado: 26/10/2025]. Disponible en:

<https://github.com/topics/python-projects>.

[12] UNIVERSIDAD EUROPEA DE MADRID. Lista de requisitos que debe cumplir vuestro proyecto [En línea]. 2023 [Citado: 26/10/2025]. Disponible en: [Documento interno proporcionado en el campus virtual].

[13] AENOR. 2010. AEN/CTN 157 - PROYECTOS. Normas y Publicaciones. [En línea] 2010. [Citado el: 25 de abril de 2013.]

<http://www.aenor.es/aenor/normas/ctn/fichactn.asp?codigonorm=AEN/CTN%20157>.

[PÁGINA INTENCIONADAMENTE EN BLANCO]