

# **Ambiente Controlado PyME**

De la cruz Nahuel, Gonzalez Agustin Elias, Gomez Miño David  
41.063.583, 42.629.319, 41.130.037  
Lunes, Grupo 3

Universidad Nacional de La Matanza,  
Departamento de Ingeniería e Investigaciones Tecnológicas,  
Florencio Varela 1903 - San Justo, Argentina

## **Resumen.**

La aplicación Android presentada se integra con un sistema embebido utilizando MQTT para comunicar eventos y valores de sensores en tiempo real. Implementa suscripciones a tópicos específicos y maneja comandos hacia actuadores, como relé y buzzer. El diseño modular asegura escalabilidad y tolerancia a fallos, mientras que su interfaz simple garantiza facilidad de uso.

**Palabras clave:** MQTT, Android, Sistema Embebido, Suscripción, Tópicos

## **1 Introducción**

La presente aplicación móvil ha sido desarrollada como una herramienta de monitoreo y control remoto para sistemas embebidos, utilizando el protocolo de comunicación MQTT. Este protocolo, basado en una arquitectura de publicación y suscripción, garantiza una transmisión eficiente y en tiempo real entre los dispositivos involucrados.

La aplicación se integra con un sistema embebido compuesto por sensores y actuadores que reportan datos críticos y ejecutan acciones según las necesidades del usuario. Los sensores monitorean variables como temperatura, humedad, o estados específicos del sistema, mientras que los actuadores, como buzzer y relés, responden a comandos emitidos desde la aplicación.

El diseño se centra en proporcionar una experiencia de usuario intuitiva, permitiendo que la configuración y operación se realicen de forma simple y directa. Además, incluye funcionalidades clave como:

- Visualización en tiempo real de datos provenientes de sensores.
- Envío de comandos para controlar actuadores.

- Recepción de alertas en eventos críticos, permitiendo una respuesta rápida ante situaciones que requieran atención inmediata.

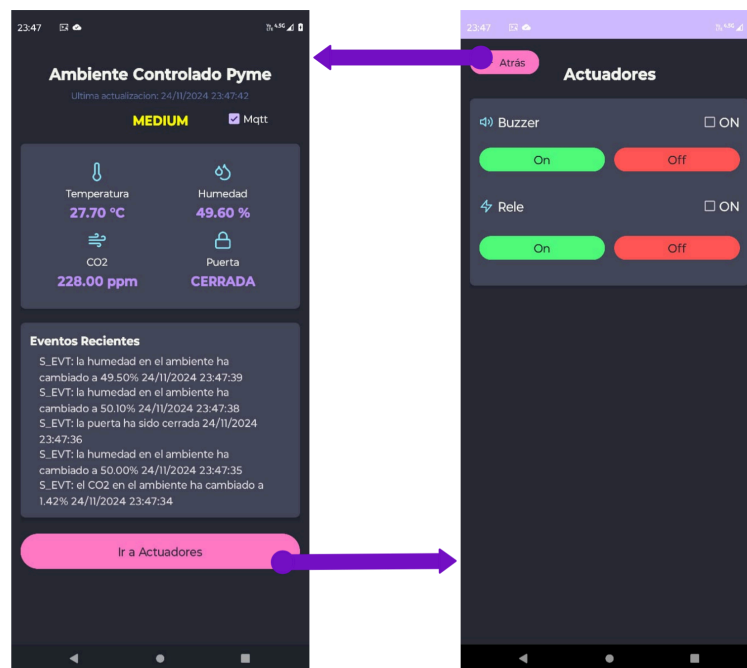
Esta aplicación está pensada para ser flexible y escalable, facilitando su adaptación a distintos entornos de sistemas embebidos. Su implementación asegura que el sistema opere de manera continua y confiable, ofreciendo al usuario una solución integral y efectiva para la gestión remota.

## 2 Desarrollo

### Repositorio GitHub y Wokwi/Tinkercad

- **GitHub:** [gdaunlam/SOA-2024-C2](https://github.com/gdaunlam/SOA-2024-C2)
- **Wokwi/Tinkercad:** <https://wokwi.com/projects/409054666641994753>

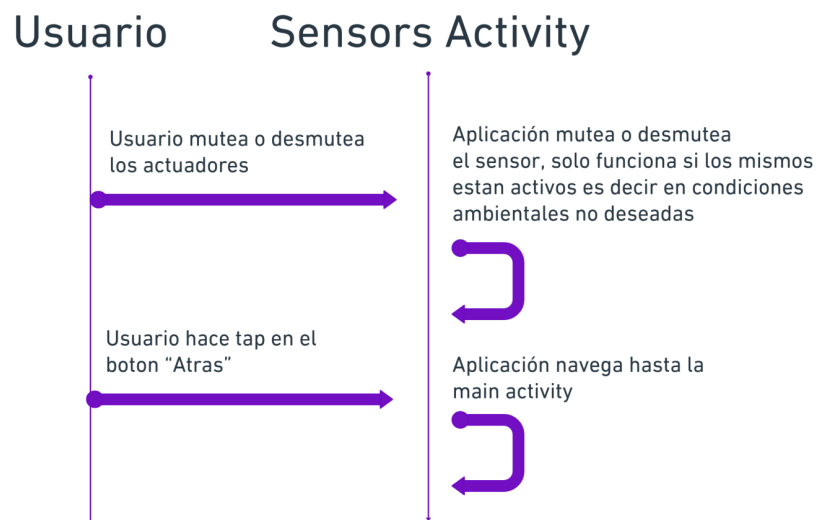
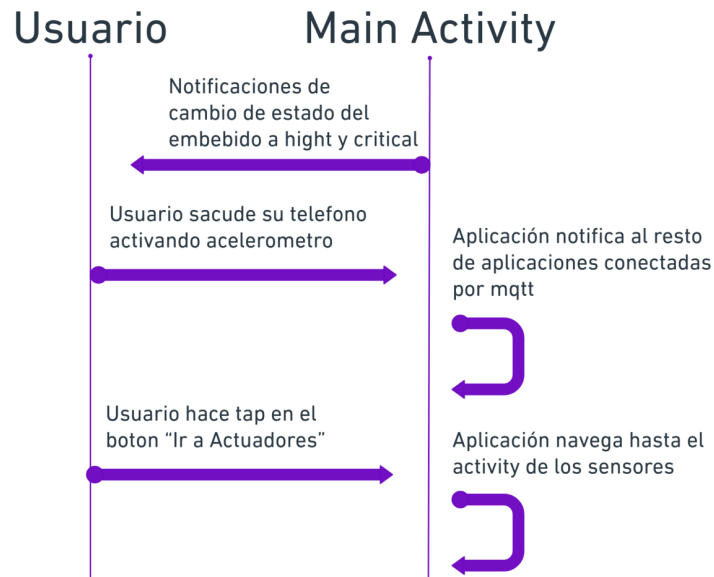
### Navegación de las Activities.



Main Activity

Sensors Activity

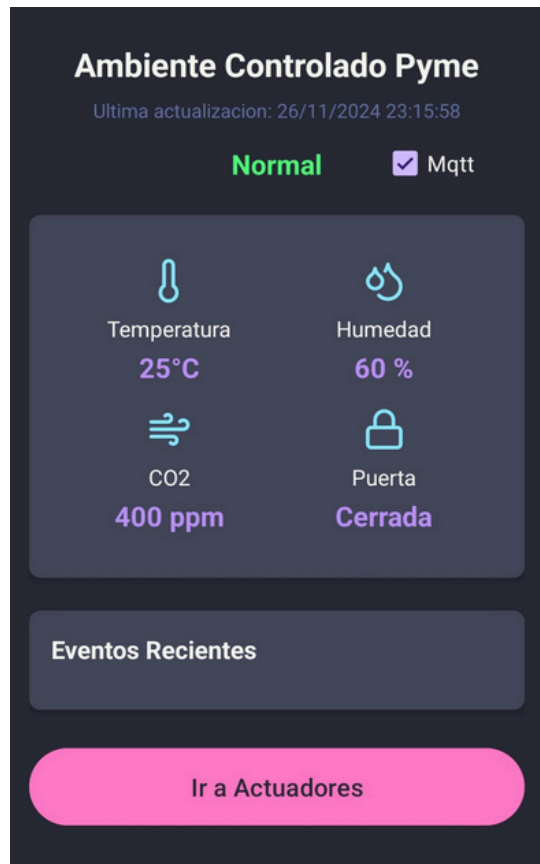
## Diagrama funcional



## Manual de Usuario de la Aplicación Android y cómo interactúa con el sistema embebido:

### Home

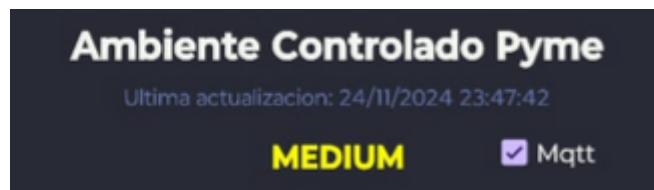
Al ingresar a la aplicación, nos vamos a encontrar con la página de inicio como la siguiente imagen:



A continuación, se explica cada uno de los elementos de esta pantalla:

- a. **Estado del sistema embebido** (Low, Mid, High, Critical). Cada estado tiene su propio color. En estado Critical, se envía una notificación de alerta a tu celular indicando el estado crítico del embebido para poder atender. Cada estado tiene su propio color. Se muestran de la siguiente manera:
  - i. LOW
  - ii. MID

- iii. **HIGH**
- iv. **CRITICAL**
- b. **Última actualización del embebido:** se muestra la última modificación del cambio de estado que ocurrió en el sistema embebido
- c. **Conexión con el servidor (MQTT):** el checkbox que está al lado derecho del estado del embebido indica si la aplicación está conectada o no al servidor del MQTT que está conectado con el embebido. No es un botón interactivo. Solamente muestra su estado. En la screenshot que se muestra a continuación, se muestra que la aplicación se encuentra conectada correctamente con el sistema.



Valores de los sensores del sistema embebido:

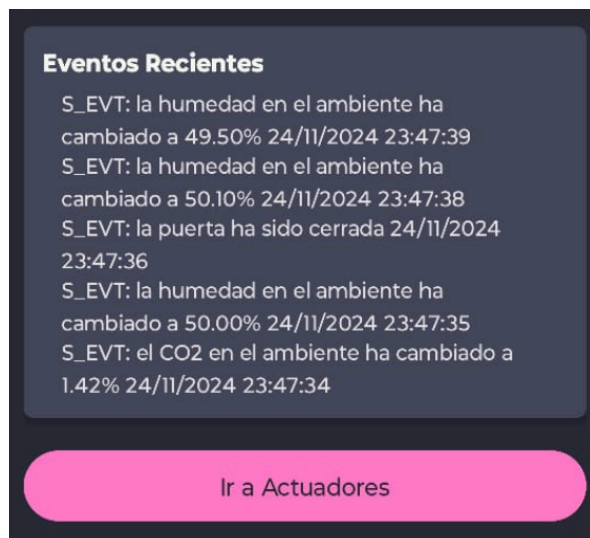


- **Temperatura del ambiente** (en °C) que registra el sensor de Temperatura y Humedad.
- **Humedad del ambiente** (%) que registra el sensor de Temperatura y Humedad.
- **CO2 o Dióxido de Carbono:** indica la cantidad de concentración de gases (en ppm) que registra el sensor de CO2.
- **Estado de la puerta:** Puede tener solamente 2 estados (CERRADA o ABIERTA). Este registro está relacionado con el sensor de distancia,

indicando si la puerta está abierta o cerrada en una tolerancia no más de 5 cm del sensor.

#### Log de los eventos:

Por último, se encuentra un cuadro que contiene un log de los distintos eventos que ocurrieron en el embebido que provocaron su cambio de estado desde el momento que se establece conexión con el sistema. Este cuadro sirve como un registro para que puedan controlar y seguir los distintos eventos por los que pasó el sistema y, además, se detalla el sensor con el valor por el cual provocó el cambio de estado.



En la captura, se detalla un ejemplo de un lote de registros de cambios de eventos, donde cada log tiene los siguientes datos:

- **El nombre del sensor y el nuevo valor encontrado.**
- **Fecha y hora del evento ocurrido.**

Por último, en la parte inferior de la pantalla, un botón para dirigirse a la pantalla donde se pueden activar y desactivar los actuadores del sistema.

#### Página de Actuadores



En esta página, se encuentran los botones para poder activar o desactivar los actuadores que posee el sistema:

- **Buzzer:** En caso de que sea necesario activar o desactivarlo. El buzzer se puede encontrar activado tanto si el embebedo se encuentra en estado Crítico como también si se detecta que la puerta fue abierta o no. También se encuentra un checkbox que indica si el buzzer se encuentra encendido o no.
- **Relé:** En caso de que sea necesario activar o desactivarlo. El relé es el encargado de encender el ventilador que va a mitigar y controlar la temperatura del ambiente. También se encuentra un checkbox que indica si el buzzer se encuentra encendido o no.
- **Botón Atrás:** un botón que te permite volver a la pantalla principal.

#### **Alerta de la aplicación integrada**

Por último, se va a explicar una funcionalidad adicional. En caso de que una persona encargada de controlar el ambiente PYME, encuentre el estado del sistema en peligro (High o Critical), existe una forma de que pueda alertar al resto del equipo de trabajo de manera rápida y sencilla.

Con sacudir el celular de izquierda a derecha o arriba hacia abajo de manera repetida, la aplicación reproduce una alarma particular en el celular.



Esta alarma se va a activar en todos los celulares que tengan la aplicación y estén conectados al servidor. De esa manera, todo el equipo de trabajo puede atender cualquier emergencia.

### Posibles escenarios

A continuación, se muestra una tabla con los posibles escenarios que se pueden presentar en cada pantalla.

Para la primera tabla, los escenarios que pueden ocurrir teniendo en cuenta los sensores que posee el embebido y el evento por el cuál llegó a tal escenario. También se encuentra una columna adicional con las medidas por el cuál ocurre cada evento.

Escenario	Eventos	Aclaracion de medidas
Se abre la ventana	en estado H y C, solamente cuando el CO2 es alto	CO >= 1000
Se cierra la ventana	en estado L y M, solamente cuando el CO2 es bajo	CO > 600 && CO < 1000 (MID) ; CO < 600 (LOW)
Se detecta puerta abierta	en estado M, solamente cuando el sensor de distancia detecta una separacion media entre la puerta y la pared	DISTANCIA > 5 cm
Se detecta puerta cerrada	en estado L, solamente cuando el sensor de distancia no detecta una separacion media entre la puerta y la pared	DISTANCIA <= 5 cm
Sube la temperatura	Sube de estado segun el valor	32 (MID) ; 35(HIGH) > 35 CRITICAL
Sube los valores de humedad	Sube de estado segun el valor	50 (LOW) ; 60 (MID) ; 70 (HIGH) ; >70 CRITICAL

En la siguiente tabla, se muestran los posibles escenarios que se pueden encontrar en los actuadores y cuál es el evento por el cuál llegó a tal escenario (FSM).

Escenario	FSM
Se enciende el buzzer	Solamente en CRITICAL
Mute buzzer / buzzer apagado	N/A
Mute buzzer / buzzer prendido	Buzzer activado pero muteado. (Si se desactiva el mute, vuelve a sonar)
Se enciende el rele	Solamente en HIGH y CRITICAL
Mute rele / rele apagado	N/A
Mute rele / rele prendido	Rele activado pero muteado. (Si se desactiva el mute, vuelve a encenderse)
Se enciende el servo	Cuando se abre la ventana (ver página de Mensajes de sensores)
Se apaga el servo	Cuando se cierra la ventana (ver página de Mensajes de sensores)

## 3 Conclusiones

- **Tolerancia a fallos:** Se implementó reconexión automática tras pérdida de conexión al broker MQTT y manejo de excepciones en tiempo real.



- **Problemas encontrados:** Inicialmente, la persistencia del cliente MQTT generaba errores al reconectar. Esto se resolvió mediante la configuración de sesiones limpias.
- **Lecciones aprendidas:** La importancia de diseñar una arquitectura robusta que facilite la recuperación ante fallos y una interacción clara para el usuario.

## 4 Referencias

1. Ing. Estaban A. Carnuccio, Ing. Mariano Volker, Ing. Raúl Villlca, Ing. Matías Adagio: Apunte Teórico sobre el Sistema Operativo Android (2024)