**Documentación: Proyecto NACHO\_LIBRE**

*(Versión 1.0 – 27/03/2025)*

**1. Descripción General**

Sistema embebido que automatiza el relleno de un comedero para cerdos mediante:

* **Hardware**: Raspberry Pi (unidad central) + ESP32 (control de sensores/actuadores).
* **Software**:
  + Backend en Python (Raspberry Pi).
  + Conexión con el servidor en PyQt6 (servidor)
  + App móvil en Flutter (control remoto).

**Objetivo principal**: Permitir al supervisor monitorear y activar el relleno del comedero de forma remota (Flutter).

**2. Diagrama de Funcionamiento**

A[App Flutter] -->|WiFi/4G| B(Raspberry Pi)

C[Interfaz PyQt6] --> B

B --> D[ESP32]

D --> E[Sensor de Nivel]

D --> F[Actuador/Motor]

* **Flujo**:
  1. El supervisor envía una orden (desde la app).
  2. La Raspberry Pi procesa la solicitud y envía comandos al ESP32.
  3. ESP32 activa el motor/servo para rellenar el comedero si el sensor indica bajo nivel.

**3. Componentes Clave**

**3.1 Hardware**

| Componente | Función |
| --- | --- |
| Raspberry Pi | Ejecuta el backend en Python, comunica app + ESP32. |
| ESP32 | Controla el sensor de nivel y el motor/actuador del comedero. |
| Sensor de nivel | Detecta cuándo el comedero está vacío. |
| Motor/Actuador | Abre/cierra el mecanismo de relleno |

**3.2 Software**

* **Backend (Python)**:
  + API REST (para la app Flutter).
  + Comunicación WIFI con ESP32.
* **PyQt6**:
  + Servidor conectado a un dominio.
* **App Flutter**:
  + Visualización del estado del comedero (lleno/vacío).
  + Botón de relleno remoto y notificaciones.

**4. Proceso de Relleno**

1. **Condición de activación**:
   * El sensor detecta nivel bajo **o** el supervisor presiona "Rellenar" (app).
2. **Secuencia**:
   * ESP32 recibe la señal y activa el motor durante X segundos (tiempo calibrado).
   * El sensor verifica el nuevo nivel y envía confirmación a la Raspberry Pi.
3. **Notificaciones**:
   * La app muestra: *"Comedero rellenado - Hora: [timestamp]"*.

**5. Protocolos de Seguridad**

* **Autenticación**: La app requiere usuario/contraseña.
* **Timeout**: Si el motor se activa por más de Y segundos, se fuerza su apagado (evita sobrecarga).
* **Registro de eventos**: Logs almacenados en la Raspberry Pi (últimos 30 días).

**6. Posibles Fallos y Soluciones**

| **Síntoma** | **Causa probable** | **Solución** |
| --- | --- | --- |
| Motor no se activa | Falta de conexión ESP32 | Verificar cables y puerto serial. |
| App no actualiza estado | WiFi/API caída | Reiniciar Raspberry Pi. |
| Falso "vacío" en sensor | Suciedad en sensor | Limpiar o recalibrar. |

**7. Notas de Mantenimiento**

* **Limpieza**: Verificar mensualmente el sensor y el mecanismo de relleno.
* **Actualizaciones**: La Raspberry Pi debe estar conectada a internet para recibir parches.