

# AviFeeder

Alejo Burnowicz, Juan Pablo Correa, Agustin Federico, Gonzalo Matellan, Nahuel Repetto

42646860, 40653000, 41882938, 43325268, 42024337  
Martes, Grupo 4

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

## Resumen.

Se trata de un dispositivo automatizado para alimentar y dar de beber a pájaros, que combina sensores y actuadores para garantizar que los recipientes de comida y agua se mantengan siempre en niveles adecuados. Un sensor de distancia por ultrasonido monitorea la cantidad de alimento en el comedero, activando un LED cuando el nivel es bajo. De manera similar, un sensor de nivel de agua detecta la cantidad disponible y enciende otro LED si requiere recarga. Además, una balanza mide con precisión el peso del alimento; si este desciende por debajo de un umbral preestablecido, un servo motor acciona una compuerta que dispensa más comida. Cuando el peso alcanza el nivel deseado, la compuerta se cierra automáticamente, evitando desperdicio. Gracias a esta combinación de sensores y actuadores, el sistema asegura alimentación e hidratación constante de forma autónoma, eficiente y precisa.

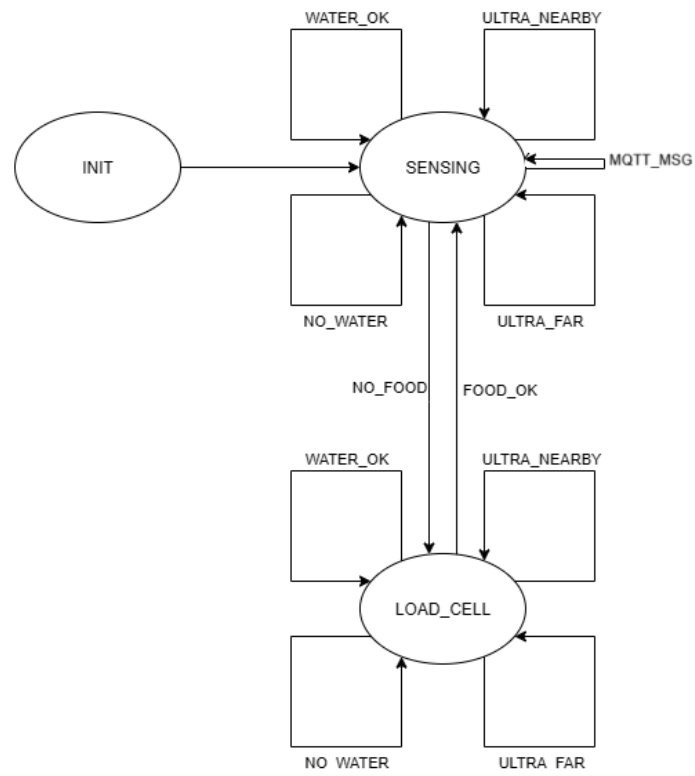
**Palabras claves:** Alimentador automatico, pajaros, sensores, actuadores, sistema embebido.

## 1 Introducción

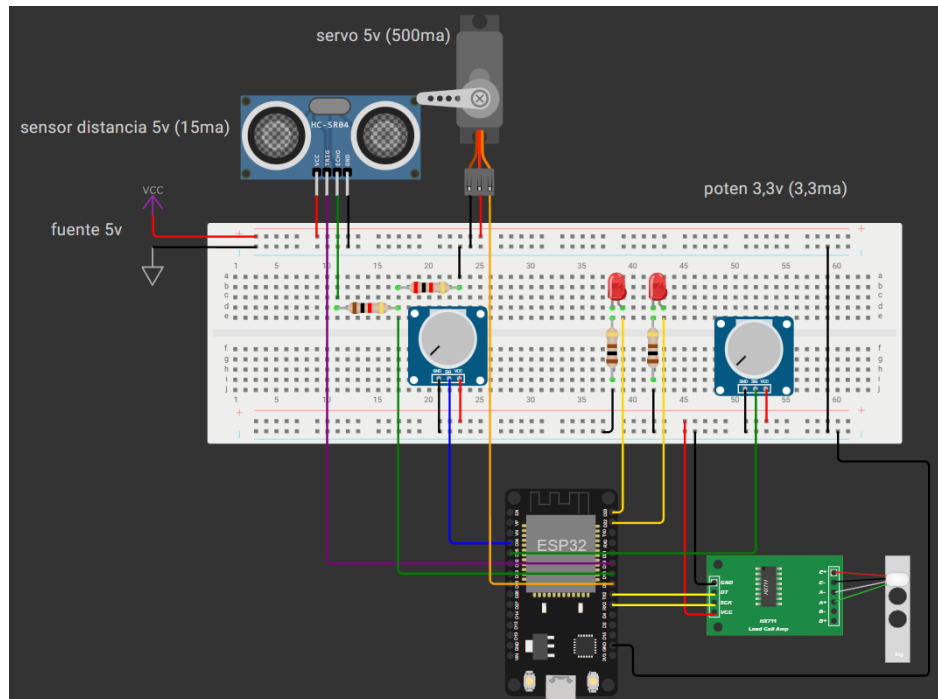
El sistema embebido diseñado para alimentar y dar de beber a pájaros automatiza tareas que normalmente requieren supervisión constante, proporcionando comodidad y eficiencia a los cuidadores de aves. Este dispositivo integra sensores y actuadores para monitorear y mantener niveles adecuados de alimento y agua. Gracias a un sensor de distancia por ultrasonido y un sensor de nivel de agua, el sistema detecta cuándo los recipientes están bajos y notifica mediante LEDs, evitando que las aves queden sin recursos esenciales. Además, una balanza integrada mide con precisión el peso del alimento y controla un servo motor que dispensa automáticamente la cantidad necesaria, cerrando la compuerta cuando se alcanza el nivel deseado. De esta manera, el sistema garantiza una alimentación e hidratación constante, minimiza desperdicios y permite un cuidado más eficiente y confiable de las aves, incluso cuando no se está presente de forma directa.

## 2 Desarrollo

### Diagrama de estados



## Diagrama de Conexiones del Circuito



[Enlace al Proyecto de Wokwi](#)

Enlace al repositorio de github del proyecto, donde se encuentran además imágenes y grabaciones de los componentes: [Repositorio AviFeeder](#).

## Manual de Usuario del Embebido Simulado

Este manual describe el uso y funcionamiento del prototipo simulado de una pajarera automatizada desarrollado en entorno virtual. El dispositivo fue diseñado para probar y validar la lógica de funcionamiento de un sistema embebido que asegura la alimentación e hidratación de aves de forma autónoma.

### 1. Encendido del Sistema

Encender virtualmente la placa ESP32, que gestiona toda la lógica del sistema (basta con presionar el botón “Start” para activar todo el circuito).

## 2. Una vez iniciada la simulación

### Sensor de Distancia Ultrasónico (HC-SR04)

- Simula la lectura del nivel de alimento en el comedero, se puedes modificar la distancia detectada manualmente desde el sensor.
- Si la distancia detectada es mayor al umbral configurado, se considera que el nivel de comida es bajo, y lo notifica encendido uno de los LEDs rojos.

### Potenciómetro

- Simular el valor del nivel de agua ya que en Wokwi no es posible simular un sensor de nivel de agua. Girar el potenciómetro para simular que varía el valor leído.
- Si el nivel es bajo (valor por debajo de cierto umbral), se enciende el otro LED rojo.

### Balanza Digital y Control del Servo Motor

- Se utiliza un módulo HX711 junto a una celda de carga simulada para medir peso (en gramos). Se puedes modificar el peso medido manualmente desde el módulo recién mencionado.
- Si el peso es menor al valor mínimo, el servo motor simulado se activa, simulando la apertura de una compuerta para recargar alimento.  
Cuando el peso alcanza cierto valor preestablecido, el servo vuelve a su posición inicial (compuerta cerrada).

### LEDs Indicadores

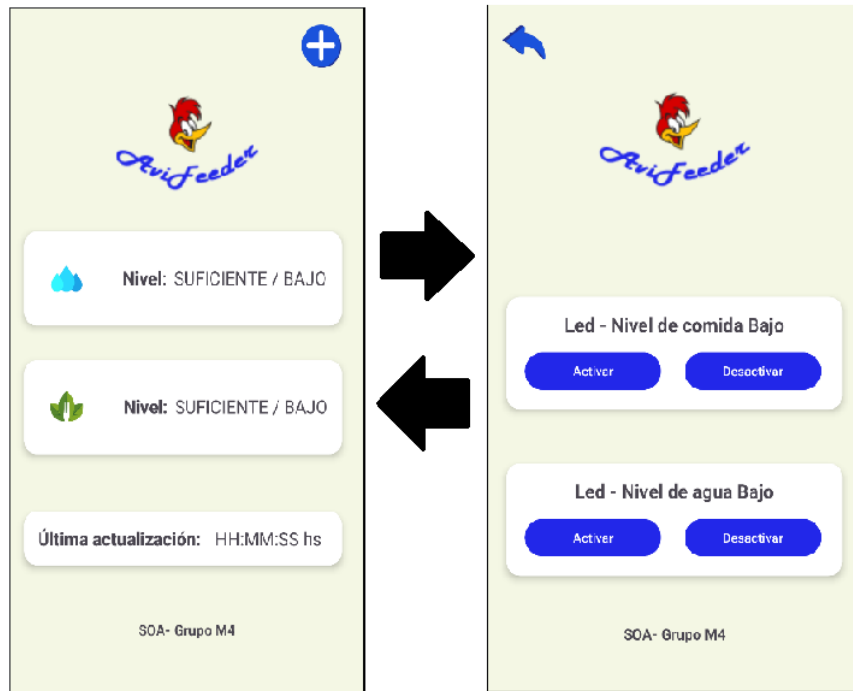
- Los LEDs cumplen función de indicadores visuales en la simulación, uno indica bajo nivel de comida (detectado por distancia) , y el otro bajo nivel de agua (simulado con potenciómetro)

## Navegación de las activities.

**MainActivity:** Pantalla principal que muestra en tiempo real el nivel de agua, nivel de comida y la última actualización recibida desde el dispositivo por MQTT. Desde esta pantalla también se puede acceder a la sección de actuadores. Se encarga de recibir los mensajes MQTT mediante un BroadcastReceiver y actualizar la interfaz.

**ActuadoresActivity:** Pantalla que permite enviar comandos al dispositivo físico mediante MQTT, como activar o desactivar los LEDs de comida y agua. Incluye detección de sacudidas mediante el acelerómetro para enviar un comando especial. También gestiona la conexión a internet mostrando un mensaje cuando no hay red disponible y se vincula internamente con el servicio MQTT para publicar comandos.

## Diagrama de navegación



## Manual de usuario de la aplicación

La aplicación fue desarrollada con el propósito de permitir al usuario monitorear de manera sencilla el estado del sistema de alimentación automático y controlar algunos de sus actuadores a distancia utilizando comunicación MQTT. Al iniciar la aplicación, el usuario es recibido por la pantalla principal, donde se muestran en tiempo real los datos que el sistema embebido envía de forma periódica. En esta vista se presentan tres valores fundamentales: el nivel de agua, el nivel de comida y el tiempo transcurrido desde la última actualización recibida. Tanto el nivel de agua como el nivel de comida se interpretan en categorías de “suficiente” o “insuficiente” a partir de los valores que proveen los sensores instalados en el dispositivo físico; mientras que el tiempo de actualización se muestra en un formato claro (HH:MM:SS), permitiendo al usuario verificar que la comunicación con el sistema embebido se mantiene activa. Todos estos datos se actualizan automáticamente y no requieren ninguna intervención del usuario.

Desde esta pantalla principal es posible acceder a la sección de actuadores mediante un botón destinado específicamente a la navegación hacia dicha área. Al ingresar a la pantalla de actuadores, el usuario obtiene la capacidad de interactuar activamente con el sistema embebido enviando comandos directos. Para ello, la aplicación dispone de botones que permiten activar o desactivar los LEDs asociados a los depósitos de comida y de agua. Al presionar cualquiera de estos controles, la aplicación envía el comando correspondiente hacia el sistema embebido utilizando el servicio MQTT en segundo plano, lo que permite que la respuesta del dispositivo físico sea prácticamente inmediata. Además del control manual a través de botones, la aplicación incorpora una funcionalidad especial basada en el sensor acelerómetro del teléfono: cuando se detecta una sacudida del dispositivo, se envía automáticamente un comando adicional denominado LED\_SHAKER, lo cual brinda una forma alternativa de interacción definida previamente en el firmware del sistema embebido.

La aplicación también considera el estado de conectividad del usuario. En caso de perderse la conexión a Internet, se muestra un mensaje persistente que advierte sobre la imposibilidad de comunicarse con el sistema embebido, el cual desaparece automáticamente cuando la conexión se restablece. Desde la misma pantalla de actuadores, el usuario puede regresar fácilmente a la pantalla principal mediante el botón de navegación correspondiente.

El funcionamiento general de la aplicación depende de un servicio MQTT que se inicia automáticamente al abrirla, encargándose de establecer y mantener la conexión con el broker. A través de este mecanismo, la aplicación recibe los datos generados por el sistema embebido y, de forma inversa, transmite los comandos que el usuario decide enviar. Todo el intercambio de información ocurre sin necesidad de que el usuario realice configuraciones adicionales, lo que simplifica su uso y garantiza una experiencia intuitiva.

### **3 Conclusión**

El desarrollo del proyecto AviFeeder representó una experiencia completa en el diseño e integración de sistemas embebidos, abarcando tanto aspectos lógicos como de implementación física. Uno de los primeros desafíos fue la correcta definición de la máquina de estados, lo cual requirió analizar en detalle los eventos, transiciones y condiciones necesarias para asegurar un funcionamiento coherente y libre de comportamientos indeseados. Posteriormente, otro reto importante fue la incorporación de la comunicación MQTT, que implicó comprender su dinámica, configurar tópicos apropiados y, especialmente, encontrar un broker público estable para realizar pruebas confiables durante todo el proceso.

En paralelo, debimos trabajar en el desarrollo de la aplicación Android, un entorno totalmente nuevo para el equipo. Esto implicó aprender su estructura, manejar la comunicación con el servicio MQTT y diseñar una interfaz capaz de mostrar información en tiempo real y enviar comandos al sistema embebido. Esta etapa amplió significativamente el alcance del proyecto al permitir control y monitoreo remoto.

A nivel hardware, la construcción de la pajarera añadió un conjunto adicional de desafíos prácticos: fue necesario planificar cuidadosamente la ubicación de cada sensor, actuador y cableado para garantizar lecturas precisas, un flujo correcto de los componentes y un montaje seguro y funcional. A ello se sumó un imprevisto importante: la rotura de la celda de carga original. Para mantener la continuidad del trabajo y poder validar la lógica del sistema, se optó por reemplazarla por un potenciómetro que simula el peso, permitiendo completar las pruebas previstas.

En conjunto, el proyecto no solo alcanzó los objetivos planteados, sino que también proporcionó un aprendizaje significativo en diseño, integración multidisciplinaria, resolución de problemas y adaptación ante imprevistos, fortaleciendo competencias esenciales para el desarrollo de soluciones embebidas reales.

## 4 Referencias

1. Wokwi, "Wokwi Docs," 2025. [En línea]. Disponible en: <https://docs.wokwi.com/>
2. SOA UNLaM, "Sistemas embebidos e Internet de las Cosas," 2025. [En línea]. Disponible en: [https://www.soa-unlam.com.ar/wiki/index.php/PUBLICO:Sistemas\\_embebidos\\_e\\_Internet\\_de\\_las\\_Cosas](https://www.soa-unlam.com.ar/wiki/index.php/PUBLICO:Sistemas_embebidos_e_Internet_de_las_Cosas)
3. Naylamp Mechatronics, "Tutorial: Transmisor de celda de carga HX711 – Balanza digital," 2025. [En línea]. Disponible en: [https://naylampmechatronics.com/blog/25\\_tutorial-trasmisor-de-celda-de-carga-hx711-balanza-digital.html](https://naylampmechatronics.com/blog/25_tutorial-trasmisor-de-celda-de-carga-hx711-balanza-digital.html)