

Diseño e implementación de un sistema de adquisición, transmisión y visualización de datos basado en CanSat

Sergio García Sánchez — Máster en Ingeniería Informática (UCM)

Director: Adrián Riesco Rodríguez

Agenda

1. Contexto y motivación
2. Objetivos
3. Trabajo relacionado & fundamentos
4. Diseño y arquitectura
5. Implementación (embebido, backend, frontend)
6. Validación y resultados
7. Conclusiones y trabajo futuro
8. Demo / Q&A

Contexto

- ¿Qué es un **CanSat**?
- Uso educativo / divulgativo
- Necesidad de **visualización en tiempo real** en proyectos educativos

Idea clave: Muchos proyectos priorizan la electrónica; la visualización suele ser ad-hoc y poco reusable.

Motivación

- Falta de una **plataforma web** reutilizable e independiente del hardware
- Necesidad de **modularidad**: adquisición, transmisión, backend y frontend desacoplados
- Facilitar pruebas, análisis y **reutilización** en otros proyectos

Objetivos

- Diseñar y construir un **prototipo CanSat** de validación
- Desarrollar una **plataforma web modular** para telemetría, mapas, orientación 3D y vídeo
- Asegurar **independencia del hardware** y despliegue sencillo (contenedores)

Trabajo relacionado (resumen)

- Herramientas genéricas (SerialPlot, Excel, etc.) vs. plataformas complejas (LabVIEW)
- Hueco: solución **ligera, web y reusable** para telemetría educativa
- Decisiones de diseño derivadas del estado del arte

Fundamentos técnicos (visión general)

- MCUs / SBCs: **Raspberry Pi Zero 2**, ESP32, Arduino (criterios de selección)
- Buses: **I²C**, **UART**, **SPI** (para sensores y radio)
- RF: **LoRa** vs. WiFi (alcance/latencia/consumo)
- Tiempo real web: **RabbitMQ**, **WebSocket**

Diseño y Arquitectura

Arquitectura general

- **Adquisición** (Raspberry Pi): sensores + GNSS + vídeo
- **Transmisión**: WiFi / LoRa
- **Backend**: Ingesta → Persistencia (DB) → **WebSocket**
- **Frontend**: métricas, gráficos en vivo, mapa, actitud 3D, vídeo

[Coloca aquí un diagrama de cajas simple de la arquitectura]

```
! [Arquitectura] (./figuras/arquitectura.png)
```

Hardware del CanSat (resumen)

- Raspberry Pi Zero 2 W
- Sensores: **barómetro**, IMU, GNSS
- **LoRa** (estación de tierra receptora)
- Cámara CSI (vídeo en tiempo real)

[Inserta fotos o esquemas de conexión]

! [Montaje] (./figuras/montaje_frontal.jpg)

! [Diagrama] (./figuras/diagrama_conexiones.png)

Software embebido (Raspberry Pi)

- Lectura de sensores (I²C / UART)
- **Publicación de eventos** → RabbitMQ (cuando hay red)
- **LoRa** como respaldo de telemetría
- Codificación y envío de **vídeo** (pipeline FFmpeg/RTMP)

[Snippet del formato de evento JSON, sin secretos]

```
{  
  "datetime": "2025-05-05T12:34:56Z",  
  "payload": { "altitude": 123.4, "yaw": 10.2, "lat": 40.4, "lon": -3.7, "...": "..." }  
}
```