Modernización de contadores de tránsito con comunicación bidireccional

Autor:
Ing. Diego Aníbal Vázquez

Director:
- (-)

Esta planificación fue realizada en el curso de Gestión de proyectos entre el 29 de abril de 2025 y el 17 de junio de 2025.

Índice

Registros de cambios

Revisión	Detalles de los cambios realizados	Fecha
0	Creación del documento	29 de abril de 2025
1	Se completa hasta el punto 5 inclusive	13 de mayo de 2025
2	Se completa hasta el punto 9 inclusive	20 de mayo de 2025
3	Se completa hasta el punto 12 inclusive	27 de mayo de 2025
4	Se completa el plan	3 de junio de 2025

Acta de constitución del proyecto

Buenos Aires, 29 de abril de 2025

Por medio de la presente se acuerda con el Ing. Diego Aníbal Vázquez que su Trabajo Final de la Carrera de Especialización en Internet de las Cosas se titulará "Modernización de contadores de tránsito con comunicación bidireccional" y consistirá en un sistema de comunicación con los contadores de tránsito, que incorpore un modelo bidireccional. El trabajo tendrá un presupuesto preliminar estimado de 600 horas y un costo estimado de $ 20.720.000, con fecha de inicio el 29 de abril de 2025 y fecha de presentación pública en marzo de 2026. Se adjunta a esta acta la planificación inicial.

Dr. Ing. Ariel Lutenberg
Director posgrado FIUBA

Subgerencia de Estudios de Demanda
Vialidad Nacional

-
Director del Trabajo Final

1. Descripción técnica-conceptual del proyecto a realizar

1.1. Contexto y motivación

Este proyecto surge a partir de una necesidad detectada en la infraestructura actual de monitoreo del tránsito vehicular utilizada en rutas nacionales.

Actualmente, uno de los modelos de contadores de tránsito empleados ha sido desarrollado internamente y cumple adecuadamente su función básica: registrar el paso de vehículos, clasificarlos por carril en livianos y pesados y transmitir los datos a un servidor central.

Sin embargo, estas unidades se comunican exclusivamente mediante enlaces GPRS tercerizados a través de un canal unidireccional. Esto impide cualquier tipo de interacción remota con los dispositivos en campo.

Frente a esta situación, se plantea el desafío de modernizar la arquitectura de comunicaciones del sistema, incorporando capacidades de comunicación bidireccional, diagnóstico remoto y respuesta operativa rápida.

1.2. Problemas identificados

- Falta de comunicación bidireccional: actualmente, no es posible enviar comandos desde el servidor a los dispositivos para ajustar su configuración, reiniciarlos o recolectar información de diagnóstico.
- Dependencia de proveedores externos: la infraestructura GPRS utilizada es tercerizada, lo que genera costos recurrentes, posibles restricciones técnicas y dificultades para gestionar incidentes de manera eficiente.
- Imposibilidad de actualización remota: cualquier modificación de parámetros de funcionamiento requiere intervención física en el dispositivo, lo que limita la flexibilidad y agilidad operativa.

1.3. Estado del arte y propuesta de valor

En el mercado existen diversas soluciones comerciales que ofrecen capacidades de gestión remota y comunicación bidireccional. Sin embargo, muchas de ellas resultan costosas.

El enfoque propuesto busca aprovechar tecnologías de código abierto y protocolos estandarizados (MQTT), con el objetivo de construir una alternativa flexible, escalable y económicamente viable adaptada al entorno específico de las rutas argentinas.

1.4. Propuesta de modernización

Se propone rediseñar el sistema de comunicaciones de los contadores de tránsito mediante la incorporación de un modelo de comunicación bidireccional y segura. Este nuevo esquema permitirá no solo el envío de datos desde los dispositivos hacia el servidor central, sino también

la recepción de comandos y actualizaciones de parámetros desde el servidor hacia los dispositivos en el campo. Además, se prevé la visualización de los datos en tiempo real conforme se transmiten.

1.5. Descripción funcional y técnica

El sistema estará compuesto por un dispositivo contador con capacidad de comunicación bidireccional mediante GPRS, que utilice el protocolo MQTT para el envío y recepción de datos. La información capturada será almacenada en una base de datos relacional y visualizada a través de una interfaz web básica alojada en el servidor central. Esta interfaz permitirá el monitoreo en tiempo real así como el envío de comandos remotos, se incluirán funciones de monitoreo de temperatura, nivel de batería y detección de errores de hardware o comunicación.

1.6. Grado de innovación

La innovación del proyecto radica en la integración de elementos existentes , como sensores y redes móviles, bajo una arquitectura abierta y centralizada, adaptable y orientada a la gestión inteligente de los datos de tránsito, con un fuerte énfasis en la autonomía y flexibilidad operativa.

En la figura 1 se presenta el diagrama en bloques del sistema. El dispositivo de conteo, compuesto por sensores y un microcontrolador, se comunica mediante un canal inalámbrico GPRS y transmite los datos hacia un broker MQTT externo. El servidor central está suscrito a este broker y recibe los datos mediante un módulo de procesamiento que los almacena en una base de datos relacional. Paralelamente, el sistema cuenta con una API REST que permite el acceso a la información desde una interfaz web y la ejecución de comandos administrativos. De esta manera, el sistema combina la eficiencia del protocolo MQTT para el envío de datos en tiempo real con la flexibilidad de una API REST para la gestión y visualización.

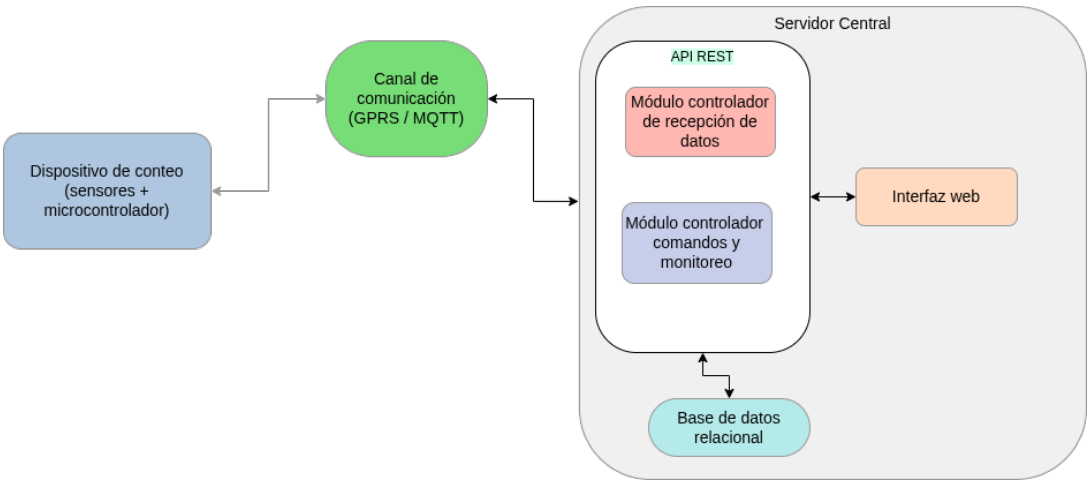

Figura 1. Diagrama en bloques del sistema.

2. Identificación y análisis de los interesados

Rol	Nombre y apellido	Organización	Puesto
Auspiciante	Vialidad Nacional	-	-
Cliente	Subgerencia de Estudios de Demanda	Vialidad Nacional	-
Impulsor	Ing. Rogelio Diego González	Vialidad Nacional	Subgerente de Estudios de Demanda
Responsable	Ing. Diego Aníbal Vázquez	FIUBA	Alumno
Colaboradora	Ing. María Clara Cutrone	Vialidad Nacional	Jefa Sección Análisis Tránsito
Colaborador	Ing. Alejandro Di Rosso	Vialidad Nacional	Jefe Sección Censos de Carga
Orientador	-	-	Director del Trabajo Final
Opositores	ATSA	-	-
Usuario final	Subgerencia de Estudios de Demanda	Vialidad Nacional	-
Usuario final	Gerencia Ejecutiva de Proyectos y Obras	Vialidad Nacional	-
Usuario final	Consultoras Externas	-	-

Cuadro 1. Identificación de los interesados.

Rol	Nombre y apellido	Observaciones
Auspiciante	Vialidad Nacional	Es exigente con la rendición de gastos. Se deberá tener especial cuidado en este aspecto.
Cliente	Subgerencia de Estudios de Demanda	Valora especialmente la autonomía operativa, la posibilidad de diagnóstico remoto y la reducción de costos operativos. No hay condiciones especiales relacionadas con la propiedad intelectual ni con la confidencialidad.
Colaboradora	Ing. María Clara Cuttrone	Suele pedir licencia debido a una familia extensa. La planificación debe considerar esta situación.

Cuadro 2. Participantes y consideraciones relevantes

.

3. Propósito del proyecto

Diseñar e implementar un sistema capaz de registrar eventos de tránsito y permitir la transmisión segura de los datos recolectados, incluso ante cortes de conexión. Además, se busca incorporar capacidades de comunicación bidireccional que habiliten diagnósticos remotos y la carga de

actualizaciones de parámetros, así como una mejor respuesta ante fallas. Con ello, se pretende mejorar la calidad de la información recolectada en rutas nacionales y facilitar el trabajo de los equipos técnicos responsables del monitoreo.

4. Alcance del proyecto

Se desarrollará un prototipo funcional que permita validar los aspectos clave del rediseño propuesto. El prototipo incluirá:

- Un dispositivo contador actualmente existente, pero incompleto, al que se le incorporará la capacidad de comunicación bidireccional.
- Un canal de comunicación GPRS.
- Implementación de un protocolo seguro (MQTT) para el envío y recepción de datos y comandos.
- Una interfaz básica en el servidor central para la visualización de datos en tiempo real y el envío de instrucciones al dispositivo.
- Funciones elementales de monitoreo remoto.
- Base de datos relacional en el servidor central para el almacenamiento estructurado de los datos recibidos desde los dispositivos de campo.

El presente proyecto no incluye el desarrollo ni la implementación de algoritmos de inteligencia artificial para estimar o reconstruir tránsitos no detectados.

5. Supuestos del proyecto

Para el desarrollo del presente proyecto se supone que:

- Se dispone de acceso a la infraestructura actual de los contadores de tránsito ya instalados, así como de la documentación técnica necesaria para su análisis y eventual integración.
- Se cuenta con conectividad intermitente por GPRS en los sitios donde se instalará el sistema, lo que permitirá validar el funcionamiento del envío diferido de datos.
- Los recursos humanos involucrados estarán disponibles durante la duración del proyecto en los tiempos y roles previstos.
- Los materiales requeridos (hardware de prueba, sensores, dispositivos de comunicación, etc.), estarán disponibles o serán reemplazables por equivalentes funcionales en caso de faltantes.
- La incorporación de comunicación bidireccional es técnicamente factible y puede realizarse sin rediseñar completamente el hardware existente.

6. Requerimientos

1. Requerimientos funcionales:
   - 1.1. El ESP32-C3 debe recibir datos por interfaz RS-232 desde el sistema de detección.
   - 1.2. Cada evento recibido debe ser encolado en memoria RAM según el orden de llegada.
   - 1.3. El ESP32-C3 debe publicar cada mensaje de la cola a un *broker* MQTT remoto usando GPRS.
   - 1.4. El protocolo MQTT debe utilizar QoS 1 o 2 para asegurar la entrega sin duplicación.
   - 1.5. Debe haber control de reintentos ante fallos de conexión sin duplicar mensajes.
   - 1.6. Si no hay conectividad GPRS disponible, los mensajes deben permanecer en la cola en memoria.
   - 1.7. Al llenarse la cola, los mensajes nuevos pueden descartarse (política FIFO).
   - 1.8. El ESP32-C3 debe suscribirse a un *topic* MQTT para recibir comandos desde el servidor.
   - 1.9. Al recibir un comando válido, el ESP32-C3 debe ejecutar una acción responder OK, error o devolver con el estado solicitado por el comando.
   - 1.10. La API REST debe suscribirse al mismo *broker* MQTT y recibir todos los eventos publicados.
   - 1.11. La API REST debe poder enviar comandos al ESP32-C3 publicando en el *topic* correspondiente.
   - 1.12. La interfaz web debe permitir visualizar eventos de tránsito recibidos desde la API REST.
   - 1.13. La interfaz web debe permitir enviar comandos al ESP32-C3 a través de la API REST y posibilitar ver el estado.
2. Requerimientos de documentación:
   - 2.1. Documentar la estructura de los mensajes RS-232 esperados (formato, delimitadores).
   - 2.2. Especificar el topic MQTT usado y los parámetros de conexión (broker, puerto, QoS).
   - 2.3. Incluir diagrama de bloques del flujo de eventos: contador → RS-232 → cola → MQTT → API REST.
   - 2.4. Incluir diagrama de bloques del flujo de eventos: API REST → MQTT → RS-232 → contador.
   - 2.5. Documentar los comandos remotos disponibles y el formato de sus respuestas.
   - 2.6. Documentar la estructura de los *endpoints* REST y sus respuestas.
3. Requerimientos de testing:
   - 3.1. Probar pérdida de conectividad GPRS y reenvío automático posterior.
   - 3.2. Verificar que no haya duplicación ni pérdida de eventos con distintos volúmenes de tráfico.
   - 3.3. Verificar saturación de la cola y descarte correcto de eventos.
   - 3.4. Probar recepción de comandos desde el servidor y respuesta correcta.
   - 3.5. Verificar que la API REST consuma correctamente los eventos desde MQTT.
   - 3.6. Probar que la interfaz web reciba eventos en tiempo real o los consulte en el backend.

4. Requerimientos de interfaz:
   4.1. La interfaz web debe ser accesible desde cualquier navegador moderno.
   4.2. La interfaz debe tener una tabla o lista para visualizar los eventos de tránsito.
   4.3. La interfaz debe permitir enviar comandos a dispositivos mediante un formulario o botón.
5. Requerimientos de interoperabilidad:
   5.1. El sistema debe poder comunicarse con un *broker* MQTT externo configurable.
   5.2. Responder al protocolo MQTT según lo acordado en el backend.
6. Requerimientos de seguridad:
   6.1. La conexión MQTT debe incluir autenticación por credenciales.
   6.2. La API REST debe requerir autenticación para el envío de comandos o consulta de datos.

7. Historias de usuarios (*Product backlog*)

Criterio para estimación de *story points*: cada historia de usuario se evalúa en tres dimensiones usando la secuencia de Fibonacci: 1, 2, 3, 5, 8, 13, 21. La suma de dificultad, complejidad y riesgo se redondea al número de Fibonacci más cercano.

1. Como usuario del sistema
   quiero ver los eventos de tránsito en una interfaz web
   para tener monitoreo visual y registro histórico.
   Story points: 8 (complejidad: 3, dificultad: 2, incertidumbre: 1)
   Prioridad: media
2. Como administrador del sistema
   quiero que los eventos se envíen al servidor vía MQTT sobre GPRS
   para recibir todos los tránsitos detectados en tiempo real o diferido.
   Story points: 8 (complejidad: 3, dificultad: 2, incertidumbre: 3)
   Prioridad: alta
3. Como desarrollador backend
   quiero que la API REST se suscriba al *broker* MQTT
   para recibir y procesar los eventos de tránsito.
   Story points: 13 (complejidad: 3, dificultad: 3, incertidumbre: 3)
   Prioridad: alta
4. Como administrador desde la interfaz web
   quiero enviar comandos al dispositivo a través de la API REST
   para controlar remotamente los dispositivos en campo.
   Story points: 8 (complejidad: 3, dificultad: 2, incertidumbre: 3)
   Prioridad: alta
5. Como usuario del sistema
   quiero usar la interfaz web para enviar comandos
   para realizar chequeos o pruebas de funcionamiento a distancia.
   Story points: 5 (complejidad: 2, dificultad: 2, incertidumbre: 1)
   Prioridad: media

6. Como desarrollador del sistema,
   quiero que el ESP32-C3 registre cada evento de tránsito recibido y respuesta a comandos por RS-232 en una cola en memoria RAM,
   para asegurar que los eventos no se pierdan si no hay conexión GPRS disponible en ese momento.
   Story points: 13 (complejidad: 5, dificultad: 2, incertidumbre: 3)
   Prioridad: alta

7. Como desarrollador del sistema
   quiero modificar el contador de tránsito en el ESP32-C3 con el objetivo de una correcta interpretación de las tramas recibidas mediante RS-232 para asegurar que los eventos de tránsito se procesen con precisión y se registren adecuadamente en la cola local, como el manejo de comandos y sus respuestas al servidor central.
   Story points: 8 (complejidad: 3, dificultad: 2, incertidumbre: 1)
   Prioridad: alta

8. Como desarrollador del sistema
   quiero que el ESP32-C3 detecte y maneje errores de comunicación en el puerto RS-232 y en la conexión GPRS y se reconecte automáticamente cuando sea necesario.
   para garantizar la estabilidad y continuidad del sistema, evitar la pérdida de eventos de tránsito y asegurar una comunicación confiable con el servidor.
   Story points: 5 (complejidad: 3, dificultad: 1, incertidumbre: 1)
   Prioridad: media

8. Entregables principales del proyecto

- Manual de usuario: guía detallada para operadores y administradores sobre el uso de la interfaz web, incluyendo la visualización de eventos de tránsito y el envío de comandos remotos.
- Manual técnico de instalación y configuración: instrucciones para la instalación física de los dispositivos ESP32-C3, configuración de la comunicación RS-232 y GPRS, y puesta en marcha del sistema.
- Código fuente del firmware (ESP32-C3) que incluirá:
  - Recepción y procesamiento de datos a través de RS-232.
  - Manejo de errores de comunicación y reconexión automática.
  - Registro de eventos en cola local para garantizar la integridad de los datos.
  - Envío de datos al servidor mediante MQTT sobre GPRS.
  - Recepción de comandos mediante MQTT sobre GPRS y respuesta
- Código fuente del backend (API REST) que incluirá:
  - Suscripción al *broker* MQTT para recibir eventos de tránsito.
  - Procesamiento y almacenamiento de los datos recibidos.
  - Envío de comandos a los dispositivos en campo.
- Código fuente del frontend que incluirá:
  - Visualización en tiempo real de los eventos de tránsito.

    - Interfaz para el envío de comandos y control remoto de dispositivos.
- Diagramas de arquitectura del sistema:
    - La arquitectura general del sistema.
    - La comunicación entre componentes (contador, ESP32-C3, servidor, interfaz web).
    - El flujo de datos desde la detección de eventos hasta su visualización.
    - El flujo de datos desde el servidor hasta el contador.
- Esquemas eléctricos y de comunicación incluyendo los diagramas de conexiones entre los dispositivos.
- Plan de pruebas y resultados.
- Memoria del trabajo final.

9. Desglose del trabajo en tareas

1. Análisis y diseño del sistema (65 h):
    1.1. Revisión de requerimientos funcionales y no funcionales (10 h)
    1.2. Diseño de arquitectura del sistema (15 h)
    1.3. Selección y evaluación de tecnologías (15 h)
    1.4. Elaboración de diagramas de flujo y casos de uso (10 h)
    1.5. Revisión y validación del diseño con los interesados (15 h)
2. Aprendizaje en tecnologías (45 h):
    2.1. Estudio del protocolo MQTT y su implementación (10 h)
    2.2. Estudio ESP32-C3 (10 h)
    2.3. Aprendizaje sobre implementación de API REST (10 h)
    2.4. Capacitación en desarrollo de interfaces web (10 h)
    2.5. Investigación sobre manejo de errores y reconexión en comunicaciones (5 h)
3. Desarrollo del firmware para ESP32-C3 (90 h):
    3.1. Implementación de envío y recepción de datos por RS-232 (15 h)
    3.2. Gestión de cola en memoria RAM para eventos (15 h)
    3.3. Publicación de eventos en *broker* MQTT sobre GPRS (20 h)
    3.4. Suscripción a tópicos MQTT para recepción de comandos (15 h)
    3.5. Ejecución y respuesta a comandos recibidos (15 h)
    3.6. Manejo de errores de comunicación y reconexión automática (10 h)
4. Testing del firmware para ESP32-C3 (20 h):
    4.1. Pruebas unitarias de módulos individuales (10 h)
    4.2. Pruebas de integración con módulos de comunicación (10 h)
5. Documentación del firmware para ESP32-C3 (10 h):

5.1. Documentación del código fuente y comentarios en línea (5 h)
5.2. Elaboración de guía de instalación y configuración (5 h)

6. Desarrollo de la API REST (75 h):
   6.1. Diseño del modelo entidad-relación y esquema de base de datos (10 h)
   6.2. Diseño de *endpoints* para recepción de eventos desde MQTT (12 h)
   6.3. Implementación de lógica para procesamiento de eventos (12 h)
   6.4. Desarrollo de *endpoints* para envío de comandos al ESP32-C3 (12 h)
   6.5. Integración con el *broker* MQTT (12 h)
   6.6. Implementación de autenticación y seguridad (10 h)
   6.7. Pruebas unitarias y de integración de la API (7 h)
7. Testing de la API REST (15 h):
   7.1. Pruebas de carga y rendimiento (5 h)
   7.2. Pruebas de seguridad y validación de datos (5 h)
   7.3. Pruebas de compatibilidad con diferentes usuarios (5 h)
8. Documentación de la API REST (10 h):
   8.1. Documentación de *endpoints* y ejemplos de uso (5 h)
   8.2. Elaboración de guía de integración para desarrolladores (5 h)
9. Desarrollo de la interfaz web (65 h):
   9.1. Diseño de la interfaz de usuario (15 h)
   9.2. Implementación de visualización de eventos de tránsito (15 h)
   9.3. Desarrollo de funcionalidad para envío de comandos (15 h)
   9.4. Integración con la API REST (10 h)
   9.5. Pruebas de usabilidad y funcionalidad (10 h)
10. Testing de la interfaz web (15 h):
   10.1. Pruebas de compatibilidad con diferentes navegadores (5 h)
   10.2. Pruebas de accesibilidad y experiencia de usuario (5 h)
   10.3. Pruebas de rendimiento y tiempos de carga (5 h)
11. Documentación de la interfaz web (10 h):
   11.1. Manual de usuario y guía de navegación (5 h)
   11.2. Documentación de instalación y despliegue (5 h)
12. Infraestructura y configuración del *broker* MQTT (25 h):
   12.1. Instalación y configuración del *broker* MQTT (10 h)
   12.2. Configuración de tópicos y políticas de QoS (10 h)
   12.3. Pruebas de comunicación entre dispositivos y *broker* (5 h)
13. Documentación de infraestructura (10 h):
   13.1. Documentación de configuración del *broker* MQTT (5 h)

13.2. Guía de mantenimiento y monitoreo (5 h)

14. Gestión del proyecto y coordinación (45 h):
    14.1. Planificación y seguimiento del proyecto (15 h)
    14.2. Reuniones de coordinación y revisión (15 h)
    14.3. Gestión de riesgos y control de calidad (10 h)
    14.4. Comunicación con los interesados y reporte de avances (5 h)
15. Pruebas e integración del sistema (45 h):
    15.1. Pruebas de integración entre módulos (15 h)
    15.2. Pruebas de sistema y validación con casos de uso (15 h)
    15.3. Pruebas de aceptación por parte del cliente (15 h)
16. Despliegue y mantenimiento inicial (45 h):
    16.1. Despliegue del sistema en entorno de producción (15 h)
    16.2. Monitoreo y soporte después de despliegue (15 h)
    16.3. Corrección de errores y ajustes finales (15 h)
17. Buffer para imprevistos (60 h):
    17.1. Tiempo reservado para manejar retrasos, cambios de alcance o problemas técnicos imprevistos.

Cantidad total de horas: 600.

10. Diagrama de Activity On Node

En la figura 2 se observa el diagrama AON, que presenta las actividades principales del proyecto y sus dependencias temporales. Las tareas se organizan en forma de red, donde las flechas indican la secuencia de ejecución. El camino crítico (resaltado en rojo) muestra las actividades cuya duración afecta directamente el plazo total del proyecto, por lo que deben gestionarse con especial atención.

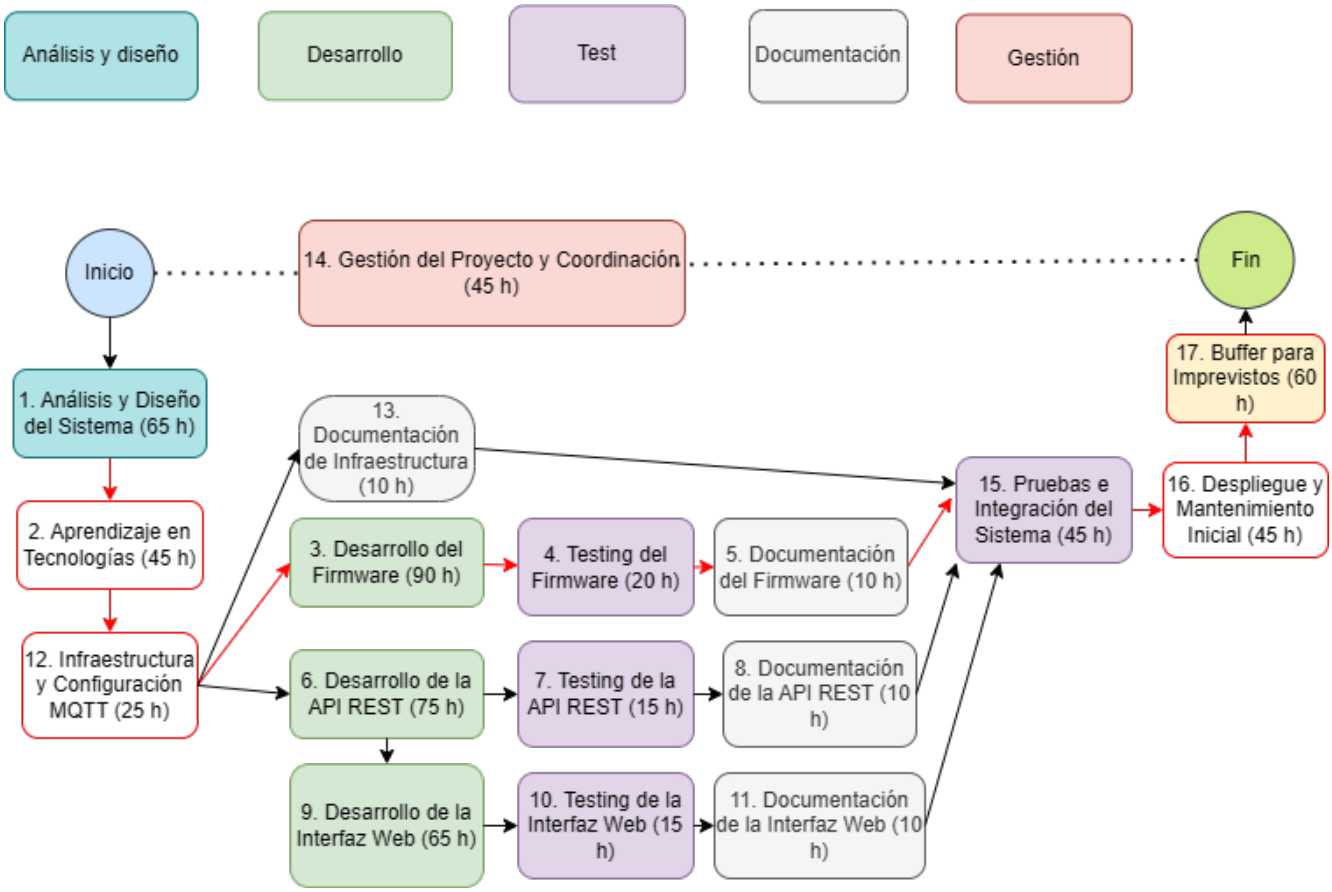

Figura 2. Diagrama de *Activity on Node*.

El camino crítico está marcado en rojo.

11. Diagrama de Gantt

En la figura 3 se presenta el diagrama de Gantt correspondiente a la planificación del proyecto. En él se detallan las tareas a realizar, sus fechas de inicio y fin, así como la duración estimada de cada una.

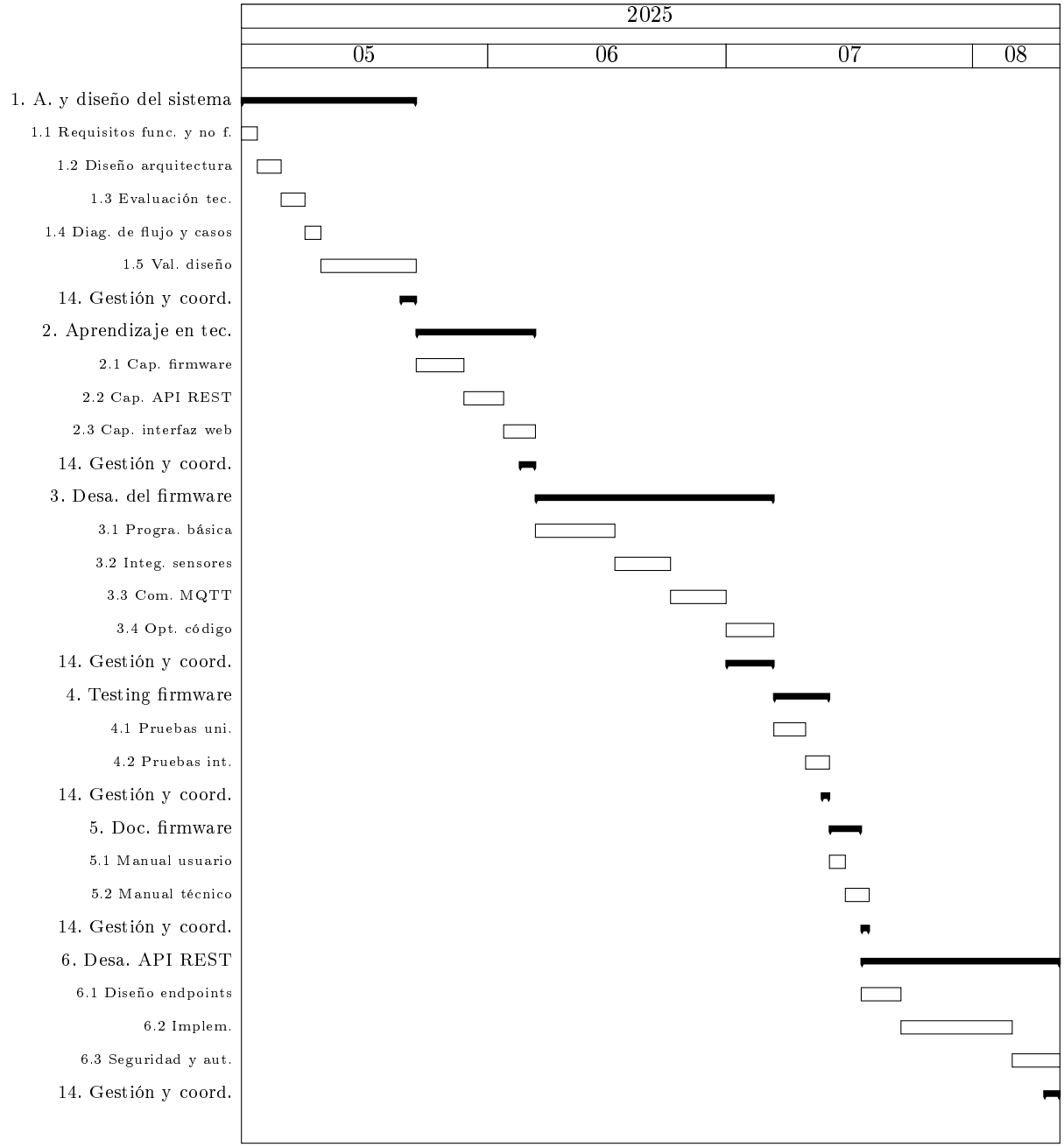

Figura 3. Diagrama de Gantt parte 1.

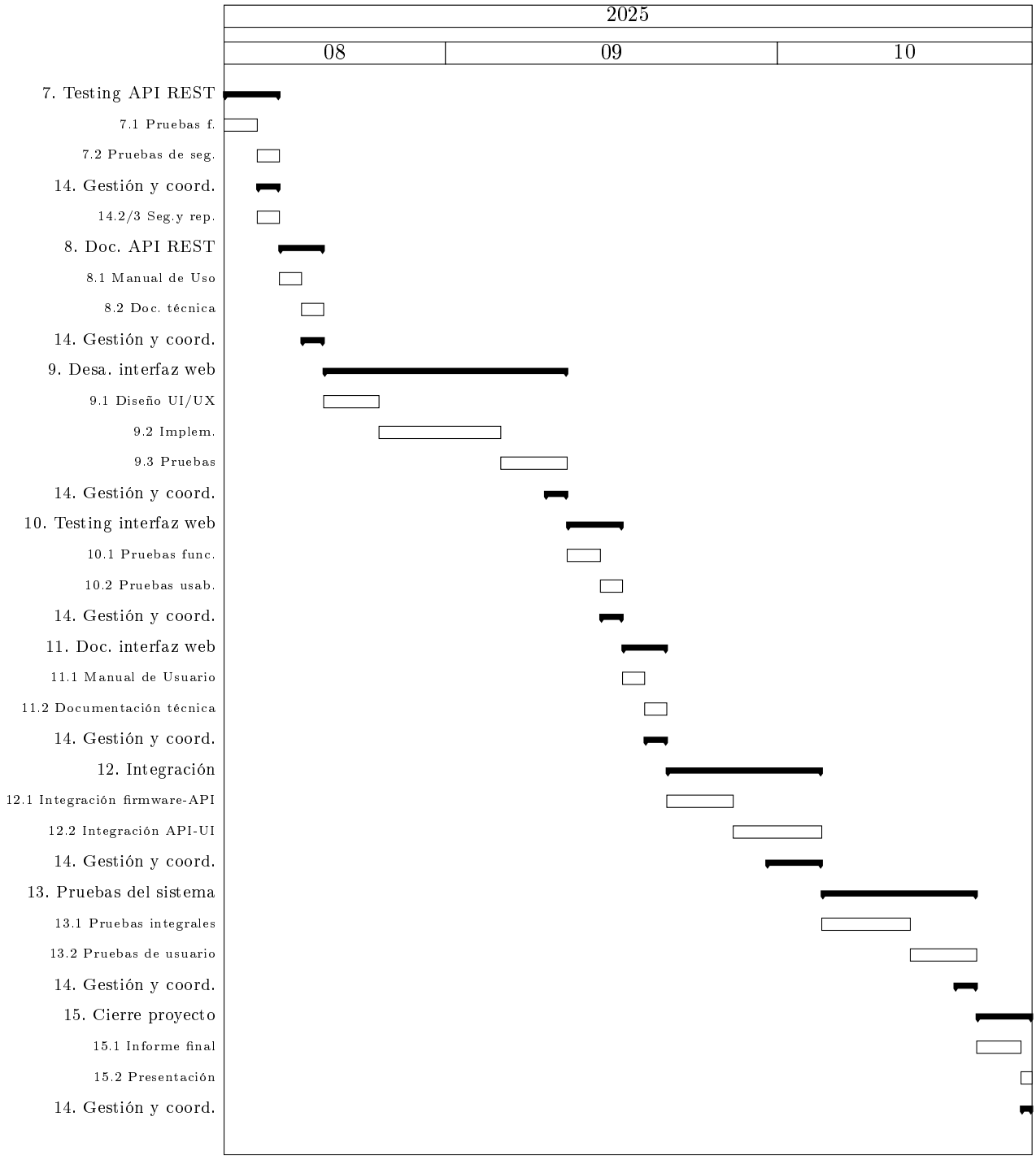

Figura 4. Diagrama de Gantt parte 2.

Supuestos considerados para la planificación:

- La jornada de trabajo estimada es de 4 horas diarias, excluyendo fines de semana.
- Las tareas se desarrollan mayormente de forma secuencial, dado que se cuenta con un único recurso humano.
- Las actividades vinculadas a la gestión del proyecto se realizan en paralelo al desarrollo técnico.

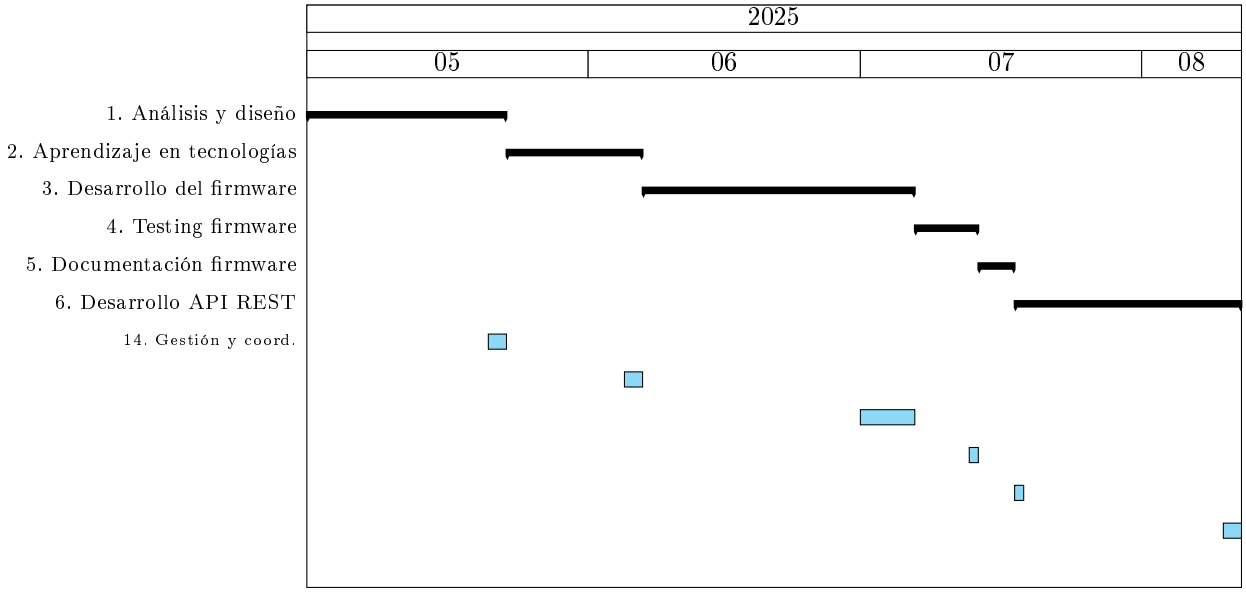

Figura 5. Diagrama de Gantt parte 1 (sin subtareas).

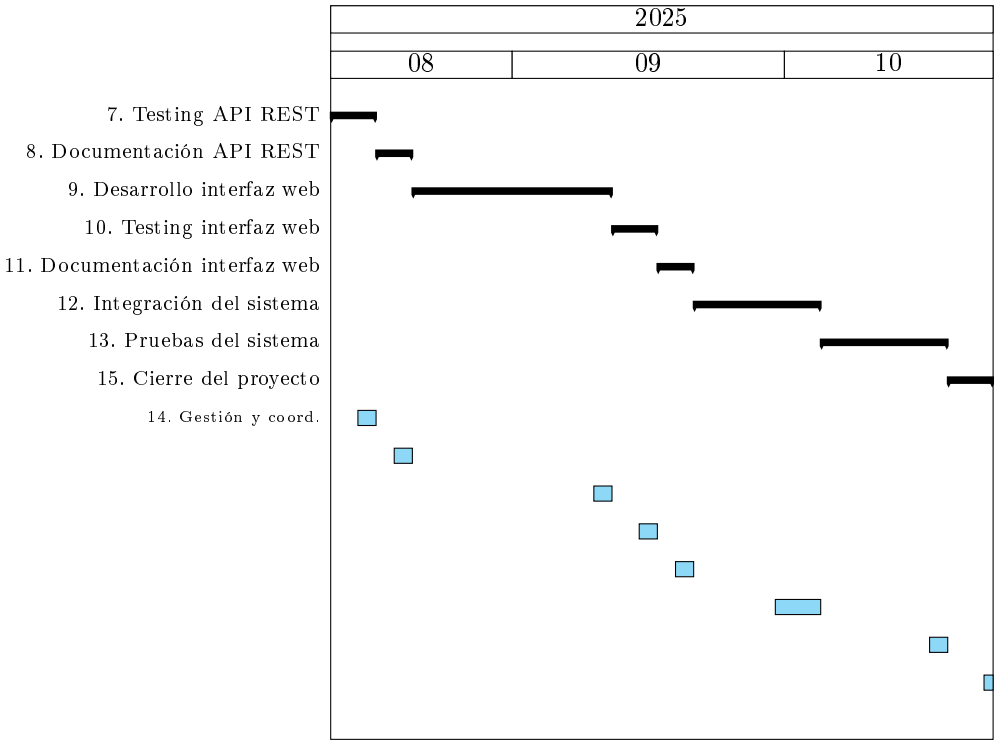

Figura 6. Diagrama de Gantt parte 2 (sin subtareas).

12. Presupuesto detallado del proyecto

COSTOS DIRECTOS			
Descripción	Cantidad	Valor unitario	Valor total
Horas de desarrollo del responsable	600	$18.000	$10.800.000
PostGrado IOT	1	$4.000.000	$4.000.000
SUBTOTAL			$14.800.000
COSTOS INDIRECTOS			
Descripción	Cantidad	Valor unitario	Valor total
40 por ciento de gastos directos	1	$5.920.000	$5.920.000
SUBTOTAL			$5.920.000
TOTAL			$20.720.000

Los valores de los precios están en pesos argentinos.

13. Gestión de riesgos

a) Identificación de riesgos y consecuencias:

Riesgo 1: incompatibilidad del módulo SIM800L con comunicación MQTT.

- Severidad (S): 8.
  Justificación: requeriría rediseñar todo el sistema de comunicación lo que impactaría en el cronograma.
- Probabilidad (O): 6.
  Justificación: el SIM800L tiene limitaciones conocidas para implementar MQTT de forma estable.

Riesgo 2: fallas prolongadas en la conexión GPRS.

- Severidad (S): 7.
  Justificación: pérdida de datos críticos de tránsito durante los períodos sin conexión.
- Probabilidad (O): 7.
  Justificación: las redes celulares en zonas rurales suelen tener intermitencias frecuentes.

Riesgo 3: saturación de la memoria RAM del ESP32-C3.

- Severidad (S): 6.
  Justificación: pérdida de datos no transmitidos.
- Probabilidad (O): 5.
  Justificación: depende del volumen de tráfico y frecuencia de eventos.

Riesgo 4: vulnerabilidades de seguridad en la comunicación.

- Severidad (S): 8.
  Justificación: podría permitir acceso no autorizado o manipulación de datos.
- Probabilidad (O): 4.
  Justificación: el uso de protocolos seguros reduce este riesgo.

Riesgo 5: incompatibilidad del protocolo RS232 con el contador desarrollado por María Clara Cutrone.

- Severidad (S): 7. Justificación: dependencia de desarrollo externo y posible incompatibilidad o falta de documentación.
- Probabilidad (O): 5.
  Justificación: riesgo moderado dado que el protocolo RS232 es estándar, pero la implementación específica puede variar.

b) Tabla de gestión de riesgos:

Criterio adoptado:
Se tomarán medidas de mitigación en riesgos con RPN >30.

c) Plan de mitigación:

Riesgo 1: incompatibilidad SIM800L/MQTT.

Riesgo	S	O	RPN	S*	O*	RPN*
Incompatibilidad del módulo SIM800L con comunicación MQTT	8	6	48	5	3	15
Fallas prolongadas en la conexión GPRS	7	7	49	5	4	20
Saturación de la memoria RAM del ESP32-C3	6	5	30	5	3	15
Vulnerabilidades de seguridad en la comunicación	8	4	32	6	2	12
Incompatibilidad del protocolo RS232 con el contador	7	5	35	5	3	15

- Mitigación: implementar protocolo HTTP como alternativa, usando POST para enviar datos.
- S*: 5 (impacto reducido al tener alternativa funcional).
- O*: 3 (solución probada y documentada).

Riesgo 2: fallas conexión GPRS.

- Mitigación: implementar almacenamiento local persistente y lógica de reintentos inteligente.
- S*: 5 (datos se recuperan post-reconexión).
- O*: 4 (mejora en manejo de desconexiones).

Riesgo 4: vulnerabilidades seguridad.

- Mitigación: implementar autenticación por *tokens* JWT.
- S*: 6 (reduce impacto de posibles ataques).
- O*: 2 (medidas comprobadas reducen vulnerabilidades).

Riesgo 5: incompatibilidad protocolo RS232 con contador.

- Mitigación: coordinar con María Clara Cutrone para validar y documentar la interfaz RS232. Realizar pruebas de integración tempranas.
- S*: 5 (reducción del impacto con mejor conocimiento).
- O*: 3 (mejora en la probabilidad al estar en contacto con la desarrolladora).

14. Gestión de la calidad

- Req #1: el ESP32-C3 debe recibir datos por interfaz RS-232 desde el sistema de detección.
  - Verificación: realizar pruebas de laboratorio con el sistema de detección enviando datos por RS-232 y monitorear que el ESP32-C3 los reciba correctamente sin pérdidas ni errores. Se usan analizadores lógicos y herramientas de *debug* para observar la comunicación interna.

  - Validación: demostrar con el cliente la recepción en tiempo real de eventos generados en el sistema de detección y confirmación del correcto procesamiento de datos.

- Req #3: el ESP32-C3 debe publicar cada mensaje de la cola a un *broker* MQTT remoto usando GPRS.

  - Verificación: simular eventos y verificar en laboratorio que el ESP32-C3 publique correctamente cada mensaje a un *broker* MQTT configurado, validando los *logs* y el orden de envío.
  - Validación: mostrar al cliente la recepción de mensajes en el sistema remoto bajo diferentes condiciones de conectividad y volumen de datos.

- Req #4: el protocolo MQTT debe utilizar QoS 1 o 2 para asegurar la entrega sin duplicación.

  - Verificación: testear la implementación con *broker* MQTT que soporte QoS 1 y 2, revisar que no haya duplicados en la recepción y confirmarlos mecanismos de ACK.
  - Validación: validar con el cliente que los datos recibidos en el backend no presentan duplicaciones y que la información es consistente.

- Req #5: debe haber control de reintentos ante fallos de conexión sin duplicar mensajes.

  - Verificación: en laboratorio interrumpir la conexión GPRS y observar que el ESP32-C3 hace reintentos controlados, sin duplicar mensajes en el *broker*.
  - Validación: el cliente valida que durante períodos de desconexión los datos no se pierden ni duplican, al comprobar reportes posteriores.

- Req #6: si no hay conectividad GPRS disponible, los mensajes deben permanecer en la cola en memoria.

  - Verificación: simular falta de conectividad y verificar que los mensajes se almacenan en cola interna y se publican una vez restaurada la conexión.
  - Validación: el cliente confirma que no se pierden eventos durante desconexiones y que la cola gestiona adecuadamente la memoria.

- Req #8: el ESP32-C3 debe suscribirse a un *topic* MQTT para recibir comandos desde el servidor.

  - Verificación: comprobar en pruebas que el ESP32-C3 se suscribe correctamente y recibe comandos publicados en el *topic* específico.
  - Validación: demostrar al cliente que los comandos enviados desde el servidor llegan y se interpretan correctamente en el dispositivo.

- Req #11: la API REST debe poder enviar comandos al ESP32-C3 publicando en el *topic* correspondiente.

  - Verificación: probar en laboratorio la API REST mediante el envío de comandos y confirmar la publicación correcta en MQTT.
  - Validación: el cliente verifica que los comandos emitidos desde la API llegan efectivamente al dispositivo y se ejecutan.

- Req #15: probar recepción de comandos desde el servidor y respuesta correcta.

  - Verificación: realizar pruebas funcionales mediante el envío de comandos y validar que el ESP32-C3 responde con OK, error o estados según corresponda.
  - Validación: el cliente valida que el sistema responde adecuadamente ante comandos válidos y maneja errores según lo esperado.

- Req #18: la conexión MQTT debe incluir autenticación por credenciales.
  - Verificación: revisar configuración de autenticación en el *broker* MQTT y probar conexiones con credenciales válidas e inválidas.
  - Validación: el cliente confirma que el acceso está restringido y solo usuarios autorizados pueden conectarse y publicar/suscribirse.

- Req #19: la API REST debe requerir autenticación para el envío de comandos o consulta de datos.
  - Verificación: probar mecanismos de autenticación (por ejemplo, *tokens* o claves) en la API REST y validar que rechaza accesos no autorizados.
  - Validación: el cliente verifica que solo usuarios autenticados pueden enviar comandos o consultar datos mediante la API.

15. Procesos de cierre

Para la realización de la reunión final de evaluación del proyecto se establecen las siguientes pautas de trabajo:

- Análisis del plan de proyecto original:
  Responsable: el director del proyecto (profesor a cargo).
  Procedimiento: se revisarán el cronograma, los entregables y los resultados del proyecto y se compararán con los reportes de avance realizados durante su desarrollo.

- Identificación de técnicas y procedimientos útiles e inútiles, problemas y soluciones:
  Responsables: el alumno con los colaboradores.
  Procedimiento: se llevará a cabo una reflexión crítica sobre las herramientas y métodos empleados. Se documentarán los problemas surgidos y las soluciones aplicadas, con el objetivo de extraer lecciones aprendidas para futuras experiencias similares. Estas conclusiones se incluirán en un documento de cierre.

- Organización de un acto simbólico de cierre y agradecimiento:
  Responsable: el director del proyecto y el alumno.
  Procedimiento: se coordinará un breve encuentro (virtual) con el director para agradecer su participación. Se enviarán agradecimientos formales y se compartirá un resumen de los resultados obtenidos.