



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

| | |
|--|-----------|
| 1. Descripción técnica-conceptual del proyecto a realizar | 5 |
| 1.1 Contexto y motivación | 5 |
| 1.2 Problemas identificados | 5 |
| 1.3 Estado del arte y propuesta de valor | 5 |
| 1.4 Propuesta de modernización | 5 |
| 1.5 Descripción funcional y técnica | 6 |
| 1.6 Grado de innovación | 6 |
| 2. Identificación y análisis de los interesados | 6 |
| 3. Propósito del proyecto | 7 |
| 4. Alcance del proyecto | 8 |
| 5. Supuestos del proyecto. | 8 |
| 6. Requerimientos | 8 |
| 7. Historias de usuarios (<i>Product backlog</i>). | 10 |
| 8. Entregables principales del proyecto | 11 |
| 9. Desglose del trabajo en tareas | 12 |
| 10. Diagrama de Activity On Node. | 14 |
| 11. Diagrama de Gantt | 15 |
| 12. Presupuesto detallado del proyecto | 18 |
| 13. Gestión de riesgos | 18 |
| 14. Gestión de la calidad | 19 |
| 15. Procesos de cierre | 20 |

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 |

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 \$ - , con fecha de inicio el 29 de abril de 2025 y fecha de presentación pública el 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 sobre TLS), 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, utilizando el protocolo MQTT sobre TLS para el envío seguro 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. Además, 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.

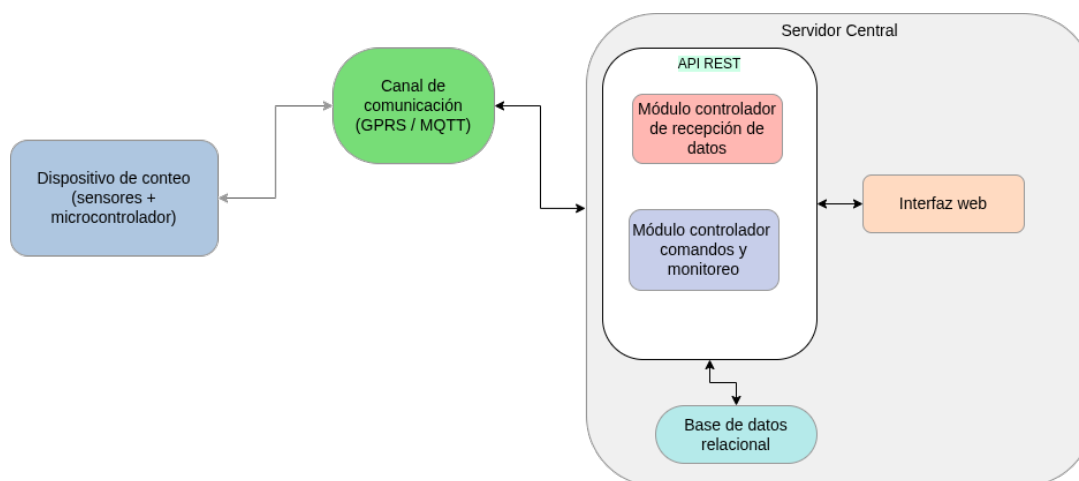


Figura 1. Diagrama en bloques del sistema.

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 Cutrone | 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 sobre TLS) 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** en 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 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 o error, también puede devolver con el estado solicitado**.
- 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 si se pudieron realizar los comandos o el estado solicitado.

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. El protocolo MQTT texto delimitado según acordado con 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. Historia usuario: **interfaz web para visualización**
Como usuario del sistema
quiero ver los eventos de tránsito en una interfaz web
para tener monitoreo visual y registro histórico de los tránsitos.
Story points: 8 (complejidad: 3, dificultad: 2, incertidumbre: 1)
Prioridad: media
- 2. Historia usuario: **envío de datos al servidor**
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. Historia usuario: **API REST que consume el MQTT**
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. Historia usuario: **envío de comandos desde la API REST**
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. Historia usuario: **interfaz web para control remoto**
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. Historia usuario: **registro de eventos de tránsito y respuesta a comandos en cola local del dispositivo ESP32-C3**

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. Historia usuario: **adaptación del contador de tránsito para interpretar protocolo RS-232**

Como desarrollador del sistema

quiero modificar el contador de tránsito en el ESP32-C3 para que interprete correctamente 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. Historia usuario: **manejo de errores de comunicación y reconexión automática en el ESP32-C3**

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. Adicionalmente realice reconexiones automáticas cuando sea necesario.

para garantizar la estabilidad y continuidad del sistema, evitando la pérdida de eventos de tránsito y asegurando 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)** Implementación del firmware que incluye:
 - 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)** Desarrollo de la API REST que:

- Se suscribe al broker MQTT para recibir eventos de tránsito.
- Procesa y almacena los datos recibidos.
- Permite el envío de comandos a los dispositivos en campo.
- **Código Fuente del Frontend (Interfaz Web)** Aplicación web que proporciona:
 - 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** Diagramas que muestran las conexiones eléctricas de los dispositivos: RS-232, ESP32-C3 y modem GPRS.
- **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 h

10. Diagrama de Activity On Node

Armar el AoN a partir del WBS definido en la etapa anterior.

Una herramienta simple para desarrollar los diagramas es el Draw.io (<https://app.diagrams.net/>). Draw.io

Indicar claramente en qué unidades están expresados los tiempos. De ser necesario indicar los caminos semi críticos y analizar sus tiempos mediante un cuadro. Es recomendable usar colores y un cuadro indicativo describiendo qué representa cada color.

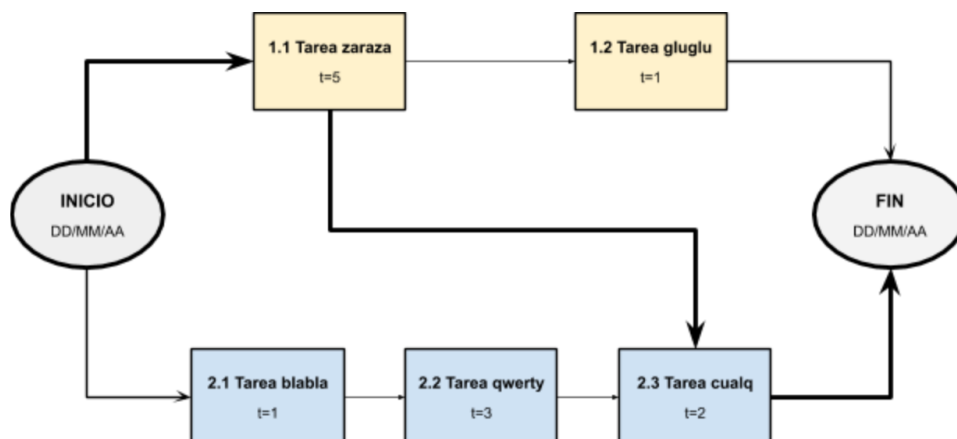


Figura 2. Diagrama de *Activity on Node*.

11. Diagrama de Gantt

Existen muchos programas y recursos *online* para hacer diagramas de Gantt, entre los cuales destacamos:

- Planner
- GanttProject
- Trello + *plugins*. En el siguiente link hay un tutorial oficial:
<https://blog.trello.com/es/diagrama-de-gantt-de-un-proyecto>
- Creately, herramienta online colaborativa.
<https://creately.com/diagram/example/ieb3p3ml/LaTeX>
- Se puede hacer en latex con el paquete *pgfgantt*
<http://ctan.dcc.uchile.cl/graphics/pgf/contrib/pgfgantt/pgfgantt.pdf>

Pegar acá una captura de pantalla del diagrama de Gantt, cuidando que la letra sea suficientemente grande como para ser legible. Si el diagrama queda demasiado ancho, se puede pegar primero la “tabla” del Gantt y luego pegar la parte del diagrama de barras del diagrama de Gantt.

Configurar el software para que en la parte de la tabla muestre los códigos del EDT (WBS).
Configurar el software para que al lado de cada barra muestre el nombre de cada tarea.
Revisar que la fecha de finalización coincida con lo indicado en el Acta Constitutiva.

En la figura 3, se muestra un ejemplo de diagrama de gantt realizado con el paquete de *pgfgantt*. En la plantilla pueden ver el código que lo genera y usarlo de base para construir el propio.

Las fechas pueden ser calculadas utilizando alguna de las herramientas antes citadas. Sin embargo, el siguiente ejemplo fue elaborado utilizando [esta hoja de cálculo](#).

Es importante destacar que el ancho del diagrama estará dado por la longitud del texto utilizado para las tareas (Ejemplo: tarea 1, tarea 2, etcétera) y el valor *x unit*. Para mejorar la apariencia del diagrama, es necesario ajustar este valor y, quizás, acortar los nombres de las tareas.

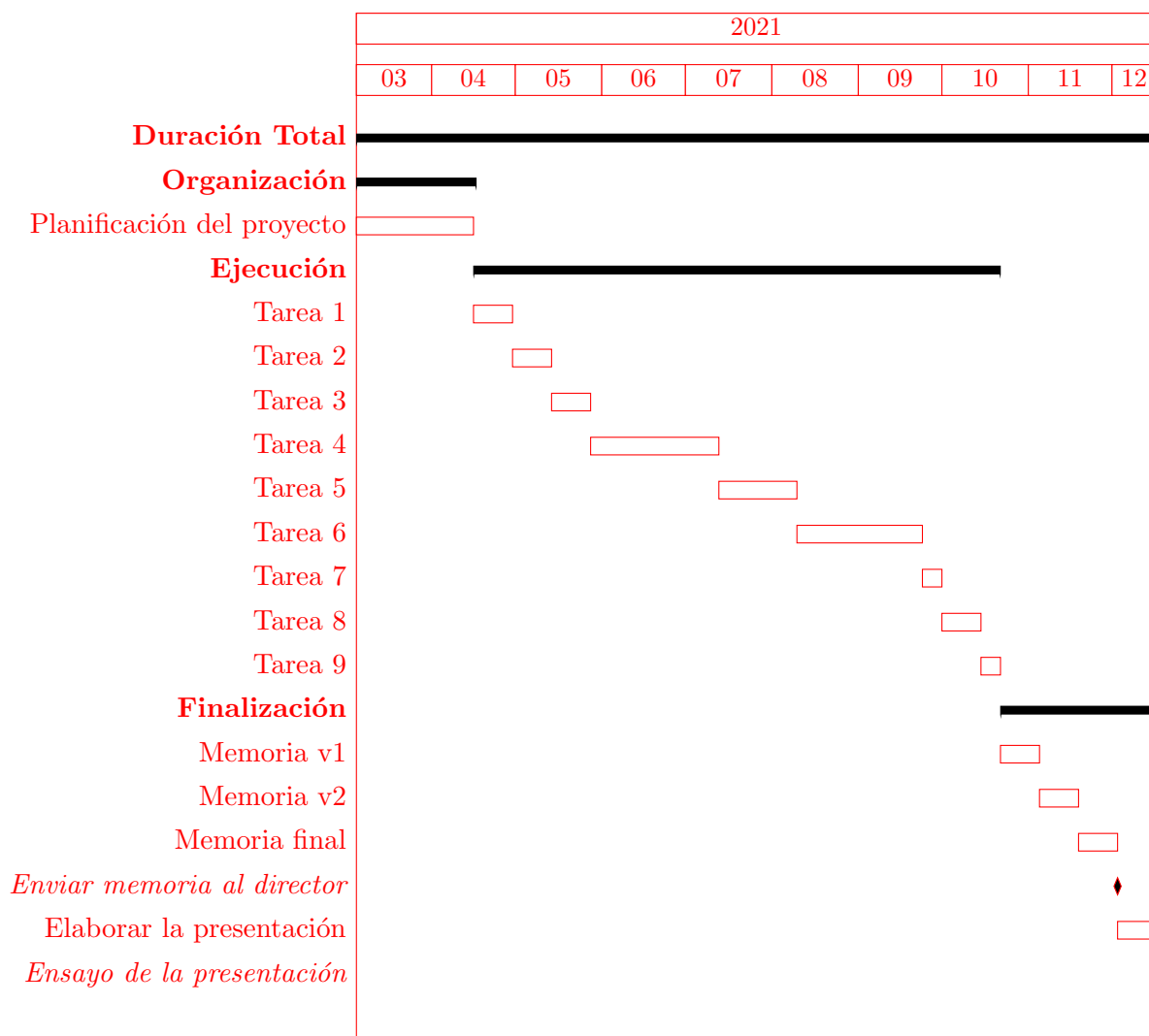


Figura 3. Diagrama de gantt de ejemplo

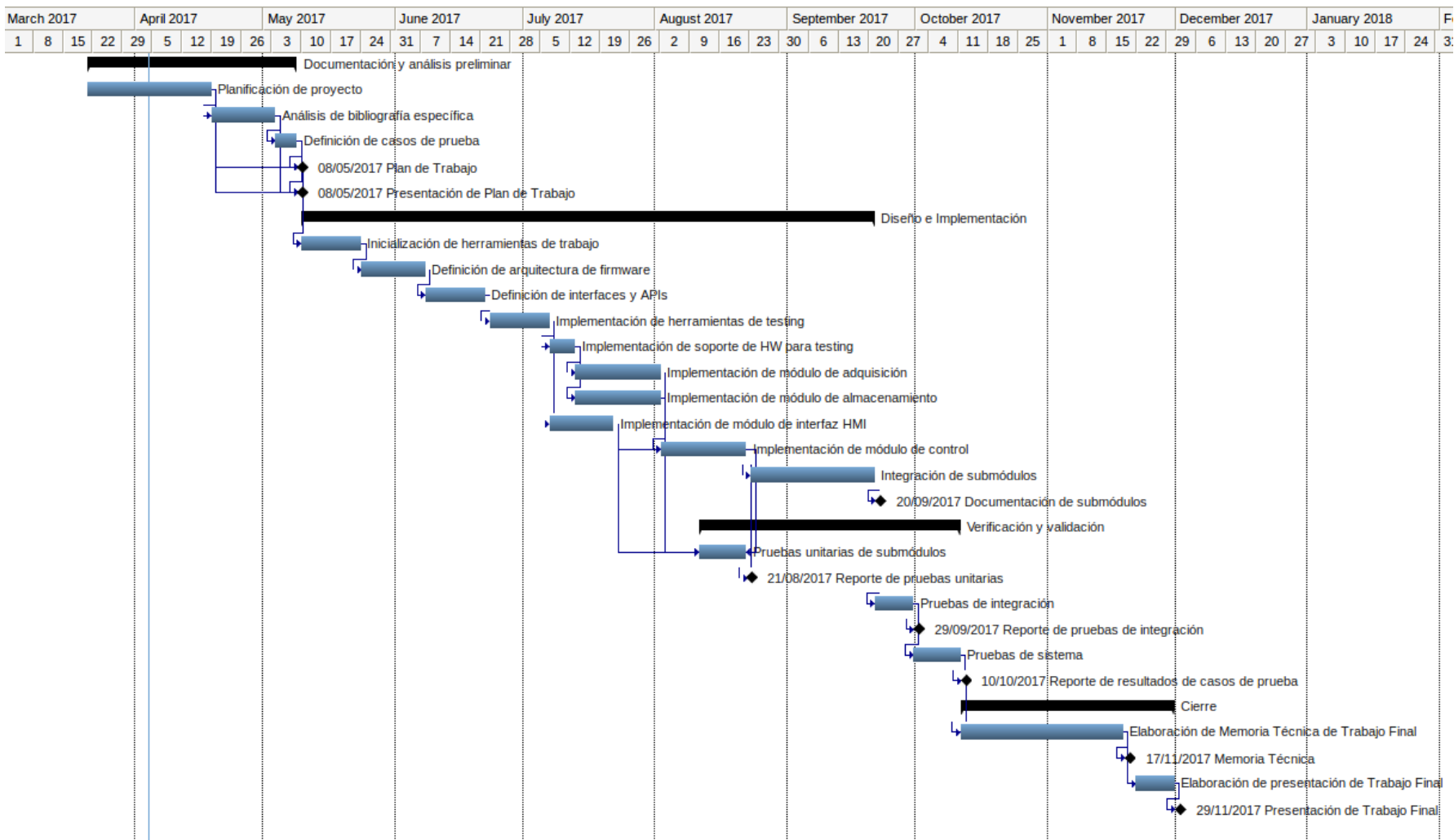


Figura 4. Ejemplo de diagrama de Gantt (apaisado).

12. Presupuesto detallado del proyecto

Si el proyecto es complejo entonces separarlo en partes:

- Un total global, indicando el subtotal acumulado por cada una de las áreas.
- El desglose detallado del subtotal de cada una de las áreas.

IMPORTANTE: No olvidarse de considerar los **COSTOS INDIRECTOS**.

Incluir la aclaración de si se emplea como moneda el peso argentino (ARS) o si se usa moneda extranjera (USD, EUR, etc). Si es en moneda extranjera se debe indicar la tasa de conversión respecto a la moneda local en una fecha dada.

| COSTOS DIRECTOS | | | |
|-------------------|----------|----------------|-------------|
| Descripción | Cantidad | Valor unitario | Valor total |
| | | | |
| | | | |
| | | | |
| | | | |
| SUBTOTAL | | | |
| COSTOS INDIRECTOS | | | |
| Descripción | Cantidad | Valor unitario | Valor total |
| | | | |
| | | | |
| | | | |
| | | | |
| SUBTOTAL | | | |
| TOTAL | | | |

13. Gestión de riesgos

a) Identificación de los riesgos (al menos cinco) y estimación de sus consecuencias:

Riesgo 1: detallar el riesgo (riesgo es algo que si ocurre altera los planes previstos de forma negativa)

- Severidad (S): mientras más severo, más alto es el número (usar números del 1 al 10). Justificar el motivo por el cual se asigna determinado número de severidad (S).
- Probabilidad de ocurrencia (O): mientras más probable, más alto es el número (usar del 1 al 10). Justificar el motivo por el cual se asigna determinado número de (O).

Riesgo 2:

- Severidad (S): X.
Justificación...

- Ocurrecia (O): Y.
Justificación...

Riesgo 3:

- Severidad (S): X.
Justificación...
- Ocurrecia (O): Y.
Justificación...

b) Tabla de gestión de riesgos: (El RPN se calcula como $RPN=S \times O$)

| Riesgo | S | O | RPN | S* | O* | RPN* |
|--------|---|---|-----|----|----|------|
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |

Criterio adoptado:

Se tomarán medidas de mitigación en los riesgos cuyos números de RPN sean mayores a...

Nota: los valores marcados con (*) en la tabla corresponden luego de haber aplicado la mitigación.

c) Plan de mitigación de los riesgos que originalmente excedían el RPN máximo establecido:

Riesgo 1: plan de mitigación (si por el RPN fuera necesario elaborar un plan de mitigación).
Nueva asignación de S y O, con su respectiva justificación:

- Severidad (S*): mientras más severo, más alto es el número (usar números del 1 al 10). Justificar el motivo por el cual se asigna determinado número de severidad (S).
- Probabilidad de ocurrencia (O*): mientras más probable, más alto es el número (usar del 1 al 10). Justificar el motivo por el cual se asigna determinado número de (O).

Riesgo 2: plan de mitigación (si por el RPN fuera necesario elaborar un plan de mitigación).

Riesgo 3: plan de mitigación (si por el RPN fuera necesario elaborar un plan de mitigación).

14. Gestión de la calidad

Elija al menos diez requerimientos que a su criterio sean los más importantes/críticos/que aportan más valor y para cada uno de ellos indique las acciones de verificación y validación que permitan asegurar su cumplimiento.

- Req #1: copiar acá el requerimiento con su correspondiente número.

- Verificación para confirmar si se cumplió con lo requerido antes de mostrar el sistema al cliente. Detallar.
- Validación con el cliente para confirmar que está de acuerdo en que se cumplió con lo requerido. Detallar.

Tener en cuenta que en este contexto se pueden mencionar simulaciones, cálculos, revisión de hojas de datos, consulta con expertos, mediciones, etc.

Las acciones de verificación suelen considerar al entregable como “caja blanca”, es decir se conoce en profundidad su funcionamiento interno.

En cambio, las acciones de validación suelen considerar al entregable como “caja negra”, es decir, que no se conocen los detalles de su funcionamiento interno.

15. Procesos de cierre

Establecer las pautas de trabajo para realizar una reunión final de evaluación del proyecto, tal que contemple las siguientes actividades:

- Pautas de trabajo que se seguirán para analizar si se respetó el Plan de Proyecto original:
 - Indicar quién se ocupará de hacer esto y cuál será el procedimiento a aplicar.
- Identificación de las técnicas y procedimientos útiles e inútiles que se emplearon, los problemas que surgieron y cómo se solucionaron:
 - Indicar quién se ocupará de hacer esto y cuál será el procedimiento para dejar registro.
- Indicar quién organizará el acto de agradecimiento a todos los interesados, y en especial al equipo de trabajo y colaboradores:
 - Indicar esto y quién financiará los gastos correspondientes.