



# Cámara de seguridad IoT con conectividad LoRaWAN

Autor:

Esp. Ing. Mauricio Barroso Benavides

Director:

Mg. Ing. Gonzalo Sanchez ( FF.AA, FIUBA)

*Esta planificación fue realizada en el curso de Gestión de la Tecnología y la Innovación  
entre el 23 de junio de 2021 y el 11 de agosto de 2021.*

## Índice

1. Descripción técnica-conceptual del proyecto a realizar . . . . .	5
2. Identificación y análisis de los interesados . . . . .	7
3. Propósito del proyecto . . . . .	7
4. Alcance del proyecto . . . . .	7
5. Supuestos del proyecto. . . . .	8
6. Requerimientos . . . . .	8
7. Historias de usuarios ( <i>Product backlog</i> ). . . . .	9
8. Entregables principales del proyecto . . . . .	9
9. Desglose del trabajo en tareas . . . . .	10
10. Diagrama de Activity On Node. . . . .	11
11. Diagrama de Gantt . . . . .	11
12. Presupuesto detallado del proyecto . . . . .	14
13. Gestión de riesgos . . . . .	14
14. Gestión de la calidad . . . . .	15
15. Procesos de cierre . . . . .	16

## Registros de cambios

Revisión	Detalles de los cambios realizados	Fecha
0	Creación del documento	23 de junio de 2021
1	Se completa hasta el punto 5 inclusive	13 de julio de 2021
2	Se completa hasta el punto 9 inclusive	27 de julio de 2021

## Acta de constitución del proyecto

Buenos Aires, 23 de junio de 2021

Por medio de la presente se acuerda con el Esp. Ing. Mauricio Barroso Benavides que su Trabajo Final de la Maestría en Sistemas Embebidos se titulará “Cámara de seguridad IoT con conectividad LoRaWAN”, consistirá esencialmente en la implementación de un prototipo de una cámara de seguridad para entornos IoT con conectividad LoRaWAN, y tendrá un presupuesto preliminar estimado de 600 hs de trabajo y US\$200, con fecha de inicio 23 de junio de 2021 y fecha de presentación pública 15 de mayo de 2022.

Se adjunta a esta acta la planificación inicial.

Ariel Lutenberg  
Director posgrado FIUBA

Mg. Ing. Gonzalo Sanchez  
Director del Trabajo Final

## 1. Descripción técnica-conceptual del proyecto a realizar

La seguridad de espacios y bienes es un aspecto muy importante en la vida de las personas. Es así que en el mercado existen distintos tipos de soluciones basadas en dispositivos electrónicos, que permiten controlar y monitorear los espacios geográficos en donde se encuentran instalados. Estos dispositivos en conjunto forman sistemas de seguridad, que pueden detectar mediante distintos tipos de datos eventos no deseados.

Los dispositivos electrónicos de seguridad más utilizados son las cámaras. Estas permiten obtener imágenes en forma de videos o fotografías del sector en particular donde se encuentran. Los primeros sistemas de seguridad basados en cámaras necesitaban de un dispositivo para almacenar las imágenes que obtenían. Hoy en día las cámaras de seguridad poseen distintos tipos de conectividad hacia Internet, lo que les permite transmitir las imágenes que obtienen hacia servidores en la nube para que sean procesadas y almacenadas.

La transferencia de datos desde una cámara de seguridad hacia Internet requiere de un gran ancho de banda, por lo que es muy común que estos dispositivos posean conectividad Wi-Fi (inalámbrico) y/o Ethernet (cableado). La utilización de estas conexiones implica la existencia previa de, generalmente, una red de tipo LAN (Local Area Network) que consta principalmente de un router que se encarga de proporcionar direcciones IP (Internet Protocol) a todos los dispositivos de la red. Este escenario es muy común en entornos urbanos, donde, los proveedores de Internet proporcionan a sus usuarios conexiones estables y de gran ancho de banda, a través de redes de fibra óptica y/o cable coaxial.

Por otro lado, las zonas rurales no cuentan con la cobertura de los proveedores de Internet de las zonas urbanas, ya que a estos no les resulta rentable, y en muchos casos posible, instalar toda la infraestructura necesaria para proporcionar sus servicios. Para estos casos existen cámaras de seguridad con conectividad móvil (redes de telefonía) para transferir datos hacia Internet. Aunque esto supone un costo económico adicional que varía en función de la cantidad de datos transferidos, es una de las pocas soluciones viables que existen actualmente en el mercado. Es común también que este tipo de cámaras dispongan de una fuente de alimentación basada en baterías o paneles solares.

Entonces, el objetivo de este trabajo es diseñar e implementar una cámara de seguridad que pueda utilizarse en lugares donde no existen redes de banda ancha y tampoco una fuente de alimentación constante, y de esta manera lograr un dispositivo con la capacidad solucionar la problemática de planteada previamente.

El desarrollo de este proyecto está basado principalmente en el área de sistemas embebidos, pero también incorpora elementos propios de IoT (Internet of Things) y ML (Machine Learning) para lograr prestaciones adecuadas al contexto actual. El Internet de las cosas o IoT, por sus siglas en inglés, se refiere a la conexión de dispositivos y objetos a Internet con el objetivo de monitorearlos e interactuar con ellos de manera remota. IoT plantea algunas problemáticas como la seguridad de la información, la alimentación de dispositivos que no disponen de una fuente de energía eléctrica constante y la administración del ancho de banda disponible. El aprendizaje automático o ML, por sus siglas en inglés, permite a los sistemas computacionales realizar predicciones y tomar decisiones en función de los datos de entrada. En los últimos años la optimización que han tenido los algoritmos de ML permiten que estos puedan ser ejecutados en sistemas embebidos, lo que da como resultado una amplia gama de nuevas aplicaciones y otra perspectiva para resolver problemas antes planteados.

Dentro de un entorno IoT elegir el tipo de red para la interconexión de nodos finales con Internet es un aspecto muy importante. Es así que, en aplicaciones donde el consumo energético y el aprovechamiento de ancho de banda son importantes, LoRaWAN es el tipo de red óptimo para estos casos. Esta red permite la conexión de nodos finales hacia gateways conectados a Internet hasta una distancia teórica de 20 Km y con un ancho de banda de hasta 500 KHz (US915). La topología de red a utilizar en este proyecto sería la que observa en la figura 1.

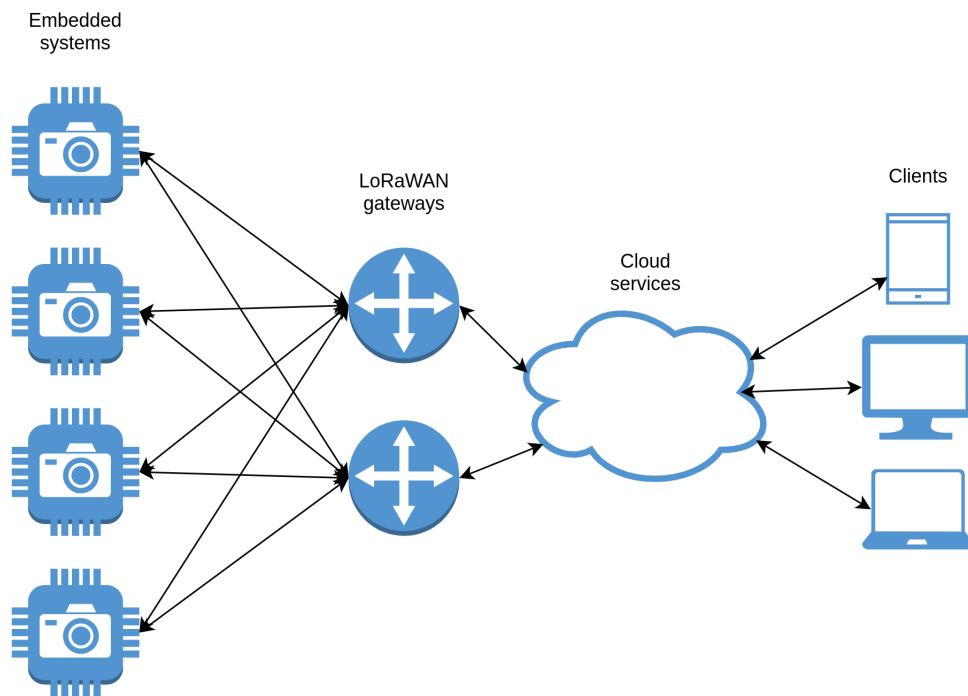


Figura 1. Diagrama en bloques del sistema

Según todas las características antes descritas, el sistema embebido implicado en este trabajo deberá tener el diagrama de bloques que se expone en la figura 2.

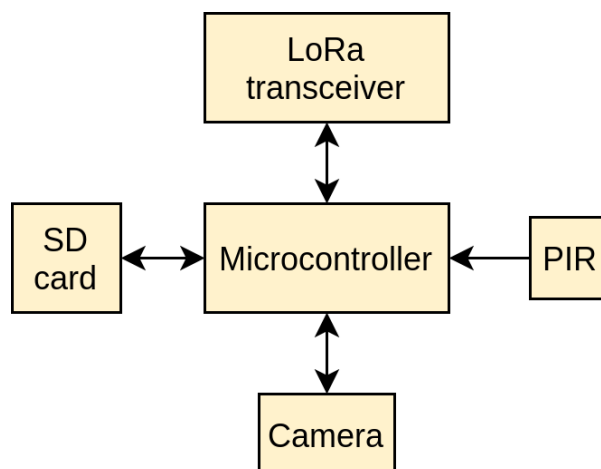


Figura 2. Diagrama en bloques del sistema

El sistema embebido tiene como componente más importante un microcontrolador que se encarga principalmente de ejecutar un modelo de machine learning para la detección de personas, sus entradas son los valores leídos por la cámara y el sensor de movimiento PIR (Passive InfraRed). Otra función del microcontrolador es la ejecución del stack LoRaWAN, para que en conjunto con el transceptor LoRa, el sistema pueda interactuar correctamente dentro de una red de este tipo. Finalmente, se cuenta con una tarjeta SD (Secure Digital) para almacenar las imágenes capturadas por la cámara.

## 2. Identificación y análisis de los interesados

Rol	Nombre y Apellido	Organización	Puesto
Responsable	Esp. Ing. Mauricio Barroso Benavides	-	Alumno
Orientador	Mg. Ing. Gonzalo Sanchez	-	Director Trabajo final
Usuario final	-	-	-

- Orientador: Tiene la capacidad de ayudar al alumno a resolver los problemas técnicos y conceptuales que se presenten a lo largo del proyecto. Es exigente con los tiempos y la calidad del desarrollo del proyecto.
- Usuario final: No posee conocimientos sobre sistemas embebidos y las tecnologías utilizadas en el proyecto, y espera dispositivos de bajo costo económico.

## 3. Propósito del proyecto

El propósito de este proyecto es diseñar e implementar un sistema embebido que funcione como cámara de seguridad en un entorno IoT con ayuda de algoritmos de ML.

Un propósito personal del autor es obtener conocimientos y experiencia sobre implementación de algoritmos de ML en sistemas embebidos.

## 4. Alcance del proyecto

Este proyecto incluye:

- Diseño e implementación del firmware del sistema embebido.
- Diseño e implementación del algoritmo de ML para reconocimiento de personas.
- Implementación del sistema embebido desarrollado dentro de una arquitectura IoT existente.
- Diseño del PCB para el prototipo funcional del sistema embebido.
- Creación de documentación sobre datos técnicos y características del sistema embebido.

Este proyecto no incluye:

- Implementación del PCB para el prototipo funcional del sistema embebido.
- Desarrollo de una aplicación para el usuario final.

## 5. Supuestos del proyecto

- Se contará con el presupuesto necesario para obtener los componentes del proyecto.
- Se dedicarán al menos 2 horas diarias para la realización del proyecto.
- Se dispondrá de una arquitectura IoT para redes LoRaWAN para la realización de pruebas y puesta en marcha del proyecto.

## 6. Requerimientos

El presente proyecto tiene dos tipos de requerimientos, funcionales y no funcionales. A continuación se citan los antes referidos.

### 1. Requerimientos funcionales

- 1.1. El sistema debe conectarse a una red LoRaWAN en la banda de frecuencia AU915 como dispositivo de clase A.
- 1.2. El sistema debe ser alimentado mediante dos baterías de tipo AA.
- 1.3. El sistema debe poseer mecanismos de seguridad implementados tanto en hardware como en software para evitar su manipulación incorrecta.
- 1.4. El sistema debe almacenar fotografías y videos obtenidos mediante su cámara en una memoria no volátil.
- 1.5. El sistema debe recibir su configuración inicial mediante interruptores físicos conectados al mismo.
- 1.6. El sistema debe recibir sus parámetros de funcionamiento mediante la red LoRaWAN a la que esta conectada.
- 1.7. El sistema debe funcionar solamente si se detecta movimiento en el sector donde se encuentra instalada.
- 1.8. El sistema debe ser compatible con la especificación LoRaWAN 1.0.2.
- 1.9. El sistema debe, con ayuda de su cámara, detectar la presencia de personas a través de un algoritmo de ML.

### 2. Requerimientos no funcionales

- 2.1. El sistema deberá tener un costo de desarrollo igual o menor a 200\$us.
- 2.2. El sistema deberá tener documentación adecuada sobre su uso y desarrollo.



## 7. Historias de usuarios (*Product backlog*)

El criterio para determinar los *storyboards* es el siguiente.

- Complejidad

- Alto: 5
- Medio: 3
- Bajo: 1

- Tiempo

- Alto: 5
- Medio: 3
- Bajo: 1

- Recursos

- Alto: 5
- Medio: 3
- Bajo: 1

1. Como usuario quiero que la duración de las baterías sea al menos de 1 año para que no suponga un costo económico adicional muy alto.

- Complejidad: 3
- Tiempo: 4
- Recursos: 1

2. Como usuario quiero que la configuración del sistema se haga en pocos pasos para evitar complicaciones en su puesta en marcha.

- Complejidad: 4
- Tiempo: 4
- Recursos: 1

3. Como desarrollador del firmware quiero que el sistema esté basado en un RTOS(Real Time Operating System) para poder escalar la aplicación de manera más sencilla.

- Complejidad: 3
- Tiempo: 4
- Recursos: 1

## 8. Entregables principales del proyecto

Los entregables del proyecto son:

- Repositorio en Github con el código fuente.
- Prototipo de pruebas del sistema embebido.
- Informe final.

## 9. Desglose del trabajo en tareas

1. Análisis y planificación preliminar (122 hs)
  - 1.1. Planificar el proyecto. (38 hs)
  - 1.2. Investigar sobre sistemas similares. (24 hs)
  - 1.3. Definir los componentes a utilizar. (6 hs)
  - 1.4. Recopilar y estudiar documentos (datasheets y application notes) de los componentes a utilizar. (30 hs)
  - 1.5. Investigar las normativas locales sobre redes que utilizan las bandas ISM(Industrial, Scientific and Medic). (12 hs)
  - 1.6. Planificar y diagramar la arquitectura a utilizar en el sistema (10 hs)
  - 1.7. Crear un repositorio en la nube para almacenar el código a desarrollar (2 hs)
2. Prototipo de pruebas (40 hs)
  - 2.1. Obtener los componentes del sistema. (12 hs)
  - 2.2. Probar la funcionalidad de todos los componentes del sistema por separado. (26 hs)
  - 2.3. Montar el prototipo de pruebas y asegurar conexiones. (2 hs)
3. LoRaWAN (66 hs)
  - 3.1. Desarrollar código para el stack LoRaWAN. (22 hs)
  - 3.2. Desarrollar código para configurar el sistema como dispositivo LoRaWAN de clase A. (18 hs)
  - 3.3. Desarrollar código para enviar y recibir paquetes de datos a través de una red LoRaWAN en la banda AU915. (16 hs)
  - 3.4. Probar el código desarrollado en el prototipo de pruebas. (10 hs)
4. ML (148 hs)
  - 4.1. Realizar un curso sobre TinyML. (40 hs)
  - 4.2. Desarrollar el modelo de ML a utilizar. (30 hs)
  - 4.3. Desarrollar código para integrar el modelo de ML en el sistema. (32 hs)
  - 4.4. Desarrollar una biblioteca de código para controlar la cámara del sistema. (24 hs)
  - 4.5. Desarrollar código para integrar la cámara del sistema con el modelo de ML. (12 hs)
  - 4.6. Probar el código desarrollado en el prototipo de pruebas. (10 hs)
5. IoT (82 hs)
  - 5.1. Configurar el entorno IoT para integrar al sistema. (14 hs)
  - 5.2. Definir el formato de los paquetes a recibir y transmitir. (2 hs)
  - 5.3. Desarrollar código para integrar el código de LoRaWAN y ML. (48 hs).
  - 5.4. Probar el código desarrollado en el prototipo de pruebas. (10 hs)
  - 5.5. Implementar el prototipo de pruebas dentro del entorno IoT. (8 hs)
6. V&V (48 hs)
  - 6.1. Medir y comprobar los parámetros eléctricos del sistema con instrumentos de laboratorio. (8 hs)

- 6.2. Probar el sistema en un entorno de trabajo controlado. (36 hs)
- 6.3. Analizar las experiencias de uso reportadas por los usuarios. (4 hs)
- 7. Cierre (106 hs)
  - 7.1. Elaborar la memoria técnica del trabajo final. (80 hs)
  - 7.2. Elaborar la presentación del trabajo final. (26 hs)

Cantidad total de horas: (612 hs)

## 10. Diagrama de Activity On Node

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

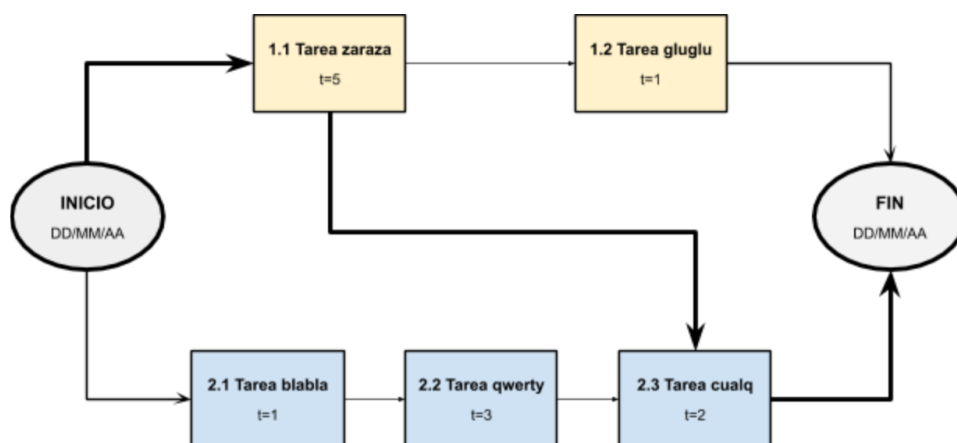


Figura 3. Diagrama en *Activity on Node*

Indicar claramente en qué unidades están expresados los tiempos. De ser necesario indicar los caminos semicríticos y analizar sus tiempos mediante un cuadro. Es recomendable usar colores y un cuadro indicativo describiendo qué representa cada color, como se muestra en el siguiente ejemplo:

## 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 4, 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.

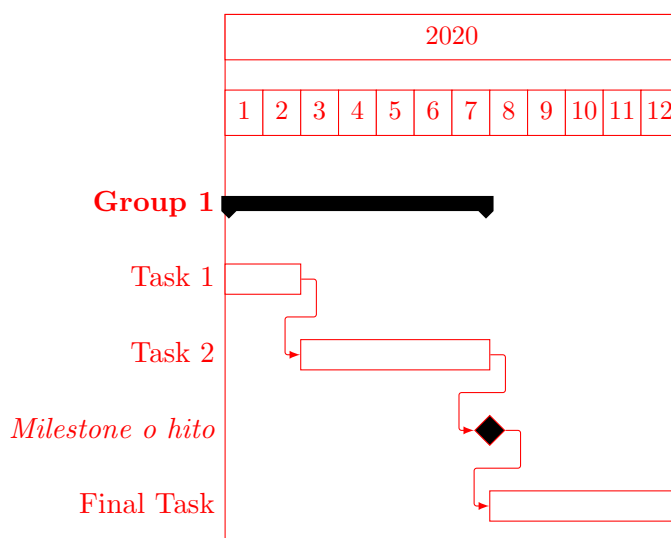


Figura 4. Diagrama de gantt de ejemplo

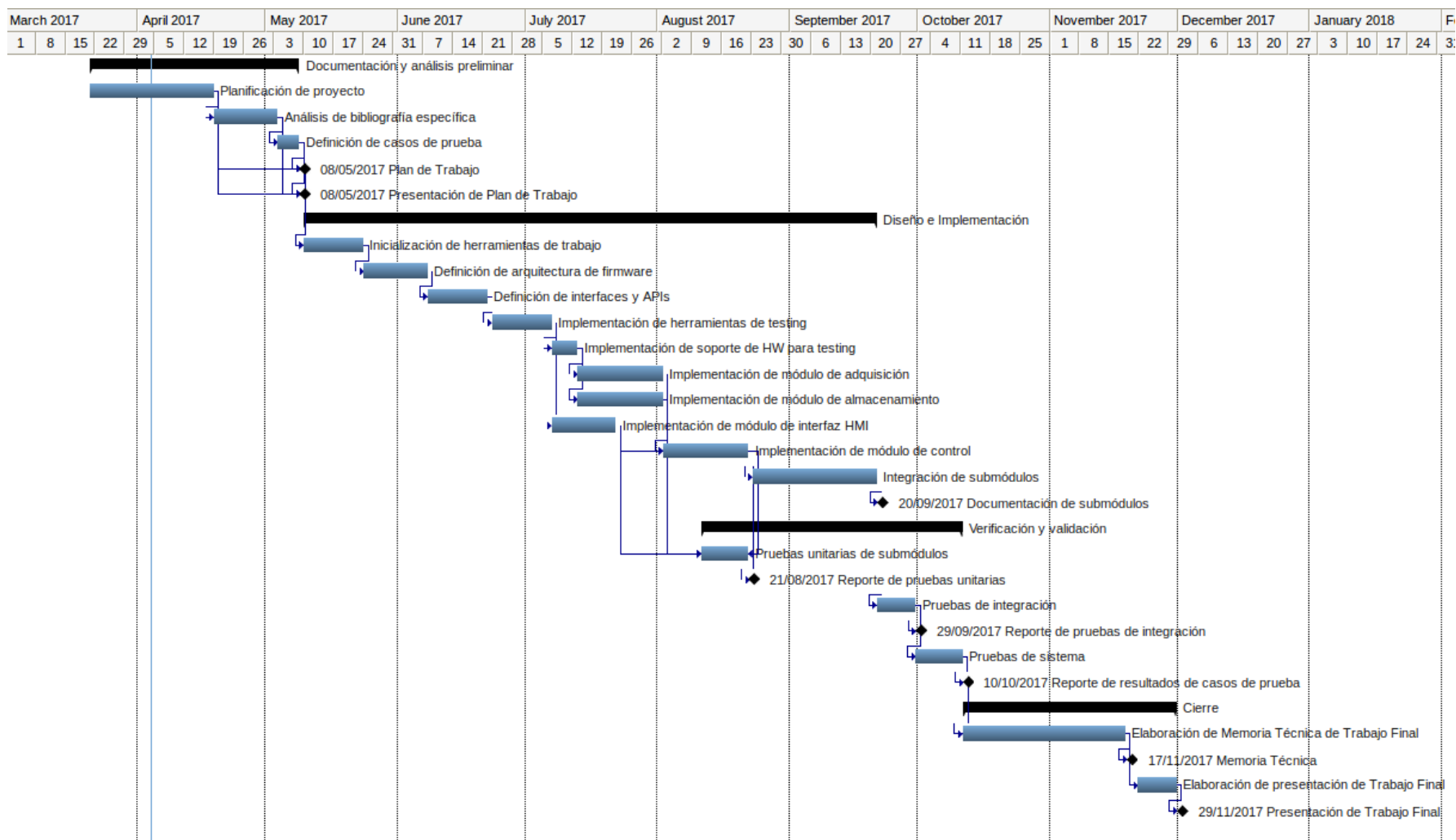


Figura 5. Ejemplo de diagrama de Gantt rotado

## 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**.

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):
- Ocurrencia (O):

Riesgo 3:

- Severidad (S):

■ Ocurrencia (O):

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

Para cada uno de los requerimientos del proyecto indique:

- Req #1: copiar acá el requerimiento.
  - 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, y 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.