



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	13
13. Gestión de riesgos	13
14. Gestión de la calidad	15
15. Procesos de cierre	17

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
3	Se completa hasta el punto 11 inclusive	4 de Agosto de 2021
4	Se completa hasta el punto 15 inclusive. Se deja pendiente el punto 14	4 de Agosto de 2021
5	Se completa el punto 14	10 de Agosto 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 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 se observa en la figura 1.

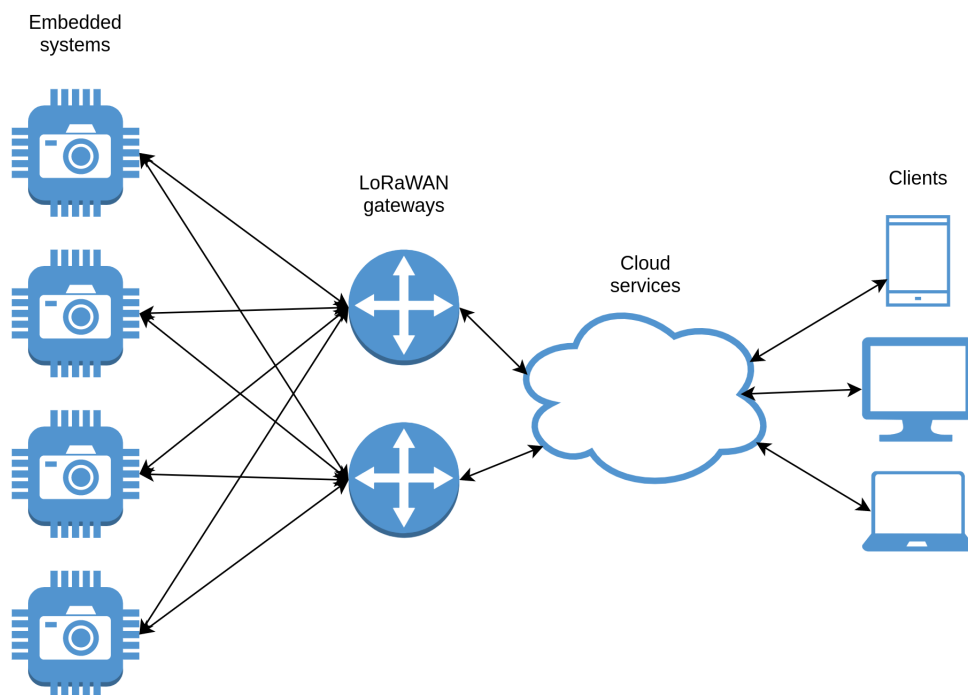


Figura 1. Diagrama en bloques del sistema

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.

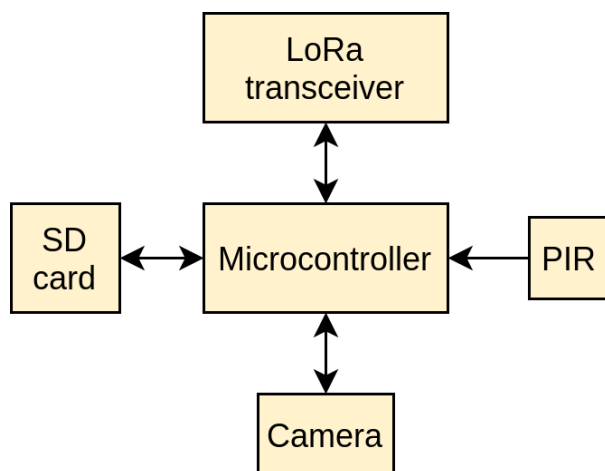


Figura 2. Diagrama en bloques del sistema

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 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 está 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 200U\$S.
- 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 Medical). (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

El diagrama de actividades se puede observar en la figura 3.

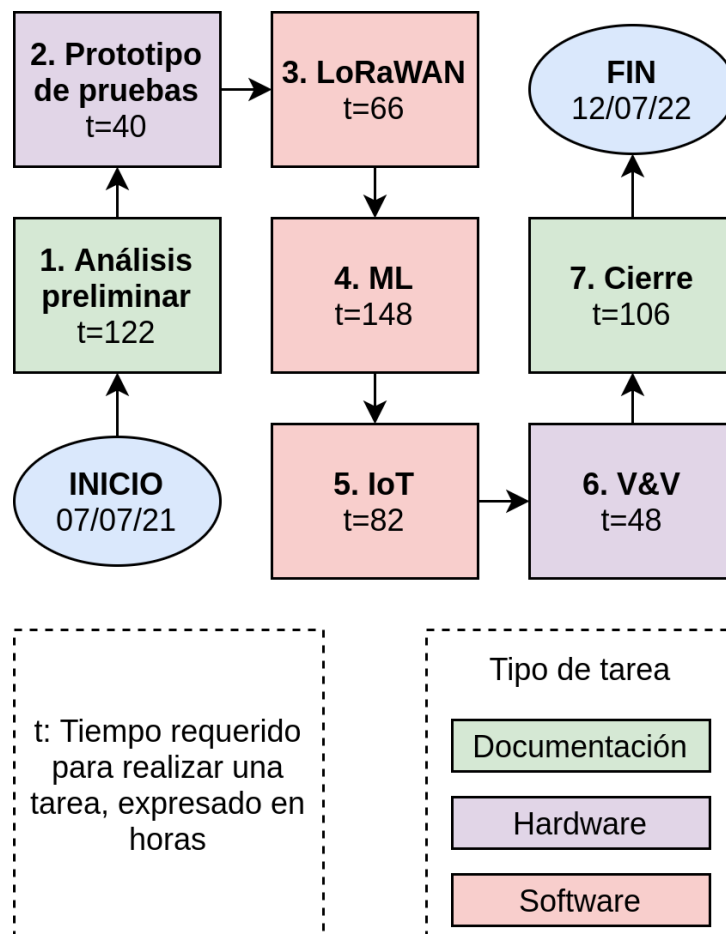


Figura 3. Diagrama en *Activity on Node*

11. Diagrama de Gantt

En la figura 11 se puede observar el diagrama de Gantt de este proyecto.

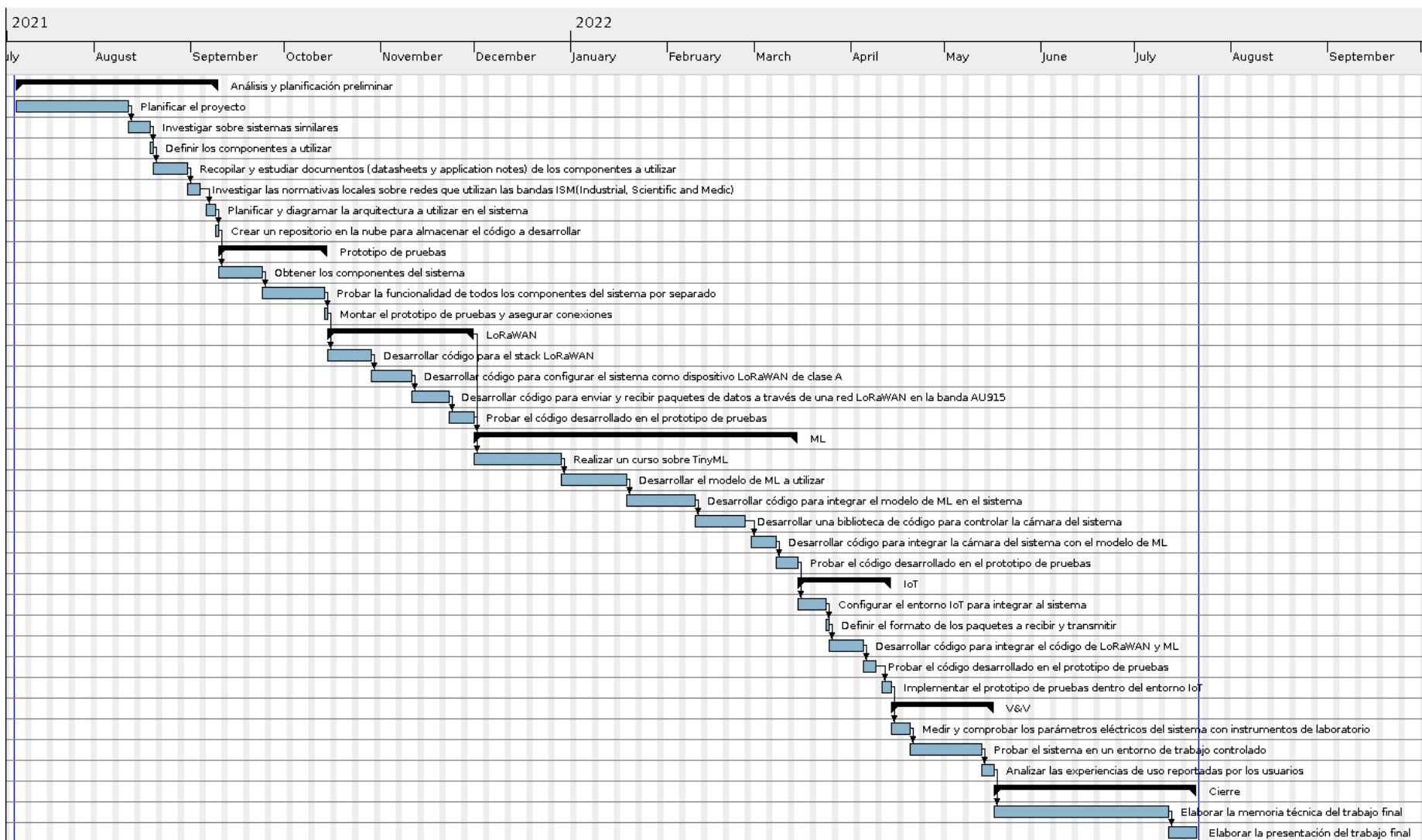


Figura 4. Diagrama de Gantt

12. Presupuesto detallado del proyecto

El presupuesto que se presenta en esta sección es un aproximado, ya que valores de los componentes que son parte del sistema embebido aún no fueron definidos. Por tanto, se consideran los costos más altos del mercado de componentes que cumplan las funciones requeridas. De esta manera se tiene un presupuesto suficiente para cumplir con el proyecto.

COSTOS DIRECTOS			
Descripción	Cantidad	Valor unitario	Valor total
Placa de desarrollo de ST (ARM Cortex M4)	1	47	47
Módulo con cámara de 5 megapíxeles	1	45	45
Módulo transceptor LoRaWAN US915	1	25	25
Módulo para tarjeta SD	1	3	3
Módulo sensor PIR	1	2	2
Módulo para tarjeta SD	1	3	3
Otros (cables, conectores, etc)	1	10	10
SUBTOTAL			135
COSTOS INDIRECTOS			
Descripción	Cantidad	Valor unitario	Valor total
50 % del costo directo	1	67,5	67,5
SUBTOTAL			67,5
TOTAL			202,5

Los valores expuestos en la tabla anterior están expresados en dólares americanos (U\$S). Esto se debe a que la mayor parte de los componentes serán importados directamente de proveedores en Estados Unidos.

13. Gestión de riesgos

a) Identificación de los riesgos y estimación de sus consecuencias: Riesgo 1: Imposibilidad de obtener los componentes necesarios.

- Severidad (S): 7. El sistema podría no funcionar de acuerdo a lo planificado.
- Probabilidad de ocurrencia (O): 3. Si no se encuentran los componentes en el mercado local, estos pueden ser importados

Riesgo 2: El tamaño del modelo de ML puede sobrepasar la cantidad de memoria del microcontrolador.

- Severidad (S): 9. Si el modelo de ML no puede ser almacenado en la memoria del microcontrolador, el sistema no podría realizar las funciones para las que fue planificado.
- Ocurrencia (O): 7. El tamaño del modelo de ML alcanzará un tamaño no adecuado en función de su complejidad.

Riesgo 3: Imposibilidad de contar con instrumentación de laboratorio.

- Severidad (S): 3. Las mediciones de alta precisión son para contar con datos técnicos exactos, que ayudarán a desarrollar una versión comercial del sistema posteriormente.
- Ocurrencia (O): 2. En el lugar de residencia se cuenta con un laboratorio de electrónica que funciona regularmente y está abierto al público previa aceptación de los encargados.

Riesgo 4: Imposibilidad de contar con los recursos económicos necesarios.

- Severidad (S): 10. No contar con el dinero necesario para comprar los componentes del sistema impediría la realización del proyecto.
- Ocurrencia (O): 2. El presupuesto requerido para el proyecto es relativamente bajo y son recursos que ya se tienen reservados.

Riesgo 5: Falta de tiempo para concluir el proyecto.

- Severidad (S): 5. Si el proyecto no se concluye en las fechas planificadas, existe la posibilidad de que se retrase indefinidamente si se presentan otros proyectos más de mayor prioridad.
- Ocurrencia (O): 6. Por cuestiones tanto laborales como familiares el proyecto podría retrasarse de manera imprevista.

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

Riesgo	S	O	RPN	S*	O*	RPN*
Imposibilidad de obtener los componentes necesarios	7	3	21	-	-	-
El tamaño del modelo de ML puede sobrepasar la cantidad de memoria del microcontrolador	9	7	63	9	2	18
Imposibilidad de contar con instrumentación de laboratorio	3	2	6	-	-	-
Imposibilidad de contar con los recursos económicos necesarios	10	2	20	-	-	-
Falta de tiempo para concluir el proyecto	5	6	30	5	4	20

Criterio adoptado: Se tomarán medidas de mitigación en los riesgos cuyos números de RPN sean mayores a 25.

c) Plan de mitigación de los riesgos que originalmente excedían el RPN máximo establecido:
Riesgo 2: Se realizará un curso de ML en microcontroladores durante el periodo de vacaciones.

- Severidad (S): 9. Este riesgo sigue siendo muy severo por más conocimientos de ML que se posean.
- Ocurrencia (O): 2. Con los conocimientos adecuados sobre ML en microcontroladores se pueden optimizar estos algoritmos, de tal forma que pueden adaptarse a la cantidad de memoria del microcontrolador a utilizar.

Riesgo 5: Durante el tiempo planificado del proyecto se trabajarán los fines de semana y feriados 4 horas diarias.

- Severidad (S): 5. La severidad no se ve afectada por la medida de mitigación.
- Ocurrencia (O): 4. Las horas extras que se destinarán al proyecto podrían asegurar su conclusión en el tiempo planificado. Su ocurrencia baja muy poco porque no se descarta la posibilidad de que surjan imprevistos que retrasen el proyecto.

14. Gestión de la calidad

1. Requerimientos funcionales

- Req #1.1: el sistema debe conectarse a una red LoRaWAN en la banda de frecuencia AU915 como dispositivo de clase A.
 - Verificación: conectar el sistema a un servidor LNS (LoRaWAN Network Server) mediante un gateway LoRaWAN en la banda AU915 para comprobar a través de la UART que los paquetes son enviados correctamente y que los datos solo son recibidos mientras las ventanas de tiempo de recepción de los dispositivos clase A están disponibles.
 - Validación: conectar el sistema a una arquitectura IoT existente a través de un gateway LoRaWAN comercial en la banda AU915 para enviar y recibir datos.
- Req #1.2: el sistema debe ser alimentado mediante dos baterías de tipo AA.
 - Verificación: con ayuda de un multímetro de precisión medir que los parámetros eléctricos requeridos por los componentes del sistema sean los requeridos por sus datasheets.
 - Validación: visualizar el buen funcionamiento del sistema a través de los LEDs de estado disponibles.
- Req #1.3: el sistema debe poseer mecanismos de seguridad implementados tanto en hardware como en software para evitar su manipulación incorrecta.
 - Verificación: mediante herramientas proporcionadas por el fabricante de la tarjeta de desarrollo, comprobar que los sectores de memoria dedicados a la seguridad funcionen correctamente.
 - Validación: intentar leer la memoria de programa del sistema y comprobar que no es posible.
- Req #1.4: el sistema debe almacenar fotografías y videos obtenidos mediante su cámara en una memoria no volátil.
 - Verificación: verificar con ayuda de herramientas de depuración que los procesos de lectura y escritura entre el sistema y la tarjeta SD se realizan con éxito.
 - Validación: probar el sistema y posteriormente extraer la tarjeta SD para visualizar las imágenes obtenidas a través de un adaptador USB y una PC.
- Req #1.5: el sistema debe recibir su configuración inicial mediante interruptores físicos conectados al mismo.

- Verificación: mediante herramientas de depuración comprobar que los sectores de memoria donde se almacena la configuración de sistema correspondan con las posiciones de los interruptores físicos.
- Validación: configurar el sistema en sus diferentes modos a través de los interruptores físicos.
- Req #1.6: el sistema debe recibir sus parámetros de funcionamiento mediante la red LoRaWAN a la que está conectada.
 - Verificación: mediante herramientas de depuración comprobar que los sectores de memoria donde se almacena la configuración de sistema corresponda con los datos que se reciben por el servidor LNS.
 - Validación: configurar el sistema en sus diferentes modos a través de una plataforma web.
- Req #1.7: el sistema debe funcionar solamente si se detecta movimiento en el sector donde se encuentra instalada.
 - Verificación: con ayuda del puerto UART imprimir el estado del sistema y la detección de movimiento a través del sensor PIR del sistema.
 - Validación: determinar si los LEDs de estado del sistema son activados por el movimiento de algún cuerpo que se encuentre en el área del sistema.
- Req #1.8: el sistema debe ser compatible con la especificación LoRaWAN 1.0.2.
 - Verificación: utilizar herramientas para probar la compatibilidad de dispositivos LoRaWAN provistas por LoRa Alliance.
 - Validación: conectar el sistema a una red LoRaWAN existente que cumpla con la especificación LoRaWAN 1.0.2.
- Req #1.9: el sistema debe, con ayuda de su cámara, detectar la presencia de personas a través de un algoritmo de ML.
 - Verificación: probar el algoritmo de ML mediante simuladores y con ayuda del puerto UART comprobar que se detecta correctamente la presencia de personas.
 - Validación: probar el sistema en un entorno real de trabajo y con ayuda de una plataforma web comprobar la detección de personas.

2. Requisitos no funcionales

- Req #2.1: el sistema deberá tener un costo de desarrollo igual o menor a 200U\$S.
 - Verificación: realizar la suma de los costos de cada componente para determinar que no se superó el presupuesto previamente propuesto.
 - Validación: realizar un detalle de los costos del sistema.
- Req #2.2: el sistema deberá tener documentación adecuada sobre su uso y desarrollo.
 - Verificación: comprobar que los datos técnicos de los documentos sean consistentes con los datasheets y application notes de los componentes utilizados.
 - Validación: presentar los documentos del proyecto a los directos interesados (director del trabajo, docentes y jurados).

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:
 - Responsable: Mauricio Barroso Benavides.
 - Actividad: evaluar si se cumplieron los requerimientos del proyecto y las fechas planificadas.
- Identificación de las técnicas y procedimientos útiles e inútiles que se emplearon, y los problemas que surgieron y cómo se solucionaron:
 - Responsable: Mauricio Barroso Benavides.
 - Actividad: evaluar las herramientas y metodologías para el desarrollo de software y hardware del proyecto, para establecer las técnicas y procedimientos que se seguirán en proyectos futuros.
- Indicar quién organizará el acto de agradecimiento a todos los interesados, y en especial al equipo de trabajo y colaboradores:
 - Responsable: Mauricio Barroso Benavides.
 - Actividad: agradecer públicamente a los profesores y responsables de la maestría, y a todos los involucrados en la realización del proyecto durante la defensa formal del mismo.