



Cámara IoT para Detección Facial con conectividad Wi-Fi

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	6
3. Propósito del proyecto	6
4. Alcance del proyecto	7
5. Supuestos del proyecto.	7
6. Requerimientos	7
7. Historias de usuarios (<i>Product backlog</i>).	8
8. Entregables principales del proyecto	9
9. Desglose del trabajo en tareas	9
10. Diagrama de Activity On Node.	10
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	16

Registros de cambios

Revisión	Detalles de los cambios realizados	Fecha
0	Creación del documento	23 de junio 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 IoT para Detección Facial con conectividad Wi-Fi”, consistirá esencialmente en la implementación del prototipo de una cámara para detectar rostros en entornos IoT con conectividad Wi-Fi, 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 diciembre 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 difusión de las redes sociales y el auge del Internet de las Cosas (IoT, Internet of Things) han aumentado significativamente la cantidad de datos disponibles en todos los aspectos de la vida humana y nos han llevado a la era de Big Data. Este gran volumen de datos tiene el potencial de mejorar nuestra comprensión del estado del mundo y permitir predicciones más precisas de un estado futuro. Los sistemas biométricos, donde la detección de rostros representa un papel auxiliar fundamental para mejorar el reconocimiento de rostros, ya que los rasgos faciales capturan una gran parte de la individualidad de una persona. De hecho, la detección de rostros ha sido un área de investigación activa en el campo de la visión artificial durante más de tres décadas, principalmente debido a la innumerable cantidad de aplicaciones que requieren la detección de rostros como primer paso.

Hoy en día, los dispositivos móviles, como teléfonos inteligentes y tabletas, brindan a los usuarios varias aplicaciones relacionadas con la tarea de detección de rostros, como grabación de video inteligente, navegación, aplicaciones para mejorar la seguridad, anotación automática de contenido visual, computación afectiva, incluido el reconocimiento y seguimiento de rostros.

Sin embargo, la disponibilidad limitada de unidades de procesamiento gráfico (GPU, Graphics Processing Unit) en dichos dispositivos impone una limitación en el rendimiento de los algoritmos que se pueden usar de manera eficiente para la tarea de detección de rostros. Las GPUs integradas en estos dispositivos móviles son más lentas que las utilizadas en computadores de escritorio, con solo una fracción de memoria y esta restricción hace que la mayoría de los algoritmos de aprendizaje profundo (DL, Deep Learning) recomendados sean inadecuados para tales aplicaciones de detección y reconocimiento de rostros.

Entonces, el objetivo de este trabajo es diseñar e implementar una solución de bajo costo económico y bajo consumo eléctrico, que pueda detectar y contar rostros humanos a través de una cámara y con ayuda de algoritmos de DL, como paso previo para la tarea de reconocimiento de rostros.

El desarrollo de este proyecto está basado principalmente en el área de sistemas embebidos, pero también incorpora elementos propios de IoT y DL para lograr prestaciones adecuadas al contexto actual. El IoT, por sus siglas en inglés, se refiere a la conexión de dispositivos 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 (ML, Machine Learning) permite a los sistemas computacionales realizar predicciones y tomar decisiones en función de los datos de entrada. DL es un conjunto de algoritmos de ML que entrena a un sistema computacional para que realice tareas complejas a través de redes neuronales convolucionales (CNN, Convolutional Neural Networks) que simulan el funcionamiento de las neuronas en los humanos. En los últimos años la optimización que han tenido los algoritmos de ML y DL 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 planteados en el pasado.

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 1.

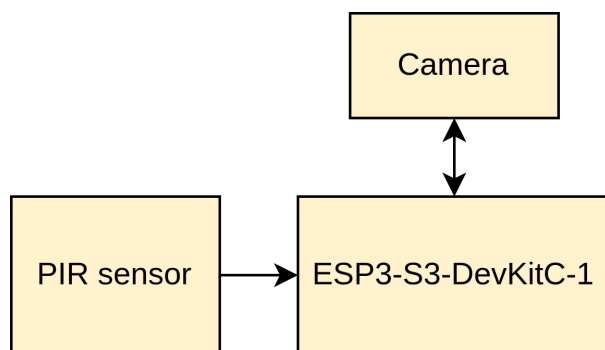


Figura 1. Diagrama en bloques del sistema

El sistema propuesto tiene como componente más importante una tarjeta de desarrollo ESP32-S3-DevKitC-1, cuyo elemento principal es un SoC (System on a Chip) ESP32-S3, que se encargará de ejecutar los modelos de DL, adquirir y procesar las imágenes obtenidas por la cámara, adquirir señales del sensor PIR (Passive InfraRed) y establecer comunicación con los servidores de AWS (Amazon Web Services) para enviar y recibir información. El sensor PIR ayuda a establecer si cerca del sistema existe algún tipo de movimiento para ejecutar las tareas previstas con el menor uso de energía posible. La cámara tiene la función de obtener imágenes y digitalizarlas para ofrecerlas al SoC.

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 mediante una cámara pueda detectar rostros humanos de las imágenes que obtiene con ayuda de uno o varios modelos de DL, para transmitir los resultados de sus predicciones a un servidor IoT.

Un propósito personal del autor es obtener conocimientos y experiencia sobre diseño e implementación de modelo de DL optimizados para sistemas embebidos.

4. Alcance del proyecto

Este proyecto incluye:

- Diseño e implementación del firmware del sistema embebido.
- Diseño e implementación de los modelos de DL para detección facial.
- Implementación del sistema embebido desarrollado dentro de una arquitectura IoT existente.
- Diseño e implementación de un dashboard para visualizar la información transmitida por el sistema embebido.
- Creación de documentación sobre datos técnicos y características del sistema embebido.

Este proyecto no incluye:

- Desarrollo 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 acceso a servicios en la nube 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 Wi-Fi existente a través de algún mecanismo de provisionamiento de credenciales de red.
- 1.2. El sistema debe establecer comunicación con los servidores de AWS.
- 1.3. El sistema debe ser alimentado mediante dos baterías AA de litio.
- 1.4. El sistema debe poseer mecanismos de seguridad implementados tanto en hardware como en firmware para evitar su manipulación incorrecta.
- 1.5. El sistema debe funcionar solamente si se detecta movimiento en el sector donde se encuentra instalada.

1.6. El sistema debe, a través de uno o varios modelos de DL, detectar la cantidad de rostros humanos presentes en una fotografía obtenida por su cámara.

2. Requerimientos no funcionales

2.1. El sistema debe tener un costo de desarrollo igual o menor a 200U\$S.

2.2. El sistema debe 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 en el futuro.

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

2. Como usuario quiero poder visualizar la información que genera el sistema en un dashboard que pueda ser accedido mediante internet.

- Complejidad: 3
- Tiempo: 5
- Recursos: 2

3. Como usuario quiero que la configuración del sistema se realice en la menor cantidad posible de pasos para evitar complicaciones en su puesta en marcha.

- Complejidad: 2
- Tiempo: 3

- Recursos: 1
- 4. 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 en el futuro.
 - Complejidad: 3
 - Tiempo: 2
 - Recursos: 1
- 5. Como desarrollador del hardware quiero que el sistema se componga de la menor cantidad posible de componentes electrónicos para optimizar el proceso y tiempo de desarrollo.
 - Complejidad: 4
 - Tiempo: 5
 - Recursos: 3

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. Recopilar y estudiar documentos de los componentes a utilizar. (30 hs)
 - 1.4. Planificar y diagramar la arquitectura a utilizar en el sistema (10 hs)
 - 1.5. Crear un repositorio en la nube para almacenar el código a desarrollar (2 hs)
2. Prototipo de pruebas (40 hs)
 - 2.1. Importar los componentes del sistema. (12 hs)
 - 2.2. Probar la funcionalidad de todos los componentes del sistema individualmente. (26 hs)
 - 2.3. Construir el prototipo de pruebas y asegurar sus conexiones. (2 hs)
3. Firmware (hs)
 - 3.1. Elegir las herramientas para desarrollar el firmware. (hs)
 - 3.2. Instalar y configurar las herramientas para desarrollar el firmware. (hs)

- 3.3. Desarrollar código para manejar la cámara del sistema. (hs)
- 3.4. Desarrollar código para manejar la conectividad Wi-Fi del sistema. (hs)
- 3.5. Desarrollar código para manejar el protocolo MQTT. (hs)
- 4. IoT (148 hs)
 - 4.1. Estudiar e implementar los servicios de AWS necesarios para el sistema. (hs)
 - 4.2. Integrar los servicios de AWS implementados con el firmware desarrollado. (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. DL (82 hs)
 - 5.1. Realizar cursos sobre DL y visión artificial . (hs)
 - 5.2. Investigar los modelos de DL aplicables al proyecto. (hs)
 - 5.3. Desarrollar los modelos de DL. (hs)
 - 5.4. Integrar los modelos de DL con el firmware desarrollado. (hs)
- 6. V&V (48 hs)
 - 6.1. Medir los parámetros eléctricos del sistema con instrumentación 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 2.

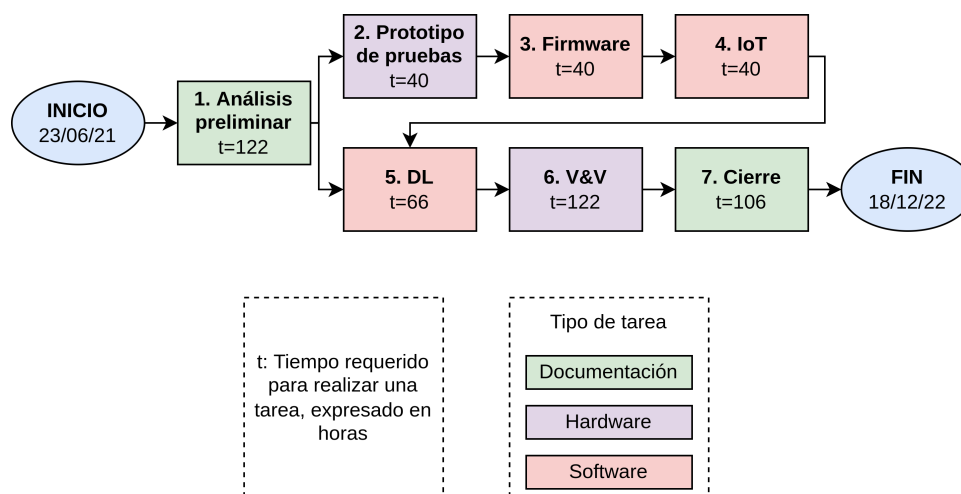


Figura 2. Diagrama en *Activity on Node*

11. Diagrama de Gantt

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

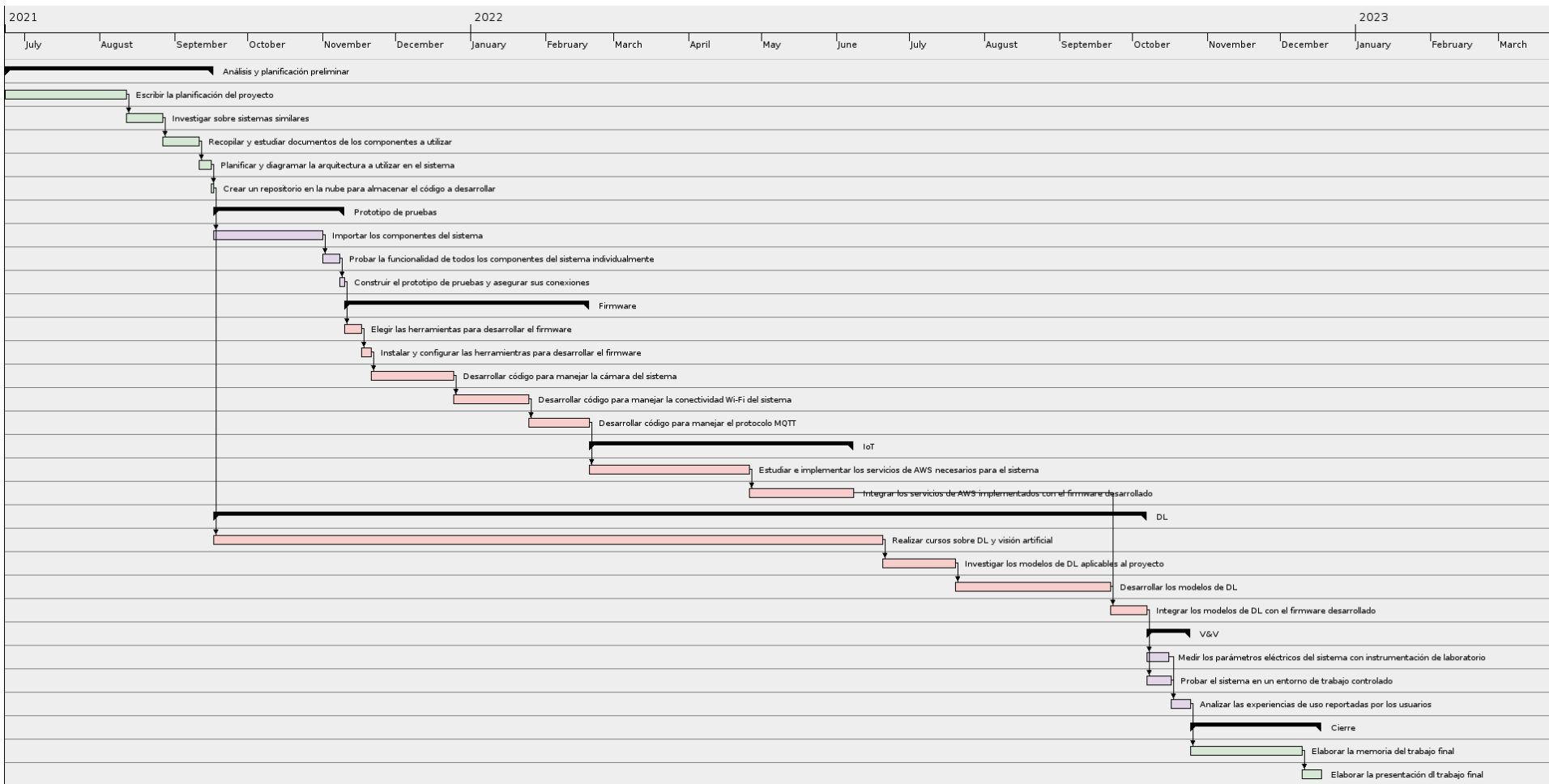


Figura 3. Diagrama de Gantt

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 ESP32-S3-DevKitC-1U-N8R8	2	15	30
Módulo con cámara OV2640	2	11,99	23,98
Sensor PIR IRA-S410ST01	2	3,49	6,98
Amplificador operacional TLV8544	2	1,82	3,64
Otros (cables, conectores, etc)	2	10	20
SUBTOTAL			84,6
COSTOS INDIRECTOS			
Descripción	Cantidad	Valor unitario	Valor total
50 % del costo directo	1	42,3	42,3
SUBTOTAL			42,3
TOTAL			126,9

Los valores expuestos en la tabla anterior están expresados en dólares americanos (U\$S). Esto es por que todos los componentes serán adquiridos de el distribuidor Mouser, ubicado 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 de los modelos de DL desarrollados puede sobrepasar la cantidad de memoria disponible.

- Severidad (S): 9. Si los modelos de DL no pueden ser almacenados en la memoria disponible, el sistema no podría realizar las funciones para las que fue planificado.
- Ocurrencia (O): 7. El tamaño de los modelos de DL alcanzarán un tamaño no adecuado en función de su complejidad y su optimización.

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 del autor 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 de los modelos de DL desarrollados puede sobrepasar la cantidad de memoria disponible	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 Wi-Fi existente a través de algún mecanismo de aprovisionamiento de credenciales de red.
 - Verificación: conectar el sistema a una red Wi-Fi existente a través de el mecanismo de aprovisionamiento de credenciales de red y comprobar a través de la salida de monitor del sistema.
 - Validación: conectar el sistema a una red Wi-Fi existente mediante una aplicación destinada para esto y comprobar su conexión a internet en la plataforma de AWS.
- Req #1.2: El sistema debe establecer comunicación con los servidores de AWS.
 - Verificación: mediante las herramientas de AWS para probar el protocolo MQTT, comprobar que los mensajes enviados desde el sistema son recepcionados correctamente.
 - Validación: con ayuda de un dashboard visualizar la información del sistema.
- Req #1.3: el sistema debe ser alimentado mediante dos baterías AA de litio.
 - 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 del LED RGB del sistema.
- Req #1.4: 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 #1.5: el sistema debe funcionar solamente si se detecta movimiento en el sector donde se encuentra instalada.
 - Verificación: con ayuda de la salida de monitor del sistema comprobar el estado del sistema y la detección de movimiento a través del sensor PIR.
 - Validación: comprobar que el LED RGB del sistema se active por el movimiento de algún cuerpo que se encuentre en el área de detección del sistema.
- Req #1.6: el sistema debe, a través de uno o varios modelos de DL, detectar la cantidad de rostros humanos presentes en una fotografía obtenida por su cámara.

- Verificación: probar los algoritmos de DL con un banco de pruebas compuesto por imágenes similares a las que serán adquiridas en un entorno real de trabajo, y comprobar su precisión con herramientas para la medición de métricas.
- Validación: probar el sistema en un entorno real de trabajo y monitorear los datos en los servidores de AWS.

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.