



“SISTEMA PARA EL MONITOREO, DETECCIÓN Y ALERTA DE
SOMNOLENCIA DEL CONDUCTOR MEDIANTE VISIÓN ARTIFICIAL,
COMUNICACIÓN INALÁMBRICA Y GEOLOCALIZACIÓN”

Proyecto Terminal 1

Autores:

Alan Eduardo Gamboa Del
Ángel
Maite Paulette Díaz Martínez
Nilsen Alfaro Parra

Asesores:

M.en C. Niels Henrik Navarrete
Manzanilla
Dr. Rodolfo Vera Amaro

03 de Noviembre 2022

Índice

1. Resumen	4
2. Introducción	7
3. Justificación	11
4. Estado del Arte	17
5. Objetivos	22
6. Marco Teórico	25
6.1. Red Neuronal	25
6.2. Redes Neuronales Convolucionales	25
6.3. Visión Artificial	28
6.4. Boosting	28
7. Diseño de la Unidad Contenedora del Módulo Central de Procesamiento	29
8. Búsqueda de Tecnologías de Geolocalización	35
9. Conclusiones	36
10. Bibliografía	37

Índice de figuras

1.	Modularidad	7
2.	Diagrama de Casos de Uso - Módulo Central de Procesamiento	8
3.	Diagrama de Actividades - Módulo Central de Procesamiento	9
4.	Diagrama de Secuencia - Módulo Central de Procesamiento	10
5.	Diagrama de Casos de Uso - Submódulo de Procesamiento de Imágenes	11
6.	Diagrama de Actividades - Submódulo de Procesamiento de Imágenes	12
7.	Puntos de referencia	14
8.	<i>Mouth Opening Ratio</i>	15
9.	Diagrama de Secuencia - Submódulo de Procesamiento de Imágenes	16
10.	Tipos de Redes Inalámbricas	20
11.	Mapa de la Ciudad de México	21
12.	Módulo estándar de una neurona artificial	25
13.	Boosting	27
14.	Boosting	28
15.	Especificaciones Físicas	29
16.	Zumbador Pasivo KY-006	30
17.	Cámara Pivariety IMX230	30
18.	Jumpers	31
19.	Micro SD	31
20.	RaspberryPi 3G/4G/ LTE Base HAT	32
21.	LE 910Cx Mini PCIe Series Linux	32
22.	LTE Main Diversity GNSS Triple Port u.FL Antenna de 100mm	33
23.	Unidad Contenedora del Módulo Central de Procesamiento	34

Índice de tablas

1.	Tabla Comparativa Lenguajes de Programación	6
2.	Agrupación de Puntos de Referencia	14
3.	Ventajas, Desventajas y Aplicaciones de las LPWAN	19
4.	Comparación entre Redes Inalámbricas	21
5.	Tecnologías de Geolocalización	35

1. Resumen

El presente documento detalla el proceso de diseño del proyecto titulado "Sistema para el monitoreo, detección y alerta de somnolencia del conductor mediante visión artificial, comunicación inalámbrica y geolocalización como Proyecto Terminal.

Uno de los factores importantes a considerar al momento de elegir un lenguaje de programación orientado a machine learning, es la popularidad el mismo, ya que esta es una señal de la aceptación por parte de la comunidad. A su vez, su el soporte es tanto o mas importante, ya que podemos darnos una idea si dicho lenguaje posee las herramientas adecuadas que se acoplan a nuestras necesidades. La velocidad de ejecución es otro factor importante, sobre todo cuando se requiere una minusiocidad en la ejecución de procesos y el cuidado de la memoria .Finalmente, la versatilidad del lenguaje es otro factor relevante, ya que, si el lenguaje fue diseñado con una determinada tarea o propósito en mente, este será mucho más eficiente y productivo.

A continuación, se listan algunos de los lenguajes de programación más populares en el campo de machine learning:

- **Python**

Python se ha convertido en el lenguaje de programación más popular del mundo, esto gracias a su sencillez aunado a una gran versatilidad. Sus utilidades abarcan desde desarrollo web, hasta la inteligencia artificial. Python es uno de los lenguajes más naturales y con sintaxis menos complicada. Por lo tanto, es fácil de aprender, leer y corregir errores. Lo mejor de todo es que Python es un lenguaje de código abierto. Lo que significa que es de libre acceso y distribución. Como resultado, tiene bibliotecas versátiles.

Una biblioteca es una colección de módulos o códigos creados para realizar operaciones específicas. Algunos módulos pueden ayudar con la interfaz gráfica de usuario, análisis de datos, gráficos, entre otras cosas. En lugar de escribir el mismo código cada vez que lo necesite, puede importarlo desde la biblioteca.

Uno de los principales factores de esta popularidad, es el enorme ecosistema de herramientas desarrolladas específicamente para machine learning tales como: Pandas, Numpy, Tensorflow, Keras, entre otras. A pesar de no ser el lenguaje más rápido o potente, lo anterior lo hace uno de los lenguajes indispensables si se piensa trabajar en el campo de machine learning.

- **C++**

A menudo hemos escuchado que C++ se aborda más cuando se trata de desarrollar juegos y sistemas grandes. Esto se debe a su característica de portabilidad y también proporciona una comprensión básica de la construcción lógica. Sin embargo, Algunos de los paquetes compatibles con C++ incluyen: Microsoft Cognitive Toolkit (CNTK) para aprendizaje profundo; Tensorflow para aprendizaje profundo; OpenCV para visión artificial; MLPack para aprendizaje automático; DyNet para redes neuronales; OpenNN para redes neuronales; Shogun para

aprendizaje automático y FANN para redes neuronales.

Sin embargo, C++ también tiene sus propias deficiencias. Está muy orientado a la sintaxis, a diferencia de Python, que es realmente amigable para los principiantes. Además de no contar con un gran soporte de bibliotecas.

■ R

Otro de los lenguajes de programación cuyo uso es considerablemente popular en el ámbito de machine learning y data análisis es R. R ofrece una variedad de beneficios desde su buen soporte de biblioteca y gráficos hasta la posibilidad crecer enormemente con muchos de los que podemos buscar ayuda, debido a su naturaleza abierta. Algunos de los paquetes clave compatibles con R son: Kernlab para operaciones basadas en regresión y clasificación; Ggplot para visualización de datos; Caret para operaciones basadas en regresión y clasificación; Plotly para visualización de datos; MLr3 para diferentes flujos de trabajo de aprendizaje automático. al igual que Python, también es comparativamente más lento que el resto de los lenguajes. Esto representa una gran desventaja para este lenguaje de programación. Además de esto, también tiene un origen débil, por lo que no es tan fácil de aprender. Para las personas que no tienen experiencia en programación les puede resultar difícil aprender R.

■ Matlab

Matlab también admite operaciones de aprendizaje automático y se puede aplicar de manera innovadora a aplicaciones que incluyen visión por computadora. MATLAB no está limitado por la sintaxis y, por lo tanto, es fácil de aprender y comprender. MATLAB también tiene muchas funciones predefinidas y una GUI para que los alumnos entiendan mejor las cosas. Además, viene con un compilador MATLAB que ayuda cuando se trata de codificar en el mismo. Además, cuando se trata de capacidades de aprendizaje automático, MATLAB lo admite de una manera única. MATLAB permite a los usuarios aplicar AutoML, lo que les permite aprovechar al máximo los modelos codificados optimizados y reducidos. Además, MATLAB también puede realizar la generación automática de código para el análisis de sensores y mucho más. MATLAB también viene con algunas restricciones. El lenguaje de programación no está fácilmente disponible o es de uso gratuito. De hecho, una versión de prueba está disponible solo por 30 días y publica que los usuarios deben comprar el paquete para un uso posterior. Por esta razón, uno puede encontrar solo investigadores como su público objetivo. No solo eso, el compilador de MATLAB es costoso de comprar y, por lo tanto, no tiene un soporte de comunidad / foro denso para ayudar a los usuarios y estudiantes.

■ Javascript

JavaScript puede no ser el lenguaje principal para desarrollo en Inteligencia Artificial, pero es gran complemento para utilizar herramientas relacionadas con Machine Learning, especialmente visualización de datos o procesos, El hecho de que TensorFlow.js se ejecute dentro

del navegador abre una gama de posibilidades interesantes tanto para empresas como para desarrolladores. Como los navegadores son un espacio interactivo: ofrece acceso a varios sensores, incluidas cámaras web y micrófonos, que pueden proporcionar imágenes y sonidos como entrada a cualquier modelo de aprendizaje automático. A pesar de todos los aspectos positivos, TensorFlow.js no tiene acceso predeterminado al sistema de archivos en el entorno del host del navegador. Esto limita los recursos de datos disponibles y puede imponer restricciones en el tamaño de los archivos y procesos.

A continuación se presenta una tabla comparativa de las características que se tomaron en cuenta para tomar una decisión sobre que lenguaje utilizar.

Lenguaje	Popularidad	Versatilidad	Velocidad	Soporte
Python	Alta	Alta	Moderada	Alta
C++	Baja	Alta	Alta	Moderado
R	Alta	Moderada	Moderado	Bajo
Matlab	Baja	Moderada	Moderada	Bajo
Javascript	Moderada	Moderada	Baja	Moderado

Tabla 1: Tabla Comparativa Lenguajes de Programación

Python es un lenguaje que permite trabajar más de un paradigma (programación estructurada y orientada a objetos). Normalmente cuando programamos un script sencillo aplicamos solo el paradigma de programación estructurada (Secuencia, selección e iteración), y cuando necesitamos programar un proyecto más complejo y queremos incluir los beneficios del paradigma orientado a objetos (reutilización de los componentes, facilidad de mantenimiento y modificación de objetos) utilizamos clases (POO).

Debido a que en el presente proyecto se realizará la integración de sistemas enfocados a Machine Learning y a su vez a la programación orientada a objetos, Python representa la mejor opción para ser implementado, ya que este lenguaje nos permite desarrollar ambos ámbitos. Por lo que se tomó la decisión de ser utilizado a lo largo de este proyecto.

2. Introducción

Una propiedad fundamental que nos ofrece el lenguaje de programación Python, y que estaremos utilizando de manera frecuente a lo largo de este proyecto, es la modularidad. Que no es más que dividir nuestros programas en partes más pequeñas (sub-programas), que habitualmente son llamados módulos. Las cuales deben tener un grado de independencia de la aplicación o programa principal. Cuando hablamos de independencia, nos referimos a que pueden ser utilizados por otras aplicaciones y no dependen de la aplicación original en sí.

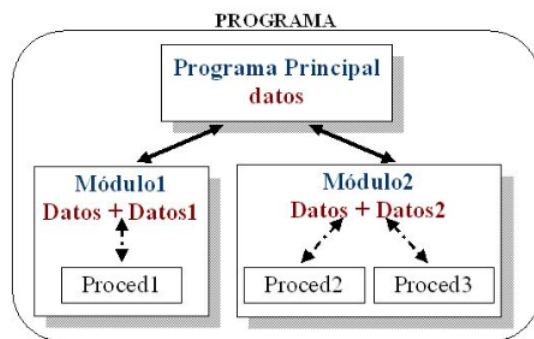


Figura 1: Modularidad

Estos módulos son archivos que alojan código independiente con una determinada función. Python cuenta con diversos módulos que podemos agregar a nuestro código y utilizarlo para facilitar y acortar el código, de manera que sea mas legible y ordenado.

Otro de los factores que consideramos al desarrollar este proyecto, es el de concurrencia de procesos. La concurrencia aparece cuando dos o más procesos ocurren al mismo tiempo. Estos procesos pueden competir.^o colaborar entre sí por los recursos del sistema.

Los temas de colaboración, sincronización, hilos y paralelismo se abordarán de una manera más profunda en el tercer reporte, cuando se tenga el diseño de los módulos faltantes.

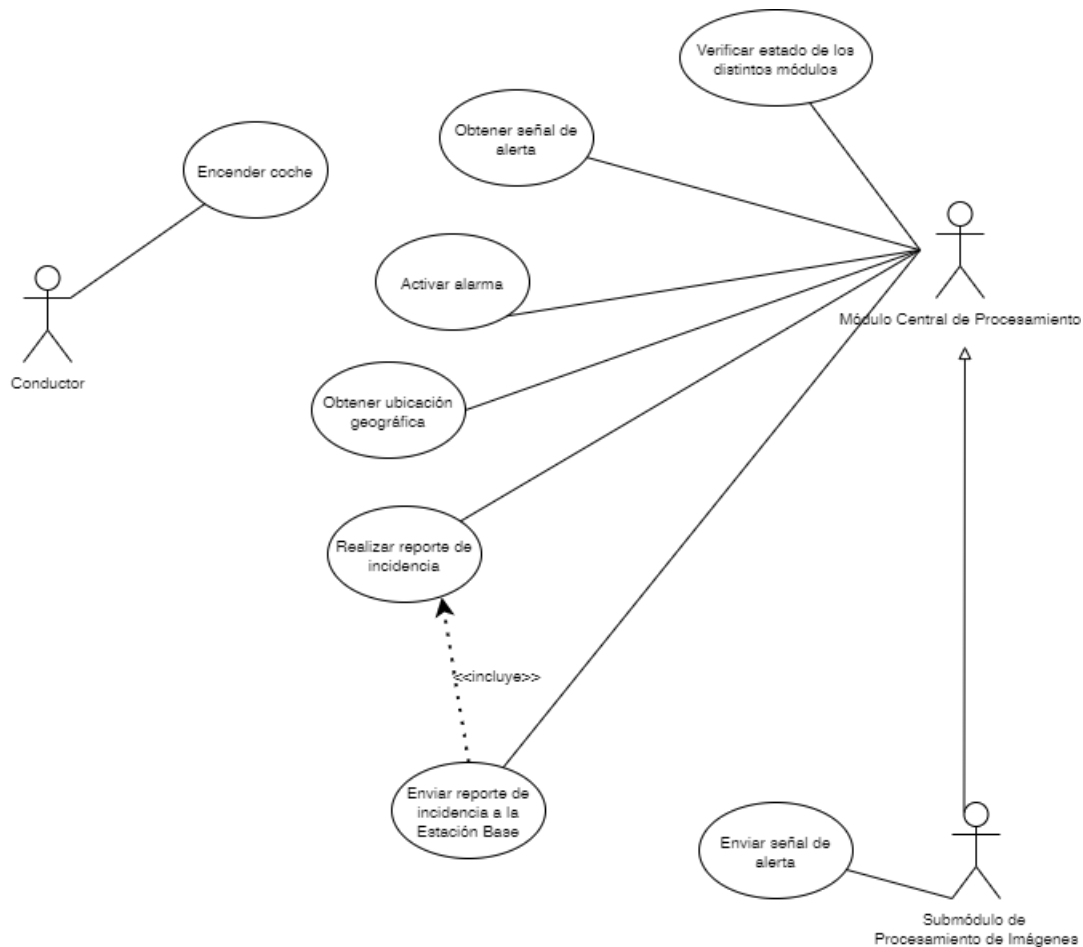


Figura 2: Diagrama de Casos de Uso - Módulo Central de Procesamiento

En el siguiente diagrama, se pueden observar las distintas tareas así como las interacciones entre estas y el Submódulo de Procesamiento de Imágenes.

Cabe recalcar, que para el presente reporte no se han determinado el diseño del Módulo de Telemetría, ni el de la Estación Base, por lo que se puede esperar el rediseño de este diagrama en un posterior reporte. Por esta misma razón, se decidió no incluir el diagrama de clases hasta que los Módulos faltantes estén diseñados.

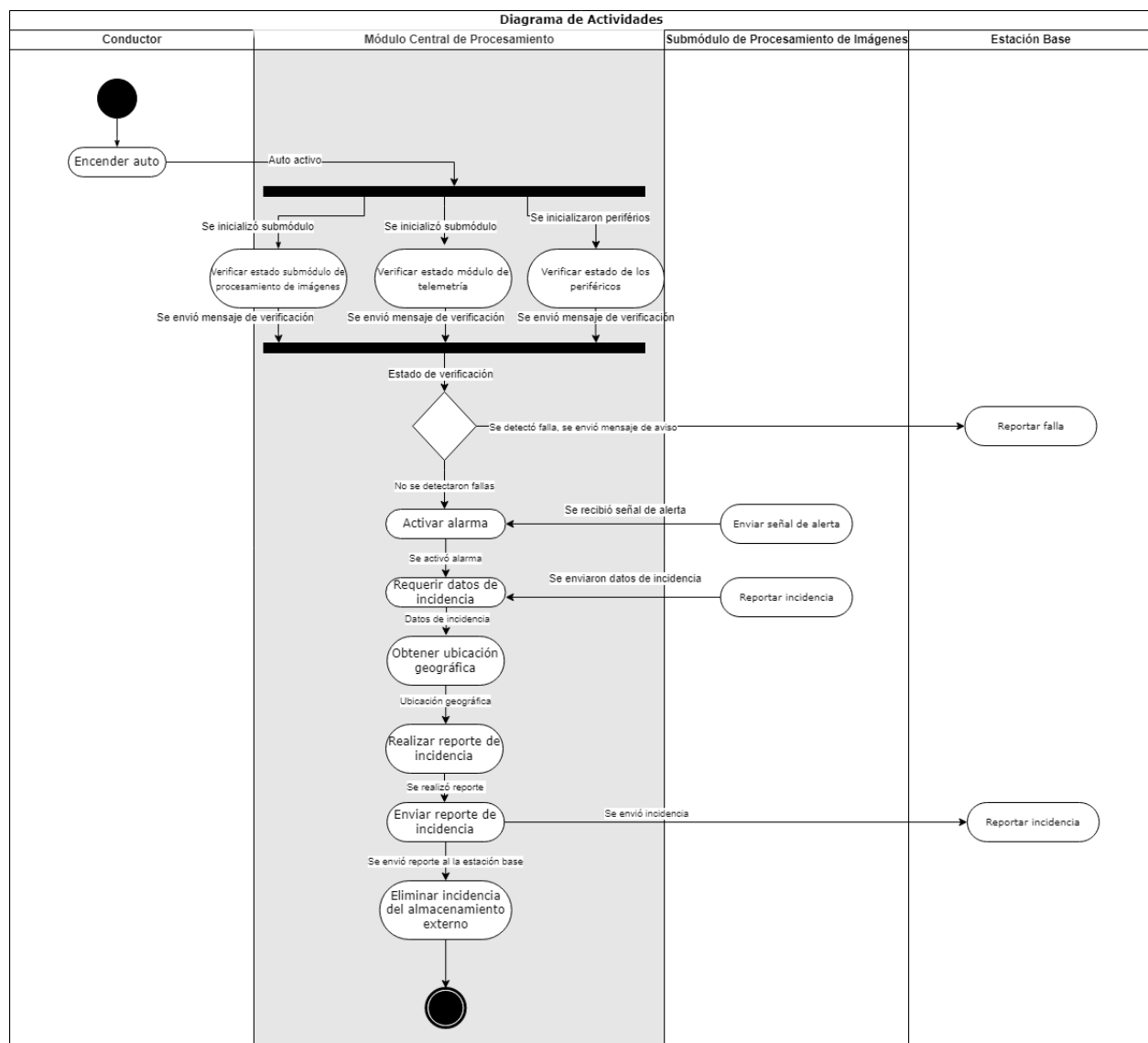


Figura 3: Diagrama de Actividades - Módulo Central de Procesamiento

En el siguiente Diagrama de Actividades se describen las actividades y sus interacciones con el Submódulo de Procesamiento de Imágenes y la Estación Base. Para que el sistema principal pueda iniciar, se necesita que el conductor encienda el auto, ya que este sistema funcionará utilizando la alimentación eléctrica.

El sistema principal comenzará por verificar el estado del Submódulo de Procesamiento de Imágenes, así como el de Telemetría y el estado de los periféricos. En caso de que cualquiera de estos presente alguna falla, se enviará un aviso a la Estación Base de que el sistema no podrá funcionar correctamente, y se etiquetará como "mantenimiento requerido". Si todos los sistemas funcionan correctamente, el Módulo de Procesamiento Central entrará en modo de espera por los datos pro-

porcionados por el Submódulo de Procesamiento de Imágenes. En caso de que este último envíe una señal de alerta de Somnolencia, el Módulo Central activará la alarma en forma de buzzer. Posteriormente, se obtendrá la ubicación geográfica con la ayuda del Módulo de Telemetría. se realizará un reporte de Incidencia que contendrá la fecha, hora, ubicación, y un pequeño videoclip del momento en que se detectó la somnolencia. Este será enviado a la Estación Base que se encargará de almacenarlo en su respectiva base de datos. Para evitar que el almacenamiento Interno del Módulo Central de Procesamiento se llene, se eliminará el reporte de incidencia siempre y cuando la Estación Base confirme que ha recibido dicho reporte. Como se indicó inicialmente el sistema estará disponible mientras el sistema se encuentre conectado a una fuente de alimentación, esto significa que el estado del sistema se encuentra conectado.

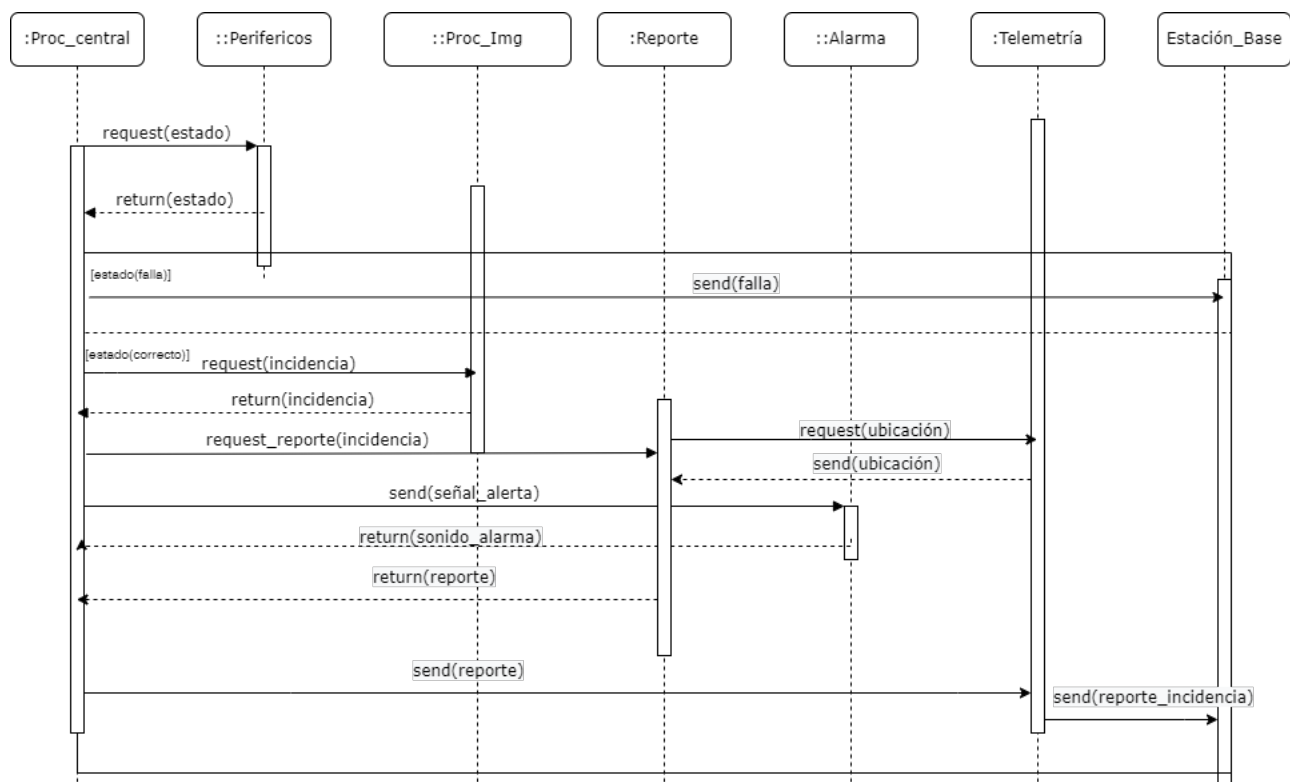


Figura 4: Diagrama de Secuencia - Módulo Central de Procesamiento

Siguiendo los procesos de la Figura 4, se procede a dar un análisis superficial en la concurrencia de los mismos. También se detallan las peticiones y respuestas de los distintos submódulos y sistemas.

3. Justificación

Para el diseño de la unidad de procesamiento nos apoyamos en la metodología UML, iniciando con un diagrama de casos de usos. Un modelo de casos de uso muestra una vista del sistema desde la perspectiva del usuario, por lo cual describe qué hace el sistema sin describir cómo lo hace. El siguiente diagrama muestra las funciones del Submódulo de Procesamiento de Imágenes basado en los requerimientos del sistema, así como la interacción con el Módulo Central de Procesamiento y el conductor, el cual es indispensable para poder realizar la detección de somnolencia.

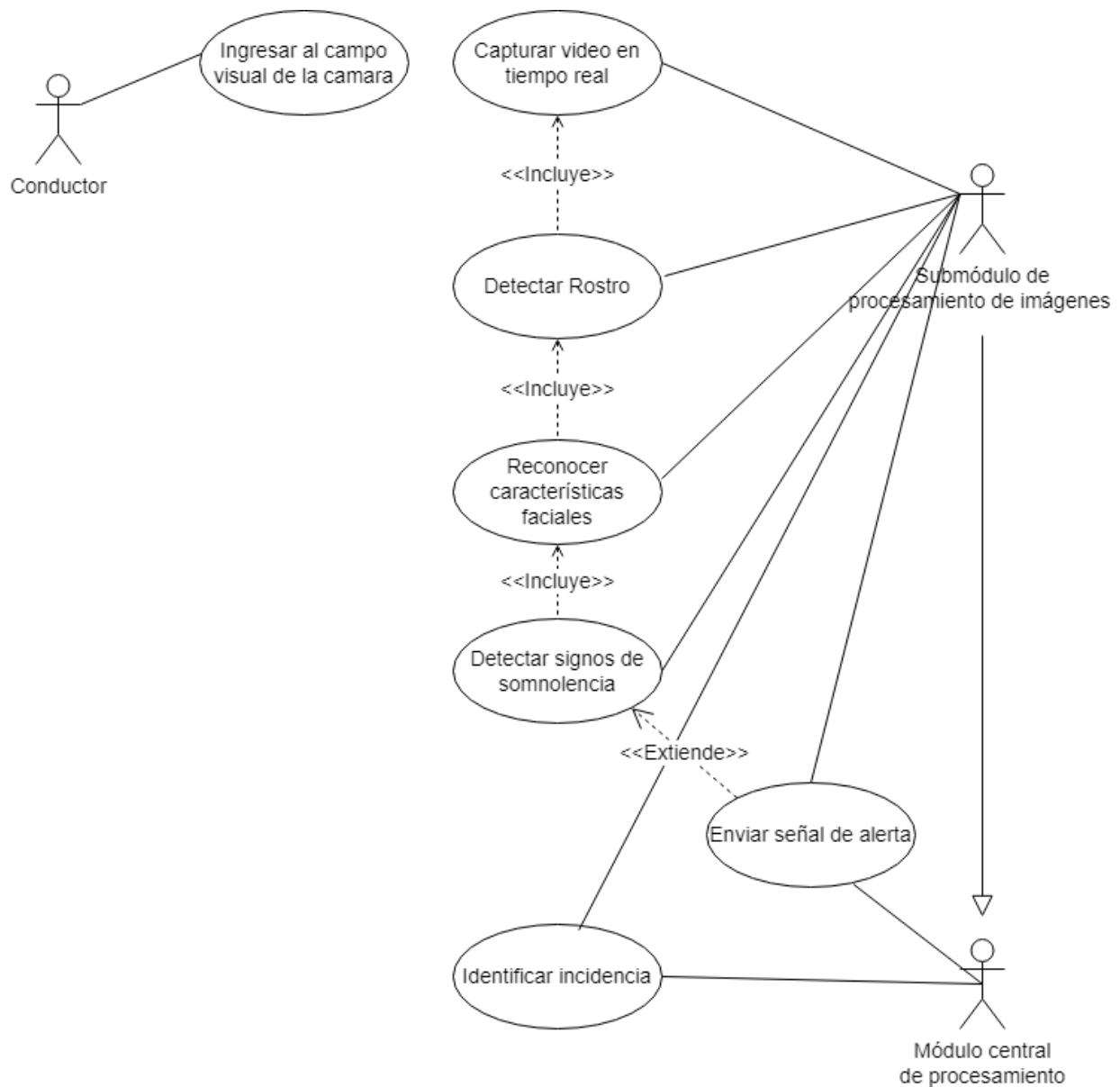


Figura 5: Diagrama de Casos de Uso - Submódulo de Procesamiento de Imágenes

A continuación se muestra el diagrama de actividades para el Submódulo de Procesamiento de Imágenes:

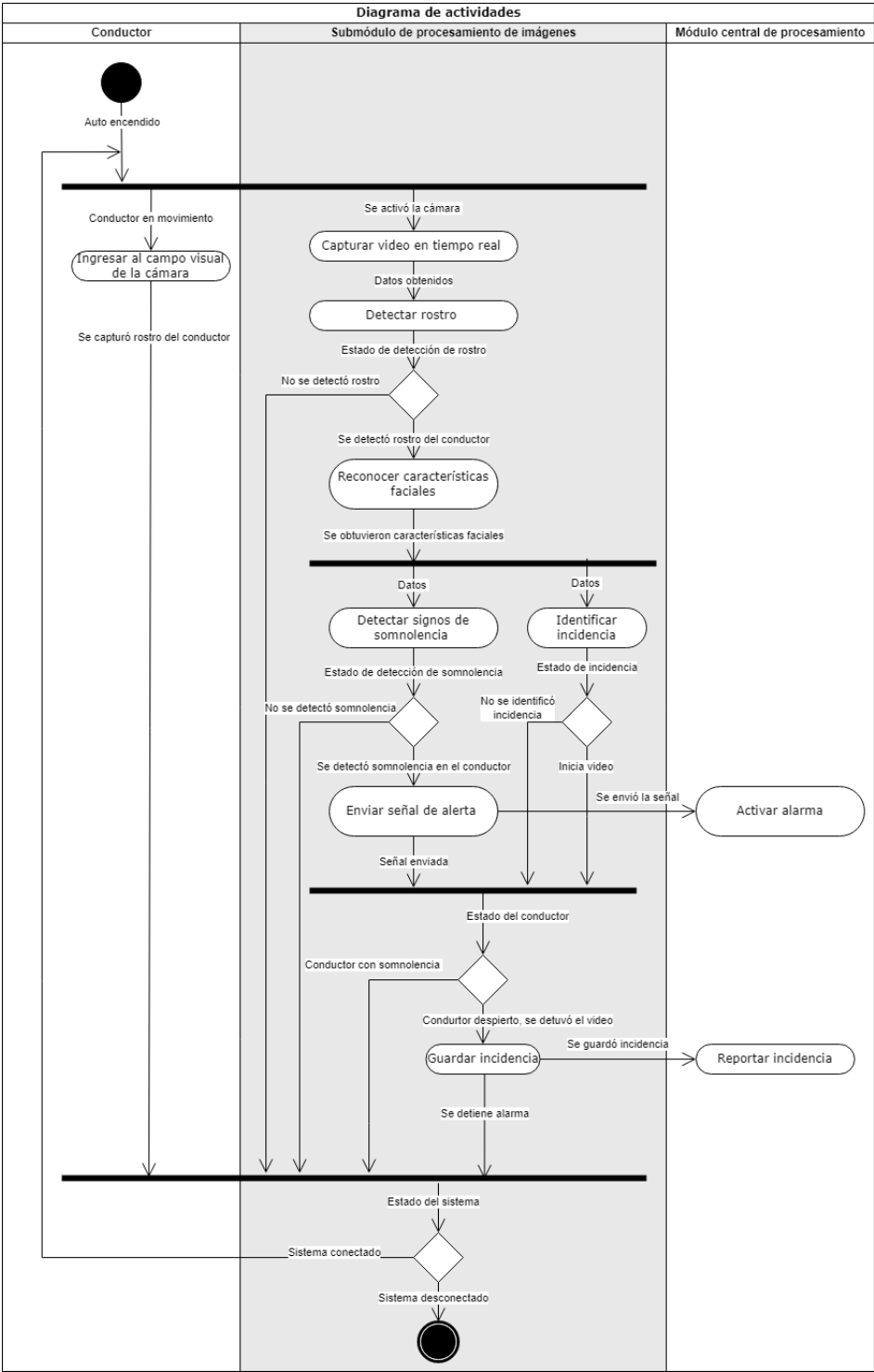


Figura 6: Diagrama de Actividades - Submódulo de Procesamiento de Imágenes

El diagrama de actividades muestra el flujo de las actividades en general, para que el sistema pueda funcionar requiere que el automóvil le proporcione una fuente de alimentación hacia el ordenador, el cual iniciara el sistema para la detección de somnolencia. Una condición importante es que el conductor se encuentre dentro del rango visual de la cámara y permanezca dentro del mismo al mismo tiempo que el Submódulo de Procesamiento de Imágenes realiza la captura de video utilizando una cámara digital. Posteriormente se realizará la detección del rostro utilizando el método Haar-Adaboost. También se utilizarán librerías como Dlib en OpenCV para entrenar al sistema, y que este sea capaz de detectar el rostro con condiciones de poca luz.

Se hará uso de una Red Neuronal Convolutiva para clasificar los ojos (abierto o cerrado) y así determinar el estado de somnolencia del conductor. Cabe mencionar, que el diseño de esta red se retomará en un reporte posterior, cuando se determine la resolución de óptima para detectar el rostro en tiempo real.

Mientras que para la boca se tomara la Relación de apertura de la boca (MOR).

En caso de que el sistema detecte somnolencia en el conductor se enviará una señal de alerta hacia el Módulo Central de Procesamiento el cual será el encargado de activar la alarma.

Como proceso paralelo a la detección de signos de somnolencia, la Identificación de una posible incidencia será la encargada de iniciar la grabación de video, con el fin de tener un fragmento de video segundos antes de detectar si el conductor presenta somnolencia.

En caso de que el conductor presente somnolencia, el video será detenido hasta detectar que el conductor mantiene los ojos abiertos por un intervalo de tiempo definido. Posteriormente el video de incidencia será guardado en la memoria y se notificará al Módulo Central de Procesamiento.

En caso de que el sistema indique que el conductor no presenta somnolencia el video será detenido y posteriormente eliminado de la memoria externa, con el fin de no saturar el espacio en la memoria. Como se indicó inicialmente el sistema estará disponible mientras el sistema se encuentre conectado a una fuente de alimentación, esto significa que el estado del sistema se encuentra conectado.

Puntos de Referencia

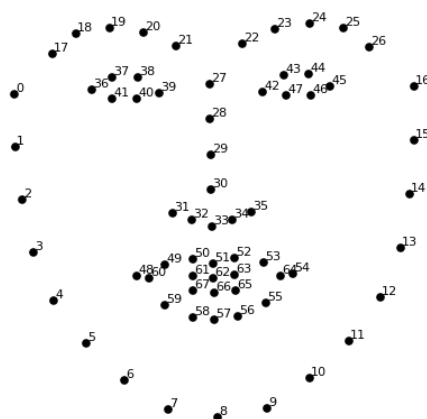


Figura 7: Puntos de referencia

A su vez, se utilizarán puntos de referencia con la ayuda de el software OpenCV y el modelo iBUG 300-W de delimitación de rostro para dividir las características que posteriormente serán extraídas y clasificadas.

En la siguiente tabla se muestra la delimitación y agrupación de los puntos de interés, tales como: ojo derecho, ojo izquierdo y boca.

Partes	Puntos de Referencia
Boca	[48-67]
Ojo Derecho	[36-41]
Ojo Izquierdo	[42-46]

Tabla 2: Agrupación de Puntos de Referencia

Mouth Opening Ratio

Bostezar es un signo cansancio. El bostezo se caracteriza por una apertura lenta y amplia de la boca. Por lo que se necesita medir el tamaño y la forma de la boca es necesario para identificar un bostezo. Para esto, se utilizara el *Mouth Opening Ratio* que es un método que utiliza puntos de referencia para medir la apertura de la boca. Entre más grande sea este valor, más es la apertura de la boca, por lo tanto se considera como un bostezo.

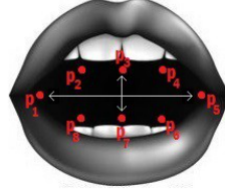


Figura 8: *Mouth Opening Ratio*

La formula general para calular el *MOR* es la siguiente:

$$MOR = \frac{\|p_2 - p_8\| + \|p_3 - p_7\| + \|p_4 - p_6\|}{2 \|p_1 - p_5\|} \quad (1)$$

Utilizando nuestros puntos de referencia, se podría calcular el MOR de la siguiente manera:

$$MOR = \frac{\|P49 - P59\| + \|P51 - P57\| + \|P53 - P55\|}{2 \|P48 - P54\|} \quad (2)$$

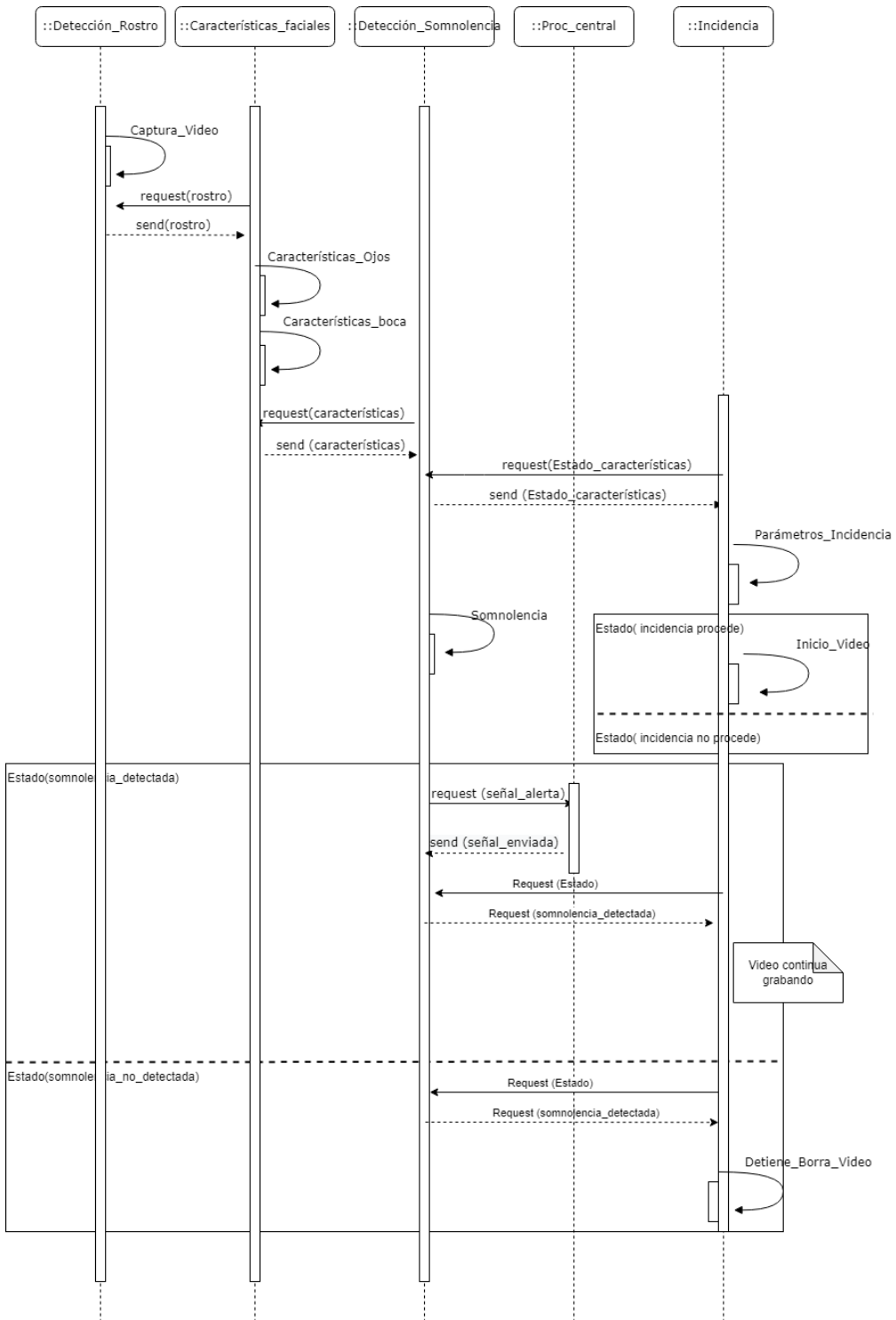


Figura 9: Diagrama de Secuencia - Submódulo de Procesamiento de Imágenes

4. Estado del Arte

Para transferir datos o información de un punto a otro sin la utilización de cableado o algún medio físico, tenemos las redes inalámbricas; y se dividen en 3 diferentes tipos dependiendo del alcance requerido:

- **La red de área personal inalámbrica (WPAN)**

Son redes de corto alcance que utilizan tecnología Bluetooth, HomeRF (*Home Radio Frequency*), Zigbee y las conexiones infrarrojas. Interconectan dispositivos compatibles como teléfonos móviles, electrodomésticos, impresoras o asistentes personales digitales (PDA) cerca de una ubicación central.

Utilizada para periféricos de tamaño pequeño, son conocidas como redes de corto alcance, con cobertura de 10 a 100 metros.

- **La red de área local inalámbrica (WLAN)**

Es una red de intercambio de datos a través de ondas electromagnéticas que se propagan en el aire, en ella se crea una red de área local con un determinado número de dispositivos conectados a un enrutador o punto de acceso que proporcionará acceso a Internet.

Son conocidas como redes de corto/mediano alcance alcanzando de 100 a 1000 metros.

Estándares usados en WLAN:

- IEEE 802.11
Es la forma más apropiada de referirnos a la tecnología Wi-Fi, especifica las normas de funcionamiento en una WLAN.
- IEEE 802.11a/b/g
Estándares identificadores de canales y frecuencias por donde se conectan los hosts a la WLAN. Con 802.11a se opera sobre las bandas de 5 GHz a 20 MHz y 2,4 GHz mientras que en la 802.11 b y g operan solamente en la banda de 2,4 GHz a una frecuencia de 25 MHz como ancho de banda.
- IEEE 802.11n
Fue el primero en implementar la tecnología MIMO (*Multiple Input - Multiple Output*) que permite usar varios canales a la vez para el envío y recepción de datos.
- IEEE 802.11ac
Se denomina WiFi 5 y la gran mayoría de aparatos trabajan sobre esta versión. Es una versión que solo opera en la banda de 5 GHz para proporcionar velocidades desde los 433 Mbps, dependiendo de la cantidad de antenas que utilice.

- IEEE 802.11ax

Es la nueva versión conocida como WiFi 6 o WiFi de 6ª generación en ella se introduce la nueva tecnología OFDMA? que mejora la eficiencia espectral de la red para WLAN en donde haya gran cantidad de usuarios conectados.

Opera sobre las frecuencias de 2,4 GHz y 5 GHz.

■ La red de inalámbrica de area amplia (WWAN)

Usan ondas de radio pero transmite a uno o varios puntos de acceso inalámbrico donde un usuario inalámbrico puede conectarse a la red, al disponer de un ancho de banda más elevado ofrece una mejor cobertura.

Como ejemplo de estas redes tenemos la tecnología 4G y 5G. Son conocidas como redes de largo alcance con cobertura de hasta 100km, pueden dar soporte a gran parte del territorio geográfico.

En esta red se incluye:

- **LPWAN(*Low Power Wide Area Network*): Red de Área Amplia de Baja Frecuencia**

Son conocidas como redes de largo alcance con cobertura de hasta 100km.

Ventajas, Desventajas y Aplicaciones de las LPWAN:

Ventajas	Desventajas	Aplicaciones
Baja Potencia. Minimiza el costo de energía lo que hace que los dispositivos remotos funcionen durante largos periodos de tiempo.	Baja velocidad de transmisión. No puede manejar grandes volúmenes de datos.	IoT. Satisfacer los desafíos de los dispositivos y sensores a pila.
Amplia cobertura. Ofrecen largo alcance sin mucha interferencia.	Problemas de atenuación de las señales. Funcionan de forma más eficiente en espacios abiertos que en zonas urbanas.	Aplicaciones de ciudades inteligentes. -Medidores de agua inteligentes -Iluminación inteligente
Bajo Ancho de Banda. Destinadas a aplicaciones con mensajes pequeños pocas veces por hora.	Confiabilidad en las transmisiones.	-
Bajo Coste. Al tener un bajo ancho de banda reduce su costo también muchos usan el rango sin licencia en las bandas ISM.	Conectividad no constante. Dificulta actividades como el control de movimiento.	-

Tabla 3: Ventajas, Desventajas y Aplicaciones de las LPWAN

A continuación, se muestra de manera gráfica los tipos de redes antes mencionadas, así como algunos de sus protocolos y tecnologías.

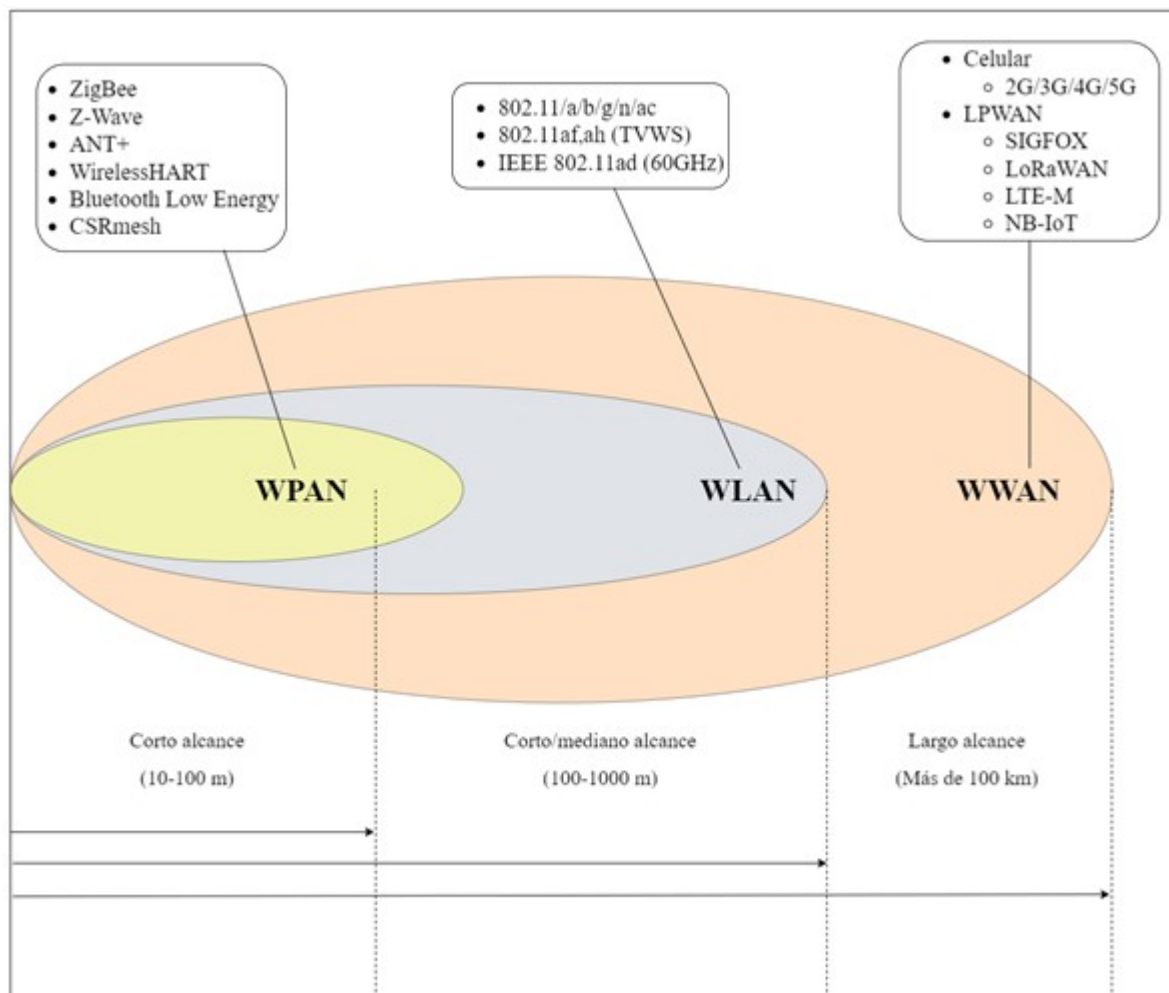


Figura 10: Tipos de Redes Inalámbricas

Dado que para el presente proyecto tiene contemplado funcionar dentro de la Ciudad de México y considerando que esta tiene una extensión de 1485 km², el tipo de red mas viable a utilizar es la LPWAN.



Figura 11: Mapa de la Ciudad de México

A continuación, se muestra un cuadro comparativo con estos tipos de redes.

	Espectro	Ancho de Banda	Costo	BitRate	Escalabilidad	Inmunidad a la Interferencia
LTE-M	Sin licencia 869 MHz 915 MHz	100kHz	Moderado	1Mbps	Alta	Moderada
NB-IoT	Sin licencia 433 MHz	<500kHz	Moderado	200kbps	Alta	Baja
SigFox	Con licencia 700-900 MHz	200kHz	Bajo	100bps	Baja	Baja
LoraWan	Con licencia 700-900 MHz	1.4MHz	Bajo	10kbps	Moderada	Muy Alta

Tabla 4: Comparación entre Redes Inalámbricas

Ya que una parte fundamental del proyecto es la transferencia de contenido multimedia, se requiere un bitrate mínimo de 0.5 Mbps.

Tomando esto en consideración, podemos observar que tanto SigFox y LoraWan no cumplen con este requisito, por lo tanto quedan descartadas para ser utilizadas.

Por lo tanto, las tecnologías que cumplen estas características son LTE-M y NB-IoT, por lo que en la siguiente sección se procede a hacer un análisis y comparación de estas dos tipos de redes inalámbricas para corroborar cual de las dos es la más óptima para ser utilizada en este proyecto.

5. Objetivos

LTE-M LTE-M es el término simplificado de la industria para el estándar tecnológico de Área Amplia de Baja Potencia (LPWA por sus siglas en inglés). Se refiere específicamente a la tecnología LTE CatM1, que es el estándar más adecuado para su uso con el IoT.

Es una tecnología de área amplia de baja potencia que admite IoT a través de dispositivos poco complejos y proporciona una cobertura extendida, al tiempo que permite la reutilización de la base instalada de LT

Las redes LTE-M coexistirán con las redes móviles 2G, 3G y 4G y se beneficiarán de todas las características de seguridad y privacidad de las redes móviles, como la confidencialidad de la identidad del usuario, la autenticación de la entidad, la privacidad, la integridad de los datos y la identificación del equipo móvil.

A continuación se describen las ventajas principales de LTE-M

- **Ahorro de Energía**

LTE-M permite que los dispositivos IoT usen un modo de ahorro de energía, que les permite ir a dormir cuando no están en uso. Y tampoco tienen que despertarse para enviar solo la actualización de ubicación, sino que pueden extender los tiempos de suspensión por un período de tiempo más largo. LTE-M también permite que los dispositivos usen recepción discontinua extendida (eDRX). Cuando el dispositivo está fuera del modo suspendido, verifica periódicamente la información del enlace descendente. Con eDRX, se aumenta el tiempo de verificación de la radio, lo que reduce el consumo de energía.

- **Cobertura**

Las redes LTE-M/Cat-M1/Cat-M2 pueden utilizar la infraestructura 4G LTE existente. Esta es una ventaja significativa, ya que más del 50 por ciento de las conexiones móviles globales se realizan en redes 4G.

Con una pérdida de acoplamiento máxima (MCL) de 156 decibelios (dB), 14 dB más alta que LTE, las redes LTE-M también ofrecen mayor cobertura y mejor penetración en interiores. La señal puede manejar muchas interferencias de edificios y otras estructuras que obstruyen su camino.

- **Velocidad de Transferencia**

Comparado con LTE, LTE-M no es particularmente rápido. Pero 1 megabit por segundo para transmisiones de enlace ascendente y descendente es excepcional en comparación con redes más antiguas como 2G y 3G (UMTS) y otras LPWAN. (Es varias veces más rápido que NB-IoT). Y eso es más que suficiente para la mayoría de las aplicaciones de IoT. Es factible para aplicaciones con necesidades de datos pequeñas e incluso casos de uso que involucran transmisión de video.

En el contexto de IoT, el alto rendimiento de datos significa que los dispositivos que usan LTE-M pueden recibir fácilmente actualizaciones por aire (OTA), y las transmisiones de datos consumirán menos energía, porque el dispositivo puede volver al modo de ahorro de energía más rápido.

- **Costo**

Las redes 4G se construyeron principalmente para teléfonos inteligentes. Las redes LTE-M se construyeron principalmente para dispositivos IoT. Como tal, los componentes necesarios para los dispositivos LTE-M son menos complejos y más asequibles que los componentes que necesitaría para un dispositivo 4G tradicional, aunque ambos utilizan la infraestructura 4G LTE.

NB-IoT

Es un protocolo inalámbrico de Internet de las cosas (IoT) que utiliza tecnología de red de área amplia de baja potencia (LPWAN). Fue desarrollado por 3GPP para comunicación inalámbrica celular que permite una amplia gama de nuevos dispositivos y servicios NB-IoT. NB-IoT es uno de los tres principales estándares 3GPP LPWAN.

El estándar de comunicación NB-IoT tiene como objetivo permitir que los dispositivos IoT funcionen a través de redes de operador, ya sea dentro de una onda portadora de comunicación del Sistema Global para Móviles (GSM) existente, en una "banda de protección" no utilizada entre canales LTE o de forma independiente.

Uno de los objetivos de NB-IoT es impulsar la extensión de la cobertura más allá de lo que ofrecen las tecnologías celulares existentes. Para ello, NB-IoT ofrece repeticiones de transmisión y diferentes configuraciones de asignación de ancho de banda en la transmisión de enlace ascendente.

La tecnología NB-IoT utiliza señales de bajo ancho de banda para comunicarse dentro de las tecnologías GSM y LTE existentes.

Los dispositivos y sensores especialmente diseñados son los componentes básicos de los sistemas NB-IoT. Estos dispositivos recopilan información de su entorno y la transmiten a estaciones base NB-IoT o nodos de transmisión.

Las estaciones base individuales están conectadas a una puerta de enlace de IoT y servidores de aplicaciones en la nube de IoT para el monitoreo centralizado y el análisis de datos.

NB-IoT emplea una nueva capa física con señales y canales para cumplir con los requisitos de cobertura extendida en áreas rurales e interiores profundos, al tiempo que permite una complejidad de dispositivo muy baja. La tecnología subyacente es mucho menos compleja que la de los módulos GSM/GPRS.

A continuación se describen las ventajas más importantes de NB-IoT:

- **Ahorro de Energía**

NB-IoT no necesita ejecutar un sistema operativo pesado, como Linux, ni hacer mucho procesamiento de señales, lo que lo hace más eficiente en términos de energía en comparación con otras tecnologías celulares.

- **Cobertura**

NB-IoT puede ayudar a admitir una gran cantidad de dispositivos mediante el establecimiento de redes NB-IoT que pueden conectarse a miles de millones de nodos. Diseñado para cobertura extendida en interiores, la menor complejidad de los dispositivos proporciona conectividad y comunicación de largo alcance.

- **Costo**

Debido a que es más fácil crear dispositivos con menor complejidad, el costo de los dispositivos es significativamente bajo, alrededor de 5 dolares por módulo.

- **Seguridad**

NB-IoT está protegido de manera muy similar a 4G, incluidas todas las funciones de autenticación basadas en cifrado y SIM.

6. Marco Teórico

6.1. Red Neuronal

Una red neuronal es un sistema que pretende emular ciertas características propias de los seres humanos, tales como la capacidad de memorizar o y asociar hechos o características. Este sistema está se basa en el concepto de *neurona*.

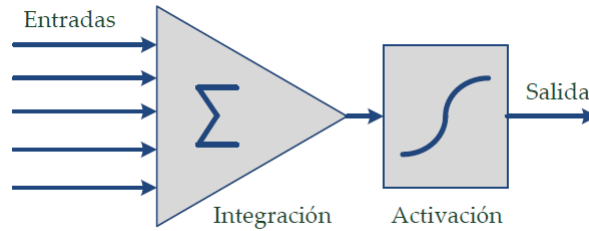


Figura 12: Modelo estándar de una neurona artificial

Un modelo simplificado de una neurona artificial consta de dos etapas. En la primera etapa, las entradas provenientes de otras neuronas son combinadas tomando en cuenta los pesos de las sinapsis. Como resultado de esta etapa surge la entrada neta o *excitación* de la neurona. En la segunda etapa, la entrada neta se utiliza para determinar el valor de salida de la neurona, que posteriormente será propagada a otras neuronas.

En la etapa de integración de las entradas, una neurona combina las distintas entradas x_i con sus pesos para así determinar su entrada neta z_j :

$$z_j = \sum_i w_{ij} x_i \quad (3)$$

Dónde w_{ij} representan los pesos sinápticos asociadas desde la i -ésima neurona hasta la j -ésima. Estos pesos tendrán valores real. Positivos para modelar conexiones excitatorias y negativos para conexiones inhibitorias.

Por otra parte, en la etapa de activación de una neurona, esta utiliza el valor asociado a su entrada neta para generar una salida y_j :

$$y_j(t) = F(y_j(t-1), z_j(t)) = F(y_j(t-1), net_j(t)) \quad (4)$$

6.2. Redes Neuronales Convolucionales

Las redes neuronales convolucionales, tambien conocidas como *redes convolutivas*, son redes neuronales artificiales que se utilizan comunmente para resolver problemas que requieren el procesamiento de imágenes. Sus casos de uso más frecuentes van desde la detección de objetos, hasta generar una descripción textual del contenido de una imagen. Particularmente, sus entradas y salidas pueden ser estructuradas. Esto quiere decir, que en lugar de recibir un vector de entradas, se

puede recibir un vector (1D), matriz (2D) o tensor ($> 2D$). En el caso de señales bidimensionales, las entradas pueden pertenecer a los píxeles de una imagen capturada por una cámara.

Como su nombre lo indica, esta red neuronal utiliza la operación de convolución. La convolución es una operación matemática que se realiza sobre dos funciones para producir una tercera que se suele interpretar versión modificada (filtrada) de las funciones originales.[1]

La convolución entre las funciones f y g se representa de la siguiente manera:

$$(f \star g)(t) = \int_{-\infty}^{\infty} f(\tau)g(t - \tau)d\tau = \int_{-\infty}^{\infty} f(t - \tau)g(\tau)d\tau \quad (5)$$

Cuando se habla de procesamiento digital de señales, se utilizan señales discretas, por lo que la integral anterior se convierte en sumatoria:

$$(f \star g)[n] = \sum_{m=-\infty}^{\infty} f[m]g[n - m] = \sum_{m=-\infty}^{\infty} f[n - m]g[m] \quad (6)$$

Comunmente, uno de los operandos de la convolución se trata de una señal que se desea procesar $x[n]$, y el otro corresponde al filtro $h[n]$, con el que se procesa la señal. Si el filtro es finito y es definido sólo sobre el dominio $\{0, 1, \dots, K - 1\}$ entonces para cada valor de la señal, se realizan K multiplicaciones y $K-1$ sumas:

$$(x \star h)[n] = \sum_{k=0}^{K-1} h[k]x[n - k] \quad (7)$$

En un caso multidimensional, si se tienen señales discretas definidas sobre dos variables y se le aplica un filtro de tamaño $K_1 \times K_2$, la convolución se calcula utilizando la siguiente expresión:

$$(x \star h)[n_1, n_2] = \sum_{k_1=0}^{K_1-1} \sum_{k_2=0}^{K_2-1} h[k_1, k_2]x[n_1 - k_1, n_2 - k_2] \quad (8)$$

En el caso particular de procesamiento digital de imágenes, las variables $[n_1, n_2]$ corresponden a coordenadas $[x, y]$ de los píxeles de una imagen. Además, el signo menos que aparece en la Ecuación 8, se suele sustituir por un signo más, por lo cual la definición de convolución se expresaría como:

$$(x \star h)[x, y] = \sum_{k_1=0}^{K_1-1} \sum_{k_2=0}^{K_2-1} h[k_1, k_2]x[x + k_1, y + k_2] \quad (9)$$

Supongamos que se tiene un imagen en blanco y negro. Dicha imagen, de 7×7 píxeles, se puede representar mediante una matriz binaria:

$$I = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Capas convolutivas

El componente clave de las redes neuronales convolucionales son las capas, que sustituyen la tradicional multiplicación de pesos por entradas. La entrada de una capa convolutiva es una señal, en el caso de imágenes, una señal bidimensional. Dicha señal es procesada realizando una convolución con una máscara o *kernel*. Los pesos correspondientes al *kernel* serán los parámetros de la capa convolutiva que será entrenada. A continuación se presenta la representación más simple de las capas de una red neuronal convolucional.

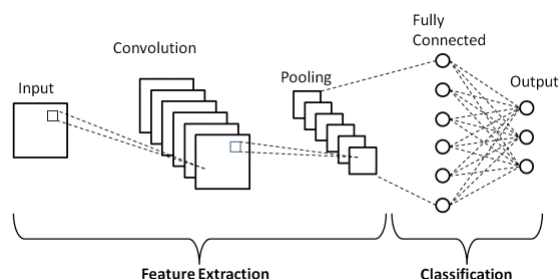


Figura 13: Boosting

- Capa Convolucional

Esta capa se utiliza para extraer las diversas características de las imágenes de entrada. En esta capa se realiza la convolución entre la imagen de entrada y un *kernel* de un tamaño particular $M \times M$. Al deslizar el filtro sobre la imagen de entrada, se toma el producto escalar entre el filtro y las partes de la imagen de entrada con respecto al tamaño de dicho kernel.

La salida se denomina mapa de características, que nos brinda información sobre la imagen, como las esquinas y los bordes. Posteriormente, este mapa de características se alimenta a otras capas para aprender otras características de la imagen de entrada.

La capa de convolución en CNN entrega el resultado a la siguiente capa una vez que se aplica la en la entrada.

- Capa de *Pooling*

El objetivo principal de esta capa es disminuir el tamaño del mapa de características convolucionado para reducir los costos computacionales. Esto se realiza disminuyendo las conexiones entre capas y opera de forma independiente en cada mapa de características. Según el método utilizado, existen varios tipos de operaciones de agrupación. Básicamente resume las características generadas por una capa de convolución.

- *Fully Connected Layer*

La capa totalmente conectada consta de los pesos y sesgos junto con las neuronas y se utiliza para conectar las neuronas entre dos capas diferentes. La

imagen de entrada de las capas anteriores se aplanan y se alimenta esta capa. Luego, el vector aplanado pasa por unas pocas capas más donde normalmente tienen lugar las operaciones de las funciones matemáticas. En esta etapa comienza el proceso de clasificación.

- *Output Layer* Esta capa se encarga de mejorar el rendimiento de un modelo de aprendizaje automático, ya que evita el sobreajuste al simplificar la red. Descarta neuronas no esenciales de la red neuronal durante el entrenamiento.

6.3. Visión Artificial

Se puede definir como el conjunto de técnicas para la extracción de información de imágenes digitales. El tipo de información obtenida de una imagen puede ser de identificación, mediciones para navegación, o aplicaciones de realidad aumentada. El campo de visión artificial puede verse como una parte de la informática. La teoría de algoritmos y el aprendizaje automático son esenciales para el desarrollo de algoritmos de visión artificial.

6.4. Boosting

El término de boosting fue presentado por primera vez por Viola y Jones y fue utilizado en su técnica de detección de rostro. Esta implica entrenar una serie de clasificadores simples cada vez más discriminatorios y luego combinar su salida.

Boosting implica construir un *clasificador* $h(x)$ como una suma de valores de *aprendices débiles*:

$$h(x) = \text{sgn} \left[\sum_{j=0}^{m-1} \alpha_j h_j(\mathbf{x}) \right] \quad (10)$$

los aprendices débiles $h_j(\mathbf{x})$ son funciones extremadamente simples de la entrada. En la mayoría de las variantes de boosting, los aprendices débiles son funciones umbrales,

$$h(x) = a_j[f_j < \theta_j] + b_j[f_j \geq \theta_j] = \begin{cases} a_j & \text{si } f_j < \theta_j \\ b_j & \text{otro,} \end{cases} \quad (11)$$

que también son conocidos como *decision stumps* consideradas como la forma más simple de un árbol de decisiones.

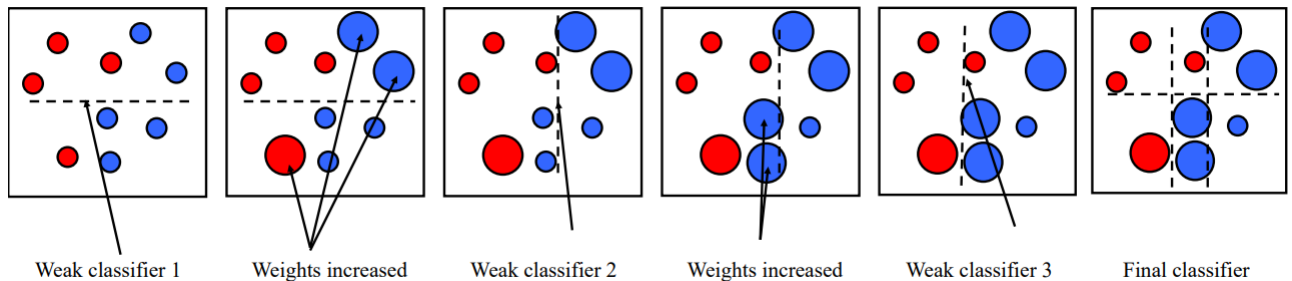


Figura 14: Boosting

Después de que cada clasificador débil es seleccionado, los *data points* que han sido clasificados de manera incorrecta aumentan su peso. El clasificador final es una combinación lineal de los clasificadores débiles.

Elementos físicos que contendrá la unidad:

De acuerdo con las especificaciones físicas de la Raspberry Pi 4 Modelo B se tiene las siguientes medidas en milímetros:

[illegible]

Figura 15: Especificaciones Físicas

Dimensiones 18 x 15mm.

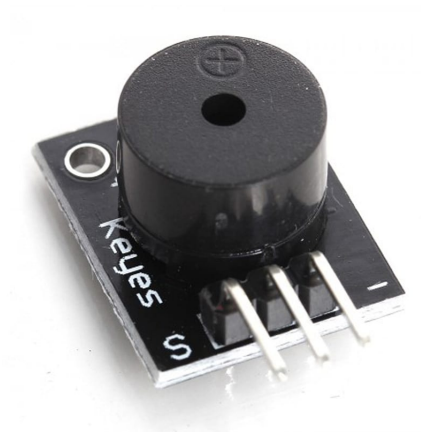


Figura 16: Zumbador Pasivo KY-006

- Cámara Arducam Pivariety IMX230
Dimensiones de la placa: 38 x 38mm.



Figura 17: Cámara Pivariety IMX230

- Cable macho-hembra



Figura 18: Jumpers

Se utilizarán 2 cables macho-hembra para la conexión del zumbador pasivo hacia los pines GND y Vcc del Raspberry Pi 4. Largo 10 cm.

- Micro SD
Dimensiones: 15 x 11 x 1 mm.

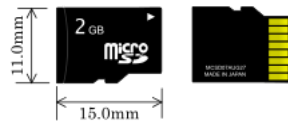


Figura 19: Micro SD

- **RaspberryPi 3G/4G/ LTE Base HAT**



Figura 20: RaspberryPi 3G/4G/ LTE Base HAT

Este HAT celular proporciona una conexión de datos simplificada para proyectos de IoT remotos, en el campo, en todo el mundo, en todas partes. Comience a usar una conexión LTE de alta velocidad con bajo consumo de energía en un factor de forma delgado con todo el software necesario para Raspberry Pi.

- **LE 910Cx Mini PCIe Series Linux**

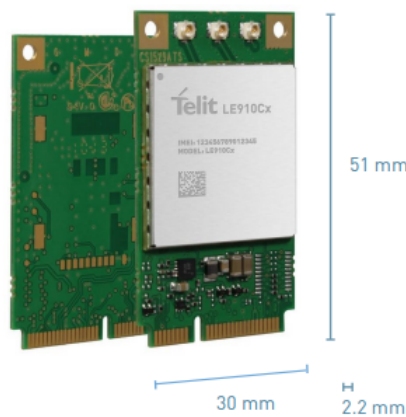


Figura 21: LE 910Cx Mini PCIe Series Linux

Los módulos Mini PCIe (mPCIe) de la serie LE910Cx son optimizadas para redes LTE de baja categoría y están disponibles en modo único de LTE o con opciones de respaldo 3G/2G.

- **LTE Main Diversity GNSS Triple Port u.FL Antenna de 100mm**



Figura 22: LTE Main Diversity GNSS Triple Port u.FL Antenna de 100mm

Antenas LTE principal y Diversity combinadas con una antena GNSS en una sola antena formada. Ajuste directo para el módulo Mini PCIe LTE de Quectel EC25, Telit LE910C1 y Telit LE910C4.

Tomando en cuenta las medidas anteriormente mencionadas, se presenta un bosquejo del diseño modular para la unidad contenedora, cabe aclarar que la siguiente propuesta puede presentar modificaciones durante el desarrollo del proyecto, con el fin de cumplir con las necesidades del sistema y presentar un diseño acorde a los cambios.

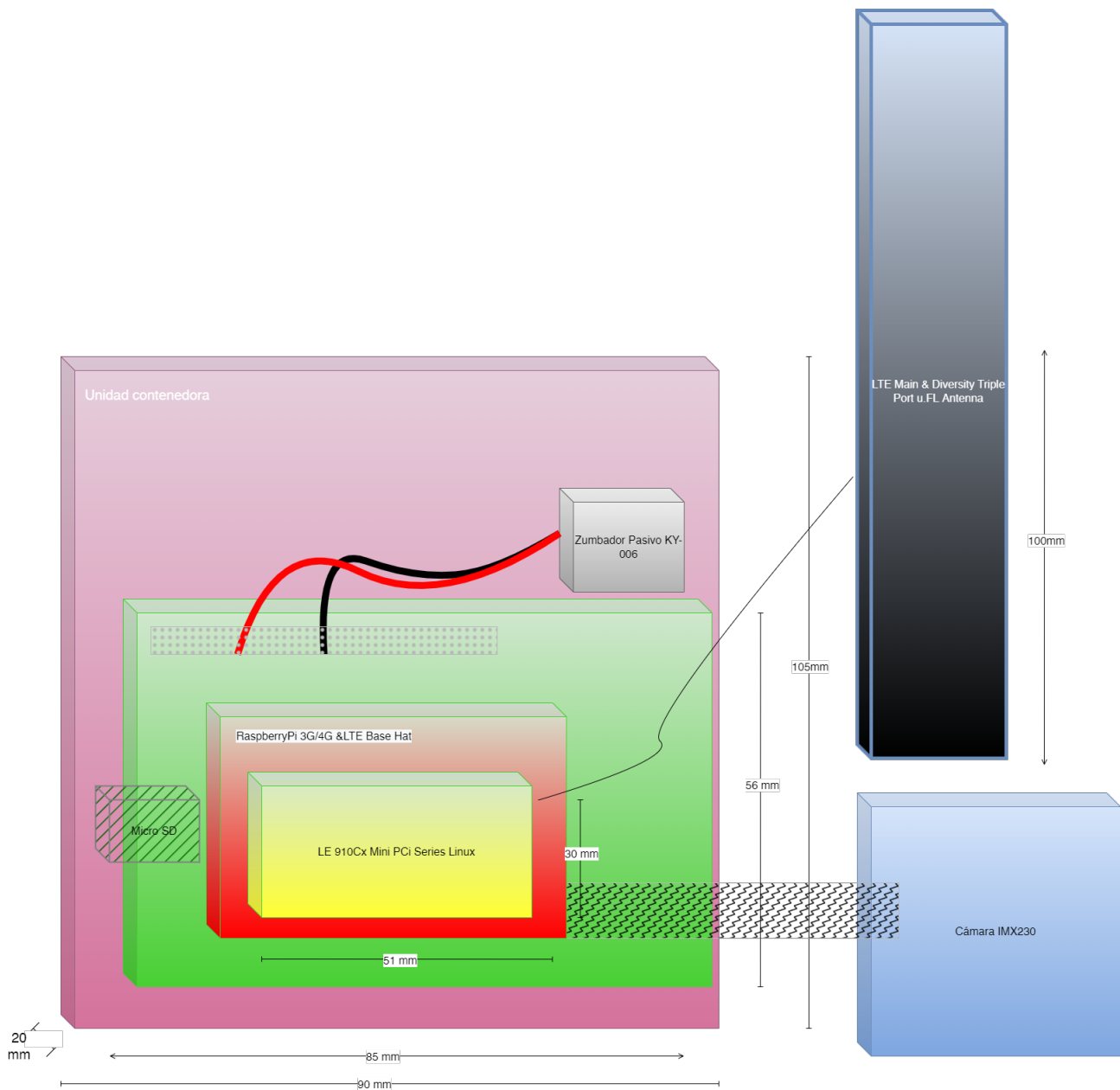


Figura 23: Unidad Contenedora del Módulo Central de Procesamiento

8. Búsqueda de Tecnologías de Geolocalización

La geolocalización implica el posicionamiento que define la localización de un objeto en un sistema de coordenadas determinado.

Los datos para la geolocalización se pueden obtener mediante la dirección IP de un usuario, su dirección MAC, el uso del RFID, la ubicación de la conexión WIFI o las coordenadas de GPS de su dispositivo.

Entre las diferentes tecnologías para la geolocalización, para este proyecto se destacan:

Tecnología	Descripción	Exactitud	Utilización Principal
<i>GPS(Global Positioning System)</i>	Sistema de posicionamiento global formado por una red de más de 30 satélites que orbitan alrededor de la tierra. Cada navegador GPS está conectado a diferentes satélites para tener los datos reales de la geolocalización en todo momento.	Es un sistema de geolocalización muy preciso, obteniendo la ubicación de manera exacta	Telefono móvil o en vehículos
<i>GSM(Global System for Mobile)</i>	Sistema de posicionamiento formado por las torres y antenas de telefonía	Es un sistema de geolocalización impreciso. Tiene un margen de error de hasta 200 metros	Comunicación móvil
<i>GPRS(General Packet Radio System)</i>	Sistema de posicionamiento de segunda generación GSM, proporciona servicios de alta velocidad	Es un sistema de geolocalización preciso pero limitado en área de cobertura	Solo disponible en tierra
<i>Wi-Fi</i>	Sistema de posicionamiento en el que todas las redes WIFI encendidas emiten una señal identificativa, comúnmente llamada dirección MAC. Este sistema usa 3 o más puntos de referencia para calcular la ubicación.	Es un sistema de geolocalización impreciso, presenta grandes márgenes de error	Interiores

Tabla 5: Tecnologías de Geolocalización

En un siguiente reporte, se analizarán con mas detalle las tecnologías que sean compatibles con la interfaz de LTE-M que utilizaremos a lo largo del proyecto.

9. Conclusiones

En un principio, se tenía contemplado utilizar la red LoRaWan para la comunicación y transferencia de archivos. Sin embargo, después de las actividades investigación, nos dimos cuenta de que esta opción no era viable, ya que el *bitrate* soportado por LoRaWan es insuficiente para la transferencia de archivos multimedia, que es una parte fundamental de nuestro sistema. Por tanto, se optó por utilizar la interfaz de LTE-M, una opción más costosa, pero que cumple los requisitos de una alta tasa de transferencia, así como una buena integración con la Raspberry Pi 4.

En el caso del Submódulo de Procesamiento de imágenes, se tenía contemplado desarrollarlo de manera separada del submódulo de *machine learning* pero después de haber realizado un análisis más profundo, se decidió integrarlo dentro de este mismo. Esto para el ahorro de recursos de procesamiento y conseguir una mejor comunicación entre procesos. En un siguiente reporte, se terminará por diseñar la Red Neuronal Convolucional, así como el diseño del Módulo de Telemetría y de la Estación Base.

Pará el diseño de la unidad de almacenamiento se obtuvo un bosquejo prototipo, sin embargo, se tiene contemplado rediseñar el modelo con el fin de mejorar y adaptarla los cambios del sistema a futuro.

10. Bibliografía

Referencias

- [1] Fernando Berzal, *Redes Neuronales y Deep Learning*, Granada, 2018
- [2] Irv Kalb, *Object-Oriented Python*, Primera Edición ,No Starch-Press, 2021
- [3] V.Moret Bonillo, *Fundamentos de Inteligencia Artificial*, Segunda Edición,Santiago de Compostela: Universidad de La Coruña Servicio de Publicaciones, 2005.
- [4] Ian Sommerville, *Ingeniería de Software*, Novena Edición , Pearson Educación de México, 2011
- [5] Kenneth E. Kendall, Julie E. Kendall, *Análisis y Diseño de Sistemas*, Octava Edición , Pearson Educación de México, 2011
- [6] Aurélien Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and Tensorflow*, Segunda Edición , O'Reilly, 2009
- [7] Kurt Demaagd, Anthony Oliver, Nathan Oostendorp y Katherine Scott *Practical Computer Vision with OpenCV*, Tercera Edición, O'Reilly, 2017
- [8] Cuno Plister, *Getting Started with Internet of Things*,Primera Edición, O'Reilly, 2011
- [9] Vilca Espinoza, R.A, *Influencia de un sistema de geolocalización en el control y monitoreo de vehículos con dispositivos GPS en una empresa logística*, 2007
- [10] S. Pachouly, N. Bhondve, A. Dalvi, V. Dhande y N. Bhamare. "Driver drowsiness detection using machine learning with visual behaviour". <https://ijert.org/papers/IJCRT2006408.pdf> (accedido el 2 de noviembre de 2022).