



“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

Índice

1. Resumen	5
2. Introducción	6
3. Planteamiento del problema	7
4. Propuesta de solución	8
5. Alcances y escenario de pruebas	9
6. Justificación	10
7. Metodología	11
8. Objetivos	12
8.1. Objetivo General	12
8.2. Objetivos Específicos	12
9. Marco Teórico	13
9.1. Raspberry Pi	13
9.2. Visión Artificial	13
9.3. Red Neuronal	13
9.4. Redes Neuronales Convolucionales	14
9.5. <i>Content Delivery Network</i>	16
10.Estado del Arte	17
10.1. <i>Driver Drowsiness Detection Using Machine Learning with Visual Behaviour</i>	17
10.2. <i>Detection Of Drowsiness And Distraction Of Drivers Using CNN</i>	17
10.3. <i>Driver Drowsiness Detection System Using Convolutional Neural Networks</i>	17
10.4. Diseño e implementación de sistema de visión artificial para alerta y detección de somnolencia mediante aprendizaje profundo aplicable en conductores de vehículos .	17
10.5. Sistema de detección de somnolencia mediante inteligencia artificial en conductores de vehículos para alertar la ocurrencia de accidentes de tránsito	18
10.6. Sistema para la detección del estado de somnolencia en seres humanos, con reconocimiento de patrones	18
10.7. Sistema de Detección de Somnolencia	18
10.8. Desarrollar un prototipo de reconocimiento facial basado en Machine Learning para detectar estado de Somnolencia en conductores de una cooperativa de transporte. .	18
10.9. Sistema basado en la detección y notificación de somnolencia en conductores de autos	19
10.10 Detección de somnolencia para conducción sin accidentes	19
10.11 Implementación de un sistema web con geolocalización para medir la transaccionalidad del portal ABC de la democracia para el consejo nacional electoral (CNE) . .	19
10.12 Diseño e implementación de un sistema de geolocalización en interiores para plataforma Android vía la red enterprise WLAN de la PUCP	19
10.13 Geolocalización con LoRa mediante multilateración	20

10.14	Propuesta de un sistema de geolocalización y monitoreo vía GPS/GSM/GPRS aplicado a un pulsómetro para personas con enfermedades cardiovasculares	20
10.15	Influencia de un sistema de geolocalización en el control y monitoreo de vehículos con dispositivos GPS en una empresa logística	21
10.16	Diseño de la red de acceso LTE en el distrito de Jesús María	21
11.	Análisis	22
11.1.	Análisis del Módulo Central de Procesamiento	23
11.1.1.	Análisis del Submódulo Visión Artificial	24
11.2.	Análisis del Módulo de Comunicaciones	25
11.3.	Análisis del Módulo de Estación Base	26
11.3.1.	Análisis de la Aplicación Web	32
12.	Diseño	42
12.1.	Diseño del Módulo Central de Procesamiento	42
12.1.1.	Diseño de la Unidad Contenedora de Procesamiento	44
12.1.2.	Diseño del Submódulo de Procesamiento de Imágenes	49
12.1.3.	Diseño del Submódulo de Deep Learning	51
12.2.	Diseño del Módulo de Comunicaciones	52
12.2.1.	Diseño del Submódulo de Transmisión de Datos	53
12.2.2.	Diseño del Submódulo de Telemetría	54
12.2.3.	Diseño del Submódulo de Cobertura	55
12.3.	Diseño de la Estación Base	56
12.3.1.	Diseño de la Aplicación Web	58
12.3.2.	Diseño de la Base de Datos	64
13.	Conclusiones	67
14.	Referencias	68

Índice de figuras

1.	Descomposición en la metodología <i>Top-Down</i>	8
2.	Descomposición en la metodología <i>Top-Down</i>	11
3.	Módulo estándar de una neurona artificial	13
4.	Arquitectura general de una Red Neuronal Convolutiva	15
5.	Ejemplo de Max Pooling	15
6.	Arquitectura de una CDN	16
7.	Boosting	22
8.	Diagrama de Casos de Usos del Módulo de Estación Base	29
9.	Diagrama de Casos de Uso	38
10.	Programa clásico vs Machine Learning	42
11.	Modularidad	42
12.	Especificaciones Físicas	44
13.	Zumbador Pasivo KY-006	44
14.	Cámara Pivariety IMX230	45
15.	Jumpers	45
16.	Micro SD	45
17.	RaspberryPi 3G/4G/ LTE Base HAT	46
18.	LE 910Cx Mini PCIe Series Linux	46
19.	LTE Main Diversity GNSS Triple Port u.FL Antenna de 100mm	47
20.	Unidad Contenedora del Módulo Central de Procesamiento	48
21.	Puntos de referencia	49
22.	<i>Mouth Opening Ratio</i>	49
23.	Arquitectura del Módulo de Estación Base	56
24.	Diagrama de Componentes AWS Amplify	57
25.	Diagrama de Secuencia Detalle Reporte Incidencia	58
26.	Diagrama de Secuencia Consultar Ubicación	58
27.	Diagrama de Secuencia Recuperar Contraseña	59
28.	Diagrama de Secuencia Confirmar Incidencia	59
29.	Diagrama de Secuencia Consultar Perfil	59
30.	Diagrama de Secuencia Registrar Conductor	60
31.	Diagrama de Secuencia Modificar Conductor	60
32.	Diagrama de Secuencia Eliminar Conductor	60
33.	Diagrama de Secuencia Estado de los Periféricos	61
34.	Página Inicio de Sesión	61
35.	Página Principal	62
36.	Página Perfil del Conductor	62
37.	Página Detalle de Incidencia	63
38.	Página Ubicación en Tiempo Real	63
39.	Estructura de un Documento en MongoDB	64
40.	Modelo propuesto para la base de datos en MongoDB	65
41.	Documento de Registro de Conductores	65
42.	Documento de registro de Incidencias	66

Índice de tablas

1.	RF01- Verificar estado funcional de periféricos	23
2.	RF02- Activar Alarma	23
3.	RF01- Capturar Video en Tiempo Real	24
4.	RF02- Delimitar región de Interés	24
5.	RF01- Guardar Incidencia	26
6.	RF02- Guardar Video	27
7.	RF03- Conectar Aplicación Web	27
8.	RF04- Desplegar Aplicación Web	28
9.	RF05- Guardar Credenciales de Usuario Administrador	28
10.	Requerimientos No Funcionales del Módulo de Estación Base	29
11.	Caso de Uso 01	30
12.	Caso de Uso 02	30
13.	Caso de Uso 03	30
14.	Caso de Uso 04	30
15.	Caso de Uso 05	31
16.	RF01- Iniciar Sesión	32
17.	RF02- Mostrar el Historial de Reportes de Incidencia	32
18.	RF02- Mostrar el Historial de Reportes de Incidencia	33
19.	RF03- Visualizar Reporte de Incidencia	33
20.	RF04- Confirmar Incidencia	34
21.	RF05 - Recuperar Contraseña	34
22.	RF06- Mostrar perfil del Conductor	35
23.	RF07- Mostrar ubicación Geográfica	35
24.	RF08- Descartar Incidencia	36
25.	RF09- Registrar Usuario	36
26.	RF10- Mostrar el Historial de Reportes de Incidencia	36
27.	RF11- Mostrar el Historial de Reportes de Incidencia	37
28.	RF12- Generar Mensaje de Alerta	37
29.	Requerimientos No funcionales - Aplicación Web	38
30.	Caso de Uso 01	39
31.	Caso de Uso 02	39
32.	Caso de Uso 03	40
33.	Caso de Uso 04	40
34.	Caso de Uso 05	40
35.	Caso de Uso 06	40
36.	Caso de Uso 07	41
37.	Caso de Uso 08	41
38.	Caso de Uso 08	41
39.	Agrupación de Puntos de Referencia	49

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.

2. Introducción

3. Planteamiento del problema

4. Propuesta de solución

Como respuesta a la problemática planteada en la sección anterior, se propone el desarrollo de un sistema portátil que sea capaz de detectar somnolencia en conductores y a su vez alertar con la intención de que este despierte.

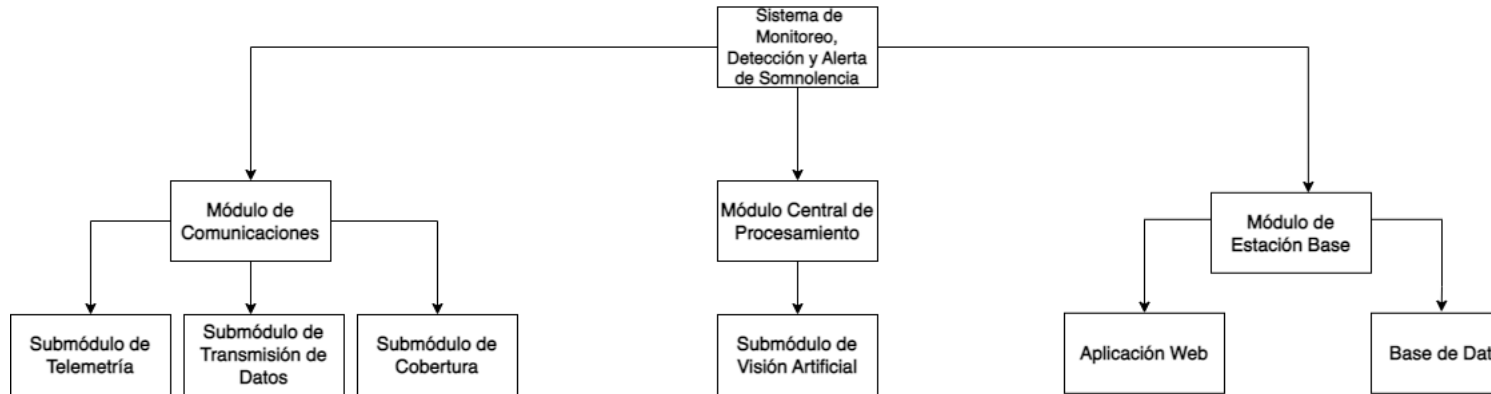


Figura 1: Descomposición en la metodología *Top-Down*

5. Alcances y escenario de pruebas

Para este trabajo, se decidió a delimitar el área de pruebas a la ciudad de México.

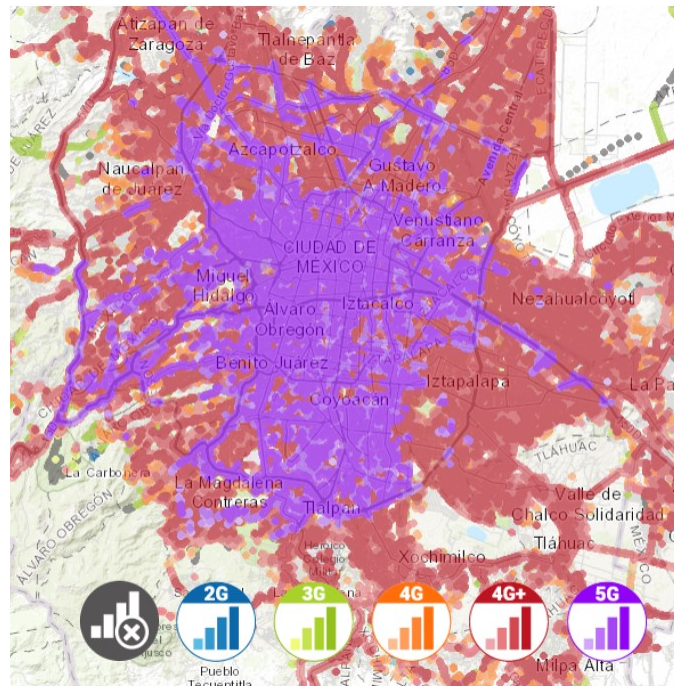


Figura 2: Descomposición en la metodología *Top-Down*

6. Justificación

De acuerdo con datos de CONAPRA (2013), en México cada año mueren en promedio 16,500 mexicanos por percances viales. Con base en estimaciones hechas por el Secretariado Técnico del Consejo Nacional para Prevención de Accidentes (STCONAPRA), los accidentes viales le cuestan al país alrededor de 150 mil millones de pesos que representan el 1.7 % del Producto Interno Bruto (PIB), sumando costos directos e indirectos.^[1]

El estado de somnolencia es una de las principales causas de los accidentes de tránsito, lo cual tiene consecuencias, no solo para los conductores, sino que también para los pasajeros y personas que circulan en la vía pública. La somnolencia al momento de conducir se trata de un fenómeno complejo que implica disminuciones en los niveles de alerta y conciencia por parte del conductor. Esto se convierte en un detonador de accidentes pues disminuye en el sujeto la capacidad de identificar situaciones de riesgo y por tanto evitar el peligro.^[1]

7. Metodología

Para el presente proyecto

■ Top-Down

La metodología de diseño *Top-Down* consiste en dividir un problema complejo en problemas o partes más pequeñas con mayor facilidad de resolución. A este proceso se le llama *Descomposición*. Estas partes pueden ser llamadas módulos o subrutinas. Cada módulo puede ser probado individualmente.

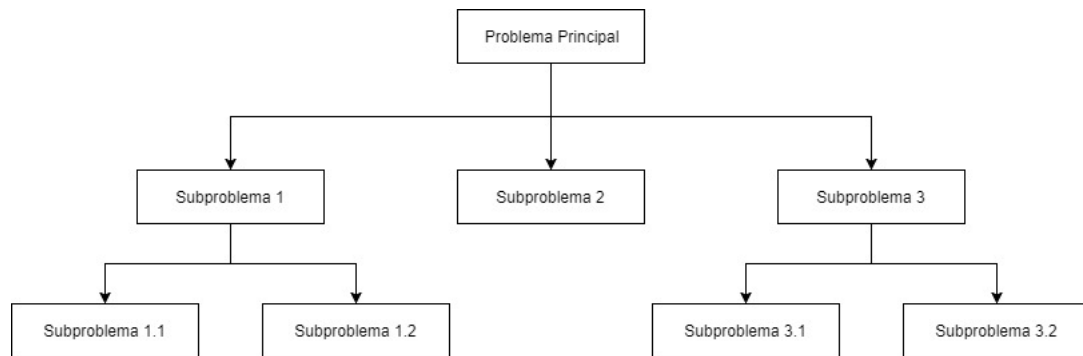


Figura 3: Descomposición en la metodología *Top-Down*

Por lo tanto, en este proyecto se comenzará entendiendo el problema general para posteriormente poder realizar una subdivisión de problemas más pequeños para ser resueltos de manera individual.

■ UML

UML es el estándar de la industria para modelar sistemas orientados a objetos[11]. UML incluye un conjunto de herramientas que permite visualizar la construcción de un sistema orientado a objetos. Al trabajar por iteraciones, se aborda de manera cada vez con más detalle el diseño de sistema, hasta que se defina con claridad las relaciones de cada uno de los objetos del sistema. Uno de las principales ventajas de UML es la reutilización de objetos, esto permite reducir costo de desarrollo así como de mantenimiento de los programas.

8. Objetivos

8.1. Objetivo General

Desarrollar e implementar un sistema para el monitoreo y detección de síntomas de somnolencia en conductores con vehículos particulares, con el fin de alertar al conductor. Además de implementar un sistema de telecomunicaciones y geolocalización que en conjunto con un sistema de administración lleve a cabo la gestión de registros de incidencias de cada conductor.

8.2. Objetivos Específicos

Para Trabajo Terminal 1:

- Diseñar un sistema de visión artificial que sea capaz de detectar la somnolencia en conductores.
- Diseñar un sistema de geolocalización en tiempo real usando redes de telecomunicaciones.
- Diseñar un sistema de administración para la gestión de usuarios, almacenamiento de datos y consultas de las incidencias de cada conductor.
- Diseñar un sistema de comunicaciones que permita la interconexión entre el sistema de visión artificial y el sistema de administración.

Para Trabajo Terminal 2:

- Implementar un sistema de visión artificial que sea capaz de detectar la somnolencia en conductores.
- Implementar un sistema de geolocalización en tiempo real usando redes de telecomunicaciones.
- Implementar un sistema de administración para la gestión de usuarios, almacenamiento de datos y consultas de las incidencias de cada conductor.
- Implementar un sistema de comunicaciones que permita la interconexión entre el sistema de visión artificial y el sistema de administración.

9. Marco Teórico

9.1. Raspberry Pi

La Raspberry Pi es una computadora de bajo costo y con un tamaño compacto, puede ser conectada a un monitor de computador o un TV, y usarse con un mouse y teclado estándar. Es un pequeño computador que corre un sistema operativo linux capaz de permitirle a las personas aprender a programar en lenguajes como Scratch y Python. Es capaz de hacer la mayoría de las tareas típicas de un computador de escritorio, desde navegar en internet, reproducir videos en alta resolución, manipular documentos de ofimática, hasta reproducir juegos. [1]

9.2. 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[5]. 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. La visión artificial está compuesta generalmente por un conjunto de procesos destinados a realizar el análisis de imágenes: captación de imágenes, memorización de la información, proceso e interpretación de los resultados[5].

9.3. 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*.

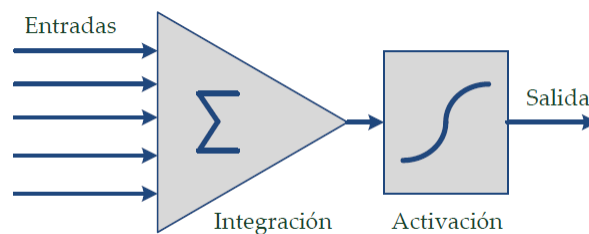


Figura 4: Modelo estándar de una neurona artificial

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 :

$$z_j = \sum_i w_{ij} x_i \quad (1)$$

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 :

$$y_j(t) = F(y_j(t-1), z_j(t)) = F(y_j(t-1), net_j(t)) \quad (2)$$

9.4. 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 pixeles 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.[7]

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

$$(f \star g)(t) = \int_{-\infty}^{\infty} f(\tau)g(t - \tau)d\tau = \int_{-\infty}^{\infty} f(t - \tau)g(\tau)d\tau \quad (3)$$

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:

$$(x \star h)[x, y] = \sum_{k_1=0}^{K_1-1} \sum_{k_2=0}^{K_2-1} h[k_1, k_2]x[x + k_1, y + k_2] \quad (4)$$

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.

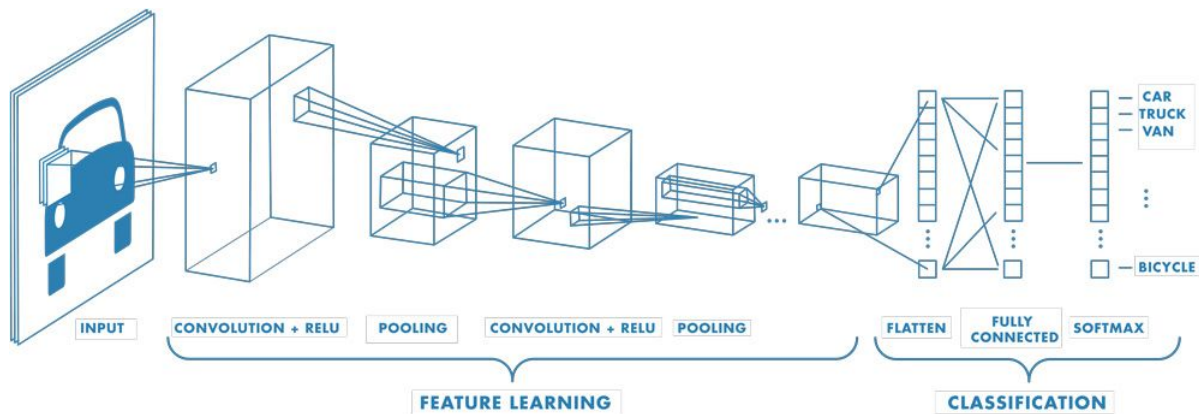


Figura 5: Arquitectura general de una Red Neuronal Convolutiva

■ Capa Convolutiva

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 $M \times M$.

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. El tipo de pooling más utilizado es el *max-pooling*. Este devuelve el valor máximo del fragmento de la imagen filtrada por el kernel.

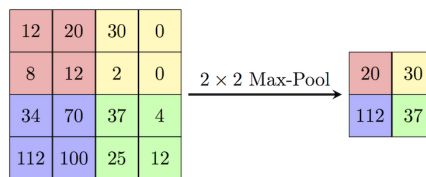


Figura 6: Ejemplo de Max Pooling

■ *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 aplanada y se alimenta a 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.

- *Funciones de activación*

Son funciones matemáticas que determinan la salida de una capa de la red neuronal, y se utilizan en capas convolucionales así como en las capas completamente conectadas. Estas funciones se utilizan para asignar los valores de las salidas de cada capa entre valores definidos, para así facilitar que el modelo se adapte a una gran variedad de datos.

- *Flattening*

Flattening consiste en convertir los datos de salida de la capa convolucional a una matriz unidimensional para ser llevada a la capa siguiente.

9.5. *Content Delivery Network*

Una Red de Distribución de Contenido, o CDN por sus siglas en inglés, es de una red de servidores que se encarga de mejorar el rendimiento en la carga y descarga de contenido de la red. Los componentes de una CDN se encuentran distribuidos en diferentes nodos. Estos componentes pueden ser servidores y se encuentran con contenido replicado alrededor del mundo. El trabajo de una CDN consiste en redirigir la solicitudes de los clientes a un servidor ubicado geográficamente cerca de estos. En consecuencia, los clientes obtienen los datos solicitados a una velocidad más rápida.

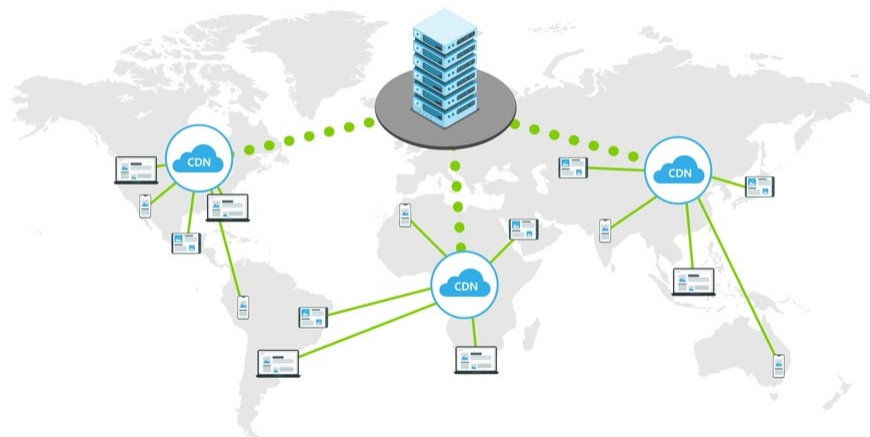


Figura 7: Arquitectura de una CDN

Además de esto, una CDN es capaz de detectar cambios en la información existente, así como detectar la disponibilidad de nuevo contenido en los servidores de origen.

10. Estado del Arte

10.1. *Driver Drowsiness Detection Using Machine Learning with Visual Behaviour*

Este trabajo de investigación realizado por el *International Journal of Creative Research Thoughts* propone un sistema de detección de signos de somnolencia en conductores utilizando un modelo de Red Neuronal Convolutacional para detectar la posición de los ojos, y OpenCV junto con Dlib para la detección de la boca y realizar el conteo del número de bostezos por minuto. Para alimetre a la red Neuronal, se utilizaron el conjunto de datos de NTHU-DDD. También fue utilizado el método PERCLOS para obtener el número de parpadeos del sujeto de estudio. Este trabajo concluye que las mayores dificultades a la hora de detección de rostros fueron el uso de gafas oscuras, así como cambios en la iluminación.

10.2. *Detection Of Drowsiness And Distraction Of Drivers Using CNN*

En este trabajo realizado por los estudiantes de la Universidad de Mumbai, se implementó el aprendizaje automático y el paquete Keras para construir un modelo de CNN, el cual, clasifica si el conductor se encuentra somnoliento o distraído, el sistema emite un tono de alerta al detectar correctamente la somnolencia, dando al conductor una alerta temprana. Se utilizó el clasificador Open CV Haar-Cascade, un clasificador en cascada basado en características, usando sus funciones integradas, se detectó el rostro y la región de los ojos.

10.3. *Driver Drowsiness Detection System Using Convolutional Neural Networks*

En este trabajo realizado en la Universidad Anurag en el año 2022 se presenta una forma de analizar y anticipar la somnolencia del conductor mediante la aplicación de una red neuronal convolutacional sobre la cara del conductor de un marco de secuencia. Se usó un conjunto de datos para dar forma y aprobar el modelo, usando redes convolucionales 3D basadas en modelos de múltiples capas de arquitectura de red neuronal repetitiva para detectar la somnolencia del conductor. Tras una sesión de entrenamiento, se obtuvo una precisión que se acerca al 92 % de aceptación.

10.4. **Diseño e implementación de sistema de visión artificial para alerta y detección de somnolencia mediante aprendizaje profundo aplicable en conductores de vehículos**

En este trabajo realizado en la facultad de Ingeniería de la Universidad Nacional de Trujillo, se desarrolló un sistema de extracción de características faciales tales como pestañeo, cabeceo y bostezos. Para la extracción de regiones de interés se utilizaron cascadas Haar, y la clasificación de estas características se realizó utilizando un modelo de red LeNet. Además el trabajo incluye la creación de una base de datos de las regiones de interés de la cara utilizando imágenes propias y también utilizando conjunto de datos externos. Para el desarrollo de este trabajo se utilizó el lenguaje de programación Python, junto con las librerías de OpenCV, Tensorflow y Keras.

10.5. Sistema de detección de somnolencia mediante inteligencia artificial en conductores de vehículos para alertar la ocurrencia de accidentes de tránsito

Este proyecto fue realizado por estudiantes de la Escuela Profesional de Ingeniería de Perú y consistió en llevar a cabo un sistema para la detección de la somnolencia y la distracción del conductor. El sistema se desarrolló utilizando C# con EmguCV para detectar la distracción y orientación de los ojos utilizando técnicas de visión artificial. Además, cuenta con un sistema de alarma compuesto por un zumbador de 12v, que se activa a recibir la orden el microcontrolador al procesar el sistema de visión artificial junto a una red neuronal.[12]

10.6. Sistema para la detección del estado de somnolencia en seres humanos, con reconocimiento de patrones

En este artículo se muestra la implementación de un sistema de detección del estado de somnolencia en seres humanos, a través de la identificación de patrones faciales y la frecuencia de parpadeo de los ojos. Utilizando técnicas de inteligencia artificial, visión por computadora y un sistema embebido con cámara integrada para la adquisición de imágenes. El cual permite detectar en tiempo real el estado de fatiga de un conductor automovilístico y su grado de somnolencia, todo con el objetivo de disminuir la tasa de accidentes viales causados precisamente por la somnolencia en México. Se hizo uso del lenguaje de programación Python, bibliotecas como OpenCV, Dlib y Scipy, las cuales, fueron requeridas debido a los modelos predefinidos que establecen una mayor precisión en la detección de puntos faciales específicos, utilizando como referencia el método de predicción de 68 puntos específicos del rostro. El sistema propuesto tiene la característica de funcionar con luz de día en una primera etapa, y la idea es poder implementarlo en cualquier tipo de vehículo automotriz a un costo accesible a la mayoría de los propietarios de vehículos automotrices.

10.7. Sistema de Detección de Somnolencia

En este trabajo de fin de grado, realizado por un alumno de la Universidad de La Laguna en el año 2022 se estudió el uso de los modelos de Redes Neuronales para la clasificación de imágenes. Enfocado en la resolución del problema de la somnolencia en los conductores. En el cual se utilizó el lenguaje de programación Python, junto con diversas librerías que facilitan la integración del modelo, y otras que ayudan en la captura de las imágenes.

10.8. Desarrollar un prototipo de reconocimiento facial basado en Machine Learning para detectar estado de Somnolencia en conductores de una cooperativa de transporte.

En este trabajo, realizado por alumnos de la Universidad de GUAYAQUIL en el año 2020-2021 se plantea el desarrollo un prototipo para la detección de la somnolencia del conductor de una cooperativa de transporte el cual dicho conductor se había sobrepasado el límite de horas de trabajo, usando la técnica de reconocimiento facial, basándose específicamente en el estado de los ojos. Para ello se realizó una investigación bibliográfica relacionada con patrones biométricos, inteligencia artificial y programación mediante Machine Learning, así como, las principales variables que

permiten identificar un estado de somnolencia. Dentro de esta investigación también se determinan cuáles son los algoritmos a utilizar (CV2, Imutils, etc.) siendo la herramienta Python y su librería principal Visión por Computadora las seleccionadas para el estudio. [16]

10.9. Sistema basado en la detección y notificación de somnolencia en conductores de autos

En este trabajo, realizado por alumnos de la Universidad de Córdoba en el año 2015 se plantea el desarrollo que permite alertar a los conductores en estado de somnolencia leve, utilizando la tecnología de reconocimiento de objetos de Kinect y la librería OpenCV con el lenguaje C# para el reconocimiento de imágenes. Además, se diseña una aplicación móvil del sistema operativo Android para la notificación de somnolencia utilizando una conexión socket tipo TCP. Para el procesamiento de imágenes, se utilizó el algoritmo Viola-Jones, el cual se basa en una nueva forma de representación de imágenes llamada ?Integral Image?, permitiendo que las características del detector se conmuten rápidamente. [13]

10.10. Detección de somnolencia para conducción sin accidentes

En este trabajo de fin de grado, realizado por un alumno de la Universidad de Alicante en el año 2022 se propuso crear una aplicación real para detectar la somnolencia al volante haciendo énfasis a un bajo coste económico. Donde se realizó una comparativa entre una solución mediante Machine Learning (ML) y Principal Component Analysis (PCA).

10.11. Implementación de un sistema web con geolocalización para medir la transaccionalidad del portal ABC de la democracia para el consejo nacional electoral (CNE)

En este proyecto de titulación de la Universidad de Guayaquil del año 2016, se implementa un sistema de geolocalización para poder tener una mejor visión a través de mapas virtuales de las personas y de las zonas geográficas de los productos o servicios que se quieran introducir, apoyado en un cien por ciento de las herramientas que Google Maps ofrece, se utilizó una aplicación popular y gratuita para poder analizar los datos obtenidos con la localización, Google Analytics. En este proyecto, sólo se pretendía visualizar mapas en la web, por lo que sólo fue necesario usar los servicios que ofrece Google Maps por ser una aplicación gratuita con la ayuda del api Javascript de Google Maps, se tuvieron en cuenta las limitaciones con respecto al código libre, pero éstas no afectaron el desempeño del sistema; los datos obtenidos se recopilaban en una base de datos en MySQL.

10.12. Diseño e implementación de un sistema de geolocalización en interiores para plataforma Android vía la red enterprise WLAN de la PUCP

En este proyecto de titulación de la Pontifica Universidad Católica Del Perú del 2016 se desarrolló una aplicación móvil capaz de geolocalizar a un usuario dentro de las instalaciones de la universidad usando la técnica de Huellas de Señal (fingerprinting), que minimiza el error debido a reflexiones y

obstáculos, basada en el estimador de máxima verosimilitud (ML por sus siglas en inglés Maximum Likelihood) junto a las mediciones de señal de los Access Points cercanos usando la red Wi-Fi. La tecnología de radiofrecuencia que se usó fue la de redes inalámbricas de área local, Wi-Fi. Que ofrece conectividad por radiofrecuencia, con alcance local a un dispositivo que envíe datos Ethernet desde la ubicación del mismo hasta una conexión a la red fija, que en este caso la universidad contaba con 32 access points que recibirían la señal de datos a través de cobre o fibra. La técnica del fingerprinting, que está dirigida a geolocalización en interiores, consiste en un mapeo de datos que se encuentran en un escenario para luego asociarlos a una localización y almacenarlos en una base de datos, para estimar la localización más probable se utilizó el algoritmo de ML basado en el teorema de Bayes de probabilidad. En las conclusiones señalan que este sistema obtuvo una precisión del 100 % en la estimación del ambiente con un error menor de 2.4m en las pruebas realizadas.

10.13. Geolocalización con LoRa mediante multilateración

En esta tesis de la Universidad de la República de Uruguay del 2018 se desarrolló un prototipo para la geolocalización animal en estudios de comportamiento. Para la geolocalización se usó una tecnología nueva emergente llamada LoRa, tecnología de radiofrecuencia de bajo consumo energético y largo alcance; usando un método de multilateración para la geolocalización, técnica de navegación basada en la medición de la diferencia de distancia a dos estaciones en posiciones conocidas, el proyecto se conformó por tres puntos fijos o access points y nodos, que son los dispositivos llevados por los animales de radio frecuencia. Los nodos envían señales de radio frecuencia cada cierto tiempo, la señal emitida se recibe por los puntos fijos que, a su vez, esa señal genera metadatos que se almacenan en una base de datos para posteriormente realizar los cálculos en Geolocator , sistema de geolocalización desarrollado en Python, y así, obtener la posición geográfica del nodo. Se concluyó que el uso de LoRa para la geolocalización tiene gran potencial y que el alcance de las señales fue eficiente, de igual manera se trata de una tecnología de bajo consumo y que mantuvo los dispositivos con energía por meses.

10.14. Propuesta de un sistema de geolocalización y monitoreo vía GPS/GSM/GPRS aplicado a un pulsómetro para personas con enfermedades cardiovasculares

Tesis del Instituto Politécnico Nacional de la Escuela Superior de Ingeniería Mecánica y Eléctrica Unidad Zacatenco del año 2018, donde se empleó un sensor de pulso cardiaco el cual proporciona información en tiempo real de los latidos del corazón mientras que un microcontrolador procesa los datos y, en caso de que se obtengan los datos de que se está presentando una taquicardia, el microcontrolador solicita la ubicación al módulo GPS y envía un mensaje de texto a través de GSM/GPRS a la persona designada.

10.15. Influencia de un sistema de geolocalización en el control y monitoreo de vehículos con dispositivos GPS en una empresa logística

Tesis del año 2015 de la Universidad César Vallejo de Perú donde se investiga de manera profunda las características y detalles de la tecnología GPS para determinar la influencia de un sistema de geolocalización en el control y monitoreo de vehículos con esta tecnología en una empresa logística.

10.16. Diseño de la red de acceso LTE en el distrito de Jesús María

Tesis del año 2017 de la Pontifica Universidad Católica del Perú que tiene por objetivo analizar y diseñar una red usando la tecnología LTE, al desarrollar el proyecto se conocen las ventajas y desventajas de usar esta red así como la posibilidad de implementarla, también se realizan los cálculos de las coberturas y se dan a conocer las velocidades de enlace y el uso de diversos anchos de banda para su despliegue.

11. Análisis

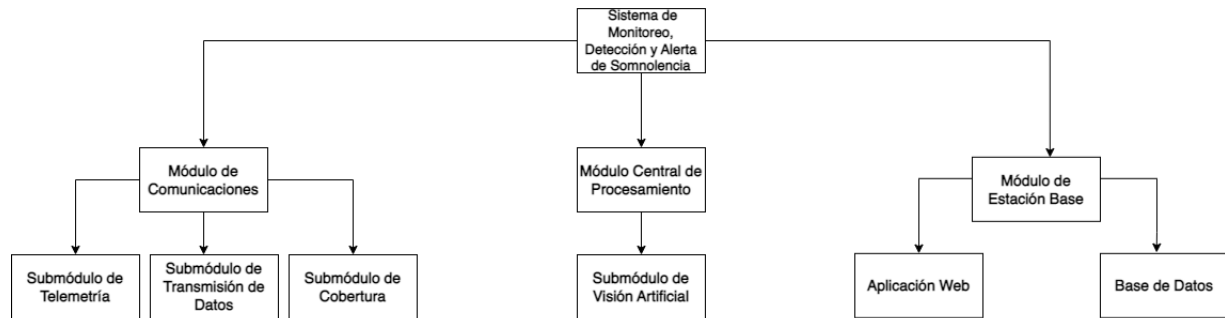


Figura 8: Boosting

11.1. Análisis del Módulo Central de Procesamiento

ID	Nombre corto del Requerimiento
<i>RF01</i>	<i>Verificar estado funcional de periféricos</i>
Descripción: El sistema verificará el estado funcional de los distintos periféricos utilizados en el Módulo Central de Procesamiento	
Elementos: <ul style="list-style-type: none">• Raspberry Pi• Cámara Digital• Zumbador Pasivo• Antena Sixfab	
Solución del Requerimiento: Utilizando métodos de comprobación ofrecidos por cada uno de las librerías de los periféricos, se comprobará que funcionen de manera correcta, en caso contrario se enviará una alerta al módulo de Estación Base.	

Tabla 1: RF01- Verificar estado funcional de periféricos

ID	Nombre corto del Requerimiento
<i>RF02</i>	<i>Activar Alarma</i>
Descripción: El sistema activará el zumbador en caso de que el Submódulo de	
Elementos: <ul style="list-style-type: none">• Raspberry Pi• Zumbador Pasivo	
Solución del Requerimiento: Utilizando métodos de comprobación ofrecidos por cada uno de las librerías de los periféricos, se comprobará que funcionen de manera correcta, en caso contrario se enviará una alerta al módulo de Estación Base.	

Tabla 2: RF02- Activar Alarma

11.1.1. Análisis del Submódulo Visión Artificial

■ Requerimientos Funcionales

ID	Nombre corto del Requerimiento
<i>RF01</i>	<i>Capturar video en tiempo real</i>
Descripción: El sistema capturará video en todo momento	
Elementos: <ul style="list-style-type: none">● Conductor● Incidencias● Administrador● MongoDB	
Solución del Requerimiento: Con el uso de una cámara digital, se capturará video en tiempo real.	

Tabla 3: RF01- Capturar Video en Tiempo Real

ID	Nombre corto del Requerimiento
<i>RF02</i>	<i>Delimitar una región de Interés</i>
Descripción: El sistema capturará video en todo momento	
Elementos: <ul style="list-style-type: none">● OpenCV● Cascadas Haar	
Solución del Requerimiento: Utilizando el software de OpenCV y .	

Tabla 4: RF02- Delimitar región de Interés

11.2. Análisis del Módulo de Comunicaciones

11.3. Análisis del Módulo de Estación Base

El módulo de la estación base tiene como objetivo que el usuario administrador, visualice y confirme el estado de los reportes de incidencia que se hayan presentado por parte del conductor, por tal motivo, se realizará una aplicación web, la cual se conectará a una base de datos NoSQL. En ella se guardarán los reportes de incidencias y se podrán visualizar por medio de la aplicación web. Cabe aclarar que el video de la incidencia se almacenará en la nube, ya que al ser contenido multimedia no se puede guardar en la base de datos, únicamente se guardará el URL para acceder al video. Las credenciales del usuario serán almacenadas en Amazon Cognito, con la cuales podrán iniciar sesión dentro de la Aplicación Web, la cual estará alojada en un servidor web.

El sistema contará con los siguientes requerimientos:

Requerimientos Funcionales del Módulo de Estación Base

ID	Nombre corto del Requerimiento
<i>RF01</i>	<i>Guardar Incidencia</i>
Descripción: La base de datos almacenará la información de cada reporte de incidencia que se envíe desde el Módulo Central de Procesamiento.	
Elementos: <ul style="list-style-type: none">• Id del Conductor• Nombre de Conductor• Apellidos de Conductor• Número de Incidencias• Fecha• Hora• Estado de la Incidencia• Ubicación• URL del video	
Solución del Requerimiento: El Módulo de Procesamiento Central puede insertar datos en la base datos.	

Tabla 5: RF01- Guardar Incidencia

ID	Nombre corto del Requerimiento
<i>RF02</i>	<i>Guardar Video</i>
Descripción: La base de datos almacenará la información de cada reporte de incidencia que se envíe desde el Módulo Central de Procesamiento.	
Elementos: <ul style="list-style-type: none">• Video de incidencia	
Solución del Requerimiento: El Módulo de Procesamiento Central puede insertar el video en el almacenamiento de objetos, mientras que la base de datos obtiene la URL del video guardado.	

Tabla 6: RF02- Guardar Video

ID	Nombre corto del Requerimiento
<i>RF03</i>	<i>Conectar Aplicación Web</i>
Descripción: La base de datos almacenará la información de cada reporte de incidencia que se envíe desde el Módulo Central de Procesamiento.	
Elementos: <ul style="list-style-type: none">• Node.js• React• Express• MongoDB	
Solución del Requerimiento: El usuario administrador puede realizar la inserción, modificación, eliminación y consulta de datos desde la aplicación web.	

Tabla 7: RF03- Conectar Aplicación Web

ID	Nombre corto del Requerimiento
<i>RF04</i>	<i>Desplegar Aplicación Web</i>
Descripción: La aplicación web se alojará en una red de entrega de contenido (CDN), disponible con una URL.	
Elementos: <ul style="list-style-type: none">• AWS Amplify• GitHub• Amazon S3	
Solución del Requerimiento: El usuario administrador puede acceder a la interfaz de la aplicación web, haciendo uso de la URL en el navegador web.	

Tabla 8: RF04- Desplegar Aplicación Web

ID	Nombre corto del Requerimiento
<i>RF05</i>	<i>Guardar Credenciales de Usuario Administrador</i>
Descripción: Se guardarán únicamente las credenciales de los usuarios administradores que podrán acceder a la aplicación web.	
Elementos: <ul style="list-style-type: none">• Amazon Cognito	
Solución del Requerimiento: El usuario administrador puede iniciar sesión en la aplicación web.	

Tabla 9: RF05- Guardar Credenciales de Usuario Administrador

Requerimientos No Funcionales del Módulo de Estación Base

ID	Nombre del requerimiento	Descripción
<i>RNF01</i>	Disponibilidad	La disponibilidad del sistema será continua, el usuario podrá acceder a la información las 24 horas del día.
<i>RNF02</i>	Interoperabilidad	El sistema será capaz de intercambiar información con el Módulo Central de Procesamiento a través del Módulo de Telemetría.
<i>RNF03</i>	Seguridad	Las credenciales de acceso al sistema podrán ser cambiadas únicamente por el administrador.
<i>RNF04</i>	Usabilidad	El sistema estará enfocado a la visualización de reportes, por lo que este tendrá una interfaz intuitiva y amigable para el usuario. El sistema proporcionará mensajes de advertencia orientados al usuario, en caso de ocurrir un error en el Módulo Central de Procesamiento.

Tabla 10: Requerimientos No Funcionales del Módulo de Estación Base

Análisis de Casos de Uso

Con base en los requerimientos se realizó el siguiente diagrama de casos de uso, el cual muestra las actividades y la interacción con el Módulo de Estación base y el Módulo Central de Procesamiento.

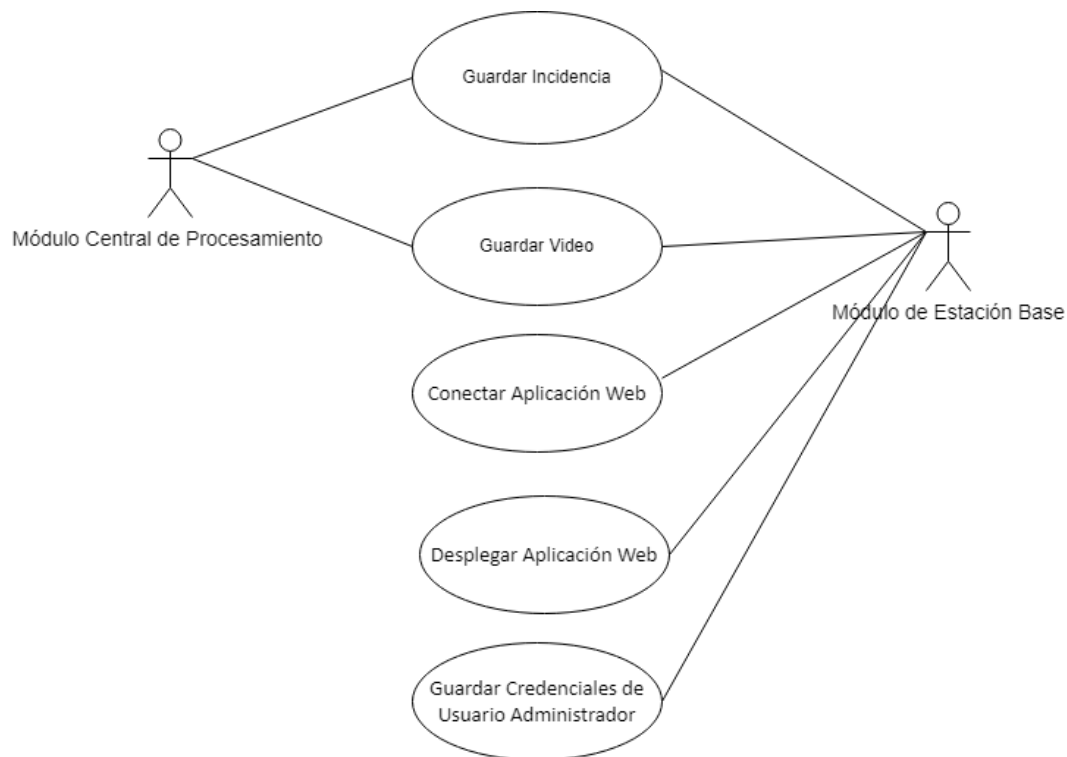


Figura 9: Diagrama de Casos de Usos del Módulo de Estación Base

Requerimientos No Funcionales del Módulo de Estación Base

Nombre del Caso de Uso	<i>Guardar Incidencia</i>
Actores	<i>Módulo Central de Procesamiento, Módulo de Estación Base</i>
Descripción: Se guardarán únicamente las credenciales de los usuarios administradores que podrán acceder a la aplicación web.	
Precondiciones: <ul style="list-style-type: none">● Amazon Cognito	
PostCondiciones: El usuario administrador puede iniciar sesión en la aplicación web.	

Tabla 11: Caso de Uso 01

Nombre del Caso de Uso	<i>Guardar Video</i>
Actores	<i>Módulo Central de Procesamiento, Módulo de Estación Base</i>
Descripción: Se guardarán únicamente las credenciales de los usuarios administradores que podrán acceder a la aplicación web.	
Precondiciones: <ul style="list-style-type: none">● Amazon Cognito	
PostCondiciones: El usuario administrador puede iniciar sesión en la aplicación web.	

Tabla 12: Caso de Uso 02

Nombre del Caso de Uso	<i>Conectar Aplicación Web</i>
Actores	<i>Módulo de Estación Base</i>
Descripción: Se guardarán únicamente las credenciales de los usuarios administradores que podrán acceder a la aplicación web.	
Precondiciones: <ul style="list-style-type: none">● Amazon Cognito	
PostCondiciones: El usuario administrador puede iniciar sesión en la aplicación web.	

Tabla 13: Caso de Uso 03

Nombre del Caso de Uso	<i>Desplegar Aplicación Web</i>
Actores	<i>Módulo Central de Procesamiento, Módulo de Estación Base</i>
Descripción: Se guardarán únicamente las credenciales de los usuarios administradores que podrán acceder a la aplicación web.	
Precondiciones: <ul style="list-style-type: none">● Amazon Cognito	
PostCondiciones: El usuario administrador puede iniciar sesión en la aplicación web.	

Tabla 14: Caso de Uso 04

Nombre del Caso de Uso	<i>Guardar Credenciales de Usuario Administrador</i>
Actores	<i>Módulo Central de Procesamiento</i>
Descripción: Se guardarán únicamente las credenciales de los usuarios administradores que podrán acceder a la aplicación web.	
Precondiciones: <ul style="list-style-type: none">• Amazon Cognito	
PostCondiciones: El usuario administrador puede iniciar sesión en la aplicación web.	

Tabla 15: Caso de Uso 05

11.3.1. Análisis de la Aplicación Web

■ Análisis de Requerimientos Funcionales

ID	Nombre corto del Requerimiento
<i>RF01</i>	<i>Iniciar Sesión</i>
Descripción: El sistema permitirá iniciar sesión en la aplicación web	
Elementos: <ul style="list-style-type: none">• Amazon Cognito• Administrador• Credenciales• MongoDB	
Solución del Requerimiento: <p>El sistema, con la ayuda de los servicios de Amazon Cognito, comprobará que las credenciales ingresadas por parte del usuario administrador se encuentren en la base de datos. En caso contrario, la aplicación web indicará que ese usuario no se encuentra registrado en la base de datos.</p>	

Tabla 16: RF01- Iniciar Sesión

ID	Nombre corto del Requerimiento
<i>RF02</i>	<i>Mostrar el Historial de Reportes de Incidencia</i>
Descripción: El sistema desplegará en forma de lista todas las incidencias que se tengan registradas en la base de datos	
Elementos: <ul style="list-style-type: none">• Conductor• Incidencias• Administrador• MongoDB	
Solución del Requerimiento: <p>El sistema recuperará de la base de datos todas las incidencias que se tengan registradas.</p>	

Tabla 17: RF02- Mostrar el Historial de Reportes de Incidencia

ID	Nombre corto del Requerimiento
<i>RF03</i>	<i>Visualizar Reporte de Incidencia</i>
Descripción: El sistema desplegará los detalles específicos de cada incidencia registrada.	
Elementos: <ul style="list-style-type: none">• Incidencia• Administrador• Conductor	
Solución del Requerimiento: <p>El usuario administrador podrá visualizar los reportes individuales de incidencias de cada conductor al hacer click en cualquiera de las incidencias mostrada en la lista principal. Los reportes contendrán información sobre la fecha, hora, ubicación y un video corto del momento en que fueron detectados síntomas de somnolencia.</p>	

Tabla 18: RF02- Mostrar el Historial de Reportes de Incidencia

ID	Nombre corto del Requerimiento
<i>RF03</i>	<i>Visualizar Reporte de Incidencia</i>
Descripción: El sistema desplegará los detalles específicos de cada incidencia registrada.	
Elementos: <ul style="list-style-type: none">• Incidencia• Administrador• Conductor	
Solución del Requerimiento: <p>El usuario administrador podrá visualizar los reportes individuales de incidencias de cada conductor al hacer click en cualquiera de las incidencias mostrada en la lista principal. Los reportes contendrán información sobre la fecha, hora, ubicación y un video corto del momento en que fueron detectados síntomas de somnolencia.</p>	

Tabla 19: RF03- Visualizar Reporte de Incidencia

ID	Nombre corto del Requerimiento
<i>RF04</i>	<i>Confirmar Incidencia</i>
Descripción: El sistema permitirá al administrador confirmar la incidencia, esto para descartar que se trate de un falso positivo.	
Elementos: <ul style="list-style-type: none">• Administrador• Incidencia• Conductor• MongoDB	
Solución del Requerimiento: <p>El usuario administrador podrá confirmar la incidencia después de haber revisado el videoclip del momento de somnolencia con la intención de descartar falsos positivos. Esto será posible ingresando a una incidencia específica mostrando sus detalles.</p>	

Tabla 20: RF04- Confirmar Incidencia

ID	Nombre corto del Requerimiento
<i>RF05</i>	<i>Recuperar Contraseña</i>
Descripción: El sistema contará con una opción para recuperar la contraseña del administrador en caso de que sea olvidada la contraseña.	
Elementos: <ul style="list-style-type: none">• Amazon Cognito• Email• Administrador• MongoDB	
Solución del Requerimiento: <p>El sistema requerirá que el usuario administrador ingrese el correo con el que fue registrado. Posteriormente se le enviará un código de recuperación de contraseña a ese correo. El administrador ingresará se código en el apartado de recuperar contraseña y así podrá ingresar una nueva contraseña.</p>	

Tabla 21: RF05 - Recuperar Contraseña

ID	Nombre corto del Requerimiento
<i>RF06</i>	<i>Mostrar Perfil Conductor</i>
Descripción: El sistema permitirá al usuario administrador visualizar los datos de cada conductor registrado en la base de datos.	
Elementos: <ul style="list-style-type: none">• Perfil• Conductor	
Solución del Requerimiento: <p>El administrador podrá consultar cada uno de los perfiles de los conductores registrados en la base de datos dando click en el nombre del mismo. En dicho perfil se mostrarán datos como nombre, apellido, así como el número de incidencias de dicho conductor, con sus respectivos detalles.</p>	

Tabla 22: RF06- Mostrar perfil del Conductor

ID	Nombre corto del Requerimiento
<i>RF07</i>	<i>Mostrar Ubicación Geográfica</i>
Descripción: La aplicación web, con ayuda de los datos proporcionados por el módulo de telemetría, mostrará la ubicación en tiempo real de un conductor.	
Elementos: <ul style="list-style-type: none">• Ubicación en Tiempo Real• Módulo de Telemetría	
Solución del Requerimiento: <p>El sistema permitirá al administrador consultar la ubicación de los conductores en tiempo real. Para esto el administrador deberá de ingresar previamente al perfil del conductor del cual desea consultar dicha ubicación.</p>	

Tabla 23: RF07- Mostrar ubicación Geográfica

ID	Nombre corto del Requerimiento
<i>RF08</i>	<i>Descartar Incidencia</i>
Descripción: El sistema permitirá catalogar una incidencia como Falsa si fuera el caso.	
Elementos: <ul style="list-style-type: none">• Incidencia• Administrador	
Solución del Requerimiento: En caso de presentarse una incidencia falsa, después de haber sido revisada por el administrador, esta podrá ser catalogada como falsa incidencia y será eliminada automáticamente de la base de datos.	

Tabla 24: RF08- Descartar Incidencia

ID	Nombre corto del Requerimiento
<i>RF09</i>	<i>Registrar Usuario</i>
Descripción: El sistema le permitirá al administrador registrar nuevos conductores.	
Elementos: <ul style="list-style-type: none">• Incidencia• Administrador	
Solución del Requerimiento: En caso de presentarse una incidencia falsa, después de haber sido revisada por el administrador, esta podrá ser catalogada como falsa incidencia y será eliminada automáticamente de la base de datos.	

Tabla 25: RF09- Registrar Usuario

ID	Nombre corto del Requerimiento
<i>RF10</i>	<i>Modificar Usuario</i>
Descripción: El sistema le permitirá al administrador modificar los datos del conductor.	
Elementos: <ul style="list-style-type: none">• Nombre• Apellido• Conductor• Administrador	
Solución del Requerimiento: El administrador podrá editar los datos de los conductores como nombre o apellido.	

Tabla 26: RF10- Mostrar el Historial de Reportes de Incidencia

ID	Nombre corto del Requerimiento
<i>RF11</i>	<i>Eliminar Usuario</i>
Descripción: El sistema le permitirá al administrador eliminar los datos del conductor. Elementos: <ul style="list-style-type: none">• Conductor• Administrador	
Solución del Requerimiento: El sistema le permitirá al eliminar los datos de conductores de la base de datos.	

Tabla 27: RF11- Mostrar el Historial de Reportes de Incidencia

ID	Nombre corto del Requerimiento
<i>RF12</i>	<i>Generar Mensaje de Alerta</i>
Descripción: El sistema alertará con un mensaje en caso de que algún componente en alguna Unidad Contenedora no funcione correctamente. Elementos: <ul style="list-style-type: none">• Módulo Central de Procesamiento• Administrador• Alerta	
Solución del Requerimiento: El sistema enviará un mensaje de alerta al administrador indicando que Unidad presenta algún desperfecto, esto con la intención de sea reparada posteriormente.	

Tabla 28: RF12- Generar Mensaje de Alerta

■ Análisis de Requerimientos no Funcionales

A continuación se mencionan los requerimientos no funcionales de la aplicación web.

Requerimiento No Funcional	Descripción
Disponibilidad	La aplicación web deberá estar disponible las 24 horas del día
Interoperabilidad	La aplicación deberá ser accesible desde cualquier sistema operativo mientras se cuenta con un navegador web en su versión más actual
Eficiencia	La aplicación tendrá tiempos de respuesta menores a 100ms.
Escalabilidad	La aplicación web deberá ofrecer la oportunidad de agregar nuevas funcionalidades y soportar un mayor número de usuarios a futuro
Estabilidad	La aplicación deberá ofrecer un buen funcionamiento mientras se tenga una conexión a internet mínima de 1Mbps

Tabla 29: Requerimientos No funcionales - Aplicación Web

■ Análisis de Casos de Uso

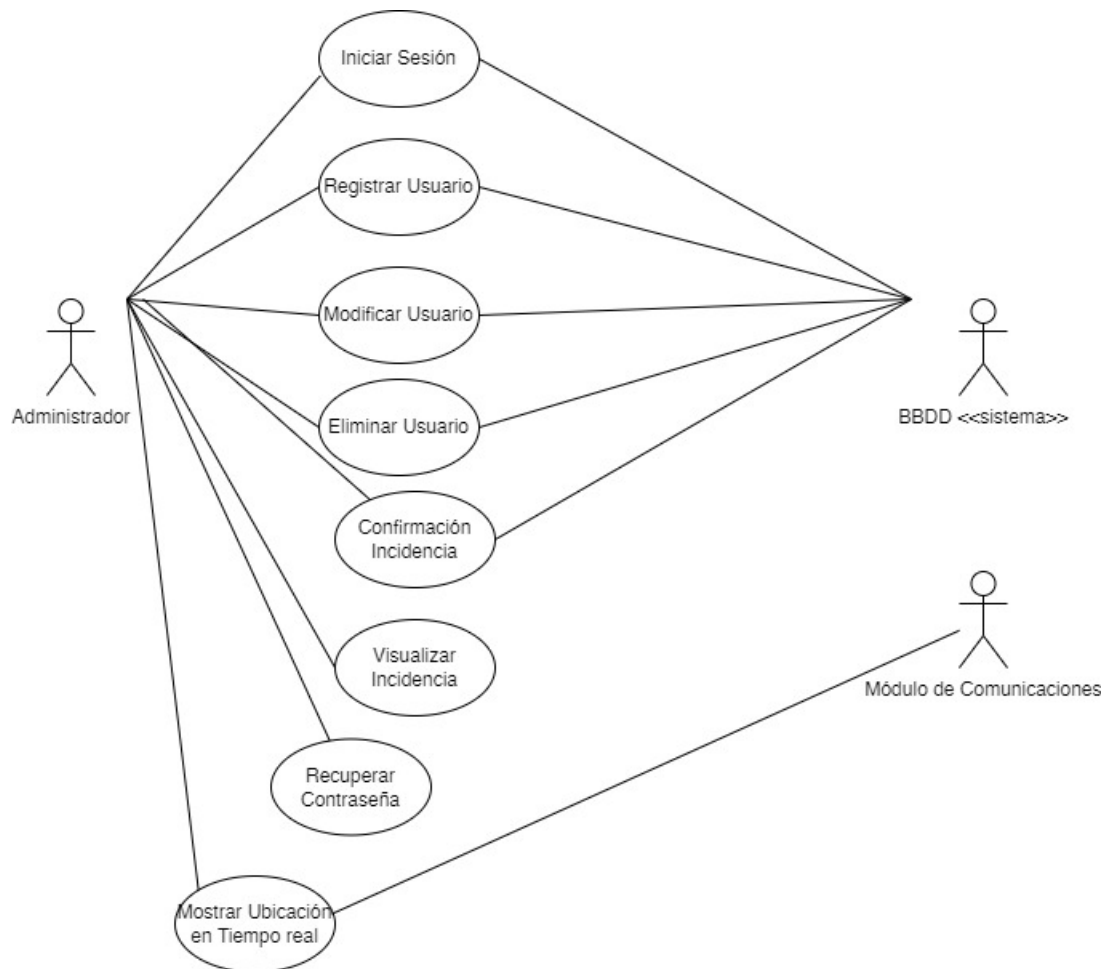


Figura 10: Diagrama de Casos de Uso

Nombre del Caso de Uso	<i>Iniciar Sesión</i>
Actores	<i>Administrador, BBDD</i>
Descripción: El administrador ingresará los datos correspondientes para iniciar sesión en la aplicación web.	
Precondiciones: <ul style="list-style-type: none">• El administrador deberá tener sus credenciales registradas en Amazon Cognito	
Entradas: <ul style="list-style-type: none">• Email• Contraseña	

Tabla 30: Caso de Uso 01

Nombre del Caso de Uso	<i>Registrar Usuario</i>
Actores	<i>Administrador, BBDD</i>
Descripción: Se guardarán únicamente las credenciales de los usuarios administradores que podrán acceder a la aplicación web.	
Precondiciones: <ul style="list-style-type: none">• Amazon Cognito	
Entradas: El usuario administrador puede iniciar sesión en la aplicación web.	

Tabla 31: Caso de Uso 02

Nombre del Caso de Uso	<i>Modificar Usuario</i>
Actores	<i>Administrador, BBDD</i>
Descripción: Se guardarán únicamente las credenciales de los usuarios administradores que podrán acceder a la aplicación web.	
Precondiciones: <ul style="list-style-type: none">• Amazon Cognito	
PostCondiciones: El usuario administrador puede iniciar sesión en la aplicación web.	

Tabla 32: Caso de Uso 03

Nombre del Caso de Uso	<i>Eliminar Usuario</i>
Actores	<i>Administrador, BBDD</i>
Descripción: Se guardarán únicamente las credenciales de los usuarios administradores que podrán acceder a la aplicación web.	
Precondiciones: <ul style="list-style-type: none">• Amazon Cognito	
PostCondiciones: El usuario administrador puede iniciar sesión en la aplicación web.	

Tabla 33: Caso de Uso 04

Nombre del Caso de Uso	<i>Eliminar Usuario</i>
Actores	<i>Administrador, BBDD</i>
Descripción: Se guardarán únicamente las credenciales de los usuarios administradores que podrán acceder a la aplicación web.	
Precondiciones: <ul style="list-style-type: none">• Amazon Cognito	
PostCondiciones: El usuario administrador puede iniciar sesión en la aplicación web.	

Tabla 34: Caso de Uso 05

Nombre del Caso de Uso	<i>Confirmar Incidencia</i>
Actores	<i>Administrador, BBDD</i>
Descripción: Se guardarán únicamente las credenciales de los usuarios administradores que podrán acceder a la aplicación web.	
Precondiciones: <ul style="list-style-type: none">• Amazon Cognito	
PostCondiciones: El usuario administrador puede iniciar sesión en la aplicación web.	

Tabla 35: Caso de Uso 06

Nombre del Caso de Uso	<i>Visualizar Incidencia</i>
Actores	<i>Administrador, BBDD</i>
Descripción: Se guardarán únicamente las credenciales de los usuarios administradores que podrán acceder a la aplicación web.	
Precondiciones: <ul style="list-style-type: none">• Amazon Cognito	
PostCondiciones: El usuario administrador puede iniciar sesión en la aplicación web.	

Tabla 36: Caso de Uso 07

Nombre del Caso de Uso	<i>Recuperar Contraseña</i>
Actores	<i>Administrador, BBDD</i>
Descripción: Se guardarán únicamente las credenciales de los usuarios administradores que podrán acceder a la aplicación web.	
Precondiciones: <ul style="list-style-type: none">• Amazon Cognito	
PostCondiciones: El usuario administrador puede iniciar sesión en la aplicación web.	

Tabla 37: Caso de Uso 08

Nombre del Caso de Uso	<i>Mostrar Ubicación en Tiempo Real</i>
Actores	<i>Administrador Módulo de Comunicaciones</i>
Descripción: Se guardarán únicamente las credenciales de los usuarios administradores que podrán acceder a la aplicación web.	
Precondiciones: <ul style="list-style-type: none">• Amazon Cognito	
PostCondiciones: El usuario administrador puede iniciar sesión en la aplicación web.	

Tabla 38: Caso de Uso 08

12. Diseño

12.1. Diseño del Módulo Central de Procesamiento

■ Elección del Lenguaje de Programación

Debido a que el presente proyecto posee un enfoque orientado a la inteligencia artificial, se debe de tener en claro que esta difiere al paradigma de la programación convencional. En la programación convencional, o estructurada, el usuario le indica a la máquina los datos y las reglas que debe seguir, y la máquina se encarga de obtener los resultados los parámetros mencionados. Sin embargo, en el paradigma de la programación orientada a inteligencia artificial, este enfoque cambia. Ahora se le indican los datos y los resultados, para obtener finalmente las reglas con las que se llegan a dichos resultados.

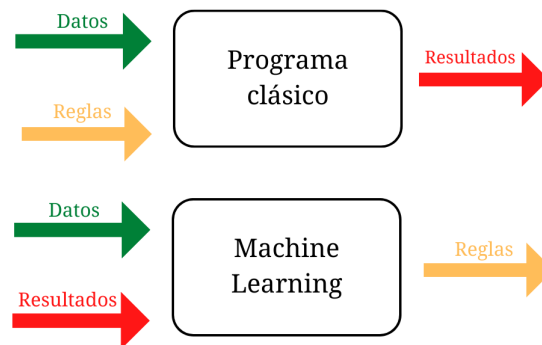


Figura 11: Programa clásico vs Machine Learning

■ Arquitectura

Cómo se mencionó anteriormente en la sección de metodología, a lo largo de este proyecto se hará uso de la metodología *Top-Down*, en dónde uno de sus principios fundamentales radica en la modularidad. Esta no es más que la habilidad de poder dividir programas en partes más pequeñas o sub-programas llamados módulos. Estos módulos deben posser cierto grado de independencia de la aplicación principal, y a su vez poder ser utilizados por otras aplicaciones.

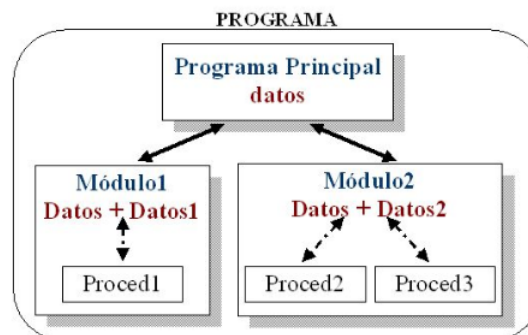


Figura 12: Modularidad

Python es un lenguaje de programación que permite trabajar con más de un paradigma(programación estructurada y orientada a objetos).[\[8\]](#). Normalmente cuando se programa un script sencillo, se aplica el paradigma de programación estructurada. Sin embargo, al tratarse de un problema más complejo, se suele utilizar el paradigma orientado a objetos.

12.1.1. Diseño de la Unidad Contenedora de Procesamiento

Para el diseño de la unidad contenedora del módulo de procesamiento, se tomaron en cuenta los elementos físicos que estarán dentro de la unidad y sus respectivas medidas. Cabe mencionar que los elementos que respectan al modelo del ordenador, el modelo de la cámara, el modelo del zumbador y el modelo de la microSD fueron previamente seleccionados en base a los requerimientos del sistema.

Elementos físicos que contendrá la unidad:

- Raspberry Pi 4 Modelo B

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

Dimensiones 85 x 53mm.

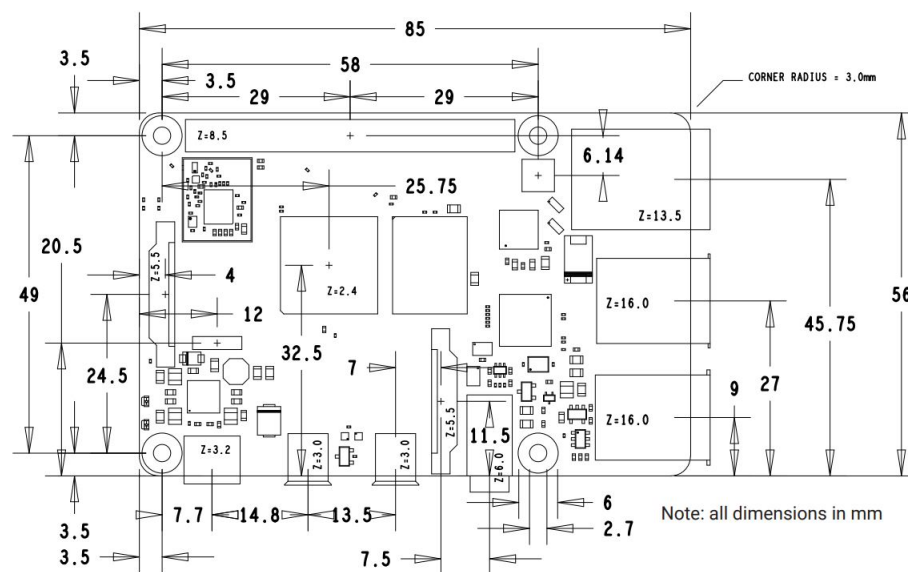


Figura 13: Especificaciones Físicas

- Zumbador Pasivo KY-006

Dimensiones 18 x 15mm.

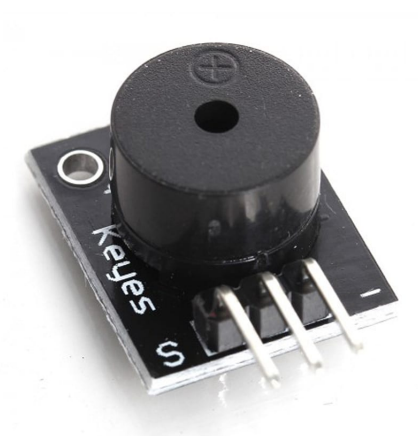


Figura 14: Zumbador Pasivo KY-006

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



Figura 15: Cámara Pivariety IMX230

- Cable macho-hembra



Figura 16: 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 17: Micro SD

- RaspberryPi 3G/4G/ LTE Base HAT



Figura 18: 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, además de contar con todo el software necesario para desarrollar en una Raspberry Pi.

- LE 910Cx Mini PCIe Series Linux

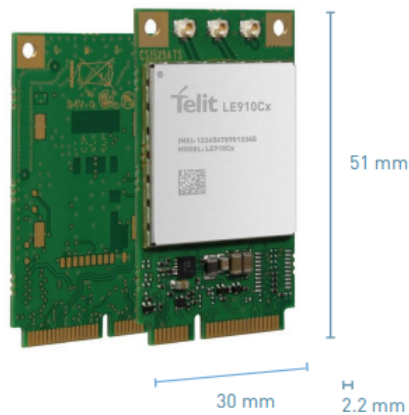


Figura 19: LE 910Cx Mini PCIe Series Linux

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 20: 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.

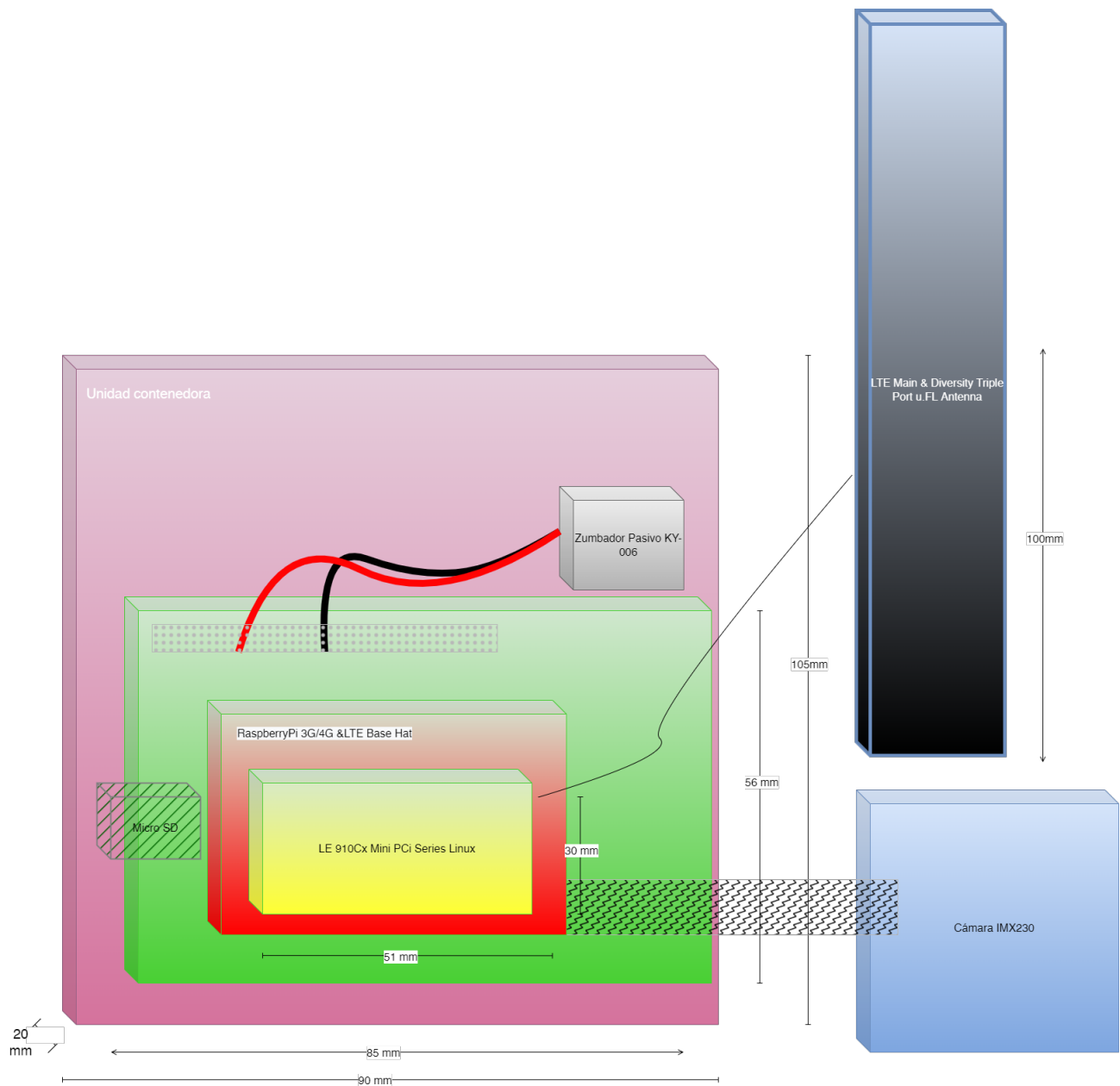


Figura 21: Unidad Contenedora del Módulo Central de Procesamiento

12.1.2. Diseño del Submódulo de Procesamiento de Imágenes

■ Puntos de Referencia

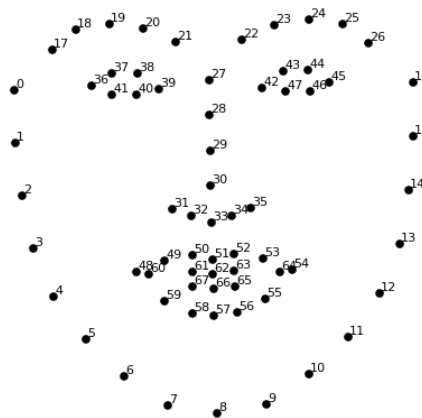


Figura 22: Puntos de referencia

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 posteriormente 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 39: 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 aperatura de la boca, por lo tanto cumple con las características de un bostezo.

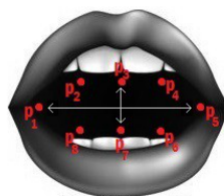


Figura 23: *Mouth Opening Ratio*

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

$$MOR = \frac{\|p_2 - p_8\| + \|p_3 - p_7\| + \|p_4 - p_6\|}{2 \|p_1 - p_5\|} \quad (5)$$

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

$$MOR = \frac{\|P49 - P59\| + \|P51 - P57\| + \|P53 - P55\|}{2 \|P48 - P54\|} \quad (6)$$

12.1.3. Diseño del Submódulo de Deep Learning

12.2. Diseño del Módulo de Comunicaciones

12.2.1. Diseño del Submódulo de Transmisión de Datos

12.2.2. Diseño del Submódulo de Telemetría

12.2.3. Diseño del Submódulo de Cobertura

12.3. Diseño de la Estación Base

Con base en los requerimientos, se plantea realizar el diseño de la arquitectura del Módulo de Estación Base, integrando el Módulo de Procesamiento Central. El cual hará énfasis en la organización y comunicación de los elementos que lo conforman.

El siguiente diagrama muestra la arquitectura del sistema de Estación base, el cual se compone de las interacciones de los elementos que permitirán al usuario acceder e interactuar con la aplicación web. El diagrama también muestra la participación del Módulo de Procesamiento Central ya que el registro de incidencias y el envío del video de incidencia serán realizados por dicho módulo. La información de los reportes de incidencias y el registro de los conductores serán almacenados en Mongo DB, mientras que los videos de incidencia se almacenarán en Amazon S3. Las credenciales de los Usuarios Administradores que tendrán permitido acceder a la aplicación se almacenarán en Amazon Cognito. Para la aplicación web, el *backend* será desarrollado en node.js mientras que el *frontend* será desarrollado en React, posteriormente los archivos se almacenarán en un repositorio dentro de Github. La aplicación web será alojada y desplegada desde AWS Amplify, lo cual le permitirá al cliente acceder a la aplicación desde una URL.

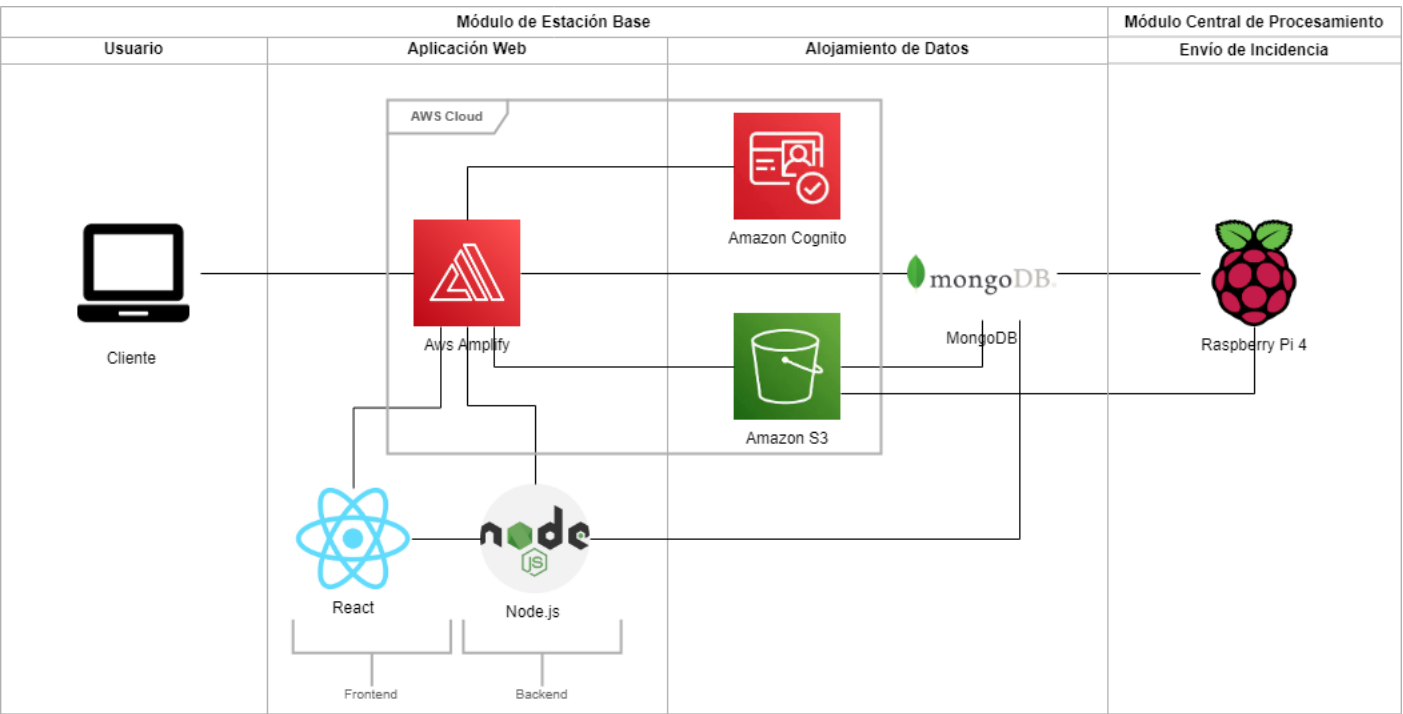


Figura 24: Arquitectura del Módulo de Estación Base

Elección del servidor de alojamiento Después de haber analizado los distintos proveedores de alojamiento así como sus características, llegamos a la conclusión que el que mejor cumple con los requisitos del proyecto es AWS Amplify. Ya que este contiene una suite de herramientas que facilitan la implementación de una aplicación web, como por ejemplo S3, que es un servicio de alojamiento en la nube utilizado para archivos multimedia de gran tamaño. [4]

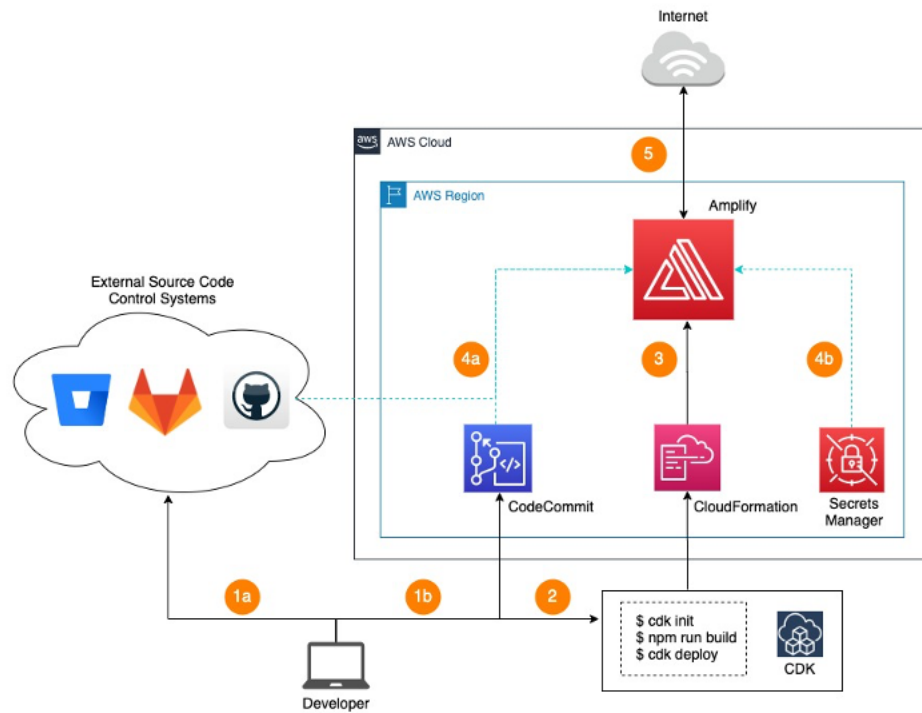


Figura 25: Diagrama de Componentes AWS Amplify

Otro factor fundamental por el cual se decidió utilizar la suite de AWS Amplify, es que esta cuenta con su propia CDN, lo cual significa que los tiempos de respuesta al usar la aplicación web deben de ser mínimos.

12.3.1. Diseño de la Aplicación Web

Para el diseño y desarrollo de la Aplicación web, se decidió utilizar el lenguaje Javascript, ya que este cuenta con las herramientas y/o frameworks que mejor se adaptan a los requisitos previamente definidos. A su vez, se decidió utilizar el NodeJs para el sistema backend.

Node.js, es un entorno en tiempo de ejecución multiplataforma basado en JavaScript. Este es controlado por eventos, permitiendo establecer y gestionar múltiples conexiones al mismo tiempo. Gracias a esta característica, el bloqueo de procesos no existe. NodeJs trabaja fundamentalmente bajo dos características: *asincronía*, que permite la ejecución de varios procesos al mismo tiempo, y *Entrada/salida sin bloqueo*, que significa poder trabajar con múltiples solicitudes sin bloquear un hilo para una sola solicitud. [2]

NodeJs es capaz de manejar distintas peticiones sin que tener que esperar a que una petición sea respondida para continuar con la siguiente petición. De ahí la elección de este entorno de Javascript para el desarrollo de la aplicación web.

A continuación se muestran diversos diagramas de secuencias que describen los procesos de los requerimientos definidos en la sección de Análisis.

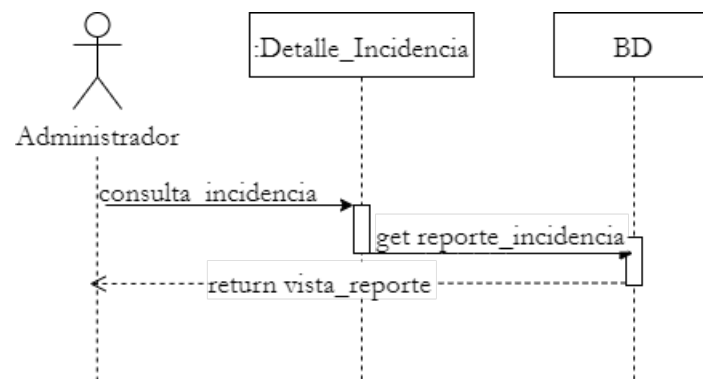


Figura 26: Diagrama de Secuencia Detalle Reporte Incidencia

El Administrador podrá consultar el reporte de incidencia de cada uno de los conductores. Accediendo a la base de de datos de MongoDB

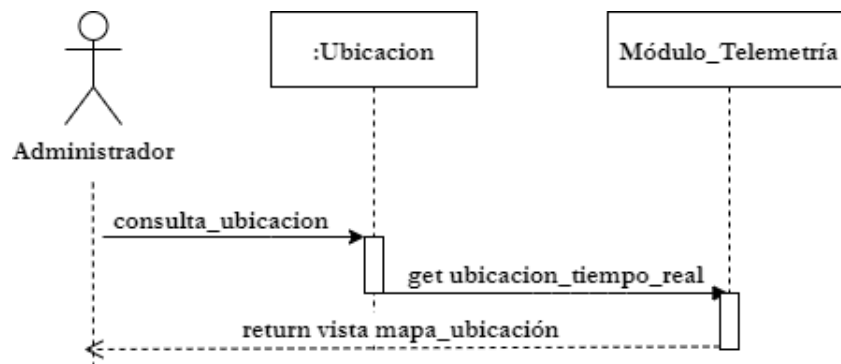


Figura 27: Diagrama de Secuencia Consultar Ubicación

De igual manera, el Administrador podrá consultar la ubicación en tiempo real del conductor.

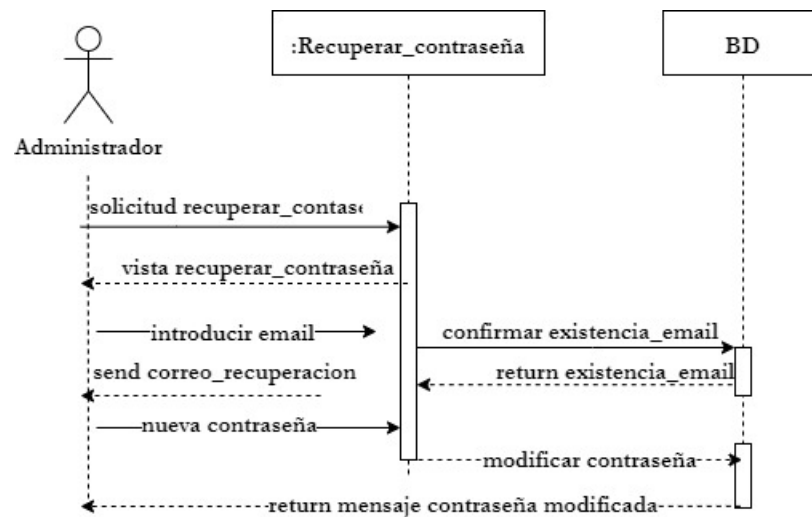


Figura 28: Diagrama de Secuencia Recuperar Contraseña

También podrá recuperar su contraseña en caso de que esta sea olvidada.

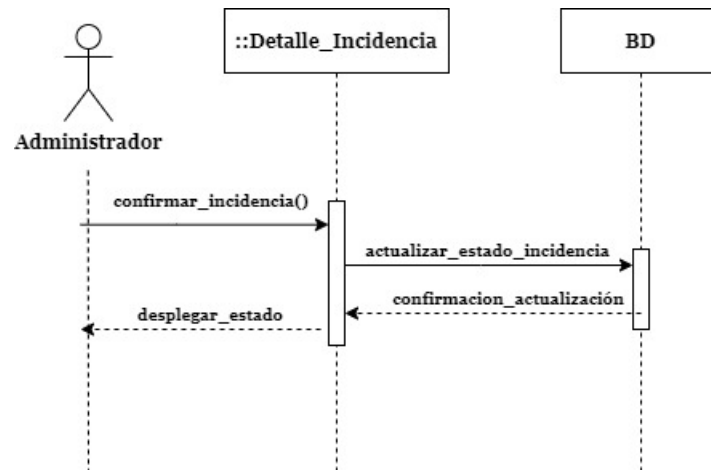


Figura 29: Diagrama de Secuencia Confirmar Incidencia

Para evitar falsos positivos, el Administrador podrá confirmar una incidencia una vez mirando el videoclip del incidente.

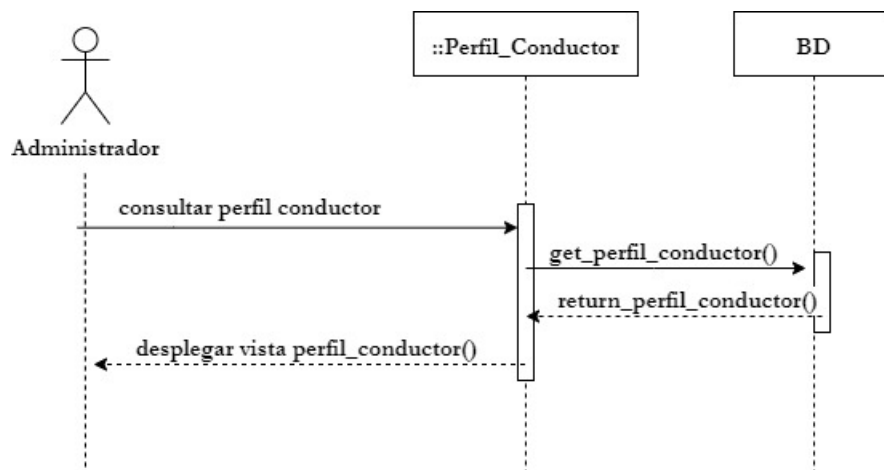


Figura 30: Diagrama de Secuencia Consultar Perfil

El Administrador podrá consultar el perfil de cada conductor.

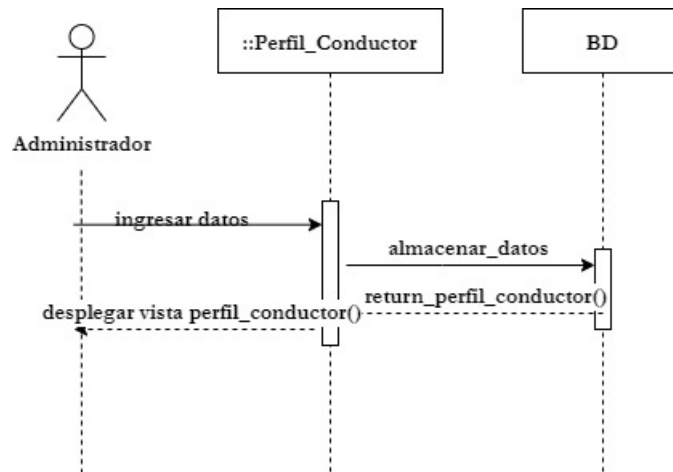


Figura 31: Diagrama de Secuencia Registrar Conductor

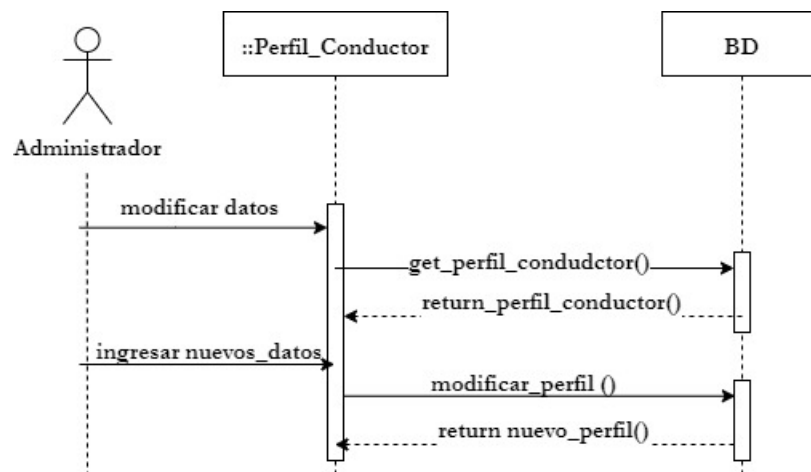


Figura 32: Diagrama de Secuencia Modificar Conductor

En caso de que los datos del conductor sean incorrectos, el administrador podrá modificarlos.

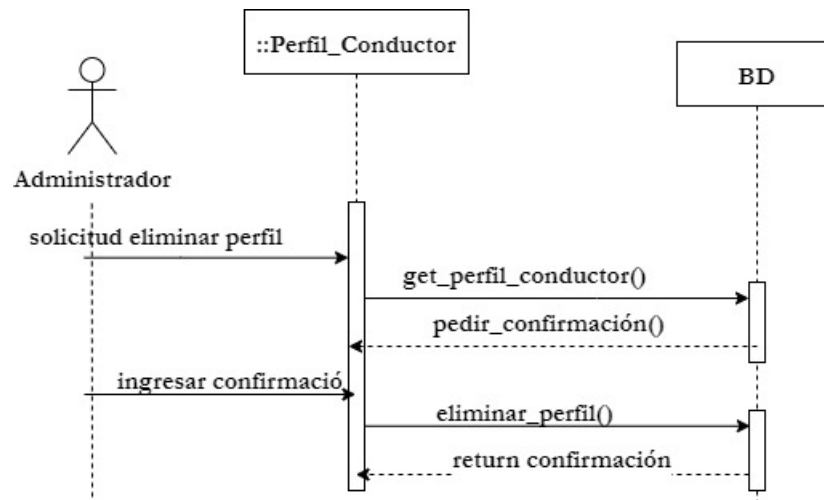


Figura 33: Diagrama de Secuencia Eliminar Conductor

El Administrador también tendrá la opción de dar de baja a un conductor.

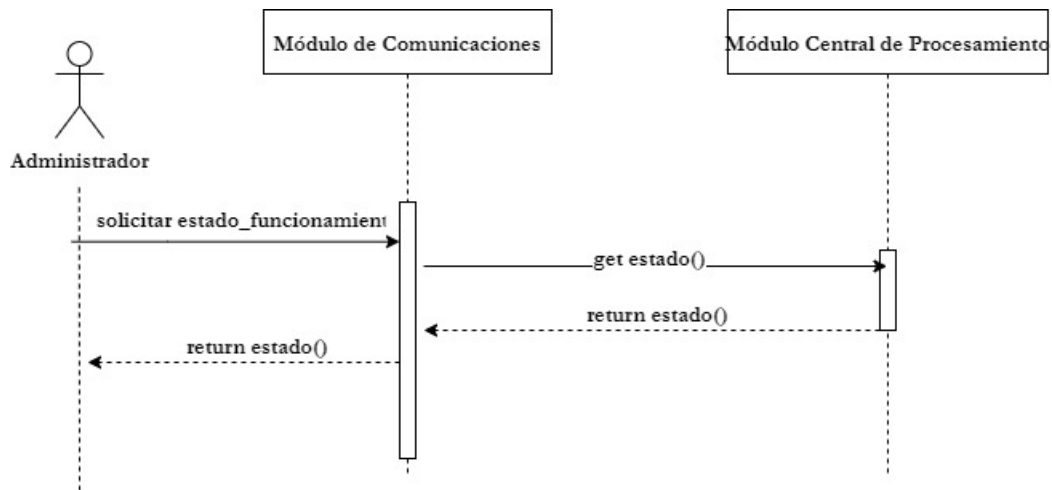


Figura 34: Diagrama de Secuencia Estado de los Periféricos

La aplicación web indicará si alguno de los periféricos no se encuentra en correcto funcionamiento, lanzando una alerta indicando el número de Unidad y a qué conductor pertenece.

A continuación se muestra el diseño propuesto de cada una de las vistas que tendrá la aplicación web.

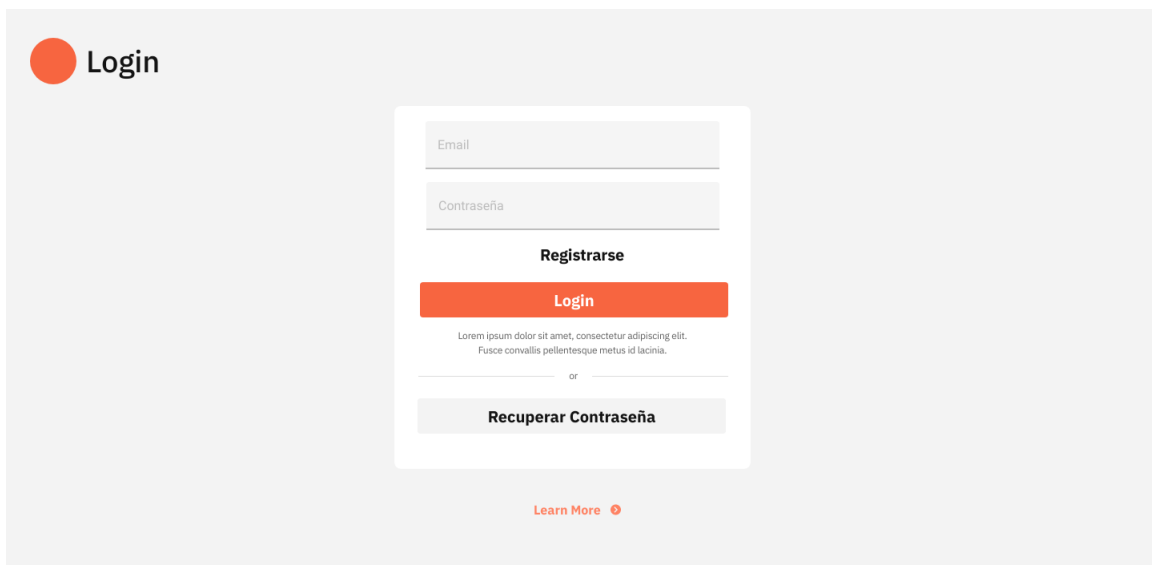


Figura 35: Página Inicio de Sesión

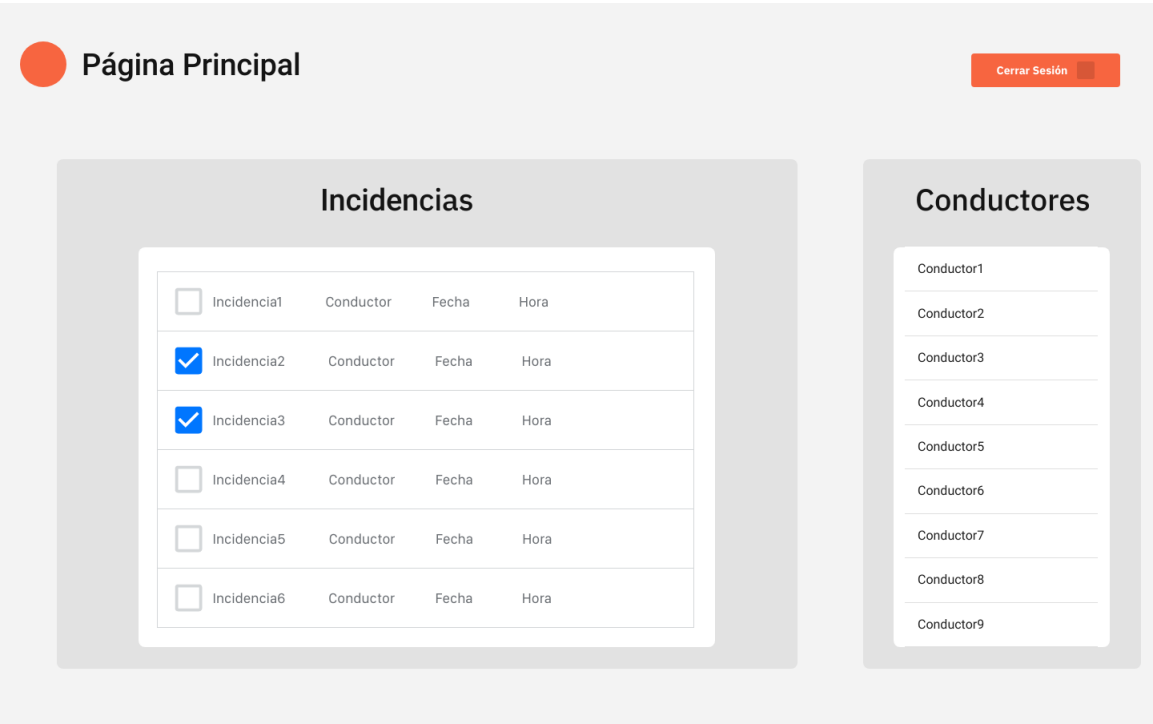


Figura 36: Página Principal

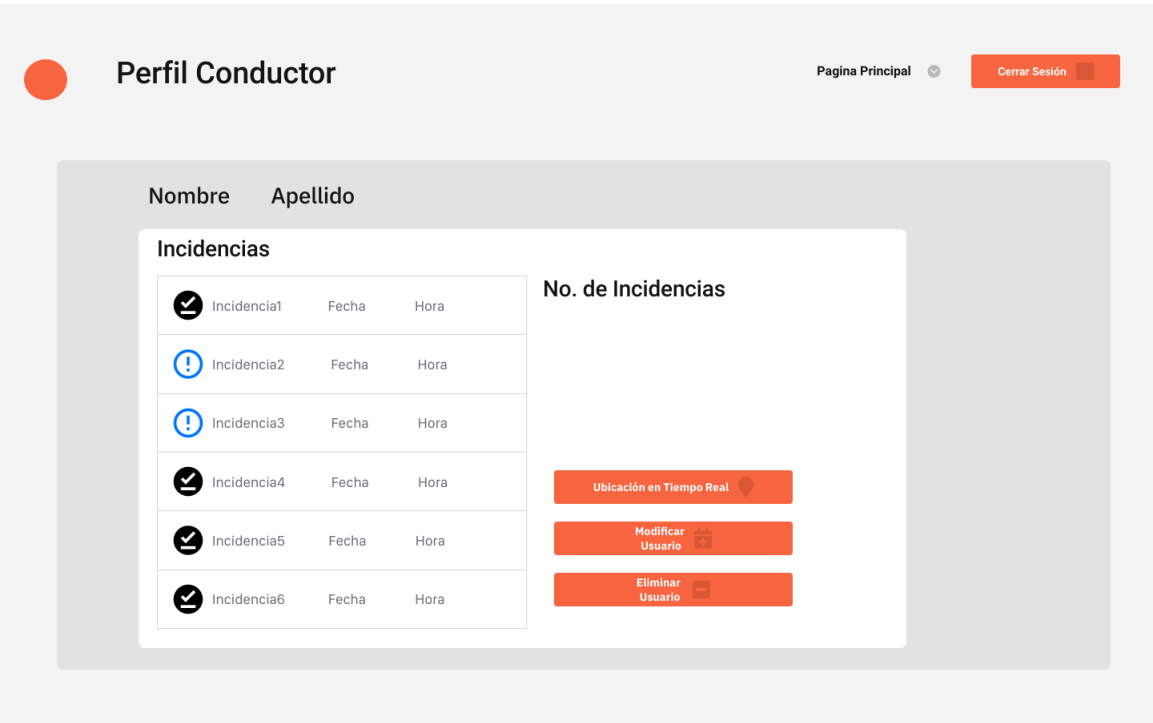


Figura 37: Página Perfil del Conductor

Reporte de Incidencia

Página Principal Cerrar Sesión

ID INCIDENCIA

Nombre Apellido

Fecha Hora

Ubicación (Coordenadas)

UBICACIÓN EN TIEMPO REAL

Confirmar Descartar

Figura 38: Página Detalle de Incidencia

Ubicación en Tiempo Real

Página Principal Cerrar Sesión

Nombre Apellido

Figura 39: Página Ubicación en Tiempo Real

12.3.2. Diseño de la Base de Datos

Elección del manejador de Base de Datos

■ MongoDB

Debido a la escalabilidad horizontal, facilidad de implementación, bajo coste y flexibilidad que ofrece los gestores de bases de datos NoSQL para adaptarse a las necesidades del proyecto, se optó por la elección de un gestor de este tipo. Además, se contemplaron las posibles modificaciones dentro del modelado de la base de datos, las cuales tendrán como motivo, cumplir los requisitos del sistema y presentar la información necesaria en la aplicación web. Estas modificaciones (en caso de ser necesarias) podrán ser fácilmente aplicadas usando un SGBD NoSQL ya que ofrecen una mayor facilidad de realizar cambios dentro del esquema a diferencia de un SGBD SQL.

Dentro de los SGBD NoSQL se eligió MongoDB, ya que la aplicación web será desarrollada utilizando como base el lenguaje JavaScript y el framework Node.js para realizar el *backend* de la aplicación. Por lo cual MongoDB, se integrará de manera funcional en el desarrollo del proyecto. Además, MongoDB es de código abierto, y es una herramienta gratuita hasta cierto punto, por lo cual no se requerirá una licencia para la implementación. Otra característica importante es que es una base de datos multiplataforma, esto nos permitirá registrar información desde el sistema operativo Linux, que se ocupará en la Raspberry Pi y por otro lado se ocupará Windows en la aplicación web para la inserción, modificación, lectura y eliminación de los datos en la base de datos.[6]

MongoDB utiliza "documentos" para registrar los datos, estos documentos son almacenados dentro de las colecciones, las cuales, en una base de datos relacional se asemeja a una tabla. Las bases de datos en MongoDB están conformadas por una o más colecciones de documentos.

MongoDB almacena registros de datos como documentos BSON. BSON es una representación binaria de documentos JSON, aunque contiene más tipos de datos que JSON.

Un registro en MongoDB es un documento, que es una estructura de datos compuesta por pares de campo y valor. Los documentos MongoDB son similares a los objetos JSON. Los valores de los campos pueden incluir otros documentos, matrices y matrices de documentos.

```
{
  name: "sue",
  age: 26,
  status: "A",
  groups: [ "news", "sports" ]
}
```



Figura 40: Estructura de un Documento en MongoDB

Basado en la estructura NOSQL de MongoDB, se propone el siguiente modelo para la base de datos, la cual contará con dos colecciones: Conductores e Incidencias. Cabe mencionar que en dicho modelo no se contempla a los administradores, debido a que se utilizará el sistema

de Amazon Cognito, que ofrece herramientas de autenticación, registro, verificación e inicio de sesión para el usuario.

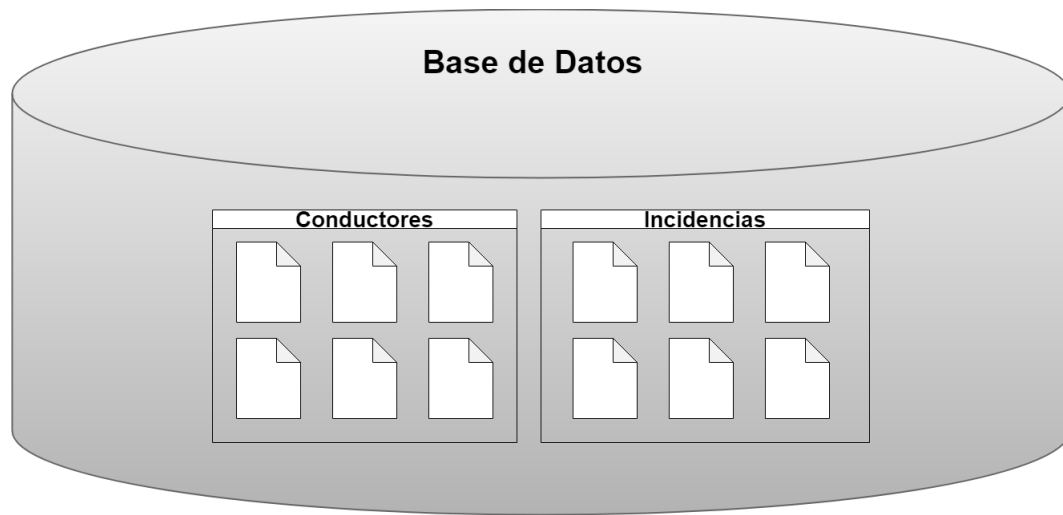


Figura 41: Modelo propuesto para la base de datos en MongoDB

El modelo de documento para la colección Conductores, se usará para registrar el nombre y apellido de cada conductor, así mismo se contabilizará el número de incidencias que presenté cada uno de ellos, el contador del conductor en marcha aumentará cada vez que el Módulo Central de Procesamiento envíe un reporte de incidencia a la base de datos. Si el Usuario Administrador del Módulo de la Estación Base revisa el video de la incidencia y lo cataloga como Descartar entonces se restará la incidencia, si esta es catalogada como Confirmar, el contador permanecerá igual. Es importante mencionar que el registro de cada conductor se realizará desde la Aplicación Web. A continuación, se muestran los campos y un ejemplo de los valores que pueden ser ingresados dentro del documento de registro de conductores.

```
{
  "_id" : 1,
  "NombreConductor" : "Nombre Conductor",
  "ApellidoConductor" : "ApellidoMaterno ApellidoPaterno",
  "NúmeroIncidencias" : 0
}
```

Figura 42: Documento de Registro de Conductores

El modelo de documento para la colección Incidencias, se usará para registrar el Reporte de Incidencia que presente un conductor, el cual será enviado desde el Módulo Central de Procesamiento, por tanto, se realizará una consulta previa a la colección Conductores para obtener el id, nombre y apellido del conductor en marcha. Posteriormente se realizará el reporte de la incidencia, el cual contendrá el Id del conductor, la fecha y hora de la incidencia, el nombre y apellido del conductor, y el estado de incidencia, este último muestra si la incidencia fue catalogada como descartada o confirmada por parte del Usuario Administrador después de revisar el video de la incidencia. Como se mencionó anteriormente, MongoDB permite anidar documentos, por lo que, dentro de este documento se encontrará anidado

un segundo documento llamado Detalles, el cual registrará la Ubicación donde sé detectó la incidencia y la URL donde se encontrará almacenado el video.

```
{  "_id" : 1,
    "_idConductor" : "1",
    "NombreConductor" : "Alan",
    "ApellidoConductor" : "Gamboa del Angel",
    "FechaHora" : "2022-11-16T13:30:00.000Z",
    "Estado_Incidencia" : null,
    "Detalles": {
      "Ubicación" : [ -73.856077, 40.848447 ],
      "Video" : "URL"
    }
}
```

Figura 43: Documento de registro de Incidencias

Cabe señalar que los anteriores modelos muestran el campo Id, sin embargo, MongoDB asigna automáticamente un Id, si el id no es especificado, esto lo realiza cada vez que se crea un nuevo documento dentro de una colección, el cual es un valor hexadecimal de 12 bytes representado en 24 caracteres. Por tanto, se utilizará el id que proporcione Mongo DB en los documentos.

- **Amazon S3 (*Simple Storage Service*)**

Amazon S3, es un servicio de almacenamiento de objetos en la nube. Es utilizado para almacenar datos en la nube de una forma segura, eficiente y escalable.

- **Amazon Cognito**

13. 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.

14. Referencias

Referencias

- [1] M. Paez y Emilio Abarca. *Resumen Boletines*, Instituto Mexicano del Transporte. <https://imt.mx/resumen-boletines.html?IdArticulo=449&IdBoletin=168> (accedido el 18 de marzo de 2022).
- [2] Jonathan Wexler, *Get Programming with Node.js*, Manning, 2019
- [3] Marjin Haverbeke *Eloquent Javascript*, Third Edition, No Starch Press, 2020
- [4] Amazon, *Amazon Amplify*, <https://aws.amazon.com/es/amplify/>
- [5] David A. Forsyth, Jean Ponce, *Computer Vision a Modern Approach*, Second Edition, Pearson, 2012
- [6] Kyle Banker, *MondoDB in Action*, Second Edition, Manning, 2021
- [7] Fernando Berzal, *Redes Neuronales y Deep Learning*, Granada, 2018
- [8] Irv Kalb, *Object-Oriented Python*, Primera Edición ,No Starch-Press, 2021
- [9] V.Moret Bonillo, *Fundamentos de Inteligencia Artificial*, Segunda Edición,Santiago de Compostela: Universidad de La Coruña Servicio de Publicaciones, 2005.
- [10] Ian Sommerville, *Ingeniería de Software*, Novena Edición , Pearson Educación de México, 2011
- [11] Kenneth E. Kendall, Julie E. Kendall, *Análisis y Diseño de Sistemas*, Octava Edición , Pearson Educación de México, 2011
- [12] Aurélien Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and Tensorflow*, Segunda Edición , O'Reilly, 2009
- [13] Kurt Demagd, Anthony Oliver, Nathan Oostendorp y Katherine Scott *Practical Computer Vision with OpenCV*, Tercera Edición, O'Reilly, 2017
- [14] Cuno Plister, *Getting Started with Internet of Things*,Primera Edición, O'Reilly, 2011
- [15] 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
- [16] S. Pachouly, N. Bhondve, A. Dalvi, V. Dhande y N. Bhamare. "Driver drowsiness detection using machine learning with visual behaviour". <https://ijcrt.org/papers/IJCRT2006408.pdf> (accedido el 2 de noviembre de 2022).