

Proyecto de Ciclovías Inteligentes y Transporte Sostenible en la RM Colaboración PNUD - GORE-RM

Producto 5: Pilotaje Dispositivo IoT para monitoreo y centralización de información de Ciclovías Integración Productos 3 y 4

Consultor:

Dr. Nicolás Sebastián Gálvez Ramírez
Académico Ing. Civil Telemática
UTFSM

Equipo:

Dr. Sebastián Diez
Académico
UDD

Ignacio Jorquera Navarro
Est. M.Sc. Ingeniería Informática
UTFSM

Matías Soto Sepúlveda
Est. Ing. Civil Telemática
UTFSM

16 de septiembre de 2024

Índice

1. Introducción	3
1.1. Ciclovías Inteligentes y Transporte sostenible para la RM	3
1.1.1. Productos del Proyecto	3
1.1.2. Producto 5	4
1.2. Objetivos del Producto 5	4
1.2.1. Objetivo General (OG)	4
1.2.2. Objetivos Específicos (OEx)	4
1.3. Estructura del Reporte	5
I Propuesta de Solución	6
2. Requerimientos de Solución para Producto 5	7
2.1. Arquitectura de la Solución	7
2.2. Antecedentes Producto 5	10
2.2.1. Medición de Contaminantes a bajo costo	10
2.2.2. Visión por Computador: Detección de Objetos en Tiempo Real	11
2.2.3. YOLO para Detección de Objetos	11
2.2.4. Alternativas a YOLO	11
2.2.5. Visión por Computador: Identificación, Seguimiento y Contador de Objetos	12
2.2.6. Obtención de Imágenes en Tiempo Real	13
2.2.7. Microcontroladores IoT	13
2.2.8. Conectividad y Protección de Dispositivos	14
II Diseño e Implementación de Solución	16
3. MVP y Escalabilidad	17
3.1. MVP: Producto Viable Mínimo y sus módulos	17
3.1.1. Módulo Calidad del Aire	17
3.1.2. Módulo de Detección y Conteo de Objetos Viales	18
3.1.3. Integración y Montaje	21
3.1.4. Salida de Datos	25
3.2. Presupuesto	27
3.3. Escalabilidad, Mejoras y Costos Asociados	27
3.3.1. Clasificación de Ciclos	27
3.3.2. Clasificación de Peatones	28
3.3.3. Persistencia de Vídeo	29
III Pilotaje y Resultados	31
4. Calibración Sensores de Calidad de Aire	32
4.1. Colocación	32
4.2. Calibración de toma de datos.	34
5. Pilotaje	37
5.1. Lugar y Fechas de Despliegue	37
5.2. Instalación y Configuración de Dispositivo	37
6. Resultados Experimentales	39
6.1. Contraste entre períodos claves	39
6.2. Contraste de contaminantes atmosféricos	41
6.3. Contraste entre objetos viales y contaminantes atmosféricos	42
6.4. Progresión de detección de objetos viales	43
7. Conclusiones y Resultados Esperados	44

Índice de figuras

2.1. Arquitectura del dispositivo IoT de la propuesta.	8
2.2. Arquitectura de Propuesta para el Producto 5.	9
2.3. Integración entre Productos 3, 4 y 5.	9
2.4. Caja Norma IP67: Caja de referencia y ejemplo con Raspberry Pi	15
2.5. Carcasa con equivalente a norma IP67 para cámara Raspberry Pi 2/3	15
3.1. Módulo de calidad de Aire montado, sin BAM incluida	17
3.2. Arquitectura del Modulo de Calidad de Aire.	18
3.3. Módulo de Detección y Conteo de Objetos Viales.	18
3.4. Arquitectura del Modulo de Detección y Conteo de Objetos Viales.	19
3.5. MVP: Primera vista producto integrado.	22
3.6. Arquitectura de MVP del Producto 5.	23
3.7. MVP: Dimensiones del prototipo base.	24
3.8. MVP: Dimensiones del prototipo con rieles de montajes.	24
3.9. Arquitectura de Salida de Datos.	25
3.10 Información de calidad del aire recopilada por ámbos módulos en formato .json.	25
3.11 Front-end de visualización de datos.	26
4.1. Ubicación de Estación SINCA en Parque O'Higgins	32
4.2. Estación SINCA en Parque O'Higgins	33
4.3. Colocación de Dispositivos en SINCA Parque O'higgins	33
4.4. Calibración de Sensor MP2.5	34
4.5. Calibración de Sensor NO	35
4.6. Calibración de Sensor NO2	35
4.7. Calibración de Sensor MP2.5	35
4.8. Calibración de Sensor NO	36
4.9. Calibración de Sensor NO2	36
5.1. Lugar de Despliegue, Colegio San Agustín, Calle Contramaestre Micalvi, Ñuñoa, RM.	37
5.2. Punto de instalación en el lugar de despliegue, Colegio San Agustín, Calle Contra- maestre Micalvi, Ñuñoa, RM.	38
5.3. Ejemplo de Captura en Despliegue	38
6.1. Distribución de Automóviles a lo largo de la semana	40
6.2. Distribución de Personas a lo largo de la semana	40
6.3. Distribución de Bicicletas a lo largo de la semana	40
6.4. Distribución de Motocicletas a lo largo de la semana	40
6.5. Distribución MP2.5 a lo largo de la semana	41
6.6. Distribución NO a lo largo de la semana	41
6.7. Distribución NO2 a lo largo de la semana	41
6.8. Contraste entre cantidad de Automóviles y medición de MP2.5	42
6.9. Contraste entre cantidad de Automóviles y medición de NO	42
6.10 Contraste entre cantidad de Automóviles y medición de NO2	43
6.11 Progresión de Objetos Viales detectados durante el período de pilotaje.	43

1. Introducción

Con el fin implementar acciones sustentables de movilidad en el marco del programa Santiago Conectado y el desarrollo del Centro de Gestión Integrada de Información Regional, el GORE-RM ha firmado un convenio de colaboración con el Programa de las Naciones Unidas para el Desarrollo (PNUD) mediante la Oficina de las Naciones Unidas para la Cooperación Sur-Sur (UNOSSC).

1.1. Ciclovías Inteligentes y Transporte sostenible para la RM

La colaboración ha iniciado con la implementación del proyecto de *Ciclovías Inteligentes y Transporte sostenible para la RM*, bajo el marco del *Proyecto Ciudades (Cities Project)*, que busca generar intercambio de capacidades técnicas y buenas prácticas entre Hangzhou (China) y Santiago (Chile) para promover un transporte sostenible, inteligente y bajo en emisiones contaminantes. Específicamente el proyecto considera tres líneas de acción:

1. Desarrollo de un análisis detallado del estado del sistema de transporte regional metropolitano y elaboración de pautas técnicas sobre una red de transporte inteligente y planificada.
2. Aumento de la valoración social del transporte sostenible y fortalecimiento de las capacidades institucionales del gobierno regional y de los y las profesionales encargados de la gestión del transporte.
3. Promoción del grupo de Cooperación Sur-Sur de Ciudades sobre Transporte Sostenible y Calidad del Aire, y fomento de los intercambios y relaciones entre ciudades para compartir experiencias.

1.1.1. Productos del Proyecto

La ejecución del proyecto de *Ciclovías Inteligentes y Transporte sostenible para la RM* consta con la realización de cuatro productos esperados:

- Producto 1: Diseño e implementación de Entrevistas a actores locales clave para identificar brechas y necesidades del sistema de transporte urbano y de su modelo de gobernanza.
- Producto 2: Diagnosticar la RM en base al modelo de Hangzhou y realizar una propuesta metodológica para implementar un sistema de transporte público inteligente en el marco de lo que será la Autoridad Metropolitana de Transporte.
- Producto 3: Propuesta técnico-metodológica para Estrategia de Big Data para apoyar, al GORE RMS, en la transformación de la actual red de ciclovías en una Red Ciclista Inteligente.
- Producto 4: Desarrollar un Plan de monitoreo de contaminantes enfocado en la red de ciclovías, entregando la información de la calidad del aire a través de plataformas digitales de acceso público.

Dentro de los productos mencionados, los Productos 3 y 4 toman gran relevancia al momento de utilizar métodos inteligentes para la toma de decisiones y la información en tiempo-real para los usuarios y administradores de la infraestructura pública de transporte. Sin embargo, estos productos están orientados a fases de estrategia, diseño y planificación, sin considerar la implementación como factor fundamental del desarrollo exitoso de cada una de éstas, ni la difusión de sus resultados como valor agregado para dar a conocer a la población sobre ésta iniciativa y ayudar en la concientización de los problemas urbanos y medioambientales. Estas últimas dos falencias son abarcadas como los Productos 5 y Producto 6, respectivamente, del proyecto original.

1.1.2. Producto 5

Tal como se explica en la Sección 1.1.1, nace la necesidad de un nuevo Producto 5, el cual es definido como:

Diseño de la implementación de dispositivos de captura de datos en transportes y sus contaminantes en la red ciclovial del gran Santiago.

Este nuevo producto permite establecer un enlace práctico-técnico entre los Productos 4 y 5, demostrando un funcionamiento a escala (piloto) y permitiendo explicar y enseñar como ejecutar los diseños proporcionados por los productos enlazados a una escala mayor. Por lo tanto, **el presente reporte se enfoca en el desafío planteado en la construcción del Producto 5.**

1.2. Objetivos del Producto 5

A continuación se presentan las directrices generales y específicas que guían del desarrollo del producto 5.

1.2.1. Objetivo General (OG)

Desarrollar un diseño especializado de dispositivos de captura de datos en transportes y contaminantes asociados que sea implementado en la red ciclovial del Gran Santiago.

1.2.2. Objetivos Específicos (OEx)

Los hitos que marcan la concreción del objetivo general son los siguientes:

- **OE1:** Elaborar las especificaciones técnicas: instalación, operación y análisis de datos capturados; mediante el uso de dispositivos tecnológicos.
- **OE2:** Desarrollar un piloto a baja escala de la propuesta del sistema de captura de información.
- **OE3:** Reportar y planificar en un cronograma las tareas necesarias para la instalación y operación de la red, considerando los productos anteriores a esta iniciativa, detallando los costos de implementación.

1.3. Estructura del Reporte

El presente reporte se divide en tres partes: I) Propuesta de Solución, II) Diseño e Implementación de Solución, y III) Pilotaje y Resultados. Éstas, además, se estructuran de la siguiente forma:

- En la Sección 2, se presenta un análisis de antecedentes técnicos y prácticos de los componentes, la arquitectura de la propuesta, y la arquitectura del dispositivo principal de monitoreo.
- En la Sección 3, se presenta los componentes que incluye el producto mínimo viable (*Minimum Viable Product, MVP*), el presupuesto asociado a sus ejes, y se discute su escalabilidad, posibles mejoras y los costos asociados a éstas.
- En la Sección 4, se presenta el trabajo pre-pilotaje, para la calibración de sensores de bajo costo de calidad de aire. Se analizan temas de colocación, y el análisis del proceso de ajuste de los resultados a los obtenidos desde estaciones de monitoreo normativas.
- En la Sección 5, se entregan los detalles de despliegue del dispositivo construido, y los elementos de configuración necesario para su funcionamiento.
- En la Sección 6, se discuten los resultados más relevantes generados a partir de los datos obtenidos por el dispositivo durante el pilotaje.
- En la Sección 7, se concluye sobre la propuesta y su pertinencia con respecto a los objetivos del producto y las habilidades del equipo de trabajo.
- Finalizando, en la Sección 8, se presenta una pequeña reseña del equipo de trabajo a cargo del desarrollo del Producto 5.

Parte I

Propuesta de Solución

2. Requerimientos de Solución para Producto 5

A partir de los Términos de Referencia del Producto 5, y de los objetivos y la información levantada por la planificación del Producto 4 (Osses y cols., 2023), la presente propuesta para el proyecto se divide **en dos requerimientos**:

1. Creación de un dispositivo IoT, para ser pilotado en metodología aprender-haciendo (learning-by-doing), encargado de monitorear, levantar y transmitir información respecto a las variables ambientales y de tránsito-urbano requeridas:
 - Medición de Masa Material Particulada, específicamente MP2.5 ($2.5[\mu m]$ de diámetro).
 - Medición de Concentración de Gas, específicamente Óxidos de Nitrógeno (NO_x): NO, NO₂.
 - Reconocimiento y conteo de Vehículos Motorizados en ruta.
 - Reconocimiento y conteo de movilidad no motorizada en ruta (peatones y ciclos).
2. Informe de Salida que explique una planificación de ejecución de escalamiento del proceso de pilotaje a nivel de producción definido en los Productos 3 y 4 del proyecto global. Este reporte incluye los matices, información y decisiones determinadas, a nivel presupuestario, práctico y cronológico, obtenidas en el proceso de planificación y ejecución del piloto usando el dispositivo del Eje 1.

El primer eje permitirá cumplir con el OE1 y el OE2, definiendo especificaciones técnicas-tecnológicas para medir usando dispositivos IoT, y probando los componentes seleccionados en un proceso de pilotaje por 14 días; 7 días previo de intervención en la infraestructura ciclovial por parte del GORE-RM, y 7 días posterior al término de dicho proyecto urbano. Por su parte, el segundo eje cumplirá a cabalidad con el OE3, aterrizando a nivel conceptual, práctico, presupuestario y normativo, las labores necesarias para escalar el proceso de pilotaje, de acuerdo a los diseños propuestos en los Productos 3 y 4 del Ciclovías Inteligentes y Transporte sostenible para la RM.

2.1. Arquitectura de la Solución

Concentrándonos en la parte teórica-práctica de los ejes de la Propuesta, vale decir en la creación y despliegue del dispositivo IoT que permitirá realizar las labores de monitoreo y recopilación de información, se presentan las diferentes arquitecturas de los dispositivos y plataformas contenidos en el Producto 5.

- **Arquitectura del Dispositivo IoT:** El dispositivo IoT base para la medición integrará, tal como se muestra en la Figura 2.1, de dos módulos principales:
 1. Módulo de Calidad del Aire: Conjunto de sensores independientes para contaminantes de MP2.5, NO y NO₂, conectados a un Single-Board-Computer (desde ahora SBC), Raspberry Pi, el cual consolidará y almacenará la información localmente, generando registros con las mediciones de contaminantes en intervalos de tiempo predefinidos. Se debe además considerar que estas mediciones son afectadas considerablemente por variables meteorológicas como la temperatura y la humedad, por lo que se debe incluir la medición de éstas para monitorear el comportamiento de los sensores principales.
 2. Módulo de Detección de Objetos Viales: Una cámara IoT, adosada a una SBC Nvidia Jetson Nano, la cual contiene GPU y procesadores CUDA, para ejecutar en tiempo real algoritmos de Detección de Objetos mediante YOLO (Redmon y cols., 2016), una Red Neuronal pre-entrenada de alta precisión y velocidad, para conteo y estadísticas de vehículos en tránsito. El resultado que este dispositivo produce corresponde a la información recopilada respecto a la cantidad de vehículos y objetos detectados en forma de datos compilados en intervalos de tiempos predefinidos. En este resultado no se considera registro o transferencia de imágenes o videos.

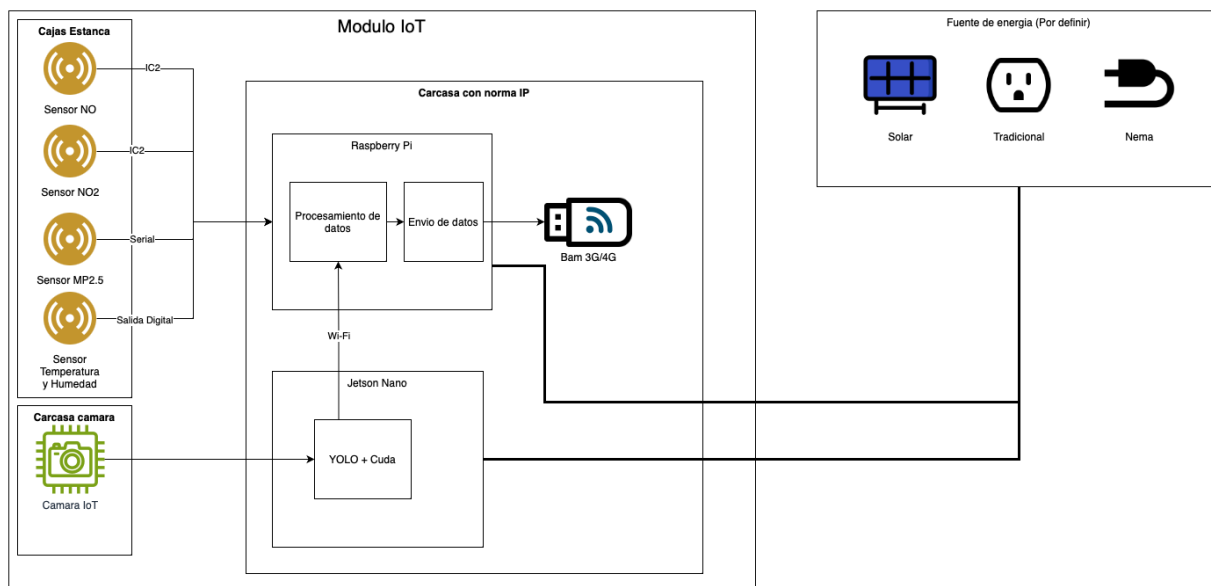


Figura 2.1: Arquitectura del dispositivo IoT de la propuesta.

Este prototipo, además contará, tres módulos secundarios de relevancia:

1. Integrador de Datos: La SBC encargada de recolectar los datos de los sensores, también servirá para recibir localmente los datos del módulo de visión por computador, de forma de centralizar en un solo lugar toda la información recolectada por los módulos principales.
2. Conectividad por Red Móvil: Aprovechando que este dispositivo está pensado para zonas urbanas, a través de un dispositivo BAM 3G o 4G adosado al integrador de datos, se pondrá a disposición un medio para poder enviar la información recolectada a algún servidor o contenedor, ya sea local o en la nube, para almacenar y analizar la data. El protocolo final, dependerá de la cantidad de información que deba ser transmitida.
3. Entrada de Energía: Las SBCs del dispositivo funcionan, cada una, con 5V AC/DC, lo cual puede ser provisto por diferentes medios. Dentro del conjunto de alternativas se consideran las siguientes:
 - Paneles Solares: Los paneles proveen energía independiente del cableado publico o privado, teniendo de esta forma un modulo prácticamente inalámbrico. Esto presenta desafíos como mantener un flujo de energía constante durante todo el día o cumplir con las especificaciones de alimentación de los dispositivos.
 - Socket NEMA: Esta opción permite aprovechar el cableado del alumbrado publico. La altura de estos conectores podrían generar problemas de logística e instalación.
 - Toma de corriente convencional: Esta opción implica el uso de una toma de corriente instalada en las cercanías, ya sea en un inmueble publico o privado. La desventaja de utilizar estas tomas de corriente radica en que están orientadas a un uso domestico y no urbano, por lo que su confiabilidad es baja.

La elección dependerá, en el prototipo, del lugar de instalación y las condiciones existentes.

- **Arquitectura del Producto 5:** Una vez construido el dispositivo IoT base, será pilotado en un ambiente reducido que intente emular la integración entre los Productos 3 y 4 del Proyecto, correspondiendo a la labor principal del Producto 5.

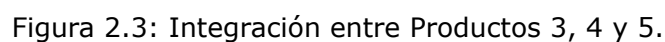
Como se muestra en la Figura 2.2, la propuesta del Producto 5 se articula de dos entornos:

- Entorno de Captura: Compuesto principalmente por el dispositivo IoT desplegado en puntos estratégicos, con tal de obtener información relevante respecto a las variables explicitadas por el Producto 4 (Osses y cols., 2023). Este dispositivo además enviará

- **Entorno Remoto de Almacenamiento y Visualización de Datos:** Compuesto de un servicio API encargado por un lado de recibir los datos de sensado y almacenarlos de forma estructurada en una base de datos relacional. Por otro lado el servicio API también dispone la información de la base de datos a un servicio de visualización el cual estará disponible para que los usuarios accedan vía navegador web. Se visualizará la data a través de una interfaz básica, para demostrar que este proceso puede ser llevado a plataformas más complejas de Analítica de Datos. Este entorno pilotará la contraparte del Producto 3.



Para finalizar las arquitecturas de diseño, se presenta la arquitectura de la integración entre los diversos productos principales del proyecto de *Ciclovías Inteligentes y Transporte sostenible para la RM* asociadas con la propuesta Producto 5 (véase Figura 2.3).



Observamos que el Producto 5 funciona como puente Técnico-Práctico entre los Productos 3 y 4, que definen dos sistemas a nivel de diseño. Nótese que el flujo de la información va desde el Producto 4 al Producto 3.

2.2. Antecedentes Producto 5

La propuesta presentada en la Sección 2.1, se sustenta en una revisión de antecedentes técnicos y científicos, en las distintas áreas que ésta cubre.

2.2.1. Medición de Contaminantes a bajo costo

A nivel de sensores ambientales, existe variada literatura científica sobre el uso de sensores para medición de materiales contaminantes. A continuación se mencionan los contaminantes más relevantes respecto al uso de sensores de bajo costo:

1. Material Particulado (MP): Componente ambiental altamente dañino, corresponde a partículas en suspensión aérea. Su forma más estudiada es el MP2.5 ($2,5[\mu m]$ de diámetro), el cual entra fácilmente al sistema respiratorio y penetra al sistema sanguíneo. Los efectos de este contaminante son: reducción de la capacidad pulmonar, aumento de los síntomas respiratorios (como alergias, irritaciones, asma, etc.), aumento del riesgo de ataques cardiovasculares y de arritmia. También, se estudia el MP10 ($10[\mu m]$ de diámetro) que se diferencia del MP2.5 por la capacidad penetrante en los pulmones, pues el MP10 afecta principalmente al sistema respiratorio y no llega a la sangre. En los últimos años, el MP1 ($1[\mu m]$ de diámetro) comenzó a ser relevante al mostrar una mayor incidencia en las afecciones respiratorias en niños que el MP2.5.
2. Óxidos de Nitrógeno (NO_x): Los óxidos de nitrógeno son gases invernaderos que combinan nitrógeno y oxígeno en distintas proporciones y que mayormente se generan cuando se queman combustibles. El estudio de los NO_x se centra en el Óxido Nítrico (NO) y en el Dióxido de Nitrógeno (NO_2). Las principales fuentes de NO_x son los automóviles, las centrales eléctricas y otras fuentes industriales. En la atmósfera, los óxidos de nitrógeno pueden contribuir a la formación de smog, provocando alteraciones de salud en la población y contribuyendo al calentamiento global.
3. Dióxido de Azufre (SO_2): Gas contaminante que tiene origen industrial cuando se queman fósiles que contienen azufre, como el petróleo, el carbón y la madera. En los últimos años la industria ha disminuido la emisión de SO_2 en diversos países por la incorporación legal de tecnologías de desulfuración.

El análisis de sensores de bajo costo y su efectividad ha sido, en los últimos 5 años, sujeto de variados estudios científicos a lo largo del orbe. Destacamos los siguientes artículos de origen nacional y extranjero:

- En (Cáquilpan y cols., 2019), la Superintendencia de Medio Ambiente (SMA) chilena formula un análisis de uso de diversos sensores de bajo costo para armar dispositivos IoT para medición y toma de decisiones. Estos dispositivos, centrados en MP2.5 y MP10, fueron desplegados en diversos puntos de la Región Metropolitana, calibrados y comparados con estaciones normativas y los propios datos generados por cada dispositivo de bajo costo construido.
- En (Feenstra y cols., 2019), un estudio similar al realizado por la SMA en Chile es hecho en la Costa Sur del Estado de California, E.E.U.U., con la finalidad de comparar doce sensores, con valores entre US\$150 y US\$3000, con respecto a los dispositivos normativos de dicho Estado.
- En (Kang y cols., 2022), se revisan 112 estudios científicos sobre sensores de bajo costo para los contaminantes mencionados anteriormente, con el objetivo de verificar si sus efectividad mejora aplicando las recomendaciones de calibración hechas por la Agencia de Protección Ambiental de E.E.U.U.

- En (Karagulian y cols., 2019), diversos métodos de calibración, basados en machine learning, son evaluados para ver el impacto tienen en sensores de bajo costo para los contaminantes mencionados anteriormente.
- En (Morawska y cols., 2018), una extensa revisión de sensores de bajo costo, su generalización y que tan lejos han llegado para resolver problemas generalizados es discutido.

En este abanico de revisión científica, las conclusiones son claras: **los sensores de bajo costo son una gran herramienta de medición**, pero deben ir a la par de procesos de calibración y comparación estaciones normativas para un mejor desempeño, incluyendo referencias de organizaciones gubernamentales de diferentes países desarrollados. A esto se le suma, que muchos de los sensores son creados para labores ad-hoc, en las cuales las variables de entorno donde son utilizadas juegan un rol fundamental en los niveles de precisión, careciendo de métodos para el monitoreo de contaminantes finos o complejos como el MP1 o CH4 (metano).

A pesar de estas limitantes, **el ingreso al mercado de estos sensores ha cambiado el paradigma de como atacar diversos problemas ambientales**, lo cual es una de las herramientas predilectas para levantar **información de entrada para decidir políticas públicas bases y decisiones de aplicación de monitoreo normativo de mayor costo**.

2.2.2. Visión por Computador: Detección de Objetos en Tiempo Real

Para el producto 5 es necesario un modelo de detección de objetos en tiempo real utilizando visión por computadora. La herramienta más utilizada para estas labores es YOLO (Redmon y cols., 2016) (You Only Look Once), un algoritmo basado en Redes Neuronales Convolucionales (CNN) que a partir de cada frame del stream de vídeo, divide la imagen en celdas e indica bajo cierta probabilidad en cada celda la detección de algún objeto que este dentro de los datos de entrenamiento del algoritmo. Algunas aplicaciones de YOLO en la industria pueden ser vigilancia (Ali y cols., 2021), Robótica (Cao y cols., 2021) o Vehículos autónomos (Golcarenenji y cols., 2023).

2.2.3. YOLO para Detección de Objetos

Los algoritmos de detección basados en YOLO se caracterizan por su flexibilidad en el uso de recursos, de tal manera que balanceando entre throughput (cantidad de *frames* procesados por segundo) y calidad de las detecciones es posible ejecutarlo en un sistema embebido que tenga soporte de procesamiento por GPU (e.g., la línea NVIDIA Jetson). Sin embargo, es importante mencionar que YOLO no es un algoritmo perfecto. La calidad de las detecciones se ve influenciada por el tamaño de cada objeto en el *frame* y por la oclusión entre objetos (Ferrante y cols., 2024), por ejemplo, se verán posibles errores en las detecciones cuando un objeto es muy pequeño o está muy alejado de la cámara, o cuando un objeto está enfrente de otro, ocultándolo parcial o completamente.

Entre las alternativas disponibles se consideran las versiones YOLOv4 (Bochkovskiy y cols., 2020), YOLOv5 (Jocher, 2020), YOLOv6 (Li y cols., 2022), YOLOv7 (Wang y cols., 2022) y YOLOv8 (Jocher y cols., 2023), las cuales no son excluyentes, ya que poseen desempeños diferentes dependiendo de la situación, vale decir, para cierto problema una versión más antigua puede tener mejor desempeño que una nueva, y viceversa. Todos los algoritmos basados en YOLO cuentan con una gran variedad de modelos pre-entrenados disponibles de forma gratuita en sus respectivos repositorios, lo que permite poder elegir entre modelos más livianos y rápidos y modelos más pesados y más lentos, pero con mejor desempeño.

2.2.4. Alternativas a YOLO

A pesar de dominar el área de detección de objetivos, existen otras alternativas, clasificadas en dos grupos: soluciones de bajos recursos basadas en redes neuronales convolucionales (o CNN, del inglés *Convolutional Neural Network*) y soluciones de altos recursos basadas en *transformers*.

En cuanto a la detección y clasificación de objetos, las soluciones basadas en *transformers* lideran el estado del arte dado la alta precisión que logran en éstas tareas (Shehzadi y cols., 2023). Sin embargo, este tipo de soluciones tienen un mayor consumo de memoria de vídeo y requieren de un mayor poder de cómputo para poder usarlas de manera efectiva, razón por la cual dejan de ser una opción viable para sistemas embebidos, como la Jetson Nano, dado que éstos no cuentan con los recursos necesarios.

Por otro lado, las soluciones basadas en CNN son una opción mucho más abordable. La simplicidad de este tipo de modelos hace que requieran menos memoria que las soluciones basadas en *transformers* y al mismo tiempo menos poder computacional, entregando la posibilidad de ejecutar estos algoritmos casi en tiempo real, con la desventaja de que la calidad de las detecciones de éste tipo de modelos es comparativamente menor a su contraparte.

Algunas de las opciones basadas en CNN que fueron consideradas son las R-CNN (del inglés *Region-based Convolutional Neural Network*) (Girshick y cols., 2014) y las SSD (del inglés *Single-Shot object Detection*) (Liu y cols., 2016):

- R-CNN es un modelo basado en CNN introducido por académicos de la UC-Berkeley en 2013 y fue uno de los primeros modelos en lograr resolver la tarea de detección de objetos con una precisión aceptable, inspirando nuevas iteraciones para poder realizar detecciones en tiempo real, como Fast R-CNN (Girshick, 2015) y Faster R-CNN (Ren y cols., 2016). Estos modelos se volvieron el estándar de la época y sus resultados eran comparables a los obtenidos con YOLOv3, sin embargo, a diferencia de YOLO, éstos modelos utilizan más recursos y confunden con mayor frecuencia el fondo o background de una imagen por objetos inexistentes (Ammar y cols., 2019), razón por la que se decidió no optar por esta alternativa.
- SSD también es un modelo basado en CNN, en particular, VGG-16 (Simonyan y Zisserman, 2015) con pasos adicionales para detectar o clasificar cada objeto en una imagen, logrando resultados comparables a los modelos basados en R-CNN e inspirando nuevas iteraciones como InceptionV1 (Ning y cols., 2017), MobilenetV1 (Howard y cols., 2017) y MobilenetV2 (Sandler y cols., 2019). Esta última es una opción común en sistemas embebidos para la clasificación de imágenes gracias a su alto desempeño (comparable con YOLOv5 small (Naftali y cols., 2022)) y bajo uso de recursos, incluso en dispositivos sin GPU. Este modelo, en particular SSD MobilenetV2, es nuestra segunda opción en caso de no obtener resultados cercanos al tiempo real con YOLO.

2.2.5. Visión por Computador: Identificación, Seguimiento y Contador de Objetos

La primera parte de la solución propuesta abordó la detección de elementos en tiempo real, mientras que la segunda parte consiste en la identificación y conteo de los elementos detectados, es decir, asignarle una "identidad" (o *ID*) a cada elemento detectado y mantener dicha identidad a medida que el objeto detectado se mueve dentro del vídeo capturado por la cámara para contar cada elemento una sola vez, tarea fundamental para realizar un buen contador de elementos.

Para abordar este problema se consideró inicialmente el uso de modelos de inteligencia artificial basados en Deep SORT (del inglés *Deep Simple Online Realtime Tracking*) (Wojke y cols., 2017), los cuales consisten en la integración de un modelo basado en CNN para la identificación de elementos (por ejemplo OSNet Re-ID (Zhou y cols., 2019) o HRNet Re-ID (G. Zhang y cols., 2021)) junto a algún algoritmo que permita hacer seguimiento de cada *ID* (por ejemplo, *KLT Feature Tracker* (Lucas y Kanade, 1981, Tomasi y Kanade, 1991)), para finalmente obtener la cantidad de elementos únicos a partir de algún criterio arbitrario, por ejemplo, en base a la cantidad total de *IDs* generadas o en base a un contador automático que se incremente cada vez que un elemento identificado cruce una determinada zona de la pantalla.

Sin embargo, dado el alto uso de recursos de ese tipo de modelos alejándonos de la solución en tiempo real que necesitamos y múltiples problemas de compatibilidad entre las librerías que usan con la Jetson Nano, principalmente dadas las diferencias entre un sistema X86 (arquitectura de un computador normal) y un sistema ARM (arquitectura de la Jetson), se optó por buscar una

alternativa basada en algoritmos tradicionales.

Es por ello que la solución elegida consiste en un algoritmo basado en la técnica *Centroid Tracking*, el cuál toma en consideración las coordenadas de los BBox (del inglés *Bounding Box*) de cada objeto detectado y sus centros en ventanas de dos *frames*, lo que permite calcular la distancia euclidiana entre todos los BBOX y centroides en un determinado lapso de tiempo y con ello hacer un seguimiento o *tracking* de cada objeto en movimiento sin que pierdan su *ID* a lo largo del tiempo.

La solución propuesta inicialmente tiene un desempeño menor por si sola al de las técnicas basadas en inteligencia artificial, especialmente porque requiere de condiciones casi ideales para operar consistentemente (por ejemplo, los objetos deben moverse, en lo posible, en líneas rectas con el menor número de cruces entre objetos posible), sin embargo, haciendo uso de una versión superior de este algoritmo, denominado ByteTrack (Y. Zhang y cols., 2021), combinado con un buen posicionamiento de la cámara y con los ajustes correctos es posible lograr resultados aceptables y comparables con los modelos basados en Deep SORT. Uno de esos ajustes consiste en determinar la zona del vídeo de la cámara con mayor precisión de seguimiento (es decir, la zona en donde se produzca la menor pérdida de *IDs* posibles), y elegir dicha zona como la región que deben cruzar los elementos para ser contados.

2.2.6. Obtención de Imágenes en Tiempo Real

Las imágenes que procesarán los algoritmos recomendados deben ser obtenidas directamente en terreno por una cámara, preferentemente en tiempo real, para lograr una medición del flujo vehicular precisa y sin retrasos. Las cámaras elegidas para la recolección de imágenes deben ser compatibles con los dispositivos utilizados, por lo que para este caso se tiene que las opciones más comunes son las siguientes:

- Cámara con conexión USB: Se conectan a los dispositivos mediante los puertos USB. Estas cámaras tienen diversos tamaños dependiendo del proveedor y suelen venir en un paquete completo que no requiere montaje. Su mayor ventaja es que son de fácil acceso, su precio no es elevado y utiliza conexión universal USB, por lo que se puede conectar a la mayoría de dispositivos. Otra ventaja de éstas cámaras, es que generalmente incluyen algoritmos integrados para estabilizar la imagen o auto enfoque de puntos de interés, mejorando la calidad de la imagen capturada a cambio de recibir la imagen final con un mínimo desfase en relación al tiempo real.
- Cámara con conexión CSI: Se conectan directamente a la interfaz serial de los dispositivos embebidos mediante un cable flex de 16 pines. Estas cámaras son de tamaño reducido y consisten únicamente de una pequeña placa y una cámara soldada en ella, por lo que es necesario considerar protección para evitar exponer sus componentes eléctricos al entorno. A diferencia de las cámaras USB, la conexión mediante CSI soporta mayores velocidades, al tener exclusividad para este tipo de componentes, ya que posee un protocolo especializado para recibir imágenes, siendo más estable y completo que lo ofrecido por el protocolo universal USB.
- Cámaras IP: En esta categoría entran todas las formas de captura de imágenes o vídeos de manera remota mediante internet. Para ello se requiere una cámara que pueda, de manera autónoma, transmitir el flujo de datos capturado hacia una red, o que se encuentre conectada a un dispositivo que realice esta labor.. La principal ventaja de esta categoría es que permite tener la cámara en terreno y el dispositivo de procesamiento en un entorno que permita optimizar sus labores de cómputo. sin embargo, existe una mayor latencia por: 1) la compresión, ruidos y errores de las imágenes para ser enviadas al dispositivo de cómputo y 2) el retraso y la inestabilidad que introduce este tipo de conexión, sobre todo si se depende de conexiones inalámbricas.

2.2.7. Microcontroladores IoT

Son necesarios dos tipos de microcontroladores:

1. Con GPU: Para esta sección se considera la plataforma Nvidia Jetson, la que corresponde a placas de desarrollo de aprendizaje automático de bajo costo y bajo consumo, diseñado para uso en aplicaciones de visión por computadora, procesamiento de lenguaje natural y aprendizaje de maquina autónomo (Corporation, 2023b). Se prefiere este tipo de placas con GPU debido a la presencia de núcleos CUDA (Corporation, 2023a), los cuales representan el factor más relevante en el rendimiento de aplicaciones *near-real time* de vision por computadoras. A continuación se muestran las alternativas disponibles.
 - Jetson AGX Xavier: Plataforma de alto rendimiento para robótica, vehículos autónomos y automatización industrial con inteligencia artificial, posee 512 núcleos de GPU y su precio oscila los \$1300 USD. Actualmente descontinuada.
 - Jetson Xavier NX: Versión de AGX Xavier de menor costo, con 384 núcleos de GPU y su precio oscila los \$600 USD. Actualmente descontinuada.
 - Jetson Nano: Plataforma de alto rendimiento, bajo costo y bajo consumo energético orientado a productos IoT, visión por computadora e inteligencia artificial. Posee 4 GB de RAM (compartida con la GPU), 128 núcleos CUDA y su precio oscila los \$250 USD. Entró en discontinuación en Enero de 2024.
 - Jetson Orin Nano: Sucesor de la Jetson Nano, está pensada como la plataforma de entrada para integrar inteligencia artificial con cámaras y robots. Posee 8 GB de RAM (compartida con la GPU), 1024 núcleos CUDA y 32 núcleos Tensor, lo que le permite ejecutar algoritmos de mayor tamaño y complejidad en comparación a su predecesor. Su precio oscila los \$500 USD.
 - Jetson AGX Orin: Su poder de cómputo se compara al de tener 8 dispositivos Jetson AGX Xavier y está pensada para llevar la inteligencia artificial generativa al "edge". Posee 32 GB de RAM (compartida con GPU), 2048 núcleos CUDA y 64 núcleos Tensor, su precio oscila los \$2500 USD.
2. Sin GPU: En este caso se barajan varias opciones de microcontroladores, será el encargado de recopilar la información de los sensores y la plataforma de visión por computadora para su posterior envío al servidor de datos.
 - Raspberry Pi: Computadora de bajo costo y alto rendimiento utilizada en amplia gama de proyectos, pequeña y con bajo consumo energético. Posee conectores I2C para la conexión de dispositivos IoT como sensores (Foundation, 2023a).
 - Arduino: Plataforma electrónica Open-Source orientada al sensado e IoT. Cuenta con pines análogos y digitales para la conexión de componentes electrónicos como sensores, transistores, diodos, etc. A través de módulos llamados "Shields" es capaz de aumentar sus funcionalidades como ampliar la cantidad de sensores o sumar un módulo Wi-Fi (Arduino, 2023).
 - Raspberry Pi Pico: Alternativa a la plataforma Arduino de menor costo pero también menores capacidades, posee pines GPIO, I2C, SPI y UART. Está basada en el Microcontrolador RP2040 (Foundation, 2023b).
 - ESP32: Placa electrónica orientada a proyectos IoT, basada en el microcontrolador Ten-silica Xtensa LX6, de bajo costo y bajo consumo energético, Wi-Fi y Bluetooth integrado y pines GPIO, I2C, SPI y UART (Systems, 2023).

2.2.8. Conectividad y Protección de Dispositivos

1. **Conectividad:** Para el modulo IoT se considera el uso de redes 3G/4G a través de una BAM USB pos-pago (proveedor por definir), esto debido a la cobertura ofrecida en el ambiente de carácter urbano donde será el pilotaje, la confiabilidad (up-time) que entregan estas redes y la poca energía que consumen en comparativa con otras tecnologías. En cuanto a protocolos de red a utilizar, para administrar y facilitar el acceso remoto a los equipos se usa SSH a través de los puertos 22 de las maquinas, mientras que el envío de datos de censado y los datos generados por el modulo de detección de objetos considera el uso de HTTP hacia el servidor de datos.

2. **Protección de dispositivos:** Tanto el modulo IoT como la cámara IoT deben ser protegidas por carcasas categorizadas bajo la norma IP del IEC (IEC, 2013). Bajo esta norma, para soportar las diversas condiciones climáticas que se presentan en la zona de instalación, las clasificaciones aceptadas para las carcasas son IP66/67/68. En la Figura 2.4 se presenta una alternativa de **caja con norma IP67** para equipos IoT, mientras que la Figura 2.5, muestra una **carcasa** para el modulo de cámara de Raspberry Pi.

- Medidas carcasa cámara IoT: 62.9 x 69.1 x 33 [mm]
- Medidas caja IP67: 40 x 30 x 18 [cm]



Figura 2.4: Caja Norma IP67: Caja de referencia y ejemplo con Raspberry Pi



Figura 2.5: Carcasa con equivalente a norma IP67 para cámara Raspberry Pi 2/3

Parte II

Diseño e Implementación de Solución

3. MVP y Escalabilidad

3.1. MVP: Producto Viable Mínimo y sus módulos

El producto viable mínimo o **MVP** (del inglés *Minimum Viable Product*) corresponde a un prototipo integrado del dispositivo IoT propuesto en la Sección 2. Este dispositivo ha sido probado por partes, previo a su integración, y será nuevamente *testeado* en laboratorio previo a su despliegue en terreno. Según los Términos de Referencia del Producto 5, las mediciones deberán realizarse por 14 días, lo que contempla 7 días previo a la intervención en la infraestructura vial por parte del GORE-RM o de los municipios asociados correspondientes, y 7 días posterior al término de dicho proyecto urbano, de tal forma de contrastar el efecto del mismo.

3.1.1. Módulo Calidad del Aire

El módulo de Calidad de Aire, véase Figura 3.1, contiene la siguiente lista de componentes:

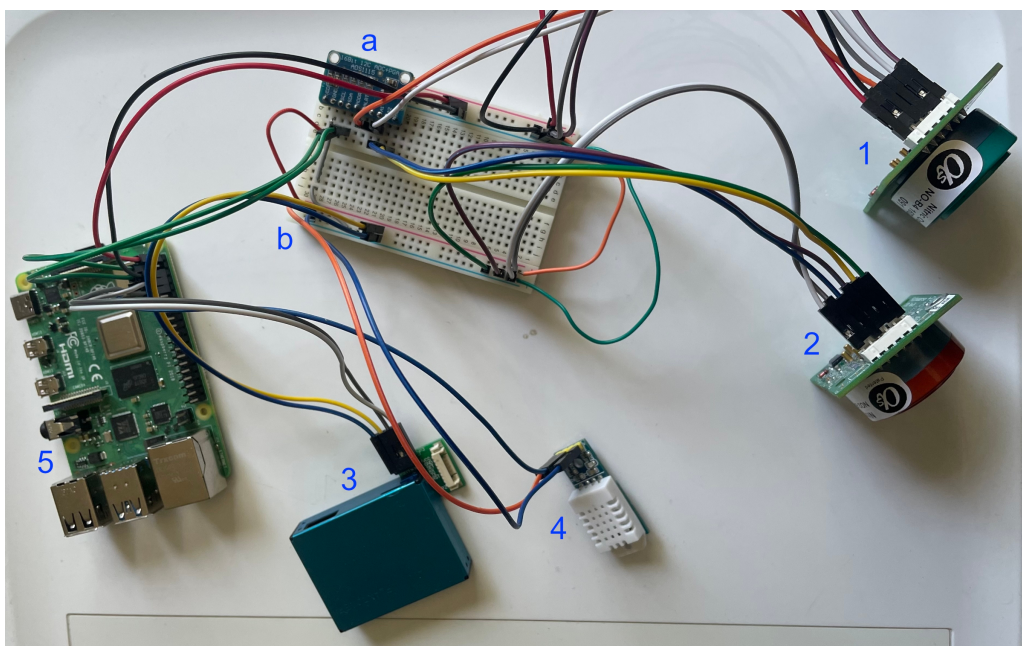


Figura 3.1: Módulo de calidad de Aire montado, sin BAM incluida

1. Sensor NO: Sensor análogo electroquímico AlphaSense NO-B4, montado sobre una placa ISB-B Modelo 01 (000-0ISB-01).
2. Sensor NO2: Sensor análogo electroquímico AlphaSense NO2-B43F, montado sobre una placa ISB-B Modelo 02 (000-0ISB-02).
3. Sensor MP2.5: Sensor digital de dispersión lumínica PlanTower PMS7003.
4. Sensor Meteorológico: Sensor digital DHT22 para medición de Temperatura y Humedad relativa del ambiente.
5. Single-Board-Computer (SBC) sin GPU: Dentro de las opciones propuestas en la Sección 2.1 se seleccionó el uso de una Raspberry Pi 4 de 4GB de RAM, debido a su versatilidad y compatibilidad en el mundo IoT, además de tener prestaciones de hardware más robustas para ejecutar tareas en paralelo. Éste dispositivo se alimenta por USB-C, con una fuente de 5v y 3A.
6. BAM SIM-Card USB: Para entregar conectividad mediante el uso de la red móvil chilena bajo formato prepago, una BAM USB marca Zte modelo Mf920u compatible con redes 3/4G se conecta al SBC seleccionado.

7. Para integrar todos los elementos del módulo, se agregaron los siguientes accesorios:

- a) Conversor Análogo Digital: Dentro los diferentes sensores se poseen señales análogas (NO, NO₂) y digitales (PMS7003, DHT22). El SBC seleccionado solo posee entradas digitales, por lo tanto el sistema debe incorporar un Conversor análogo-digital (ADC) ADS1115 de 16-bits, para que la señal análoga de ambos sensores de NO_x pueda ser recibida y manejada por la SBC.
- b) Protoboard: Para poder hacer la conexión entre los sensores y la SBC seleccionada, se utilizó una Protoboard de 400 puntos, para ordenar los circuitos y tener un lugar para montar el ADC para las señales análogas.

A modo de resumen, la arquitectura concreta del módulo de calidad de aire diseñado para el MVP, se puede observar en la Figura 3.2.

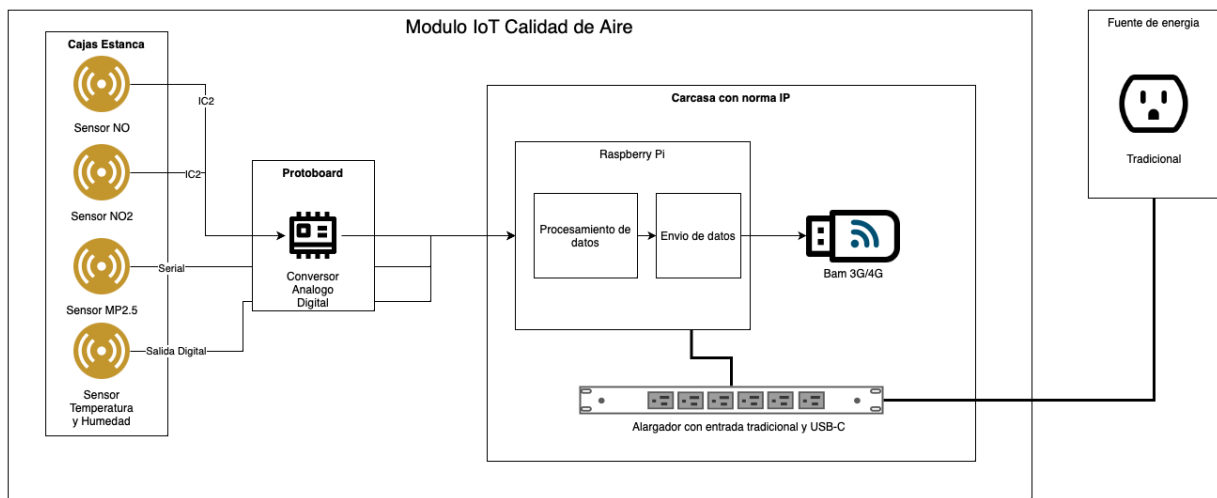


Figura 3.2: Arquitectura del Módulo de Calidad de Aire.

3.1.2. Módulo de Detección y Conteo de Objetos Viales

El módulo de Detección de Objetos Viales, véase Figura 3.3, contiene la siguiente lista de componentes:

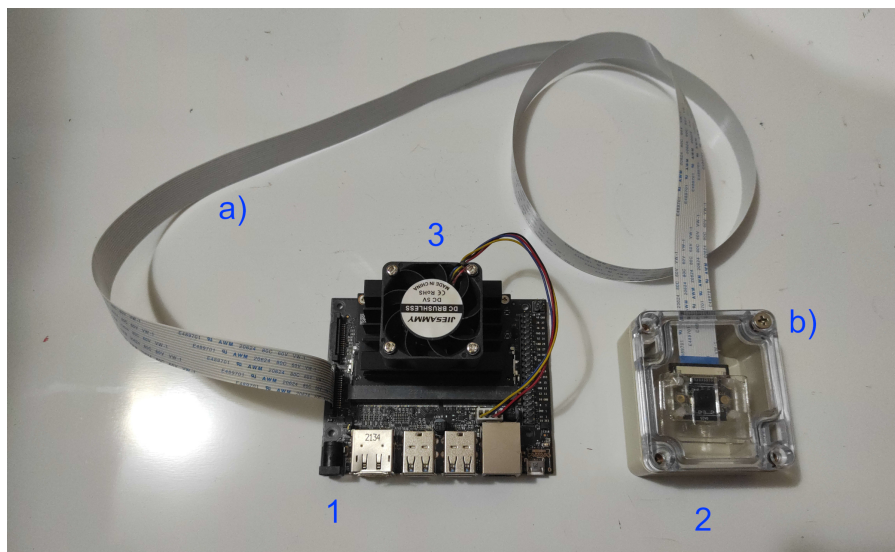


Figura 3.3: Módulo de Detección y Conteo de Objetos Viales.

1. Single-Board-Computer (SBC) con GPU: Por accesibilidad (costo, stock, etc.) se decidió por el Development Kit de Jetson Nano de 4GB. Sin embargo, se debe considerar a futuro que estos fueron descontinuados a inicios de 2024 y los módulos normales están pensados para ser descontinuados a partir del 2027, por lo que los dispositivos debiesen migrar Jetson Orin Nano para desarrollos en el mediano y largo plazo.
2. Cámara: Raspberry Pi 2 Camara con CSI, de 16 pines, modelo IMX219, con angular de 77°.
3. Refrigeración: Ventilador de 5V con 4 pines, de tamaño 40x40x20 [mm].
4. Para integrar todos los elementos del módulo, se agregaron los siguientes accesorios:
 - a) Flex 85mm de 15 pines de 1 metro de longitud.
 - b) Carcasa para cámara: caja de conexiones impermeable de plástico con tapa de acrílico transparente, de tamaño 63x58x35 [mm].

A modo de resumen, la arquitectura concreta del módulo para contar vehículos y peatones diseñado para el MVP, se puede observar en la Figura 3.4.

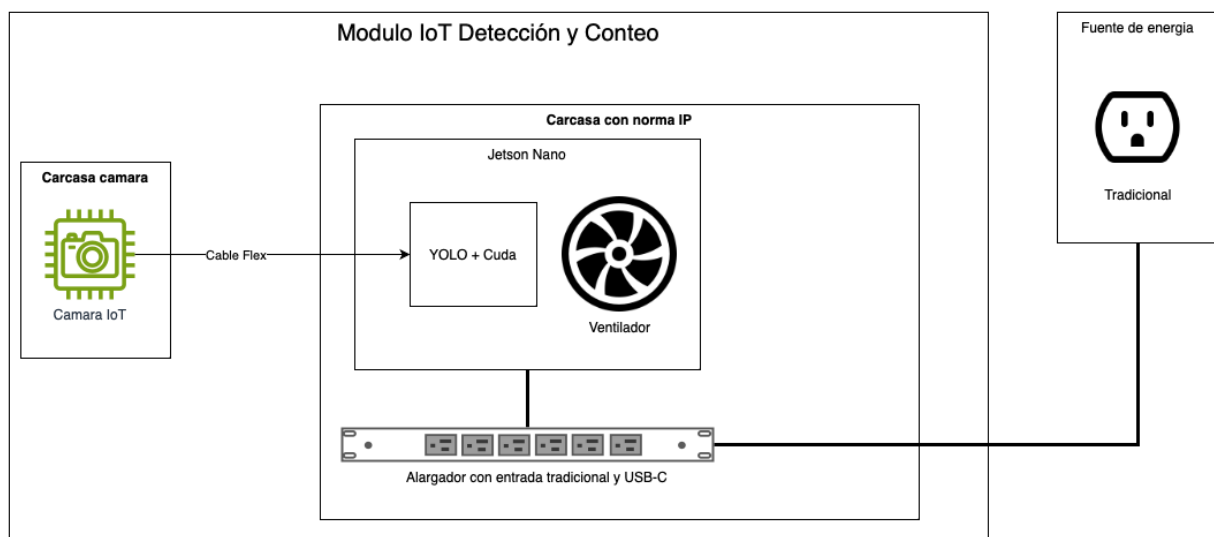


Figura 3.4: Arquitectura del Módulo de Detección y Conteo de Objetos Viales.

La SBC escogida, Jetson Nano, funciona en buenas condiciones en conjunto a YOLO, sin embargo necesita de configuraciones específicas para obtener el máximo rendimiento para las labores del producto diseñado. A continuación se presenta un análisis de las opciones probadas y las seleccionadas en diversos elementos relevantes:

1. Kit de Compilación Jetson: JetPack 4.6.1 (L4T 32.7.1) con CUDA 10.2 y OpenCV v4.5 (con *G-stream*).
2. Modos de Energía:
 - **Max Performance** (modo de energía utilizado): Requiere de una alimentación de 10W (sin incluir la energía de periféricos como la cámara o el ventilador) para que haga uso completo de sus 4 procesadores y de todos sus núcleos CUDA. Al usar este modo con una alta carga computacional, como es el caso del algoritmo para detectar y contar elementos, se genera una mayor temperatura en las componentes internas del SBC, haciendo que se apague forzosamente al superar el límite de seguridad. Para asegurar un funcionamiento continuo y evitar que la Jetson Nano alcance temperaturas extremas es necesario el uso de un sistema de refrigeración como componente principal, en este caso, un ventilador de 5V.
 - **Low Power**: Requiere de una alimentación de 5W (sin incluir la energía de periféricos como la cámara o el ventilador) para que haga uso de 2 de sus 4 procesadores y una

fracción de sus núcleos CUDA. Este modo produce mayor estabilidad y menos calor que el modo *Max Performance* mientras sacrifica el ya restringido poder de cómputo del SBC.

3. Lenguaje de Programación: El desarrollo de la solución se hizo usando el lenguaje de programación *C* junto al *framework Deepstream 6.0.1*.
4. Versión de YOLO: Se implementa YOLOv8, la cuál presenta múltiples mejoras en el algoritmo de detección de objetos en comparación a versiones anteriores, y permitió lograr el mejor balance entre velocidad de procesamiento y calidad de las detecciones, bajo la condición de *near-real time*. Cabe considerar que el modelo Lite de YOLOv4 logró alcanzar la mayor velocidad de ejecución tras sacrificar en gran medida la calidad de las detecciones, mientras que los modelos pequeños y medianos de las versiones 7 y 8 de YOLO lograron los mejores resultados, siendo la versión 8 ligeramente mejor en escenarios con más ruido o flujo vehicular.
5. Algoritmo de Conteo de Objetos: Algoritmo ByteTrack descrito en la Sección 2.2.5.

En las pruebas realizadas logramos observar la cantidad de imágenes por segundo que procesa cada una de las variantes de YOLO probadas para resolver la tarea de detección de elementos: YOLOv4, YOLOv5, YOLOv7 y YOLOv8, registrando un resumen de los mejores modelos encontrados en el Cuadro 1.

Modelo	Variante	Cuantización	Image Size	FPS	Factor de detección
YOLOv8	M	FP16	640	4	50.2
YOLOv8	S	FP16	640	9	44.9
YOLOv7	Tiny	FP32	640	12	38.7
YOLOv5	S	FP16	640	10	37.4
YOLOv4	Tiny	FP32	640	39	38.1

Cuadro 1: Comparación de la variante candidata elegida para cada modelo de YOLO en base a la calidad de las detecciones. La fila coloreada representa el modelo con mejor calidad observada en las detecciones realizadas durante la fase de pruebas.

Una de las ventajas de la Jetson Nano es que permite resolver operaciones en Punto Flotante de 16 bits (FP16) con mínima pérdida de calidad en los resultados, por lo que aprovechamos esa ventaja y cuantizamos todos los modelos que originalmente se encontraban representados en Punto Flotante de 32 bits (FP32), acelerando en gran medida la velocidad de cómputo. Esta cuantización podría haber sido más agresiva, llegando a Enteros de 8 bits (INT8), en caso de haber usado un SBC de la generación de la Jetson Orin. Otra característica de la serie Jetson Orin, es que es al menos 80 veces rápida que la Jetson Nano utilizada (declarado por el equipo de *Nvidia*), por lo que los FPS registrados en el Cuadro 1 al menos duplicarán su valor si se ejecutaran en estos nuevos dispositivos, permitiendo la integración de modelos que presenten mejor desempeño para obtener mejores resultados en *near-real time*.

Otro ajuste que permite acelerar la velocidad de cómputo es reducir el Image Size o resolución interna de YOLO. Sin embargo, esto reduce considerablemente la calidad de las detecciones, en particular, de los objetos más pequeños como las bicicletas o los peatones.

En conclusión, tal como se muestra en el Cuadro 1, dentro de las mejores variantes de cada modelo probado, se observa como el modelo YOLOv8-M es el que genera las mejores detecciones de elementos, pero con un procesamiento menor de FPS, siendo imposible una detección *near-real time*. Esto se debe a un trade-off por los recursos necesarios para tener un gran nivel de detección, implicando un mayor tiempo de procesamiento que reduce la cantidad de información o imágenes que se puede procesar en una unidad de tiempo.

Una alternativa que encontramos para poder incrementar el número de FPS de cada modelo es introduciendo un parámetro llamado *Frame Skip* al algoritmo. Este parámetro se encarga de

hacer que nuestro detector procese 1 *frame* y luego se salte los siguientes N *frames*, lo cual produce un aumento de FPS proporcional al valor de N con una pérdida mínima en la calidad de las detecciones cuando $N \leq 4$, haciendo posible utilizar YOLOv8-M con una velocidad de procesamiento entre los 15 y los 20 FPS y YOLOv8-S con una velocidad de 30 FPS, satisfaciendo la necesidad del *near-real time*.

En el Cuadro 2 se observa el desempeño de 4 alternativas probadas para la tarea de identificación y seguimiento de elementos: Deep Sort con OSNet, Deep Sort con HRNet, Centroid Tracking y ByteTrack (véase Sección 2.2.5). Éste cuadro, que incluye el parámetro *Frame Skip* con $N = 4$, contiene los mejores valores obtenidos. Considerando que todas las opciones son válidas al momento de tener un seguimiento del movimiento de los objetos detectados, nos decantamos por ByteTrack, ya que al utilizar solo 0.02 segundos para realizar esta labor, afecta mínimamente al proceso de detección de objetos y al mismo tiempo logra un desempeño considerablemente superior al visto con el algoritmo Centroid Tracking. Sin embargo, estos métodos necesitan ajustes *in situ* para obtener resultados satisfactorios.

Modelo	Variante	SPF
Deep SORT	OSNet	0.6
Deep SORT	HRNet	0.2
Algoritmo	Centroid Tracking	0.01
Algoritmo	ByteTrack	0.02

Cuadro 2: Comparación de las técnicas candidatas para la tarea de identificación, seguimiento y conteo de elementos. La columna SPF (del inglés *Seconds Per Frame*) representa la cantidad de segundos que le toma a cada algoritmo procesar cada frame del vídeo, entre menor sea este valor, mejor.

Sobre los lenguajes de programación y *frameworks* utilizados, los resultados del Cuadro 1 se obtuvieron al ejecutar YOLOv4 usando *Tensorflow* en *Python*, al igual que las dos variantes de Deep SORT presentes en el Cuadro 2, mientras que los resultados de YOLOv5, v7 y v8 se obtuvieron al ejecutar los respectivos modelos en C usando el framework *Deepstream*.

Luego de hacer comparaciones internas con YOLOv5 en *Python* con *PyTorch* y en C con *Deepstream* se determinó que existe una mejora en el uso de recursos de memoria RAM y una mejora cercana al 10 % en la velocidad de cómputo en la versión desarrollada en C utilizando los pipelines propuestos por *Nvidia* en la documentación de *Deepstream*.

Finalmente, la configuración elegida para el piloto del producto 5 consiste en utilizar YOLOv8-S como nuestro detector de objetos, acompañado por el algoritmo de *ByteTrack* para la identificación, tracking y conteo de objetos en *Deepstream*. La solución propuesta con la configuración elegida alcanza una velocidad de cómputo cercana a las 24 imágenes por segundo al utilizar un *Frame Skip* con $N = 2$, velocidad suficiente para satisfacer la necesidad de *near-real time*. Se elige YOLOv8-S por sobre los otros modelos ya que consideramos que es un balance adecuado entre velocidad de cómputo, calidad de detecciones y pérdida de información al tener un bajo valor de N en comparación al requerido para ejecutar YOLOv8-M ($N = 4$) en *near-real time*.

3.1.3. Integración y Montaje

Para integrar los módulos de Calidad de Aire y de Detección y Conteo de Objetos Viales en una única unidad, tal como se muestra en la Figura 3.5, se han utilizado los siguientes elementos:

1. Protección SBC: Gabinete Eléctrico Metálico con Puerta, bajo norma IP65, de 250x200x150 [mm] para resguardar e integrar todo el armado electrónico de ambos módulos.
2. Protección Sensores: Dos Cajas Estancas con tapa y conos de 150x100x70 [mm], bajo norma IP55, para resguardar los sensores. Una de las cajas estanca aloja los sensores análogos electroquímicos y, la otra, almacena los sensores digitales.



Figura 3.5: MVP: Primera vista producto integrado.

3. Protección Cámara: Caja estanca con tapa de acrílico transparente, bajo norma IP65, de 63x58x35 [mm] para resguardar la cámara que va conectada a la Jetson Nano para llevar a cabo el conteo de elementos. Esta caja va asegurada a un brazo plástico, y éste va asegurado al gabinete eléctrico.
4. Extensor de cable de energía: Dentro de las opciones de obtención de energía eléctrica, se decidió por construir el prototipo que utilice un único enchufe tradicional de tipo C (16A) o L (10A), especificados en la Norma Chilena de Electricidad NCH-2027-1-2008 y NCH-2027-2-2008. De esta forma se energiza todos los módulos del sistema con un extensor de conexiones, que además posee salida USB-C, que se ha integrado dentro del producto.
5. Accesorios y componentes internas:
 - a) Repisa 3D para SBC's: Impresión 3D de un soporte para almacenar las SCBs de ambos módulos de forma de aprovechar de mejor forma el espacio.
 - b) Soporte 3D para Sensores: Impresión 3D de un soporte para poder colocar las cajas estancas de conjunta, y que sus tapas no topen con el movimiento de la puerta del Gabinete Eléctrico.
 - c) Sensores de temperatura, humedad y PM 2.5.
 - d) Sensores de NO y NO2.
 - e) Rieles de fijación: Rieles perforados que se adosan al gabinete eléctrico para montar el equipo, con amarras de tipo abrazadera, en cualquier poste o perfil metálico.

A modo de resumen, la arquitectura concreta del MVP se observa en la la Figura 3.6.

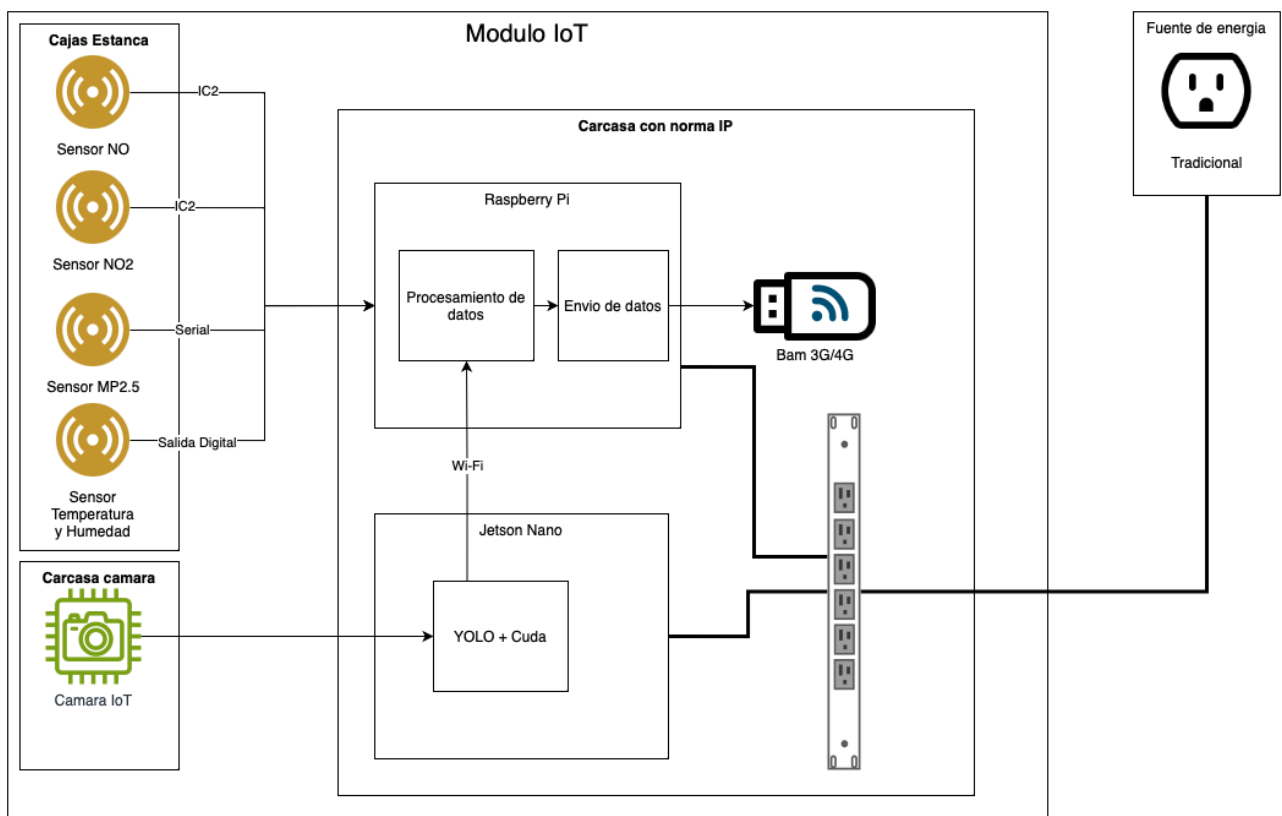


Figura 3.6: Arquitectura de MVP del Producto 5.

La integración genera un MVP único, de peso aproximado 4.2 [kg], el cual posee las dimensiones especificadas en las Figura 3.7 para uso sin fijación, y en la Figura 3.8, para uso con fijaciones.

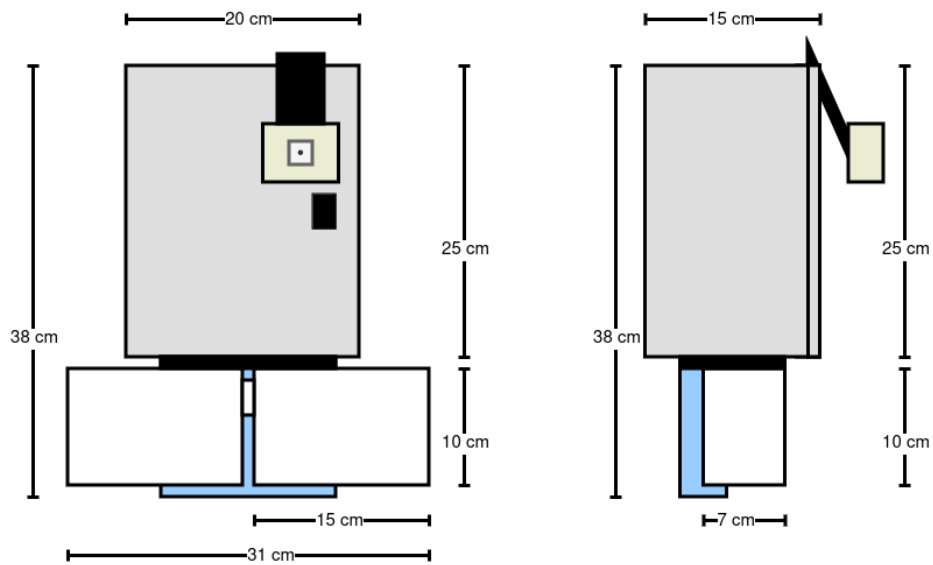


Figura 3.7: MVP: Dimensiones del prototipo base.

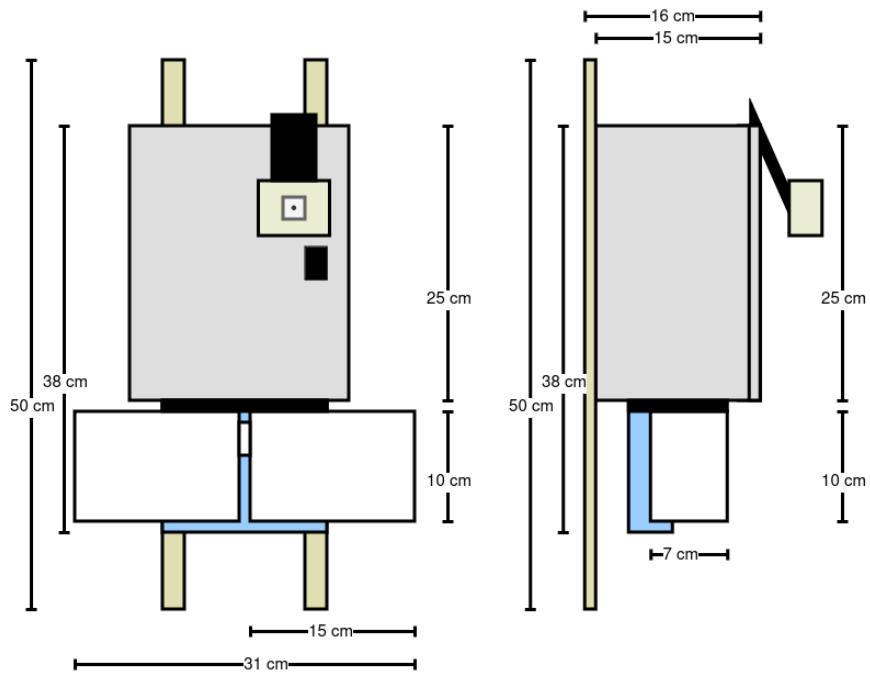


Figura 3.8: MVP: Dimensiones del prototipo con rieles de montajes.

3.1.4. Salida de Datos

El funcionamiento de ambos módulos en conjunto, resumido en la arquitectura de la Figura 3.9, pasa por una serie de procesos para terminar en una visualización estandarizada de la información obtenida:

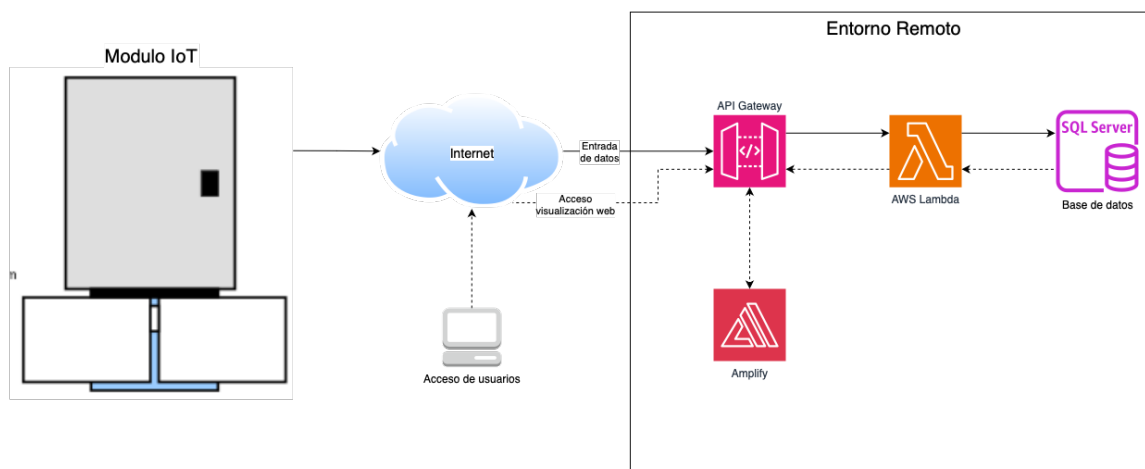


Figura 3.9: Arquitectura de Salida de Datos.

1. Comunicación Jetson Nano - Raspberry Pi: Dado que la Raspberry Pi obtiene conectividad a través de un dispositivo externo (BAM), se utiliza su dispositivo WI-FI como un hotspot inalámbrico, de tal forma que la Jetson Nano se pueda conectar a él y así poder transferirle la información relacionada al conteo vial a la Raspberry Pi.
2. Consolidación de la información: Una vez reunida la información de ambos módulos en la Raspberry Pi, esta la consolida en un archivo `.json` (véase Figura 3.10). Luego, este archivo es enviado a un servidor por medio de una Interfaz (API) para ser almacenada en una base de datos y posteriormente manejada por los distintos componentes que puedan utilizar esta información.

```
{
  "Id": 1,
  "Device": "test",
  "Pm1_0cf1": 172.27586206896552,
  "Pm2_5cf1": 250.06896551724137,
  "Pm10cf1": 256.6034482758621,
  "Pm1_0": 114.05172413793103,
  "Pm2_5": 165.91379310344828,
  "Pm10": 170.31034482758622,
  "N0_3": 30794.379310344826,
  "N0_5": 8407.224137931034,
  "N1_0": 1415.6206896551723,
  "N2_5": 100.32758620689656,
  "N5_0": 9.206896551724139,
  "N10": 3.5344827586206895,
  "No": 26.304682065534806,
  "No2": 15.733238761032702,
  "Temp": 31.02068965517241,
  "Hum": 30.95862068965519,
  "Ts": 1701990697.446332,
  "Date_of_sample": "2024-02-26T22:12:56Z"
}
```

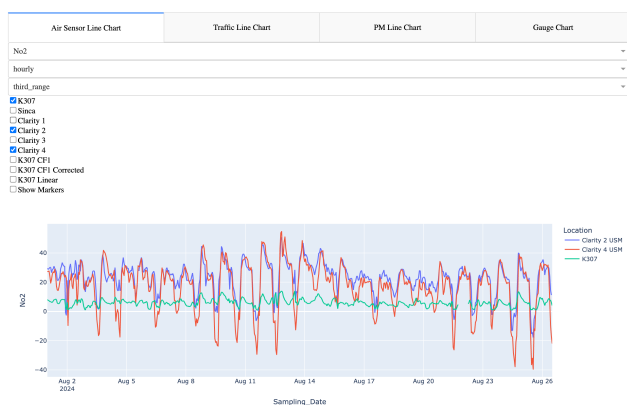
```
{
  "Id": 4,
  "Device": "JetsonNano_1",
  "Pedestrian": 12,
  "Bicycle": 2,
  "Motorcycle": 5,
  "Car": 26,
  "Truck": 4,
  "Bus": 5,
  "Date_of_sample": "2024-03-02T15:27:52Z"
}
```

Figura 3.10: Información de calidad del aire recopilada por ámbos módulos en formato `.json`.

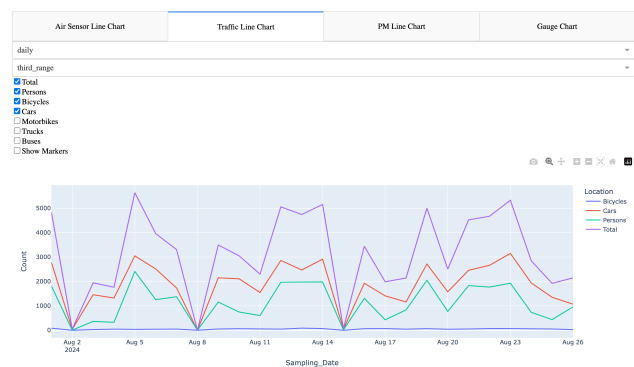
3. Interfaz (API) de lectura y escritura:

- a) AWS Lambda: Se desarrolló una API en *GoLang* para gestionar las escrituras y lecturas a la base de datos, y se optó por el servicio *AWS Lambda* para dejarla en línea y contar con una solución *serverless* y estable dentro del corto plazo para llevar a cabo el piloto. Los costos asociados a este servicio rondan los 0.2 USD por cada millón de consultas, por lo que resulta conveniente dada la baja escala del piloto inicial.
 - b) Gateway: Este servicio de AWS es fundamental para tener un punto de acceso abierto hacia la API montada en AWS Lambda, y tiene un valor de servicio de 1 USD por cada un millón de consultas que se hagan a la API mediante el punto de acceso de *Gateway*.
 - c) Base de Datos: Para almacenar los datos recopilados por los sensores se utiliza el servicio *Free Tier* de *Turso*, el cuál permite levantar hasta un máximo de 500 bases de datos SQLite en 3 ubicaciones distintas, incluyendo Santiago, con una capacidad total de 9 GB. Este servicio tiene como restricción que el número mensual de lecturas a las bases de datos está limitado a un billón de consultas, mientras que el número mensual de escrituras a 25 millones. A modo de referencia, si la Raspberry Pi enviara datos cada 1 minuto por 30 días, la cantidad de escrituras realizadas rondaría las 43200.
 - d) Amplify: Esta solución de AWS permite alojar plataformas Web o Móvil y se utiliza para mantener en línea el visualizador visto en la Figura 3.11. Los costos asociados a este servicio rondan los 0.3 USD por cada un millón de consultas.
4. Visualizador de Datos: Haciendo uso de la API, un visualizador de datos puede acceder a la información de la base de datos y utilizarla para mostrar la información más relevante obtenida a través de tablas y gráficos. En este prototipo, un pequeño *dashboard* con gráfico de línea, tacómetros y revisión de tablas puede ser accedido de forma remota (véase Figura 3.11). Cabe recordar que el despliegue exitoso de esta parte corresponde al Producto 3 del proyecto PNUD - GORE-RM.

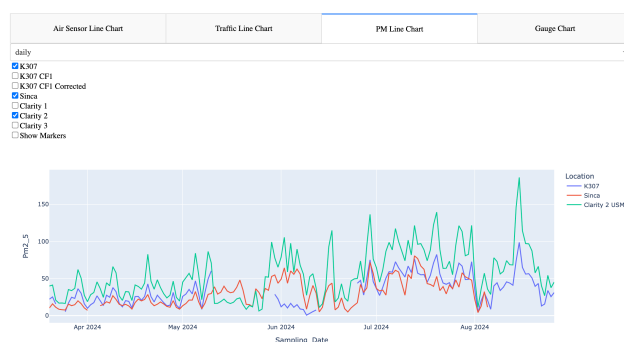
Air Sensor and Traffic Data Visualization



Air Sensor and Traffic Data Visualization



Air Sensor and Traffic Data Visualization



Air Sensor and Traffic Data Visualization

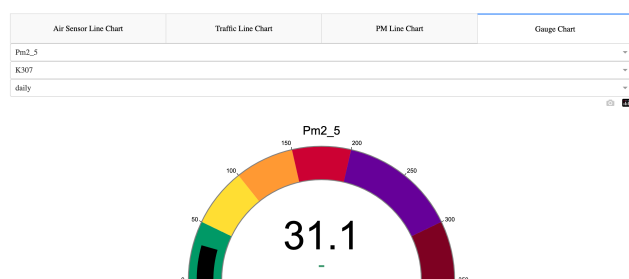


Figura 3.11: Front-end de visualización de datos.

3.2. Presupuesto

A continuación se resume el presupuesto del núcleo del Producto 5 (véase Cuadro 3) ¹.

Componentes	Monto por unidad (USD)	Unidades	Monto total
Nvidia Jetson Nano	250	2	500
Raspberry Pi Camera Module 3	30	2	60
Raspberry Pi 4	150	2	300
Sensor MP2.5	250	2	500
Sensor NO	200	2	400
Sensor NO2	200	2	400
Carcasas Norma IP	150	2	300
Accesorios	100	2	200
HH Equipo de Trabajo	800	3	2400
Total			5060

Cuadro 3: Presupuesto de Componentes Producto 5

En este presupuesto no se considera lo siguiente:

- Los costos de instalación del dispositivo en las zonas de pilotaje. Debido a que se realizará a través de gestiones del GORE-RM con las unidades administrativas locales correspondiente.
- Los costos de servicio de recepción, almacenamiento y visualización de datos. Los cuales serán alojados en un servidor a cargo del Consultor de la presente propuesta. Esta labor debe ser provista, en el despliegue definitivo de la red de monitoreo, por quienes se adjudiquen la implementación del Producto 3.

3.3. Escalabilidad, Mejoras y Costos Asociados

En esta sección se analizan distintos requerimientos que la contraparte ha consultado para ver la factibilidad de incluirlas en el proyecto actual.

3.3.1. Clasificación de Ciclos

Para la clasificación de ciclos puede ser necesario re-entrenar el modelo clasificador si es que este no cuenta con la información previa de éstos (como es, por ejemplo, el caso de los patines o los scooters).

Si se opta por re-entrenar el modelo para agregarle nueva información, hay que tener presente que se necesitan entre 2000 y 10000 imágenes del nuevo objeto para que el modelo empiece a reconocerlo consistentemente. Estos números son referenciales, es posible requerir una mayor cantidad si es que la proporción entre las imágenes del nuevo objeto y las imágenes de todos los otros objetos dentro del modelo es muy pequeña para evitar que el nuevo objeto quede sub-representado y sea potencialmente ignorado por el modelo final.

Para la recopilación de imágenes se pueden considerar las siguientes opciones:

- Utilizar imágenes disponibles en internet: Si son objetos comunes, es posible que exista algún dataset en plataformas de acceso abierto². No hay requisitos específicos para las imágenes en sí, éstas deben contener al objeto que se quiere detectar y pueden presentarse con distintos niveles de iluminación, ángulos y tamaños o resoluciones, siempre y cuando sigan el formato de etiquetado requerido por el modelo, en este caso, YOLO³. En caso de que éstas no cumplan con el formato del modelo, se requerirá un grupo de personas para que las re-etiquete siguiendo el formato correcto, dado que si bien existen herramientas que

¹Valores estimados al 09 de Marzo de 2024

²<https://huggingface.co/datasets>

³https://docs.ultralytics.com/yolov5/tutorials/train_custom_data/

facilitan ésta tarea⁴, el etiquetado para entrenamiento es un proceso manual que requiere conocimiento experto y no puede ser automatizado completamente.

En el caso de querer usar imágenes de internet, tener en consideración que algunos autores o sitios web protegen su propiedad intelectual para evitar que se utilicen en el entrenamiento de modelos o en cualquier acción que involucre inteligencia artificial, por lo que queda a criterio y responsabilidad de todas las partes involucradas verificar el origen de las imágenes utilizadas y el cumplimiento de las restricciones que puedan existir sobre cada una.

- Recopilar y etiquetar un conjunto de imágenes manualmente (tomadas en terreno): Dependiendo de la cantidad y del tipo de objetos nuevos que se quieren agregar al clasificador se podría requerir una cantidad considerable de tiempo, por lo general son proyectos de mediano a largo plazo, considerando por lo menos una ventana de 6 meses dado que además de tener que recopilar las imágenes, se necesita que un grupo de personas las vaya etiquetando una por una. Las imágenes recopiladas y el etiquetado deben seguir las mismas especificaciones planteadas en el punto anterior.

Ambas opciones para la recopilación de imágenes pueden ser lideradas por un/a Ingeniero/a de Datos que supervise y valide los procesos de recolección y etiquetado de imágenes (sueldo base estimado entre \$879.039 y \$4.596.608 según nivel de *seniority*⁵).

En caso de que se opte por la segunda opción (recopilar imágenes manualmente) se necesitaría también un equipo de personas que se reparta la tarea de salir a terreno a recopilar imágenes con una cámara. El tiempo requerido por imagen dependerá de que tan difícil sea encontrar el objeto de interés y de que tan difícil sea movilizar personas hacia esas áreas, por lo que en estos casos se define un salario entre \$5.000 y \$6.000 por hora, por persona (incluir también posibles costos asociados al transporte o al equipamiento), tomando como referencia el sueldo base de un/a Ingeniero/a de Datos *Junior* o recién ingresado/a en el área (al rededor de \$900.000) dado que la toma de fotos se considera como una tarea de recopilación y/o extracción de datos.

Para el proceso de etiquetado de imágenes se suele asignar al menos tres personas por imagen para reducir el riesgo de etiquetas incorrectas y detectar factores de riesgo oportunamente, como por ejemplo, imágenes irrelevantes, ambigüedad en el tipo de objeto que se quiere detectar, etiquetas incorrectas, entre otros. En estos casos se suele asignar un sueldo base equivalente al de un/a Ingeniero/a de Datos *Junior* o recién ingresado/a en el área (cercano a los \$900.000) por persona dado que el etiquetado se considera como una tarea de preparación y procesamiento de datos, y se espera que en promedio una persona etiquete cerca de 20 a 25 imágenes por hora luego de acostumbrarse al flujo de trabajo⁶. Es decir, en caso de tener 10000 imágenes habría que etiquetar 30000 imágenes (3 personas por imagen) y se necesitarían entre 1200 y 1500 horas de trabajo aproximadamente, sin considerar factores irregulares que puedan afectar el desarrollo de ésta tarea.

Para el re-entrenamiento del modelo, lo que se suele hacer es arrendar una instancia o un servidor en la nube que cuente con múltiples GPU de alta gama. A modo de referencia, Amazon cuenta con un servicio de este tipo y ofrece servidores en la nube con 8 GPU A100 bajo un sistema de pago on-demand, en el que cada hora de uso está evaluada entre 32.77 y 40.96 USD⁷, tal que un entrenamiento corto o no riguroso del modelo mediano de YOLO usando la configuración de servidor descrita toma aproximadamente 1.5 días o cerca de 36 horas, lo que tendría un costo aproximado entre 1179.72 y 1474.56 USD por entrenamiento⁸.

3.3.2. Clasificación de Peatones

La solución que se está desarrollando permite detectar personas y clasificarlas como peatones, pero avanzar más allá y separar a las personas por grupo etario, grupo étnico o género requeri-

⁴<https://roboflow.com/annotate>

⁵https://www.glassdoor.com.mx/Sueldos/chile-data-engineer-sueldo-SRCH_IL.0,5_IN49_KO6,19.htm - (última revisión: 30 de octubre 2023)

⁶Información y cifras basadas en trabajos previos de etiquetado por parte de los miembros del equipo de trabajo

⁷<https://aws.amazon.com/ec2/instance-types/p4/> - (última revisión: 23 de octubre 2023)

⁸<https://github.com/ultralytics/yolov5/pull/5194#issuecomment-957488176>

ría, como mínimo, poder capturar la cara de cada persona con el mayor nivel de detalle posible, lo que de por sí sería complicado si consideramos una cámara fija en altura como es en este caso.

Y por otra parte, requeriría de dispositivos con poder de cómputo mayor al de un SBC para asegurar compatibilidad con los modelos utilizados y un nivel de calidad sólido y consistente en las clasificaciones sin sacrificar el registro de datos en tiempo real, dada la complejidad y los requerimientos de las soluciones modernas (Kuprashevich y Tolstykh, 2023), las cuales necesitan máquinas con tarjetas de vídeo o poder de cómputo de alta gama para poder procesar múltiples imágenes en tiempo real.

Además de eso, requeriría de la implementación de un nuevo modelo especializado para esa tarea debido a que se escapa de las capacidades que tiene YOLO como segmentador y clasificador de imágenes ligero.

Dado que estas soluciones no son compatibles con el equipamiento presentado en las propuestas (tanto a nivel de *hardware* como de *software*) y la falta de soporte existente para resolver el problema de clasificación de peatones se omite estimación de gastos. Esta tarea por sí sola ameritaría un proyecto de investigación y desarrollo dedicado para determinar la factibilidad y el equipamiento necesario para llevarla a cabo.

3.3.3. Persistencia de Vídeo

En contexto de registro de vídeo para su revisión en casos de siniestros conlleva las siguientes implicancias de costos:

- Almacenamiento extra: Se deben costear los gastos de computación y almacenamiento en forma remota para los archivos de vídeo, también definir la duración de los registros. El servidor responsable de esta tarea puede ser un equipo físico, lo que implica costear el equipo y costos energéticos y de administración. Este tipo de servidores físicos para almacenar datos suelen tener un costo inicial cercano a los 2000 USD⁹, sin embargo cuentan con una cantidad inicial limitada de espacio para almacenamiento, por lo que habrá que considerar gastos adicionales para aumentar el espacio disponible cada vez que se llene esa capacidad y para agregar configuraciones de redundancia que permitan proteger los datos recopilados en caso de que algún disco presente fallas con el tiempo (cerca de 200 USD por cada disco duro adicional, la capacidad y el precio varía de proveedor a proveedor). Para asegurar la disponibilidad del servidor es necesario un equipo UPS (uninterruptible power supply), los cuales tienen valores cercanos a los 1200 USD¹⁰. Otro factor a considerar en términos de conectividad y cómputo es que un solo servidor puede no ser suficiente si la cantidad de dispositivos enviándole información supera el ancho de banda o el poder de procesamiento del servidor.

La otra opción es un servidor en la nube (AWS, Azure, GCP, etc.), lo cual implica gastos mensuales de mantención y administración. A modo de referencia, un servicio de almacenamiento básico en AWS de acceso frecuente usando S3 tiene un valor de 0.0405 USD por GB almacenado al mes y un costo inicial de 0.15 USD por GB descargado al mes, en caso de que se quiera descargar y acceder al contenido almacenado¹¹.

En ambos casos considerar que una sola grabación en 1080P 30Hz (resolución estándar de una cámara *Full HD*) utiliza aproximadamente 20 GB de almacenamiento por cada 24 horas de contenido, dependiendo del nivel de compresión usado en el vídeo. Esto para 100 puntos en un mes de vídeo se traduce en aproximadamente 2400 USD mensuales.

- Conectividad: El volumen de datos generados por la persistencia es mayor en varias ordenes de magnitud con respecto a lo planteado para el pilotaje, por lo que es necesario un protocolo de conexión a internet de mayor ancho de banda como lo podría ser 5G, una conexión cableada a cada módulo IoT o acceso a una red Wi-Fi privada. A modo de referencia, un

⁹<https://www.dell.com/en-us/shop/dell-poweredge-servers/sr/servers>

¹⁰<https://a.co/d/b3wAhKd>

¹¹https://aws.amazon.com/s3/pricing/?did=ap_card&trk=ap_card

modem 5G de exterior tiene un valor de entrada cercano a los 500 USD¹² y un plan de datos móviles 5G ilimitado en Entel tiene un valor de \$29.990 al mes¹³ (sin considerar promociones ni descuento por los primeros meses). Los costos asociados a una posible conexión cableada son dependientes a la infraestructura disponible en la zona de pilotaje y posteriormente en otros puntos de instalación, debido a que esta infraestructura varía de punto a punto se considera descartable como opción existiendo la disponibilidad de la red móvil.

- Implementos extras: Para que los datos de medición del modelo de detección de objetos no se vean alteradas por el registro de vídeo, es necesario ampliar el hardware así estos procesos se realicen de forma paralela, como por ejemplo una segunda cámara para cada modulo o un microcontrolador adicional, lo cual aumentaría el costo del modulo en 180 USD (Raspberry Pi 4 más Camera Module) según lo especificado en la sección 3.3.
- Acceso a los datos: Es necesario definir un esquema de acceso a los datos, es decir, como obtener segmentos de vídeos de fechas especificas en horas especificas, mantener un esquema de datos que permita esto y que no intervenga con procesos de ingesta de vídeo. El primer costo asociado es el tiempo de planificación y desarrollo de esta solución (HH del equipo en desarrollo de software). El segundo costo corresponde al despliegue de esta solución, el equipo considera una API hosteada en Lambda de AWS la cual retorna los links a los vídeos solicitados según filtros de fecha y hora. El costo de Lambda comienza en los 0.20 USD cada millón de solicitudes¹⁴.

Se recomienda un análisis detallado de parte de la contraparte para concluir si es estrictamente necesario integrar estas funcionalidades adicionales al producto 5, debido al alto costo y/o poca viabilidad presentada y la existencia de soluciones tradicionales, que no afecten el rendimiento del sistema original, debiesen ser más económicas como cámaras de seguridad para la persistencia de vídeo.

¹²<https://store.zyxel.com/catalogsearch/advanced/result?search=NR7101> - última revisión: 30 de octubre 2023

¹³<https://www.entel.cl/planes/oferta-portabilidad/> - última revisión: 30 de octubre 2023

¹⁴<https://aws.amazon.com/es/lambda/pricing/> - última revisión: 30 de octubre 2023

Parte III

Pilotaje y Resultados

4. Calibración Sensores de Calidad de Aire

4.1. Colocación

Si bien los sensores de bajo costo vienen con pruebas y calibraciones de laboratorio realizadas por el fabricante, la recomendación general es ajustar las mediciones de éstos a las efectuadas por instrumentos normativos del país de utilización, vale decir, reducir al mínimo la diferencia entre los valores entregado por ambos componentes. De esta forma, el primer paso consiste en generar una colocación del dispositivo de bajo costo, en un radio de no mayor a 2[m] respecto a instrumentos especializados, con la finalidad de asegurar que la toma de muestras para ambos componentes ocurran en el mismo entorno, para luego poder realizar los ajustes mencionados,

En el caso del dispositivo diseñado, se realizó una colocación con los instrumentos normativos del Ministerio de Medio Ambiente (MMA) ubicados en la estación de monitoreo del Sistema de Información Nacional de Calidad del Aire ubicada en el Parque O'Higgins (véase Figura 4.1), dado que ésta cuenta con mediciones correspondientes a los sensores de bajo costo incluíos en el piloto del Producto 5: MP2.5, NO y NO₂.

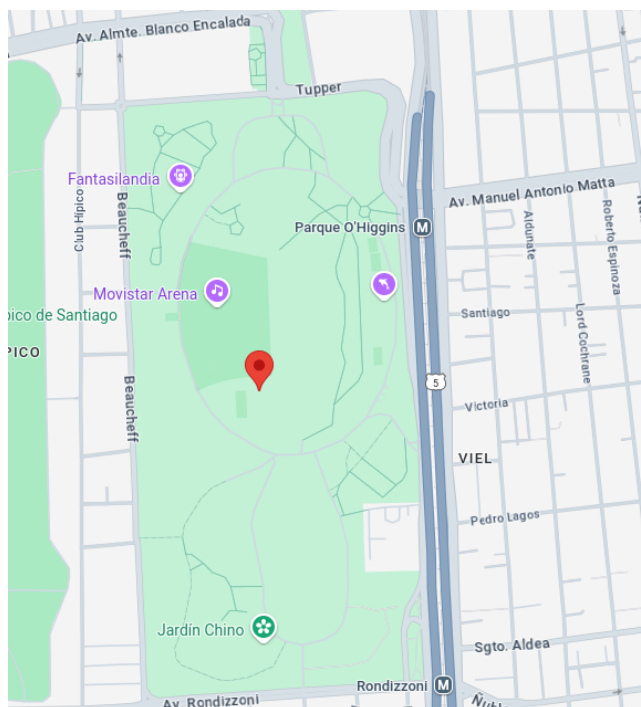


Figura 4.1: Ubicación de la Estación de Monitoreo del SINCA en Parque O'higgins.

La colocación de los sensores se hizo en el techo de la estación del SINCA (véase Figura 4.2), en donde se encuentran la toma de muestras de los instrumentos normativos, a una distancia menor de 2 [m] de éstas, y considerando una altura desde el suelo entre 2.5 y 3 [m] (véase Figura 4.3).



Figura 4.2: Estación de Monitoreo del SINCA ubicada en Parque O'Higgins.



Figura 4.3: Colocación de dispositivos en Estación de Monitoreo del SINCA en Parque O'higgins. Los tres dispositivos con paneles solares negros son sensores del proveedor Clarity. El dispositivo con el gabinete eléctrico gris corresponde al prototipo con los sensores de bajo costo diseñado para el Producto 5.

4.2. Calibración de toma de datos.

Según lo discutido en la Sección 2.2.1, se dedicó un periodo de colocación de siete semanas (entre 20/03/2024 y 10/05/2024, inclusive) para calibrar los sensores de bajo costo, en las dependencias del SINCA ubicadas en Parque O'Higgins, contrastando y ajustando las mediciones del prototipo respecto a las publicadas por el Ministerio de Medio Ambiente. Este ajuste fue realizado utilizando técnicas de aprendizaje estadístico, en particular, regresiones lineales para los tres contaminantes de de interés: MP2.5, NO y NO₂.

El uso de la técnica de regresión lineal permite obtener transformaciones por medio de una recta necesaria para aproximar los valores numéricos de los sensores de bajo costo a los valores entregados por los dispositivos normativos presentes en el SINCA. Los resultados del ajuste muestran que la aproximación para MP2.5 y NO tienen un coeficiente de determinación R^2 de 0.83 y 0.86, respectivamente, mientras que para NO₂ ronda el 0.57 (véase Cuadro 4). Entre más cercano se encuentren los valores de R^2 a 1, mejor serán las aproximaciones de los sensores calibrados. Mientras más se acerque el MSE a cero, más cercanos serán los valores calibrados a los medidos en la estación SINCA del Parque O'higgins.

	PM _{2,5}			NO			NO ₂		
	Train	Val	Test	Train	Val	Test	Train	Val	Test
R^2	0,80	0,77	0,83	0,87	-84,88	0,86	0,77	-15,21	0,57
MSE	17,59	18,59	15,74	212,70	92980,66	207,42	31,60	1994,67	96,76

Cuadro 4: Coeficientes de determinación (R^2) y Errores Cuadráticos Medios (MSE) obtenidos con la calibración para las etapas de entrenamiento, validación y testing de PM 2.5, NO y NO₂.

El ajuste logrado, a través de regresión lineal, se puede observar en la Figura 4.4, la Figura 4.5 y la Figura 4.6 para MP2.5, NO y NO₂, respectivamente. En estas imágenes se observa que los puntos azules, correspondientes al cruce entre el valor obtenido por los sensores de bajo costo y el valor reportado por el SINCA, se ajusta de buena forma a la recta roja, entregada por los regresores utilizados. Entre éstas, se destaca que la calibración de MP2.5 es la que presenta mayor estabilidad a simple vista, mientras que las de NO y NO₂ presentan algunas anomalías ("outliers") que se alejan de la curva de ajuste propuesta. Indagando en los datos generados durante la colocación, se encontraron múltiples oportunidades donde la humedad relativa del ambiente fue superior al 85 %, sobrepasando el umbral máximo reportado en los sensores de NO y NO₂, siendo este un posible factor en las anomalías observadas durante la calibración.

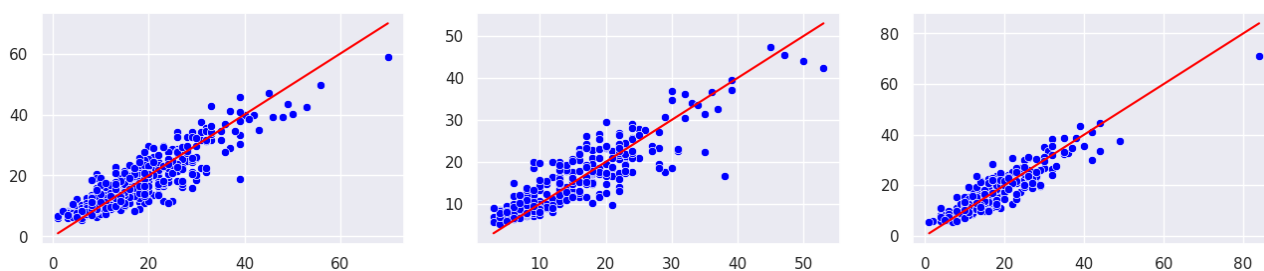


Figura 4.4: Calibración de Sensores MP2.5: Curva de aproximaciones obtenidas sobre el sensor PLANTOWER PMS7003. La recta roja es la curva de ajuste entregada por la regresión lineal, y los puntos azules representan los cruces entre el dato del sensor bajo costo y el valor en ese momento reportado por SINCA. Los resultados se dividen en tres conjuntos de datos: entrenamiento, validación y testing. Entre más se acerquen los puntos azules a la recta roja, mejor es la calibración.

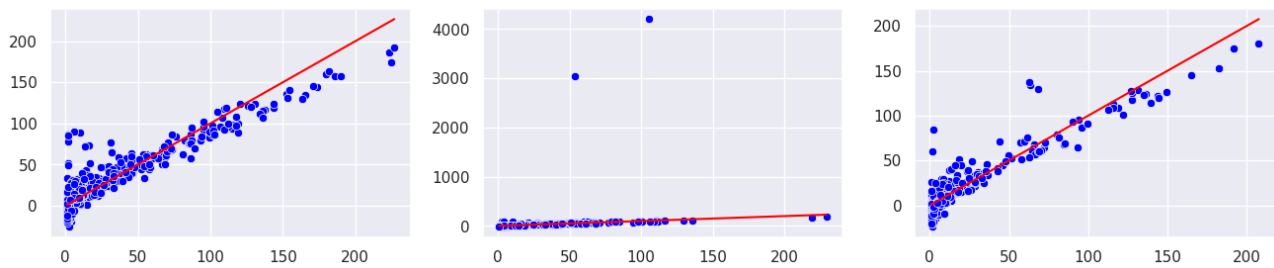


Figura 4.5: Calibración de Sensor NO: Curva de aproximaciones obtenidas sobre el sensor Alpha-Sense NO-B4. La recta roja es la curva de ajuste entregada por la regresión lineal, y los puntos azules representan los cruces entre el dato del sensor bajo costo y el valor en ese momento reportado por SINCA. Los resultados se dividen en tres conjuntos de datos: entrenamiento, validación y testing. Entre más se acerquen los puntos azules a la recta roja, mejor es la calibración.

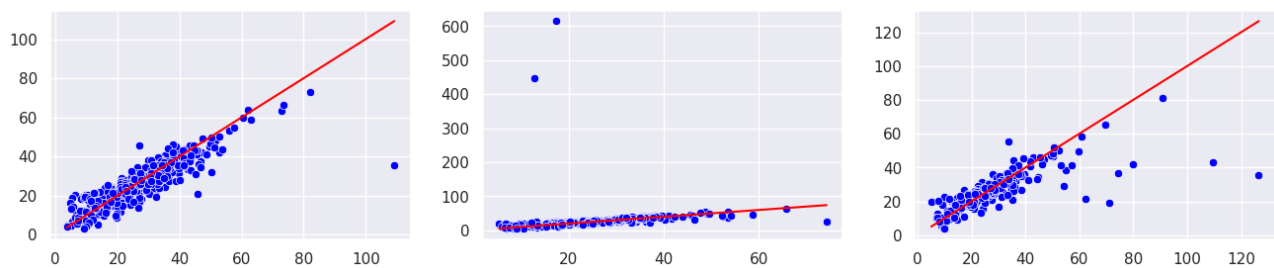


Figura 4.6: Calibración de Sensor NO₂: Curva de aproximaciones obtenidas sobre el sensor Alpha-Sense NO₂-B43F. La recta roja es la curva de ajuste entregada por la regresión lineal, y los puntos azules representan los cruces entre el dato del sensor bajo costo y el valor en ese momento reportado por SINCA. Los resultados se dividen en tres conjuntos de datos: entrenamiento, validación y testing. Entre más se acerquen los puntos azules a la recta roja, mejor es la calibración.

Pese a estas anomalías, la calibración realizada permite obtener aproximaciones visualmente similares a las reportadas por los instrumentos normativos del SINCA (véase Figura 4.7, Figura 4.8 y Figura 4.9). Esto nos da a entender que podemos aproximar los mismos patrones de crecimiento y decrecimiento de MP_{2.5}, NO y NO₂ que tendrían los dispositivos especializados del SINCA por una fracción del costo, y en algunos casos, lograr las mismas mediciones que esos dispositivos, lo que justificaría el uso masivo de estos sensores en múltiples lugares a la vez.

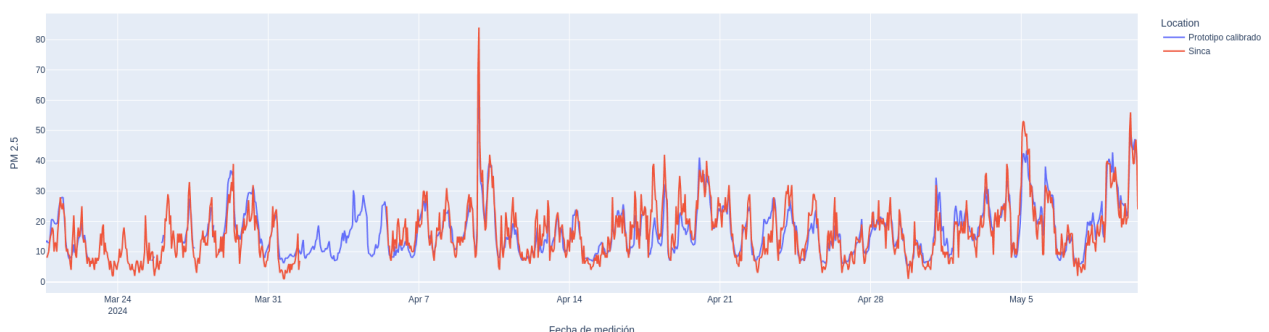


Figura 4.7: Calibración Sensor MP_{2.5}: Comparación entre mediciones obtenidas por el prototipo del Producto 5 y las mediciones reportadas por la estación del SINCA Parque O'Higgins. Entre mayor sea el traslape entre las curvas, mejor es la calibración.

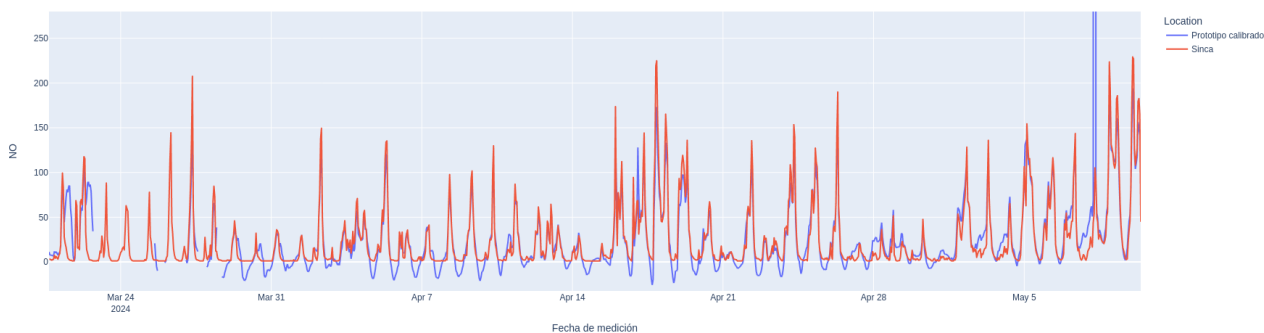


Figura 4.8: Calibración de Sensor NO: Comparación entre mediciones obtenidas por el prototipo del Producto 5 y las mediciones reportadas por la estación del SINCA Parque O'higgins. Entre mayor sea el traslape entre las curvas, mejor es la calibración.

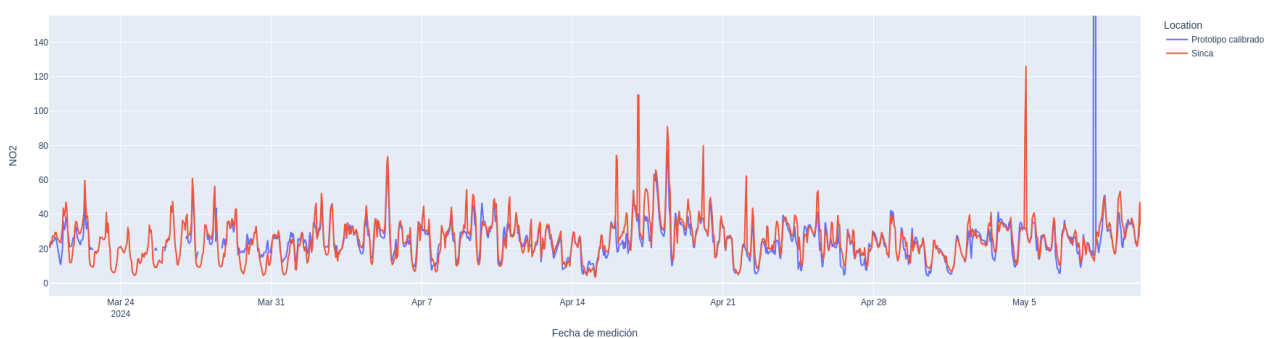


Figura 4.9: Calibración Sensor NO2: Comparación entre mediciones obtenidas por el prototipo del Producto 5 y las mediciones reportadas por la estación del SINCA Parque O'higgins. Entre mayor sea el traslape entre las curvas, mejor es la calibración.

5. Pilotaje

5.1. Lugar y Fechas de Despliegue

El MVP se desplego en una zona de alta afluencia escolar en la comuna de Ñuñoa, en donde el GORE-RM en conjunto con el Municipio de dicha comuna, buscan medir el impacto de medidas de cambio de flujo en calles (vía reversible y/o cierre acceso vehicular) de calles aledañas a establecimientos escolares durante las horas de ingreso y salida.

La zona propuesta en primera instancia corresponde a la Calle Contra maestre Micalvi (véase Figura 5.1), en donde se encuentra el Colegio San Agustín. En esta calle, entre el 01/08/2024 y el 30/08/2024, tuvo un piloto de bidireccionalidad a unidireccionalidad entre 06:30 a 08:30 hrs. Además, esta calle es aledaña a dos zonas de alta afluencia en la comuna: Calle Dublé Almeyda y Calle Ramon Cruz Montt.

El dispositivo fue desplegado en las siguientes semanas en las que se realizó el plan piloto, considerando tres periodos clave:

1. Vacaciones de invierno: Desde el 26/06/2024 hasta el 07/07/2024, inclusive.
2. Retorno a clases: Desde el 08/07/2024 hasta el 31/07/2024, inclusive.
3. Cambio de direccionalidad en la calle observada: Desde el 01/08/2024 hasta 25/08/2024, inclusive. Este efecto aplica solo desde las 06:30 hasta las 08:30 de Lunes a Viernes.

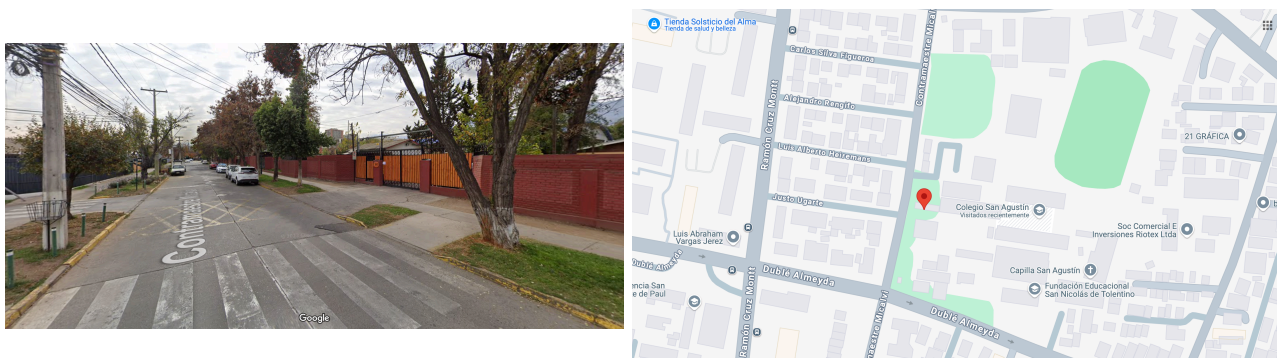


Figura 5.1: Lugar de Despliegue, Colegio San Agustín, Calle Contra maestre Micalvi, Ñuñoa, RM.

5.2. Instalación y Configuración de Dispositivo

El lugar de despliegue fue seleccionado con el objetivo es monitorear las variaciones de flujo de vehículos motorizados, peatones y ciclos en horarios equivalentes cuando el piloto de cambio de direccionalidad mencionado, no esté rigiendo y cuando sea efectivamente implementada. La instalación realizada (véase Figura 5.2, considera un lugar adecuado respecto requisitos mínimos de funcionamiento: altura, ángulo de cámara, y energización.

La Figura 5.3 muestra una captura de la Calle Contra maestre Micalvi observada desde la cámara conectada al prototipo. El prototipo se instaló a unos 2 metros de la calle y a unos 3 metros de altura. Entre menor sea la distancia hacia la zona de interés, mejor se podrá distinguir a los distintos elementos que nos interesa contar, en especial a las personas y las bicicletas que de por sí tienen menor tamaño que autos, camiones y buses. El ángulo elegido permite observar la calle en su totalidad y contiene una sola dirección, lo que favorece la detección de vehículos al disminuir la cantidad de cruces entre ellos. También logra capturar partes de las veredas y el cruce de peatones, lo que favorece a la identificación y conteo de peatones. En ésta figura también se puede observar que hay zonas irrelevantes, como los techos de las casas o la presencia de árboles. Para evitar que éstas zonas entorpezcan el proceso de conteo se definen regiones de interés denotadas por las líneas moradas de la captura, de tal manera que los elementos que nos interesan solo sean contados cuando interactúen con estas zonas.



Figura 5.2: Punto de instalación en el lugar de despliegue, Colegio San Agustín, Calle Contra-maestre Micalvi, Ñuñoa, RM.

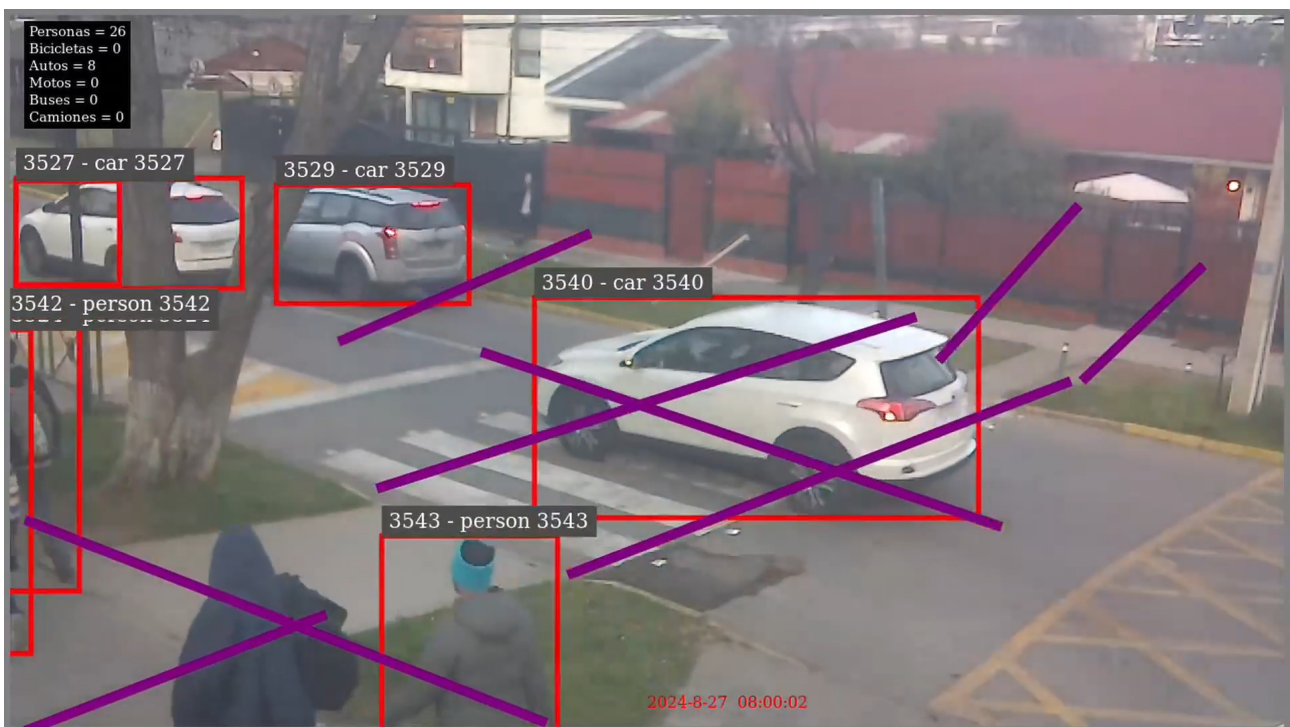


Figura 5.3: Captura en Despliegue en el Lugar de Despliegue. El prototipo ejecuta sobre lo capturado los algoritmos de detección (cajas rojas con etiqueta); identificación y seguimiento (ID única asociada a cada elemento dentro de las cajas rojas); y conteo de elementos. Cuando un elemento cruza una de las líneas moradas dibujadas en la imagen aumenta el contador de la etiqueta correspondiente (esquina superior izquierda) y se registra la ID del elemento para evitar volver a contarlo en el futuro.

6. Resultados Experimentales

En la presente sección, se analizan diversos resultados levantados a partir de los datos recolectados por el dispositivo. A modo de ejemplo, se muestran distintos tipos de gráficos que pueden ser solicitados, tanto como un análisis pre-eliminar de éstos datos recolectados.

Al finalizar el plan piloto se hace una recopilación de las mediciones y conteos durante todo el periodo, registrando en total 81733 entradas de los sensores que miden PM 2.5, NO y NO₂, y 82715 entradas en el contador de vehículos. Estas mediciones corresponden al promedio de las mediciones y a la suma de elementos observados por minuto durante el periodo de 8.5 semanas entre el 26/07/2024 a las 05:00 horas y el 25/08/2024 a las 23:59 horas. La diferencia en la cantidad de registros de ambos dispositivos se puede deber a fallas inesperadas en el funcionamiento, como por ejemplo, desconexión de internet, reinicio de los dispositivos, errores en el programa o errores de memoria. Cabe señalar, que el Módulo de Calidad del Aire tiene menos requerimientos energéticos que el Módulo de Detección y Conteo de Objetos Viales, resultando esencial en casos de cortes de energía, ya que la vuelta en funcionamiento del primero es mucho más rápida que la del segundo.

6.1. Contraste entre períodos claves

Para hacer una comparación entre los distintos períodos de muestreo se generaron gráficos que contrastan las distribuciones de automóviles, personas y bicicletas a lo largo de una semana dentro de cada uno de los tres periodos claves del plan piloto (véase Figura 6.1, Figura 6.2 y Figura 6.3). En particular, se observa que durante la semana de 01/07/2024 (líneas azules) éstos tres elementos viales presentan menor actividad entre los días Lunes y Jueves en comparación a las otras semanas, lo que resulta esperable considerando que esa semana corresponde al periodo de vacaciones de invierno escolares. Por otra parte, durante las semanas del 22/07/2024 (líneas rojas) y del 12/08/2024 (líneas verdes) se ve un flujo regular de personas y vehículos, especialmente entre las 06:00 hrs y las 09:00 hrs, horario en el que comienzan a llegar funcionarios, estudiantes y apoderados para el inicio de la jornada escolar, repitiéndose entre 14:00 y 18:00 hrs, cuando terminan éstas jornadas.

Respecto al cambio de direccionalidad en la calle Contramaestre Micalvi (véase Figura 6.1), se observa que el primer peak de concurrencia (entre 06:00 hrs. y 09:00 hrs.) se mantiene similar entre la semana del 22/07/2024 (líneas rojas) y la semana del 12/08/2024 (líneas verdes), entendiéndose que no disminuyó la densidad vehicular en esa calle y que las personas que circulan en automóviles lograron adaptarse al cambio de direccionalidad durante esas horas de la mañana. Considerar que la diferencia del jueves y viernes se debe al fin de semana largo del 14/08/2024.

El flujo de ciclistas (véase Figura 6.3) se mantiene constante durante los tres periodos observados, lo que puede sugerir que estas personas son, en mayor medida, peatones sin relación con el colegio dado que su número tampoco varió durante la semana de vacaciones de invierno, y que tampoco se logró observar un incremento de bicicletas como alternativa a los automóviles al restringir la dirección de la calle Contramaestre Micalvi. Este fenómeno es similar respecto a motocicletas (véase Figura 6.4).

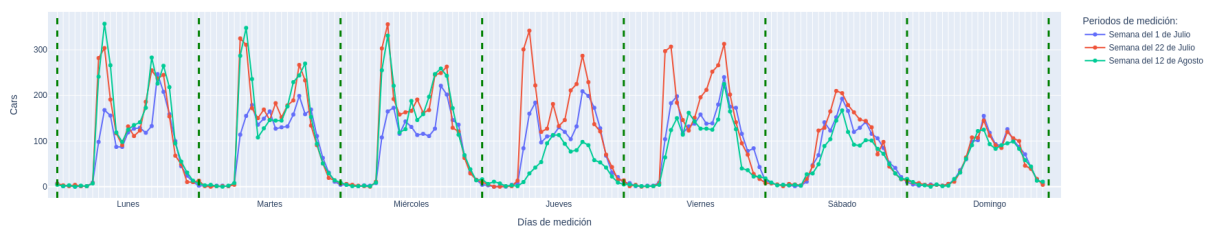


Figura 6.1: Distribución de automóviles a lo largo una semana para las semanas del 01/07/2024 (azul), 22/07/2024 (rojo) y 12/08/2024 (verde).

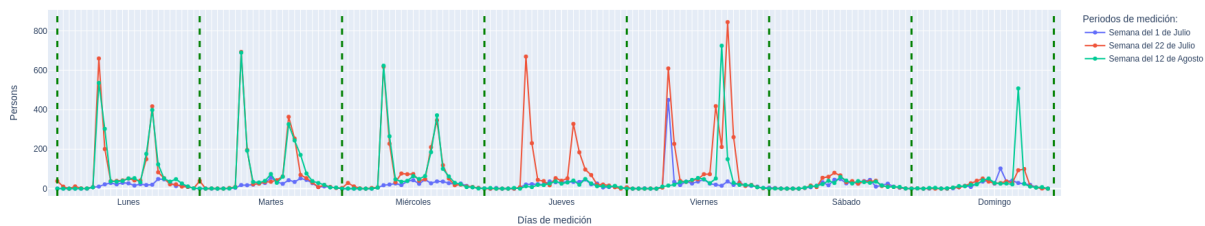


Figura 6.2: Distribución de personas a lo largo de una semana para las semanas del 01/07/2024 (azul), 22/07/2024 (rojo) y 12/08/2024 (verde).

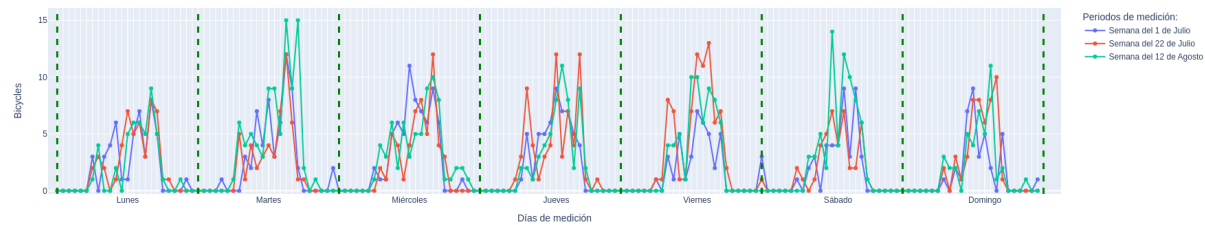


Figura 6.3: Distribución de bicicletas a lo largo de una semana para las semanas del 01/07/2024 (azul), 22/07/2024 (rojo) y 12/08/2024 (verde).

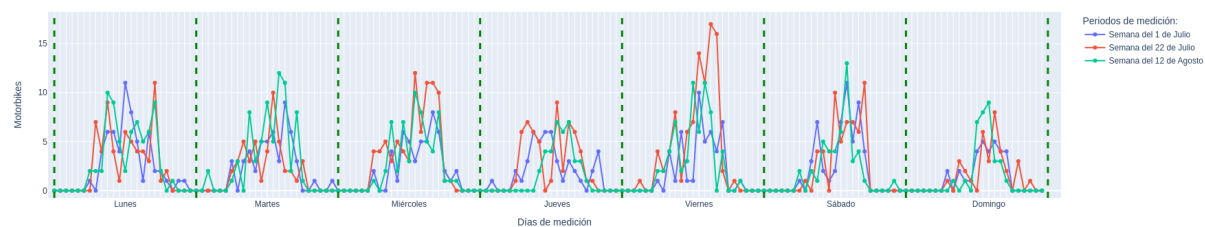


Figura 6.4: Distribución de motocicletas a lo largo de una semana para las semanas del 01/07/2024 (azul), 22/07/2024 (rojo) y 12/08/2024 (verde).

6.2. Contraste de contaminantes atmosféricos

De forma similar al contraste realizado a objetos viales, se puede comparar las mediciones de contaminantes atmosféricos y ver su evolución en las semanas escogidas con anterioridad. De esta forma, podemos ver que MP2.5, mantiene una tendencia estable, salvo el día Miércoles y Jueves de la semana del 12/08/2024 (líneas verdes), días que estuvieron precedidos por Alertas Ambientales decretadas por la autoridad metropolitana. De tal forma, el aumento de los contaminantes se debe al retorno de funcionamiento de entidades prohibidas durante dichos eventos. Este comportamiento de estabilidad también se observa en los NO_x (NO y NO_2), salvo la particularidad del descenso durante los días Jueves y Viernes de la semana del 12/08/2024 (líneas verdes), por el fin de semana largo correspondiente a aquella fecha.

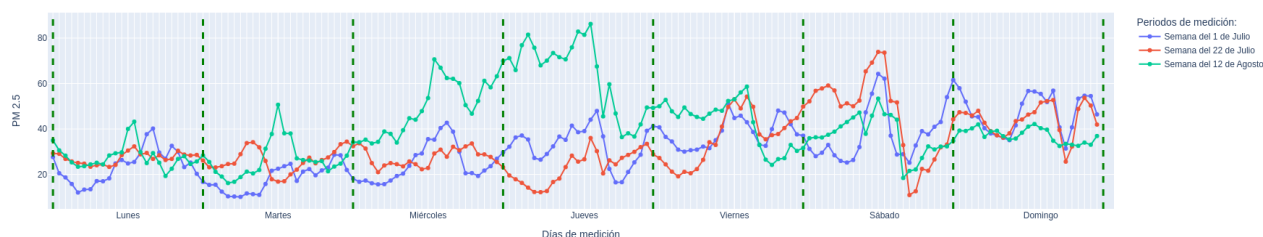


Figura 6.5: Distribución de MP2.5 a lo largo de una semana para las semanas del 01/07/2024 (azul), 22/07/2024 (rojo) y 12/08/2024 (verde).

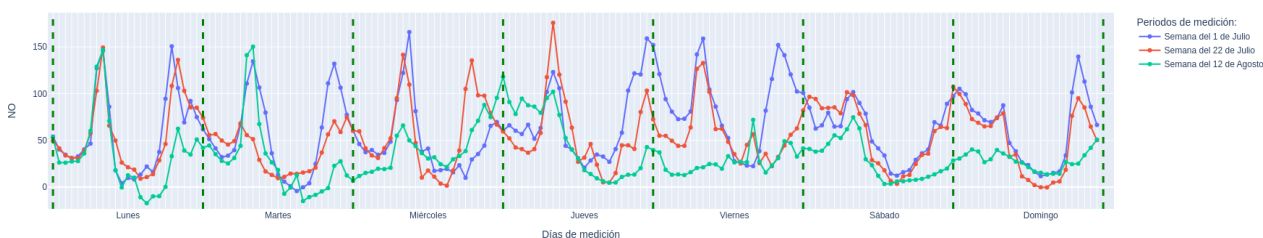


Figura 6.6: Distribución de NO a lo largo de una semana para las semanas del 01/07/2024 (azul), 22/07/2024 (rojo) y 12/08/2024 (verde).

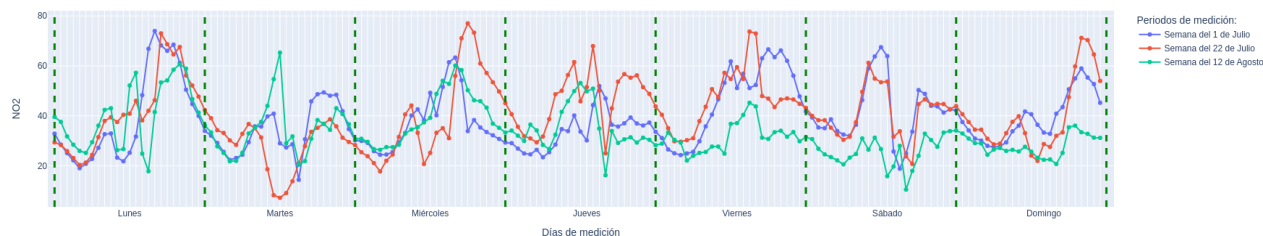


Figura 6.7: Distribución de NO_2 a lo largo de una semana para las semanas del 01/07/2024 (azul), 22/07/2024 (rojo) y 12/08/2024 (verde).

6.3. Contraste entre objetos viales y contaminantes atmosféricos

Uno de los elementos interesantes a analizar, es el contraste, durante el periodo de pilotaje, del flujo vehicular y los distintos contaminantes atmosféricos para verificar si tienen incidencia en los niveles de contaminación detectados en la zona. Al observar la Figura 6.8, se observa cierto grado de independencia entre las cantidades de automóviles y las mediciones de MP2.5, no pudiendo observar de forma evidente algún aporte de contaminación local a lo detectado. Además, este comportamiento es esperado, debido a que el MP2.5 es producido por múltiples fuentes, entre ellas las fábricas, la quema de madera, y en menor medida, los vehículos.

En el caso del NO_x (véase Figura 6.9 y Figura 6.10), se observa que en el período, las mayores mediciones fueron registradas entre el 26 de junio y el 10 de julio, lo que corresponde a gran parte del periodo de vacaciones, y a la vez al periodo que registró la menor cantidad de automóviles. Posterior al periodo de vacaciones, las mediciones se mantienen en rangos estables con muy poca variación, salvo por los días de lluvia, por lo que no se puede determinar relación directa entre el número local de automóviles y los niveles NO_x detectados, en el período y lugar de pilotaje. Esto se debe a que, si bien los NO_x se producen en mayor medida por procesos de combustión (uso de bencina), los datos locales (conteo de vehículos) no tienen un efecto en la medición local de los contaminantes, que corresponden a una visión global de la contaminación atmosférica en la hoya hidrográfica de la Región Metropolitana.

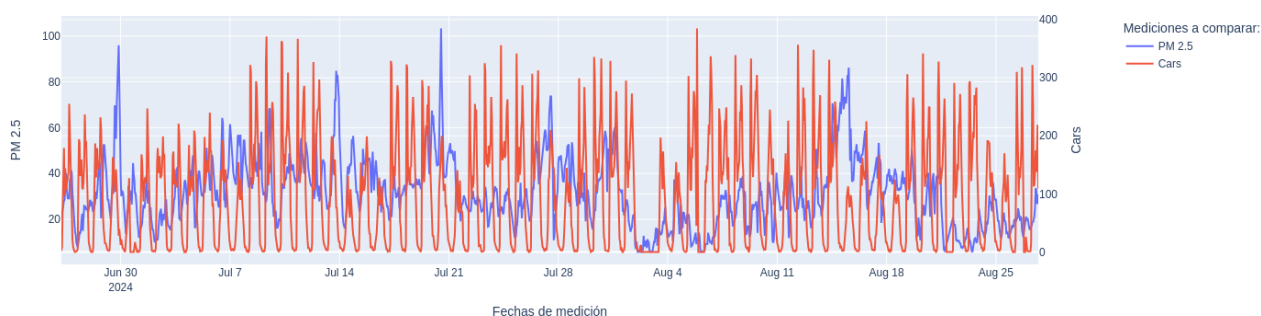


Figura 6.8: Contraste entre la cantidad de automóviles observados y las mediciones de MP2.5 registradas a lo largo del plan piloto.

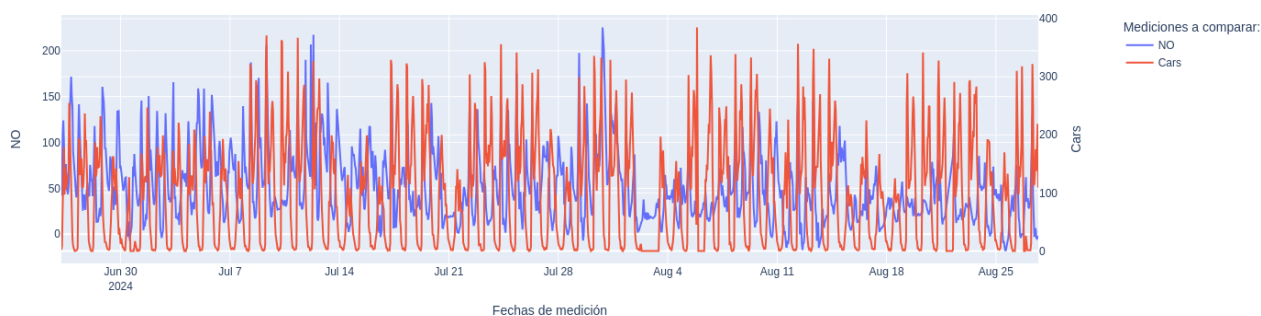


Figura 6.9: Contraste entre la cantidad de automóviles observados y las mediciones de NO registradas a lo largo del plan piloto.

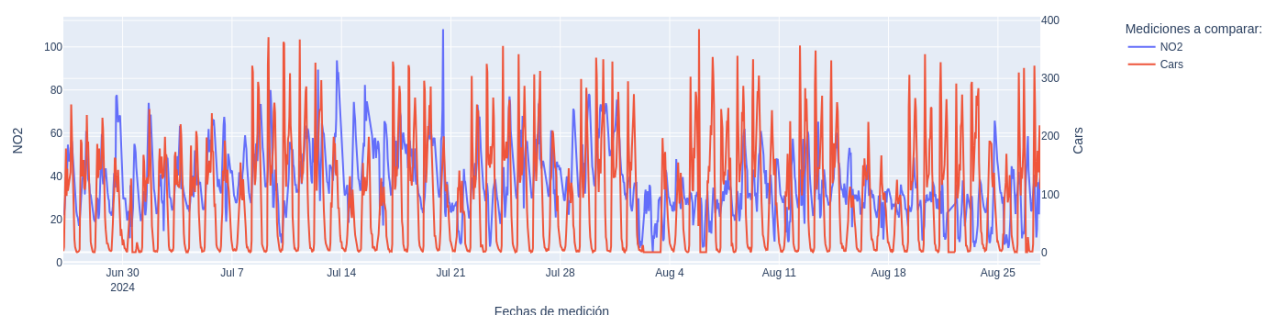


Figura 6.10: Contraste entre la cantidad de automóviles observados y las mediciones de NO2 registradas a lo largo del plan piloto.

6.4. Progresión de detección de objetos viales

Por último, tal como muestra la Figura 6.11, se observa la progresión de los diferentes objetos viales requeridos. Como se había anticipado, durante el periodo de vacaciones es cuando hay menor movimiento a nivel general, mientras que durante los periodos de clases se puede observar que el mayor flujo ocurre entre las 6 y las 9 AM, y entre las 2 PM y las 6 PM. Además, los mayores flujos de objetos corresponden a Peatones y Automóviles, siendo el resto de las categorías, mucho menor, en el sector del pilotaje.

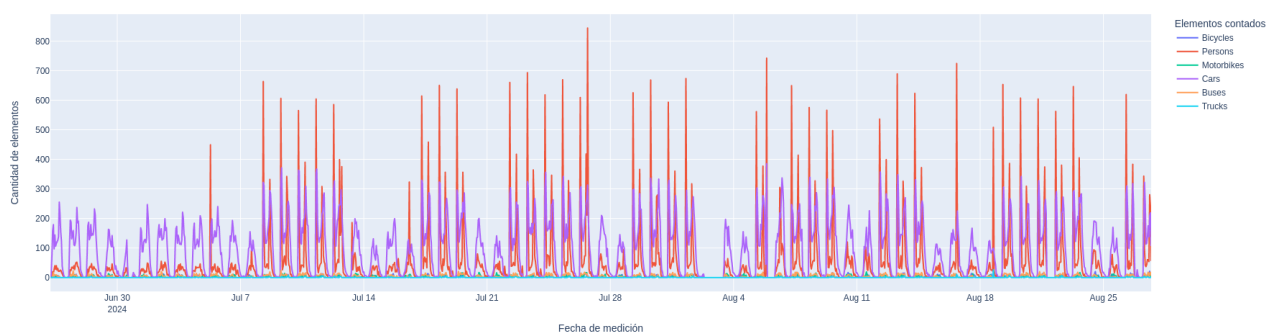


Figura 6.11: Progresión de Objetos Viales detectados durante el período de pilotaje.

7. Conclusiones y Resultados Esperados

A partir de los antecedentes recolectados, se construye un dispositivo inteligente, a partir de los antecedentes tecnológicos analizados, el cual permite cumplir con los objetivos del Producto 5 dentro del Proyecto GORE-PNUD. Este producto, considera un prototipo IoT, capaz de interconectar y disponibilizar distintas fuentes de información como eje principal de obtención de data sobre variables ambientales y de tránsito urbano, transmitiéndolas a un ambiente remoto para su posterior análisis y almacenamiento. La construcción y puesta en marcha de un piloto, permite cumplir los dos objetivos específicos iniciales del Producto 5, mientras que la experiencia adquirida en todo el proceso, y los ajustes del mismo, permiten reportar exitosamente los pasos para cumplir con el objetivo específico restante.

Los antecedentes y el pilotaje permiten trazar líneas de escalamiento del dispositivo IoT, con la finalidad de incluir nuevas funcionalidades o establecer plazos de instalación masiva. Dentro de estas posibilidades, nos encontramos con barreras tecnológicas que no permiten el desarrollo de ciertas funcionalidades como la clasificación por rango etario o por género, como con el incremento de los costos para poder tener persistencia de vídeo o clasificación vehículos no motorizados de tipo ciclos. La implementación de estas características excede los recursos del proyecto y se recomienda analizar detalladamente si estas funcionalidades son necesarias al momento de escalar el prototipo diseñado o es mejor establecer productos tradicionales separados para cumplir con ellas.

La fase de construcción, despliegue y pilotaje, ha servido para ajustar las expectativas y el funcionamiento del dispositivo construido y generar un conjunto de recomendaciones para su uso y reproducción, así como negociar e innovar con nuevas iniciativas que se adapten a la realidad del sector a aplicar (definida por región, tipo de transporte, calles, clima, etc).

En cuanto a los resultados generados, se cuenta con un producto capaz de monitorear y reportar en condiciones *near real-time* información ambiental y de transporte urbano, que se ajusta a las directrices establecidas en el Producto 4, y enviar, en ventanas asincrónicas de tiempo, información consolidada para nutrir el sistema diseñado por el Producto 3 del proyecto *Ciclovías Inteligentes y Transporte Sostenible en la RM*, dando por cumplido el principal objetivo del Producto 5 de forma exitosa.

8. Equipo de Trabajo

El equipo de trabajo que sostiene esta propuesta, está compuesto por las siguientes personas:

- Nicolás Gálvez Ramírez: Dr en Informatique (Univ. Angers), Dr. en Ing. Informática (UTFSM), Académico del Departamento de Electrónica, líder de la carrera de Ingeniería Civil Telemática en Campus Santiago San Joaquín (CSSJ), y parte del equipo académico del Programa de Data Analytics; en la Universidad Técnica Federico Santa María. Sus trabajos se concentran en la Integración Tecnológica, Optimización, Inteligencia Artificial y Ciencia de Datos, incluyendo:
 - La implementación de técnicas de Optimización y Machine Learning en el ámbito del Transporte (Díaz Urrea y cols., 2023) y la generación automática de código (Gálvez Ramírez y cols., 2020),
 - La generación de solvers automáticos para problemas de optimización con diferentes lógicas de primer orden, y
 - La detección de flujos urbanos de desechos a través de reconocimiento de imágenes.
- Sebastian Diez: Dr. en Ingeniería, especialista en Contaminación Atmosférica de la Universidad Tecnológica Nacional de Buenos Aires, Argentina. Académico de la Universidad del Desarrollo en Chile, y Investigador Científico Asociado a la Universidad de York en Reino Unido. Experto en mediciones de calidad de aire y toma de decisiones para políticas públicas ambientales, donde destaca:
 - Uso de sensores de bajo costo en zonas urbanas, y el manejo de los errores que las mediciones ambientales pueden generar, respecto a mediciones normativas.
 - Análisis de las transformaciones urbanas en latino-américa respecto a las actividades contaminantes cotidianas que afectan a la población.
- Ignacio Jorquera Navarro: Ing. Civil Informático y alumno regular del Magíster en Ciencias de la Ingeniería Informática. Actualmente se desempeña como Ayudante de Investigación y posee vasta experiencia en Optimización y Aprendizaje Estadístico (Supervisado, No Supervisado y Reforzado) y el uso de herramientas como Z3, PyTorch, YOLO, TensorFlow, Keras, etc. Sus trabajos recientes incluyen:
 - Investigación y desarrollo de la plataforma web “CoTranslate” (Carvallo y cols., 2023a) enfocada en facilitar tareas de etiquetado y traducción de oraciones entre español, inglés, mapudungún y rapanui (Carvallo y cols., 2023b) en base al paradigma “No Language Left Behind” (Costa-jussà y cols., 2022), con la finalidad de generar un ciclo de aprendizaje reforzado humano-máquina para mejorar la calidad de las traducciones de lenguas sub-representadas mediante la inteligencia artificial.
 - Investigación y desarrollo de un probador de ropa virtual haciendo uso de inteligencia artificial para resolver problemas enfocados en el procesamiento de imágenes en tiempo real mediante una cámara, como reconocimiento facial, estimación de pose y forma corporal de una persona, generación de mallas 3D, entre otros.
- Matías Soto Sepúlveda: Alumno de primera Generación de Ingeniería Civil Telemática en Campus San Joaquín, siendo uno de los mejores alumnos de la carrera entre todas las generaciones. Se ha caracterizado por tener gran habilidad en la integración entre capas físicas y de aplicación en la creación de dispositivos inteligentes (IoT), incluyendo técnicas de inteligencia artificial. Actualmente se desempeña como ayudante de las áreas de redes de computadores y sensores, y de sistemas de computación. Sus trabajos recientes incluyen:
 - Desarrollo de software de integración y visualización de datos de sensores de temperatura y aceleración para el monitoreo de estaciones de correa transportadora para el área minera.
 - Diseño de arquitectura de solución en la nube para ingesta y visualización de datos del ministerio de cultura de Chile.
 - Desarrollo de solución informática para conectar plataforma web con sistema de recomendación basado en inteligencia artificial orientado al uso médico.

Referencias

- Ali, Z. A., Narejo, S., Pandey, B., Esenarro vargas, D., Rodriguez, C., y Anjum, M. R. (2021). Weapon detection using yolo v3 for smart surveillance system. *Mathematical Problems in Engineering*, 2021, 9975700. Descargado de <https://doi.org/10.1155/2021/9975700> doi: 10.1155/2021/9975700
- Ammar, A., Koubaa, A., Ahmed, M., y Saad, A. (2019). Aerial images processing for car detection using convolutional neural networks: Comparison between faster R-CNN and yolov3. *CoRR*, abs/1910.07234. Descargado de <http://arxiv.org/abs/1910.07234>
- Arduino. (2023). *Arduino*. <https://www.arduino.cc/>. (Last Visited: 2023-10-18)
- Bochkovskiy, A., Wang, C., y Liao, H. M. (2020). Yolov4: Optimal speed and accuracy of object detection. *CoRR*, abs/2004.10934. Descargado de <https://arxiv.org/abs/2004.10934>
- Cao, Z., Liao, T., Song, W., Chen, Z., y Li, C. (2021). Detecting the shuttlecock for a badminton robot: A yolo based approach. *Expert Systems with Applications*, 164, 113833. Descargado de <https://www.sciencedirect.com/science/article/pii/S0957417420306436> doi: <https://doi.org/10.1016/j.eswa.2020.113833>
- Carvalho, A., Jorquera, I., y Aspillaga, C. (2023a). *Cotranslate*. <https://github.com/OpenCENIA/CoTranslate>. (Last Accessed: Oct 14, 2023)
- Carvalho, A., Jorquera, I., y Aspillaga, C. (2023b, 07). Cotranslate: A web-based tool for crowdsourcing high-quality sentence pair corpora. *SoftwareX*, 23, 101508. doi: 10.1016/j.softx.2023.101508
- Corporation, N. (2023a). *Cuda cores*. <https://developer.nvidia.com/cuda-zone>. (Last Visited: 2023-10-18)
- Corporation, N. (2023b). *Nvidia jetson platform*. <https://developer.nvidia.com/embedded-computing>. (Last Visited: 2023-10-18)
- Costa-jussà, M. R., Cross, J., Çelebi, O., y cols. (2022). *No language left behind: Scaling human-centered machine translation*.
- Cáquilpan, V., G., A., S., S., R., K., G., y Sierralta Jorquera, C. (2019, 10). Advantages and challenges of the implementation of a low-cost particulate matter monitoring system as a decision-making tool. *Environmental Monitoring and Assessment*, 191. doi: 10.1007/s10661-019-7875-4
- Díaz Urra, R., Gálvez Ramírez, N., y Castro, C. (2023). A framework for the generation of complex scenario instances in the urban transit routing problem. *Ann. Math. Artif. Intell.*, 91(2-3), 153–175. Descargado de <https://doi.org/10.1007/s10472-022-09797-z> doi: 10.1007/s10472-022-09797-z
- Feenstra, B., Papapostolou, V., Hasheminassab, S., Zhang, H., Boghossian, B. D., Cocker, D., y Polidori, A. (2019). Performance evaluation of twelve low-cost pm2.5 sensors at an ambient air monitoring site. *Atmospheric Environment*, 216, 116946. Descargado de <https://www.sciencedirect.com/science/article/pii/S1352231019305850> doi: <https://doi.org/10.1016/j.atmosenv.2019.116946>
- Ferrante, G. S., Vasconcelos Nakamura, L. H., Sampaio, S., Filho, G. P. R., y Meneguette, R. I. (2024, 16 de Jan). Evaluating yolo architectures for detecting road killed endangered brazilian animals. *Scientific Reports*, 14(1), 1353. Descargado de <https://doi.org/10.1038/s41598-024-52054-y> doi: 10.1038/s41598-024-52054-y
- Foundation, R. P. (2023a). *Raspberry pi*. <https://www.raspberrypi.org/>. (Last Visited: 2023-10-18)
- Foundation, R. P. (2023b). *Raspberry pi pico*. <https://www.raspberrypi.com/products/raspberry-pi-pico/>. (Last Visited: 2023-10-18)

- Gálvez Ramírez, N., Monfroy, É., Saubion, F., y Castro, C. (2020). Improving complex SMT strategies with learning. *Int. Trans. Oper. Res.*, 27(2), 1162–1188. Descargado de <https://doi.org/10.1111/itor.12650> doi: 10.1111/itor.12650
- Girshick, R. (2015). *Fast r-cnn*.
- Girshick, R., Donahue, J., Darrell, T., y Malik, J. (2014). *Rich feature hierarchies for accurate object detection and semantic segmentation*.
- Golcarenenji, G., Martinez-Alpiste, I., Wang, Q., y Alcaraz-Calero, J. M. (2023). Robust real-time traffic light detector on small-form platform for autonomous vehicles. *Journal of Intelligent Transportation Systems*. Descargado de <https://www.sciencedirect.com/science/article/pii/S1547245023000439> doi: <https://doi.org/10.1080/15472450.2023.2205018>
- Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., ... Adam, H. (2017). *Mobilenets: Efficient convolutional neural networks for mobile vision applications*.
- IEC. (2013). *Degrees of protection provided by enclosures (ip code)*.
- Jocher, G. (2020). *Ultralytics yolov5*. Descargado de <https://github.com/ultralytics/yolov5> doi: 10.5281/zenodo.3908559
- Jocher, G., Chaurasia, A., y Qiu, J. (2023, enero). *Ultralytics YOLO*. Descargado de <https://github.com/ultralytics/ultralytics>
- Kang, Y., Aye, L., Ngo, T. D., y Zhou, J. (2022). Performance evaluation of low-cost air quality sensors: A review. *Science of The Total Environment*, 818, 151769. Descargado de <https://www.sciencedirect.com/science/article/pii/S0048969721068455> doi: <https://doi.org/10.1016/j.scitotenv.2021.151769>
- Karagulian, F., Barbieri, M., Kotsev, A., Spinelle, L., Gerboles, M., Lagler, F., ... Borowiak, A. (2019). Review of the performance of low-cost sensors for air quality monitoring. *Atmosphere*, 10(9). Descargado de <https://www.mdpi.com/2073-4433/10/9/506> doi: 10.3390/atmos10090506
- Kuprashevich, M., y Tolstykh, I. (2023). *Mivolo: Multi-input transformer for age and gender estimation*.
- Li, C., Li, L., Jiang, H., Weng, K., Geng, Y., Li, L., ... Wei, X. (2022). *Yolov6: A single-stage object detection framework for industrial applications*.
- Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., y Berg, A. C. (2016). Ssd: Single shot multibox detector. En *Lecture notes in computer science* (p. 21–37). Springer International Publishing. Descargado de http://dx.doi.org/10.1007/978-3-319-46448-0_2 doi: 10.1007/978-3-319-46448-0_2
- Lucas, B., y Kanade, T. (1981, 04). An iterative image registration technique with an application to stereo vision (ijcai). En (Vol. 81).
- Morawska, L., Thai, P. K., Liu, X., Asumadu-Sakyi, A., Ayoko, G., Bartonova, A., ... Williams, R. (2018). Applications of low-cost sensing technologies for air quality monitoring and exposure assessment: How far have they gone? *Environment International*, 116, 286–299. Descargado de <https://www.sciencedirect.com/science/article/pii/S0160412018302460> doi: <https://doi.org/10.1016/j.envint.2018.04.018>
- Naftali, M. G., Sulistyawan, J. S., y Julian, K. (2022). *Comparison of object detection algorithms for street-level objects*.
- Ning, C., Zhou, H., Song, Y., y Tang, J. (2017). Inception single shot multibox detector for object detection. , 549–554. doi: 10.1109/ICMEW.2017.8026312
- Osses, M., Loo Oyarzún, M.-I., y Diez, S. (2023). *Producto 4, plan de monitoreo de contaminantes enfocado en la red de ciclovías, informe de avance*.
- Redmon, J., Divvala, S., Girshick, R., y Farhadi, A. (2016). You only look once: Unified, real-time object detection. En *2016 IEEE conference on computer vision and pattern recognition (cvpr)* (p. 779–788). doi: 10.1109/CVPR.2016.91

- Ren, S., He, K., Girshick, R., y Sun, J. (2016). *Faster r-cnn: Towards real-time object detection with region proposal networks*.
- Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., y Chen, L.-C. (2019). *Mobilenetv2: Inverted residuals and linear bottlenecks*.
- Shehzadi, T., Hashmi, K. A., Stricker, D., y Afzal, M. Z. (2023). *Object detection with transformers: A review*.
- Simonyan, K., y Zisserman, A. (2015). *Very deep convolutional networks for large-scale image recognition*.
- Systems, E. (2023). *Esp32*. <https://www.espressif.com/en/products/hardware/esp32/overview>. (Last Visited: 2023-10-18)
- Tomasi, C., y Kanade, T. (1991). *Detection and tracking of point features*.
- Wang, C.-Y., Bochkovskiy, A., y Liao, H.-Y. M. (2022). *Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors*.
- Wojke, N., Bewley, A., y Paulus, D. (2017). *Simple online and realtime tracking with a deep association metric*.
- Zhang, G., Ge, Y., Dong, Z., Wang, H., Zheng, Y., y Chen, S. (2021). Deep high-resolution representation learning for cross-resolution person re-identification. *IEEE Transactions on Image Processing*, 30, 8913–8925. Descargado de <http://dx.doi.org/10.1109/TIP.2021.3120054> doi: 10.1109/tip.2021.3120054
- Zhang, Y., Sun, P., Jiang, Y., Yu, D., Yuan, Z., Luo, P., ... Wang, X. (2021). Bytetrack: Multi-object tracking by associating every detection box. *CoRR*, abs/2110.06864. Descargado de <https://arxiv.org/abs/2110.06864>
- Zhou, K., Yang, Y., Cavallaro, A., y Xiang, T. (2019). *Omni-scale feature learning for person re-identification*.