



## Desarrollo de un proyecto con Inteligencia Artificial Nivel 3

Este template ha sido diseñado como una guía detallada para el desarrollo de proyectos de inteligencia artificial (IA), siendo su objetivo proporcionar una estructura clara y completa que permita a las estudiantes seguir un proceso ordenado y eficiente en el avances de sus proyectos, desde la definición del problema hasta la implementación y evaluación del modelo de inteligencia artificial.

El template está dividido en secciones que abarcan cada una de las etapas clave en el ciclo de desarrollo de un proyecto de IA, incluyendo la definición y contextualización del problema, la recolección y exploración de datos, el modelado y entrenamiento del modelo, la evaluación y disponibilización del mismo como API, así como la planificación, gestión y documentación del proyecto.

Cada sección contiene una descripción detallada de los elementos que se deben considerar en cada etapa, junto con espacios para que completen con la información específica de su proyecto. Además, se incluyen sugerencias y puntos de reflexión que pueden ayudar a mejorar el desarrollo y resultado final.

El objetivo de este documento es facilitar el proceso asegurando que se cubran todos los aspectos necesarios para lograr resultados exitosos y de calidad. Es importante que sigan este template de manera rigurosa y completa, adaptándolo a las particularidades de su proyecto y utilizando la información proporcionada como guía para tomar decisiones informadas en cada etapa del proceso.

Sobre este mismo documento compartido se contempla el versionado para visualizar la evolución a lo largo del cursado como así también la revisión mediante comentarios y sugerencias de cambios que permitan la corrección de forma dinámica entre estudiantes y docentes.

# PROGRAMA M1000IA

Sé parte de la Próxima Generación  
Innovadora de la Fuerza de Trabajo Futura



## Índice

Índice	2
Información del Proyecto	3
Definición del Problema	4
Planificación del Proyecto	5
Gestión de Proyecto	6
Recolección de Datos	7
Ética y Privacidad	8
Exploración de Datos (EDA)	9
Modelado	10
Entrenamiento del Modelo	11
Evaluación del Modelo	12
Disponibilización del Modelo como API	13
Implementación	14
Evaluación Final del Proyecto	15
Próximos Pasos y oportunidades de mejora	16
Mantenimiento y actualización	17
Seguimiento Post-Implementación	18
Reflexiones	19

# PROGRAMA M1000IA

Sé parte de la Próxima Generación  
Innovadora de la Fuerza de Trabajo Futura



## Información del Proyecto

- Nombre del Proyecto: Detección de vehículos de emergencia.
- Integrantes:
  - Kary Francia, analista de datos y planificación/científica de datos junior.  
kfrancia@ceibal.edu.uy
- Representante: Kary Francia.
- Organización: No aplica.
- Posibles proyectos:
  - Detección/diagnóstico por imágenes
  - Predicción de demanda de productos
  - Predicción poblacional
  - **Detección de vehículos (proyecto seleccionado)**
  - Predicción de demanda de repuestos para soporte de dispositivos electrónicos usando series temporales

# PROGRAMA M1000IA

Sé parte de la Próxima Generación  
Innovadora de la Fuerza de Trabajo Futura



## Definición del Problema

- Descripción del problema:

El problema consiste en desarrollar un sistema de conteo de vehículos basado en técnicas de machine learning e inteligencia artificial que pueda identificar y contar vehículos de manera precisa y automatizada a partir de imágenes de videos capturados por cámaras de vigilancia de tráfico u otros dispositivos de captura de imágenes.

Este sistema debe ser capaz de manejar una variedad de condiciones de iluminación, climáticas y tipos de vehículos ('Ambulancia', 'Bus', 'Auto', 'Moto' y 'Camión'), y debe ser escalable para su implementación en diferentes ubicaciones y entornos. Además, debe cumplir con restricciones de tiempo real para su aplicación en sistemas de gestión de tráfico.

La precisión en el conteo de vehículos es crucial para la toma de decisiones informadas en la gestión del tráfico, la planificación urbana y la infraestructura de transporte. Por lo tanto, el desarrollo de un sistema automatizado y preciso para la detección y conteo de vehículos es fundamental para mejorar la eficiencia del transporte urbano y contribuir a la seguridad vial en áreas urbanas congestionadas.

- Contexto del problema:

En entornos urbanos y de tránsito, contar vehículos es crucial y relevante para la gestión del tráfico, la planificación urbana, y la toma de decisiones en infraestructuras de transporte.

Tradicionalmente, este conteo se ha realizado manualmente o mediante métodos mecánicos costosos y poco precisos. Sin embargo, con el avance de la tecnología, el uso de técnicas de visión por computadora y machine learning han surgido como una alternativa prometedora para automatizar este proceso de conteo.

El desarrollo de un sistema que permita determinar con precisión la cantidad de vehículos que transitan por una arteria específica, podría tener implicaciones significativas. Por ejemplo, este sistema podría ayudar a analizar y prever demoras en la circulación y evaluar la peligrosidad de intersecciones específicas, dependiendo de la ubicación de las cámaras. Además, serviría para gestionar el tráfico mediante la modificación de la duración de los semáforos optimizando el flujo vehicular y evitando congestionamientos. Este sistema también podría facilitar la movilidad de vehículos de emergencia, como ambulancias, mejorando la eficiencia y la respuesta en situaciones críticas.

# PROGRAMA M1000IA

Sé parte de la Próxima Generación  
Innovadora de la Fuerza de Trabajo Futura



- Impacto del problema:

La gestión actual del sistema de tránsito tiene un impacto directo en la cantidad y gravedad de accidentes que suceden, pero además impacta en la calidad de vida de las personas que viven en ciudades de alta densidad poblacional, en la contaminación producida por los automóviles y transporte urbano y en los tiempos de demora del personal de asistencia.

Una solución efectiva a este problema puede facilitar la implementación de sistemas de transporte inteligente y contribuir a la reducción de emisiones contaminantes al optimizar el flujo del tráfico.

- Alcance del problema:

- El proyecto se centra en el desarrollo de un sistema avanzado que utilice técnicas de machine learning para detectar y contar vehículos en un área específica. Esto incluye la implementación de algoritmos de detección de vehículos a partir de imágenes de videos capturados por cámaras de vigilancia de tráfico u otros dispositivos de captura de imágenes.
- El sistema debe ser capaz de detectar los tipos de vehículos con la máxima precisión posible, incluyendo 'Ambulancia', 'Bus', 'Auto', 'Moto' y 'Camión'.
- El sistema debe ser capaz de manejar una variedad de condiciones ambientales, como diferentes niveles de iluminación y condiciones climáticas. Se busca optimizar el rendimiento del sistema bajo diferentes condiciones para garantizar resultados precisos en diversas situaciones.
- El sistema debe ser escalable para su implementación en diferentes ubicaciones y entornos urbanos. Además, debe cumplir con restricciones de tiempo real para su aplicación en sistemas de gestión de tráfico, lo que garantiza una respuesta rápida y eficiente.
- Se lleva a cabo una evaluación exhaustiva del sistema desarrollado para medir su precisión y eficacia en la detección y conteo de vehículos.

# PROGRAMA M1000IA

Sé parte de la Próxima Generación  
Innovadora de la Fuerza de Trabajo Futura



- **Objetivos del problema:**

Desarrollar un modelo de inteligencia artificial utilizando un modelo preentrenado con YOLOv8 capaz de identificar con precisión la presencia de vehículos en imágenes, que pasan por una determinada ubicación durante un período de tiempo; a fin de poder implementar medidas que mejoren la eficiencia en el tránsito de vehículos de emergencia (ambulancias).

- **Requerimientos y restricciones:**

**Requerimientos:**

- El sistema debe ser capaz de detectar vehículos con una precisión alta para garantizar la fiabilidad de los datos obtenidos.
- El sistema debe ser capaz de distinguir entre diferentes categorías de vehículos, como ambulancias, buses, autos, motos y camiones.
- El sistema debe ser robusto y capaz de manejar una variedad de condiciones ambientales, incluyendo diferentes niveles de iluminación y condiciones climáticas adversas.
- El sistema debe ser escalable para su implementación en diferentes ubicaciones y entornos urbanos, permitiendo su despliegue en áreas de alto tráfico.
- El sistema debe cumplir con restricciones de tiempo real para su aplicación en sistemas de gestión de tráfico, proporcionando resultados rápidos y actualizados.
- El sistema debe ser capaz de adaptarse a cambios en el flujo de tráfico y en las condiciones ambientales sin comprometer su rendimiento.

**Restricciones:**

- El sistema debe operar dentro de los límites de los recursos computacionales disponibles, incluyendo capacidad de procesamiento y almacenamiento.
- El desarrollo y despliegue del sistema deben mantenerse dentro de un presupuesto establecido para garantizar su viabilidad económica.
- El sistema depende de la disponibilidad de datos de imágenes de videos capturados por cámaras de vigilancia de tráfico u otros dispositivos de captura de imágenes.
- El sistema debe cumplir con las regulaciones de privacidad y protección de datos aplicables, garantizando el anonimato de los usuarios de la vía pública.
- El sistema debe ser compatible con sistemas existentes de gestión de tráfico y otros sistemas de infraestructura de transporte.



# PROGRAMA M1000IA

Sé parte de la Próxima Generación  
Innovadora de la Fuerza de Trabajo Futura



- Beneficios esperados:

- Reducción en los tiempos de viaje y mejora en la seguridad vial (reducción de accidentes): Una gestión del tráfico más eficiente y una planificación urbana optimizada pueden contribuir a mejorar la seguridad vial al reducir el riesgo de accidentes y mejorar la fluidez del tráfico.
- Respuesta de emergencia más rápida y reducción de embotellamientos: Al contar con datos precisos y actualizados sobre el flujo de vehículos, las autoridades de tráfico podrán tomar decisiones informadas para mejorar la gestión del tráfico, reducir las congestiones y optimizar los tiempos de viaje.
- Optimización de la planificación urbana: La información obtenida puede utilizarse para optimizar la planificación urbana, incluyendo la identificación de áreas que requieren mejoras en la infraestructura vial y la planificación de rutas de transporte público.
- Toma de decisiones en infraestructuras de transporte: Los datos precisos sobre el flujo vehicular pueden utilizarse para la toma de decisiones relacionadas con la construcción y mantenimiento de infraestructuras de transporte, como carreteras, puentes y túneles.
- Reducción de costos y aumento de eficiencia: Al automatizar el proceso se reducirán los costos asociados con el empleo de métodos tradicionales de conteo manual o mecánico. Además, se mejorará la eficiencia al obtener datos de manera más rápida y precisa.
- Facilitación de la implementación de sistemas de transporte inteligente: Los datos obtenidos pueden utilizarse para implementar sistemas de transporte inteligente, como semáforos adaptativos y sistemas de gestión de peajes, que mejorarán la experiencia de los usuarios de la vía pública.

# PROGRAMA M1000IA

Sé parte de la Próxima Generación  
Innovadora de la Fuerza de Trabajo Futura



## Planificación del Proyecto

- **Cronograma:**

### **Definición del problema, planificación del proyecto y gestión del proyecto (1 semana):**

- Establecer objetivos y alcance del proyecto.
- Identificar requerimientos y restricciones.
- Definir metodología de trabajo y herramientas a utilizar.

### **Recopilación de datos, ética y privacidad (1 semana):**

- Identificar fuentes de datos de imágenes o vídeos de tráfico.
- Recopilar y preparar conjuntos de datos etiquetados para entrenamiento del modelo.

### **Exploración de Datos (1 semana)**

### **Preprocesamiento de datos y modelado (2 semanas):**

- Investigar y seleccionar el algoritmo para detección de objetos.
- Entrenar el modelo utilizando los conjuntos de datos preparados.
- Optimizar y ajustar hiperparámetros del modelo para mejorar la precisión y eficiencia.

### **Entrenamiento del modelo (1 semana):**

- Ampliar el modelo para distinguir entre diferentes categorías de vehículos.
- Entrenar el modelo para clasificar vehículos en categorías como automóviles, camiones, bicicletas y motocicletas.

### **Evaluación del modelo (2 semanas):**

- Evaluar el desempeño del modelo utilizando conjuntos de datos de prueba para validar la precisión y robustez del modelo.

### **Disponibilización del modelo (1 semana):**

- Realizar ajustes finales en el modelo basados en los resultados de las pruebas y retroalimentación recibida.

### **Exposición final del proyecto en encuentro sincrónico (1 semana):**



# PROGRAMA M1000IA

Sé parte de la Próxima Generación  
Innovadora de la Fuerza de Trabajo Futura



- Preparar presentación para compartir los resultados del proyecto.

## **Cierre de documento de proyecto con correcciones y completado de secciones faltantes (1 semana):**

- Finalizar la documentación y realizar la entrega final del proyecto.

- **Recursos necesarios:**

Hardware: Laptops o computadoras para el procesamiento, entrenamiento y ejecución del modelo.

Software: Lenguaje Python con acceso a sus bibliotecas para el desarrollo y entrenamiento del modelo; entornos de desarrollo integrado (IDE) como Jupyter Notebook o Google Colab para la escritura y ejecución de código; software de gestión de proyectos como Jira o Trello para coordinar las tareas y el progreso del equipo.

Datos: Conjuntos de datos etiquetados de imágenes o vídeos de tráfico para el entrenamiento y validación del modelo; y datos de prueba que simulan las condiciones reales de tráfico para validar el rendimiento del sistema.

Infraestructura de TI: Servidores o servicios en la nube para el almacenamiento y procesamiento de grandes volúmenes de datos; software de control de versiones como Git para el seguimiento de cambios en el código y la colaboración en equipo.

# PROGRAMA M1000IA

Sé parte de la Próxima Generación  
Innovadora de la Fuerza de Trabajo Futura



## Gestión de Proyecto

- Responsabilidades del equipo: [Describir las responsabilidades asignadas a cada miembro del equipo]

Kary Francia - Analista de Datos y Planificación, Científica de Datos Jr: Encabezará el proceso de análisis de datos, modelado, y disponibilización del modelo.

- Comunicación y coordinación:

Se utilizarán reuniones semanales virtuales para sincronizar avances y resolver impedimentos. Para la comunicación diaria y coordinación de tareas, se emplea WhatsApp, permitiendo la creación de canales específicos por áreas de responsabilidad. Trello facilitará la gestión de tareas y el seguimiento del progreso en tiempo real, asignando tareas específicas a cada miembro y estableciendo fechas límite.

- Seguimiento y control:

El seguimiento del progreso se realizará a través de dashboard en Trello (O si prefieren un board en git), actualizado en tiempo real, permitiendo una vista general del estado del proyecto. Se establecerán indicadores de rendimiento específicos para cada fase del proyecto, revisados durante las reuniones semanales. Se documentará y reportará los resultados de las pruebas y ajustes necesarios.

- Riesgos Identificados:

- Retrasos en la recopilación de conjuntos de datos etiquetados.
- Sesgo en los datos de entrenamiento.
- Dificultades en la selección y optimización de algoritmos.
- Problemas de compatibilidad o rendimiento de hardware/software.
- Incumplimiento de regulaciones de privacidad y ética.
- Interpretabilidad y explicabilidad del modelo.
- Cobertura de pruebas insuficiente.
- Retrasos debidos a ciclos de prueba y corrección prolongados.

- Plan de mitigación de riesgos:

# PROGRAMA M1000IA

Sé parte de la Próxima Generación  
Innovadora de la Fuerza de Trabajo Futura



Se realizarán pruebas de compatibilidad de hardware/software al inicio del proyecto para mitigar problemas futuros. El equipo de desarrollo se asegurará de que el entorno de trabajo sea adecuado.

Se implementará un protocolo estricto de ética y privacidad desde el comienzo del proyecto para garantizar el cumplimiento de todas las regulaciones pertinentes.

Para fomentar la adopción del modelo, se desarrollará una estrategia de comunicación para resaltar los beneficios y aplicaciones del modelo.

Para la interpretabilidad y aplicabilidad del modelo, se integrarán herramientas y técnicas de explicabilidad de IA desde las primeras fases del desarrollo. Esto facilitará una mayor transparencia y confianza en las decisiones tomadas por el modelo.

Dado el tiempo y recursos limitados, se deberán priorizar las pruebas basándose en el riesgo y la criticidad de las diferentes partes del sistema. Esto asegura que los esfuerzos de prueba se concentren en las áreas más propensas a fallos o aquellas que tendrían el mayor impacto negativo si fallaran.

# PROGRAMA M1000IA

Sé parte de la Próxima Generación  
Innovadora de la Fuerza de Trabajo Futura



## Recolección de Datos

- **Fuente de los datos:**

Los datos provienen de [kaggle.com](https://www.kaggle.com), una fuente accesible en línea. Este conjunto de datos muestra una amplia gama de vehículos que comprende cinco clases distintas: Ambulancia, bus, auto, moto y camión . Las imágenes fueron capturadas en una variedad de escenarios del mundo real, lo que las hace ideales para el entrenamiento efectivo de modelos de detección de objetos.

- **Tipo de datos:**

Las imágenes y sus etiquetas son consideradas como tipos de datos no estructurados. Todas las imágenes tienen una resolución uniforme de 416 x 416 píxeles.

- **Formato de los datos:**

El conjunto de datos se presenta en formato JPG para las imágenes y en formato TXT para las etiquetas. Es importante destacar que las anotaciones están estructuradas en el formato YOLOv8.

- **Volumen de los datos:**

El conjunto de datos consiste en un total de 1254 imágenes que se dividen en tres conjuntos principales: el conjunto de entrenamiento que comprende 878 imágenes, el conjunto de validación que consta de 90 imágenes y el conjunto de test que contiene 126 imágenes. Todas las imágenes tienen una resolución uniforme de 416 x 416 píxeles. Cada imagen viene acompañada de un archivo de etiquetas en formato TXT que proporciona detalles sobre la clase del objeto, la ubicación mediante coordenadas y las dimensiones (ancho x alto) de los vehículos presentes en la imagen. En conjunto, estas características conforman el volumen de datos disponible para el análisis y desarrollo del modelo.

# PROGRAMA M1000IA

Sé parte de la Próxima Generación  
Innovadora de la Fuerza de Trabajo Futura



- **Calidad de los datos:**

La calidad de los datos se evalúa en base a varios aspectos. En primer lugar, este conjunto de datos presenta coherencia y uniformidad en la resolución de las imágenes (416 x 416 píxeles) lo que contribuye a una fácil comparación y procesamiento. Además, la información detallada proporcionada en los archivos de etiquetas, que incluye la clase del objeto, las coordenadas de ubicación y las dimensiones de los vehículos, sugiere una anotación meticulosa y precisa de los datos. En general, la disponibilidad de contar con estos datos bien estructurados y anotados adecuadamente sugiere que es un dataset de calidad y que es propicio para el desarrollo y la evaluación del modelo de detección y conteo de vehículos.

- **Posibilidad de obtener más datos:**

En caso de que este equipo de trabajo considere obtener más datos, se tienen en cuenta las siguientes posibilidades:

- Recopilación de datos adicionales: Se podría realizar una recopilación propia de imágenes y etiquetarlas manualmente. Esto podría implicar la captura de imágenes de vehículos en diferentes entornos y condiciones para aumentar la diversidad y representatividad del conjunto de datos.

- Generación de datos sintéticos: Otra opción es generar datos sintéticos utilizando técnicas de síntesis de imágenes, como el aumento de datos mediante transformaciones geométricas o la generación de imágenes sintéticas mediante el uso de modelos generativos, como las redes generativas adversarias (GAN).

- **Posibilidad de validar y corregir datos:**

Es posible validar los datos realizando una comparación con la fuente original para asegurar que no haya discrepancias significativas. Por ejemplo, haciendo una verificación de la integridad y coherencia de las imágenes y etiquetas en el conjunto de datos y mediante una evaluación de la consistencia en el formato y la estructura de estos. La corrección se daría con la actualización de los datos utilizando la información de la fuente original para corregir cualquier discrepancia identificada, con la incorporación de nuevas imágenes y etiquetas actualizadas desde la fuente original para reemplazar o complementar el conjunto de datos existente.



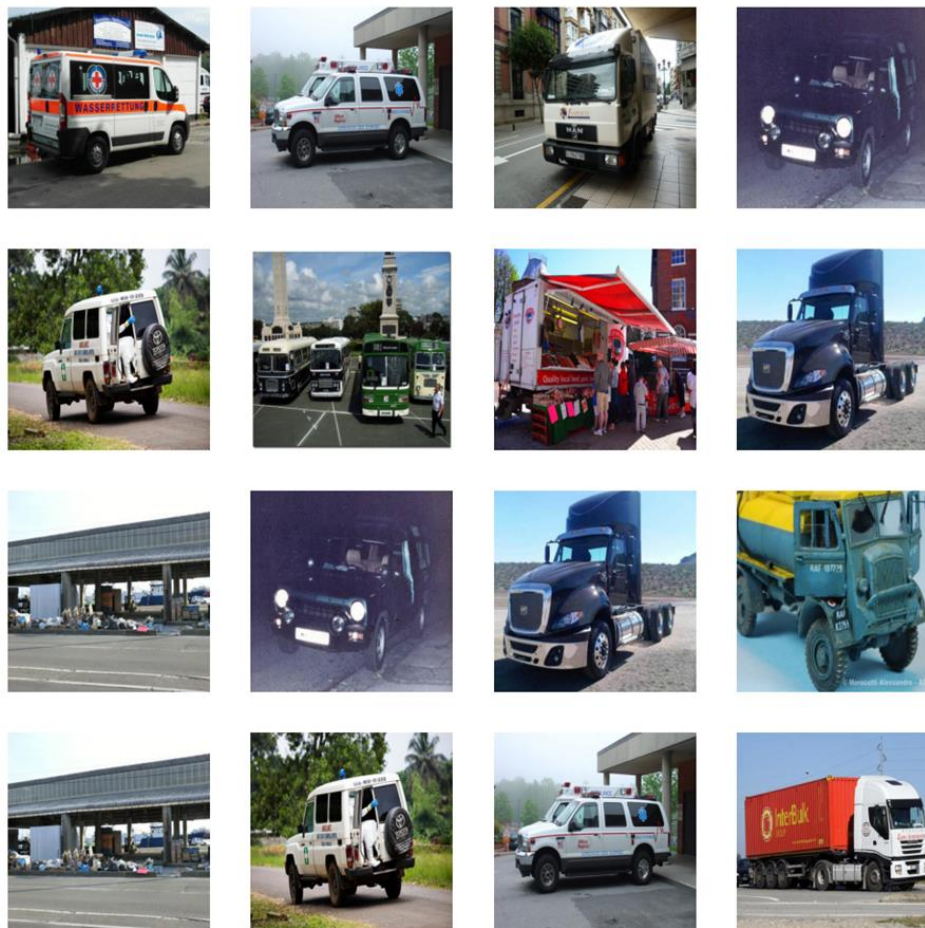
# PROGRAMA M1000IA

Sé parte de la Próxima Generación  
Innovadora de la Fuerza de Trabajo Futura



- Ejemplos de datos:

**Datos de imagen:**



**Datos de etiqueta:**

```
2 0.0985576923076923 0.23557692307692307 0.19110576923076922 0.32091346153846156
2 0.4375 0.5600961538461539 0.8209134615384616 0.8052884615384616
2 0.21995192307692307 0.20072115384615385 0.1045673076923077 0.19831730769230768
2 0.34975961538461536 0.22235576923076922 0.2127403846153846 0.34615384615384615
2 0.9579326923076923 0.27524038461538464 0.0829326923076923 0.14903846153846154
2 0.9711538461538461 0.3233173076923077 0.056490384615384616 0.14182692307692307
```

- Enlace a repositorio con los datos:

Los datos están almacenados en un repositorio en Google Drive, accesible a través del siguiente enlace: [Repositorio de datos](#). Desde allí, se puede acceder a la carpeta que contiene el conjunto de datos.



# PROGRAMA M1000IA

Sé parte de la Próxima Generación  
Innovadora de la Fuerza de Trabajo Futura



## Ética y Privacidad

- **Consideraciones éticas:**

En el desarrollo de este proyecto, se tienen en cuenta diversas consideraciones éticas para garantizar la integridad y el respeto hacia todas las partes involucradas. Esto incluye el manejo responsable de los datos, garantizando la privacidad y confidencialidad de la información personal, así como el cumplimiento de las regulaciones y normativas pertinentes. Se ha trabajado para evitar sesgos en el proceso de análisis y toma de decisiones, asegurando la equidad y la imparcialidad en los resultados obtenidos. Esto incluye la consideración de factores como la representatividad de los diferentes tipos de vehículos, condiciones de iluminación y ubicaciones geográficas en los datos utilizados.

Otro aspecto fundamental ha sido la transparencia en el proceso de desarrollo del modelo. Se ha proporcionado documentación clara sobre el funcionamiento del algoritmo de detección y conteo de vehículos, así como la metodología utilizada para recopilar los datos. Se ha fomentado la apertura y la divulgación de los resultados obtenidos, permitiendo una evaluación independiente y la participación de la comunidad científica y el público en general.

- **Privacidad de los datos:**

Para garantizar la privacidad de los datos se asegura de que los datos recopilados no contengan información personal identificable, como matrículas de vehículos o rostros de personas. Si es necesario utilizar datos adicionales que puedan contener información sensible, como imágenes de vehículos en vías públicas, serán pixeladas las áreas que pudieran identificar a individuos o las matrículas de los vehículos.

- **Cumplimiento normativo:**

Con respecto al cumplimiento normativo, este proyecto se adhiere estrictamente a todas las normativas y regulaciones aplicables en materia de privacidad y protección de datos. Esto incluye el cumplimiento con el Reglamento General de Protección de Datos (GDPR) en la Unión Europea, así como cualquier otra legislación local o nacional relevante en nuestras jurisdicciones operativas. Todas las integrantes del equipo nos comprometemos a respetar los derechos de privacidad de los individuos y a garantizar la seguridad y confidencialidad de los datos utilizados.



## Exploración de Datos (EDA)

- Análisis Exploratorio:

Para realizar un análisis exploratorio detallado de los datos, se procedió a importar el conjunto de datos de entrenamiento y prueba utilizando la biblioteca `os`. El conjunto de datos está compuesto por un total de 1254 imágenes, distribuidas en 878 imágenes para entrenamiento, 250 imágenes para validación y 126 imágenes para test. Todas las imágenes tienen una resolución de 416 x 416 píxeles.

En una primera fase del análisis, se visualizaron las primeras 25 imágenes del conjunto de datos de entrenamiento mediante una matriz de subgráficos de 5x5. Se observó que todas las imágenes presentaban el mismo tamaño de resolución, es decir, 416 píxeles.

Al examinar las etiquetas asociadas a las imágenes, se verificó que no existían datos faltantes. Cada imagen está acompañada de una etiqueta que proporciona detalles sobre la clase del objeto, siendo 0: 'Ambulancia', 1: 'Bus', 2: 'Auto', 3: 'Moto', 4: 'Camión'; así como la ubicación mediante coordenadas y las dimensiones (ancho x alto) de los vehículos presentes en la imagen, representadas por los valores `[class_id, x_center, y_center, width, height]`. Es importante destacar que se evidenció que los datos correspondientes a las etiquetas se encuentran normalizados, dentro del rango de 0 a 1.

Para comprender mejor la distribución de las etiquetas, se realizaron estadísticas descriptivas del número de etiquetas por imagen, así como de las coordenadas y dimensiones de cada vehículo en las etiquetas. Los resultados obtenidos fueron los siguientes:

Estadísticas de número de etiquetas por imagen:

Media: 2

Mediana: 1

Mínimo: 1

Máximo: 9

Moda: 1

Estadísticas de coordenada x:

Media: 0.498

Mediana: 0.5

# PROGRAMA M1000IA

Sé parte de la Próxima Generación  
Innovadora de la Fuerza de Trabajo Futura



Mínimo: 0.019  
Máximo: 0.993  
Desviación estándar: 0.256

Estadísticas de coordenada y:  
Media: 0.504  
Mediana: 0.510  
Mínimo: 0.021  
Máximo: 0.907  
Desviación estándar: 0.171

Estadísticas de ancho:  
Media: 0.393  
Mediana: 0.293  
Mínimo: 0.010  
Máximo: 0.998  
Desviación estándar: 0.303

Estadísticas de altura:  
Media: 0.379  
Mediana: 0.305  
Mínimo: 0.010  
Máximo: 0.998  
Desviación estándar: 0.278

- Distribución de datos:

Número de etiquetas por imagen:

- La media es 2, lo que indica que, en promedio, hay 2 etiquetas por imagen.
- La mediana es 1, lo que sugiere que la mitad de las imágenes tienen 1 etiqueta por lo menos.
- El mínimo es 1 y el máximo es 9, lo que indica una variabilidad en la cantidad de etiquetas por imagen.
- La moda es 1, lo que implica que 1 es el valor más común de etiquetas por imagen.

Coordenada x:

- La media de las coordenadas x es aproximadamente 0.49, lo que sugiere que las coordenadas x tienden a distribuirse alrededor de este valor.

# PROGRAMA M1000IA

Sé parte de la Próxima Generación  
Innovadora de la Fuerza de Trabajo Futura



- La mediana de las coordenadas x es 0.5, lo que indica que la mitad de las coordenadas x están por debajo de este valor y la otra mitad están por encima.
- El rango de las coordenadas x va desde 0.019 hasta 0.993, lo que muestra una amplia dispersión de los datos a lo largo del eje x.
- La desviación estándar de las coordenadas x es 0.256, lo que indica que hay una considerable variabilidad en las coordenadas x.

Coordenada y:

- La media de las coordenadas “y” es aproximadamente 0.5, lo que sugiere una distribución similar a las coordenadas x.
- La mediana de las coordenadas y es 0.51, lo que indica que la distribución de las coordenadas “y” está ligeramente sesgada hacia valores más altos.
- El rango de las coordenadas “y” va desde 0.021 hasta 0.907, mostrando una amplia dispersión similar a las coordenadas x.
- La desviación estándar de las coordenadas “y” es 0.171, también indicando una variabilidad considerable en las coordenadas y.

Ancho:

- El ancho promedio de los vehículos es aproximadamente 0.39, con una mediana de 0.29, lo que sugiere una distribución sesgada hacia valores más altos.
- El rango de los anchos va desde 0.01 hasta 0.99, lo que muestra una amplia variabilidad en los tamaños de los vehículos.
- La desviación estándar del ancho es 0.303, lo que indica una variabilidad considerable en los anchos de los vehículos.

Altura:

- La altura promedio de los vehículos es aproximadamente 0.379, con una mediana de 0.305, sugiriendo una distribución sesgada hacia valores más altos, similar al ancho.
- El rango de las alturas va desde 0.01 hasta 0.99, mostrando una amplia variabilidad en los tamaños de los vehículos.
- La desviación estándar de la altura es 0.278, lo que indica una variabilidad considerable en las alturas de los vehículos.

- **Valores atípicos:**

Dado que las etiquetas de los datos se encuentran normalizadas en un rango entre 0 y 1, y considerando la consistencia en la calidad e integridad de los datos, no se detectan valores atípicos en las coordenadas x y ancho. Sin embargo, se

# PROGRAMA M1000IA

Sé parte de la Próxima Generación  
Innovadora de la Fuerza de Trabajo Futura



observan valores atípicos en la coordenada “y” y la altura de los vehículos, sugiriendo la posibilidad de que estos representen vehículos excepcionalmente altos como camiones grandes, vehículos de carga pesada o autobuses.

- **Valores faltantes:**

Se exploraron las carpetas que contienen los datos del dataset train y test para recopilar los nombres de archivos de imágenes y sus respectivas etiquetas. Posteriormente, se llevó a cabo una verificación para determinar si alguna imagen carecía de su correspondiente etiqueta, notificando cualquier discrepancia identificada. Una vez completada esta fase, las etiquetas fueron cargadas en un DataFrame de pandas para someterlas a un análisis más detallado. Finalmente, se realizó una exhaustiva revisión para detectar posibles datos faltantes en el DataFrame. El resultado de esta verificación confirmó que no se encontraron valores faltantes en las etiquetas del conjunto de datos de train y test.

- **Técnicas de imputación de valores:**

No se aplican técnicas de imputación debido a la ausencia de valores faltantes en los datos.

- **Correlaciones:**

De la exploración de correlaciones entre features se observaron los siguientes hallazgos:

- La correlación muy baja entre las coordenadas X e Y (0,06) sugiere que la posición de un objeto en el eje X no tiene relación lineal significativa con su posición en el eje Y. En otras palabras, donde aparece un objeto en el eje horizontal no influye significativamente en dónde aparece en el eje vertical.
- La ausencia de correlación indica que la posición horizontal de un objeto no tiene una relación lineal con su ancho (-0,00). Esto podría significar que, independientemente de dónde se encuentra un objeto a lo largo del eje X, su tamaño en términos de ancho no se ve afectado.
- Similar al punto anterior, una correlación muy baja (0,02) sugiere que la posición horizontal de un objeto no está relacionada con su altura. El lugar donde aparece un objeto en el eje X no proporciona información sobre su tamaño vertical.
- La correlación baja entre la coordenada Y y el ancho sugiere una relación lineal muy débil (0,10). Esto podría indicar que, en algunos casos, la posición

# PROGRAMA M1000IA

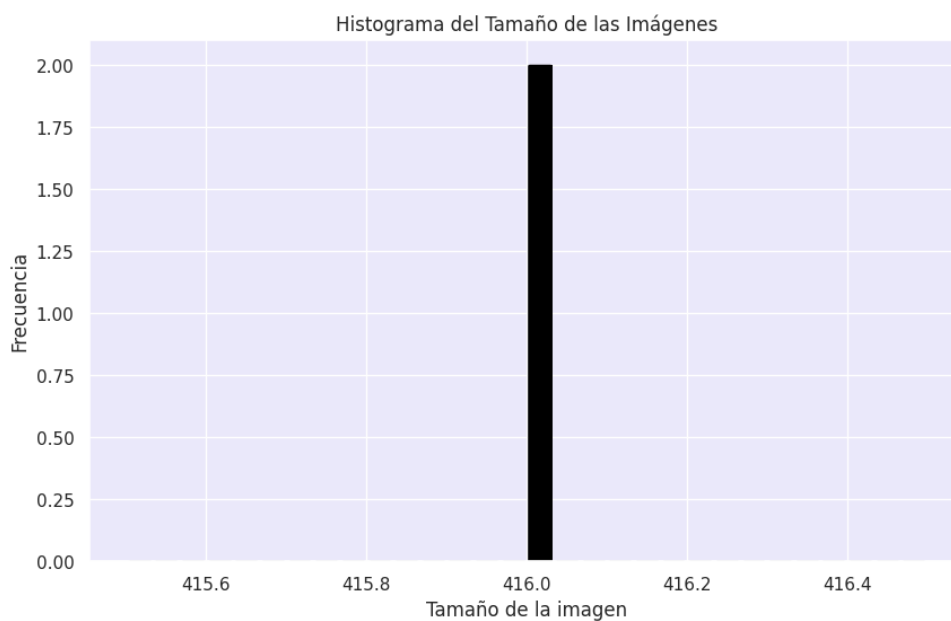
Sé parte de la Próxima Generación  
Innovadora de la Fuerza de Trabajo Futura



vertical de un objeto puede tener una ligera influencia en su ancho, pero esta relación es muy débil y puede no ser consistente.

- Al igual que el punto anterior, una correlación baja (0,08) sugiere una relación lineal muy débil entre la posición vertical de un objeto y su altura. De nuevo, esto indica que la posición en el eje Y no proporciona mucha información sobre el tamaño vertical del objeto.
- Existe una correlación fuerte entre el ancho y la altura que sugiere que los objetos tienden a crecer proporcionalmente en ambas dimensiones. En el contexto de detección de objetos, esto podría indicar que muchos objetos tienen formas similares y que, si un objeto es ancho, también tiende a ser alto. Por ejemplo, un autobús o un camión en las imágenes probablemente será tanto ancho como alto.

- Visualización de datos:



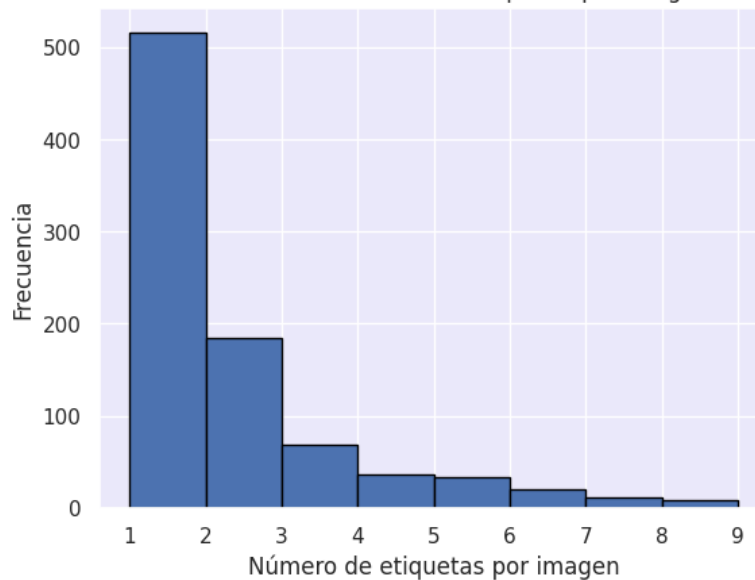


# PROGRAMA M1000IA

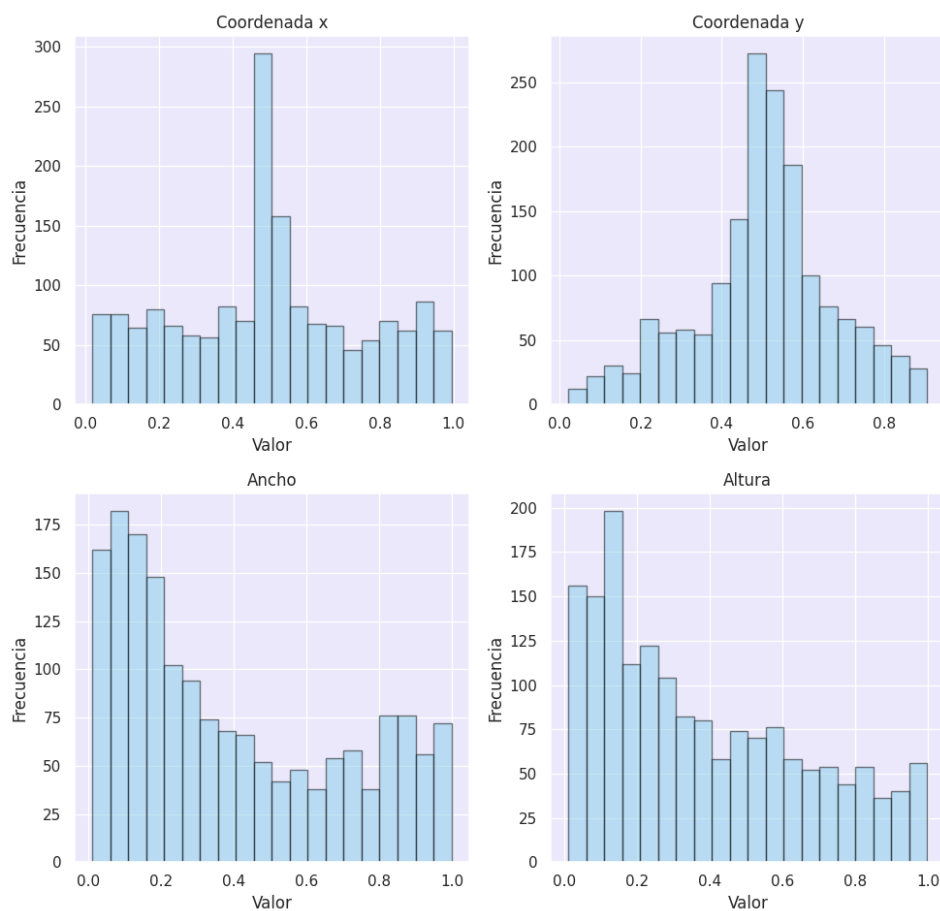
Sé parte de la Próxima Generación  
Innovadora de la Fuerza de Trabajo Futura



Distribución del número de etiquetas por imagen



Distribución de Coordenadas y Dimensiones

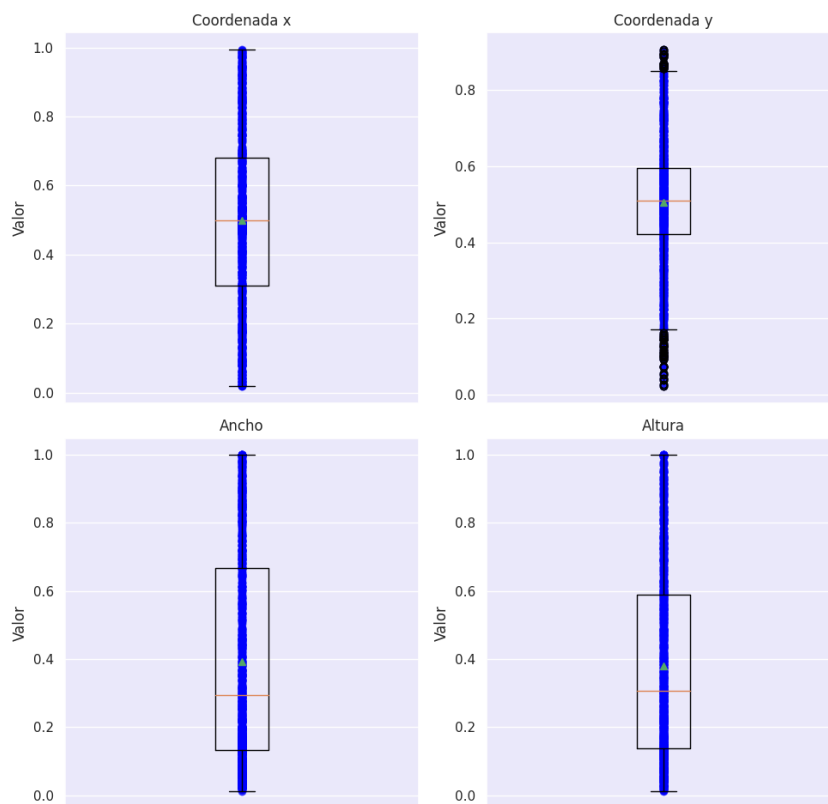


# PROGRAMA M1000IA

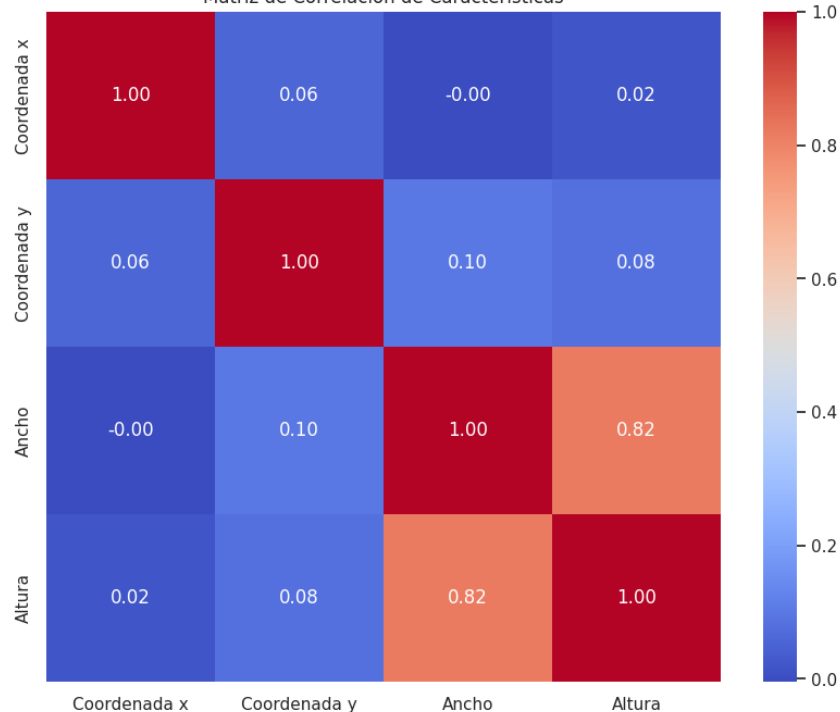
Sé parte de la Próxima Generación  
Innovadora de la Fuerza de Trabajo Futura



Boxplot y Distribución de Puntos de Coordenadas y Dimensiones



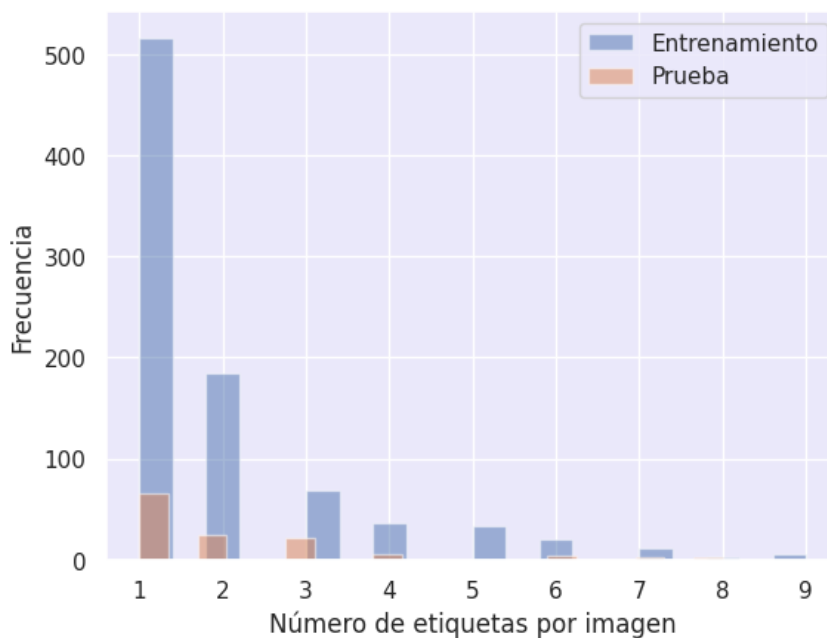
Matriz de Correlación de Características



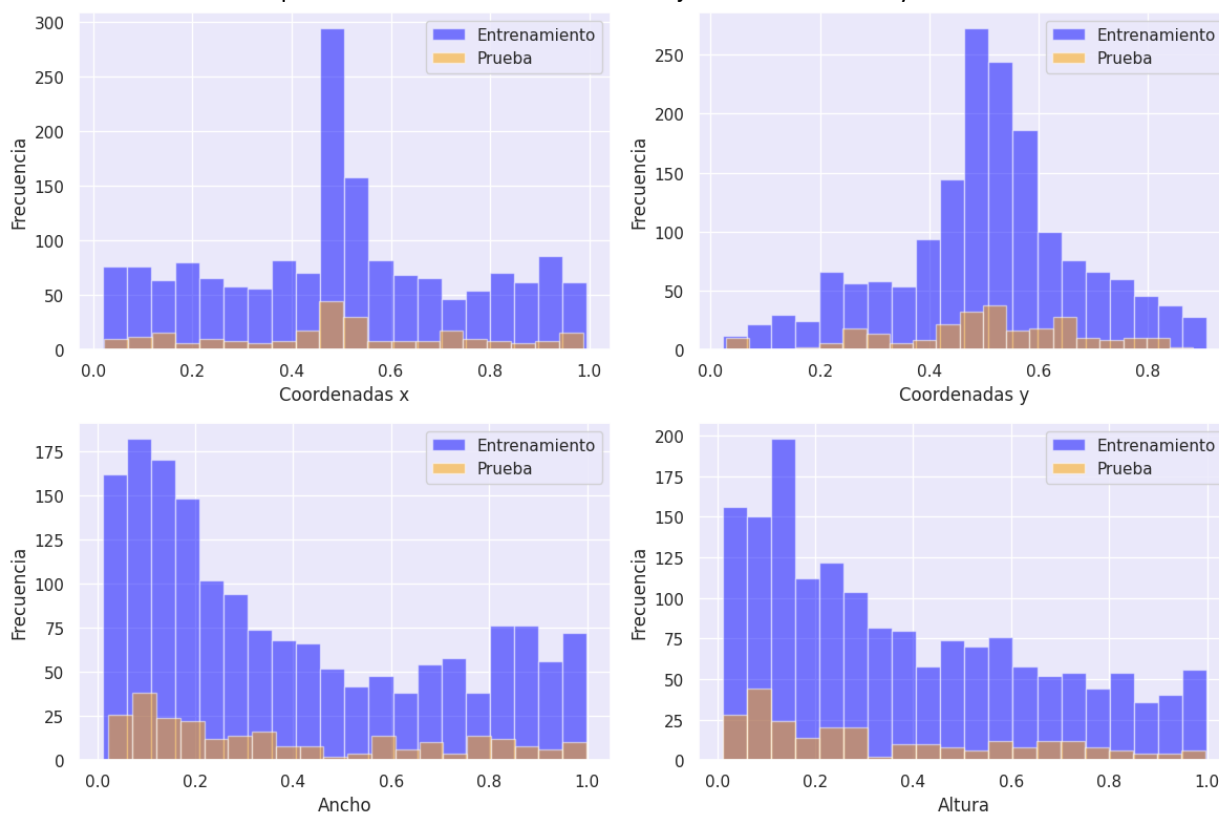
Comparación de distribución entre el conjunto de datos train y test

# PROGRAMA M1000IA

Sé parte de la Próxima Generación  
Innovadora de la Fuerza de Trabajo Futura



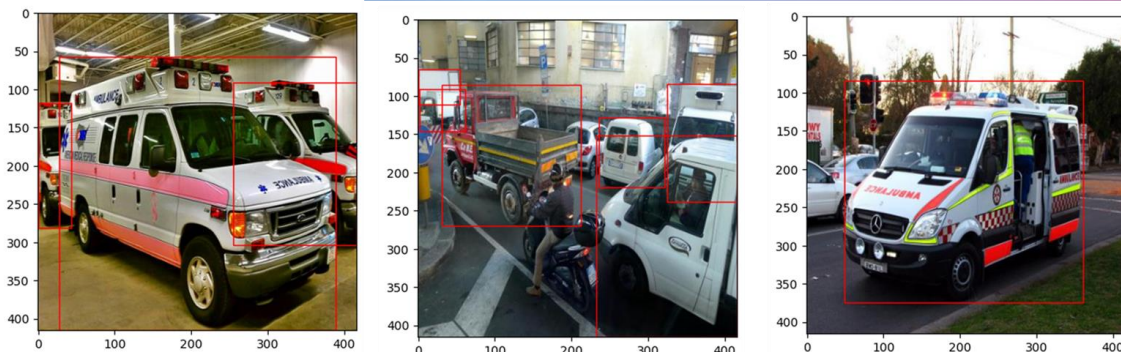
Comparación de distribución entre el conjunto de datos train y test



Representación gráfica de las imágenes con su etiqueta correspondiente

# PROGRAMA M1000IA

Sé parte de la Próxima Generación  
Innovadora de la Fuerza de Trabajo Futura



- Conclusiones de la EDA:

En base al Análisis Exploratorio de Datos, se pueden extraer las siguientes conclusiones:

- La cantidad de etiquetas por imagen varía considerablemente en el conjunto de datos, con un promedio de 2 etiquetas por imagen y una mediana de 1. Esto sugiere que las imágenes contienen una cantidad significativa de objetos de interés, lo que puede ser importante para el diseño y entrenamiento del modelo.
- Las coordenadas x e y, así como las dimensiones de los vehículos, se encuentran normalizadas en el rango de 0 a 1. Esto facilita la comparación y el procesamiento de las etiquetas en problemas de detección de objetos. Las estadísticas descriptivas de estas características proporcionan información útil sobre la distribución espacial y el tamaño de los objetos en las imágenes.
- Todas las imágenes del conjunto de datos tienen la misma resolución de 416 x 416 píxeles. Esta uniformidad simplifica el procesamiento y análisis de las imágenes, ya que no es necesario realizar ajustes adicionales para alinear las resoluciones.
- Se verificó que no existen datos faltantes en las etiquetas asociadas a las imágenes. Esto es crucial para garantizar la integridad y la calidad de los datos utilizados en el entrenamiento del modelo.
- Las bajas correlaciones entre las coordenadas (X e Y) y las dimensiones (ancho y altura) sugieren que los objetos no tienen una ubicación preferente específica que influya en su tamaño. Esto puede ser útil para saber que los objetos en las imágenes no están distribuidos de una manera que depende de su tamaño, lo cual es un buen indicativo de variabilidad en las posiciones de los objetos. Por otro lado, la fuerte correlación entre ancho y altura indica que los objetos en las imágenes tienden a mantener proporciones similares.

# PROGRAMA M1000IA

Sé parte de la Próxima Generación  
Innovadora de la Fuerza de Trabajo Futura



Este tipo de información es útil para los algoritmos de detección y clasificación, ya que pueden aprovecharse de estas proporciones consistentes para mejorar la precisión.

- Comparaciones:

Se comparó la distribución del número de etiquetas por imagen y las características de las etiquetas (coordenadas y dimensiones) de los conjuntos de datos de entrenamiento y prueba, encontrando una tendencia similar en su distribución. Esta similitud sugiere una consistencia entre los conjuntos de datos que facilita la transferencia de conocimiento de modelos preentrenados. Es decir, los modelos preentrenados en un conjunto de datos pueden adaptarse o fine-tunearse más fácilmente en otro conjunto de datos relacionado sin perder su capacidad de generalización.

Además, la estabilidad en la distribución de las características entre los conjuntos de datos de entrenamiento y prueba sugiere que el modelo debería comportarse de manera coherente en diferentes entornos de datos.

- Enlace a notebook con EDA y respectivos datos:

El notebook con el EDA correspondiente está en un repositorio en Google Drive, accesible a través del siguiente enlace: [Notebook](#)



## Modelado

- Selección del modelo:

Se eligió la arquitectura YOLOv8 debido a su eficacia y rendimiento en la detección de objetos en imágenes. Esta arquitectura es conocida por su precisión y eficiencia computacional, lo que la hace adecuada para aplicaciones en tiempo real.

Se entrenaron tres modelos utilizando diferentes variantes de YOLOv8: YOLOv8n, YOLOv8s y YOLOv8m. Estas variantes difieren en la profundidad y complejidad de la red neuronal, lo que puede afectar su rendimiento y velocidad de inferencia. Cada variante está diseñada con diferente profundidad y complejidad de red. Modelos más profundos (como YOLOv8m) suelen tener un rendimiento mejorado debido a su capacidad para aprender representaciones más complejas, pero a menudo a costa de una mayor carga computacional y una velocidad de inferencia más lenta. Por otro lado, modelos más simples (como YOLOv8n o YOLOv8s) pueden ofrecer una velocidad de inferencia más rápida, aunque posiblemente con una ligera disminución en el rendimiento.

- Justificación de la selección del modelo en relación a otros:

La elección de YOLOv8 se debe a su capacidad para manejar múltiples clases de objetos con alta precisión y eficacia. En comparación con otros modelos como YOLOv7 o Faster R-CNN, YOLOv8 ha demostrado tener un rendimiento comparable o superior en tareas de detección de objetos, además de ser más rápido en términos de velocidad de inferencia.

Por otro lado, la elección de entrenar múltiples variantes de YOLOv8 se hizo para evaluar cómo la profundidad y complejidad de la red afectan el rendimiento en la tarea de detección de objetos. Esto permite determinar cuál de las variantes ofrece el mejor equilibrio entre precisión y eficacia.

- Preprocesamiento de datos:

No se requirió ningún procesamiento adicional de los datos, debido a que estos se encuentran correctamente etiquetados. Además, sus etiquetas ya se encuentran normalizadas dentro del rango de 0 a 1.

- Arquitectura del modelo:

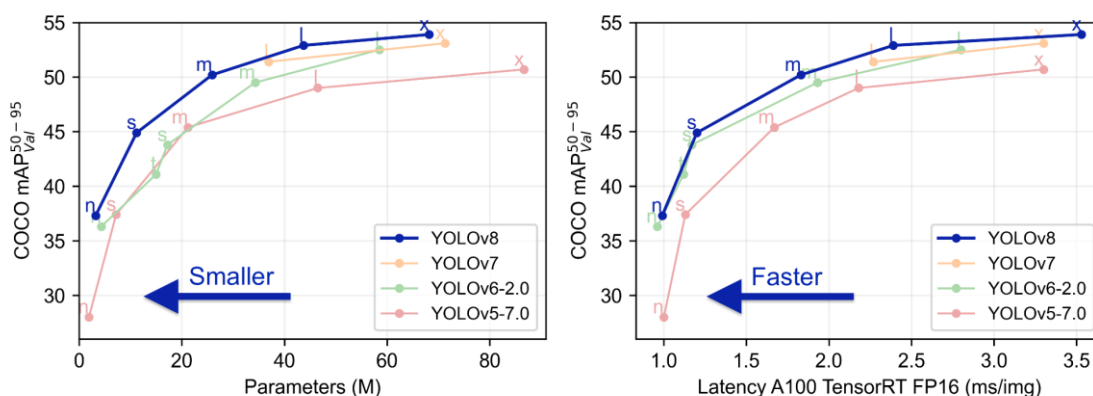


# PROGRAMA M1000IA

Sé parte de la Próxima Generación  
Innovadora de la Fuerza de Trabajo Futura



YOLOv8 utiliza una arquitectura de red neuronal convolucional profunda para la detección de objetos. Esta consta de múltiples capas convolucionales y de agrupación, seguidas por capas completamente conectadas y capas de detección para identificar y localizar objetos en las imágenes. ([Fuente de información: Ultralytics YOLO Docs](#))



- Hiperparámetros del modelo:

Se utilizaron los mismos hiperparámetros para los tres modelos (YOLOv8n, YOLOv8s y YOLOv8m):

- 'epochs': 100 (número de épocas o iteraciones completas a través del conjunto de datos durante el entrenamiento)
  - 'imgsz': 416 (las imágenes tienen un tamaño de 416 píxeles)
  - 'patience': 10 (Este valor representa la paciencia para detener el entrenamiento si no se observa una mejora en el rendimiento del modelo. En este caso, el entrenamiento se detendrá después de 10 épocas consecutivas sin mejorar.)
  - Se utilizan además los parámetros predeterminados para la función de pérdida, learning rate y batch size.
- Enlace a notebook con preparación de los datos, instanciación del modelo, configuración de arquitectura e hiperparámetros:

El notebook con la preparación de los datos, la instanciación del modelo, configuración de arquitectura e hiperparámetros correspondientes está en un repositorio en Google Drive, accesible a través del siguiente enlace: [Notebook](#)

# PROGRAMA M1000IA

Sé parte de la Próxima Generación  
Innovadora de la Fuerza de Trabajo Futura



## Entrenamiento del Modelo

- División de datos (split):

El conjunto de datos ya se encontraba dividido en un 70% para entrenamiento, 20% para validación y 10% para prueba. Esta división se realizó para garantizar un buen rendimiento del modelo y evitar el sobreajuste.

- Método de entrenamiento:

Se utilizó el método de entrenamiento que emplea el modelo preentrenado de YOLOv8, el cual está dado por lotes (batch training) con backpropagation para ajustar los pesos de la red neuronal en cada modelo. Además, este emplea la función de pérdida predeterminada y un optimizador para minimizar la pérdida durante el entrenamiento.

- Técnica de validación:

Se utilizó la técnica de validación en un conjunto de datos de validación independiente para evaluar el rendimiento de cada modelo durante el entrenamiento.

- Duración del entrenamiento:

Cada modelo se entrenó durante 100 épocas.

- Enlace a notebook con entrenamiento del modelo, split de datos, gráfica de métrica de evaluación en conjunto de entrenamiento:

El notebook con el entrenamiento del modelo, split de datos, gráfica de métrica de evaluación en el conjunto de entrenamiento correspondientes está en un repositorio en Google Drive, accesible a través del siguiente enlace: [Notebook](#)

# PROGRAMA M1000IA

Sé parte de la Próxima Generación  
Innovadora de la Fuerza de Trabajo Futura



## Evaluación del Modelo

- Métricas de evaluación utilizadas:

Se utilizaron las siguientes métricas para evaluar el rendimiento del modelo:

- Matriz de confusión: proporciona una visión detallada sobre el rendimiento del modelo al mostrar las predicciones correctas e incorrectas para cada clase.
- Precisión (P): medida de la exactitud de las predicciones positivas del modelo. Indica qué proporción de las predicciones positivas realizadas por el modelo son realmente correctas. Una alta precisión significa que el modelo tiene pocos falsos positivos.
- Recall o sensibilidad: mide la capacidad del modelo para identificar correctamente todas las instancias positivas.
- mAP50 (Mean Average Precision con IoU = 0,50): métrica que combina la precisión y recall para evaluar el rendimiento del modelo. En este caso cuando el umbral de intersección sobre Unión es 0,5.
- mAP50-95 (Mean Average Precision con IoU = 0,50 a 0,95): métrica que combina la precisión y recall para evaluar el rendimiento del modelo. En este caso la media de la precisión promedio es calculada en múltiples umbrales IoU, que van desde 0,5 hasta 0,95. Esta métrica proporciona una evaluación más robusta y exigente del rendimiento del modelo.

# PROGRAMA M1000IA

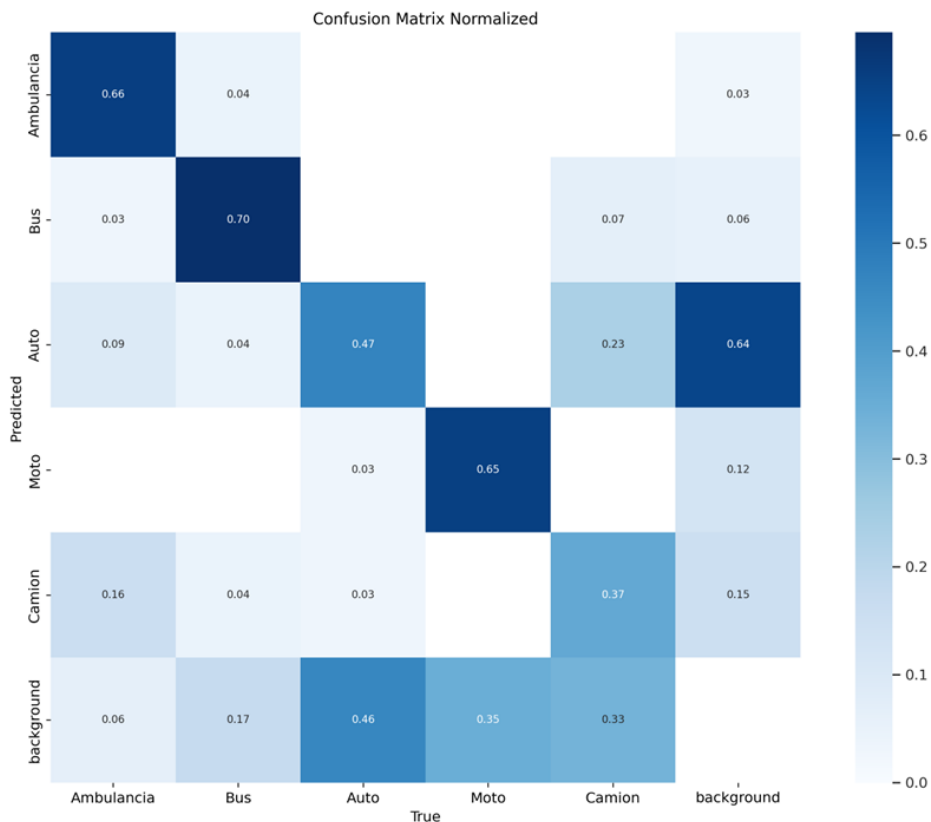
Sé parte de la Próxima Generación  
Innovadora de la Fuerza de Trabajo Futura



- Resultados de la evaluación:

- Modelo YOLOv8n:**

Matriz de confusión: Los porcentajes de acierto fueron los siguientes:  
Ambulancia (66%), Bus (70%), Auto (47%), Moto (65%), Camión (37%).



Class	Images	Instances	Box(P	R	mAP50	mAP50-95):
all	250	454	0.562	0.575	0.564	0.423
Ambulancia	50	64	0.882	0.702	0.819	0.708
Bus	30	46	0.618	0.652	0.635	0.522
Auto	90	238	0.479	0.487	0.401	0.271
Moto	42	46	0.405	0.565	0.525	0.308
Camion	38	60	0.425	0.469	0.439	0.305

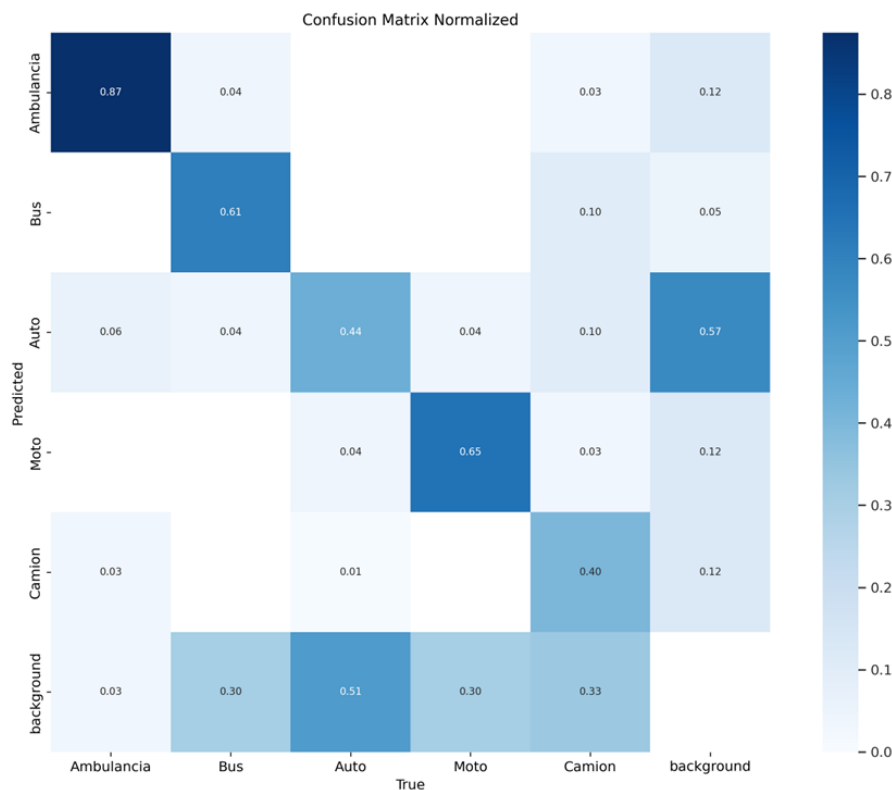
# PROGRAMA M1000IA

Sé parte de la Próxima Generación  
Innovadora de la Fuerza de Trabajo Futura



## - Modelo YOLOv8s:

Matriz de confusión: Los porcentajes de acierto fueron los siguientes:  
Ambulancia (87%), Bus (61%), Auto (44%), Moto (65%), Camión (40%).



Class	Images	Instances	Box(P	R	mAP50	mAP50-95):
all	250	454	0.659	0.618	0.636	0.473
Ambulancia	50	64	0.806	0.91	0.911	0.756
Bus	30	46	0.711	0.652	0.724	0.595
Auto	90	238	0.576	0.437	0.462	0.324
Moto	42	46	0.516	0.652	0.617	0.359
Camion	38	60	0.687	0.44	0.469	0.329

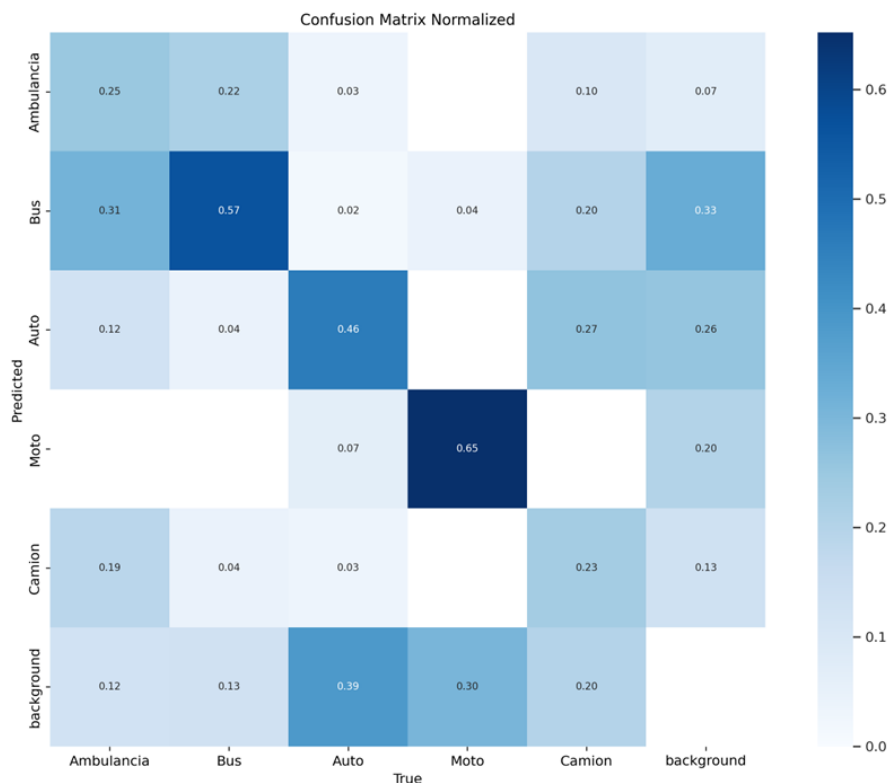
# PROGRAMA M1000IA

Sé parte de la Próxima Generación  
Innovadora de la Fuerza de Trabajo Futura



## - Modelo YOLOv8m:

Matriz de confusión: Los porcentajes de acierto fueron los siguientes:  
Ambulancia (25%), Bus (57%), Auto (46%), Moto (65%), Camión (23%).



Class	Images	Instances	Box(P	R	mAP50	mAP50-95) :
all	250	454	0.287	0.479	0.382	0.265
Ambulancia	50	64	0.527	0.418	0.549	0.412
Bus	30	46	0.148	0.783	0.582	0.436
Auto	90	238	0.373	0.437	0.365	0.249
Moto	42	46	0.181	0.49	0.284	0.155
Camion	38	60	0.204	0.267	0.127	0.0729

Con respecto a la matriz de confusión: Se observa que YOLOv8s tiene la mayor cantidad de aciertos en la clase ambulancia (0.87), superando considerablemente a las otras versiones en la detección de ambulancias. Las imágenes de ambulancias que fueron clasificadas incorrectamente por YOLOv8s tenían una característica en común que la vista de la ambulancia era desde la parte trasera. Este tipo de error no es relevante para el presente proyecto, ya que el objetivo es integrar en un futuro este modelo a un sistema en cámaras de semáforo.



# PROGRAMA M1000IA

Sé parte de la Próxima Generación  
Innovadora de la Fuerza de Trabajo Futura



## MODELO YOLOV8s

Error de clasificación:



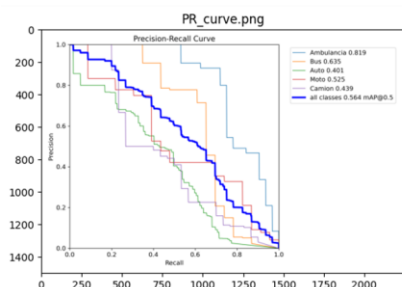
Acierto de clasificación:



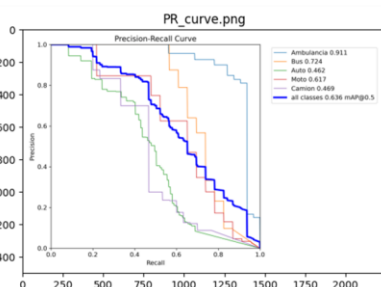
Con respecto a la Precisión y Recall: Se observa que YOLOv8s tiene la mayor precisión (0,659) y un buen recall (0,618), lo que indica que este modelo hace predicciones precisas y es bastante bueno para encontrar la mayoría de los objetos. Por otro lado, YOLOv8n tiene una precisión y recall moderados, lo que sugiere un rendimiento equilibrado, pero no tan alto como el de YOLOv8s. Finalmente, YOLOv8m muestra el rendimiento más bajo en términos de precisión y recall, lo que indica que no es tan efectivo en la tarea de detección de objetos.

En cuanto al mAP50 y mAP50-95: Se observa que YOLOv8s también tiene el mayor mAP50 (0,636) y mAP50-95 (0,473), lo que sugiere que es el mejor modelo en términos de detección precisa de objetos a diferentes umbrales de IoU. Por otro lado, YOLOv8n tiene un mAP50 y mAP50-95 razonables, pero no tan altos como el de YOLOv8s. Finalmente, YOLOv8m tiene los valores más bajos, indicando un rendimiento inferior en comparación con los otros dos modelos.

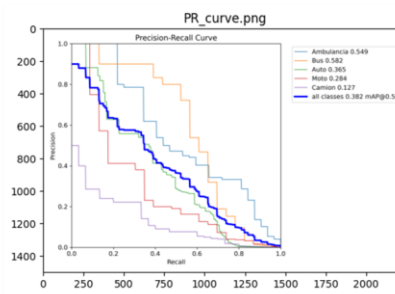
yolov8n



yolov8s

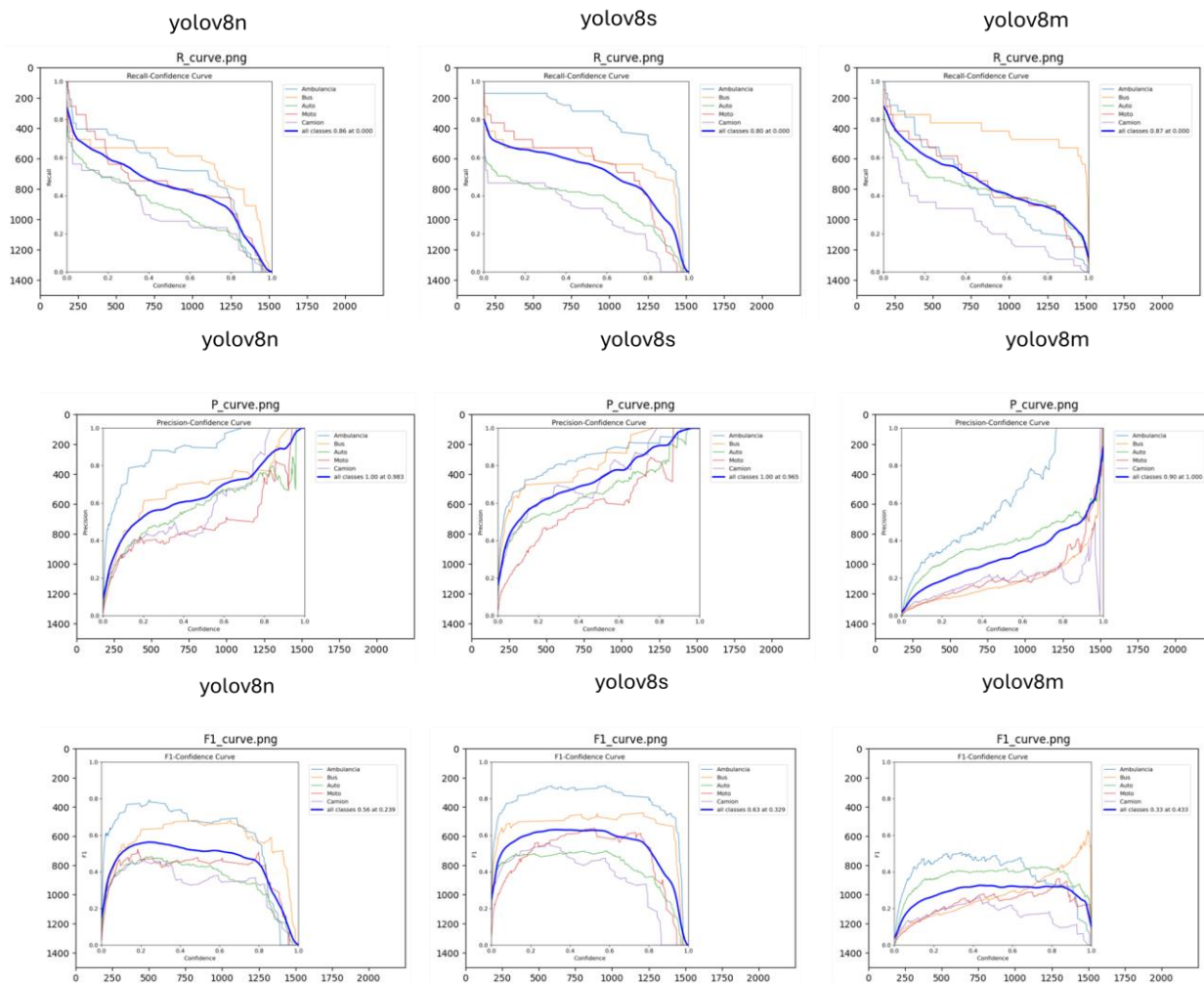


yolov8m



# PROGRAMA M1000IA

Sé parte de la Próxima Generación  
Innovadora de la Fuerza de Trabajo Futura



En resumen, se infiere que YOLOv8s es el modelo más adecuado para resolver el problema del proyecto, debido que ofrece el mejor rendimiento general en términos de precisión, recall y mAP, siendo además una condición a favor de la solución planteada que el modelo en la clase ambulancia obtuvo el porcentaje de acierto más alto.

- Comparación con otros modelos:

Para este proyecto, se optó por utilizar las variantes YOLOv8n, YOLOv8s y YOLOv8m debido a su reconocido rendimiento en tareas de detección de objetos y su balance entre precisión y velocidad de inferencia. Tras evaluar estas tres variantes, los resultados mostraron que YOLOv8s ofrece el mejor desempeño en términos de precisión (P), recall (R), mAP50 y mAP50-95.

Siendo, el objetivo principal del proyecto identificar el mejor modelo dentro de la familia YOLOv8 para la detección de vehículos en imágenes, teniendo en

# PROGRAMA M1000IA

Sé parte de la Próxima Generación  
Innovadora de la Fuerza de Trabajo Futura



cuenta los recursos disponibles y las necesidades específicas del proyecto, en ese sentido, los resultados obtenidos con YOLOv8s cumplen con los requisitos de precisión y recall establecidos para esta aplicación.

Por otro lado, entrenar y evaluar modelos adicionales como Faster R-CNN, SSD y RetinaNet implicaría un uso significativo de recursos computacionales y tiempo. Dado que YOLOv8s ya ofrece resultados satisfactorios que cumplen con los estándares necesarios, se decidió enfocar los recursos en la optimización y aplicación de este modelo. Es por ello que, al momento de realizar el testeo, se ha demostrado que YOLOv8s supera a YOLOv8n y YOLOv8m en todas las métricas clave, proporcionando un rendimiento robusto para la tarea específica de detección de vehículos. Por lo tanto, se considera que la comparación interna es suficiente para respaldar la selección del modelo.

Representación gráfica del testeo realizado con el modelo YOLOv8s



- Enlace a notebook con evaluación del modelo, gráfica de métrica de evaluación en conjunto de entrenamiento, testeo y validación:

El notebook con el entrenamiento del modelo, split de datos, gráfica de métrica de evaluación en el conjunto de entrenamiento correspondientes está en un repositorio en Google Drive, accesible a través del siguiente enlace: [Notebook](#)





## Disponibilización del Modelo como API

- Descripción de la API:

La API de Detección de Vehículos emplea el modelo YOLOv8s para identificar diferentes tipos de vehículos en imágenes, tales como Ambulancias, Buses, Autos, Motos y Camiones. Esta API se ha implementado utilizando FastAPI, un framework moderno y eficiente para la creación de APIs en Python.

Al subir una imagen, la API la procesa y devuelve la misma imagen con los vehículos detectados resaltados mediante cuadros delimitadores y etiquetas que indican el tipo de vehículo y la confianza de la detección. Además, si se detecta una ambulancia, la API genera un mensaje adicional informando que el semáforo se pondrá en verde para facilitar el tránsito del vehículo de emergencia. Para otros tipos de vehículos, se genera un mensaje indicando que el semáforo mantiene su programación de tráfico normal.

- Documentación de la API:

La API cuenta con la siguiente documentación:

**app.py:**

- Carga del Modelo: Carga el modelo YOLOv8s con los pesos entrenados (best.pt).
- Iniciación de la API: Configura y arranca la API con FastAPI.
- Función car\_detect: Procesa las imágenes proporcionadas para detectar vehículos utilizando el modelo. Genera un mensaje adicional dependiendo del tipo de vehículo detectado (vehículo de emergencia o vehículo regular).
- Endpoint: POST /detect/: Recibe una imagen y devuelve la imagen procesada con los vehículos detectados y los cuadros delimitadores correspondientes. Incluye un mensaje adicional si se detecta una ambulancia.

**Front\_streamlit.py:**

- Definición de la URL de la API: Especifica la URL de la API para el cliente.
- Carga y Envío de la Imagen: Permite al usuario subir una imagen (jpg, jpeg, png) y enviar una solicitud a la API.
- Recepción y Visualización de Resultados: Muestra la imagen procesada con las detecciones y el mensaje adicional.

# PROGRAMA M1000IA

Sé parte de la Próxima Generación  
Innovadora de la Fuerza de Trabajo Futura



**Run\_api.sh:** Script para iniciar la interfaz de Streamlit y la API de FastAPI.

## Requisitos y Respuestas:

- file (required): Imagen a procesar; debe ser un archivo de tipo jpg, jpeg o png.
- Respuestas:  
200 OK: Devuelve la imagen procesada con las detecciones.  
500 Internal Server Error: Devuelve un mensaje de error si algo sale mal durante el procesamiento de la imagen.

### • Uso de la API:

Para utilizar la API de detección de vehículos, se debe seguir estos pasos:

- Iniciar la API y la interfaz de usuario: en la terminal ejecutar: `./run_api.sh`.
- Acceder a la interfaz de usuario: Luego abrir un navegador y dirigirse a `http://localhost:8501/`. Y aparecerá la siguiente página en el entorno local:

## Detección de Vehículos

Sube una imagen de un vehículo y el modelo detectará los objetos:

Sube una imagen



Drag and drop file here

Limit 200MB per file • JPG, JPEG, PNG

Browse files

- Subir una imagen y detectar vehículos: En la interfaz de usuario de Streamlit, sube una imagen en formato JPG, JPEG o PNG. Y dar clic en el

botón

Detectar Vehículos

- Resultados:

La imagen procesada con los vehículos detectados se mostrará en la pantalla. Para cada vehículo detectado, se mostrará un cuadro delimitador con la etiqueta del tipo de vehículo y la probabilidad de confianza. Si se detecta una ambulancia, se mostrará el mensaje de alerta: "Vehículo de emergencia, semáforo en verde". Para otros vehículos, se mostrará el mensaje: "Semáforo mantiene la programación de tráfico normal".

# PROGRAMA M1000IA

Sé parte de la Próxima Generación  
Innovadora de la Fuerza de Trabajo Futura



- Ejemplos de uso de la API:

Ejemplo de un tipo de vehículo que no es de emergencia:

## Detección de Vehículos

Sube una imagen de un vehículo y el modelo detectará los objetos:

Sube una imagen



Drag and drop file here

Limit 200MB per file • JPG, JPEG, PNG

Browse files



imag3.jpg 239.7KB



Imagen subida

Detectar Vehículos



Imagen con detecciones

Mensaje del semáforo: Semáforo mantiene la programación de tráfico normal

Ejemplo de un vehículo de emergencia:



# PROGRAMA M1000IA

Sé parte de la Próxima Generación  
Innovadora de la Fuerza de Trabajo Futura



## Detección de Vehículos

Sube una imagen de un vehículo y el modelo detectará los objetos:

Sube una imagen



Drag and drop file here

Limit 200MB per file • JPG, JPEG, PNG

Browse files



ambulance-2018-crop.jpg 1.2MB



Imagen subida

Detectar Vehículos



Imagen con detecciones

Mensaje del semáforo: Vehículo de emergencia detectado, semáforo en verde

# PROGRAMA M1000IA

Sé parte de la Próxima Generación  
Innovadora de la Fuerza de Trabajo Futura



- Enlace a código python que despliega la aplicación como API:

El código python que despliega la aplicación como API está en un repositorio en Google Drive, accesible a través del siguiente enlace: [API](#)

# PROGRAMA M1000IA

Sé parte de la Próxima Generación  
Innovadora de la Fuerza de Trabajo Futura



## Implementación

- Integración del modelo:

El modelo YOLOv8 ha sido integrado en el sistema mediante FastAPI, lo que permite disponibilizar sus capacidades de detección de vehículos a través de una API RESTful. A continuación, se detalla cómo se ha llevado a cabo esta integración:

- Carga del modelo: Se carga el modelo YOLOv8s utilizando la librería ultralytics con los pesos entrenados almacenados en el archivo best.pt. Este archivo contiene los parámetros optimizados para la detección de vehículos como Ambulancias, Buses, Autos, Motos y Camiones.
- Despliegue de la API: Se utiliza FastAPI para crear un servidor que escucha las solicitudes en el endpoint /detect/. La API recibe una imagen, la procesa usando el modelo YOLOv8, y devuelve la imagen con los vehículos detectados junto con cuadros delimitadores y etiquetas de confianza.
- Interfaz de usuario: Se ha desarrollado una interfaz de usuario con Streamlit que permite a los usuarios subir imágenes y visualizar los resultados de la detección de vehículos directamente en su navegador. La interfaz está conectada a la API para enviar las imágenes y recibir las respuestas procesadas.
- Mensajes adicionales: Se ha implementado lógica adicional para generar mensajes de alerta en función del tipo de vehículo detectado. Si se detecta una ambulancia, se muestra un mensaje indicando que el semáforo se pondrá en verde para facilitar el tránsito del vehículo de emergencia. Para otros tipos de vehículos, se muestra un mensaje indicando que el semáforo mantiene la programación de tráfico normal.

Enlace al video con el modelo implementado: [Disponibilización del modelo](#)

- Pruebas de integración:

Para garantizar que la integración del modelo en el sistema sea exitosa, se han realizado varias pruebas de integración:

# PROGRAMA M1000IA

Sé parte de la Próxima Generación  
Innovadora de la Fuerza de Trabajo Futura



- Pruebas unitarias: Se probaron componentes individuales como la función *car\_detect* que realiza la detección de vehículos y genera las imágenes procesadas. Además, se verificó que el modelo carga correctamente y realiza detecciones precisas en un entorno controlado.
- Pruebas de API: Se enviaron solicitudes POST al endpoint */detect/* con diferentes imágenes para asegurar que la API procese las imágenes correctamente y devuelva las respuestas esperadas. Se realizó la validación de los códigos de respuesta HTTP y el formato de las imágenes devueltas.
- Pruebas de interfaz de usuario: Se verificó la funcionalidad completa de la interfaz de usuario desarrollada con *Streamlit*. Además, se realizó pruebas de carga de imágenes, envío de solicitudes a la API y visualización de resultados. Finalmente, se hizo la validación de la correcta visualización de los mensajes de alerta según el tipo de vehículo detectado.
- Pruebas de casos límite: Se probaron imágenes de baja calidad, imágenes sin vehículos y diferentes resoluciones de imágenes para asegurar que el sistema maneje adecuadamente estos casos. Se realizó la verificación de que el sistema no falle y proporcione retroalimentación adecuada en situaciones de error.
- Pruebas de rendimiento: Se evaluó el tiempo de respuesta de la API y la capacidad del sistema para manejar múltiples solicitudes concurrentes. Se vio la optimización del rendimiento para garantizar tiempos de respuesta rápidos y una experiencia de usuario fluida.

Las imágenes con diferentes resoluciones y condiciones de iluminación, nitidez, etc, para la realización de pruebas se encuentran en el siguiente enlace:

[Imágenes para pruebas de integración](#)



## Evaluación del proyecto

- Cumplimiento de objetivos:

El proyecto de detección de vehículos con YOLOv8 y FastAPI ha cumplido con éxito los objetivos establecidos:

- El modelo ha sido correctamente integrado y expuesto a través de una API RESTful utilizando FastAPI. Los usuarios pueden enviar imágenes a la API y recibir imágenes procesadas con vehículos detectados.
- Se desarrolló una interfaz de usuario amigable utilizando Streamlit, permitiendo a los usuarios subir imágenes, realizar detecciones y visualizar los resultados de manera interactiva.
- Se añadió funcionalidad para generar mensajes de alerta específicos para ambulancias, indicando que el semáforo se pondrá en verde, y mensajes genéricos para otros tipos de vehículos, indicando la programación normal del semáforo.
- Se realizaron pruebas unitarias, de integración, de rendimiento y de casos límite, asegurando que el sistema funcione correctamente bajo diversas condiciones.

- Lecciones aprendidas:

Durante el desarrollo del proyecto, se aprendieron varias lecciones valiosas:

1. Establecer objetivos y metas claras, de tal forma de no perder la finalidad del modelado de los datos.
2. Tener un dataset de calidad y variado para poder conseguir un modelo robusto.
3. Tener el código modular y bien organizado para que facilite la disponibilización del mismo. Además de que facilitará a futuro las pruebas, el mantenimiento y la expansión del proyecto.
4. La interfaz que se desarrolle debe ser intuitiva y fluida para su adopción y el uso efectivo por parte de los usuarios finales.
5. Es importante realizar pruebas exhaustivas en diferentes escenarios y con diferentes tipos de datos para asegurar la robustez y fiabilidad del sistema.

# PROGRAMA M1000IA

Sé parte de la Próxima Generación  
Innovadora de la Fuerza de Trabajo Futura



- Éxito en la resolución del problema:

El problema planteado ha sido resuelto con éxito utilizando el modelo YOLOv8, obteniéndose los siguientes logros:

- El modelo detecta con alta precisión varios tipos de vehículos, incluyendo ambulancias, buses, autos, motos y camiones.
- La API no solo detecta vehículos, sino que también genera mensajes de alerta específicos para vehículos de emergencia, demostrando una potencial aplicación en la gestión del tránsito.

En resumen, el proyecto ha logrado resolver el problema planteado de manera efectiva, cumpliendo todos los objetivos establecidos y proporcionando una herramienta valiosa para la detección y gestión de vehículos en entornos urbanos.



# PROGRAMA M1000IA

Sé parte de la Próxima Generación  
Innovadora de la Fuerza de Trabajo Futura



## Próximos pasos y oportunidades de mejora

- Sugerencias para mejorar el modelo:

Para mejorar el modelo se sugiere lo siguiente:

- Incorporar más datos de diferentes condiciones de iluminación, clima y ángulos de cámara puede mejorar la precisión y robustez del modelo.
- Usar técnicas avanzadas de augmentación de datos para generar más variabilidad en los datos de entrenamiento y mejorar la capacidad del modelo para generalizar.
- Explorar y probar métodos de detección de objetos más avanzados o híbridos que puedan complementar a YOLOv8.
- Establecer un proceso continuo de evaluación y mejora del modelo basado en los datos de rendimiento y retroalimentación de los usuarios.

- Oportunidades futuras:

Se identifican las siguientes oportunidades:

- Explorar la integración de la API con sistemas de gestión de tráfico en tiempo real para mejorar la eficiencia del tráfico urbano y la respuesta a emergencias.
- Ampliar el modelo para detectar más tipos de vehículos y otros objetos relevantes en el contexto urbano, como peatones y señales de tráfico.
- Añadir funcionalidades como el seguimiento de vehículos a lo largo del tiempo y la predicción de trayectorias.
- Optimizar el modelo y la API para ser implementados en dispositivos edge, permitiendo su uso en cámaras de tráfico y otros dispositivos IoT.
- Buscar colaboraciones con expertos en tráfico, urbanismo y seguridad para desarrollar soluciones integrales y multidisciplinarias.

# PROGRAMA M1000IA

Sé parte de la Próxima Generación  
Innovadora de la Fuerza de Trabajo Futura



## Mantenimiento y actualización

- Plan de mantenimiento:

Una vez que se implemente el modelo, se establecerá un sistema de monitoreo continuo para la API y el modelo de tal forma de detectar y resolver problemas de manera efectiva. Asimismo, se revisará periódicamente el rendimiento del modelo y la API para asegurar que se sigan cumpliendo con los requisitos para los que fue destinado y proporcionar a su vez el soporte técnico necesario. Además, se tiene en consideración la creación de una plataforma para que los usuarios reporten problemas o soliciten las mejoras correspondientes.

- Actualización del modelo:

Se planea actualizar el modelo regularmente, con la utilización de nuevos conjuntos de datos de entrenamiento, propios del despliegue en la vida real, a fin de que se pueda reflejar los cambios en el entorno y con ello mejorar la precisión del modelo. Asimismo, se tienen en cuenta, actualizar y mejorar la infraestructura de la API para aumentar su capacidad, seguridad y rendimiento. Por otro lado, se tendrá en consideración, recoger y analizar la retroalimentación por parte de los usuarios para identificar áreas de mejora y actualizar el modelo en consecuencia.

- Re-entrenamiento del modelo:

Se planea en un futuro realizar el re-entrenamiento del modelo de manera periódica usando datos actualizados y técnicas mejoradas de entrenamiento. Se tiene en consideración, que este re-entrenamiento se realizará sobre todo cuando el modelo presente problemas de precisión, cuando haya cambios significativos en el entorno o cuando se incorporen nuevas funcionalidades que requieran ajustes en el modelo.

# PROGRAMA M1000IA

Sé parte de la Próxima Generación  
Innovadora de la Fuerza de Trabajo Futura



## Seguimiento Post-Implementación

- Monitorización del modelo en producción:
  - Se sugiere implementar herramientas de monitoreo en tiempo real como Prometheus para supervisar el rendimiento del modelo y la API. Esto incluye la monitorización de la latencia de respuesta, el uso de recursos como CPU, memoria, etc. y las tasas de error.
  - Configurar alertas automatizadas para notificar a los administradores cuando se detecten anomalías en el rendimiento del modelo, por ejemplo cuando haya un aumento en la tasa de error o una disminución en la precisión de las detecciones.
  - Crear dashboards personalizados para visualizar las métricas claves de rendimiento del modelo y la API, y mediante ellos hacer una evaluación rápida del estado del sistema.
- Resolución de problemas:
  - **Establecimiento de un Equipo de Soporte Técnico:** Creación de un equipo dedicado que esté disponible para resolver problemas reportados por los usuarios, capacitado para responder con rapidez y ofrecer soluciones efectivas.
  - **Desarrollo de Procedimientos para la Respuesta a Incidentes:** Formulación de procedimientos claros y documentados que incluyan identificación de problemas, priorización de incidentes según su impacto, asignación de recursos para la resolución y comunicación con los usuarios afectados.
  - **Análisis Post-Mortem:** Realización de análisis detallados después de resolver incidentes críticos para identificar las causas raíz y tomar medidas preventivas, mejorando así la resiliencia del sistema.
  - **Establecimiento de un Loop de Retroalimentación:** Implementación de un ciclo de retroalimentación con los usuarios para recoger comentarios y sugerencias sobre el rendimiento del modelo y la API, y usar esta información para identificar áreas de mejora y priorizar actualizaciones necesarias.

### Optimización Continua del Modelo:

# PROGRAMA M1000IA

Sé parte de la Próxima Generación  
Innovadora de la Fuerza de Trabajo Futura



- **Evaluación Regular del Modelo:** Implementación de un proceso iterativo y sistemático para evaluar regularmente el rendimiento del modelo y la efectividad de las actualizaciones.
- **Recalibración de Hiperparámetros y Datos de Entrenamiento:** Recalibración periódica de hiperparámetros y la incorporación de nuevos datos de entrenamiento para refinar continuamente el modelo.
- **Ajustes en la Infraestructura Tecnológica:** Realización de ajustes necesarios en la infraestructura tecnológica para mantener el modelo actualizado y optimizado.
- **Uso de Técnicas Avanzadas de Aprendizaje Automático:** Aplicación de técnicas avanzadas para adaptar y mejorar el modelo de forma proactiva en respuesta a cambios en los patrones de datos y retroalimentación de los usuarios, asegurando la relevancia y precisión del modelo.

- Optimización continua:

Se implementará un enfoque de actualización incremental, caracterizado por la realización de mejoras y optimizaciones continuas en pequeña escala. Esto incluirá desde ajustes en los hiperparámetros del modelo hasta la incorporación de nuevos conjuntos de datos de entrenamiento y el perfeccionamiento de la infraestructura de API.

Para evaluar el impacto de estos cambios, se llevarán a cabo pruebas A/B. Se comparará el rendimiento de diferentes versiones del modelo en el entorno de producción, permitiendo seleccionar la configuración que ofrezca el rendimiento más óptimo.

# PROGRAMA M1000IA

Sé parte de la Próxima Generación  
Innovadora de la Fuerza de Trabajo Futura



## Reflexiones

- Experiencia personal de cada una de las integrantes: [Compartir una reflexión personal sobre la experiencia en el proyecto]

Kary Francia: Mi experiencia en el proyecto ha sido profundamente enriquecedora al poder desarrollar una solución que integra tecnologías avanzadas como YOLOv8s para la detección de vehículos y FastApi. Como Data Science Jr., este proyecto representó mi primer contacto con modelos de este tipo, lo cual enriqueció considerablemente mis conocimientos, ya que antes no había trabajado con datos no estructurados. Lograr hacer la implementación de la API y en la integración del modelo YOLOv8s me permitió comprender la importancia crítica de la robustez, eficiencia y escalabilidad en los sistemas de visión por computadora, especialmente en aplicaciones del mundo real. La resolución de desafíos técnicos, como la optimización del rendimiento del modelo y la gestión de errores, fue fundamental para aprender a garantizar un producto final de alta calidad. Además, adquirí un entendimiento profundo sobre la disponibilización del modelo y la resolución de problemas post-implementación, aspectos que son vitales para comprender el ciclo completo de desarrollo de aplicaciones de inteligencia artificial.

- Impacto en la industria o sociedad:

El presente proyecto tiene el potencial de impactar positivamente tanto en la industria como en la sociedad en varios aspectos claves:

- Eficiencia en gestión de emergencias: La capacidad de detectar y priorizar vehículos de emergencia como ambulancias puede mejorar significativamente los tiempos de respuesta en situaciones críticas, como accidentes de tráfico o emergencias médicas. Esto podría salvar vidas y reducir el impacto negativo en la sociedad.
- Seguridad vial: Al detectar y clasificar diferentes tipos de vehículos como autos, camiones y motos, la API puede contribuir a sistemas avanzados de asistencia al conductor y de gestión de tráfico. Esto promueve una conducción más segura y eficiente en las carreteras, reduciendo el riesgo de accidentes.

# PROGRAMA M1000IA

Sé parte de la Próxima Generación  
Innovadora de la Fuerza de Trabajo Futura



- Automatización de procesos: En la industria, la automatización de la detección de vehículos puede optimizar operaciones logísticas y de transporte, mejorando la eficiencia operativa y reduciendo costos.
- Innovación tecnológica: Implementar tecnologías modernas como YOLOv8s y FastAPI en un proyecto real demuestra el potencial de estas herramientas en aplicaciones prácticas. Esto puede inspirar a otros desarrolladores y empresas a explorar y adoptar soluciones avanzadas de inteligencia artificial.