UNIVERSIDAD DE CONCEPCIÓN
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE INGENIERÍA INFORMÁTICA
Y CIENCIAS DE LA COMPUTACIÓN

ANÁLISIS E INTERPRETABILIDAD DE IMÁGENES EN SEGURIDAD LABORAL MEDIANTE MACHINE LEARNING

POR

Jaime Ignacio Ansorena Carrasco

Memoria de Título presentada a la Facultad de Ingeniería de la Universidad de Concepción para optar al título profesional de Ingeniero Civil Informático

Patrocinante
Pedro Pablo Pinacho Davidson

Copatrocinantes
Fernando Andree Tercero Gutiérrez Gómez
Pablo Esteban Aqueveque Navarro

Agosto 2025
Concepción, Chile

Este trabajo fue parcialmente financiado por Proyecto ANID Fondecyt N°11230359, “AN IMMUNE INSPIRED MODEL OF INTRUSION PREVENTION SYSTEM (IPS) FOR COLLABORATIVE AND DISTRIBUTED ENVIROMENTS

«A mi familia y amigos, por su apoyo incondicional;
y a Wendy, por acompañarme en cada paso de este camino.»

Resumen

Esta memoria de título presenta el diseño e implementación de una prueba de concepto para el monitoreo automatizado del cumplimiento de normativas de seguridad laboral. La solución combina detección visual con YOLOv11 para identificar personas y elementos de protección personal, interpretación contextual mediante modelos multimodales de lenguaje (MLLM) y un componente de Retrieval-Augmented Generation (RAG) que fundamenta las evaluaciones en documentación técnica y normativa.

El prototipo fue validado en un entorno industrial real, con participación de distintos profesionales del área. Los resultados indican que la detección alcanzó un mAP50 de 0.775. La evaluación cualitativa de las salidas de los modelos de lenguaje mostró excelentes criterios de coherencia e integridad, mientras que la justificación y relevancia fueron levemente inferiores. Pese a estas restricciones, los resultados evidencian que la interpretación contextual y la generación de alertas aportan valor operativo para la supervisión y la reducción de incidentes.

El autor, Jaime Ignacio Ansorena Carrasco, declara haber utilizado herramientas de inteligencia artificial generativa, como ChatGPT, de manera ética y responsable para apoyar la realización de este trabajo. A continuación, se detalla el uso otorgado:

[✓] **Redacción, Estructuración, Mejora del Texto y Corrección Ortográfica**: Uso de IA para reescribir ideas originales, organizar secciones, mejorar la coherencia, claridad, estilo, corregir errores gramaticales y ortográficos.

[✓] **Traducción**: Uso de IA para traducir textos a distintos idiomas.

[ ] **Generación de Ideas**: Uso de la IA como fuente de inspiración o para explorar enfoques novedosos en el desarrollo del trabajo. Siempre que se han utilizado ideas específicas provenientes de la IA, se ha citado adecuadamente su origen.

[✓] **Asesoría Técnica o Conceptual**: Consulta sobre conceptos técnicos o metodológicos complejos. La información proporcionada por la IA ha sido revisada, contrastada y validada con fuentes académicas o científicas adecuadas para asegurar su precisión y pertinencia.

[ ] **Otros usos**:

El autor declara que todo contenido generado o asistido por IA ha sido revisado, adaptado y validado para asegurar su originalidad y pertinencia. Él es el único responsable del trabajo presentado y se compromete a que las fuentes utilizadas sean debidamente citadas.

Índice general

Índice de tablas

Índice de figuras

Capítulo 1. Introducción

La seguridad laboral constituye un área crítica para la prevención de accidentes y la protección de la salud de los trabajadores, especialmente en sectores de alto riesgo como la construcción, la manufactura y la atención médica. De acuerdo con la Organización Internacional del Trabajo (OIT), cada año se registran aproximadamente 2,93 millones de muertes relacionadas con el trabajo y más de 395 millones de lesiones no mortales a nivel mundial [1].

Para comprender los factores que originan los accidentes laborales, es habitual distinguir entre condiciones inseguras, acciones inseguras y omisiones. Esta clasificación facilita la identificación de áreas prioritarias de intervención. En este contexto, la pirámide de Bird [2] surge como un modelo fundamental: postula que por cada accidente grave o fatal, existen numerosos accidentes menores y cientos de actos o condiciones inseguras previas, lo que enfatiza la necesidad de actuar preventivamente sobre estos niveles inferiores (Figura 1.1).

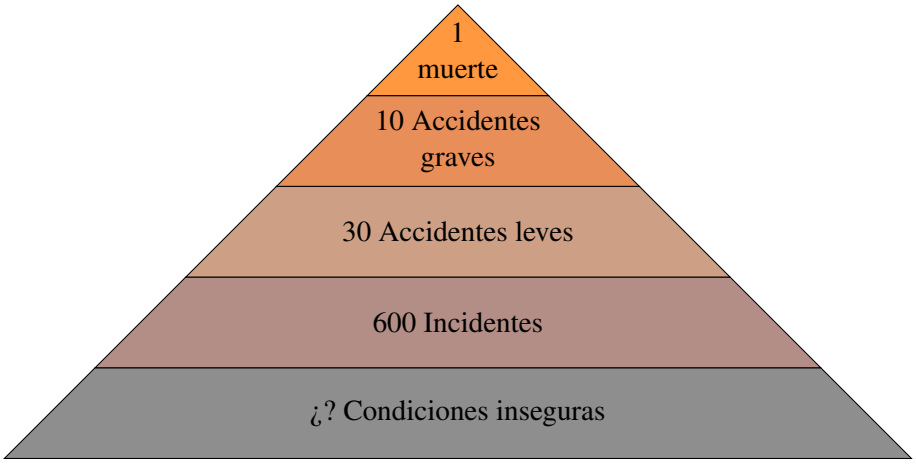


Figura 1.1: Pirámide de Bird.

Organismos internacionales como el National Institute for Occupational Safety and Health (NIOSH) y los Centers for Disease Control and Prevention (CDC) han evidenciado que muchas situaciones de riesgo derivan de la insuficiente implementación de medidas preventivas [3], [4]. Para abordar estos riesgos, estos organismos han elaborado guías y estándares globales, orientados a la reducción de accidentes y enfermedades laborales [5].

En el caso de Chile, el marco normativo se compone principalmente de la Ley N°16.744, que obliga a los empleadores a adoptar medidas para proteger la vida y salud de sus trabajadores [6], y del Decreto Supremo N°594 del Ministerio de Salud, que regula las condiciones sanitarias y ambientales de los lugares de trabajo según su nivel de riesgo [7].

No obstante, los indicadores nacionales siguen evidenciando desafíos. Por ejemplo, en 2023 la Superintendencia de Seguridad Social (SUSESO) reportó 207.477 accidentes laborales, de los cuales el 72 % correspondió a accidentes del trabajo [8]. Las caídas de altura permanecen como la causa principal de accidentes graves y la falta de supervisión efectiva sigue siendo una debilidad recurrente en muchas organizaciones [9]. Como se muestra en la Figura 1.2, a pesar de la disminución sostenida en las tasas de accidentes laborales, aún persiste un número significativo de incidentes, lo que evidencia la necesidad de fortalecer los mecanismos de prevención y supervisión.

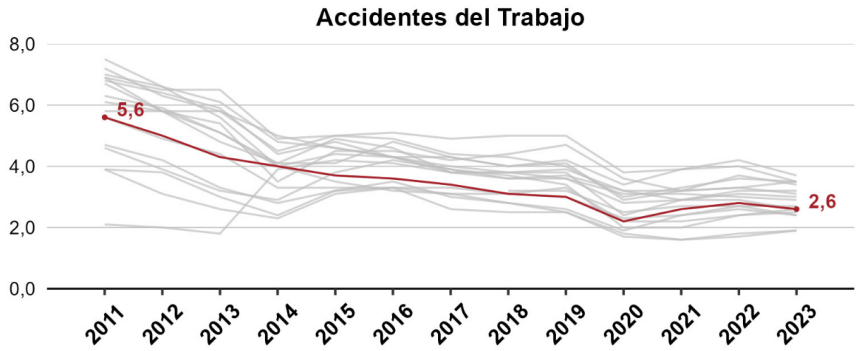


Figura 1.2: Evolución de la Tasa de Accidentabilidad en Chile. El eje de la ordenada representa la tasa de accidentes por cada 100 trabajadores. Cada línea gris muestra la evolución para un sector económico distinto, mientras que la línea roja destaca el promedio nacional. Fuente: [8]

Frente a este escenario, el monitoreo automatizado se presenta como una oportunidad para fortalecer la aplicación y seguimiento de las normativas de seguridad laboral. Los avances recientes en inteligencia artificial, especialmente en modelos de visión como CLIP [10], LLaVA [11], y modelos multimodales como GPT-4o y Gemini, han demostrado capacidades sobresalientes en razonamiento visual y clasificación guiada por texto, facilitando así la supervisión continua y automatizada de la seguridad laboral.

1.1. Presentación del problema

Para reducir los riesgos asociados al contexto laboral, los enfoques tradicionales están basados en la seguridad conductual (Behavior-Based Safety, BBS) que han puesto énfasis en inspecciones manuales acompañadas de acciones correctivas, las cuales han sido efectivas, pero presentan ciertas limitaciones:

1. **Escalabilidad:** Es difícil supervisar simultáneamente múltiples trabajadores en áreas amplias o con alta rotación, resultando en altos costos operacionales.
2. **Contexto:** La necesidad de comprender cuándo es obligatorio un elemento de seguridad depende de múltiples factores contextuales como la tarea específica, la ubicación o el nivel de riesgo.
3. **Flexibilidad:** Las diferencias normativas entre sectores (construcción, minería, industria) dificultan la adaptación de sistemas genéricos.

Los avances en visión computacional y la implementación de sistemas de monitoreo han mitigado algunas de estas limitaciones. Sin embargo, las soluciones suelen identificar solo un rango limitado de peligros y dependen de complejas reglas para la toma de decisiones. Por ello, existe una necesidad de desarrollar sistemas capaces no solo de detectar elementos visualmente, sino también de interpretar el contexto y las implicancias normativas asociadas.

Además, estos sistemas enfrentan retos técnicos: altas tasas de falsos positivos y negativos al detectar objetos y situaciones de riesgo, variabilidad ambiental y elevados costos de procesamiento. A esto se suman aspectos éticos relacionados con la privacidad y la aceptación por parte de los trabajadores.

Por lo tanto, se requiere avanzar hacia sistemas automatizados capaces no sólo de detectar visualmente elementos de seguridad, sino también de interpretar el contexto normativo y operativo en que se producen las situaciones de riesgo. En este sentido, la aplicación de técnicas de *Machine Learning* sobre registros visuales de entornos industriales constituye una oportunidad para superar las limitaciones de escalabilidad, flexibilidad y precisión descritas.

1.2. Objetivos

1.2.1. Objetivo general

Desarrollar un sistema para el análisis automatizado del cumplimiento de normativas de seguridad laboral, mediante detección visual e interpretación contextual.

1.2.2. Objetivos específicos

1. Analizar la normativa vigente en seguridad laboral chilena, identificando los requerimientos asociados a las normativas laborales.
2. Revisar el estado del arte de los enfoques de clasificación visual e interpretación contextual aplicados a seguridad laboral, incluyendo distintos modelos de Machine Learning y técnicas de análisis de imágenes.
3. Diseñar un sistema que integre captura visual, detección de elementos relevantes y análisis del contexto normativo de seguridad laboral.
4. Implementar un prototipo del sistema diseñado que permita la detección automática de cumplimiento normativo.
5. Validar el sistema en un entorno controlado.

1.2.3. Metodología

Con el fin de demostrar la viabilidad técnica y práctica de la propuesta, se adoptó un enfoque aplicado estructurado en cuatro fases:

1. **Revisión bibliográfica:** Se realizó una revisión de la literatura sobre seguridad laboral, con énfasis en la normativa vigente sobre normativas laborales y en las metodologías de análisis visual aplicadas a este contexto.
2. **Diseño del sistema:** Se definió la arquitectura del sistema, considerando la captura de imágenes o video, la detección de elementos, y la interpretación del contexto según las condiciones normativas.
3. **Implementación del prototipo:** Se desarrolló un prototipo funcional como prueba de concepto, abarcando todas las etapas del sistema propuesto. Esta decisión respondió a la extensión y complejidad de la solución en su conjunto.
4. **Validación y evaluación:** Se realizarán pruebas en un entorno real, evaluando la detección, la capacidad de interpretar situaciones normativas y la utilidad de las salidas generadas. Se compararán distintos enfoques de clasificación y se analizará su desempeño en distintos escenarios.

La Figura 1.3 resume gráficamente las etapas de la metodología aplicada, junto con los hitos que representan los resultados de cada fase.

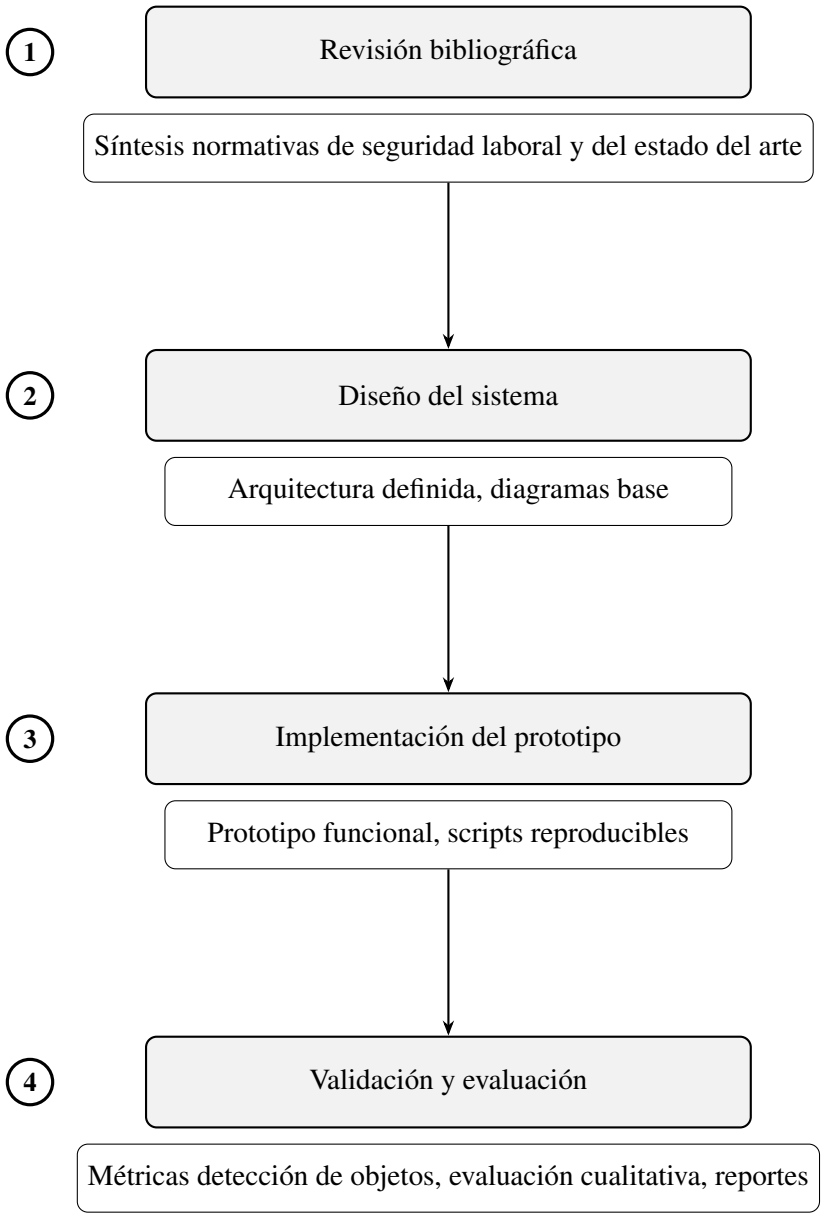


Figura 1.3: Metodología del estudio: etapas (1-4) y principales hitos asociados a cada fase.

Capítulo 2. Marco Teórico

La solución propuesta se fundamenta en diferentes componentes, que incluyen seguridad laboral, visión computacional y grandes modelos de lenguaje (LLM). Si bien se ofrecerán descripciones de estos conceptos, no se profundizará en detalle, ya que no representan el eje central de esta memoria de título.

2.1. Seguridad laboral

La seguridad laboral se entiende como el conjunto de políticas, acciones y normativas orientadas a prevenir riesgos laborales y proteger la integridad física, mental y social de los trabajadores y trabajadoras. Su objetivo principal es anticipar, evitar y reducir la ocurrencia de accidentes de trabajo y enfermedades profesionales, resguardando la calidad de vida y dignidad humana en el entorno laboral. En este sentido, la seguridad laboral no solo promueve la protección individual, sino también el bienestar colectivo, fomentando ambientes de trabajo seguros, saludables y equitativos [12].

En Chile, el marco regulatorio en materia de seguridad y salud en el trabajo se ha consolidado a través de instrumentos legales fundamentales. Entre ellos destaca la Ley N° 16.744, que establece el Seguro Social Obligatorio contra Accidentes del Trabajo y Enfermedades Profesionales.

En esta Ley, se define como accidente del trabajo toda lesión que sufra una persona “a causa o con ocasión del trabajo”, incluyendo también los accidentes de trayecto y las enfermedades profesionales derivadas directamente del ejercicio laboral.

2.2. Detección de objetos

La visión computacional es una rama fundamental dentro de la inteligencia artificial que permite a los ordenadores interpretar y comprender información proveniente de datos visuales, tales como imágenes y vídeos, mediante la aplicación de algoritmos de Machine Learning. Dentro de este campo, una de las áreas de mayor relevancia es la detección de objetos, la cual se refiere al proceso de identificar y localizar de manera precisa distintos objetos presentes en una imagen o secuencia de vídeo. La

importancia de la detección de objetos radica en su aplicabilidad transversal en diversos ámbitos, tales como la seguridad industrial [13], la conducción autónoma [14], vigilancia [15], entre otros.

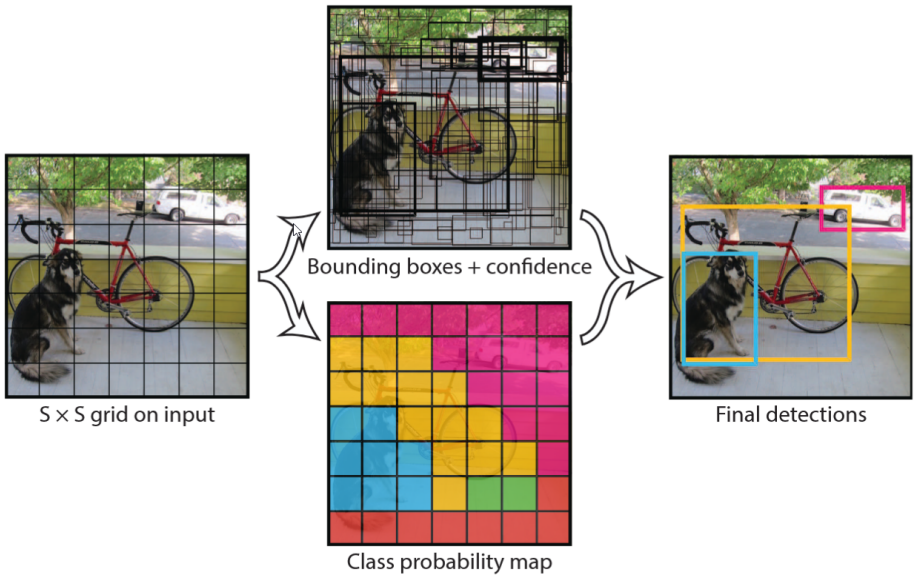


Figura 2.1: Algoritmo YOLO. Divide la imagen en una cuadrícula de tamaño S × S y, para cada celda de dicha cuadrícula, predice B cajas delimitadoras (bounding boxes), junto con la confianza (confidence) asociada a esas cajas, y C probabilidades de clase. Fuente: [16]

Un hito relevante en el desarrollo de esta disciplina fue la introducción del algoritmo You Only Look Once (YOLO), propuesto por Redmon et al [16]. A diferencia de enfoques previos, YOLO reformula la detección de objetos como un problema de regresión, en el cual una única red neuronal es capaz de predecir simultáneamente las cajas delimitadoras (bounding boxes) y las probabilidades de clase correspondientes a los objetos presentes en una imagen, como se observa en la Figura 2.1. Este enfoque unificado permite procesar la imagen completa en una sola evaluación, lo que simplifica significativamente la arquitectura del sistema y habilita el procesamiento en tiempo real.

La arquitectura YOLO [17] se divide en tres componentes. En primer lugar, el **backbone** funciona como el extractor principal de características, utilizando redes neuronales convolucionales para transformar datos de la imagen en mapas de características. En segundo lugar, el componente **neck** actúa como una etapa de procesamiento intermedia, empleando capas especializadas para agregar y mejorar las representaciones de características a través de distintas escalas. Finalmente, el componente **head** cumple la función de mecanismo de predicción, generando las salidas finales para la localización y clasificación de objetos a partir de los mapas de características.

2.3. Modelos de lenguaje

Los grandes modelos de lenguaje (LLM) son modelos generativos basados en redes neuronales profundas, entrenados en grandes volúmenes de datos textuales. Su objetivo es capturar patrones complejos del lenguaje, lo que les permite generar y comprender texto de forma coherente.

El fundamento de los LLM modernos se encuentra en la arquitectura Transformer, introducida por Vaswani et al. [18]. Esta arquitectura se caracteriza por su capacidad para procesar secuencias mediante mecanismos de atención, superando las limitaciones de las arquitecturas recurrentes tradicionales.

El proceso general seguido por los LLM se ilustra en la Figura 2.2, donde se distinguen las etapas principales: preprocesamiento, tokenización, entrenamiento, ajuste de parámetros y generación de salida.

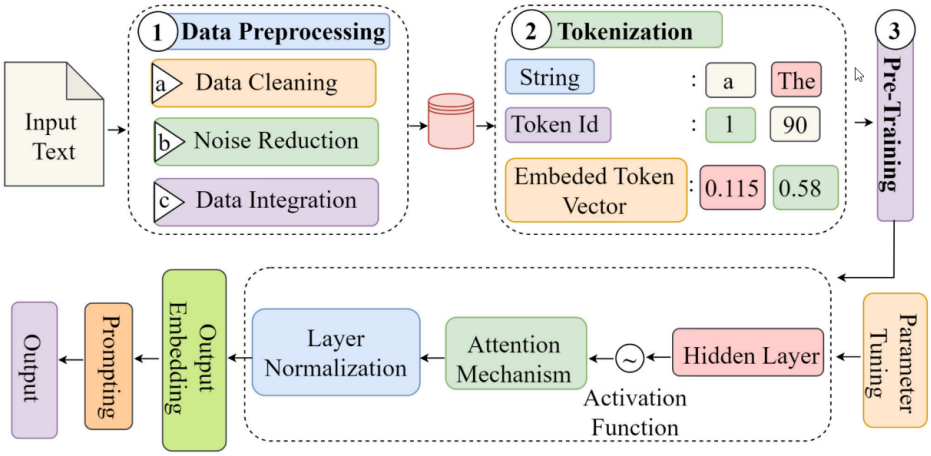


Figura 2.2: Arquitectura de los LLM. Fuente: [19].

El texto de entrada se divide en unidades llamadas tokens (palabras o fragmentos), que se transforman en vectores densos (embeddings). Se añade información posicional para que el modelo interprete el orden de las palabras en la secuencia. Una vez representados como vectores, el Transformer calcula la atención de cada token respecto a los demás, evaluando su relación y contexto.

Los Transformers se estructuran en bloques modulares de encoder y decoder, cuya configuración específica determina las capacidades y aplicaciones del modelo resultante:

- **Encoder-only:** Solo se emplea la codificación del texto de entrada, obteniendo representaciones contextuales, pero sin capacidad generativa.
- **Decoder-only:** Operan de forma autoregresiva y están optimizados para generar texto secuencialmente, prediciendo cada token en función del anterior.

- **Encoder-Decoder:** Ambos bloques trabajan de forma conjunta: el encoder transforma la entrada en una representación intermedia, que luego el decoder utiliza para generar una salida de longitud variable.

La estrategia de entrenamiento generalmente consiste en dos fases: primero, los modelos se preentrenan en grandes corpus de texto; luego, pueden adaptarse a tareas específicas con conjuntos de datos más pequeños, mejorando así su desempeño en dominios concretos.

2.3.1. Multimodalidad

Los grandes modelos de lenguaje multimodales (MLLM) expanden las capacidades de los modelos tradicionales de lenguaje, los cuales están enfocados en procesar y generar texto. Esto permite que los modelos puedan comprender y generar múltiples tipos de datos, lo que incluye texto, audio, imágenes y vídeos.

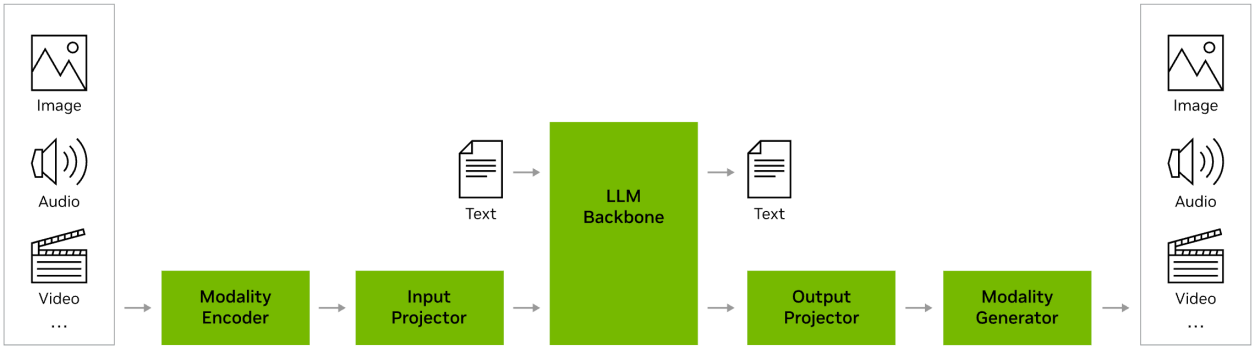


Figura 2.3: Arquitectura general de un MLLM. Los datos multimodales (texto, imágenes, audio, vídeo) se transforman mediante encoders y proyectores en embeddings que son procesados por el **LLM Backbone**, el cual actúa como núcleo central del modelo, para luego generar salidas en distintas modalidades. Fuente: NVIDIA, Multimodal Large Language Models.

Como los MLLM pueden procesar múltiples modalidades, es necesario integrar estas modalidades en una representación común. Para ello, se utilizan encoders específicos para cada tipo de entrada (texto, imágenes, audio, etc.), que transforman los datos en embeddings dentro de un espacio vectorial compartido (Figura 2.3). Posteriormente, los embeddings se combinan en un espacio conjunto, permitiendo la conversión de una modalidad a otra.

2.3.2. Aumento de contexto

Los LLM se entrenan utilizando grandes volúmenes de datos, provenientes en su mayoría de fuentes abiertas como internet. No obstante, estos modelos no suelen tener acceso a información específica o confidencial, como la documentación interna de una organización. Por ello, cuando se requiere adaptar modelos de lenguaje para casos de uso particulares, es necesario ajustar su funcionamiento con el fin de obtener los resultados y comportamientos deseados. Existen diferentes técnicas para optimizar las salidas de los LLM:

- **Prompt Engineering**: El objetivo principal de la ingeniería de prompts es elaborar instrucciones que permitan que las respuestas del modelo cumplan con los requisitos específicos del caso de uso previsto. Cabe destacar que un diseño deficiente de los prompts no puede ser compensado mediante entrenamiento adicional o mayor acceso a datos; por lo tanto, la calidad del prompt es determinante para el desempeño del modelo.
- **Finetuning**: Proceso mediante el cual un modelo previamente entrenado se reentrena utilizando un conjunto de datos más pequeño y específico, con el objetivo de dotarlo de conocimientos particulares de un dominio. Durante este procedimiento, el modelo ajusta sus parámetros y sus embeddings para adaptarse de manera más precisa al conjunto de datos seleccionado.
- **Retrieval Augmented Generation (RAG)** Método que incorpora información proveniente de fuentes externas durante el proceso de generación de respuestas [20].

  El uso de RAG permite integrar fuentes externas de información sin necesidad de reentrenar el modelo, lo que reduce costos y tiempo de desarrollo. Además, evita modificar el conocimiento general del modelo, centrándose en complementar sus respuestas, en lugar de alterar su funcionamiento interno. Esta estrategia resulta útil en escenarios donde el dominio es dinámico o está sujeto a cambios.

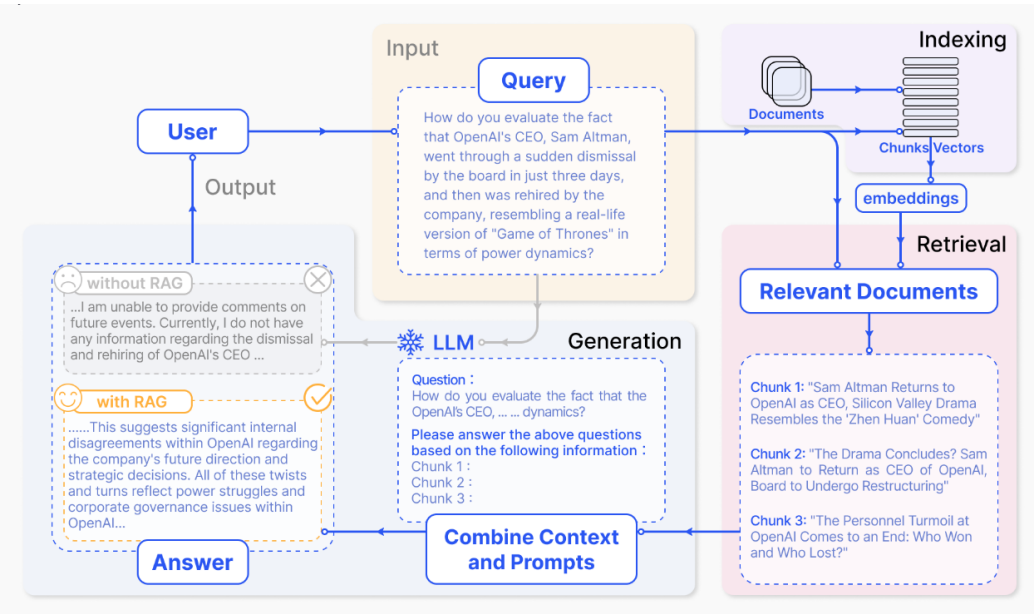


Figura 2.4: Arquitectura de RAG. Fuente: [20]

La arquitectura de RAG, como se muestra en la Figura 2.4, se divide principalmente en 3 etapas:

- **Indexación y almacenamiento**

  Los documentos de referencia se dividen en fragmentos (chunks), que luego son procesados y convertidos en vectores mediante modelos de embeddings. Estos vectores se almacenan en una base de datos especializada (vector database) para facilitar la búsqueda eficiente por similitud semántica.

- **Recuperación de información relevante**

  Cuando el usuario realiza una consulta, esta se transforma también en un vector y el sistema busca en la base de datos los fragmentos más relevantes o similares a la consulta. Estos fragmentos recuperados contienen el contexto actualizado o especializado necesario para enriquecer la respuesta.

- **Generación aumentada**

  Finalmente, tanto la pregunta original como los fragmentos recuperados se integran y se envían como entrada al modelo de lenguaje, el cual genera una respuesta informada y con respaldo documental, superando las limitaciones de conocimiento del modelo original.

Capítulo 3. Revisión bibliográfica

Las soluciones tradicionales para analizar el cumplimiento normativo en materia de seguridad laboral se han basado principalmente en técnicas de visión artificial, destacando modelos como YOLO por su eficiencia y precisión en tiempo real.

En este contexto, en Mohona et al. [21] se propone un framework de detección de EPP utilizando YOLOv8, atendiendo tanto al conjunto de datos como a la precisión y eficiencia del modelo. Se enfoca principalmente en la generación y preparación del dataset CHV (Color Helmet & Vest), el entrenamiento del modelo YOLOv8 y la evaluación comparativa entre variantes de YOLOv8. Todos los modelos presentan mayor precisión en relación con otros sistemas contemporáneos.

En los últimos años, se han explorado enfoques más avanzados que integran modelos multimodales para el análisis de escenarios complejos. Por ejemplo, Tami et al. [22] proponen un sistema que utiliza modelos multimodales como Gemini-Pro-Vision y LLaVA (Large Language and Vision Assistant), una arquitectura capaz de procesar y entender instrucciones tanto visuales como textuales, para el análisis de videos de conducción. El proceso metodológico inicia con la extracción de fotogramas desde las cámaras de los vehículos, los cuales son sometidos primero a un análisis multimodal orientado a la identificación de riesgos potenciales. Posteriormente, los fotogramas y la información textual extraída se utilizan para realizar una clasificación adicional mediante preguntas estructuradas del tipo *¿Qué?, ¿Cuál? y ¿Dónde?*. Estas entradas se combinan y se envían a los modelos multimodales a través de prompts diseñados para enfocar la respuesta en la detección de eventos críticos.

En Chen et al. [13] se presenta **Clip2Safety**, un framework que integra detección de objetos y modelos de lenguaje visual-textual (Vision-Language Models, VLMs), como CLIP, para evaluar el uso correcto de Equipos de Protección Personal en entornos laborales diversos. El sistema opera en cuatro módulos principales:

1. **Reconocimiento del escenario**: identifica el tipo de entorno (por ejemplo, fábrica química, hospital o sitio de construcción) para determinar qué EPP es obligatorio en cada caso.
2. **Generación de visual prompts**: crea las “prompts” visuales específicas según el escenario detectado, dirigiendo la atención del modelo hacia los elementos relevantes.

3. **Detección de objetos**: emplea un modelo de detección (ej. YOLO) para localizar personas y sus EPP en la escena.

4. **Verificación de atributos fina**: extrae las regiones de imagen correspondientes a las personas y genera embeddings visuales usando CLIP. Estos embeddings se comparan con embeddings textuales derivados de las prompts que describen los EPP requeridos, disciplinando el cumplimiento de cada atributo (material, forma, uso correcto, etc.)

Clip2Safety fue validado en seis escenarios reales, logrando una precisión superior a la de los modelos VLM basados en pregunta-respuesta.

Recientemente, en Chen et al. [23] se postula ChatCH, un framework que combina un sistema de identificación de peligros en construcción basado en modelos visión-lenguaje (VLM) con finetuning, y un método para la generación automatizada de informes de riesgos. Para validar su enfoque, desarrollaron el Construction Hazard Dataset (CHD), compuesto por 1308 imágenes reales de peligros en obra, distribuidas en 32 categorías específicas. En los experimentos, ChatCH logra precisiones superiores a los modelos preentrenados Qwen2-VL-7B y CLIP, evidenciando mejoras significativas (precisión de 89,4 %). Además, demuestra capacidades de aprendizaje con pocos ejemplos (few-shot learning), lo cual refuerza su aplicabilidad práctica.

Capítulo 4. Solución propuesta

En esta memoria de título se propone un sistema automatizado para el análisis del cumplimiento de normativas de seguridad laboral usando detección de objetos y MLLM. A diferencia de los enfoques revisados en la sección 3, este sistema no solo realiza detección visual de elementos, sino que también interpreta el contexto y evalúa el nivel de riesgo asociado a cada situación.

4.1. Centro para la Industria 4.0 (C4i)

El Centro para la Industria 4.0 (C4i)[1] es el centro tecnológico de la Universidad de Concepción que conecta la investigación académica con las necesidades de la industria, impulsando la transformación digital y tecnológica de diversos sectores productivos. Su labor facilita la adopción de tecnologías avanzadas, optimiza procesos, mejora la capacidad y aumenta la competitividad y rentabilidad de las empresas nacionales.

En el desarrollo de esta memoria de título, el C4i desempeñó un rol fundamental al entregar retroalimentación técnica y actuar como nexo entre la universidad y la empresa colaboradora responsable de validar el sistema. Asimismo, apoyó en la definición de los requerimientos y promovió la vinculación con profesionales de la industria.

4.2. Arquitectura de solución

Como se observa en la Figura 4.1, la solución propuesta se compone de tres módulos principales: **Entrada**, correspondiente a la etapa de ingreso y preprocesamiento de datos; **Procesamiento**, donde se realiza un análisis preliminar y evaluación multimodal; y **Salida**, en la cual se lleva a cabo la interpretación normativa y la generación de resultados o alertas.

[1] `https://c4i-udec.cl/`

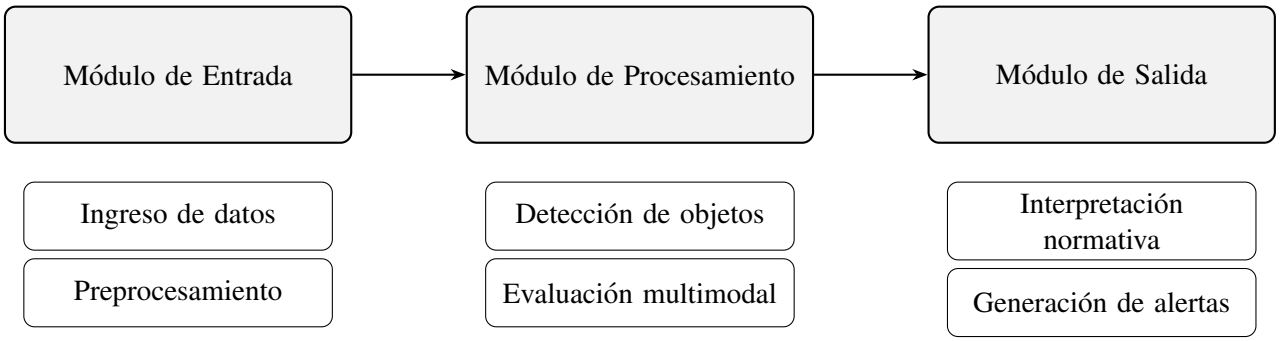


Figura 4.1: Esquema general de la arquitectura del sistema, con los principales subprocesos de cada módulo.

También se contemplan distintos escenarios de despliegue para los modelos de inteligencia artificial, ya que las condiciones operativas, los recursos disponibles y los requisitos de privacidad pueden variar según la empresa y el entorno donde se utilice el sistema:

- **Implementación local:** Permite procesar los datos de forma autónoma, sin depender de conexiones externas, lo que garantiza mayor privacidad y menores tiempos de respuesta. Esta modalidad puede incluir tanto la detección de objetos como algunos modelos de lenguaje.
- **Implementación vía API:** Indicada para modelos de lenguaje de gran tamaño cuyo procesamiento supera las capacidades locales. El uso de APIs de proveedores externos, como OpenAI o Google AI Studio, facilita el acceso a modelos de última generación, aunque implica consideraciones adicionales respecto a costos, seguridad y privacidad.
- **Implementación híbrida:** El sistema está diseñado para combinar ambas modalidades según los recursos disponibles y las necesidades del usuario. Por ejemplo, la detección inicial puede realizarse localmente y el análisis avanzado delegarse a una API externa, optimizando así el balance entre costo, privacidad y rendimiento.

Se evaluaron tanto la implementación vía API como la modalidad híbrida, seleccionando la alternativa más adecuada de acuerdo con los requisitos técnicos y operativos del entorno de despliegue.

4.2.1. Herramientas y Recursos

El desarrollo de sistemas requiere herramientas especializadas que permitan integrar distintos componentes de forma modular.

En este sistema se utilizaron diversas herramientas y recursos ampliamente adoptados en aplicaciones de inteligencia artificial, debido a su extensibilidad y amplia documentación. A continuación, se mencionan las principales herramientas empleadas.

Python

Python es un lenguaje de programación interpretado, de alto nivel y propósito general, cuya filosofía de diseño prioriza la legibilidad y la simplicidad del código. Utiliza tipado dinámico, gestión automática de memoria (garbage collection) y ofrece una extensa cantidad de librerías que cubren desde cálculo numérico y ciencia de datos hasta redes, interfaces gráficas y desarrollo web.

Python ofrece velocidad de desarrollo, una cadena de herramientas especializada para visión artificial y lenguaje, y flexibilidad de despliegue, lo cual lo hace adecuado para el sistema propuesto.

Langchain

Langchain[2] es un framework de código abierto que permite construir aplicaciones basadas en LLMs en Python y JavaScript. Está diseñado para facilitar la integración de estos modelos en diversas aplicaciones, como chatbots, análisis de documentos y otros casos de uso, mediante componentes modulares que se organizan en pipelines o "chains".

En este sistema, Langchain actúa como una capa de abstracción que unifica la interacción con distintos modelos de lenguaje, tanto locales (Ollama) como accesibles vía API (OpenAI, GoogleAI). Esto permite seleccionar y cambiar de modelo según los requisitos de rendimiento y costo, sin modificar el código principal. Además, Langchain integra motores vectoriales como FAISS y Chroma, lo que habilita búsquedas semánticas eficientes sobre grandes volúmenes de documentos.

Langsmith

Langsmith[3] es una plataforma de observabilidad y evaluación diseñada para aplicaciones basadas en LLMs. Ofrece trazabilidad, métricas, alertas y herramientas de depuración que facilitan monitorear y analizar el comportamiento de los modelos. Además, permite realizar evaluaciones estructuradas para medir cuantitativamente el rendimiento, lo que resulta clave para validar y mejorar la calidad de los resultados generados en el sistema propuesto.

Streamlit

Streamlit[4] es un framework de código abierto en Python que permite crear aplicaciones web interactivas orientadas a datos con pocas líneas de código. Facilita la integración de widgets, gráficos, tablas y

[2] `https://python.langchain.com/docs/`
[3] `https://docs.smith.langchain.com/`
[4] `https://docs.streamlit.io/`

archivos, sin necesidad de conocimientos en lenguajes de frontend como HTML, CSS o JavaScript. El script se ejecuta de arriba hacia abajo con cada interacción del usuario, permitiendo la actualización en tiempo real de la interfaz. Además, soporta aplicaciones multipágina y personalización de temas, lo que lo hace ideal para prototipos, visualizaciones exploratorias y dashboards.

En este sistema, tanto la visualización de cada etapa del framework como el dashboard se implementaron usando esta herramienta.

Ollama

Ollama[5] es una herramienta y motor LLM de código abierto que permite ejecutar localmente modelos de lenguaje como Llama3, Qwen y Gemma. Su interfaz principal es una línea de comandos con un servidor REST, que expone endpoints para chat, embeddings e inferencia multimodal. En este sistema, los modelos locales se ejecutan mediante Ollama, que agrega una capa de abstracción para la gestión y ejecución de modelos, permitiendo prescindir de servicios en la nube, reducir costos y garantizar la privacidad de los datos.

Ultralytics YOLO

Ultralytics YOLO[6] es la versión más reciente de la reconocida serie YOLO (You Only Look Once) para la detección y segmentación de objetos en tiempo real. Integra nuevas funciones y mejoras sobre versiones anteriores, incrementando el rendimiento, la flexibilidad y la eficiencia. Esta implementación admite diversas tareas de visión por computador, como detección, segmentación, estimación de pose, seguimiento y clasificación.

En este proyecto se utiliza Ultralytics YOLO para la detección de objetos, debido a su documentación extensa, su enfoque de código abierto y su activa comunidad de desarrollo.

Roboflow

Roboflow[7] es una plataforma integral que facilita la creación y despliegue de modelos de visión por computador. Ofrece herramientas para el etiquetado, procesamiento y gestión de datos, así como para el entrenamiento y despliegue de modelos.

Se utilizó Roboflow para el etiquetado de imágenes, aprovechando sus funciones asistidas por inteligencia artificial, lo que permitió realizar el proceso de manera más rápida y eficiente en comparación

[5] `https://ollama.com/`
[6] `https://docs.ultralytics.com/`
[7] `https://universe.roboflow.com/`

con métodos manuales.

4.2.2. Restricciones

Debido a la naturaleza de la solución propuesta, la principal restricción de este sistema es de tipo técnico, ya que la calidad de los resultados depende directamente del desempeño y las capacidades de los modelos empleados.

Técnicas El entrenamiento de modelos de detección como YOLO requiere de imágenes de calidad y bien anotadas. La falta de datos representativos del entorno industrial puede limitar los resultados alcanzables. Asimismo, las versiones locales de LLM y las variantes ligeras de YOLO cuentan con menos parámetros que los modelos avanzados disponibles vía API. Aunque los modelos de mayor tamaño pueden ofrecer mayor precisión y capacidades visuales superiores, su uso implica altos requisitos computacionales y, en el caso de los modelos accesibles mediante API, costos adicionales asociados.

Gracias al grupo de *IA & Ciberseguridad* de la Universidad de Concepción, fue posible acceder a dos GPU RTX 4090 para el entrenamiento y uso de modelos YOLO, así como para la ejecución de modelos multimodales. Sin embargo, este recurso no es suficiente para ejecutar modelos de gran escala, lo que limita la escalabilidad del sistema en escenarios industriales más complejos. En este sentido, se proyecta la adopción de técnicas de optimización de modelos (cuantización, pruning, distillation) y estrategias de despliegue en *edge computing*, como implementación local, que permitan mantener un balance adecuado entre rendimiento, costo y portabilidad.

Temporales El sistema debe ser desarrollado, validado y documentado durante el primer semestre de 2025, lo que restringe tanto la profundidad de las pruebas en entornos reales como la extensión del sistema final. Este marco temporal condiciona la posibilidad de realizar estudios de confiabilidad y escalamiento. Como proyección, se recomienda extender las validaciones hacia implementaciones piloto en distintos entornos industriales, lo que permitirá robustecer el sistema y generar evidencia sobre su aplicabilidad práctica a gran escala.

4.3. Arquitectura del prototipo

El sistema procesa videos de cámaras de seguridad, seleccionando automáticamente los frames más representativos. Luego, utiliza un modelo YOLO para detectar equipos de protección personal y per-

sonas, y utiliza MLLM junto con un sistema RAG para evaluar las condiciones de seguridad en base a documentos normativos.

Como se observa en la Figura 4.2, el sistema integra estos módulos en un flujo automatizado que va desde la captura y análisis de imágenes hasta la generación de alertas y reportes detallados, combinando visión computacional y LLM.

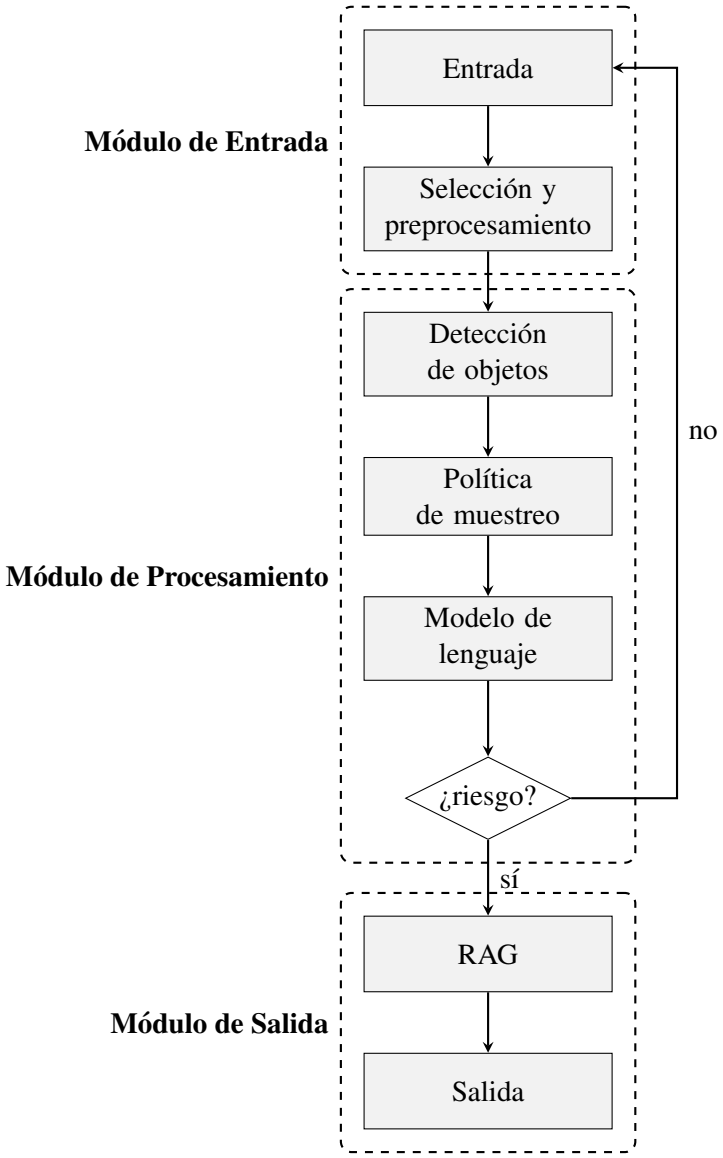


Figura 4.2: Flujo del sistema propuesto.

4.3.1. Entrada

El punto de partida del sistema es el ingreso y preprocesamiento de los datos. Esta etapa es fundamental, ya que determina la calidad y representatividad de la información sobre la cual operarán los

módulos posteriores. A continuación, se describen los mecanismos implementados.

Entrada de datos

El sistema inicia su funcionamiento a partir de la entrada de video, la cual puede corresponder tanto a una grabación preexistente como a una transmisión en tiempo real proveniente de las cámaras de seguridad. Para optimizar el procesamiento, se implementa un muestreo temporal que consiste en extraer un cuadro por segundo (**1 fps**) del video de entrada. Esto responde a la necesidad de asegurar una cobertura de los eventos relevantes. De este modo, es posible analizar la escena sin perder información crítica, ya que se capturan los momentos más relevantes. Al mismo tiempo, mejora la eficiencia al evitar la sobrecarga que implicaría procesar todos los cuadros por segundo.

Los análisis se realizan en batch o lotes. Un batch corresponde a un conjunto de K frames tomados en instantes de tiempo sucesivos o muestreados a intervalos regulares. Formalmente, un batch de tamaño K que inicia en el instante t_i se define como:

$$\text{Batch}_i = \{f_{t_i}, f_{t_{i+1}}, f_{t_{i+2}}, \ldots, f_{t_{i+K-1}}\} \tag{4.1}$$

donde f_{t_j} representa el frame capturado en el instante t_j.

Optical Flow Técnica de visión computacional utilizada para estimar el movimiento aparente de objetos, superficies y bordes en una secuencia de imágenes. Matemáticamente, se define como el campo de velocidades de desplazamiento de los píxeles entre dos cuadros. En la práctica, el cálculo entrega un valor numérico que representa la magnitud del movimiento detectado entre estos dos cuadros, como se muestra en la Figura 4.3. Si este valor es cercano a cero, se interpreta que no existe movimiento relevante en la escena. Por el contrario, cuando el valor supera un umbral previamente definido, el sistema determina que existe actividad significativa. Esta lógica permite filtrar los segmentos del video que presentan cambios importantes, reduciendo la cantidad de cuadros a analizar en detalle.

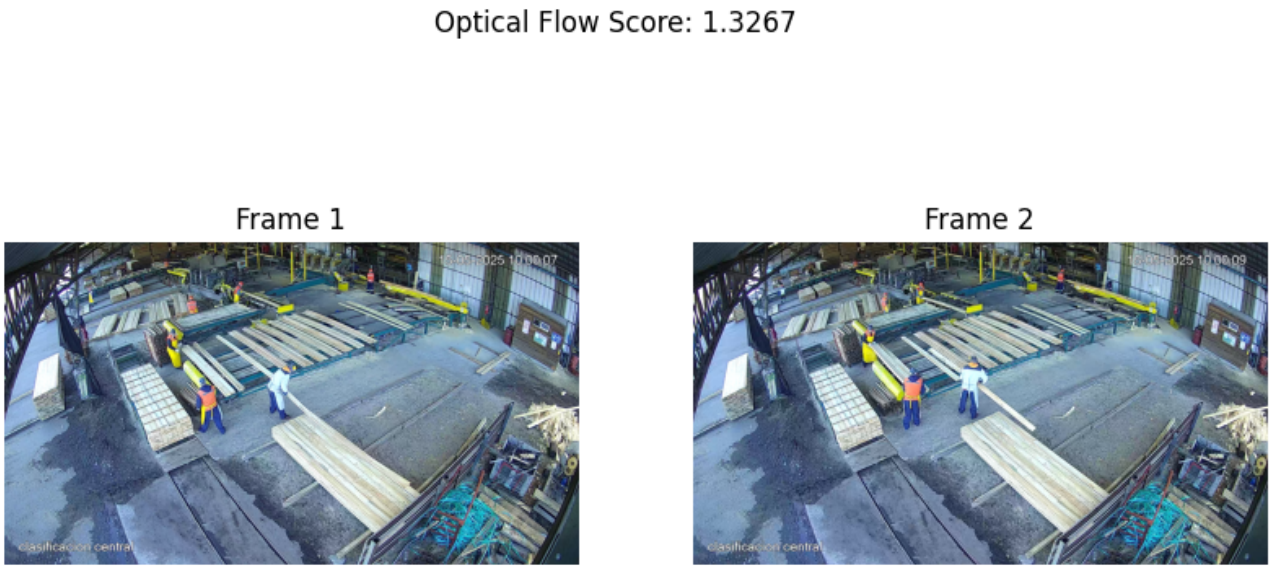


Figura 4.3: Cálculo de optical flow entre dos cuadros.

Selección y Procesamiento

Una vez completado un batch y aplicada la técnica de optical flow, se procede a seleccionar un subconjunto representativo de estos cuadros usando uno de los siguientes métodos:

Structural Similarity Index (SSIM) Métrica que permite comparar el grado de similitud entre dos imágenes. Evalúa la similitud considerando tres componentes: luminosidad, contraste y estructura de las imágenes.

El proceso consiste en comparar secuencialmente los frames del batch y seleccionar como "representativos" aquellos cuya similitud estructural con el último frame seleccionado sea inferior a un umbral definido. Esto asegura que los cuadros elegidos muestren diferencias visuales significativas.

Formalmente, dado un umbral τ, un frame f_j se selecciona si:

$$\text{SSIM}(f_j, f_{j-1}) \leq \tau \tag{4.2}$$

Contrastive Language-Image Pretraining (CLIP) Utiliza redes neuronales entrenadas para mapear imágenes y textos a un espacio semántico compartido. En este sistema, se emplea el modelo CLIP para extraer "embeddings" o representaciones vectoriales de cada cuadro dentro del batch, capturando información visual relevante y abstracta más allá de la similitud de píxeles.

El proceso de selección se basa en aplicar técnicas de agrupamiento (clustering) sobre los embeddings

obtenidos. Luego, se identifica el cuadro más representativo ("medoide") de cada grupo, formando así un subconjunto que captura las distintas situaciones presentes en el batch analizado.

La elección entre utilizar ***SSIM o CLIP*** como método de selección depende de los recursos computacionales disponibles y de los requisitos del análisis. CLIP está optimizado para ejecutarse en GPU y puede aprovechar la aceleración gráfica para procesar grandes volúmenes de datos, además de ofrecer una mayor capacidad de análisis semántico. Por otro lado, SSIM es una métrica más sencilla, no requiere GPU y puede ejecutarse de forma eficiente en CPU, siendo adecuado para sistemas con recursos limitados o cuando se busca rapidez en el procesamiento.

4.3.2. Procesamiento

En esta etapa, el sistema transforma los cuadros representativos extraídos de los videos en información útil para la evaluación. El procesamiento abarca desde la detección de objetos relevantes, pasando por la aplicación de políticas de muestreo para priorizar los casos más críticos, hasta el análisis contextual mediante modelos de lenguaje.

Detección de objetos

Con los frames representativos seleccionados, el sistema recurre a un modelo de detección de objetos basado en **YOLOv11** para identificar objetos relevantes. Esta elección se justifica por la naturaleza de YOLO como algoritmo de una sola pasada, capaz de realizar inferencia de manera rápida y eficiente en comparación con otros modelos que requieren mayor procesamiento, como los LLMs.

En particular, se emplea un modelo entrenado a partir de la versión `yolo11x.pt`, diseñado y afinado específicamente para detectar personas y equipos de protección personal. Las clases objetivo incluyen: `person`, `hardhat`, `safety-vest`, `safety-goggles`, `safety-gloves`, `earmuffs` y `phone`. El dataset utilizado consta de 3,412 imágenes para entrenamiento, etiquetadas manualmente, 155 para validación y 136 para pruebas. El entrenamiento del modelo incluyó técnicas de aumento de datos para mejorar su capacidad de generalización frente a condiciones variables de las cámaras.

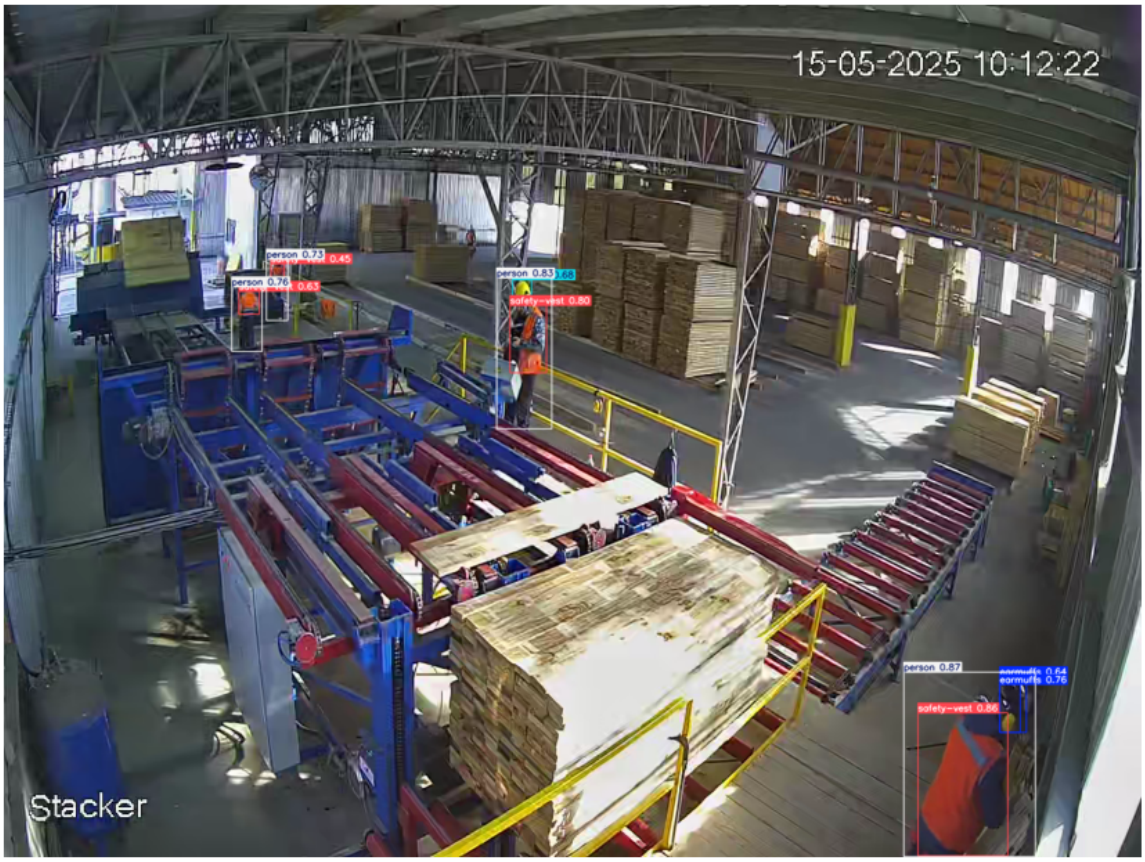


Figura 4.4: Detección de objetos con cajas delimitadoras, etiquetas de clase y nivel de confianza.

La ejecución del modelo sobre cada frame genera cajas delimitadoras (bounding boxes) y las etiquetas de clase correspondientes para cada objeto detectado, como se ejemplifica en la Figura 4.4. Estas detecciones se usan posteriormente para evaluar el nivel de cumplimiento de las normativas, generar alertas o alimentar modelos de análisis posteriores.

Política de muestreo

La política de muestreo corresponde al conjunto de reglas empleadas para decidir qué cuadros y situaciones deben ser analizados en profundidad por el sistema. Estas políticas pueden basarse en distintos criterios, tales como la relación entre la cantidad de personas detectadas y la cantidad de elementos de protección personal identificados en cada frame, o la detección de personas cruzando zonas definidas como restringidas.

Por ejemplo, una política consiste en comparar el número de personas presentes en una escena con la cantidad de cascos o chalecos detectados; si el sistema identifica una discrepancia, el lote de imágenes asociado es marcado para un análisis posterior más detallado mediante modelos de lenguaje.

Sin embargo, la definición de reglas de muestreo requiere un conocimiento de dominio específico para cada actividad, cuya formalización excede los plazos y el alcance de este trabajo. Por esta razón, y con el objetivo de centrar el desarrollo en la capacidad de interpretación contextual de los modelos de lenguaje, se opta por no implementar esta capa de filtrado preliminar.

El diseño del sistema, no obstante, mantiene la flexibilidad para que estas políticas de muestreo puedan ser incorporadas a futuro.

Modelo de Lenguaje

El sistema incorpora una arquitectura de doble nivel para el análisis mediante modelos de lenguaje, implementando un modelo "ligero" para evaluación inicial y un modelo de mayor capacidad para análisis detallado. Esta estructura busca reducir costos computacionales y latencia en el procesamiento, sin sacrificar la calidad del análisis.

El primer nivel utiliza un modelo orientado a verificar de manera preliminar si existe algún riesgo o situación que amerite una revisión más exhaustiva. Este modelo realiza una evaluación rápida sobre cada batch de frames representativos, generando un análisis que incluye:

- Descripción breve del entorno, condiciones observadas y elementos de protección personal.
- Evaluación binaria ("Sí" o "No") del uso correcto de elementos de protección personal.
- Evaluación binaria ("Sí" o "No") de la presencia de condiciones de riesgo.
- Puntuación de seguridad, valor numérico entre 0 y 100 (donde 0 indica situación peligrosa y 100 una condición completamente segura).

En la propuesta inicial, se abordó la protección de los datos personales, considerando el envío de imágenes a proveedores externos de LLM.

En el desarrollo de este trabajo se omitió dicho análisis en detalle, dado que la disposición y el ángulo de las cámaras utilizadas no permiten la identificación individual de los trabajadores.

Sin embargo, resulta fundamental considerar escenarios futuros donde sí exista riesgo de identificación personal. En estos casos, el uso de sistemas de monitoreo conlleva implicancias éticas y legales vinculadas a la privacidad, la transparencia en el uso de datos y la percepción de vigilancia en el entorno laboral.

Como medidas de mitigación, se proyecta la incorporación de técnicas de anonimización (difuminado de rostros, enmascaramiento de zonas sensibles) y almacenamiento seguro en servidores locales con acceso restringido.

Prompt

```
Las imágenes fueron capturadas en un aserradero durante operaciones normales.

Analiza {num_images} imágenes y devuelve **solo** un JSON con las claves del esquema.
- "batch_description": 1-2 oraciones que resuman entorno, EPP visible y riesgos.
- "overall_ppe_compliance": "Sí" o "No".
- "overall_dangerous_conditions": "Sí" o "No".
- "safety_score": número entre 0 y 100 (100 = totalmente seguro).

Contexto adicional: {custom_context}
Considera el contexto adicional en la descripción.

### Ejemplo válido
{{
    "batch_description": "Un operario utiliza una grúa horquilla para trasladar tablones;
    ↪  no lleva guantes ni antiparras y hay serrín y cables sueltos en el suelo.",
    "overall_ppe_compliance": "No",
    "overall_dangerous_conditions": "Sí",
    "safety_score": 42
}}

### Ejemplo inválido
{{
    "batch_description": "Análisis de seguridad en un aserradero."
}}
```

Salidas Estructuradas

Como este primer análisis funciona como filtro para la siguiente etapa, es necesario que los modelos generen una salida consistente y predecible. Esto se logra utilizando el concepto de salida estructurada (Structured Output), donde se instruye al modelo para responder con una estructura específica.

LangChain automatiza la vinculación entre el esquema y el modelo, además de procesar y validar la salida, facilitando la integración con distintos proveedores que soportan esta funcionalidad.

```
properties = {
    "batch_description": {
        "type": "string",
        "description": (
            "Descripción breve (1-2 oraciones) centrada en seguridad laboral: "
```

```
            "entorno, EPP observado y condiciones peligrosas."
        ),
        "maxLength": 500,
    },
    "overall_ppe_compliance": {
        "type": "string",
        "enum": ["Sí", "No"],
        "description": "Cumplimiento general de EPP en todas las imágenes.",
    },
    "overall_dangerous_conditions": {
        "type": "string",
        "enum": ["Sí", "No"],
        "description": "Presencia de condiciones peligrosas en las imágenes.",
    },
    "safety_score": {
        "type": "number",
        "description": "Puntuación global de seguridad (0-100).",
        "minimum": 0,
        "maximum": 100,
    },
}
```

Escalamiento

El sistema implementa una lógica de escalamiento basada en los resultados de este modelo. Solo cuando se detectan condiciones críticas se activa el análisis detallado:

```
if (
    llm_result.get("overall_ppe_compliance") == "No"
    or llm_result.get("overall_dangerous_conditions") == "Sí"
):
    should_run_rag = True
```

Fatiga de eventos

El sistema implementa un mecanismo de deduplicación que previene la saturación de notificaciones repetitivas. Esta combina análisis temporal y semántico para determinar si una nueva alerta debe procesarse: mantiene una ventana deslizante con las alertas recientes y calcula la similitud semántica mediante la similitud coseno con un umbral predefinido. Para implementar este mecanismo, se utiliza una variable numérica denominada `safety_score`, la cual es generada por el modelo de lenguaje

a partir de su interpretación de la imagen o lote de imágenes. Este valor, que varía entre 0 y 100, cuantifica el nivel de seguridad observado en la escena.

Las alertas críticas con `safety_score` ≤ 30, siempre se procesan independientemente de su similitud.

Sin este mecanismo, situaciones que persisten en el tiempo (como un trabajador sin casco durante varios minutos) generarían múltiples alertas idénticas, reduciendo la efectividad del sistema.

Modelos

Entre los modelos evaluados y utilizados se encuentran: `gpt4.1-mini`, `qwen2.5vl-32B` y `gemini-2.5-flash`. De estos, `qwen2.5vl-32B` corresponde a un modelo local, ejecutado mediante Ollama, mientras que `gpt4.1-mini` y `gemini-2.5-flash-lite` son modelos comerciales accesibles mediante API. La elección de estos modelos responde a su capacidad para procesar entradas multimodales, generar salidas estructuradas y ofrecer un equilibrio adecuado entre capacidades y costo (considerando el valor en USD por millón de tokens para los modelos comerciales, y la factibilidad de ejecución en la VRAM disponible para el modelo local).

4.3.3. Salida

La etapa de salida corresponde a la generación y presentación de los resultados obtenidos a partir del análisis de los cuadros seleccionados. En este punto, el sistema produce información estructurada y reportes detallados que facilitan la interpretación y toma de decisiones.

RAG

El sistema RAG constituye el segundo nivel de análisis, enfocado en la recuperación de información específica. A diferencia del primer nivel, que se apoya únicamente en el conocimiento preentrenado del modelo junto con un contexto específico, esta etapa fundamenta sus evaluaciones en documentación técnica. Esto permite generar análisis más precisos y respaldados normativamente.

Este nivel está formado por una base de conocimiento vectorizada a partir de documentos técnicos, normativas y manuales. El proceso comienza dividiendo los documentos en “chunks” o fragmentos de tamaño determinado; luego, cada fragmento se transforma en una representación numérica (vector) mediante técnicas de procesamiento de lenguaje natural.

Para la vectorización se utiliza el modelo `text-embedding-3-small` de OpenAI, seleccionado por su equilibrio entre calidad de representación semántica y eficiencia. Los embeddings generados con este

modelo se indexan utilizando FAISS (Facebook AI Similarity Search)[8], una librería especializada en búsquedas de similitud sobre vectores. FAISS destaca por su capacidad para manejar grandes volúmenes de datos y ofrecer altas velocidades de consulta, especialmente al aprovechar la aceleración por GPU.

Recuperación

El proceso de recuperación comienza utilizando la descripción del batch generada por el modelo ligero como consulta semántica. El sistema recupera los `TOP_K_RETRIEVAL` fragmentos más relevantes de la base vectorial.

El pipeline de recuperación y generación integra múltiples fuentes de información: el contexto documental recuperado, las imágenes del batch analizado y las clasificaciones generadas por YOLO, permitiendo así respuestas fundamentadas y específicas para cada caso.

Prompt

```
Imagen {i + 1} objetos: {yolo_list}

Eres un experto en seguridad ocupacional especializado en análisis de imágenes de
↪  ambientes laborales, particularmente aserraderos. Analiza la imagen usando el
↪  contexto documental proporcionado.

CONTEXTO DOCUMENTAL:
{contexto}

INSTRUCCIONES DETALLADAS:
Analiza la imagen capturada en ambiente de aserradero y proporciona un análisis
↪  exhaustivo sobre:

1. **ENTORNO LABORAL**:
   - Descripción específica del área de trabajo (ubicación, iluminación, maquinaria
   ↪  cercana)
   - Condiciones ambientales observables (orden, limpieza, espacios de trabajo)
   - Estado de las instalaciones y equipos visibles

2. **EQUIPO DE PROTECCIÓN PERSONAL (EPP)**:
   - Identifica específicamente qué EPP está presente y su estado
   - Determina qué EPP obligatorio está ausente
```

[8] https://github.com/facebookresearch/faiss

```
   - Evalúa la calidad y adecuación del EPP observado

3. **RIESGOS Y CONDICIONES INSEGURAS**:
   - Identifica condiciones peligrosas específicas (derrames, obstrucciones, superficies
    ↪ resbaladizas)
   - Detecta actos inseguros o comportamientos riesgosos
   - Evalúa riesgos de maquinaria, herramientas o equipos
   - Identifica bloqueos de salidas de emergencia o rutas de evacuación

4. **CUMPLIMIENTO NORMATIVO**:
   - Evalúa el cumplimiento con normativas de seguridad laboral
   - Considera estándares específicos para industria maderera
   - Identifica violaciones a protocolos de seguridad

5. **ACCIONES CORRECTIVAS**:
   - Prioriza máximo 3 acciones correctivas inmediatas
   - Enfócate en los riesgos más críticos identificados

SEMÁFORO DE SEGURIDAD:
- **Verde**: Cumple con todos los estándares, sin riesgos críticos
- **Amarillo**: Observaciones menores, EPP insuficiente, riesgos controlables
- **Rojo**: Incumplimientos graves, riesgos críticos inmediatos

IMPORTANTE:
- Sé específico y factual en tu análisis
- Solo menciona lo que es claramente observable en la imagen
- Si no puedes determinar algo con certeza, indícalo claramente
- Enfócate en aspectos críticos de seguridad
- Responde en español
- No inventes información que no sea visible
- El campo 'details' debe ser un resumen ejecutivo que integre todos los hallazgos
 ↪ principales

CONTEXTO ADICIONAL: {contexto_adicional}
```

Al igual que en el primer nivel, el modelo utiliza un esquema definido que garantiza la consistencia y estructuración de la información. Este esquema incluye:

- **Semáforo**: Asigna automáticamente un código cromático (verde, amarillo, rojo) que refleja la severidad de la situación detectada, basado tanto en la evaluación visual como en el cumplimiento de normativas recuperadas del contexto documental.
- **Elementos de protección personal**: Identifica y evalúa los equipos de protección presentes y ausentes, fundamentando el análisis en los requisitos específicos para la actividad y el entorno,

de acuerdo con la documentación.

- **Riesgos**: Lista y describe las condiciones inseguras detectadas, estableciendo su correspondencia con las normativa.
- **Evaluación ambiental**: Analiza las condiciones del entorno de trabajo que pueden influir en la seguridad, vinculando estas observaciones con la documentación.
- **Cumplimiento normativo**: Verifica la adherencia a las normativas recuperadas, proporcionando una fundamentación legal y técnica para las evaluaciones emitidas.
- **Acciones correctivas**: Propone acciones correctivas específicas basadas en el análisis de las imágenes.

Salida

Los resultados se presentan a través de un dashboard desarrollado con Streamlit, que permite el monitoreo en tiempo real y el análisis histórico de alertas de seguridad.

El panel principal muestra métricas organizadas en cuatro columnas, incluyendo el total de alertas y su distribución por criticidad. Las visualizaciones están implementadas con **Plotly**, una biblioteca interactiva para Python que permite crear distintos tipos de gráficos, facilitando la identificación de patrones y tendencias mediante gráficos de barras, diagramas y líneas de tiempo (Figura 4.5).

El historial de alertas se presenta mediante una tabla interactiva basada en **AgGrid**, un componente utilizado para la visualización y gestión de datos tabulares. Esta tabla soporta paginación, ordenamiento, filtrado y selección individual de cada alerta. Al seleccionar una, se despliega información detallada como el timestamp, la clasificación según el semáforo, las imágenes asociadas y la descripción correspondiente (Figura 4.6).

Todas las alertas se almacenan persistentemente en una base de datos SQLite, lo que permite su posterior consulta y análisis.

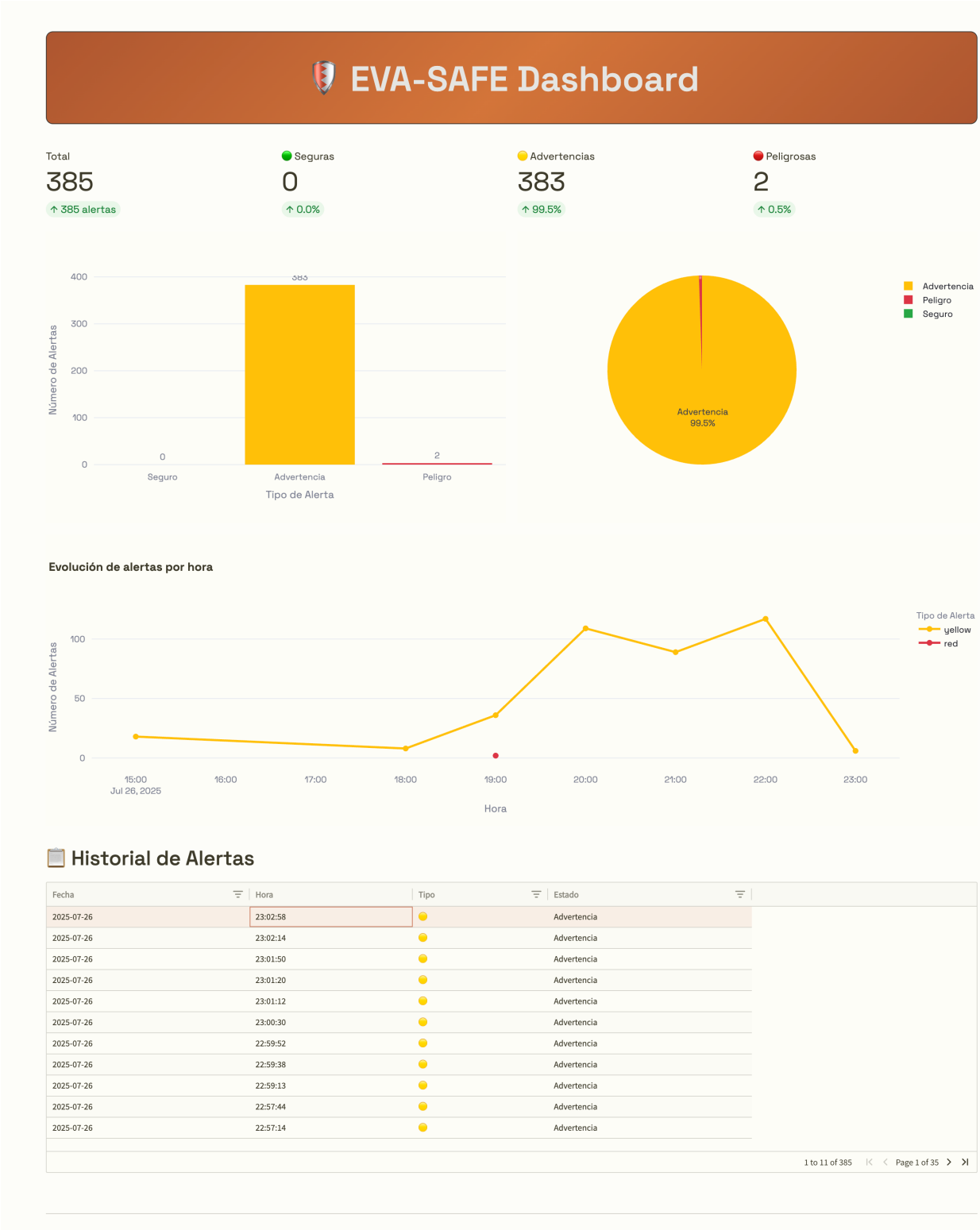
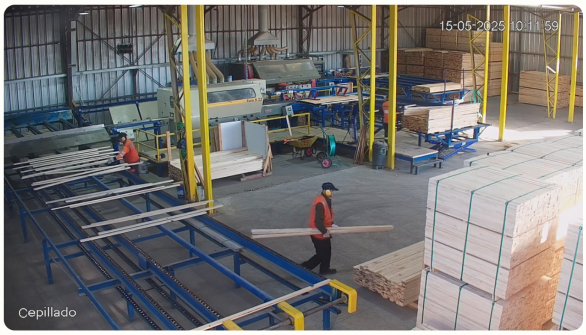


Figura 4.5: Panel principal.

Fecha y Hora: 2025-07-26 23:02:58

Estado: Advertencia



Detalles del Análisis

El semáforo de seguridad se encuentra en amarillo debido a la ausencia de guantes y gafas de seguridad en los trabajadores que manipulan madera, incumpliendo normativas y recomendaciones de la HDS, lo que representa riesgos de cortes, astillas y lesiones oculares. El entorno es ordenado y bien iluminado, con maquinaria y materiales correctamente dispuestos, aunque algunos pasillos podrían estar parcialmente obstruidos. Se observa cumplimiento en el uso de chaleco reflectante y protección auditiva, pero es crítico corregir de inmediato la falta de EPP obligatorio para garantizar la seguridad y el cumplimiento normativo. Las acciones prioritarias son exigir el uso de guantes y gafas, reforzar la capacitación en riesgos y mantener despejadas las rutas de evacuación.

Elementos de Protección Personal

Detectado: chaleco reflectante, protector auditivo tipo orejera **Faltante:** guantes de protección, gafas de seguridad

Riesgos Identificados

- Manipulación manual de madera sin guantes (riesgo de cortes, astillas)
- Ausencia de gafas de seguridad (riesgo de lesiones oculares por partículas)
- Presencia de maquinaria en movimiento (riesgo de atrapamiento)
- Posible obstrucción parcial de pasillos por materiales apilados

Cumplimiento Normativo

- Cumplimiento parcial de EPP: uso de chaleco reflectante y protección auditiva adecuados para el entorno ruidoso.
- Incumplimiento en uso de guantes y gafas de seguridad, ambos obligatorios según normativa y HDS para manipulación de madera.
- No se observan bloqueos graves de rutas de evacuación, pero se recomienda revisión periódica.

Condiciones Ambientales

Área de trabajo amplia, bien iluminada y aparentemente ordenada. No se observan derrames ni suciedad significativa. Maquinaria y materiales apilados correctamente, aunque algunos pasillos podrían estar parcialmente obstruidos.

Acciones Recomendadas

- Exigir uso inmediato de guantes y gafas de seguridad a todo el personal que manipule madera.
- Reforzar capacitación sobre riesgos de manipulación manual y uso correcto de EPP.
- Verificar y mantener libres los pasillos y rutas de evacuación de materiales apilados.

Análisis generado automáticamente por EVA-SAFE

Ver datos técnicos detallados

Figura 4.6: Información detallada de cada alerta.

4.4. Condiciones de operación y gestión del cambio

A continuación se describe los requisitos necesarios para que el sistema propuesto funcione de manera segura, eficiente y sostenible dentro de la organización. Se abordan cuatro dimensiones clave: *infraestructura técnica*, *implantación y gestión del cambio*, *usabilidad y capacitación*, y *mantención de la operación*.

4.4.1. Infraestructura y dependencias técnicas

Hardware mínimo Para un despliegue **local** capaz de ejecutar los modelos de detección y, opcionalmente, un modelo multimodal, se requiere al menos 24GB de VRAM (NVIDIA RTX 4090 o equivalente). Para organizaciones con recursos limitados, se sugiere optar por una arquitectura híbrida.

Software y dependencias

- **Sistema operativo**: Ubuntu 22.04 LTS o superior.
- **Librerías principales**: `Python` $\geq$`3.11`, `PyTorch` $\geq$`2.3`, `ultralytics`, `streamlit`, `faiss`, `langchain`, `langsmith` y `ollama` para la ejecución local de LLM.
- **Servicios externos**: Credenciales para OpenAI API o Google AI Studio.

4.4.2. Implantación y gestión del cambio

La introducción de sistemas de monitoreo automatizado afecta tanto a los procesos operativos como a la cultura organizacional. Se proponen las siguientes actividades de gestión del cambio:

1. **Plan de comunicación**: Difundir objetivos, alcance y beneficios; enfatizar que el sistema es una *herramienta preventiva* y no un mecanismo punitivo.
2. **Piloto controlado**: Implementar un entorno de prueba en un área acotada para ajustar parámetros del sistema y flujos de reporte.
3. **Políticas internas:** Actualizar reglamentos de higiene y seguridad para incorporar el uso de monitoreo con inteligencia artificial.

4.4.3. Usabilidad y capacitación de usuarios

El *dashboard* desarrollado en Streamlit ha sido concebido con un enfoque de diseño centrado en el usuario. Los elementos visuales, por ejemplo, el semáforo de alertas o los *checklists* de estado, se basan en iconografía estándar para facilitar la comprensión inmediata de los operadores.

La capacitación se organiza según perfiles funcionales. El operador recibe un bloque práctico dedicado a interpretar alertas, aplicar filtros y generar reportes. El equipo de TI, por su parte, abarca el despliegue de contenedores `Docker`, la administración de bases de datos SQLite y los procedimientos de actualización continua de modelos.

4.4.4. Mantención y soporte de la operación

Para garantizar la vigencia del modelo de detección, se establece un plan de mantenimiento que contempla un **reentrenamiento** del modelo YOLO con nuevas imágenes etiquetadas en planta. El desempeño del componente multimodal se monitorea mediante *LangSmith*. Además, se almacenará la información (imágenes y descripciones) para una posterior evaluación del sistema.

Capítulo 5. Experimentos y resultados

En este capítulo se presentan los experimentos y resultados de los dos componentes clave del sistema: la detección de objetos y la generación de texto con LLM.

Para la detección se emplea una evaluación cuantitativa porque existen etiquetas de referencia que permiten medir la coincidencia entre las predicciones y la realidad. En cambio, el valor de las respuestas generadas por los LLM depende de la utilidad percibida por los expertos. Por esto se recurre a una evaluación cualitativa basada en una escala de Likert de cinco niveles, donde se valoran distintos criterios.

Esta doble estrategia, cuantitativa para la identificación visual y cualitativa para la interpretación textual, proporciona una visión integral del desempeño del sistema.

Los datos utilizados se clasifican en:

Imágenes Se trabajó con registros visuales obtenidos a partir de videos industriales en operación y de datasets públicos, de los cuales se extrajeron imágenes y se realizó un proceso de etiquetado de personas y elementos de protección personal. Estos datos constituyeron la base para entrenar y validar el modelo de detección YOLO, permitiendo aplicar técnicas de *Machine Learning* en un escenario con variabilidad real de condiciones ambientales y operativas.

Documentación normativa El componente de interpretación contextual se fundamentó en documentos normativos nacionales e internacionales, como la Ley N°16.744 y las guías de organismos como OSHA y NIOSH. Además, se incorporó documentación interna de la empresa colaboradora, que describe en detalle los puestos de trabajo y sus requisitos de seguridad.

Estos textos fueron procesados e indexados en la base de conocimiento del módulo RAG, lo que permitió a los modelos multimodales de lenguaje combinar aprendizaje automático con referencias normativas, otorgando justificación y trazabilidad a cada hallazgo.

5.1. Detección de Objetos

El modelo YOLO fue entrenado específicamente para reconocer un conjunto de clases que actúan como indicadores del cumplimiento de las normativas de seguridad. Estas clases son:

- **Elementos de Protección Personal (EPP)**: hardhat (casco), safety-vest (chaleco de seguridad), safety-goggles (antiparras), safety-gloves (guantes) y earmuffs (protectores auditivos).
- **Sujetos y objetos de riesgo**: person (persona) para identificar a los trabajadores en la escena, y phone (teléfono), cuyo uso puede constituir una distracción peligrosa.

El rendimiento del modelo YOLO se evalúa mediante varias métricas utilizadas en visión computacional, entre las que destacan **Precision**, **Recall**, **F1 Score** y **Mean Average Precision (mAP)**.

Precision Mide la exactitud de las predicciones del modelo, es decir, la proporción de clasificaciones positivas que realmente son correctas. Se define como la razón entre los verdaderos positivos (VP) y la suma de verdaderos positivos y falsos positivos (FP):

$$\text{Precision} = \frac{\text{VP}}{\text{VP} + \text{FP}} \tag{5.1}$$

Recall Evalúa la capacidad del modelo para identificar correctamente todos los casos positivos presentes en el conjunto de datos. Se calcula como la razón entre los verdaderos positivos (VP) y la suma de verdaderos positivos y falsos negativos (FN):

$$\text{Recall} = \frac{\text{VP}}{\text{VP} + \text{FN}} \tag{5.2}$$

F1 Score Media armónica entre **Precision** y **Recall**, y resulta útil cuando se requiere un equilibrio entre ambas métricas, por ejemplo, en escenarios con clases desbalanceadas. Se expresa como:

$$\text{F1 Score} = 2 \times \frac{\text{Precisión} \times \text{Recall}}{\text{Precisión} + \text{Recall}} \tag{5.3}$$

Mean Average Precision (mAP) Métrica de referencia para detección de objetos, ya que considera tanto la capacidad del modelo para clasificar correctamente los objetos como la exactitud en la localización de sus cajas delimitadoras (bounding boxes).

La notación **mAP50** indica que se considera una detección como correcta si la superposición entre la caja predicha y la real (medida mediante la Intersección sobre Unión, IoU) alcanza al menos un 50 %.

Por otro lado, **mAP50-95** promedia el mAP utilizando múltiples umbrales de IoU, desde 0.5 hasta 0.95 con incrementos de 0.05, proporcionando así una evaluación más estricta y representativa del rendimiento del modelo.

La fórmula general para el mAP es:

$$\text{mAP} = \frac{1}{n} \sum_{i=1}^{n} AP_i \tag{5.4}$$

donde AP_i corresponde al Average Precision de cada clase y n es el número total de clases evaluadas.

5.1.1. Resultados y Discusión

Métrica	Precision	Recall	F1 Score	mAP50	mAP50-95
Valor	0.902	0.670	0.769	0.775	0.621

Tabla 5.1: Resultados de evaluación del modelo YOLOv11 sobre el conjunto de validación.

Los resultados obtenidos, resumidos en la Tabla 5.1, evidencian que el modelo alcanza un buen rendimiento en la detección de elementos de protección personal y personas. Se observa un **mAP50** de 0.775 y un **mAP50-95** de 0.621, lo que refleja una adecuada capacidad del modelo para identificar y localizar correctamente los objetos en la mayoría de los casos.

El valor de **Precision** (0.902) destaca la baja proporción de falsas detecciones, es decir, el modelo rara vez identifica un objeto cuando no está presente. Sin embargo, el **Recall** (0.670) revela que aún existen instancias en las que el modelo no detecta algunos objetos. El **F1 Score** de 0.769 refleja un balance adecuado entre precisión y recall, validando el desempeño general del sistema.

La curva Precision-Recall (Figura 5.1) muestra variaciones en el desempeño por clase, evidenciando mejores resultados en la detección de “hardhat” y “phone”, mientras que la clase “safety-gloves” presenta el menor rendimiento. Esto podría deberse a una menor representación de dicha clase en los datos de entrenamiento, así como a la mayor dificultad visual asociada al tamaño o a la apariencia del objeto.

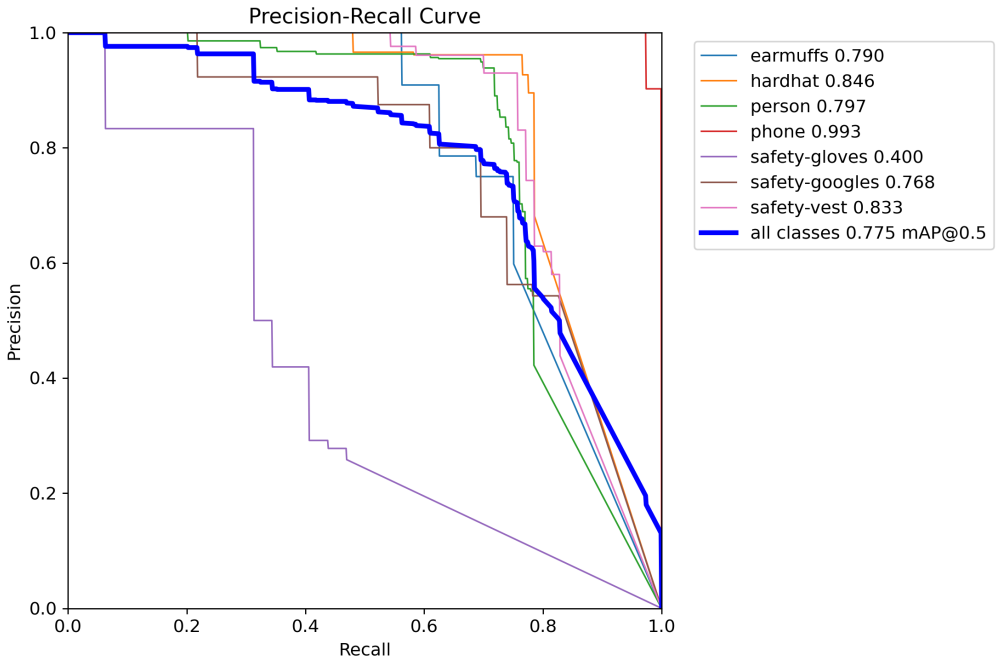


Figura 5.1: Curva Precision-Recall.

Por otro lado, la matriz de confusión normalizada (Figura 5.2) muestra que la mayoría de las predicciones se concentran en la diagonal principal, lo que indica que el modelo clasifica correctamente la mayor parte de los objetos. Sin embargo, se observan confusiones relevantes entre “earmuffs” y “hardhat”, atribuibles a la superposición que ocurre con estos objetos o a características visuales similares, así como entre “person” y “background”. Estos resultados sugieren que sería beneficioso incrementar la cantidad y calidad de muestras para las clases con menor desempeño, con el fin de reducir ambigüedades.

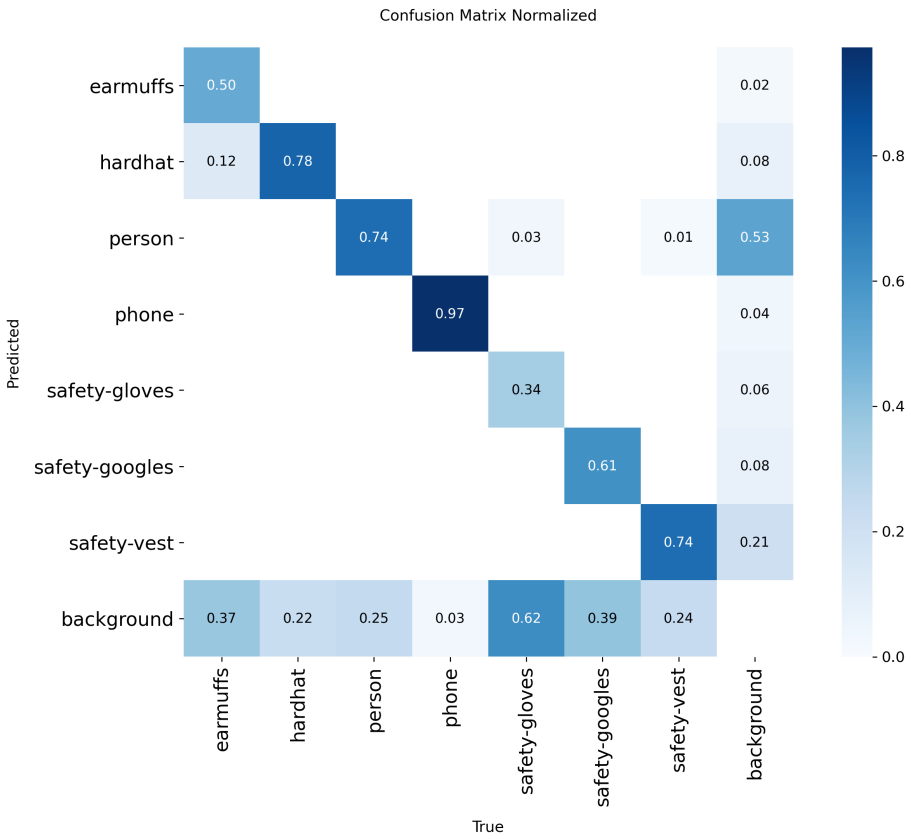


Figura 5.2: Matriz de confusión.

5.2. Modelos de Lenguaje

Se realizó una evaluación cualitativa basada en la percepción de usuarios expertos de la empresa, con el objetivo de analizar la utilidad de las respuestas generadas por los modelos de lenguaje. Este proceso consistió en aplicar una encuesta en línea, implementada mediante Google Forms, donde los participantes evaluaron distintas respuestas generadas por el componente RAG 4.3.3, utilizando una escala de Likert de cinco niveles en diversos criterios:

- **Justificación**: Evalúa qué tan bien fundamentada está la respuesta en normativa y argumentos técnicos. Considera la solidez de las referencias y la pertinencia de las normas citadas.
- **Relevancia**: Mide en qué grado la respuesta es pertinente y está directamente relacionada con la descripción de la imagen. Valora que la información aportada responda de forma específica al contexto visual presentado.
- **Coherencia**: Analiza la claridad, lógica interna y estructura del texto. Incluye la fluidez de las ideas, la corrección gramatical y la facilidad con que se entiende la argumentación.

- **Integridad**: Verifica que la respuesta sea respetuosa, imparcial y profesional. Examina la ausencia de sesgos o lenguaje inapropiado, el cumplimiento de principios éticos y el mantenimiento de un tono formal y claro.

Los criterios de evaluación fueron los siguientes:

Dimensión	1	2	3	4	5
Justificación	Sin justificación	Justificación débil	Justificación aceptable	Justificación clara y suficiente	Justificación excelente
Relevancia	Irrelevante	Poco relevante	Medianamente relevante	Relevante	Perfectamente relevante
Coherencia	Incoherente	Poco coherente	Medianamente coherente	Coherente	Muy coherente
Integridad	Poco profesional o poco ético	Mejorable en profesionalismo o ética	Aceptable profesional y ética	Profesional y ético	Totalmente profesional y ejemplar

Tabla 5.2: Escala utilizada para la evaluación cualitativa de respuestas de modelos de lenguaje.

5.2.1. Resultados y Discusión

La encuesta fue aplicada al Centro para la Industria 4.0, entidad con amplia experiencia en el sector, y respondida por dos profesionales con conocimiento especializado en la industria. Los resultados se presentan en la Tabla 5.3.

Criterio	Justificación	Relevancia	Coherencia	Integridad
Promedio	4.38	4.13	4.81	5.00

Tabla 5.3: Resultados de evaluación cualitativa sobre los modelos de lenguaje.

Las evaluaciones de **Coherencia** e **Integridad** alcanzaron buenos puntajes, evidenciando que las salidas de los modelos fueron claras, lógicas, consistentes y respetuosas. En cambio, los criterios de **Justificación** (4.38) y **Relevancia** (4.13) presentaron valores ligeramente inferiores.

En el caso de la **Justificación**, las observaciones se centraron en problemas puntuales en las referencias recuperadas por el módulo RAG, así como en la pertinencia de las normativas citadas. En ocasiones, la respuesta incluía referencias generales o parcialmente relacionadas con el contexto específico de la imagen.

Respecto a la **Relevancia**, la disminución en la valoración se atribuye principalmente a limitaciones inherentes de los MLLM, que pueden afectar la interpretación de las imágenes, y a condiciones técnicas como la distancia de la cámara respecto a la escena.

En conjunto, estos resultados indican que, aunque el desempeño general en términos de claridad, estructura y ética es adecuado, existe un margen de mejora en la precisión y pertinencia de las descripciones generadas.

5.3. Análisis Crítico

En relación con los sistemas de Mohona et al. [21] y Chen et al. [13], centrados en la detección de EPP en escenarios controlados, esta propuesta incorpora *interpretación contextual*. Combina detección con análisis multimodal y un componente RAG, de modo que las salidas no sólo señalen la presencia de EPP, sino que también incluyan referencias normativas y acciones correctivas.

Comparación con el estado del arte Mohona et al. [21] reportan un mAP50 de 0.917 con YOLOv8 sobre el dataset CHV. En contraste, el sistema propuesto obtiene un mAP50 de 0.775 con YOLOv11. La diferencia es esperable considerando condiciones menos controladas (CCTV en operación real, oclusiones, variaciones de iluminación y ángulos no ideales). Además, el entrenamiento se realizó con datos *in situ*, lo que favorece la transferencia al entorno real, pero introduce desbalance por clase.

Resultados y *trade–offs* El modelo alcanza Precision de 0.902, Recall de 0.670, mAP50 de 0.775 y mAP50–95 de 0.621, mostrando un buen compromiso *precision–recall* en clases dominantes (*person*, *hardhat*, *safety-vest*). La curva *Precision–Recall* indica peor desempeño en *safety-gloves*, consistente con su menor representación y mayor dificultad visual.

Valor agregado del componente multimodal En la evaluación cualitativa, *Coherencia* e *Integridad* presentan valores altos, mientras que *Justificación* (4.38) y *Relevancia* (4.13) disminuyen cuando el RAG recupera referencias demasiado generales o la captura limita detalles finos (distancia/ángulo de cámara).

En conjunto, estos elementos consolidan la propuesta como un aporte significativo frente a enfoques centrados exclusivamente en la detección. Aunque persisten áreas de mejora necesarias para alcanzar los niveles reportados en el estado del arte, la incorporación de análisis contextual, fundamentación normativa y mecanismos de auditoría constituye un valor diferencial con impacto práctico en entornos industriales reales.

Capítulo 6. Conclusiones

Este trabajo presentó el desarrollo de un sistema automatizado para el análisis del cumplimiento de normativas de seguridad laboral. La solución combina detección visual con YOLOv11 para identificar personas y elementos de protección personal, interpretación contextual mediante MLLM y un componente RAG que fundamenta las evaluaciones en documentación técnica y normativa.

Los resultados evidencian un desempeño adecuado en detección de objetos (`mAP50` = 0.775). En la evaluación cualitativa de los modelos de lenguaje, se obtuvieron puntajes máximos en Coherencia e Integridad, lo que refleja respuestas claras, consistentes y profesionales. Si bien Justificación (4.38) y Relevancia (4.13) fueron inferiores, la capacidad de interpretación contextual y la generación de alertas constituyen un valor agregado.

Pese a estos resultados, se identificaron ciertas limitaciones. La calidad de los resultados de YOLO depende de la cantidad y calidad de imágenes y etiquetas; además, el Recall (0.670) indica la existencia de clases no detectadas. En el componente MLLM, la Justificación se vio afectada por recuperaciones del RAG demasiado generales, mientras que la Relevancia se asoció a limitaciones de los MLLM para captar detalles finos y a condiciones de captura. El horizonte temporal del proyecto restringió la profundidad de las pruebas en entornos reales. Aunque la configuración actual de cámaras reduce el riesgo de identificación, la protección de datos personales sigue siendo un aspecto ético importante para futuros despliegues.

En relación con los objetivos específicos, todos fueron verificados de manera progresiva. El análisis de la normativa vigente se materializó en la construcción de la base de conocimiento para el RAG; la revisión del estado del arte se abordó en el Capítulo 3, sentando las bases para un diseño integrado cuya arquitectura se presentó en el Capítulo 4. Dicho diseño se concretó con la implementación de un prototipo funcional, validado a través de experimentos cuantitativos y cualitativos en el Capítulo 5. Estos resultados confirman el cumplimiento integral de los objetivos planteados y, con ello, del objetivo general de la memoria de título.

Adicionalmente, la propuesta se vincula con los Objetivos de Desarrollo Sostenible (ODS). Contribuye al ODS 3 (Salud y Bienestar) al promover la prevención de accidentes y la protección de la salud de los trabajadores; al ODS 8 (Trabajo Decente y Crecimiento Económico) al reforzar ambientes

laborales seguros; y al ODS 9 (Industria, Innovación e Infraestructura) al aplicar tecnologías avanzadas de inteligencia artificial en un contexto industrial real, ejemplificando cómo la innovación tecnológica puede fortalecer la seguridad y modernizar procesos productivos.

Como trabajo futuro, se propone ampliar y balancear el conjunto de datos para mejorar la detección, y optimizar el RAG para asegurar la pertinencia de las normas y referencias, lo que mejorará la justificación de las respuestas. Asimismo, se plantea reforzar la capacidad de los MLLM para captar detalles visuales finos y mitigar los efectos de las condiciones de captura. De igual modo, resulta necesario profundizar en el análisis de las implicancias económicas, ambientales y éticas del sistema, considerando tanto los costos de implementación como el consumo energético asociado al uso intensivo de modelos de lenguaje. En el ámbito ético, se debe considerar la privacidad de los trabajadores, la transparencia en el uso de los datos y la aceptación social de sistemas de monitoreo automatizado. Asimismo, es fundamental evitar que la herramienta sea utilizada con fines punitivos hacia los trabajadores.

Finalmente, se sugiere evaluar el finetuning de MLLM en ciertos escenarios, en línea con enfoques como **ChatCH**, con el objetivo de aumentar la adaptación y precisión al dominio. Este ajuste podría permitir consolidar la arquitectura en un único modelo en lugar del esquema de dos niveles. En paralelo, la incorporación de políticas de muestreo y un despliegue híbrido favorecerán una operación escalable y eficiente.

Bibliografía

[1] Organización Internacional del Trabajo (OIT), *Estadísticas sobre seguridad y salud en el trabajo*, 2024. dirección: https://ilostat.ilo.org/es/topics/safety-and-health-at-work/.

[2] J. Davies, A. Ross y B. Wallace, *Safety Management: A Qualitative Systems Approach*. Taylor & Francis, 2003, ISBN: 9780415303712. dirección: https://books.google.cl/books?id=h7YixY365aQC.

[3] National Institute for Occupational Safety and Health (NIOSH), *Falls in the Workplace*, 2024. dirección: https://www.cdc.gov/niosh/falls/about/?CDC_AAref_Val=https://www.cdc.gov/niosh/topics/falls/default.html.

[4] Centers for Disease Control and Prevention (CDC), *All Workplace Safety & Health Topics*, 2020. dirección: https://www.cdc.gov/niosh/topics/default.html.

[5] National Institute for Occupational Safety and Health (NIOSH), *Personal Protective Equipment (PPE)*, 2021. dirección: https://www.cdc.gov/niosh/ppe/default.html.

[6] G. de Chile, *Ley N°16.744 sobre accidentes del trabajo y enfermedades profesionales*, 1968. dirección: https://www.bcn.cl/leychile/navegar?idNorma=28650.

[7] M. de Salud de Chile, *Decreto Supremo N°594, Reglamento sobre condiciones sanitarias y ambientales básicas en los lugares de trabajo*, 2000. dirección: https://www.bcn.cl/leychile/navegar?idNorma=167766.

[8] S. de Seguridad Social (SUSESO), *Estadísticas de Accidentes del Trabajo 2023*, 2024. dirección: https://www.suseso.cl/607/articles-732522_archivo_01.pdf.

[9] K. A. L. Group, *Fatal Falls in Construction Caused by Employers Failing to Plan for Worker Safety*, 2022. dirección: https://www.kenallenlaw.com/es/2022/11/fatal-falls-in-construction-caused-by-employers-failing-to-plan-for-worker-safety/.

[10] A. Radford, J. W. Kim, C. Hallacy et al., *Learning Transferable Visual Models From Natural Language Supervision*, 2021. arXiv: 2103.00020 [cs.CV]. dirección: https://arxiv.org/abs/2103.00020.

[11] H. Liu, C. Li, Q. Wu e Y. J. Lee, *Visual Instruction Tuning*, 2023. arXiv: 2304.08485 [cs.CV]. dirección: https://arxiv.org/abs/2304.08485.

[12] S. de Previsión Social, *Seguridad y Salud en el Trabajo*, https://previsionsocial.gob.cl/organizaciones/empresas-y-empleadores-empleadoras/seguridad-y-salud-en-el-trabajo/, [Accessed 28-07-2025], 2025.

[13] Z. Chen, H. Chen, M. Imani, R. Chen y F. Imani, *Vision Language Model for Interpretable and Fine-grained Detection of Safety Compliance in Diverse Workplaces*, 2024. arXiv: 2408.07146 [cs.CV]. dirección: https://arxiv.org/abs/2408.07146.

[14] J. Mao, S. Shi, X. Wang y H. Li, *3D Object Detection for Autonomous Driving: A Comprehensive Survey*, 2023. arXiv: 2206.09474 [cs.CV]. dirección: https://arxiv.org/abs/2206.09474.

[15] K. J. Oguine, O. C. Oguine y H. I. Bisallah, *YOLO v3: Visual and Real-Time Object Detection Model for Smart Surveillance Systems(3s)*, 2022. arXiv: 2209.12447 [cs.CV]. dirección: https://arxiv.org/abs/2209.12447.

[16] J. Redmon, S. Divvala, R. Girshick y A. Farhadi, *You Only Look Once: Unified, Real-Time Object Detection*, 2016. arXiv: 1506.02640 [cs.CV]. dirección: https://arxiv.org/abs/1506.02640.

[17] R. Khanam y M. Hussain, *YOLOv11: An Overview of the Key Architectural Enhancements*, 2024. arXiv: 2410.17725 [cs.CV]. dirección: https://arxiv.org/abs/2410.17725.

[18] A. Vaswani, N. Shazeer, N. Parmar et al., *Attention Is All You Need*, 2023. arXiv: 1706.03762 [cs.CL]. dirección: https://arxiv.org/abs/1706.03762.

[19] M. A. K. Raiaan, M. S. H. Mukta, K. Fatema et al., «A Review on Large Language Models: Architectures, Applications, Taxonomies, Open Issues and Challenges,» *IEEE Access*, vol. 12, págs. 26 839-26 874, 2024. DOI: 10.1109/ACCESS.2024.3365742.

[20] Y. Gao, Y. Xiong, X. Gao et al., *Retrieval-Augmented Generation for Large Language Models: A Survey*, 2024. arXiv: 2312.10997 [cs.CL]. dirección: https://arxiv.org/abs/2312.10997.

[21] R. T. Mohona, S. Nawer, M. S. I. Sakib y M. N. Uddin, «A YOLOv8 Approach for Personal Protective Equipment (PPE) Detection to Ensure Workers Safety,» en *2024 3rd International Conference on Advancement in Electrical and Electronic Engineering (ICAEEE)*, 2024, págs. 1-6. DOI: 10.1109/ICAEEE62219.2024.10561752.

[22] M. A. Tami, H. I. Ashqar y M. Elhenawy, *Using Multimodal Large Language Models for Automated Detection of Traffic Safety Critical Events*, 2024. arXiv: 2406.13894 [cs.CV]. dirección: https://arxiv.org/abs/2406.13894.

[23] Q. Chen y X. Yin, «Tailored vision-language framework for automated hazard identification and report generation in construction sites,» *Advanced Engineering Informatics*, vol. 66, pág. 103 478, 2025, ISSN: 1474-0346. DOI: https://doi.org/10.1016/j.aei.2025.103478. dirección: https://www.sciencedirect.com/science/article/pii/S1474034625003714.

Anexo A. Anexo

Modelo	Entrada	Salida	Contexto	Salida	Corte
gpt-4.1	$3.00	$12.00	1,047,576 tokens	32,768 tokens	May 2024
gpt-4.1-mini	$0.80	$3.20	1,047,576 tokens	32,768 tokens	May 2024
gemini-2.5-flash-lite	$0.10	$0.40	1,000,000 tokens	64,000 tokens	Ene 2025

Tabla A.1: Precios y especificaciones de los modelos de lenguaje (valores en USD por millón de tokens).

Componente	Costos 1	Costos 2
GPU	NVIDIA RTX 4090 - $3,199.99	NVIDIA RTX 3060 - $409.99
CPU	Intel Core i9-13900K - $589.99	Intel Core i5-12400F - $189.99
Placa Madre	ASUS Z790 - $299.99	ASUS B660 - $149.99
RAM	64GB DDR5 - $299.99	32GB DDR4 - $129.99
Almacenamiento	SSD NVMe 2TB - $199.99	SSD NVMe 1TB - $99.99
Fuente de Poder	1000W - $199.99	650W - $89.99
Gabinete	ATX - $149.99	ATX - $79.99
Total Estimado	≈ **$4,940**	≈ **$1,150**

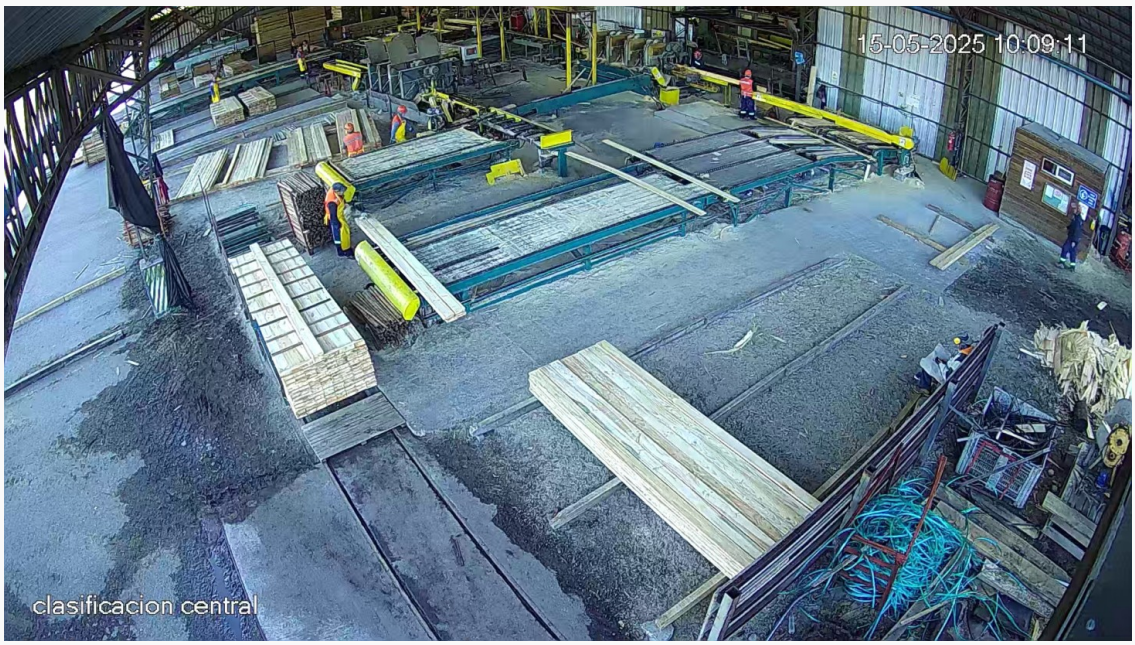
Tabla A.2: Presupuesto de hardware.

Precios referenciales en US$ de PCPartPicker.
Costos 1 considera el uso de LLM locales y modelos de detección de objetos.
Costos 2 considera solo modelos de detección de objetos.

Contexto: Manipulador de madera
Estado: Peligro

El semáforo de seguridad es ROJO debido a la manipulación manual de madera sobre maquinaria en movimiento sin el uso completo de EPP obligatorio (solo se observa casco de seguridad, faltando protección auditiva, guantes y protección ocular), lo que representa un riesgo crítico inmediato de lesiones graves. El área presenta desorden con piezas de madera sueltas, aumentando el riesgo de caídas, atrapamientos y golpes. Las condiciones ambientales muestran iluminación adecuada pero con zonas de sombra y acumulación de materiales, lo que afecta la seguridad y el cumplimiento normativo. Se identifican incumplimientos en el uso de EPP, orden y limpieza, y señalización de riesgos. Se recomienda como acciones prioritarias: exigir el uso completo de EPP, limpiar y organizar el área de trabajo, e instalar resguardos y señalización adecuada para mitigar los riesgos críticos observados.

Figura A.1: Reporte generado.



Contexto: Manipulador de madera
Estado: Advertencia

El semáforo de seguridad es amarillo debido a la presencia de EPP parcial (casco y chaleco reflectante), pero ausencia visible de guantes anticorte y protección ocular, ambos obligatorios en la manipulación de madera. El entorno está razonablemente ordenado y bien iluminado, aunque hay acumulación de aserrín, restos de madera y mangueras/cables en el suelo, lo que representa riesgos de tropiezo y caída. No se observan bloqueos de salidas, pero falta señalización clara de zonas de riesgo. El cumplimiento normativo es parcial, con deficiencias en el uso de EPP y señalización. Se priorizan como acciones correctivas: exigir el uso completo de EPP, retirar obstáculos del suelo y mejorar la señalización de seguridad.

Figura A.2: Reporte generado.

Contexto: Manipulador de madera
Estado: Advertencia

El semáforo de seguridad es amarillo debido a la presencia de EPP parcial (casco y chaleco reflectante), pero ausencia visible de protección auditiva, ocular y guantes, todos obligatorios en aserraderos. El entorno está ordenado y bien iluminado, sin obstrucciones ni derrames, pero existe riesgo relevante por la proximidad de personas a maquinaria en operación y tránsito de montacargas. No se observan bloqueos de rutas de evacuación ni uso de dispositivos prohibidos. Se recomienda reforzar el uso de EPP completo, mejorar la segregación entre peatones y maquinaria, y supervisar el cumplimiento de guantes y calzado de seguridad para reducir riesgos críticos.

Figura A.3: Reporte generado.

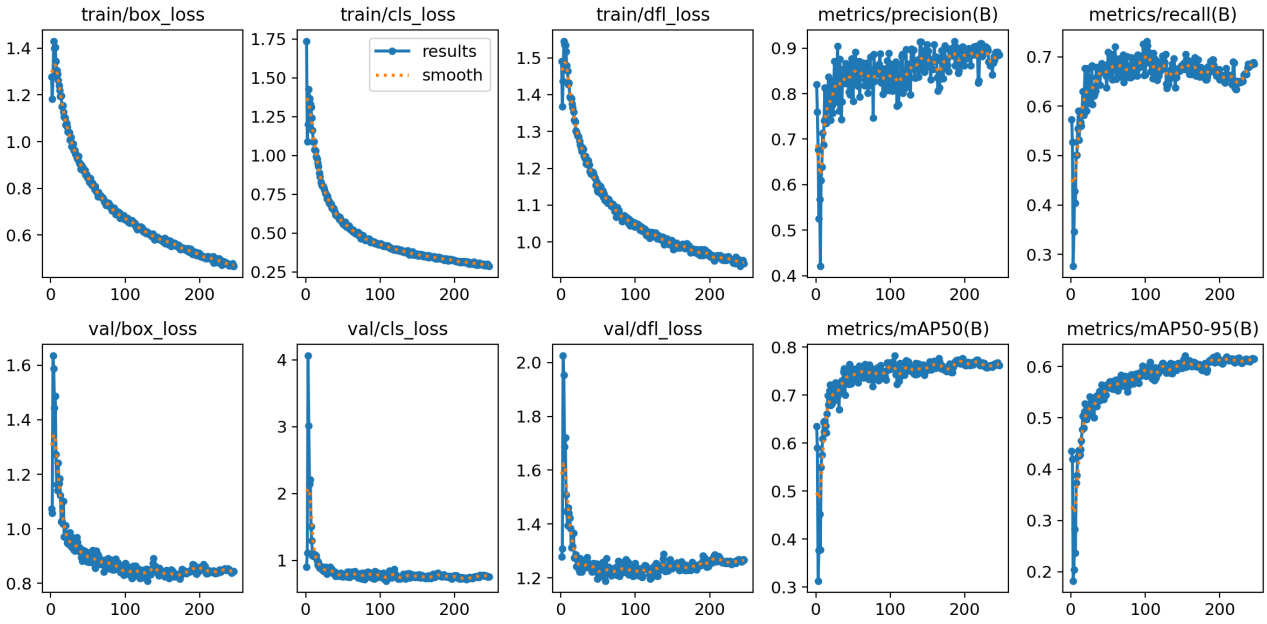


Figura A.4: Resultados detección de objetos.

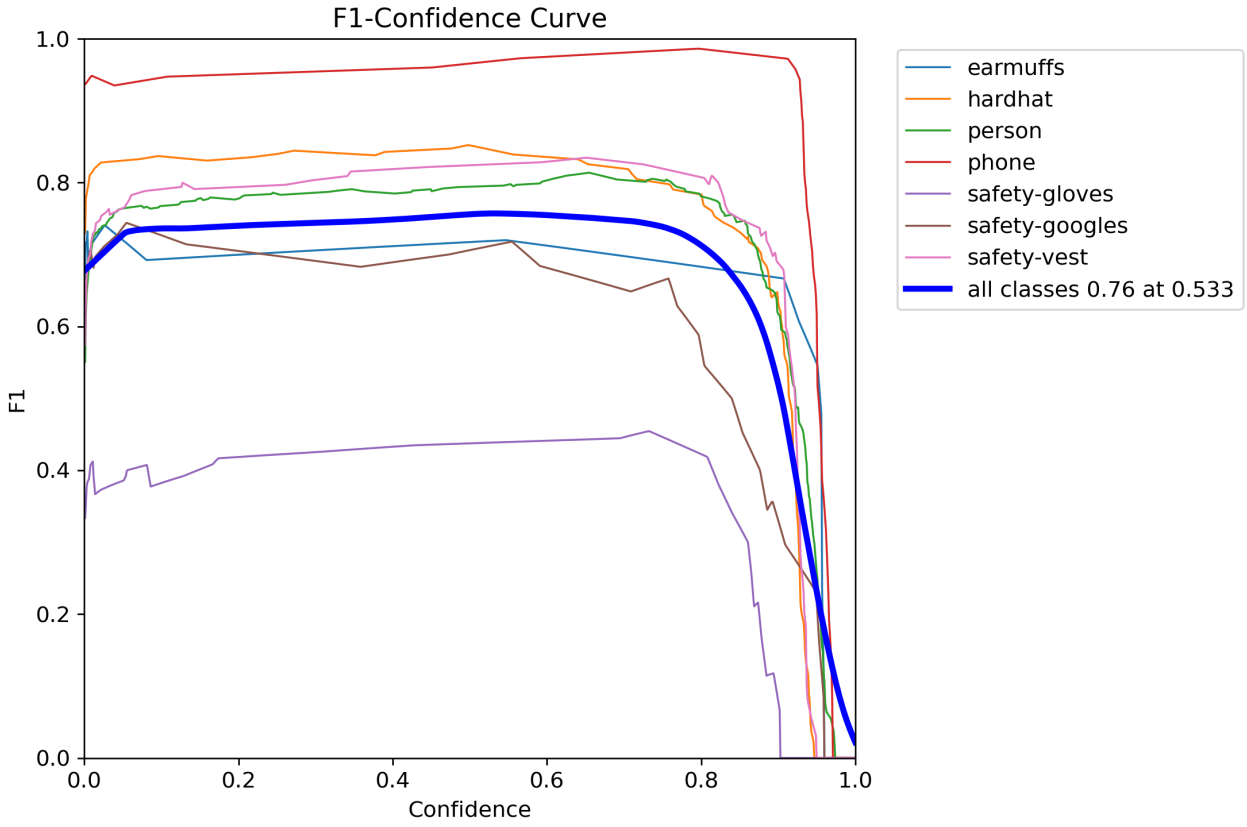


Figura A.5: Curva F1.

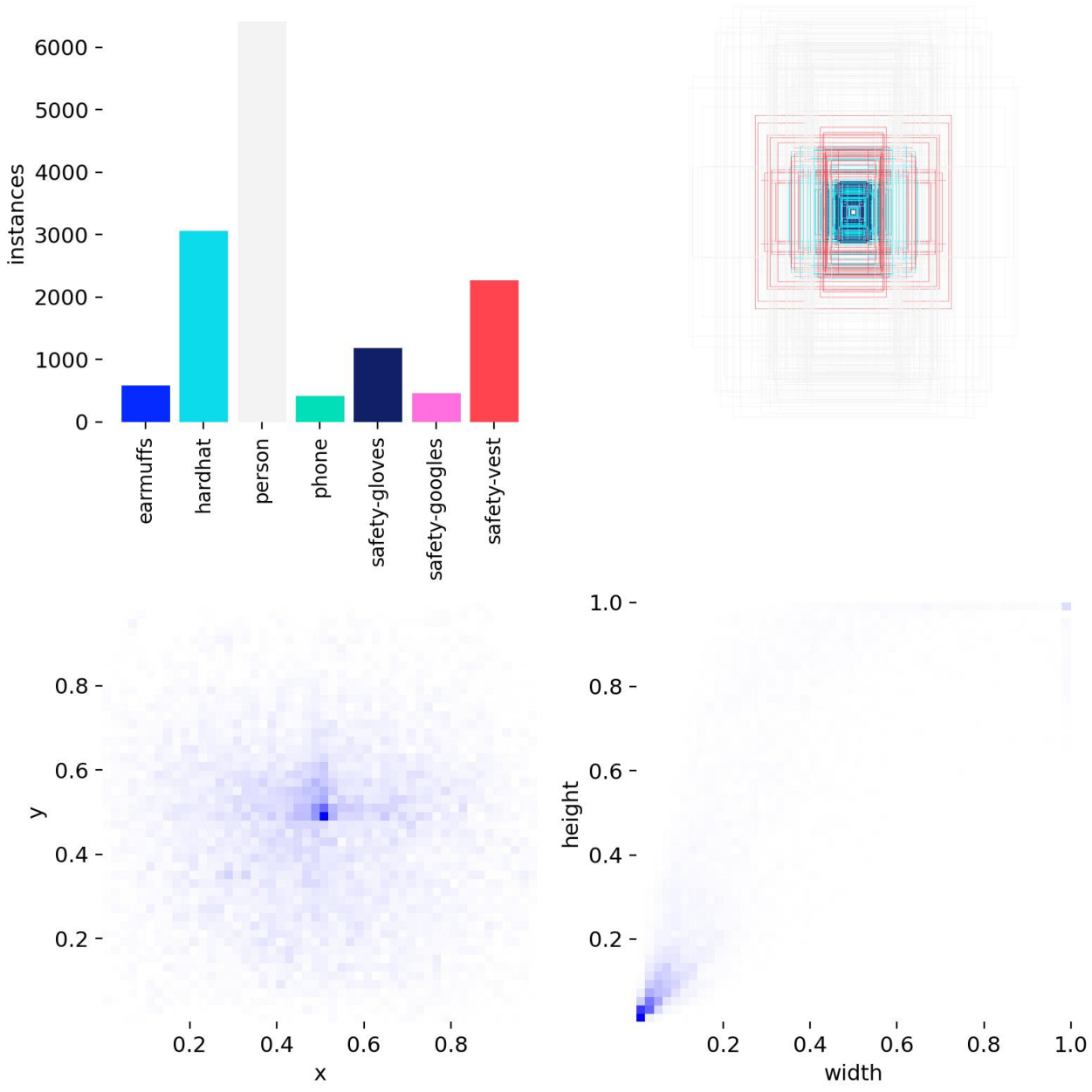


Figura A.6: Información de las etiquetas.

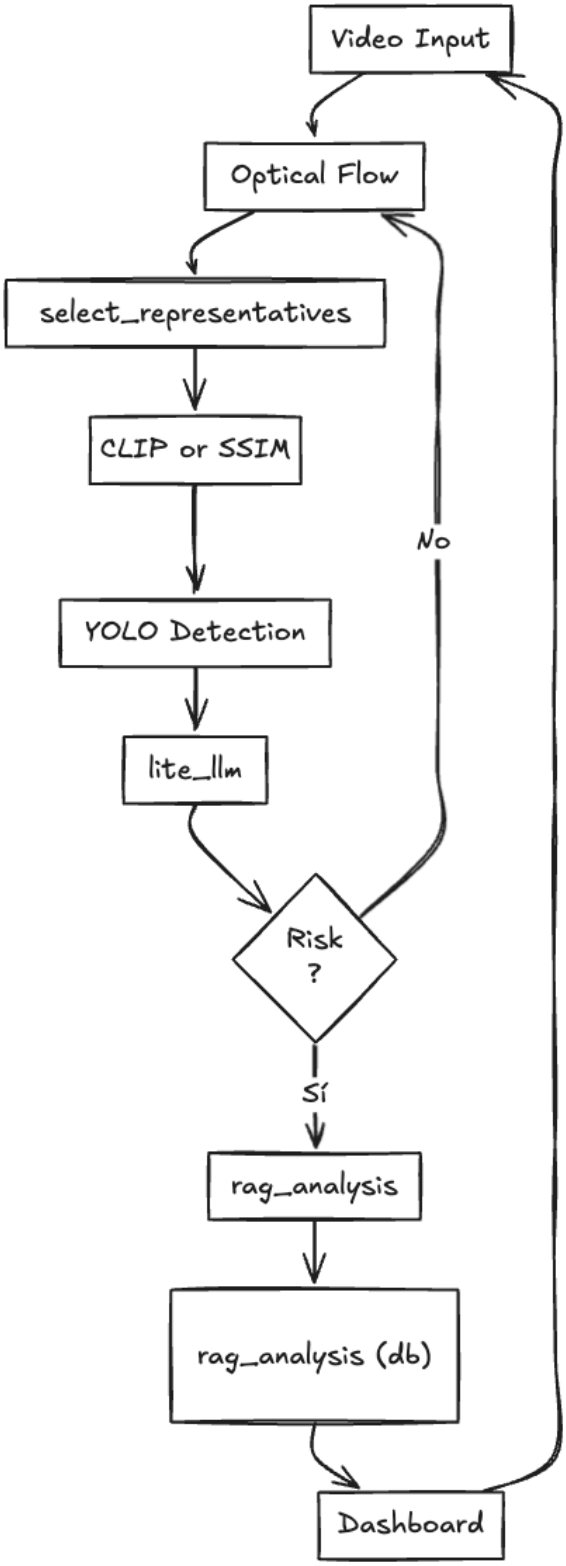


Figura A.7: Flujo del sistema.