**Desarrollo de los objetivos**

**O1. Normalización del conocimiento (Ontología de riesgos)**

**Propósito.** Estandarizar la identificación de riesgos y las recomendaciones aplicables mediante una ontología editable por el área de SST, sin depender del código.

**Alcance.** Primera versión centrada en:

* Atrapamiento/atropellamiento por equipos móviles.
* Contacto con partes móviles/maquinaria.
* Caídas al mismo nivel por obstáculos (orden y aseo).

**Metodología.**

1. Revisión de incidentes y condiciones típicas del entorno.
2. Definición de disparadores visuales (clases detectadas) y condiciones lógicas all\_of / any\_of.
3. Priorización de controles según Jerarquía (eliminación, sustitución, ingeniería, administrativos, EPP).
4. Vinculación a normas base (Decreto 1072/2015, Resolución 0312/2019 e ISO 45001:2018).
5. Validación con responsable SST.

**Implementación.**

* Archivo de referencia: risk\_ontology.yaml.
* Estructura: normas → riesgos → disparadores → recomendaciones por Jerarquía.
* El motor de reglas aplica la ontología a la lista de clases detectadas, sin reglas “quemadas” en el código.

**Evidencias.**

* Ontología versionada en el repositorio.
* Pruebas en la UI con imágenes representativas, verificando que el riesgo inferido coincide con el criterio del experto.

**Criterios de aceptación.**

* Activación correcta de riesgos a partir de los disparadores definidos.
* Recomendaciones ordenadas por Jerarquía de Control.
* Citas normativas consistentes y trazables.

**O2. Presentación textual de hallazgos y recomendaciones**

**Propósito.** Entregar una salida clara, breve y accionable para actas e informes, enfocada en decisiones prácticas.

**Metodología.**

1. Definición de un estilo de redacción factual (qué se observó, riesgo asociado, controles priorizados, referencia normativa).
2. Implementación de la capa textual que recibe clases detectadas, riesgos inferidos y conjunto de controles.
3. Integración en la UI de imágenes y video, y exposición por API para consumo externo.

**Implementación.**

* UI (ui\_gradio.py y ui\_gradio\_video.py): imagen anotada, listado de clases, riesgos, controles y resumen redactado.
* API (app\_fastapi.py):
  + POST /analyze → JSON con detecciones, riesgos y recomendaciones.
  + POST /analyze-chat → incluye resumen textual consolidado.

**Ejemplo de redacción (estilo).**  
“Se evidenció interacción persona–equipo móvil en área de maniobras. Priorizar barreras físicas y demarcación de rutas (ingeniería), plan de tráfico y permisos (administrativos), y uso de chaleco de alta visibilidad (EPP). Referencia normativa: Decreto 1072/2015; Resolución 0312/2019; ISO 45001:2018.”

**Evidencias.**

* Capturas de la UI con casos por categoría de riesgo.
* Respuestas de /analyze-chat archivadas para auditoría.

**Criterios de aceptación.**

* Coherencia entre riesgos detectados y controles propuestos.
* Texto comprensible, sin ambigüedades, priorizado por Jerarquía.
* Referencias normativas claras (sin transcribir textos extensos).

**O3. Evaluación objetiva del desempeño**

**Propósito.** Medir de forma reproducible el desempeño frente a etiquetas de experto, con métricas estándar.

**Metodología.**

1. Construcción de un conjunto de verdad terreno (ground truth) a nivel de imagen con los riesgos esperados.
2. Evaluación con métricas cuantitativas (Precision, Recall, F1 micro) y acuerdo interevaluador (Cohen’s Kappa).
3. Análisis de falsos positivos/negativos para retroalimentar dataset, ontología y umbrales.

**Implementación.**

* Formato de entrada: ground\_truth.csv con columnas image\_path y risks (separadas por “;” cuando hay múltiples).
* Script de evaluación: evaluator.py (compara salida del sistema vs. ground truth y calcula métricas).
* Reporte: totales de aciertos/errores y ejemplos representativos para revisión conjunta con SST.

**Evidencias.**

* CSV etiquetado por experto.
* Informe de métricas y casos de discrepancia con explicación.

**Criterios de aceptación.**

* Alcanzar umbrales acordados con SST (referencia inicial: F1 micro ≥ 0,70; Kappa ≥ 0,60).
* Documentar ajustes y su impacto tras cada ciclo de mejora.

**Gobierno, control de cambios y uso en equipo**

* **Propiedad de la ontología:** área de SST; cambios mediante control de versiones y revisión por pares.
* **Pesos del modelo:** models/best.pt incluido en el repositorio (se recomienda Git LFS).
* **Datos:** no se versionan datasets sensibles; se utiliza datasets/ local y una muestra sst\_sample/ para demostración.
* **Compartición:** clonación del repositorio, instalación de dependencias y ejecución de la UI. La ruta por defecto usa models/best.pt.

**Limitaciones y consideraciones**

* Desempeño dependiente de la similitud entre el dataset y el entorno real (iluminación, ángulos, oclusiones).
* Clases finas (p. ej., cables o derrames) requieren más datos y etiquetado consistente.
* Herramienta de apoyo a inspección; la decisión final permanece en el responsable SST.