|  |  |  |  |
| --- | --- | --- | --- |
| ***Interesados en el proyecto*** | | | |
| ***Interesado*** | ***Nombre*** | ***Función*** | ***Teléfono*** |
| Director del proyecto | Yersson Calderon Romero | Supervisar el proyecto y tomar decisiones estratégicas. | 925271857 |
| Gerente del proyecto |  | Gestionar las actividades diarias y coordinar al equipo. |  |
| Coordinador de cronograma y presupuesto |  | Planificar y controlar el cronograma y presupuesto del proyecto. |  |
| Docente del curso | Americo Estrada Sanchez | Evaluador del proyecto | - |

**Repositorio:** [**https://github.com/Taller-de-proyectos-I/IDEA-PROYECTO.git**](https://github.com/Taller-de-proyectos-I/IDEA-PROYECTO.git)

**ÍNDICE**

[**1. PLANTEAMIENTO DE PROPUESTAS 2**](#_uezntphoj2en)

[1.1. propuesta 1 2](#_z36acm61mvof)

[1.2. propuesta 2 2](#_hm735v2jta3d)

[1.3. propuesta 3 2](#_69qujv6ny6ts)

[1.4. propuesta 4 2](#_8do6rvxrc82o)

[**2. COMPARATIVA ENTRE PROPUESTAS PARA SU ANÁLISIS 2**](#_8hs7arnhc029)

[**3. EVALUACION Y ANALISIS GRUPAL 2**](#_ebxn6hdxld74)

[**4. JUSTIFICACIÓN 2**](#_j4okbkvz8ots)

[**5. DIAGRAMA DE ISHIKAWA BASADO EN ERRORES 2**](#_319ngwgnnwa8)

[**6. DESARROLLO DE LA METODOLOGÍA ÁGIL 2**](#_kn255f6cucu2)

[6.1. Historias de usuario 2](#_xh361z95hw)

[6.2. Estimación de tiempos 2](#_pq2alevlwoho)

[6.3. Estimación de épicas 2](#_sga0vxueo5sj)

[6.4. Product backlog según prioridades y estimaciones 2](#_c3np63o0t6rr)

[6.5. Cronograma: diagrama de gantt 2](#_f2x7ngu3h60u)

# PLANTEAMIENTO DE PROPUESTAS

## Propuesta 1

**Título de la Propuesta:**

Sistema Inteligente de Reconocimiento de Vehículos y Alerta Automática para la Gestión de Seguridad Vial en la Vía de Acceso a la Universidad Continental – Saylla, Cusco 2025

**Propuesto por:** Yersson Calderon Romero

**Problema:**

En la vía de acceso frente a la Universidad Continental, en Saylla (Cusco), se presentan con frecuencia congestionamientos, incidentes viales y fugas de vehículos que dificultan la respuesta inmediata de las autoridades, como el serenazgo o la Policía Nacional. Esa situación pone en riesgo tanto a peatones como a conductores, especialmente considerando que la infraestructura y señalización de la zona aún son novedosas para muchos.

Según el Observatorio Nacional de Seguridad Vial, en el año 2023 se registraron en todo el país 87 083 siniestros de tránsito, con 58 000 personas heridas y 3 316 fallecidas.

Además, Cusco se posicionó como la segunda región con más siniestros fatales, con 182 casos, solo detrás de Lima (370).

**Descripción de la solución Propuesta:**

Se propone desarrollar un sistema inteligente de monitoreo vehicular, instalado en la vía de acceso a la Universidad Continental – Saylla, que utilice cámaras y visión por computadora (CNN, YOLO) para reconocer en tiempo real las placas, tipo y características de los vehículos. Este sistema utilizará:

1. Cámaras inteligentes junto con modelos de visión artificial (por ejemplo, YOLO para detección de vehículos).
2. Algoritmos de reconocimiento automático de placas (ALPR) y clasificación de tipo de vehículo utilizando CNN u OCR como EasyOCR.
3. Módulo de alerta automática, que enviará información clave (placa, tipo, hora, ubicación, una fotografía del vehículo) al serenazgo o la PNP en caso de detectar:

- Vehículos involucrados en accidentes que intentan fugarse.

- Maniobras sospechosas o incumplimiento de normas viales.

1. Una base de datos centralizada para almacenar registros históricos y permitir análisis posteriores.
2. Canales de notificación en tiempo real (como una aplicación web) para que las autoridades puedan actuar rápidamente.

**Requerimientos del producto de alto nivel:**

1. Detectar y reconocer vehículos en tiempo real mediante cámaras instaladas en la vía de acceso a la universidad.
2. Identificar placas vehiculares automáticamente usando algoritmos OCR e inteligencia artificial.
3. Clasificar el tipo de vehículo (auto, camioneta, moto, bus) a partir de las imágenes captadas.
4. Registrar evidencia (imagen, hora, ubicación, tipo de vehículo y placa) en una base de datos centralizada.
5. Generar alertas automáticas en caso de fuga tras un accidente o comportamiento sospechoso.
6. Enviar notificaciones en tiempo real a serenazgo o PNP a través de un SMS o aplicación web.
7. Acceder a un panel de monitoreo web donde las autoridades puedan visualizar los incidentes y el historial.
8. Almacenar reportes históricos para análisis de tráfico, zonas de riesgo y estadísticas.
9. Soportar integración con sistemas de seguridad existentes (como cámaras municipales o bases de datos policiales).
10. Funcionar de manera continua (24/7) con capacidad de autodiagnóstico básico (alerta si la cámara falla o no hay conexión).

**Requerimientos Tecnológicos:**

**Hardware:**

1. Laptop con las siguientes características mínimas:
   * Procesador: Intel Core i7 (10ma gen o superior).
   * Memoria RAM: 16 GB min.
   * Tarjeta Gráfica (GPU): NVIDIA RTX 3070.
2. Almacenamiento:
   * SSD de 512 GB.
   * HDD/SSD adicional de 1 TB
3. Cámara Web / Cámara IP (mín. 1080p, 30 fps)
4. Router y conexión a internet estable
5. UPS (Sistema de alimentación ininterrumpida) – Para evitar pérdidas de datos en caso de cortes eléctricos.

**Software:**

1. Lenguajes de Programación:
2. Python 3.10+
3. Frameworks de Visión Artificial e IA:
4. OpenCV (procesamiento de imágenes).
5. TensorFlow 2.x / PyTorch (entrenamiento y ejecución de CNN, ej. YOLOv8).
6. EasyOCR (para lectura de placas).
7. Frameworks Web / Backend:
8. Flask (para API REST y microservicios).
9. Base de Datos:
10. PostgreSQL o MySQL (para gestión de registros históricos).
11. Frontend / Dashboard:
12. HTML5, CSS3, JavaScript.
13. Notificaciones y Comunicación:
14. API de Telegram (bot para alertas).

**Estado del Arte: Análisis de la competencia:**

|  |
| --- |
| **1. Sistema de Reconocimiento Automático de Placas Vehiculares (ANPR) – ParkPow**   * Nombre: Deep License Plate Recognition * Versión / Año: GitHub Open Source Project (2021, actualizado en 2023) * Funcionalidades:   + Detecta vehículos y extrae automáticamente la matrícula de imágenes o video en tiempo real.   + Compatible con múltiples países y formatos de placas. Permite integración mediante API. * Limitaciones:   + Enfocado principalmente en la lectura de placas.   + No incluye sistema de alertas automáticas ni clasificación avanzada del vehículo (tipo, color). * Ventaja de nuestra propuesta:   + Integramos no solo la detección de placas, sino también la clasificación de tipo de vehículo.   + Incorporamos un módulo de alerta inmediata a serenazgo/PNP, adaptado al contexto local (Saylla – Cusco).   + Implementamos un panel web con historial y notificaciones en tiempo real, no solo extracción de datos.   Referencia: [GitHub – parkpow/deep-license-plate-recognition](https://github.com/parkpow/deep-license-plate-recognition?utm_source=chatgpt.com)  **2. An efficient and layout-independent automatic license plate recognition system based on the YOLO detector” — Laroca et al., 2021 (IET Intelligent Transport Systems)**  Año / versión: 2021, enfoque layout-independent (independiente del diseño de la placa).  Funcionalidades: unifica detección de LP + clasificación del layout y aplica reglas de post-proceso para mejorar OCR; evalúa 8 datasets (China, Europa, Brasil, etc.) con 96.9% de tasa promedio y > 70 FPS en GPU alta, manteniendo tiempo real con múltiples vehículos.  Dónde nuestra propuesta mejora:   * Añadimos clasificación de tipo de vehículo (auto/camión/bus/moto) y atributos (color/marca opcional) para correlacionar con infracciones o eventos. * Entrenamiento fino con datos locales (dominios .pe) y evaluación A/B en la intersección frente a Continental para reducir falsos positivos por obras/señalización nueva.   Referencia: [raysonlaroca.github.io](https://raysonlaroca.github.io/papers/laroca2021efficient-published.pdf) |

## Propuesta 2

**Título de la Propuesta:**

Implementación de sistema de visión por computadora con CNN para diagnóstico de enfermedades en hojas de cultivos andinos en comunidades de Cusco.

**Propuesto por: Yersson Calderon Romero**

**Problema:**

En comunidades agrícolas del Cusco, los agricultores muchas veces no identifican a tiempo las enfermedades o plagas que afectan las hojas de sus cultivos. Esto implica pérdidas económicas, menor productividad y un riesgo latente para la seguridad alimentaria.

La FAO (Organización de las Naciones Unidas para la Alimentación y la Agricultura) estima que hasta el 40 % de los cultivos alimentarios a nivel global se pierden cada año por enfermedades y plagas de las plantas ([FAO, 2021](https://agn.gt/fao-enfermedades-y-plagas-de-las-plantas-causan-la-perdida-del-40-de-los-cultivos/?utm_source=chatgpt.com)).

En el caso del Perú, el SENASA (Servicio Nacional de Sanidad Agraria) reporta que los agricultores de la sierra enfrentan con frecuencia la aparición de enfermedades como tizón tardío en papa, roya amarilla en trigo y manchas foliares en maíz, lo que afecta directamente a pequeños productores que no cuentan con sistemas modernos de monitoreo ([SENASA, 2020](https://www.senasa.gob.pe/senasacontigo/minagri-mantiene-el-control-de-plagas-y-enfermedades-en-la-agricultura/?utm_source=chatgpt.com)).

**Descripción de la solución Propuesta:**

Se propone implementar un sistema de visión por computadora con redes neuronales convolucionales (CNN) que permita diagnosticar de manera temprana enfermedades en hojas de cultivos clave como papa, maíz y tomate.  
  
El sistema permitirá a los agricultores:

* Capturar o subir una foto de la hoja.
* Analizar la imagen mediante un modelo CNN entrenado.
* Obtener un diagnóstico preliminar con precisión superior al 85 %.
* Recibir recomendaciones iniciales para el manejo de la enfermedad detectada.

El enfoque inicial será sobre cultivos andinos de alta relevancia económica, con la posibilidad de ampliar progresivamente a otros como uva, fresa o pimiento.

**Requerimientos del producto de alto nivel:**

* Capturar o cargar imágenes de hojas de cultivo.
* Detectar si la hoja está sana o enferma.
* Identificar el tipo de enfermedad en los cultivos priorizados (papa, maíz, tomate).
* Mostrar recomendaciones iniciales de manejo.
* Guardar historial de diagnósticos por usuario.
* Ofrecer una interfaz accesible en web.
* Permitir actualización del modelo CNN con nuevos datos.
* Generar reportes de enfermedades más frecuentes en cada zona.
* Incluir progresivamente nuevos cultivos según disponibilidad.
* Funcionar en comunidades rurales con baja conectividad (soporte offline).

**Requerimientos Tecnológicos:**

**Hardware:**

* Computadoras de arquitectura x86 o x64.
* GPU NVIDIA (mínimo 8GB VRAM, RTX 3060 superior)
* Memoria RAM ≥16GB.
* Almacenamiento SSD (mínimo 200GB).
* Smartphones
* Servidor en la nube

**Software:**

* Lenguajes de programación: Python (principal), JavaScript
* Frameworks de Deep Learning: TensorFlow/Keras, PyTorch.
* Librerías de procesamiento de imágenes: OpenCV, Pillow, Albumentations (data augmentation).
* Framework web: Flask o FastAPI para API de predicción.
* Control de versiones: GitHub.
* Entornos de ejecución: Google Colab, VS Code, Jupyter Notebook.

**Estado del Arte: Análisis de la competencia:**

|  |
| --- |
| 1. **PlantVillage App (Penn State University, 2019)**    * Detecta enfermedades en diferentes cultivos usando inteligencia artificial.    * Diferencia: nuestra propuesta está enfocada en cultivos andinos y pensada para funcionar en zonas rurales con baja conectividad.    * Fuente oficial: [https://plantvillage.psu.edu/?utm\_source](https://plantvillage.psu.edu/?utm_source=chatgpt.com) 2. **Plantix (App, 2020, Alemania/India)**    * Permite identificar más de 400 enfermedades y deficiencias en cultivos.    * Diferencia: requiere conexión a internet y no está adaptada al contexto rural del Perú.    * Fuente oficial: [https://plantix.net/en/?utm\_source](https://plantix.net/en/?utm_source=chatgpt.com) |

## Propuesta 3 (Andree)

## Propuesta 4 (Alejandro)

# 

# COMPARATIVA DE LAS PROPUESTAS

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Aspecto** | Propuesta 2: “Sistema Inteligente de Reconocimiento de Vehículos y Alerta Automática para la Gestión de Seguridad Vial en la Vía de Acceso a la Universidad Continental – Saylla, Cusco 2025” | Propuesta 1: “Implementación de un sistema de visión por computadora con CNN para diagnosticar enfermedades en hojas de cultivos andinos, logrando al menos 85% de precisión en comunidades agrícolas de Cusco” |  |  |
| **Problema Identificado** | Congestionamientos, incidentes viales y fugas de vehículos en la vía de acceso a la Universidad Continental (Saylla), dificultan la respuesta rápida de autoridades. En Cusco hubo 182 siniestros fatales en 2023 (2da región más alta del país). | En comunidades agrícolas del Cusco, los agricultores no identifican a tiempo las enfermedades en las hojas de cultivos, ocasionando pérdidas económicas, menor productividad y riesgo a la seguridad alimentaria. |  |  |
| **Oportunidad** | Uso de visión por computadora y reconocimiento automático de placas para apoyar a serenazgo/PNP en seguridad vial, con alertas en tiempo real y almacenamiento de evidencia. | Aplicar visión por computadora para ofrecer a pequeños agricultores una herramienta tecnológica accesible, que permita diagnósticos rápidos y confiables, incluso en zonas rurales con baja conectividad. |  |  |
| **Solución Propuesta** | |  | | --- | | Sistema de cámaras + CNN/YOLO + OCR (EasyOCR) para reconocer placas y vehículos, generar alertas automáticas y enviar notificaciones a autoridades. Incluye panel de monitoreo web y base de datos histórica. | | Un sistema basado en redes neuronales convolucionales (CNN) que analice imágenes de hojas (papa, maíz, tomate), determine si están sanas o enfermas, identifique la enfermedad y brinde recomendaciones iniciales con precisión ≥ 85%. |  |  |
| **Requerimientos Funcionales** | - Detectar y reconocer vehículos en tiempo real.  - Identificar placas automáticamente.  - Clasificar tipo de vehículo (auto, moto, bus, camión).  - Registrar evidencia (foto, hora, ubicación, placa).  - Generar alertas automáticas en caso de fuga o sospecha.  - Enviar notificaciones en tiempo real (SMS/app web).  - Acceso a panel de monitoreo.  - Guardar reportes históricos.  - Integración con sistemas de seguridad existentes.  - Funcionar 24/7 con autodiagnóstico básico. | - Capturar o cargar imágenes de hojas.  - Detectar si está sana o enferma.  - Identificar tipo de enfermedad.  - Mostrar recomendaciones iniciales.  - Guardar historial de diagnósticos.  - Interfaz web accesible.  - Permitir actualización del modelo.  - Generar reportes por zona.  - Incluir nuevos cultivos.  - Funcionar offline. |  |  |
| **Requerimientos Tecnológicos** | **Hardware:**  - Laptop i7 10ma gen+, RAM 16GB, GPU RTX 3070, SSD 512GB + 1TB extra.  - Cámaras IP (1080p, 30fps), router, UPS.  **Software:**  - Python 3.10+.  - OpenCV, TensorFlow/PyTorch, YOLOv8, EasyOCR.  - Flask (API REST).  - PostgreSQL/MySQL.  - HTML, CSS, JS (frontend).  - API Telegram, Twilio/SMTP (alertas). | - **Hardware:** PC x86/x64, GPU ≥ RTX 3060 (8GB VRAM), RAM ≥16GB, SSD ≥200GB, smartphones, servidor en la nube.  - **Software:** Python, JavaScript, TensorFlow/Keras, PyTorch, OpenCV, Pillow, Albumentations, Flask/FastAPI, GitHub, Colab, VS Code, Jupyter. |  |  |
| **Plataforma Backend** | Flask para API REST y microservicios, integrando panel de monitoreo y notificaciones. | Flask para la API de predicción y gestión de diagnósticos. |  |  |
| **Tecnologías de Mapa** | |  | | --- | | Integración con mapas de calor en panel web para mostrar zonas de mayor frecuencia de incidentes. (LeafletJS / Google Maps API). | | Uso de mapas interactivos (ej. LeafletJS o Google Maps API) para mostrar distribución geográfica de enfermedades detectadas en cada zona. |  |  |
| **Gamificación** | |  | | --- | | Puede implementarse como incentivos a conductores responsables (ej. puntos por cumplimiento de normas en campañas universitarias). | | No aplica de manera directa, pero podría añadirse un módulo opcional con insignias o logros para agricultores que registren diagnósticos frecuentes o adopten prácticas recomendadas. |  |  |
| **Impacto Social** | * Incrementa la seguridad vial en la zona universitaria, protege a peatones y conductores, y fortalece la capacidad de respuesta de autoridades locales. | Mejora la productividad agrícola, reduce pérdidas económicas, contribuye a la seguridad alimentaria y democratiza el acceso a tecnologías de IA en comunidades rurales del Cusco. |  |  |
| **Competencia y Avances Similares** | - **Deep License Plate Recognition (ParkPow, 2021-2023):** reconocimiento de placas en tiempo real. Diferencia: nuestra propuesta integra alertas inmediatas y clasificación de vehículos.  - **Laroca et al. (2021, YOLO-based ANPR):** alta precisión en detección de placas, pero sin conexión a sistemas locales. Diferencia: adaptado al contexto de Cusco, con panel y notificaciones en tiempo real. | |  | | --- | | - **PlantVillage (2019):** Diagnóstico de cultivos con IA, pero requiere internet y no está enfocado en cultivos andinos.  - **Plantix (2020):** Detecta más de 400 enfermedades, pero no está adaptada al Perú ni funciona offline.  Diferencia: propuesta enfocada en cultivos andinos, soporte offline y adaptación local. | |  |  |

# 

# EVALUACION Y ANALISIS GRUPAL

| **Nombre del Proyecto** | **Conocimientos Teóricos y Técnicos Referentes a lo Planteado** | **Impacto de la Solución a Implementar** | **Preferencia Personal por el Tema** | **Factibilidad Técnica de la Solución** | **Total** |
| --- | --- | --- | --- | --- | --- |
| **Propuesta 1:** | alejandro  Yersson  andre4  Yersson 5  total:  13 | alejandro  Yersson  andre3  total: 12 | alejandro  Yersson  andre3  total: 8 | alejandro  Andre 3  Yersson 4 total: 11 | 36 |
| **Propuesta 2:** | alejandro  andre4  Yersson 5  total: 10 | alejandro  andre3  Yersson 4  total: 10 | fabricio1  andre3  Yersson 2  total: 6 | alejandro  andre4  Yersson 1  total: 8 | 34 |
| **Propuesta 3:** | alejandro  andre3  Yersson 4  total: 12 | alejandro  andre3  Yersson 3  total: 11 | alejandro  Yersson 1  total: 5 | alejandro  andre4  Yersson 2  total: 10 | 33 |
| **Propuesta 4:** | alejandro  andre3  Yersson 2  total: 9 | alejandro  andre4  Yersson 4  total: 11 | alejandro  andre3  Yersson 2  total: 5 | alejandro  andre3  Yersson 2  total: 10 | 35 |

# JUSTIFICACIÓN E IMPORTANCIA

# DIAGRAMA DE ISHIKAWA BASADO EN ERRORES

# DESARROLLO DE LA METODOLOGÍA ÁGIL

## Historias de usuario

## Estimación de tiempos

## Estimación de épicas

## Product backlog según prioridades y estimaciones

## Cronograma: diagrama de gantt