|  |  |  |  |
| --- | --- | --- | --- |
| ***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:** Andree

**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 4 (Alejandro)

# 

# COMPARATIVA DE LAS PROPUESTAS

|  |  |  |  |
| --- | --- | --- | --- |
| **Aspecto** | Propuesta 1: “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 2 “Implementación de sistema de visión por computadora con CNN para diagnóstico de enfermedades en hojas de cultivos andinos en comunidades 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 2  Yersson 5  total:  7 | Alejandro 4  Yersson 4  total: 8 | Alejandro 2  Yersson 4  total: 6 | Alejandro 3  Yersson 2  total: 5 | 26 |
| **Propuesta 2:** | Alejandro 3  Andre 3  total: 6 | Alejandro 5  Andre 5  total: 10 | Fabricio 5  Andre 5  total: 10 | Alejandro 4  Andre 4  total: 8 | 34 |
| **Propuesta 3:** | Andre 2  Yersson 3  total: 5 | Andre 3  Yersson 2  total: 5 | Andre 2  Yersson 1  total: 3 | Andre 2  Yersson 3  total: 5 | 18 |

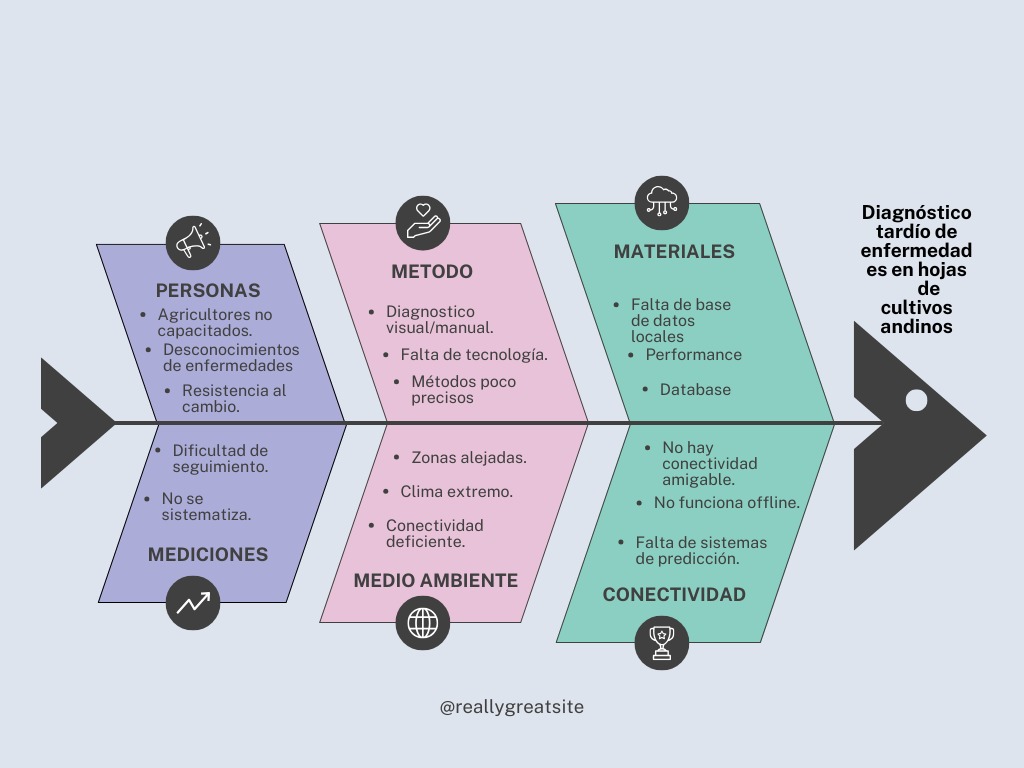
# JUSTIFICACIÓN E IMPORTANCIA

Desde el punto de vista de la Gestión de Proyectos, la selección de esta propuesta responde a su viabilidad, impacto social y alineación con los Objetivos de Desarrollo Sostenible (ODS). El proyecto de detección temprana de enfermedades en cultivos mediante redes neuronales convolucionales se ejecutará considerando buenas prácticas de planificación, control y gestión de riesgos, lo que permitirá optimizar recursos limitados (tiempo, presupuesto y alcance) y garantizar entregables en menos de 10 semanas. Además, la relevancia del problema asegura un alto valor agregado para las comunidades rurales, donde la reducción de pérdidas agrícolas se traduce directamente en mejoras económicas y de seguridad alimentaria.

Desde el punto de vista de la Ingeniería de Software, se desarrollará un modelo de inteligencia artificial basado en CNN implementado en frameworks como PyTorch y TensorFlow, utilizando técnicas de preprocesamiento y aumento de datos para garantizar la precisión del sistema. El proyecto implica fases propias de la disciplina: análisis de requerimientos, diseño, construcción, pruebas y despliegue de un prototipo funcional. Se busca no solo generar un producto mínimo viable (MVP) con resultados cuantitativos (precisión, recall y exactitud del modelo), sino también sentar las bases para futuras ampliaciones, como la integración en aplicaciones móviles accesibles a los agricultores.

Desde nuestro punto de vista como equipo, seleccionamos esta idea porque une innovación tecnológica con un impacto social real. Entre las alternativas evaluadas, esta propuesta nos motiva por su contribución directa a la solución de una problemática que afecta tanto a comunidades andinas como al país en general. Además, nos permite aplicar de manera práctica los conocimientos adquiridos en inteligencia artificial, visión por computadora y gestión de proyectos. Creemos que, más allá de la calificación académica, este trabajo puede ser un punto de partida hacia soluciones más amplias que apoyen al agricultor peruano frente a los retos de las plagas y el cambio climático.

# DIAGRAMA DE ISHIKAWA BASADO EN ERRORES



# 

# DESARROLLO DE LA METODOLOGÍA ÁGIL

## Historias de usuario

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Historia de Usuario** | **Criterios de Aceptación** | **Prioridad** | **Estimación** | **Observaciones** |
| HU1 | Como agricultor, quiero tomar una foto de la hoja de mi cultivo desde el celular, para detectar si está sana o enferma. | La app debe permitir capturar imagen desde cámara o galería y mostrar resultado inmediato. | Alta | 5 pts | Requiere cámara funcional y buena iluminación. |
| HU2 | Como agricultor, quiero recibir el nombre de la enfermedad detectada, para saber qué está afectando mi cultivo. | El sistema debe mostrar el nombre de la enfermedad con al menos 85% de precisión. | Alta | 3 pts | Depende del modelo CNN entrenado. |
| HU3 | Como agricultor, quiero ver recomendaciones iniciales de manejo, para actuar rápido y reducir pérdidas en mi producción. | La app debe mostrar al menos una acción sugerida para cada enfermedad detectada. | Alta | 5 pts | Requiere asesoría técnica para recomendaciones. |
| HU4 | Como usuario rural, quiero usar la aplicación en modo offline, para diagnosticar mis cultivos sin conexión. | El sistema debe funcionar con un modelo descargado y guardar datos localmente. | Alta | 8 pts | Importante en comunidades con baja conectividad. |
| HU5 | Como agricultor, quiero almacenar un historial de diagnósticos, para hacer seguimiento a mis cultivos. | La app debe guardar fecha, imagen y resultado del diagnóstico. | Media | 5 pts | Puede requerir almacenamiento local o en la nube. |
| HU6 | Como técnico agrícola, quiero generar un reporte de las enfermedades más frecuentes en la zona, para planificar acciones preventivas. | El sistema debe exportar reportes en PDF o Excel con estadísticas. | Media | 8 pts | Orientado a usuarios avanzados. |
| HU7 | Como administrador del sistema, quiero actualizar el modelo CNN con nuevos datos, para mejorar la precisión de los diagnósticos. | La app debe permitir subir nuevos modelos entrenados y reemplazar el anterior. | Alta | 13 pts | Necesita pipeline de entrenamiento. |
| HU8 | Como agricultor, quiero recibir alertas cuando aparezcan enfermedades recurrentes en mi zona, para prevenir contagios. | El sistema debe generar alertas según frecuencia de diagnósticos en la comunidad. | Media | 8 pts | Puede requerir GPS o ubicación aproximada. |
| HU9 | Como agricultor, quiero que la app detecte si la hoja pertenece al cultivo correcto, para evitar diagnósticos erróneos. | El sistema debe verificar que la hoja pertenece al cultivo seleccionado antes de clasificar. | Alta | 5 pts | Previene errores de clasificación cruzada. |
| HU10 | Como agricultor, quiero poder elegir el tipo de cultivo antes de analizar la hoja, para recibir un diagnóstico más preciso. | La app debe desplegar lista de cultivos disponibles antes del análisis. | Alta |  | Mejora la clasificación del modelo. |
| HU11 | Como agricultor, quiero compartir mis diagnósticos con un técnico agrícola mediante WhatsApp o correo, para recibir asesoría personalizada. | La app debe permitir compartir resultados en formato imagen o PDF desde el celular. | Media | 5 pts | Requiere integración con apps externas. |
| HU12 | Como agricultor, quiero ver ejemplos visuales de hojas sanas y enfermas, para comparar con mis cultivos. | La app debe mostrar imágenes de referencia junto al resultado del diagnóstico. | Alta | 5 pts | Ayuda al aprendizaje del usuario. |
| HU13 | Como usuario, quiero poder cambiar el idioma de la aplicación (ej. quechua/español), para usarla en mi idioma preferido. | La app debe permitir cambiar el idioma desde configuración. | Media | 8 pts | Mejora accesibilidad en comunidades rurales. |
| HU14 | Como técnico agrícola, quiero acceder a un panel web con estadísticas agregadas, para analizar tendencias regionales. | El sistema debe mostrar gráficos y datos consolidados por cultivo y zona. | Media | 13 pts | Útil para instituciones o municipalidades. |
| HU15 | Como agricultor, quiero recibir consejos preventivos de cuidado de mis cultivos, para reducir la probabilidad de enfermedades. | La app debe mostrar tips generales según el cultivo registrado. | Media | 3 pts | Se puede basar en información agronómica. |
| HU16 | Como investigador, quiero visualizar una matriz de confusión de los diagnósticos, para evaluar la precisión del modelo CNN. | El sistema debe generar métricas de rendimiento como precisión, recall y matriz de confusión. | Media | 8 pts | Útil para validar el desempeño científico del modelo. |
| HU17 | Como agricultor, quiero poder adjuntar notas o comentarios a cada diagnóstico, para llevar un mejor registro de mis cultivos. | La app debe permitir añadir texto asociado al historial de diagnósticos. | Media | 5 pts | Útil para seguimiento personalizado. |
| HU18 | Como administrador, quiero gestionar la lista de cultivos disponibles en la app, para asegurar que solo se muestren los más relevantes. | El sistema debe permitir activar/desactivar cultivos en el menú. | Media | 5 pts | Mantiene la app enfocada. |
| HU19 | Como agricultor, quiero que el sistema me indique el nivel de severidad de la enfermedad en la hoja, para priorizar acciones. | La app debe clasificar la enfermedad en leve, moderada o grave. | Alta | 8 pts | Requiere entrenamiento adicional con etiquetas de severidad. |
| HU20 | Como agricultor, quiero recibir notificaciones cuando la app detecte una nueva enfermedad que no está en mi historial, para actuar de inmediato. | El sistema debe enviar notificaciones push en el dispositivo. | Alta | 5 pts | Requiere conexión ocasional a internet. |
| HU21 | Como investigador, quiero exportar los datos anonimizados de diagnósticos, para analizar patrones de propagación de enfermedades. | El sistema debe permitir exportar registros en CSV/Excel sin datos personales. | Baja | 8 pts | Aporta al análisis científico. |
| HU22 | Como agricultor, quiero que la app me muestre el porcentaje de confianza del diagnóstico, para decidir si debo consultar a un técnico. | La app debe mostrar el nivel de certeza (ej. 90%). | Alta | 3 pts | Transparencia para el usuario. |
| HU23 | Como técnico agrícola, quiero filtrar reportes por cultivo y enfermedad, para identificar zonas críticas. | El panel debe permitir aplicar filtros dinámicos sobre los datos. | Media | 5 pts | Complementa las estadísticas globales. |
| HU24 | Como agricultor, quiero poder subir varias fotos de diferentes hojas del mismo cultivo, para tener un diagnóstico más confiable. | La app debe analizar múltiples imágenes antes de dar un resultado final. | Alta | 8 pts | Mejora precisión de diagnósticos. |
| HU25 | Como agricultor, quiero que la app funcione en dispositivos de bajo costo, para no depender de celulares caros. | La app debe estar optimizada para funcionar en móviles con 2GB RAM. | Alta | 5 pts | Importante para adopción rural. |
| HU26 | Como agricultor, quiero que el sistema me sugiera si debo aplicar fertilizantes o pesticidas básicos, para dar el primer paso en el tratamiento. | La app debe mostrar recomendaciones básicas y no invasivas. | Media | 5 pts | Requiere validación con expertos agrícolas. |
| HU27 | Como administrador, quiero hacer copias de seguridad automáticas de la base de datos, para garantizar la seguridad de la información. | El sistema debe tener backups periódicos configurados. | Alta | 8 pts | Importante para mantenimiento. |
| HU28 | Como técnico agrícola, quiero que la app almacene imágenes anónimas de diagnósticos confirmados, para mejorar futuros modelos. | El sistema debe guardar imágenes clasificadas de manera opcional y anónima. | Media | 5 pts | Sirve como retroalimentación para reentrenar el modelo. |
| HU29 | Como agricultor, quiero que la app me indique si la hoja analizada es ilegible o tiene mala calidad, para repetir la captura. | El sistema debe mostrar advertencia en caso de imágenes borrosas o incompletas. | Alta | 5 pts | Evita diagnósticos erróneos. |
| HU30 | Como agricultor, quiero poder imprimir mis reportes de diagnóstico, para archivarlos y mostrarlos en reuniones comunales. | La app debe exportar reportes en formato PDF con opción de impresión. | Media | 5 pts | Útil en capacitaciones y talleres. |

## Estimación de tiempos

1. **Estimación inicial:**

* Total de puntos de historia de usuario: 190 pts (suma de todas tus HU).
* Conversión adoptada: 1 punto = 1 hora de esfuerzo efectivo.
* Estimación total: 190 × 1 hora = 190 horas.

1. **Capacidad del Equipo por Semana:**

* Número de integrantes: 3
* Horas disponibles por persona: 4 horas/día × 5 días = 20 horas/semana/persona
* Capacidad total del equipo: 20 × 3 = 60 horas/semana

1. **Verificación para ≤ 10 Semanas:**

* Total de horas requeridas: 190 horas
* Capacidad total en 10 semanas: 60 × 10 = 600 horas
* Resultado: El equipo tiene más que suficiente capacidad para completar el proyecto en 10 semanas.

1. **Distribución del Trabajo:**

* Horas por semana necesarias: 190 ÷ 10 = 19 horas/semana
* Horas por persona por semana: 19 ÷ 3 = 6.3 horas/semana/persona
* Cada integrante del equipo necesita dedicar en promedio ~6 a 7 horas por semana, lo cual es factible dentro de la carga académica.

## Estimación de épicas

**Épica 1: Captura y Diagnóstico de Imágenes (Core CNN)**

* HU1: Captura de foto de hoja.
* HU2: Mostrar nombre de enfermedad.
* HU9: Verificación de cultivo correcto.
* HU10: Selección de cultivo antes del análisis.
* HU22: Porcentaje de confianza en el diagnóstico.
* HU24: Análisis de múltiples imágenes.
* HU29: Validación de calidad de imagen.

**Épica 2: Recomendaciones y Apoyo al Agricultor**

* HU3: Recomendaciones iniciales.
* HU12: Compartir diagnóstico con técnico.
* HU15: Ejemplos visuales de hojas sanas/enfermas.
* HU18: Consejos preventivos.
* HU26: Sugerencias básicas de fertilización/pesticidas.
* HU19: Clasificación de severidad de la enfermedad.

**Épica 3: Funcionalidades Offline y Usabilidad en Campo**

* HU4: Modo offline.
* HU5: Guardar historial de diagnósticos.
* HU17: Añadir notas a diagnósticos.
* HU25: Optimización para móviles de bajo costo.
* HU30: Exportar reportes imprimibles.
* HU13: Cambio de idioma (Quechua/Español).

**Épica 4: Entrenamiento, Evaluación y Mantenimiento del Modelo CNN**

* HU7: Actualización del modelo CNN.
* HU14: Panel web de estadísticas básicas.
* HU16: Visualización de métricas (matriz de confusión, precisión, recall).
* HU23: Filtrar reportes por cultivo/enfermedad.
* HU27: Backups automáticos.
* HU28: Almacenamiento de imágenes anónimas para reentrenar.

**Épica 5: Alertas y Reportes Comunitarios**

* HU6: Reportes PDF/Excel.
* HU8: Alertas por enfermedades recurrentes.
* HU20: Notificaciones de nuevas enfermedades.
* HU11: Exportar dataset de imágenes.
* HU21: Exportar datos anonimizados.

**Planificación (≤ 10 semanas)**

|  |  |  |
| --- | --- | --- |
| **Semana** | **Actividad Principal** | **Épicas / HU involucradas** |
| 1 | Preparación: diseño de arquitectura, setup de dataset (39 categorías), entorno de entrenamiento (GPU, PyTorch). | Épica 4 (HU7, HU16 setup) |
| 2 | Implementación básica del modelo CNN (sano/enfermo + verificación de cultivo) | Épica 1 (HU1, HU2, HU9) |
| 3 | Clasificación por enfermedad + confianza en diagnóstico | Épica 1 (HU10, HU22) |
| 4 | Múltiples imágenes + validación de calidad | Épica 1 (HU24, HU29) |
| 5 | Recomendaciones iniciales + ejemplos visuales | Épica 2 (HU3, HU15) |
| 6 | Consejos preventivos + severidad de enfermedad + fertilización básica | Épica 2 (HU18, HU19, HU26) |
| 7 | Modo offline + historial de diagnósticos + notas | Épica 3 (HU4, HU5, HU17) |
| 8 | Optimización móviles + exportación de reportes + idioma Quechua/Español | Épica 3 (HU25, HU30, HU13) |
| 9 | Mantenimiento del modelo: actualización + métricas + backups + dataset anónimo | Épica 4 (HU7, HU16, HU27, HU28) |
| 10 | Alertas comunitarias y reportes para investigadores | Épica 5 (HU6, HU8, HU20, HU11, HU21) |

## Product backlog según prioridades y estimaciones

**Ordenado de mayor a menor prioridad.**

**Prioridad Alta (MVP – lo esencial para que funcione la app)**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Historia de Usuario** | **Estimación** | **Observaciones** |
| HU1 | Tomar foto de la hoja desde celular (cámara/galería). | 5 pts | Base del sistema |
| HU2 | Mostrar el nombre de la enfermedad detectada. | 3 pts | Núcleo del diagnóstico |
| HU9 | Verificar que la hoja corresponda al cultivo correcto. | 5 pts | Evita diagnósticos cruzados |
| HU10 | Elegir el tipo de cultivo antes del análisis. | 5 pts | Mejora precisión del modelo |
| HU22 | Mostrar el porcentaje de confianza del diagnóstico. | 3 pts | Transparencia al usuario |
| HU12 | Mostrar ejemplos visuales de hojas sanas y enfermas. | 5 pts | Refuerzo educativo |
| HU19 | Mostrar el nivel de severidad de la enfermedad (leve, grave, etc.). | 8 pts | Mejora en la toma de decisiones |
| HU24 | Analizar múltiples fotos de un mismo cultivo para diagnóstico más confiable. | 8 pts | Aumenta precisión |
| HU29 | Indicar si la hoja tiene mala calidad (borrosa/incompleta). | 5 pts | Evita diagnósticos erróneos |
| HU25 | Optimización para celulares de bajo costo (2GB RAM). | 5 pts | Inclusión digital |
| HU20 | Notificación inmediata ante nueva enfermedad en el cultivo. | 5 pts | Acciones rápidas |
| HU4 | Uso en modo offline (modelo descargado, datos locales). | 8 pts | Clave en zonas rurales |
| HU7 | Actualizar el modelo CNN con nuevos datos. | 13 pts | Mantenimiento esencial |
| HU27 | Configurar copias de seguridad automáticas del sistema. | 8 pts | Seguridad de información |

**Prioridad Media (funcionalidades de apoyo y valor agregado)**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Historia de Usuario** | **Estimación** | **Observaciones** |
| HU3 | Recomendaciones iniciales de manejo por enfermedad. | 5 pts | Acción rápida |
| HU15 | Consejos preventivos de cuidado de cultivos. | 3 pts | Prevención |
| HU26 | Sugerencias básicas de fertilización/pesticidas. | 5 pts | Requiere validación agronómica |
| HU5 | Historial de diagnósticos con fecha e imagen. | 5 pts | Seguimiento del agricultor |
| HU17 | Adjuntar notas o comentarios a diagnósticos. | 5 pts | Personalización |
| HU13 | Cambiar idioma (Quechua/Español). | 8 pts | Accesibilidad |
| HU6 | Generar reportes PDF/Excel con estadísticas. | 8 pts | Orientado a técnicos |
| HU11 | Compartir diagnósticos con un técnico (WhatsApp/correo). | 5 pts | Requiere integración externa |
| HU23 | Filtros por cultivo y enfermedad en reportes. | 5 pts | Herramienta analítica |
| HU16 | Visualizar matriz de confusión y métricas del modelo. | 8 pts | Evaluación científica |
| HU28 | Guardar imágenes anónimas para mejorar dataset. | 5 pts | Retroalimentación al modelo |
| HU18 | Administrar lista de cultivos disponibles. | 5 pts | Mantener foco en app |
| HU30 | Imprimir reportes en PDF para uso comunal. | 5 pts | Útil en talleres |
| HU8 | Alertas comunitarias por enfermedades recurrentes. | 8 pts | Impacto social |
| HU14 | Panel web con estadísticas regionales y gráficos. | 13 pts | Uso institucional |

**Prioridad Baja (funcionalidades futuras / secundarias)**

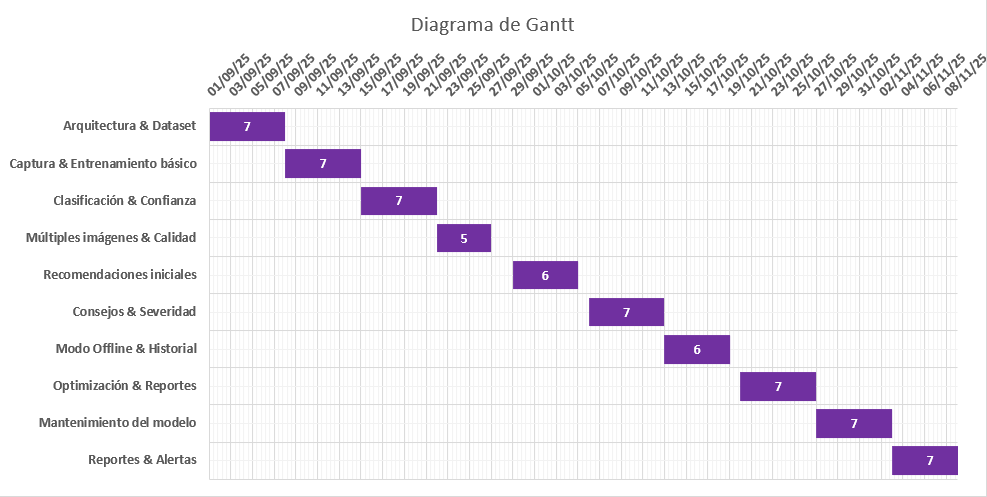
|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Historia de Usuario** | **Estimación** | **Observaciones** |
| HU21 | Exportar datos anonimizados para investigación científica. | 8 pts | Aporta a ciencia |
| HU30 | Exportar diagnósticos en formato Excel o PDF (duplicado en HU30, ajustar si necesario). | 5 pts | Funcionalidad extra |
| HU11 | Notificaciones de nuevas enfermedades (ya cubierta parcialmente en HU20). | 2-3 pts | Se puede integrar más adelante |

## 

## Cronograma: diagrama de Gantt

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nombre Actividad** | **Actividades** | **Inicio** | **Días** | **Final** | **Épicas/HU** |
| Arquitectura & Dataset | Definición de arquitectura del sistema; Preparación del dataset (39 categorías); Configuración de entorno de entrenamiento (GPU, PyTorch) | 01/09/2025 | 7 | 07/09/2025 | Épica 4 (HU7, HU16 setup) |
| Captura & Entrenamiento básico | Implementar módulo de captura de imágenes (cámara/galería); Entrenamiento inicial del CNN (sano/enfermo); Verificación de cultivo correcto | 08/09/2025 | 7 | 14/09/2025 | Épica 1 (HU1, HU2, HU9) |
| Clasificación & Confianza | Extender modelo CNN para clasificación por enfermedad; Mostrar nombre de enfermedad detectada; Implementar porcentaje de confianza | 15/09/2025 | 7 | 21/09/2025 | Épica 1 (HU10, HU22) |
| Múltiples imágenes & Calidad | Incorporar análisis de múltiples imágenes por cultivo; Validar calidad de imágenes capturadas (borrosas/incompletas) | 22/09/2025 | 5 | 28/09/2025 | Épica 1 (HU24, HU29) |
| Recomendaciones iniciales | Agregar módulo de recomendaciones iniciales; Implementar ejemplos visuales de hojas sanas/enfermas | 29/09/2025 | 6 | 05/10/2025 | Épica 2 (HU3, HU15) |
| Consejos & Severidad | Incluir consejos preventivos por cultivo; Clasificación de severidad (leve, moderada, grave); Sugerencias básicas de fertilización/pesticidas | 06/10/2025 | 7 | 12/10/2025 | Épica 2 (HU18, HU19, HU26) |
| Modo Offline & Historial | Implementar modo offline; Guardar historial de diagnósticos; Añadir notas en diagnósticos | 13/10/2025 | 6 | 19/10/2025 | Épica 3 (HU4, HU5, HU17) |
| Optimización & Reportes | Optimización de la app para móviles de bajo costo (2GB RAM); Exportación de reportes PDF/impresión; Incorporar idioma Quechua/Español | 20/10/2025 | 7 | 26/10/2025 | Épica 3 (HU25, HU30, HU13) |
| Mantenimiento del modelo | Implementar actualización del modelo CNN; Generar métricas de rendimiento (matriz de confusión, precisión, recall); Configurar backups automáticos; Guardar imágenes anónimas | 27/10/2025 | 7 | 02/11/2025 | Épica 4 (HU7, HU16, HU27, HU28) |
| Reportes & Alertas | Generar reportes PDF/Excel para técnicos e instituciones; Implementar alertas comunitarias por brotes; Notificaciones de nuevas enfermedades; Exportar dataset/datos anonimizados | 03/11/2025 | 7 | 09/11/2025 | Épica 5 (HU6, HU8, HU20, HU11, HU21) |

**Diagrama de Gantt:**



REFERENCIAS:

1. LOZADA-PORTILLA, H., et al. Clasificación de enfermedades en hojas de papa utilizando técnicas de aprendizaje profundo con CNN y modelos híbridos. 2025. Disponible en: <http://scielo.senescyt.gob.ec/scielo.php?script=sci_arttext&pid=S2631-26542025000100142>
2. FERNANDEZ FERNANDEZ, K.D. y PINGLO CABEZAS, W.R. Redes neuronales convolucionales para la detección de enfermedades en cultivos de arroz: una revisión sistemática. 2025. Disponible en: <https://repositorio.uss.edu.pe/bitstream/handle/20.500.12802/14298/Fernandez%20Fernandez%20Kenedy%20David%20&%20Pinglo%20Cabezas%20Williams%20Rafael.pdf?sequence=1&isAllowed=y>
3. DIALNET. Detección de enfermedades en cultivos de papa usando procesamiento digital de imágenes. Disponible en: <https://dialnet.unirioja.es/descarga/articulo/8228807.pdf>
4. FLYPIX.AI. Detección de enfermedades de cultivos impulsada por IA. Disponible en: <https://flypix.ai/es/blog/crop-disease-detection/>
5. UNIVERSIDAD DE LOS ANDES. Sistema de visión por computadora para la monitorización y gestión de cultivos con visión artificial. Disponible en: <https://repositorio.uniandes.edu.co/entities/publication/05a1c5a2-b619-447b-9883-24fb71ebdf27>
6. MIOTI. Agricultura Inteligente: El poder del Computer Vision. 2024. Disponible en: <https://mioti.es/es/blog-agricultura-inteligente-el-poder-del-computer-vision/>
7. ACOFI. Sistemas de visión por computadora aplicados a cultivos tropicales con UAV. Disponible en: <https://acofipapers.org/index.php/eiei/article/download/2270/1807/5809>
8. PRATEC. Agricultura y cultura andina: tecnologías campesinas aplicadas a la agricultura andina. Disponible en: <https://www.pratec.org/wpress/pdfs-pratec/agricultura-y-cultura-andina-final.pdf>
9. FLORES MENDOZA, J.C. y MEJÍA CARHUAJULCA, G. Agricultura y cultura andina. Universidad Nacional Pedro Ruiz Gallo. Disponible en: <https://repositorio.unprg.edu.pe/bitstream/handle/20.500.12893/11478/Flores_Mendoza_Juan_Carlos%20y%20Mejia_Carhuajulca_Gianfranco.pdf?sequence=1&isAllowed=y>
10. BANCO INTERAMERICANO DE DESARROLLO (BID). Agricultura de precisión: tecnología para enfrentar el cambio climático. 2024. Disponible en: <https://blogs.iadb.org/innovacion/es/agricultura-de-precision-tecnologia-para-enfrentar-el-cambio-climatico/>