FASE 3

DISEÑO E IMPLEMENTACIÓN DEL PROYECTO

NEXA

Escuela de Informática y Telecomunicaciones

Noviembre 2025

1. Identificación del Proyecto

|  |
| --- |
| Nombre de Proyecto |
| Nexa – Gestión Logística y de Abastecimiento de Gendarmería |

1. Integrantes del Equipo de Trabajo

|  |  |  |  |
| --- | --- | --- | --- |
| **N°** | **Rut** | **Apellidos** | **Nombres** |
| 1 | 20913129-3 | Guerrero Ayala | Yeremi Andres |
| 2 | 21524701-5 | Campusano Olivares | Tihare Betzabé |
| 3 | 18841597-0 | Yáñez Salinas | Octavio Yáñez |

1. Registro de Control de Cambios

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Revisión** | **Fecha** | **Paginas** | **Descripción del Cambio** | **Autor** |
| 1 | 15-09-2024 | Todas | Primera revisión del informe del proyecto | TCA |
| 2 | 22-09-2025 | Todas | Segunda revisión del informe del proyecto | TCA |
| 3 | 14-10-2025 | Todas | Revisión y corrección final del informe completo | TCA |

**INDICE DE CONTENIDO**

[LISTA DE TABLAS 3](#_Toc415423693)

[LISTA DE FIGURAS 4](#_Toc620546246)

[LISTA DE DIAGRAMAS 5](#_Toc1499131903)

[GLOSARIO 5](#_Toc81959362)

[1 Diseño e Implementación del Proyecto 6](#_Toc319069262)

[1.1 Resumen 7](#_Toc102269412)

[1.2 Abstrac 7](#_Toc809132082)

[1.3 Introducción 8](#_Toc1773070797)

[1.4 Problemática a solucionar o necesidad a satisfacer 9](#_Toc2015430125)

[1.5 Objetivos del Proyecto (general y específicos) 10](#_Toc2019018909)

[1.6 Competencias del Perfil de Egreso 10](#_Toc1305873086)

[1.7 Asignación de roles 11](#_Toc406855105)

[1.8 Metodología utilizada en el Proyecto. 12](#_Toc504873239)

[1.9 Creación de cronograma asociado al Proyecto 17](#_Toc1685206775)

[1.10 Riesgos Asociados al Proyecto 19](#_Toc1975037698)

[1.11 Implementación del Proyecto 22](#_Toc539956490)

[Diseño y Arquitectura de la solución 22](#_Toc1109600155)

[Requerimientos técnicos y Diagramas 25](#_Toc1526070389)

[Resultados de la solución 33](#_Toc2074828999)

[1.12 Definición de Recursos y Costos asociados al Proyecto 37](#_Toc755949307)

[CONCLUSIÓN 39](#_Toc824870492)

[BIBLIOGRAFÍA 40](#_Toc1834621930)

[ANEXOS 42](#_Toc2077864768)

# LISTA DE TABLAS

[Tabla 1 Proyecto General 15](#_Toc210402372)

[Tabla 2 Proyecto General 16](#_Toc210402373)

[Tabla 3 Sprint 1 17](#_Toc210402374)

[Tabla 4 Sprint 2 17](#_Toc210402375)

[Tabla 5 Sprint 3 18](#_Toc210402376)

[Tabla 6 Sprint 4 18](#_Toc210402377)

[Tabla 7 Sprint 5 18](#_Toc210402378)

[Tabla 8 Sprint 6 19](#_Toc210402379)

[Tabla 9 Costos 1 43](#_Toc210402380)

[Tabla 10 Costos 2 44](#_Toc210402381)

# LISTA DE FIGURAS

[Ilustración 1 Carta Gantt 18](#_Toc210050162)

[Ilustración 2 Riesgos asociados al proyecto 21](#_Toc210050163)

[Ilustración 3 Herramientas del proyecto 21](#_Toc210050164)

# LISTA DE DIAGRAMAS

[Diagrama 1 Arquitectura IA 22](#_Toc210050165)

[Diagrama 2 Arquitectura del Proyecto 23](#_Toc210050166)

[Diagrama 3 Actividades 24](#_Toc210050167)

[Diagrama 4 Entidad-Relación 25](#_Toc210050168)

[Diagrama 5 Clases 26](#_Toc210050169)

[Diagrama 6 Componentes 27](#_Toc210050170)

[Diagrama 7 Despliegue 28](#_Toc210050171)

[Diagrama 8 Flujo 29](#_Toc210050172)

[Diagrama 9 Casos de uso 30](#_Toc210050173)

# GLOSARIO

**Scrum**: Metodología ágil de gestión de proyectos que organiza el trabajo en ciclos cortos llamados sprints, fomentando la colaboración, adaptación al cambio y mejora continua.

**Sprint**: Iteración de trabajo en Scrum con una duración definida, durante la cual se planifica, desarrolla y entrega un incremento del producto.

**Product** **Owner** (PO): Rol de Scrum encargado de representar al cliente y priorizar el backlog del producto.

**Scrum** **Master**: Rol de Scrum que facilita el cumplimiento de la metodología, eliminando impedimentos y asegurando la comunicación efectiva del equipo.

**Developer**: Rol del equipo Scrum que se encarga de diseñar, programar y probar las funcionalidades del sistema.

**Backlog del producto**: Lista priorizada de tareas, funcionalidades y mejoras que debe incluir un proyecto de software.

**API** **REST**: Interfaz de programación que permite la comunicación entre aplicaciones mediante peticiones HTTP.

**Supabase**: Plataforma de base de datos en la nube que ofrece autenticación, almacenamiento y servicios similares a Firebase.

**FastAPI**: Framework de desarrollo backend en Python, diseñado para crear APIs rápidas y seguras.

**Ionic** **React**: Framework para construir aplicaciones móviles híbridas usando React.

**Tailwind** CSS: Librería de estilos que facilita el diseño web responsivo y personalizable.

**Machine** **Learning** (ML): Rama de la inteligencia artificial que permite a los sistemas aprender y mejorar a partir de datos sin ser programados explícitamente.

**MobileNetV2**: Arquitectura de red neuronal ligera y eficiente utilizada en visión por computadora.

**Dataset**: Conjunto de datos estructurados que se utilizan para entrenar y validar modelos de inteligencia artificial.

**TensorFlow** **/** **Keras**: Librerías de código abierto utilizadas para construir y entrenar modelos de aprendizaje automático.

# Diseño e Implementación del Proyecto

## Resumen

El proyecto Nexa permitió modernizar de manera significativa la gestión de inventarios de insumos tecnológicos en Gendarmería de Chile, reemplazando los procesos manuales que tradicionalmente eran lentos, propensos a errores y poco seguros, por una plataforma digital confiable, escalable y con altos estándares de seguridad. La solución consistió en el desarrollo e implementación de un sistema integral que centralizó de forma eficiente el control de entradas y salidas de insumos, el registro de stock actualizado en tiempo real y la generación de alertas automáticas de reposición, lo que aseguró una gestión más transparente y organizada de los recursos. Este sistema integró además reportes automáticos personalizables que facilitaron la toma de decisiones estratégicas, incorporó múltiples niveles de seguridad para garantizar el acceso restringido a la información sensible, habilitó soporte para la utilización de códigos de barras con el fin de agilizar la identificación y control de productos, e incluyó un módulo de inteligencia artificial destinado al análisis predictivo y la optimización en la gestión de insumos tecnológicos. El proyecto fue desarrollado bajo la metodología ágil Scrum a lo largo de 18 semanas, organizando el trabajo en sprints que permitieron planificar, ejecutar y revisar avances de manera iterativa, fomentando la adaptación a cambios y la mejora continua en cada entrega. Asimismo, el desarrollo del sistema se alineó directamente con competencias propias de la Ingeniería en Informática, tales como diseño y programación de software, gestión de proyectos, seguridad informática, pruebas de calidad y despliegue de soluciones tecnológicas. También permitió potenciar intereses profesionales en áreas como el desarrollo fullstack, el liderazgo de equipos, la administración de metodologías ágiles y la validación de sistemas a través de testing funcional y automatizado. Como resultado, los funcionarios de Gendarmería contaron con una herramienta digital confiable y robusta que redujo considerablemente los errores humanos, optimizó los tiempos destinados a la gestión de inventarios, mejoró la trazabilidad de los insumos en todas las etapas del proceso y fortaleció la transparencia institucional. Todo esto se tradujo en una administración más eficiente de los recursos tecnológicos, la reducción de costos operativos, el aumento de la productividad interna y la consolidación de una cultura organizacional basada en el uso de la tecnología como soporte estratégico para la gestión pública, logrando un impacto positivo tanto en la labor diaria de los funcionarios como en los resultados globales de la institución.

## Abstrac

The Nexa project significantly modernized the management of technological supply inventories at the Chilean Gendarmerie, replacing the traditionally slow, error-prone, and insecure manual processes with a reliable, scalable digital platform built on high security standards. The solution consisted of the development and implementation of a comprehensive system that efficiently centralized the control of supply inputs and outputs, real-time stock updates, and automatic restocking alerts, ensuring a more transparent and organized management of resources. This system also integrated customizable automated reports to support strategic decision-making, incorporated multiple security levels to guarantee restricted access to sensitive information, enabled barcode support to streamline product identification and control, and included an artificial intelligence module designed for predictive analysis and optimization in the management of technological supplies. The project was developed under the agile Scrum methodology over the course of 18 weeks, organizing the work into sprints that made it possible to plan, execute, and review progress iteratively, fostering adaptability to changes and continuous improvement in each delivery. Furthermore, the system’s development was directly aligned with core competencies of Computer Engineering, such as software design and programming, project management, information security, quality assurance, and the deployment of technological solutions. It also reinforced professional interests in areas such as full-stack development, team leadership, the administration of agile methodologies, and system validation through both functional and automated testing. As a result, Gendarmerie officers gained access to a robust and reliable digital tool that considerably reduced human errors, optimized the time allocated to inventory management, improved supply traceability throughout all process stages, and strengthened institutional transparency. All of this translated into more efficient administration of technological resources, reduced operational costs, increased internal productivity, and the consolidation of an organizational culture grounded in the strategic use of technology to support public management, achieving a positive impact both on the daily work of officers and on the institution’s overall outcomes.

## Introducción

El proyecto Nexa surge como respuesta a la necesidad de modernizar los procesos de control y administración de insumos tecnológicos en Gendarmería de Chile. Actualmente, la institución enfrenta dificultades derivadas de la falta de organización y de un sistema de inventariado formal, ya que la recepción de insumos se realizaba sin un registro ordenado ni trazabilidad adecuada. Estas falencias impactan directamente en la eficiencia administrativa y en el uso responsable de los recursos disponibles.

En este contexto, Nexa propone el desarrollo de una plataforma digital segura, eficiente y escalable, que permita centralizar la gestión del inventario mediante la digitalización de los procesos de registro, control de entradas y salidas, generación de reportes, alertas de reposición y trazabilidad de recursos. Asimismo, incorpora tecnologías innovadoras como la inteligencia artificial para el análisis de imágenes, aumentando la precisión en el control del estado de los insumos. El proyecto se enmarca en un entorno público donde la gestión eficiente de recursos resulta crítica, y su implementación busca no solo optimizar tiempos y reducir errores, sino también aportar transparencia y fortalecer la eficiencia institucional. En definitiva, Nexa se presenta como una solución tecnológica alineada con las necesidades actuales de la organización y con el desarrollo profesional de las competencias en ingeniería informática, contribuyendo a la modernización de la gestión logística en el sector público.

## Problemática a solucionar o necesidad a satisfacer

La gestión de inventarios de insumos tecnológicos en Gendarmería de Chile presenta importantes deficiencias que dificultan el control eficiente de los recursos. Actualmente, muchos de los procesos se realizan de forma manual o sin un sistema formal de registro, lo que genera una serie de inconvenientes: Falta de organización y sistematización: los insumos son recepcionados y utilizados sin un inventariado claro ni procesos estandarizados. Duplicidad y pérdida de información: la ausencia de un sistema centralizado provoca inconsistencias en los registros, lo que conlleva a la compra duplicada de productos o a la falta de disponibilidad en momentos críticos. Errores humanos frecuentes: la gestión manual incrementa la probabilidad de equivocaciones en el registro de entradas, salidas y stock. Ausencia de trazabilidad: no existe un seguimiento confiable que permita conocer el origen, destino o estado actual de los recursos. Falta de reportes confiables: la carencia de información consolidada dificulta la toma de decisiones administrativas y la rendición de cuentas en auditorías. Estos antecedentes reflejan una necesidad urgente de modernización en los procesos de administración de insumos, ya que las falencias actuales no solo afectan la eficiencia administrativa, sino que también implican un uso inadecuado de los recursos públicos, comprometiendo la transparencia y la eficiencia institucional. En este escenario, surge la necesidad de implementar una solución tecnológica integral que permita digitalizar, organizar y automatizar los procesos de gestión de inventario. Esta iniciativa se convierte en un paso clave para garantizar la trazabilidad, reducir costos, evitar pérdidas y fortalecer la capacidad de control de la institución.

## Objetivos del Proyecto (general y específicos)

El objetivo general del proyecto fue diseñar e implementar un sistema de inventario para Gendarmería de Chile que permitiera registrar, controlar y gestionar de manera precisa y eficiente los bienes bajo su custodia, optimizando los procesos de administración de recursos y fortaleciendo la transparencia institucional.

Para dar cumplimiento al objetivo general, se definieron los siguientes objetivos específicos:

* Analizar las necesidades actuales del proceso de control de inventario.
* Optimizar los tiempos de gestión mediante un sistema digital centralizado.
* Implementar funcionalidades de búsqueda y categorización para mejorar la trazabilidad de los bienes.
* Garantizar la integridad y seguridad de los datos mediante respaldos y medidas de protección.
* Generar reportes automáticos que respalden la toma de decisiones estratégicas.
* Integrar un módulo de inteligencia artificial capaz de identificar el estado de conservación de los objetos a través de imágenes.

## Competencias del Perfil de Egreso

El proyecto APT, orientado al desarrollo de una plataforma tecnológica para la gestión de inventarios en Gendarmería de Chile, se vincula directamente con el perfil de egreso de la carrera de Ingeniería en Informática, al integrar las capacidades de diseñar, implementar y administrar soluciones tecnológicas que responden a necesidades organizacionales reales y específicas. De esta manera, a través de su ejecución, el proyecto permitió potenciar y fortalecer las competencias del perfil de egreso que se detallarán a continuación.

El perfil de egreso contempla tanto la formación técnica como las competencias analíticas, comunicacionales y de gestión, las cuales fueron esenciales para abordar la problemática detectada: la ineficiencia en el control, registro y seguimiento de los recursos institucionales.

En primer lugar, la competencia de administrar la configuración de entornos, servicios de aplicaciones y bases de datos fue aplicada de manera directa, ya que la solución se basa en una base de datos centralizada y segura, que requiere una adecuada configuración para garantizar disponibilidad, continuidad operativa y protección de la información sensible.

Asimismo, la capacidad de proponer soluciones informáticas a partir de un análisis integral de los procesos organizacionales se refleja en la evaluación de las falencias existentes (duplicidad de registros, errores humanos, falta de trazabilidad) y en el diseño de un sistema modular que incorpora funcionalidades de control de entradas y salidas, generación de reportes y alertas automáticas, optimizando los flujos de trabajo internos.

La construcción de modelos de datos y la programación de rutinas para la manipulación y actualización de la información constituyeron la base técnica del proyecto, dado que el control de inventario exige estructuras de datos escalables, consultas eficientes y mecanismos que aseguren la trazabilidad y disponibilidad de los insumos en tiempo real.

Por otra parte, la competencia asociada al desarrollo de software bajo estándares de calidad y buenas prácticas de la industria se materializó en la implementación de pruebas unitarias y funcionales, que permitieron validar la fiabilidad del sistema, reducir errores y asegurar que el producto final cumpla con los requisitos técnicos y operativos establecidos.

De manera complementaria, la aplicación de principios de seguridad informática y gestión de vulnerabilidades resultó clave para proteger los datos institucionales, estableciendo niveles diferenciados de acceso y garantizando la confidencialidad e integridad de la información dentro de la plataforma.

Finalmente, las competencias transversales —como la comunicación efectiva, la capacidad de trabajo colaborativo, la gestión de proyectos informáticos y la generación de ideas innovadoras— permitieron planificar, documentar y presentar el proyecto de forma clara, organizada y con enfoque en la mejora continua y el impacto positivo institucional.

En conclusión, el proyecto APT no solo guarda una relación estrecha con el perfil de egreso de la carrera, sino que constituye una aplicación práctica y concreta de las competencias profesionales desarrolladas durante la formación académica. Dichas competencias fueron determinantes para resolver la problemática de gestión de inventarios en Gendarmería, aportando una solución tecnológica eficiente, segura y alineada con los estándares actuales de la ingeniería en informática.

## Asignación de roles

**Tihare Campusano – Scrum Master, Developer**   
Tihare desempeñó el rol de Scrum Master, encargándose de facilitar la metodología ágil Scrum durante todo el desarrollo del proyecto. Se aseguró de que las reuniones se realizaran correctamente, promovió la comunicación efectiva entre los integrantes del equipo, resolvió impedimentos que podían frenar el avance y veló por el cumplimiento de los tiempos establecidos en cada sprint. Su liderazgo permitió mantener la organización y la disciplina del equipo, garantizando un flujo de trabajo constante y ordenado. Además, como Developer, participó activamente en la programación de módulos clave del sistema, contribuyendo en la implementación de funcionalidades críticas y en la resolución de incidencias técnicas.

**Yeremi Guerrero – Product Owner, Developer**   
Yeremi asumió el rol de Product Owner, representando la voz del cliente y asegurando que el sistema respondiera a las necesidades reales de Gendarmería de Chile. Fue responsable de definir y priorizar el Product Backlog, clarificar los requerimientos, validar entregas y garantizar que cada funcionalidad desarrollada generara valor para la institución. Además, impulsó la visión del producto, mantuvo la alineación entre los objetivos estratégicos y las tareas técnicas y validó que el proyecto cumpliera los estándares de calidad esperados. En paralelo, como Developer, participó directamente en el diseño e implementación de interfaces, integraciones y pruebas de las funcionalidades, aportando tanto desde la visión estratégica como desde la ejecución técnica.

**Octavio Yáñez – Developer**   
Octavio desempeñó el rol de Developer, encargándose de implementar funciones específicas relacionadas con la descarga de reportes. Para ello trabajó siguiendo una guía detallada preparada por el Scrum Master, donde se especificaban claramente las funcionalidades que debía desarrollar. Su labor se centró en interpretar correctamente dichas indicaciones, adaptar el código existente cuando fue necesario y asegurar que las funciones implementadas cumplieran con los requisitos establecidos, aportando así al avance del Sprint.

## Metodología utilizada en el Proyecto.

**Proyecto General**

En este proyecto se decidió utilizar la metodología Scrum porque permite organizar el trabajo en sprints cortos (2.5 semanas cada uno), entregar avances de manera periódica y adaptarse a cambios en los requerimientos del cliente (en este caso, Gendarmería de Chile). Además, Scrum asegura que cada integrante del equipo tenga un rol claro (Product Owner, Scrum Master, Developers), lo que mejora la comunicación y coordinación. La elección también responde a que el sistema requiere entregables incrementales y validación continua para garantizar que las funcionalidades críticas estén listas y probadas durante el proceso, no solo al final.

|  |  |
| --- | --- |
| **Elemento** | **Detalle** |
| **Proyecto** | Sistema de Inventario para Gendarmería de Chile |
| **Metodología** | Scrum |
| **Duración** | 6 sprints semanales (2.5 semanas cada uno) |
| **Herramienta de Gestión** | Teamwork |

Tabla 1 Proyecto General

**Comparación metodologías agiles**

Tras un análisis entre metodologías agiles existentes como Scrum, Kanban, XP (Extreme Programming) y Lean, se destacaron sus principales características, ventajas, desventajas y escenarios de aplicación.

* **Scrum** resulta ideal para proyectos que requieren organización en sprints cortos, roles definidos y entregas incrementales, lo que favorece la comunicación y la mejora continua.
* **Kanban** ofrece flexibilidad y control visual del flujo de trabajo, aunque carece de una estructura formal de roles y tiempos, siendo más útil en proyectos de soporte o mantenimiento.
* **XP** se centra en la calidad técnica del software mediante prácticas de programación estrictas como TDD y pair programming, garantizando un código robusto, pero exigiendo alta disciplina del equipo.
* **Lean**, por su parte, busca optimizar recursos eliminando desperdicios y maximizando el valor para el cliente, aunque requiere una fuerte cultura de mejora continua para ser efectivo.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Metodología** | **Características principales** | **Ventajas** | **Desventajas** | **Uso recomendado** |
| **Scrum** | Basada en sprints, roles definidos (Product Owner, Scrum Master, Developers), entregas incrementales. | Claridad de roles, mejora continua, entregables frecuentes. | Puede ser rígida si el equipo no se adapta a los roles. | Proyectos con equipos pequeños-medianos, donde se requiere orden y entregas periódicas. |
| **Kanban** | Flujo continuo, uso de tableros visuales para priorizar tareas. | Flexibilidad, control visual del trabajo. | No define roles ni tiempos, puede generar desorden. | Proyectos de mantenimiento o soporte, con tareas cambiantes. |
| **XP (Extreme Programming)** | Centrado en calidad del código, buenas prácticas de programación (pair programming, TDD). | Mayor calidad técnica, fomenta pruebas automatizadas. | Exige alta disciplina, curva de aprendizaje. | Proyectos con alto componente técnico y necesidad de calidad en software. |
| **Lean** | Basado en la eficiencia y eliminación de desperdicios. | Optimiza recursos, enfoque en valor para el cliente. | Puede ser difícil de implementar si no hay cultura de mejora continua. | Procesos de desarrollo donde la optimización y reducción de costos son clave. |

Tabla 2 Comparativa de metodologías

Ya para concluir, esta comparación permite evidenciar que, para este proyecto específico, Scrum es la metodología más adecuada, ya que combina organización, entregas periódicas y adaptabilidad, aspectos esenciales para el desarrollo de un sistema de inventario confiable y escalable para Gendarmería de Chile.

**Sprints**

**Sprint 1: Planificación y Configuración Inicial**

Esta tabla presenta el primer sprint, donde se definió la configuración inicial del entorno, la base de datos y la estructura del sistema. El propósito fue sentar las bases técnicas y organizativas para permitir un desarrollo fluido en los siguientes sprints.

|  |  |
| --- | --- |
| **Categoría** | **Detalle** |
| Tareas | * Configuración del entorno de desarrollo * Diseño de base de datos * Estructura inicial del sistema * Definición de roles y permisos |
| Responsables | Yeremi Guerrero (Product Owner, Developer)  Tihare Campusano (Scrum Master, Developer)  Octavio Yáñez (Developer) |
| Entregables | * Entorno de desarrollo configurado * Modelo de base de datos inicial * Estructura básica del sistema |

Tabla 3 Sprint 1

**Sprint 2: Registro de Entradas y Salidas**

El segundo sprint se enfocó en el desarrollo del módulo de entradas y salidas. Esta fase fue clave porque constituye el núcleo del sistema de inventario, asegurando que desde un inicio se pudiera registrar el movimiento de los recursos.

|  |  |
| --- | --- |
| **Categoría** | **Detalle** |
| **Tareas** | - Desarrollo de módulo de registro de entradas y salidas  - Implementación de validaciones básicas  - Pruebas unitarias iniciales |
| **Responsables** | Yeremi Guerrero (Product Owner, Developer)  Tihare Campusano (Scrum Master, Developer) |
| **Entregables** | - Módulo funcional de registro de entradas y salidas  - Documentación de pruebas realizadas |

Tabla 4 Sprint 2

**Sprint 3: Gestión de Stock y Alertas**

En este sprint se abordó la gestión de stock y alertas automáticas, agregando valor al sistema mediante un control en tiempo real y avisos de reposición. Esto mejora la eficiencia y evita quiebres de stock.

|  |  |
| --- | --- |
| **Categoría** | **Detalle** |
| **Tareas** | - Implementación de control de stock en tiempo real  - Desarrollo de sistema de alertas automáticas de reposición  - Integración con módulo de entradas y salidas |
| **Responsables** | Yeremi Guerrero (Product Owner, Developer)  Tihare Campusano (Scrum Master, Developer)  Octavio Yáñez (Developer) |
| **Entregables** | - Sistema de gestión de stock operativo  - Alertas automáticas funcionales |

Tabla 5 Sprint 3

**Sprint 4: Reportes y Seguridad**

El cuarto sprint se centró en la seguridad y la generación de reportes. Se buscó garantizar tanto la integridad de los datos como la información útil para la toma de decisiones.

|  |  |
| --- | --- |
| **Categoría** | **Detalle** |
| Tareas | * Desarrollo de reportes automáticos personalizables * Implementación de medidas de seguridad y respaldos * Pruebas de seguridad y validación de reportes |
| Responsables | Yeremi Guerrero (Product Owner, Developer)  Tihare Campusano (Scrum Master, Developer)  Octavio Yáñez (Developer) |
| Entregables | - Reportes automáticos en funcionamiento  - Sistema de seguridad y respaldos implementado |

Tabla 6 Sprint 4

**Sprint 5: Integración de Inteligencia Artificial**

Aquí se integró un módulo de Inteligencia Artificial, lo que representa un valor agregado innovador. El modelo entrenado con datos reales permite anticipar necesidades y optimizar la gestión de inventario.

|  |  |
| --- | --- |
| **Categoría** | **Detalle** |
| **Tareas** | - Desarrollo e integración del módulo de inteligencia artificial  - Entrenamiento del modelo con datos reales  - Pruebas de precisión y ajuste del modelo |
| **Responsables** | Tihare Campusano (Scrum Master, Developer) |
| **Entregables** | - Módulo de IA integrado y funcional  - Resultados de pruebas y ajustes realizados |

Tabla 7 Sprint 5

**Sprint 6: Finalización y Despliegue**

El último sprint estuvo dedicado a la fase final de despliegue, corrigiendo errores y realizando pruebas de integración y aceptación. El objetivo fue entregar un sistema completo, operativo y documentado para su uso inmediato.

|  |  |
| --- | --- |
| **Categoría** | **Detalle** |
| **Tareas** | - Ajustes finales y corrección de errores  - Pruebas de integración y aceptación  - Despliegue en entorno de producción |
| **Responsables** | Yeremi Guerrero (Product Owner, Developer)  Tihare Campusano (Scrum Master, Developer) |
| **Entregables** | - Sistema desplegado y operativo  - Documentación final y manual de usuario |

Tabla 8 Sprint 6

Product Backlog inicial

* Registro de entradas y salidas de inventario
* Control de stock en tiempo real
* Alertas automáticas de reposición
* Reportes automáticos personalizables
* Seguridad y respaldos de datos
* Integración de inteligencia artificial para análisis de conservación
* Interfaz de usuario amigable y responsive
* Pruebas de calidad y validación

Temas de retrospectiva

* Comunicación y colaboración del equipo
* Cumplimiento de plazos y estimaciones
* Calidad del código y pruebas realizadas
* Herramientas y recursos utilizados
* Lecciones aprendidas y áreas de mejora

## Creación de cronograma asociado al Proyecto

La carta Gantt presentada corresponde a la planificación del proyecto APT, que busca desarrollar un sistema de gestión de inventario para Gendarmería de Chile. En ella se detallan las fases, actividades y entregables principales, organizados a lo largo del semestre académico (agosto – noviembre de 2025). El cronograma se ha estructurado en cinco grandes etapas:

**Inicio**  
Comprende la identificación de la problemática, la definición de objetivos y alcance, así como la asignación de roles. Esta fase inicial asegura una base sólida para el desarrollo del proyecto.

**Planificación**  
Incluye la investigación y análisis de la situación actual, la recopilación de requerimientos funcionales y no funcionales, el diseño de la arquitectura del sistema y la elaboración de diagramas y mockups. El objetivo es obtener una visión clara y organizada de la solución antes de pasar a su construcción.

**Ejecución**  
Abarca la construcción de la base de datos, la programación del sistema, la integración de componentes de inteligencia artificial y la documentación técnica. En esta etapa se materializa el sistema propuesto.

**Control y Seguimiento**  
Considera la realización de pruebas de funcionalidad, integración, rendimiento y seguridad, junto con la corrección de errores detectados. Además, se supervisa el avance respecto al cronograma para garantizar que el proyecto cumpla con los objetivos en tiempo y forma.

**Cierre**  
Incluye la preparación del entorno de producción, migración de datos, despliegue del sistema, capacitación de usuarios y entrega final. Finalmente, se formaliza el cierre del proyecto con la documentación y retroalimentación correspondiente.

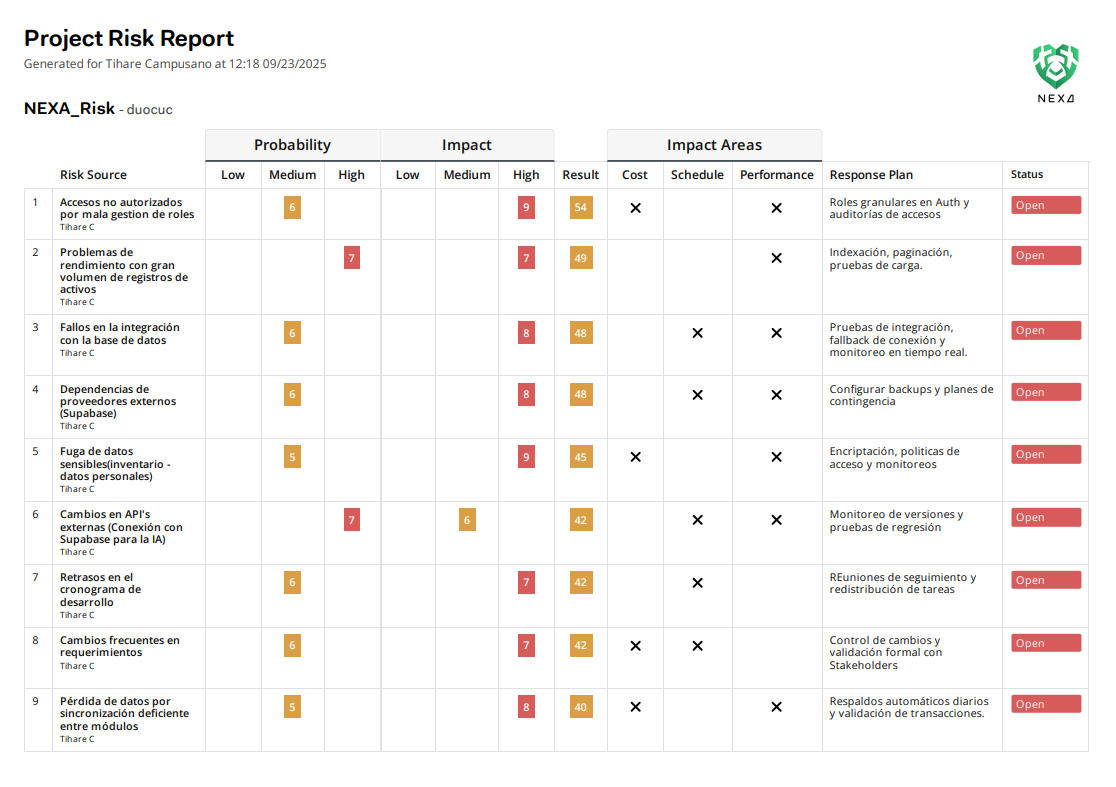
Escala de tiempo, Gráfico de barras

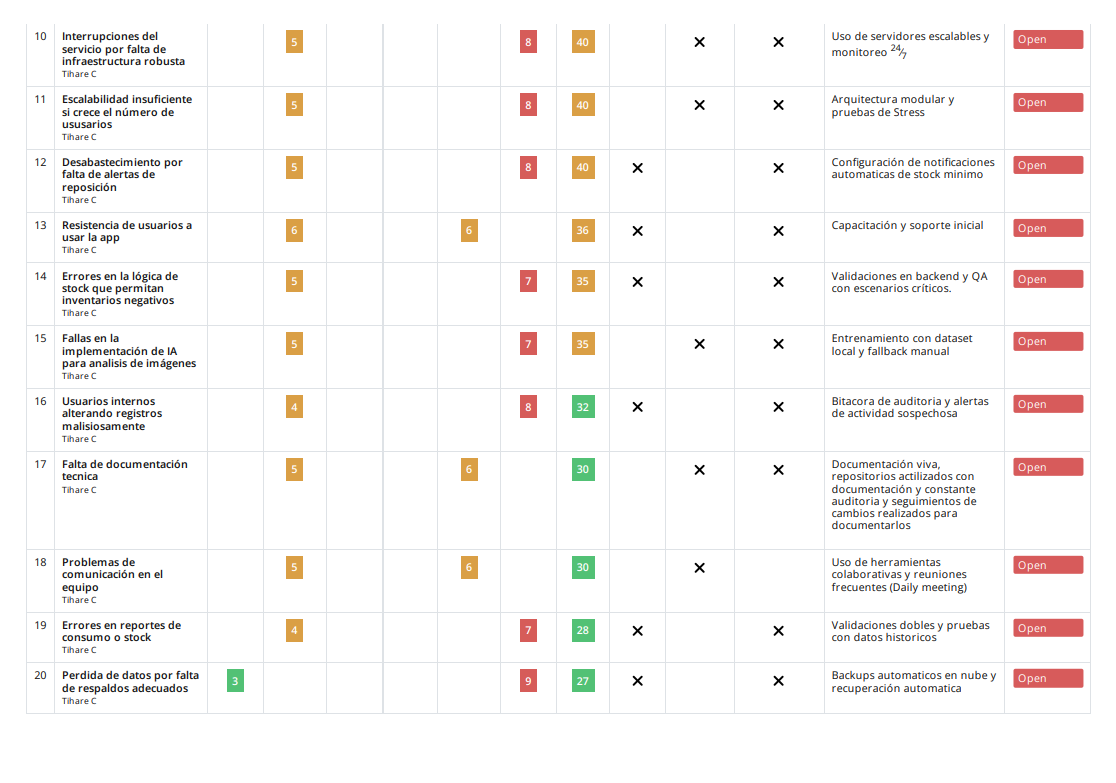
El contenido generado por IA puede ser incorrecto.

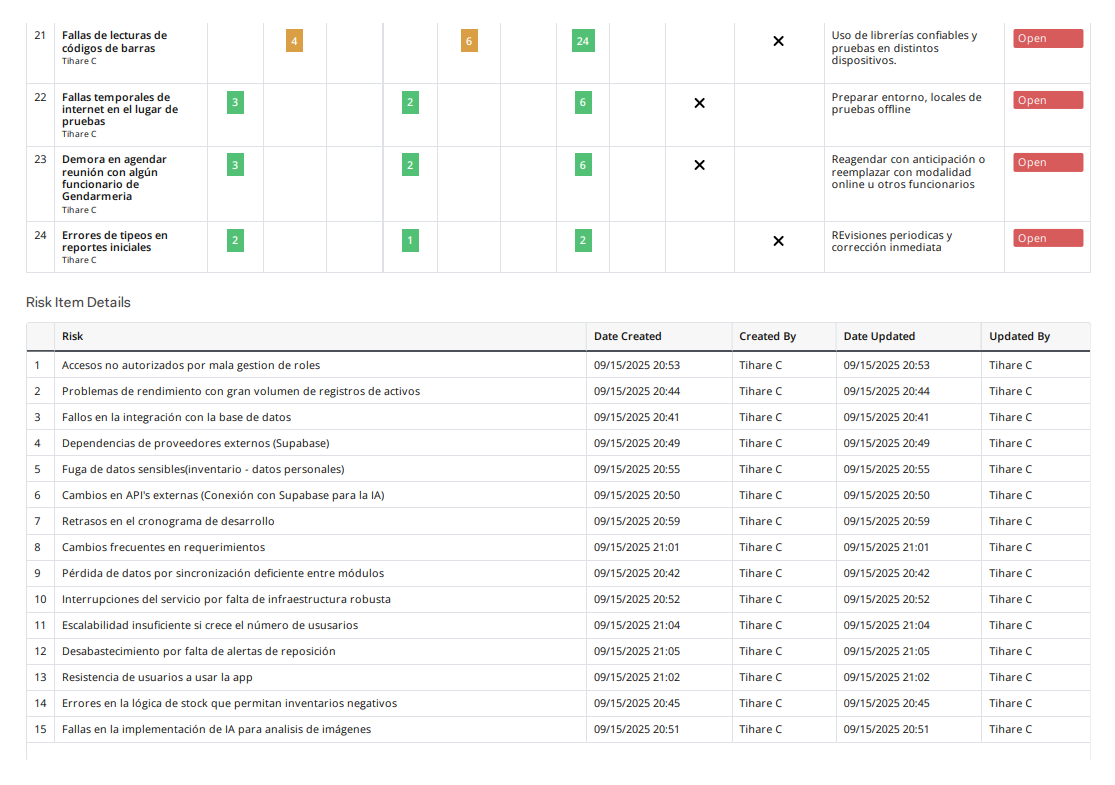
Ilustración 1 Carta Gantt

## Riesgos Asociados al Proyecto

En este apartado se presenta la **matriz de riesgos del proyecto**, la cual incluye la identificación de posibles amenazas que podrían afectar el desarrollo del sistema de gestión de inventario. La tabla contempla la **descripción de cada riesgo**, su **probabilidad de ocurrencia** (baja, media o alta), el **impacto potencial** que tendría sobre el proyecto y el **resultado** de combinar ambas variables, lo que permite establecer un nivel de **criticidad**. Asimismo, se especifican las **áreas de impacto** (costo, cronograma y rendimiento) y se detallan los **planes de mitigación o respuesta** propuestos para cada riesgo. De esta forma, la matriz facilita la gestión proactiva de incidentes mediante la definición de **estrategias preventivas y de contingencia**, lo que asegura un mayor control sobre los factores que pueden comprometer los objetivos del proyecto.







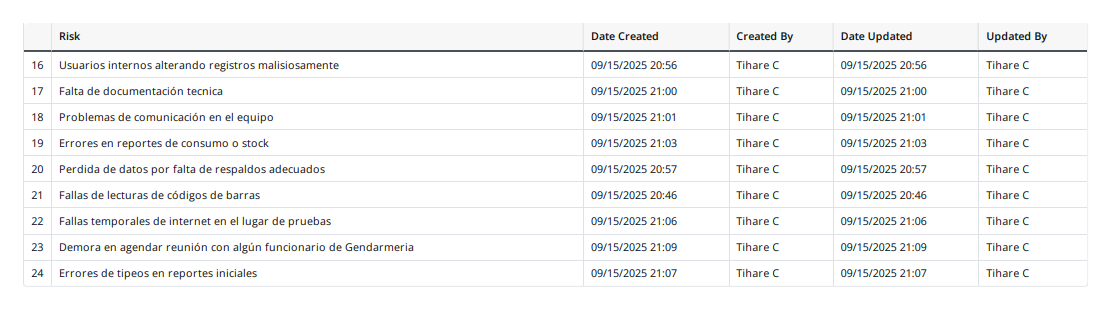


Ilustración 2 Riesgos asociados al proyecto

## Implementación del Proyecto

### Diseño y Arquitectura de la solución

**Dispositivos y herramientas**

El desarrollo de la aplicación NEXA se gestiona por el Equipo utilizando VS Code para codificación y Git / Teamwork para control de versiones y colaboración, todo dentro de un entorno de Desarrollo general. El Frontend está construido sobre TypeScript y React, utilizando Ionic Framework para la capa de la aplicación móvil y Tailwind CSS para los estilos. La persistencia de datos y el Backend son manejados por Supabase, que actúa como la Base de Datos. Para la Distribución final, la aplicación se empaqueta como un APK Android. Además, NEXA incluye un módulo de Inteligencia Artificial desarrollado en Python que utiliza TensorFlow / Keras para el entrenamiento del modelo. La inferencia optimizada se realiza usando MobileNetV2 y se despliega en el entorno de producción mediante TFLite (para inferencia local), y es contenerizada en Docker para ser ejecutada en la nube a través de Google Cloud Run (para inferencia remota).

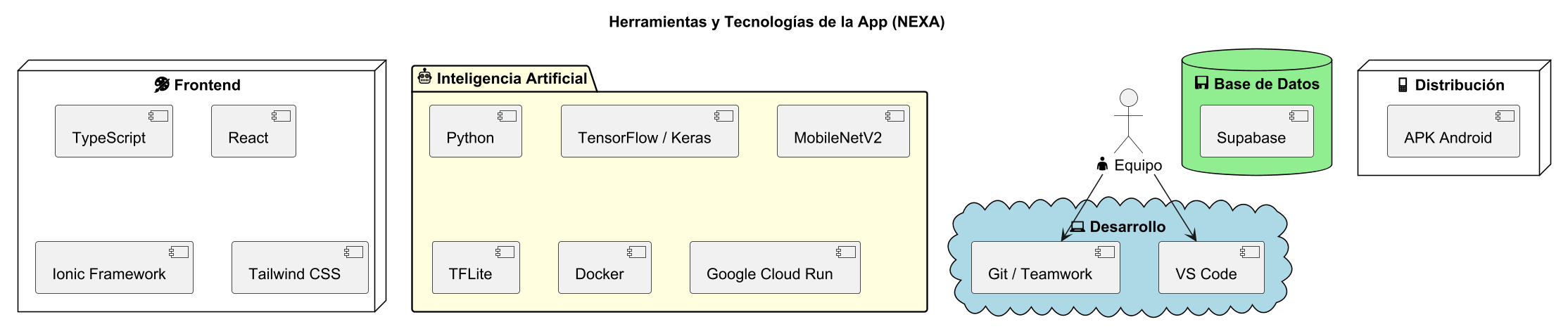


Ilustración 3 Herramientas del proyecto

**Diagrama de arquitectura IA**

La arquitectura de inferencia para la clasificación de productos está diseñada para ser escalable y accesible a través de la nube. El proceso inicia cuando el Usuario utiliza la App Móvil (Ionic React) para enviar una imagen, generando una Petición de Inferencia (REST) que se dirige al Punto de Acceso Público alojado en Google Cloud Run. Este servicio serverless ejecuta un Contenedor Docker (inventario-ia-api:v1

x, construido sobre la imagen tiangolo/uvicorn-gunicorn-fastapi) bajo demanda, aprovechando el Escalamiento a Cero. Dentro del contenedor, la API (FastAPI) es el componente central que Carga y Predice utilizando el Modelo TFLite optimizado. Tras la clasificación, la API opcionalmente Guarda/Consulta Datos en la Base de Datos (Supabase) antes de generar la Respuesta de Clasificación (JSON), la cual es finalmente procesada por la App Móvil para Mostrar el Resultado al usuario. Esta estructura garantiza una predicción rápida, un consumo eficiente de recursos y alta disponibilidad.

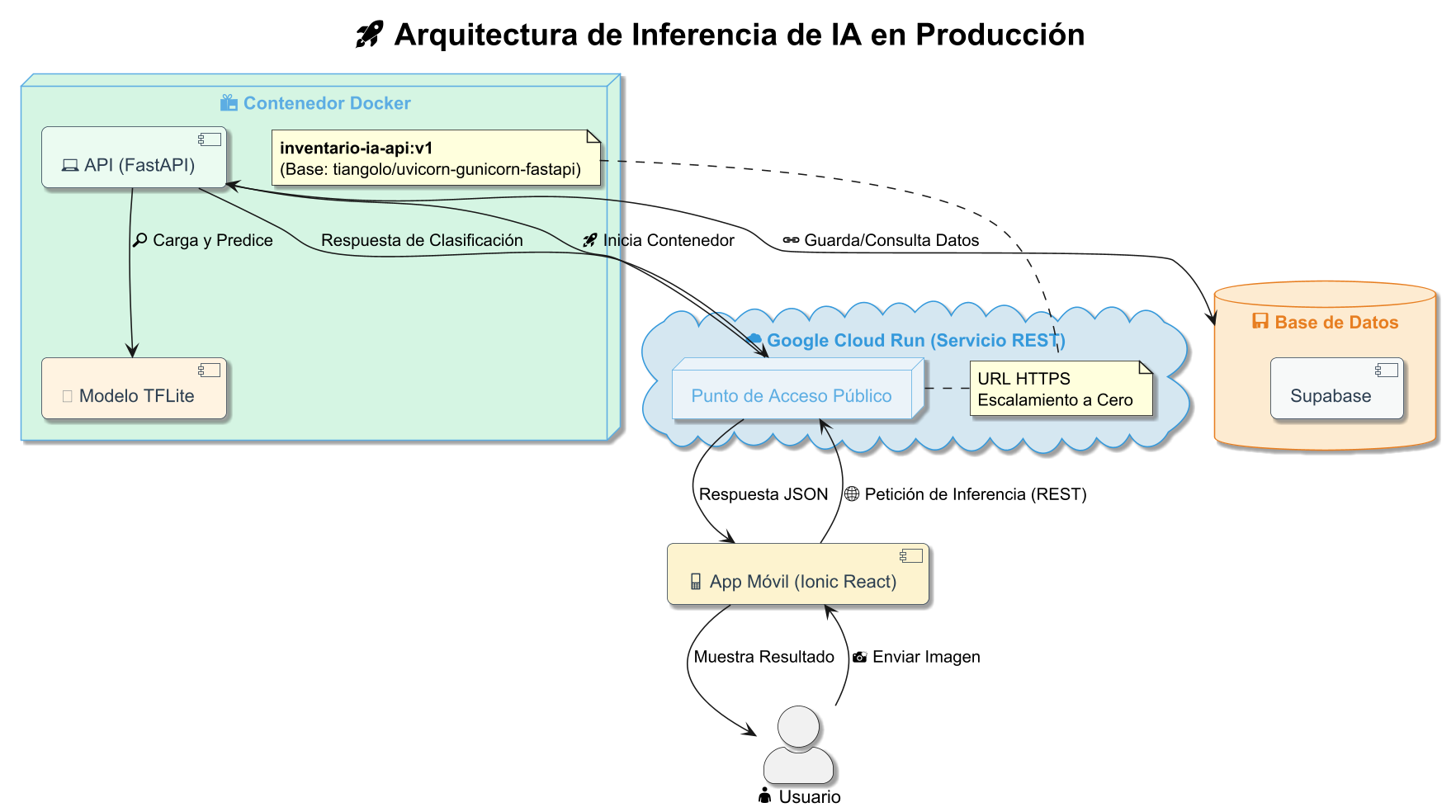


Diagrama 1 Arquitectura IA

**Diagrama de arquitectura**

El diagrama de arquitectura del Proyecto APT muestra un sistema donde el Frontend (Ionic React) se conecta directamente con dos servicios independientes: el Backend Principal y el Backend de IA. El Backend Principal (FastAPI) gestiona la lógica de negocio, incluyendo la autenticación de usuarios y la administración del inventario, comunicándose con Supabase para realizar operaciones de base de datos. Paralelamente, el Backend de IA (contenedor Docker desplegado en Cloud Run) recibe directamente desde el Frontend las imágenes a clasificar y devuelve el estado del producto. Ambos servicios responden al Frontend por separado, permitiendo una arquitectura modular, segura y eficiente, donde cada backend cumple un rol específico sin depender del otro.

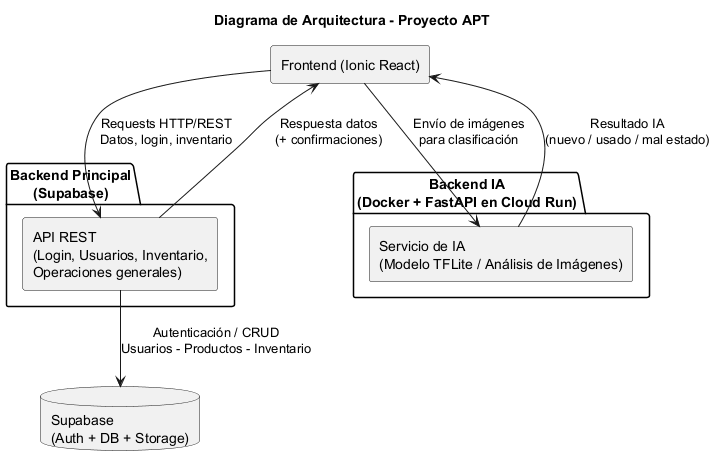


Diagrama 2 Arquitectura del Proyecto

### Requerimientos técnicos y Diagramas

|  |  |
| --- | --- |
| **Diagrama de Actividades**  El diagrama de actividades del Proyecto APT muestra el flujo completo de uso de la aplicación, desde la apertura hasta las diferentes funcionalidades disponibles para el usuario. El proceso inicia en la pantalla de Login, donde se validan las credenciales. Si son correctas, el usuario accede a la pantalla Identifícate, ingresa su nombre y apellido, se registra en la base de datos y finalmente llega al Home de la aplicación. Si las credenciales son incorrectas, se muestra un mensaje de error y el flujo termina. En el Home, el usuario puede visualizar la tabla de productos y elegir entre varias acciones: buscar productos, registrar nuevos, editar existentes, generar reportes o acceder a su perfil. En el módulo de registro de productos, la información puede ingresarse de tres maneras: manualmente, mediante escaneo de código QR con pistola, o usando la cámara del móvil. Si el producto ya existe, el sistema autocompleta sus datos. Además, el usuario puede tomar fotografías del producto, que son analizadas por un módulo de Inteligencia Artificial para determinar su estado (nuevo, usado o en mal estado) antes de guardar la información en la base de datos. El módulo de edición permite seleccionar un producto existente, modificar campos como nombre, código, modelo o marca, y guardar los cambios. El módulo de reportes permite elegir el tipo de informe, generar un PDF y descargarlo. Finalmente, desde el perfil, el usuario puede ver datos personales como alias, correo electrónico, dispositivos asociados, sesiones activas y fecha del último inicio de sesión. Este flujo refleja cómo la aplicación gestiona la interacción del usuario, la administración de productos, la integración con inteligencia artificial y la generación de reportes, garantizando un recorrido completo, seguro y eficiente dentro del sistema. |  |

Diagrama 3 Actividades

**Diagrama Entidad - Relación**

El Diagrama Entidad-Relación (ER) del Proyecto APT define la estructura de datos para la gestión de inventario y análisis de calidad. La entidad Usuario almacena información de acceso y se relaciona con Movimiento, detallando qué usuario realiza una transacción de inventario. El núcleo es la entidad Producto, que contiene todos los datos del artículo (como codigo\_barras, stock, estado) y se asocia tanto con Categoría (permitiendo la organización jerárquica mediante parent\_id) como con Movimiento (que registra entradas y salidas). La entidad Inspeccion es clave para el análisis, ya que verifica un Producto y utiliza la información de ModeloIA (que detalla el nombre y versión del modelo de inteligencia artificial) para registrar el resultado del análisis, incluyendo la etiqueta, la confianza y los detalles del proceso.

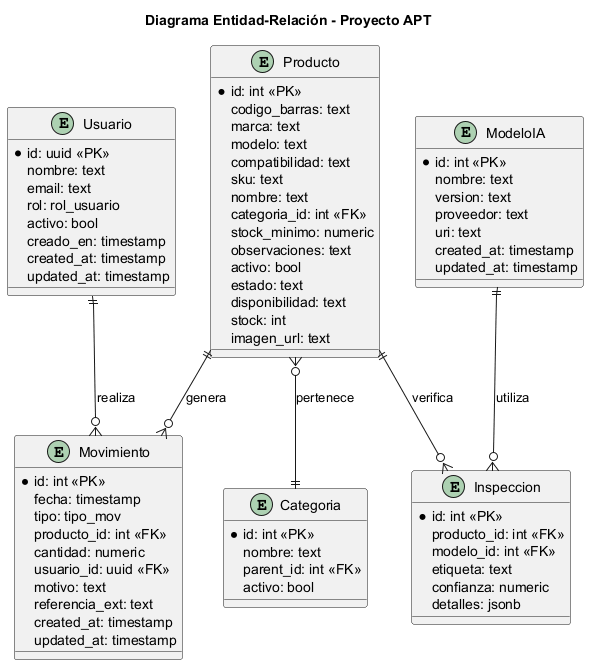


Diagrama 4 Entidad-Relación

**Diagrama de clase**

El diagrama de clases del Proyecto APT representa la arquitectura orientada a objetos del sistema, enfocada en la gestión de inventario y la trazabilidad de la calidad, donde la clase central es Producto, que contiene atributos detallados del artículo (stock, imagen\_url) y se relaciona con la clase Categoría para su organización. La interacción del usuario se gestiona a través de la clase Usuario y su método registrarMovimiento(), lo que da lugar a la creación de instancias de la clase Movimiento, la cual tiene una relación de muchos a muchos con Producto para registrar entradas y salidas. Finalmente, el módulo de calidad se compone de la clase ModeloIA, que define los metadatos de los modelos de inteligencia artificial, y la clase Inspeccion, que registra los resultados del análisis (etiqueta, confianza, detalles) manteniendo una relación de muchos a muchos tanto con Producto como con ModeloIA.

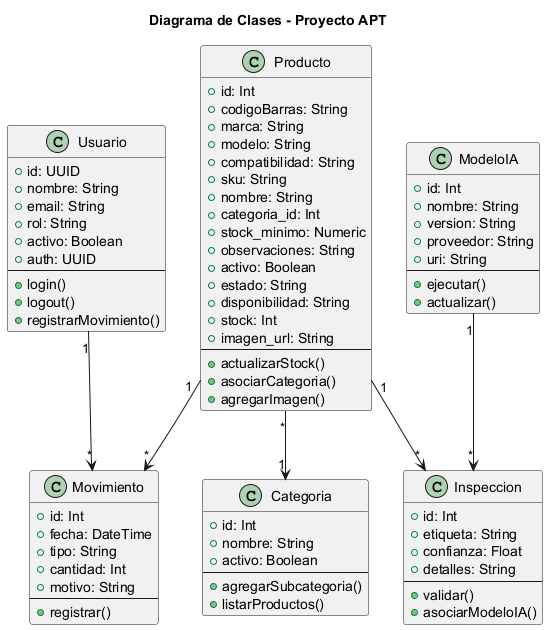


Diagrama 5 Clases

**Diagrama de componentes**

El diagrama de componentes del Proyecto APT representa los módulos principales del sistema y cómo se interconectan para soportar todas las funcionalidades de la aplicación. En el Frontend, desarrollado con Ionic React, se incluyen la UI (Pantallas) para la interacción del usuario, la Gestión de Estado (Context/Redux) para mantener centralizado el estado de la aplicación y el Cliente API (REST/GraphQL) para comunicarse con el Backend. El Backend API, implementado en FastAPI, se organiza en módulos especializados: Autenticación (BEAuth), que valida credenciales mediante Firebase Auth; Inventario (BEInv), que realiza operaciones CRUD sobre los productos en Supabase; Integración con IA (BEIA), que envía imágenes al Servicio de IA y recibe el estado de los productos; y Comunicación con DB (BEDB), que administra la conexión directa con la base de datos. El Servicio de IA se encarga del análisis de imágenes, proporcionando información sobre el estado de los productos al Backend. Cada componente cumple un rol específico y su integración asegura el funcionamiento completo del sistema, abarcando desde la experiencia del usuario y la gestión de datos hasta la seguridad y el análisis inteligente, garantizando un flujo eficiente y coherente de la información en toda la aplicación.

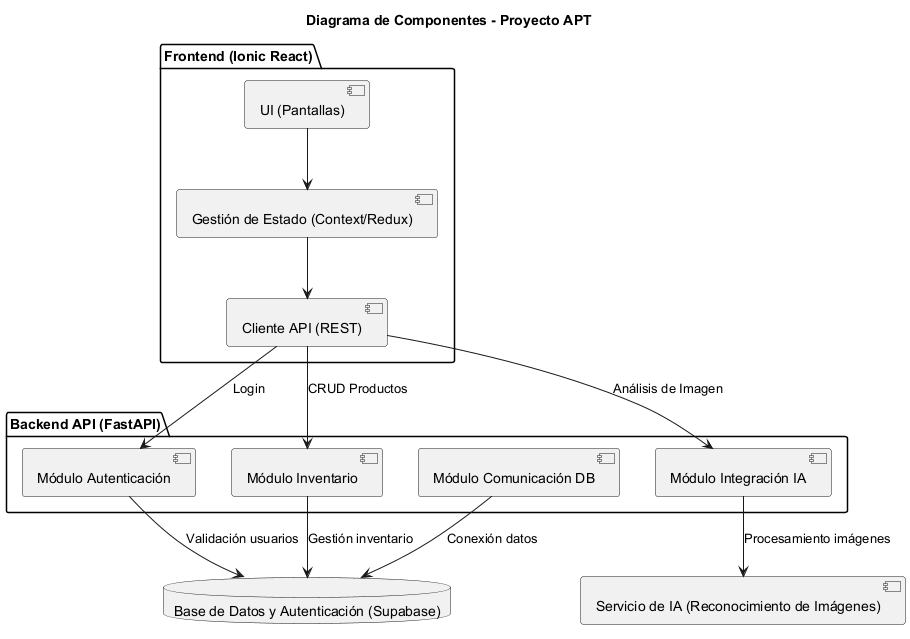


Diagrama 6 Componentes

**Diagrama de despliegue**

El diagrama de despliegue del Proyecto APT ilustra la infraestructura donde se ejecutan los distintos componentes del sistema y cómo se interconectan. En el nodo Cliente se encuentran el navegador web y la aplicación móvil desarrollada en Ionic React, que constituyen los puntos de acceso para el usuario final y se comunican mediante HTTP/HTTPS con el Servidor Backend, donde reside la API REST implementada en FastAPI. Este servidor procesa las solicitudes, gestiona la lógica de negocio y se conecta con los servicios en la nube, incluyendo Supabase, que proporciona la base de datos y la autenticación de usuarios, y el Servicio de IA (por ejemplo, TensorFlow con MobileNetV2), encargado del procesamiento y análisis de imágenes. En conjunto, esta arquitectura cliente-servidor con soporte en la nube asegura un almacenamiento seguro, autenticación confiable y análisis inteligente de productos a través de servicios externos, garantizando un flujo de información eficiente y seguro en toda la aplicación.

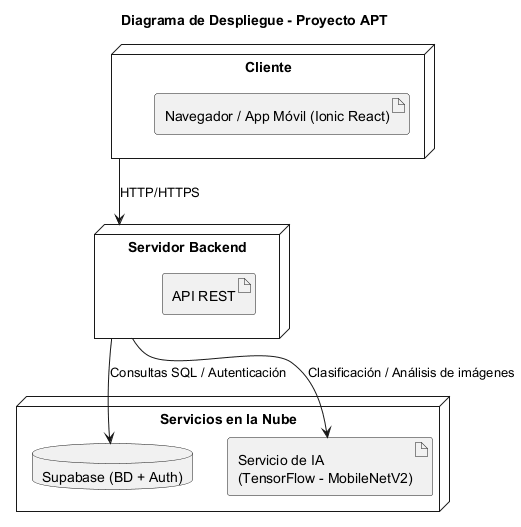


Diagrama 7 Despliegue

|  |  |
| --- | --- |
| **Diagrama de flujo**  El diagrama de flujo del Proyecto APT muestra paso a paso la interacción del usuario con la aplicación, desde la apertura hasta la ejecución de sus distintas funcionalidades. El flujo inicia cuando el usuario abre la app y accede a la pantalla de Login, donde se validan sus credenciales; si son correctas, el usuario ingresa su nombre y apellido, se registra en la base de datos y obtiene acceso al Home, mientras que si las credenciales son incorrectas, se muestra un mensaje de error y el flujo termina. Desde el Home, el usuario puede realizar varias acciones: buscar productos, ingresando un término y visualizando los resultados; generar reportes, seleccionando el tipo de informe, generando un PDF y descargándolo; registrar productos, ya sea de forma manual, mediante escaneo con pistola o con la cámara del móvil, con autocompletado si el producto ya existe, seguido del análisis del estado del producto mediante inteligencia artificial y su almacenamiento en la base de datos; editar productos, seleccionando un ítem, modificando campos como nombre, código, modelo o marca y guardando los cambios; y visualizar su perfil, donde se muestran alias, correo electrónico, dispositivos asociados y sesiones activas. Este flujo refleja de manera integral cómo la aplicación gestiona la autenticación, la administración de productos, la generación de reportes y la gestión de perfiles, asegurando un recorrido completo, eficiente y coherente para el usuario dentro del sistema. |  |

Diagrama 8 Flujo

**Casos de uso**

El diagrama de casos de uso del Sistema NEXA refleja las principales funcionalidades disponibles para el rol de Administrador, quien posee un conjunto de privilegios para gestionar y mantener el sistema. Entre las operaciones más relevantes se encuentran la gestión de usuarios y roles, el respaldo y restauración de la base de datos, así como la configuración de umbrales de reposición y parámetros de la inteligencia artificial (IA). Asimismo, el administrador puede iniciar y cerrar sesión, generar reportes que incluyen la visualización del inventario, y acceder a la auditoría y registros del sistema, los cuales se pueden extender con la revisión de evaluaciones de IA. Este modelo asegura un control integral, permitiendo al administrador supervisar el estado del sistema y garantizar su correcto funcionamiento.

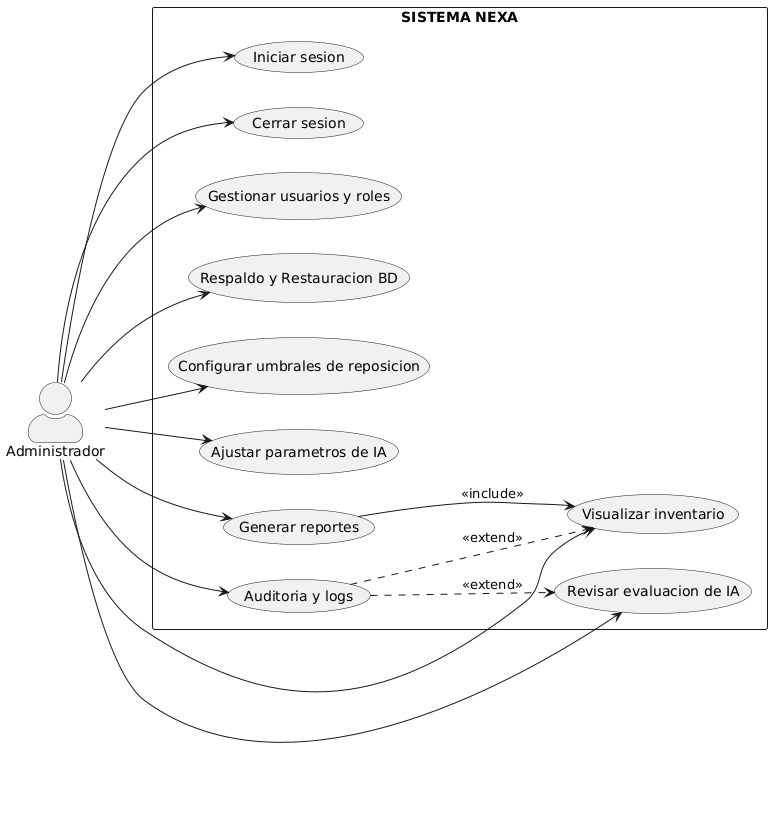


Diagrama 9 Casos de uso

### Resultados de la solución

**Aplicación completada**

El desarrollo del proyecto ha finalizado con éxito, por lo que la aplicación se encuentra totalmente terminada y completada. A continuación, se procederá a presentar una visión detallada de las interfaces de usuario (UI) y los elementos visuales clave de la aplicación. Esta demostración permitirá apreciar la calidad estética, la usabilidad y la coherencia del diseño, mostrando cómo la arquitectura visual facilita la interacción y la experiencia del usuario con la plataforma tecnológica desarrollada.

[](https://www.youtube.com/embed/-XJQdS5Pih0?feature=oembed)

**Aplicación NEXA**

Al iniciar el desarrollo de la aplicación, la integrante Tihare Campusano fue responsable de implementar un script en Python que automatiza por completo la preparación del proyecto. Este script genera la estructura base, creando carpetas y archivos esenciales tanto para el frontend como para el backend. En el caso del frontend, configura un entorno con Ionic, React, Tailwind y TypeScript, además de instalar dependencias necesarias y generar archivos preconfigurados como tailwind.config.js y postcss.config.js. Para el backend, establece un proyecto en FastAPI con una organización modular de directorios (api, core, services) y archivos iniciales como main.py, config.py y un servicio básico de IA. Asimismo, el script prepara la sección de Machine Learning con carpetas para datos, notebooks y modelos, y añade un área de documentación con archivos de soporte. Finalmente, genera un README.md, un .gitignore adaptado y asegura la instalación de las herramientas requeridas, garantizando así un entorno listo para comenzar con el desarrollo de la aplicación.

[generar\_inventario.py](https://github.com/Tihare-Campusano/Script-para-generar-cod-base-de-App-NEXA/blob/main/Script%20para%20generar%20cod%20base%20de%20App%20NEXA/generar_inventario.py) / [Video\_de\_script\_funcionando](https://youtu.be/4MeD01NGOQc?si=niLRf4ud9LzhzPRg)

Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto.

Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto.

**Base de Datos Supabase**

El integrante Yeremi Guerrero, Diseño la base de datos del proyecto de esta manera porque era fundamental asegurar la organización, trazabilidad y seguridad de la información de inventario en Gendarmería. Definí tablas principales como usuarios, categorías, productos, movimientos, modelos\_ia e inspecciones, ya que cada una cumple un rol específico en la gestión integral del sistema.

Opté por esta estructura porque permite:

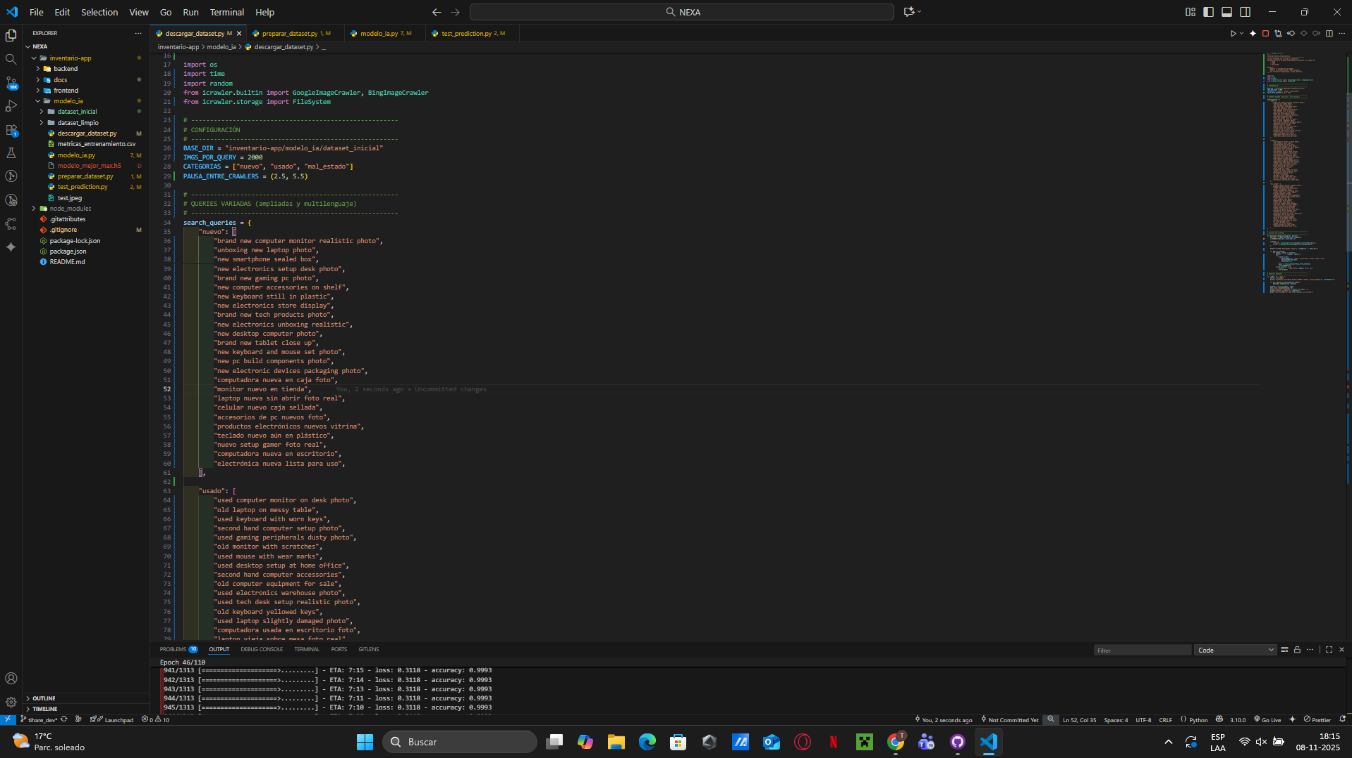
* **Mantener integridad y consistencia** de los datos mediante llaves foráneas y restricciones.
* **Facilitar la trazabilidad** de entradas y salidas con la tabla de movimientos (kardex).
* **Optimizar consultas** a través de vistas como v\_stock\_actual y v\_resumen\_diario, que entregan información inmediata y consolidada.
* **Incorporar evidencia y análisis inteligente**, con tablas de imágenes e inspecciones vinculadas a modelos de IA, asegurando que el estado de los insumos sea validado tanto manual como automáticamente.

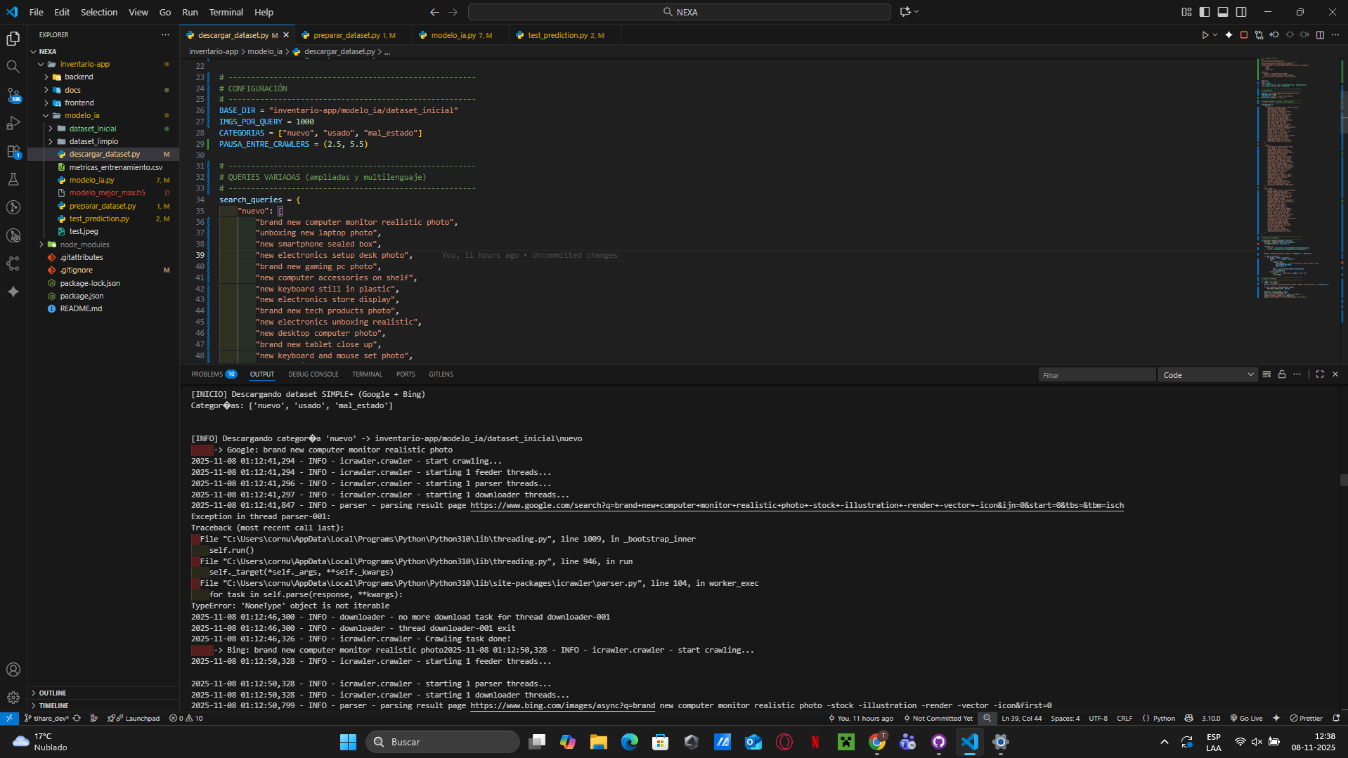
De esta manera, la base de datos no solo respalda el funcionamiento del sistema NEXA, sino que también garantiza escalabilidad, eficiencia en la consulta de datos y soporte a los módulos de inteligencia artificial que forman parte del proyecto.

**Modelo IA**

En la etapa de preparación del modelo de inteligencia artificial, la integrante Tihare Campusano implementó el script en Python descargar\_dataset.py, cuyo objetivo principal es la descarga masiva de imágenes (estimando más 80.000 imágenes) para crear un dataset estructurado. Este script utiliza la librería icrawler empleando tanto el motor de búsqueda Google como Bing para descargar imágenes basándose en múltiples y variados términos de búsqueda (queries en español e inglés). El proceso organiza automáticamente las imágenes en tres categorías principales: nuevo, usado y mal\_estado. Para maximizar la cantidad, el código intenta descargar + 1000 imágenes por cada query definida (a diferencia de las 20 mencionadas), e incluye filtros básicos para excluir ilustraciones (-stock -illustration...) y una pausa aleatoria entre descargas para evitar ser bloqueado, lo que resulta en un dataset inicial masivo y clasificado para el entrenamiento del modelo.

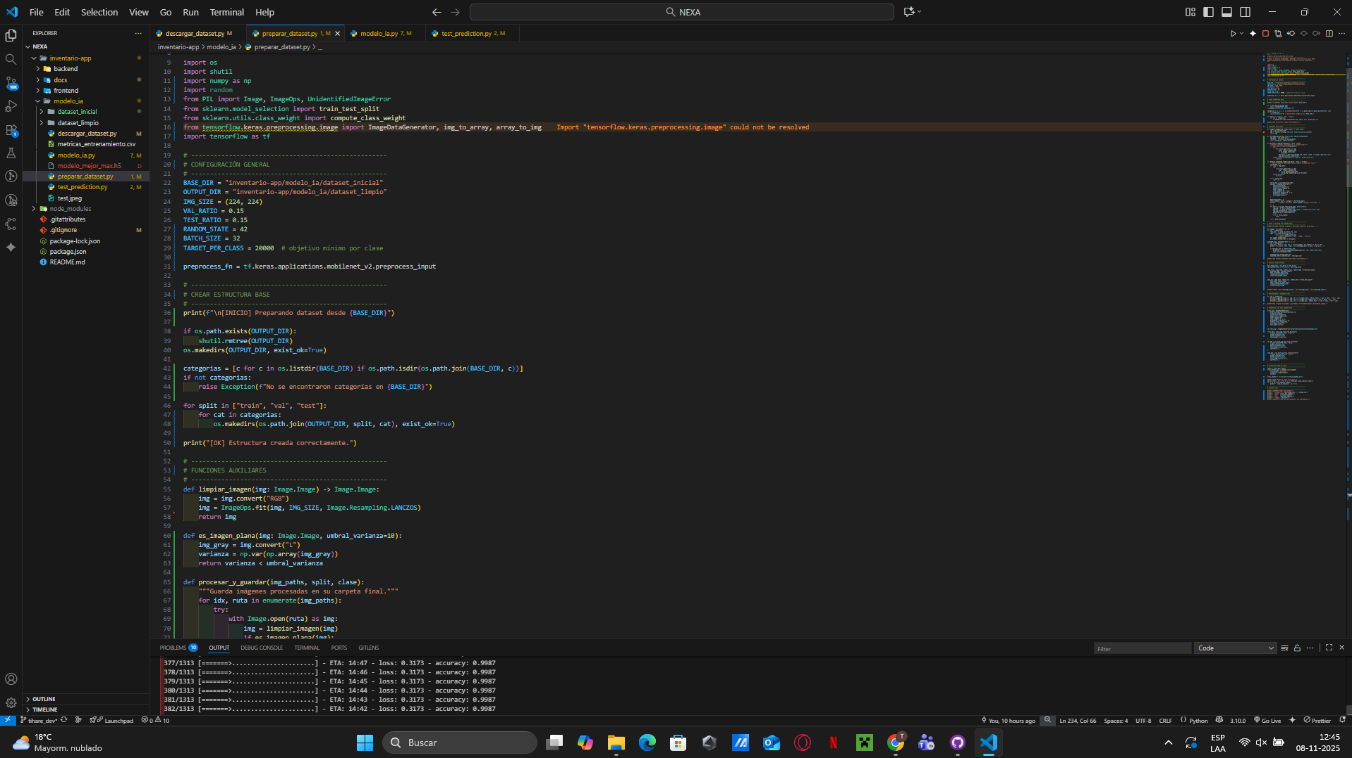
[Descargar\_dataset.py](https://github.com/Tihare-Campusano/NEXA/blob/tihare_dev/inventario-app/modelo_ia/descargar_dataset.py) / [Video\_de\_script\_funcionando](https://youtu.be/GjgzHVGW3QU?si=yCd1LTdp52H90SQj)

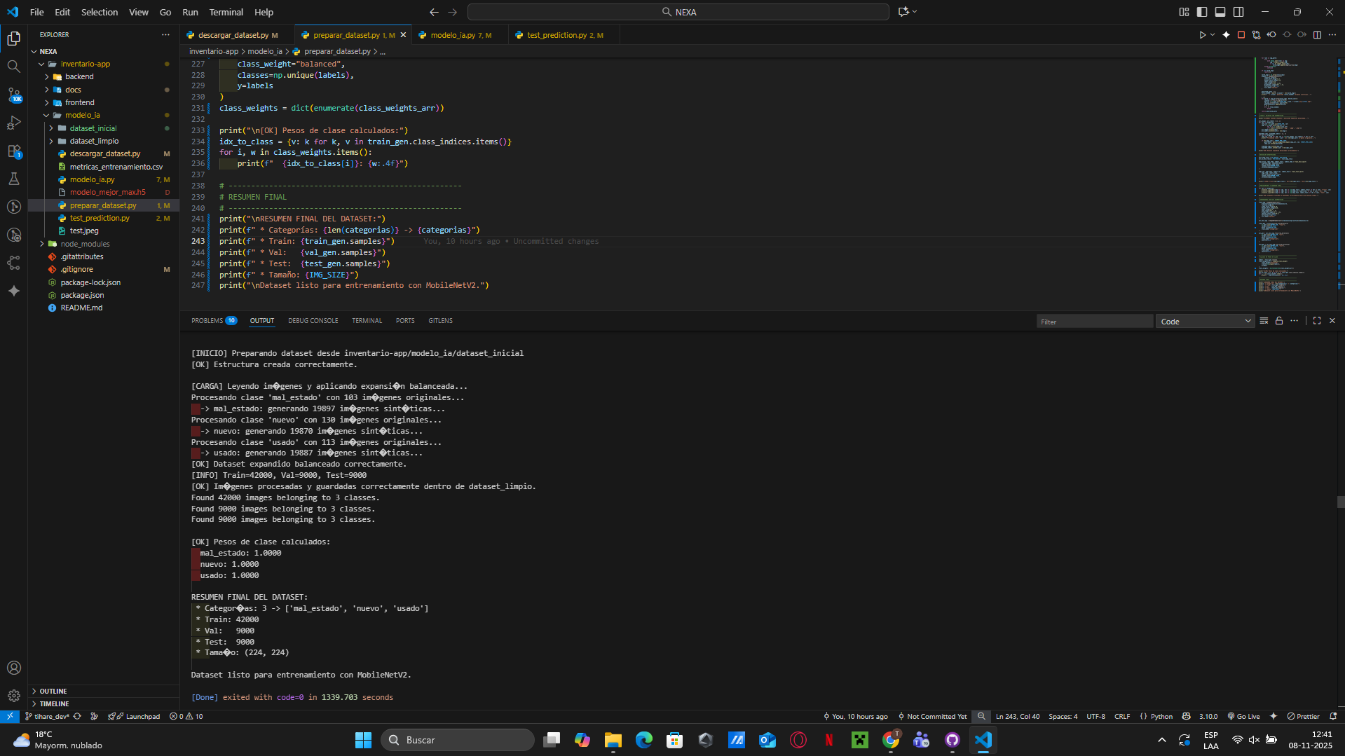




Continuando con la preparación del dataset, la integrante Tihare Campusano fue la encargada de desarrollar el script en Python preparar\_dataset.py que automatiza la limpieza, el balanceo, la división y la preparación final del conjunto de datos para el modelo de visión por computadora. Este script procesa el dataset inicial en varias etapas: primero realiza la limpieza al convertir cada imagen a RGB, redimensionarla al tamaño estándar de 224 X 224 píxeles, y descartar imágenes completamente planas o fallidas. Para lograr un dataset balanceado, el código utiliza ImageDataGenerator de Keras para generar imágenes sintéticas (aumento de datos) en tiempo de ejecución para las categorías que no alcanzan el objetivo de 20.000 imágenes por clase. Luego, el script realiza una división estratificada utilizando Scikit-learn para crear automáticamente los subconjuntos train (70%), val (15%) y test (15%), manteniendo la proporción de clases. Finalmente, prepara generadores de datos con ImageDataGenerator para la carga eficiente (aplicando aumento de datos adicional en el conjunto de entrenamiento) y calcula los pesos de clase para mitigar cualquier desequilibrio residual durante el entrenamiento del modelo\_ia.

[Preparar\_dataset.py](https://github.com/Tihare-Campusano/NEXA/blob/tihare_dev/inventario-app/modelo_ia/preparar_dataset.py) / [Video\_de\_script\_funcionando](https://youtu.be/CgPppeHI3uQ?si=dUocFxgesu9_r-MW)





En la etapa de desarrollo del modelo de inteligencia artificial, la integrante Tihare Campusano fue responsable de implementar el script en Python **modelo\_ia.py** que automatiza el entrenamiento, la optimización avanzada y la exportación del modelo de clasificación de productos. Este script utiliza **TensorFlow** y **Keras**, apoyándose en la arquitectura **MobileNetV2** preentrenada en **ImageNet** mediante la técnica de **Transfer Learning** para aprovechar y adaptar conocimientos a la clasificación específica de productos (nuevo, usado, mal estado).

**Proceso de Desarrollo y Optimización**

El proceso implementado en el script está diseñado para maximizar la precisión (objetivo de 80-90%) e incluye optimizaciones avanzadas:

1. Preparación de Datos y Generadores:

Se definen los parámetros principales como el tamaño de imagen (224 X 224), la tasa de aprendizaje inicial y las épocas. Se cargan los conjuntos de entrenamiento, validación y prueba desde el dataset\_limpio usando generadores de datos que aplican Data Augmentation al conjunto de entrenamiento y se utilizan los pesos de clase precalculados para mitigar desequilibrios residuales durante el entrenamiento.

1. Construcción del Modelo con MobileNetV2:

Se utiliza MobileNetV2 como modelo base, cargando los pesos de ImageNet y excluyendo la capa superior original. Sobre esta base, se añaden un Head de Clasificación personalizado con capas densas más grandes (de 512 y 256 neuronas) que incluyen capas de BatchNormalization y Dropout junto con Regularización L2 para mejorar la estabilidad y prevenir el sobreajuste (overfitting). La capa final es una capa densa con activación softmax para las tres categorías.

1. Entrenamiento en Dos Fases:

El entrenamiento sigue la metodología de Transfer Learning en dos fases:

* + **Fase 1 (Head Training):** La base de **MobileNetV2** se mantiene **congelada** (base\_model.trainable = False), entrenando únicamente las nuevas capas añadidas (el *Head*) con el optimizador **Adam** y una tasa de aprendizaje base (**1e-4**).
  + **Fase 2 (Fine-Tuning):** El modelo base se hace **entrenable** (base\_model.trainable = True), pero se **congelan las primeras 50 capas** (las más abstractas) y se dejan entrenables las capas posteriores (más específicas). Se vuelve a compilar con una tasa de aprendizaje mucho menor (**BASE\_LR / 20.0**) para un ajuste fino y gradual de los pesos existentes.

1. Compilación y Callbacks:

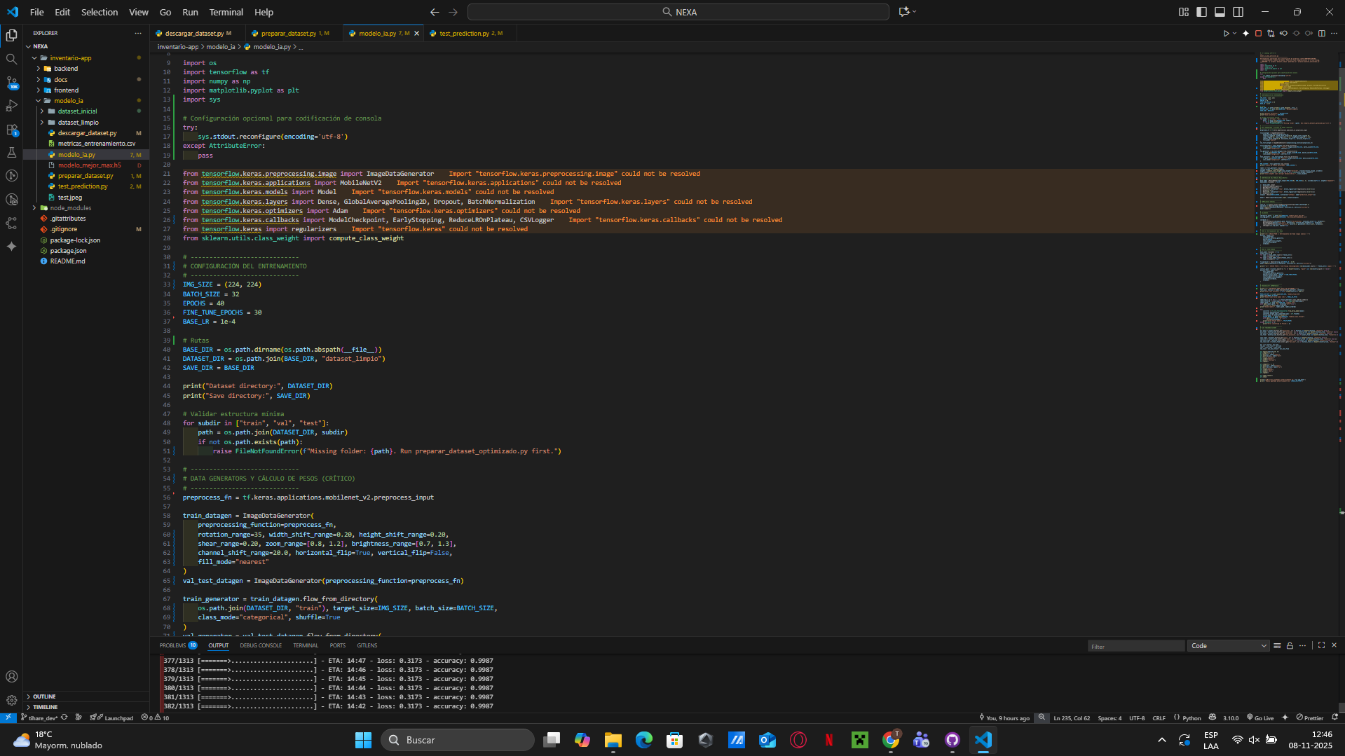
El modelo se compila utilizando el optimizador Adam, la función de pérdida categorical\_crossentropy y la métrica de accuracy. Se incorporan ModelCheckpoint para guardar la mejor versión basada en la precisión de validación y EarlyStopping para detener el entrenamiento si no hay mejora. Además, se utiliza CSVLogger para el registro del historial.

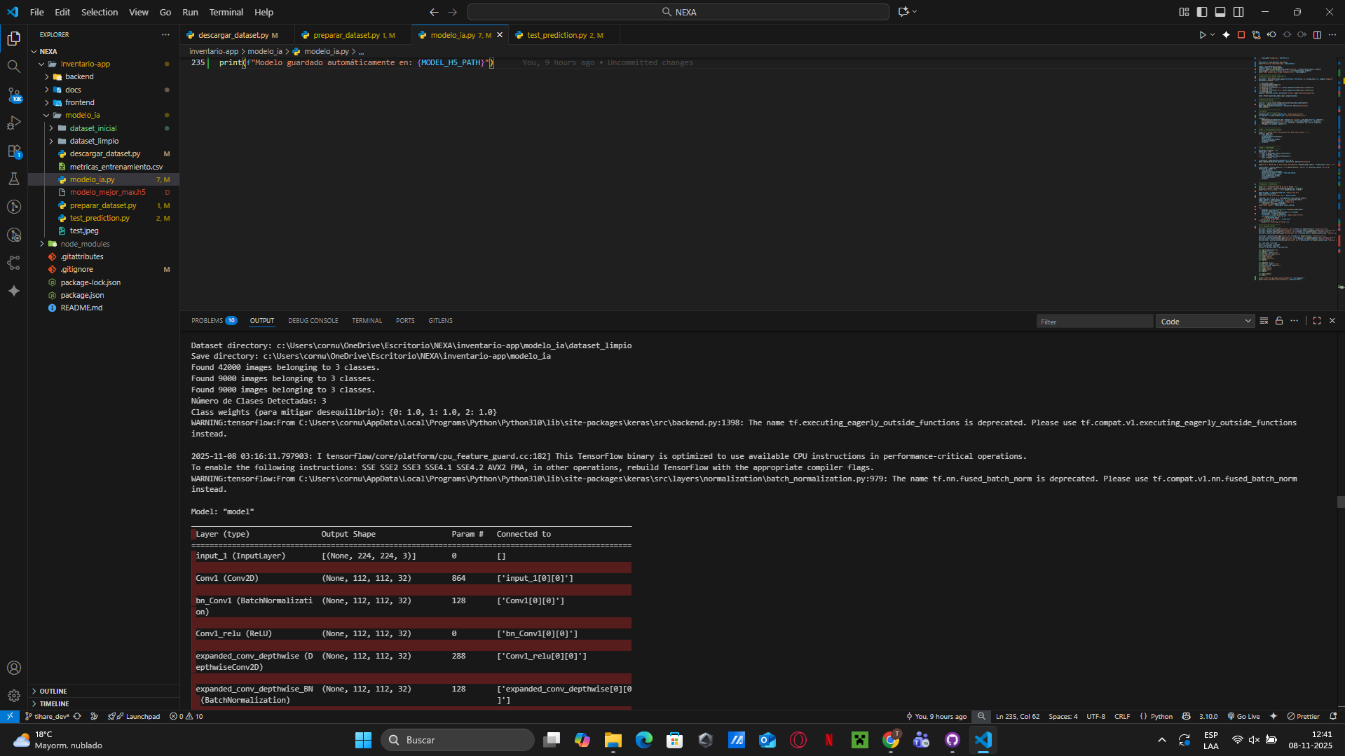
1. Evaluación y Exportación Final:

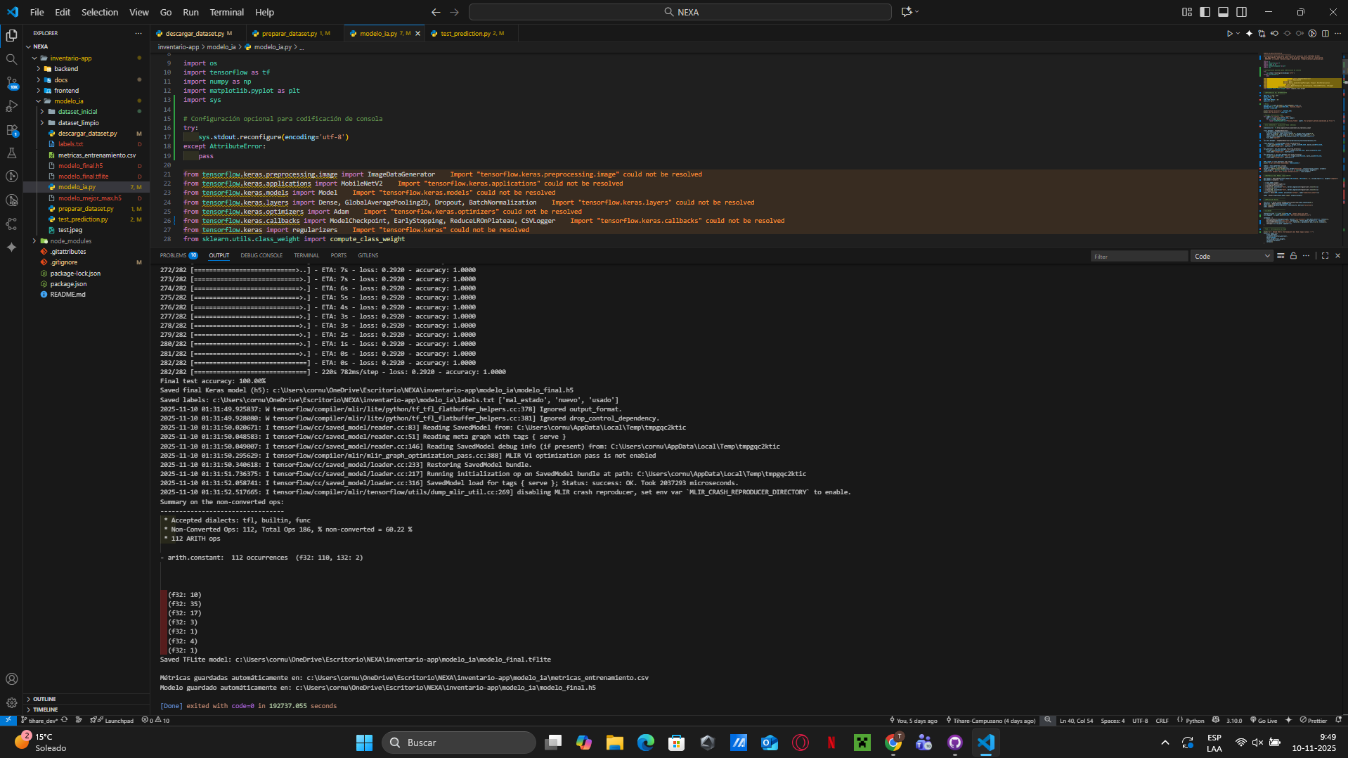
Una vez terminado el entrenamiento en ambas fases, se realiza una evaluación final en el conjunto de prueba (test\_generator). El modelo final se guarda en formato estándar Keras (.h5) y, posteriormente, se convierte a formato TensorFlow Lite (.tflite), optimizado para el despliegue eficiente en la aplicación móvil.

Este script no solo automatiza el entrenamiento, sino que incorpora **optimizaciones críticas** para asegurar que el modelo de IA final sea de alta precisión y esté listo para su implementación en la aplicación móvil.

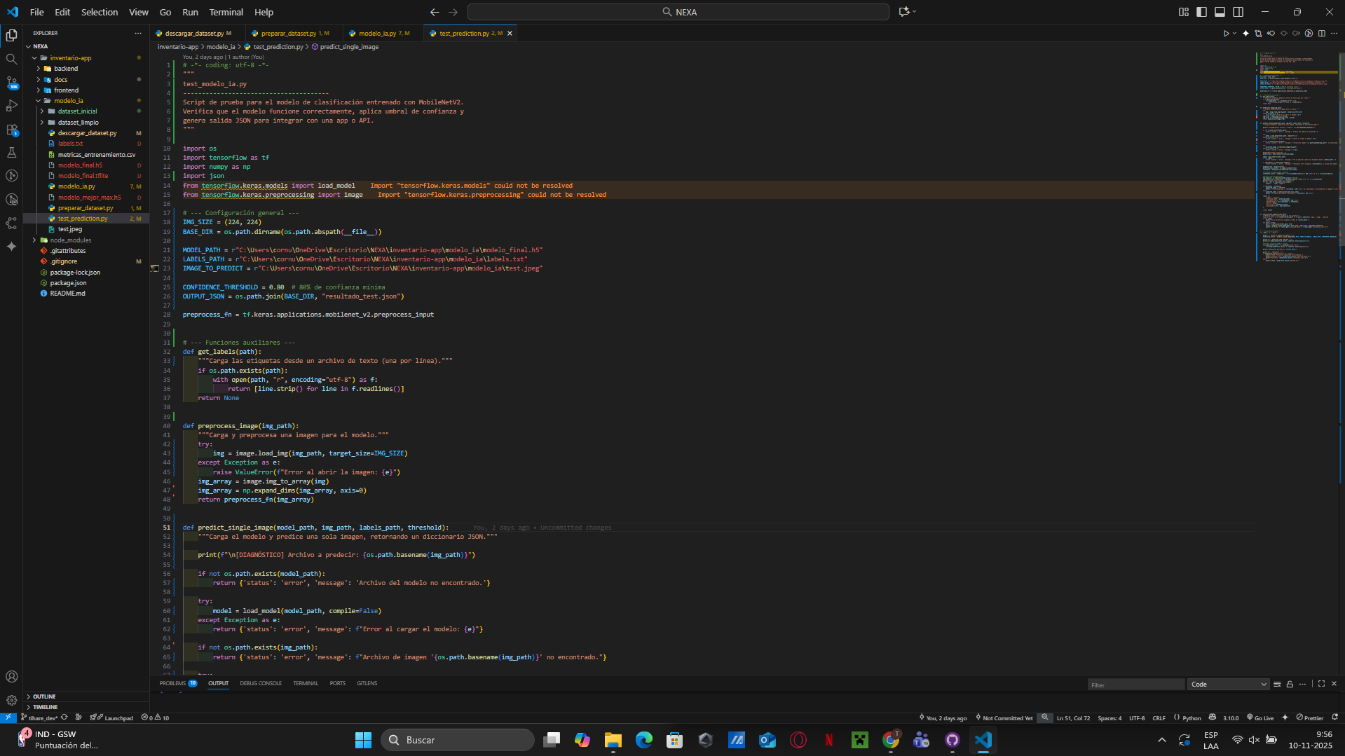
[modelo\_ia.py](https://github.com/Tihare-Campusano/NEXA/blob/tihare_dev/inventario-app/modelo_ia/modelo_ia.py) / [Video\_de\_script\_funcionando](https://youtu.be/OvEgtfgA2n0)

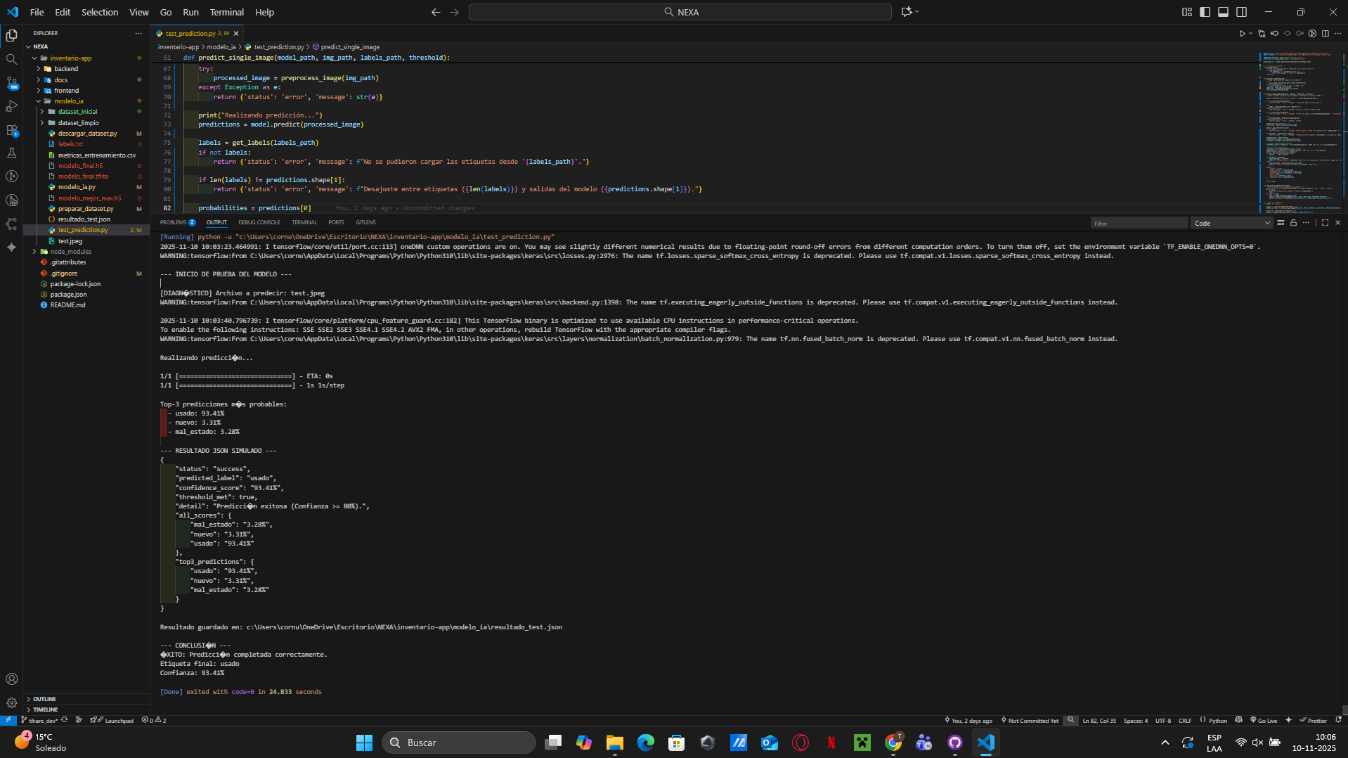






Para finalizar, la integrante Tihare Campusano implementó el script test\_modelo\_ia.py, el cual cumple el propósito de probar de forma aislada la funcionalidad del modelo de clasificación de inventario y generar una salida estandarizada. Este script primero carga el modelo Keras entrenado (modelo\_final.tflite) y las etiquetas de clase, y luego se encarga del preprocesamiento de la imagen de prueba (redimensionamiento a 224 X 224 píxeles y aplicación de la función específica de MobileNetV2). La función principal ejecuta la clasificación y realiza un post-procesamiento crítico al aplicar un umbral de confianza del 80% (CONFIDENCE\_THRESHOLD = 0.80): si la confianza es inferior a este valor, la etiqueta final se establece como "INCIERTO" en lugar de la clase predicha. Finalmente, toda la información de la predicción, incluyendo la etiqueta final, la puntuación de confianza y el detalle de todas las puntuaciones, se organiza en un objeto JSON para facilitar su consumo por parte de la aplicación móvil o sistemas externos.





La implementación del *backend* de inferencia de IA fue ejecutada por la integrante Tihare Campusano, con el objetivo de establecer un entorno de producción estable y escalable. Para lograrlo, se utilizó Docker para contenerizar la aplicación FastAPI, la cual integra un modelo TFLite de clasificación de productos y la conexión a Supabase. Este proceso incluyó la creación de un Dockerfile robusto que empleó la imagen base optimizada tiangolo/uvicorn-gunicorn-fastapi, resolviendo problemas de dependencias y configuración de *workers*. Una vez construida la imagen Docker (inventario-ia-api:vX), esta se registró y se desplegó en **Google Cloud Run** , lo que configuró automáticamente el servicio inventario-ia-api. Esta plataforma *serverless* garantiza la máxima disponibilidad, el escalado a cero cuando no hay tráfico, y provee el punto de acceso público (URL HTTPS) necesario para que el *frontend* consuma el servicio de clasificación en tiempo real.

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

Una captura de pantalla de una computadora

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

## Definición de Recursos y Costos asociados al Proyecto

La estimación de recursos y costos del proyecto considera tanto el aspecto humano como el tecnológico. En cuanto a los recursos humanos, se contemplan tres perfiles clave: Scrum Master/Developer, Product Owner/Developer y un Developer adicional. Cada uno tiene un costo mensual de $700.000 CLP y, dado que la duración real del proyecto según el cronograma es de 3,5 meses, el costo individual asciende a $2.450.000 CLP, lo que en conjunto representa un total de $7.350.000 CLP. En relación con los recursos tecnológicos, se incluyen los gastos necesarios para garantizar la operatividad del sistema: la publicación en Google Play Console con un pago único de $25.000 CLP, el servicio de hosting básico durante cuatro meses por $20.000 CLP, el uso del plan Pro de Supabase por el mismo período por $100.000 CLP y la adquisición de un dominio web anual por $15.000 CLP. En total, los costos tecnológicos ascienden a $160.000 CLP. De esta forma, la inversión global del proyecto alcanza los $7.510.000 CLP, asegurando tanto el capital humano necesario como la infraestructura tecnológica adecuada para cumplir los objetivos establecidos dentro de los plazos previstos en la carta Gantt.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Recurso Humano** | **Cantidad** | **Costo unitario** | **Tiempo** | **Costo total** |
| Scrum Master / Developer | 1 | $700.000 CLP/mes | 3.5 meses | $2.450.000 |
| Product Owner / Developer | 1 | $700.000 CLP/mes | 3.5 meses | $2.450.000 |
| Developer | 1 | $700.000 CLP/mes | 3.5 meses | $2.450.000 |
| **Total Recursos Humanos** |  |  |  | **$7.350.000 CLP** |

Tabla 9 Costos 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Recursos Tecnológicos** | **Cantidad** | **Costo unitario** | **Tiempo** | **Costo total** |
| Google Play Console (publicación app) | 1 | $25.000 CLP | Único | $25.000 |
| Hosting básico (Hostinger) | 1 | $5.000 CLP/mes | 3.5 meses | $20.000 |
| Supabase (plan Pro) | 1 | $25.000 CLP/mes | 3.5 meses | $100.000 |
| Dominio web | 1 | $15.000 CLP | Anual | $15.000 |
| Docker Team | 1 | $14.900 CLP / mes | 3.5 meses | $52.150 |
| Google Cloud run (Se paga por uso de recursos) | 1 | $3.000 CLP / mes | 3.5 meses | $10.500 |
| **Total Recursos Tecnológicos** |  |  |  | **$222.650 CLP** |

Tabla 10 Costos 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Recursos** | **Cantidad** | **Costo unitario** | **Tiempo** | **Costo total** |
| Recursos Humanos | 1 |  | 3.5 meses | **$7.350.000 CLP** |
| Recursos | 1 |  | 3.5 meses | **$222.650 CLP** |
| Utilidad(15%) | 1 |  |  | $1.135.897 CLP |
| **Total Recursos Humanos** |  |  |  | **$**8.708.547 **CLP** |

# CONCLUSIÓN

El proyecto NEXA – Gestión Logística y de Abastecimiento de Gendarmería representó un esfuerzo integral de diseño, implementación y validación de un sistema de inventario moderno que responde a una necesidad real de la institución. A través de la aplicación de metodologías ágiles como Scrum, fue posible organizar el trabajo de manera iterativa y lograr avances continuos, permitiendo un desarrollo ordenado y flexible ante los cambios que surgieron durante el proceso. Uno de los logros más relevantes fue la automatización de procesos críticos que antes se ejecutaban manualmente, como el registro de entradas y salidas de insumos, la actualización de stock en tiempo real y la generación de reportes. Estas funcionalidades no solo incrementan la eficiencia, sino que también fortalecen la transparencia y trazabilidad en la gestión de recursos públicos, aspecto fundamental en un organismo como Gendarmería. Asimismo, la incorporación de inteligencia artificial aportó un valor diferenciador, permitiendo la clasificación automática del estado de los productos mediante imágenes, lo que facilita la toma de decisiones más informadas y oportunas. Este componente innovador evidencia la capacidad del equipo para integrar tecnologías de vanguardia y aplicarlas a contextos reales del sector público. En términos de aprendizaje, este proyecto consolidó competencias clave del perfil de egreso en Ingeniería Informática, como la gestión de proyectos, la programación fullstack, el aseguramiento de la calidad del software y la implementación de medidas de seguridad. También fortaleció habilidades transversales como el trabajo en equipo, la comunicación efectiva y la capacidad de resolución de problemas. En conclusión, el desarrollo de NEXA demuestra cómo la tecnología puede ser un pilar estratégico en la modernización de instituciones públicas, optimizando tiempos, reduciendo costos y mejorando la transparencia. Más allá de un sistema de inventario, este proyecto constituye un aporte real al fortalecimiento de la gestión pública y marca un precedente para futuros desarrollos tecnológicos aplicados en el sector gubernamental.

# BIBLIOGRAFÍA

* Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., & Thomas, D. (2001). Manifesto for Agile Software Development. <https://agilemanifesto.org/>
* Schwaber, K., & Sutherland, J. (2020). The Scrum Guide. Scrum.org. <https://scrumguides.org/>
* Chollet, F. (2018). Deep Learning with Python. Manning Publications.
* Supabase. (2025). Supabase Documentation. <https://supabase.com/docs>
* TensorFlow. (2025). TensorFlow Documentation. <https://www.tensorflow.org/>
* Ionic Framework. (2025). Ionic Documentation. <https://ionicframework.com/docs>
* FastAPI. (2025). FastAPI Documentation. <https://fastapi.tiangolo.com/>
* Sommerville, I. (2011). Ingeniería de Software (9a ed.). Pearson.
* Pressman, R. S. (2014). Ingeniería del Software: Un enfoque práctico. McGraw-Hill.
* Kruchten, P. (2003). The Rational Unified Process: An Introduction. Addison-Wesley.
* Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer.
* Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.
* Keras. (2025). Keras Documentation. <https://keras.io/>
* Firebase. (2025). Firebase Documentation. <https://firebase.google.com/docs>
* Hostinger. (2025). Documentation & Tutorials. <https://www.hostinger.com/tutorials>

# ANEXOS

* [Manual de usuario](https://docs.google.com/document/d/10DO4xNrqlbllR7XWHyZe3ug1a4CaFscme9ME615fQYM/edit?usp=sharing)
* [Implementación IA](https://docs.google.com/document/d/1CQOhjWpgdNJEuHf7ng9L2JWJN_yvwMSP/edit?usp=sharing&ouid=112309288411702756749&rtpof=true&sd=true)
* [Documentación DB](https://docs.google.com/document/d/1_bSzOYffXsO3OSPqO3Xvlv6DZ_uDAaTw/edit?usp=sharing&ouid=112309288411702756749&rtpof=true&sd=true)
* [Guía técnica NEXA](https://docs.google.com/document/d/1MUJNyMuudrGaFmHgQ9dFOv0wtg2LRHrC/edit?usp=sharing&ouid=112309288411702756749&rtpof=true&sd=true)
* [Requerimientos](https://duoccl0.sharepoint.com/:x:/s/NEXAProyectodeCapstone/EVdzoSq6-LROihlvJg-rO4UBccTff_BuVEnA4XJSspckkg?e=fjiPgQ)
* [Prototipo Figma](https://www.figma.com/design/ePFoUyqZOREHeFDsnLPPZK/NEXA?node-id=0-1&t=op8m8C1cskA0ywQA-1)
* [Repositorio del proyecto](https://github.com/Tihare-Campusano/NEXA/tree/tihare_dev)
* [Presentación del proyecto](https://www.canva.com/design/DAG01XMeoqM/-vT3hgZEuxtAKlR6t-iPgg/edit?utm_content=DAG01XMeoqM&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton)