

Escuela Superior Politécnica del Litoral

Facultad de Ingeniería en Electricidad y Computación

Sistema de gestión de inventario y seguimiento automatizado de elementos en bodega

TECH-327

Proyecto Integrador

Previo a la obtención del Título de:

Ingeniero en Ciencias de la Computación

Presentado por:

Alcivar Castro José Emanuel

Viscarra Zambrano Daniel Rodrigo

Guayaquil – Ecuador

Año: 2023 - 2024

Dedicatoria

Este trabajo va dedicado a mi Mamá, a mi Papá y a mis Hermanos, quienes a lo largo de estos años siempre tuvieron fe en mí y estuvieron dándome su apoyo y consejos. A mi papá, Marcos Alcivar, que cuando dije que quería ir a Espol, me inscribió en un curso de nivelación diciéndome "porque sé que tienes talento para esto"; a mi mamá, Nelly Castro, que desde el primer día de clases del pre ya me llamaba "mi ingeniero politécnico"; a mis hermanos, Helen y Elías, que con sus acciones y palabras me decían "tú puedes, tú puedes"; a mí, por saber valorar el apoyo, por no rendirme y por haberme esforzado para lograr la meta.

Papá, Mamá, Hermanos, ¡¡El equipo lo logró, misión cumplida, tres de tres!!

José Emanuel Alcivar Castro

Dedicatoria

El presente proyecto se lo dedico a mi padre, mi madre, mi hermana y a mi novia, que estuvieron apoyándome siempre durante toda mi etapa universitaria.

Daniel Viscarra

Agradecimientos

Agradezco a Dios, por la salud, la vida y la sabiduría para lograr esta meta; a mis padres y mis hermanos, por todo su apoyo, consejos y confianza en estos años; a mis pastores, Milton Acosta y Gioconda Ibarra, quienes en cada oportunidad me preguntaban si iba bien en la universidad e incluso me saludaban con un “¿qué tal, ingeniero?”; a mis amigos, con quienes pasé buenos momentos en esta etapa universitaria.

Agradezco también a mi primo, que es como mi hermano, con quien en medio de tardes/noches de videojuegos compartimos experiencias universitarias e ideas de superación.

Gracias a todos.

José Emanuel Alcivar Castro

Agradecimientos

Agradezco a mis padres, a mi hermana y a mi novia por darme fuerza, sabiduría, ánimos y apoyo en toda esta etapa de mi vida.

A mis profesores de la universidad que me dieron el conocimiento y me enseñaron como debo aplicarlo.

Finalmente, a todas las personas que nos guiaron y nos brindaron apoyo durante la realización de este proyecto.

Daniel Viscarra

Declaración Expresa

Nosotros, José Emanuel Alcivar Castro y Daniel Rodrigo Viscarra Zambrano, acordamos y reconocemos que:

La titularidad de los derechos patrimoniales de autor (derechos de autor) del proyecto de graduación corresponderá al autor o autores, sin perjuicio de lo cual la ESPOL recibe en este acto una licencia gratuita de plazo indefinido para el uso no comercial y comercial de la obra con facultad de sublicenciar, incluyendo la autorización para su divulgación, así como para la creación y uso de obras derivadas. En el caso de usos comerciales se respetará el porcentaje de participación en beneficios que corresponda a favor del autor o autores. La titularidad total y exclusiva sobre los derechos patrimoniales de patente de invención, modelo de utilidad, diseño industrial, secreto industrial, software o información no divulgada que corresponda o pueda corresponder respecto de cualquier investigación, desarrollo tecnológico o invención realizada por nosotros durante el desarrollo del proyecto de graduación, pertenecerán de forma total, exclusiva e indivisible a la ESPOL, sin perjuicio del porcentaje que nos corresponda de los beneficios económicos que la ESPOL reciba por la explotación de nuestra innovación, de ser el caso.

En los casos donde la Oficina de Transferencia de Resultados de Investigación (OTRI) de la ESPOL comunique a los autores que existe una innovación potencialmente patentable sobre los resultados del proyecto de graduación, no se realizará publicación o divulgación alguna, sin la autorización expresa y previa de la ESPOL.

Guayaquil, 15 de febrero del 2024.



José Emanuel Alcivar Castro



Daniel Rodrigo Viscarra Zambrano

Evaluadores

Luis Eduardo Mendoza M., Ph.D.

Profesor de Materia

Miguel Andrés Realpe R., Ph.D.

Tutor de proyecto

Resumen

El Centro de Investigación, Desarrollo e Innovación de Sistemas Computacionales (CIDIS) es un centro de investigación de la ESPOL que cuenta con varios recursos como: equipos, herramientas y repuestos tecnológicos. Estos se encuentran almacenados o son usados en proyectos de investigación por los miembros del centro.

El CIDIS mantiene un inventario para todos estos recursos; sin embargo, la gestión de éste ocasiona que sea dificultoso determinar de manera fiable qué recursos hay y su stock, así como sus ubicaciones y las personas que actualmente los están usando, debido a que los datos son registrados de manera manual en una plantilla de inventario realizada en Excel.

Para solucionar estas falencias, se optó por implementar un prototipo funcional de un sistema de inventario automatizado que permita un manejo eficiente de los registros de CIDIS, con la finalidad de optimizar las tareas de registro, consulta y seguimiento de los recursos de la organización.

Las tecnologías usadas para el desarrollo del sistema fueron las siguientes: Angular para la interfaz gráfica, Express para la API y gestión de información, MySQL como motor de base de datos, servicio CAS de ESPOL para la autenticación de los usuarios y, un modelo básico de Deep Learning para realizar detección y reconocimiento de los equipos almacenados en bodega.

Como resultado, se obtuvo un sistema que solo puede ser accedido por usuarios de Espol que pertenezcan al CIDIS. Este sistema se encuentra dividido en secciones, en base a los roles de los usuarios que se encuentran dentro del proceso de gestión del inventario (Administrador, Custodio, Usuario General).

En conclusión, el prototipo resultó ser un sistema que proporciona mayor seguridad y fiabilidad en el manejo del inventario de los recursos; además, agiliza los procesos de préstamo dentro del CIDIS.

Palabras Clave: Recursos, consultas, registros, solicitud de préstamo, usuarios.

Abstract

The Centro de Investigación, Desarrollo e Innovación de Sistemas Computacionales (CIDIS) is a research center of the ESPOL that has several resources such as: equipment, tools and technological spare parts. These are stored or used in research projects by members of the center.

CIDIS maintains an inventory for all these resources; however, the management of this inventory makes it difficult to reliably determine what resources there are and their stock, as well as their locations and the people who are currently using them, because the data are recorded manually in an Excel inventory template. To solve these shortcomings, the decision was made to implement a functional prototype of an automated inventory system that allows efficient management of CIDIS records, in order to optimize the tasks of registration, consultation and monitoring of the organization's resources.

The technologies used for the development of the system were the following: Angular for the graphical interface, Express for the API and information management, MySQL as database engine, ESPOL CAS service for user authentication and, a basic Deep Learning model to perform detection and recognition of equipment stored in warehouse.

As a result, we obtained a system that can only be accessed by Espol users belonging to CIDIS. This system is divided into sections, based on the roles of the users within the inventory management process (Administrator, Custodian, General User).

In conclusion, the prototype turned out to be a system that provides greater security and reliability in the management of the inventory of resources, in addition to streamlining the loan processes within the CIDIS.

Keywords: *Resources, queries, records, loan requests, users.*

Índice general

1.	Resumen.....	I
2.	<i>Abstract</i>	III
3.	Índice general.....	IV
4.	Índice de figuras.....	VII
5.	Índice de tablas	IX
1.	Capítulo 1.....	1
1.1	Introducción	2
1.2	Descripción del problema.....	2
1.3	Justificación del problema.....	3
1.4	Objetivos	5
1.4.1	Objetivo general.....	5
1.4.2	Objetivos específicos	5
1.5	Marco teórico	6
1.5.1	Sistema de inventario.....	6
1.5.2	Experiencia de usuario (UX)	7
1.5.3	Trabajos previos.....	7
1.5.4	Deep Learning.....	9
1.5.5	Modelo YOLO	10
1.5.6	Modelo Detectron2	10
1.5.7	Modelo Retinanet.....	11
2.	Capítulo 2.....	12
2.1	Análisis.....	13
2.1.1	Requerimientos	13
2.1.2	Usuarios y beneficiarios de la solución	14
2.1.3	Análisis de confiabilidad y compatibilidad.....	14

2.1.4	Análisis de los modelos de IA	16
2.1.5	Dataset (modelo de Deep Learning)	17
2.2	Historias de usuario	17
2.3	Diseño de la solución (Sistema de inventario)	19
2.3.1	Escenarios	20
2.3.2	Vista Lógica	20
2.3.3	Vista de Desarrollo o Implementación	23
2.3.4	Vista de Procesos	26
2.3.5	Vista de Despliegue	27
2.4	Diseño de la solución (Modelo de Deep Learning)	28
2.4.1	Esquema general de pasos	28
2.4.2	Diagramas de casos de uso	30
2.4.3	Entrenamiento	31
2.5	Plan de Implementación	31
3.	Capítulo 3	34
3.1	Desarrollo	35
3.2	Resultados y análisis (Sistema de inventario)	37
3.2.1	Vista de Login	37
3.2.2	Vista del administrador	38
3.2.3	Vista del custodio	39
3.2.4	Vista del usuario general	40
3.3	Resultados y análisis (Módulo de IA)	40
3.3.1	Entrenamiento	40
3.3.2	Detecciones correctas	43
3.3.3	Detecciones incorrectas	44

3.3.4	Ambiente de desarrollo.....	45
3.4	Pruebas con el cliente final.....	46
3.5	Análisis de costos.....	47
4.	Capítulo 4.....	48
4.1	Conclusiones.....	49
4.2	Recomendaciones.....	49
6.	Referencias.....	51
7.	Apéndice A.....	56
	Requerimientos del proyecto.....	56
8.	Apéndice B.....	58
	Historias de usuario.....	58

Índice de figuras

Figura 2.1: Diagrama de casos de uso del sistema de inventario.	20
Figura 2.2: Tablas de la sección de usuarios y roles.	21
Figura 2.3: Tablas de la sección de registros de préstamos y solicitudes.....	22
Figura 2.4: Diagrama de bloques general del sistema de inventario.	23
Figura 2.5: Diagrama de componentes del sistema de inventario.	26
Figura 2.6: Diagrama de despliegue del sistema de inventario.	27
Figura 2.7: Pasos generales del desarrollo del modelo de IA.....	28
Figura 2.8: Diagrama de casos de uso (situación actual de CIDIS).	30
Figura 2.9: Diagrama casos de uso (Solución con modelo de IA).	30
Figura 3.1: Tareas asignadas a cada integrante del equipo de desarrollo.	35
Figura 3.2: Commits en el Frontend.....	36
Figura 3.3: Commits en el Backend.	36
Figura 3.4: Ventana de Login del sistema.	37
Figura 3.5: CAS de Espol incorporado en el sistema.....	37
Figura 3.6: Ventana para un usuario no registrado en el sistema.	38
Figura 3.7: Vista de un administrador	38
Figura 3.8: Vista de un custodio.....	39
Figura 3.9: Vista de un usuario general.....	40
Figura 3.10: Matriz de confusión del modelo de IA.....	41
Figura 3.11: Matriz de confusión normalizada.....	42
Figura 3.12: Gráficos de métricas del modelo.....	43
Figura 3.13: Detección correcta de un CPU.....	43

Figura 3.14: Detección correcta de una memoria RAM.....	44
Figura 3.15: Error al detectar GPUs.	44
Figura 3.16: Falso positivo para una PSU.	45

Índice de tablas

Tabla 2.1: Historia de usuario, para usuario con rol administrador.....	19
Tabla 2.2: Plan de implementación de las historias de usuario.	33

1. Capítulo 1

1.1 Introducción

En un mundo empresarial cada vez más dinámico y competitivo, la eficiencia en la gestión de recursos se ha convertido en una prioridad estratégica para las organizaciones. Uno de los recursos más críticos para cualquier entidad es su inventario: un conjunto complejo y dinámico de activos que debe ser gestionado con precisión para garantizar la continuidad operativa y la satisfacción de los usuarios finales.

La implementación de un sistema de inventario efectivo no solo es una necesidad, sino un imperativo para las organizaciones que buscan mantenerse ágiles; y para poder implementarlo de una manera correcta y eficiente, se necesitan tanto los conocimientos y habilidades de un profesional de computación, como un correcto análisis de requerimientos de la correspondiente entidad que requiera el sistema.

1.2 Descripción del problema

El Centro de Investigación, Desarrollo e Innovación de Sistemas Computacionales (CIDIS), es un centro de investigación de la ESPOL que mantiene en su inventario varios recursos, mismos que pueden ser equipos, herramientas, repuestos, etc. Un grupo de estos posee un código de inventario de ESPOL y se encuentran registrados en el sistema de la universidad. Sin embargo, el grupo restante, entre los que hay equipos pequeños, repuestos, partes y accesorios, no posee dicho registro, lo cual dificulta su seguimiento y organización. Adicionalmente, el sistema de inventario de ESPOL no es accesible para el personal del CIDIS y tampoco cuenta con los requerimientos específicos del Centro.

Para distribuir responsabilidades, se designan custodios de los bienes, encargados de velar por la integridad, el uso, ubicación y cuidado del recurso encargado. De la forma en que

actualmente se realizan las actividades de gestión de inventario en CIDIS, es difícil determinar de manera fiable qué recursos hay y su stock, así como sus ubicaciones y las personas que actualmente los están usando. Los datos se registran manualmente en una plantilla de inventario realizada en Excel y las cantidades de stock registradas en dicha plantilla suelen diferir de las que realmente existen en bodega. Las razones para estas inconsistencias son variadas. Por ejemplo, retiradas accidentales de recursos, procesos defectuosos, retiros no autorizados, etc. Por ello, la gestión de inventario de la bodega solamente puede ser tan buena como lo sea la calidad de los datos disponibles.

Respecto a lo expuesto, se requiere implementar un sistema de gestión de inventario enfocado en las necesidades del CIDIS. Adicionalmente, se solicita un módulo demo de reconocimiento automatizado de equipos en bodega.

1.3 Justificación del problema

Dentro de las actividades consideradas cruciales para el correcto funcionamiento de un negocio, compañía, centro u organización, está el control de inventario. Con un correcto control de inventario se puede garantizar que se conozca con certeza los recursos disponibles, su respectivo stock y demás información que se considere necesaria para los fines que se persiguen.

La implementación de un sistema de manejo de inventario trae marcados beneficios ante problemas que comúnmente ocurren cuando no se cuenta con un sistema o, si es precario y/o deficiente. Entre estos beneficios podemos listar [15, 16, 17, 19]:

- **Permite optimización financiera:** Implementar un correcto sistema de inventario hace posible que se mantenga un control de los recursos disponibles. De esta forma se evitan

compras innecesarias o insuficientes y, por lo tanto, permite que las pérdidas de capital por recursos no utilizados se reduzcan.

- **Hace posible tomar mejores decisiones:** Tener de manera precisa la información acerca de la disponibilidad de los recursos, posibilita la toma informada y estratégica de decisiones en cuanto a compras, reabastecimiento y gestión del stock.
- **Reduce los errores humanos:** Automatizar procesos ayuda a la reducción de inconsistencias, producto de errores humanos como, por ejemplo, un mal registro de datos. Esto puede generar un efecto positivo en la precisión de los registros y, de este modo, potenciar la eficiencia operativa.
- **Prevenir robos y pérdidas:** Mantener un registro a detalle de los recursos puede ayudar a la prevención de pérdidas y robos, al tener un control de la información relacionada a cada recurso. Incluso puede ayudar a que miembros del centro y terceros desistan de incurrir en actos fraudulentos.
- **Mejorar eficiencia operativa:** Las tareas de inventario relacionadas al registro y seguimiento de los recursos pueden ser realizadas de forma más ágil y eficiente, mediante la implementación de un sistema; esto permite ahorrar tiempo y recursos de modo que el personal pueda centrarse en otras actividades estratégicas de la organización [15, 16, 17]. El CIDIS posee una gran influencia en el futuro de muchos trabajos de investigación, ya que es el encargado de proporcionar equipos, recursos y herramientas para proponer, desarrollar e implementar muchos proyectos. Por lo tanto, si el centro de investigación posee mayor conocimiento de la disponibilidad de sus recursos, agilizará las decisiones con respecto a la planificación de proyectos de investigación futuros, a diferencia de otros departamentos que no posean esta información.

Estos son los beneficios que traería la implementación de un sistema de inventario en CIDIS, que daría al Centro una base sólida para la gestión eficaz de sus recursos y contribuiría a la eficacia de sus operaciones.

1.4 Objetivos

1.4.1 Objetivo general

Implementar un prototipo funcional de un sistema de inventario automatizado que permita un manejo eficiente de los registros de CIDIS, con la finalidad de optimizar las tareas de registro, consulta y seguimiento de los recursos de la organización.

1.4.2 Objetivos específicos

1. Realizar análisis de los requisitos del sistema de inventario
2. Desarrollar un sistema de Frontend y Backend que permita realizar las respectivas operaciones CRUD de la información relacionada a cada recurso.
3. Entrenar un modelo de Deep Learning para realizar la detección y reconocimiento de los equipos presentes en imágenes de la bodega.
4. Implementar un programa demo que haga uso del modelo entrenado y reporte la información de existencias de los recursos.
5. Crear documentación técnica detallada para futuras referencias y mantenimiento del sistema.

1.5 Marco teórico

1.5.1 Sistema de inventario

Los inventarios son todos los elementos o productos utilizados en la producción, como materias primas y productos en proceso; así como en actividades de apoyo, como suministros de mantenimiento y reparación; y para satisfacer las necesidades del cliente, como productos terminados y repuestos [9]. Ellos constituyen una de las inversiones más significativas de las empresas en comparación con sus otros activos, ya que son esenciales para su correcta operatividad. Sin embargo, en la práctica empresarial, muchos gerentes cometen el error de subestimar la importancia de llevar a cabo una gestión eficiente de los mismos [9].

Se conoce como sistema de inventario al conjunto de normas, métodos y procedimientos que son utilizados con la intención de realizar la debida planificación y control de aquellos productos o materiales que son utilizados por la organización [1].

Existen 5 metas principales de los sistemas de inventario [10]: asegurar operaciones independientes, adecuar la producción a las fluctuaciones en la demanda, permitir flexibilidad en los programas de producción, establecer un margen de seguridad para enfrentar retrasos en la entrega de materias primas y disminuir los costos mediante compras en grandes volúmenes.

Una forma de implementar esto es mediante un sistema compuesto de Frontend y Backend que realizan las respectivas operaciones CRUD (Create, Read, Update and Delete) con la información contenida en la correspondiente base de datos.

1.5.2 *Experiencia de usuario (UX)*

En el ámbito de las páginas web, las aplicaciones móviles y otros recursos similares, UX hace referencia al conjunto de atributos que permiten conocer qué tan satisfactoria es la navegación por un sistema; esto puede involucrar el funcionamiento técnico del sistema, la organización estructural y el valor agregado que aporta el contenido que el sistema le brinda al usuario [14].

Entre los pilares que conforman esto, tenemos [14]:

- **Atractivo visual:** Es importante que el sistema cuente con diseño que sea visualmente agradable, llamativo y coherente.
- **Utilidad:** El sistema debe ser altamente funcional, brindar al usuario un valor agregado que le ayude a resolver problemas y satisfacer necesidades.
- **Credibilidad:** Es importante que el sistema esté libre de errores y cumpla con las expectativas que genera.
- **Intuitividad:** El diseño debe permitir que el usuario identifique fácilmente las funcionalidades que posee el sistema y pueda sacarle el máximo provecho.

1.5.3 *Trabajos previos*

En la Universidad de los Llanos, se realizó una tesis que tuvo como objetivo principal, mejorar las diversas actividades relacionadas con el registro preciso de dispositivos electrónicos a través de un sistema de gestión [6]. Para lograr esto, se inició con una fase inicial de investigación y orientación, durante la cual se analizaron las características teóricas y tecnológicas necesarias para el programa. Con respecto a la creación y de gestión de la base de datos, se realizó con MySQL Workbench, la cual mostró resultados muy favorables con respecto a transacciones y tiempos de búsquedas; además que no presentó ningún inconveniente al momento de trabajar con otras herramientas. Para el desarrollo web se usó Laravel, que demostró ser una herramienta eficaz

gracias a su gran compatibilidad con MySQL Workbench, además de que facilitó el manejo de las rutas de la aplicación, así como la generación de URLs amigables que ayudaron a mejorar el posicionamiento web. La implementación de este proyecto demostró ofrecer un servicio de mayor calidad al proporcionar una mejora en la administración [6].

La Escuela Superior Politécnica Agropecuaria de Manabí ha registrado en su archivo digital una tesis que tuvo como objetivo el desarrollo e implementación de un sistema de gestión de inventario, compra y venta de productos agrícolas y generación de reportes, para facilitar la toma de decisiones en la Asociación de Mujeres Comunitarias del Cantón Tosagua [7]. Con respecto al desarrollo del Backend se utilizó como Api Rest, Visual Studio C#, debido a su alta compatibilidad con plataformas, rendimiento, seguridad, escalabilidad y facilidad de uso; además, puede consumir y devolver datos en diferentes formatos, como JSON, XML o HTML. Para la creación de manejo y base de datos se usó SQL Server, el cual mostró resultados óptimos al momento de realizar búsquedas o modificaciones en datos. En cuanto al Frontend se utilizó Angular, ya que proporcionó un ambiente de desarrollo tanto para aplicación web como para móvil; además, poseía una alta compatibilidad con los navegadores y también proporcionaba componentes personalizados y reutilizables. En base a encuestas de satisfacción realizadas, se pudo constatar que la operatividad y usabilidad del sistema fue eficaz y eficiente a la hora de manejar la gestión de los procesos [7].

En el repositorio digital de la Universidad Técnica de Ambato, está disponible una tesis que tiene como objetivo mejorar la atención al cliente, al utilizar un sistema que controla bases de datos confiables [8]. La importancia de emplear programas de gestión para optimizar actividades

es resaltada en este estudio. Para la modelación de la base de datos se utilizó SQL Server, ya que proporcionó un potente entorno gráfico de administración que permitió crear entidades (Tablas) y la racionalización de estas de una forma fácil, y de usar procedimientos almacenados que se utilizaron frecuentemente para el manejo de consultas, ya sea de búsqueda, ingreso, modificación y eliminación de productos, administradores, clientes, servicios y ventas. Para el desarrollo web se utilizó visual estudio 2010 debido a su eficacia y que está orientado a objetos, lo que permitió la reutilización de código para crear constructores propios, funciones, clases etc. La implementación total del proyecto resultó en una percepción positiva, lo que a su vez aumentó la eficacia de la empresa en la sistematización de procesos [8].

1.5.4 Deep Learning

También conocido como Aprendizaje Profundo es, de manera básica, una red neuronal con tres o más capas. Dichas neuronas le dan a la red la capacidad de “aprender” mediante el análisis de grandes cantidades de datos. En estas redes, una sola capa ya le da la capacidad de realizar predicciones aproximadas; sin embargo, el aumento de capas, a las que se conoce como capas ocultas, permite que la precisión de la predicción se optimice y refine [2].

Mediante el uso de estas redes neuronales, los algoritmos de aprendizaje profundo son capaces de imitar las acciones realizadas por el cerebro humano. En estos algoritmos cada una de las capas ocultas recibe y procesa la información que obtiene de la capa anterior, permitiendo que la red pueda “aprender” de los datos que está procesando [3].

1.5.5 *Modelo YOLO*

Dentro del campo de la Visión Artificial, la detección de objetos es una técnica que consiste en localizar, dentro de una imagen de entrada, objetos individuales que pertenecen a diferentes clases predeterminadas. Esta localización consiste en trazar un cuadro delimitador alrededor del objeto que ha sido detectado [4].

You Only Look Once (YOLO), es un algoritmo que hace uso de Deep Learning y redes neuronales convolucionales para detectar objetos presentes en imágenes, la principal característica distintiva entre YOLO y sus competidores es lo que está indicado en su nombre, para detectar los objetos este algoritmo solamente necesita “ver” la imagen una sola vez, esto le permite ser más rápido, aunque sacrifica algo de exactitud en las detecciones. Esta característica permite que el algoritmo pueda ser utilizado incluso en detecciones en tiempo real en videos de hasta 30 FPS [5].

1.5.6 *Modelo Detectron2*

Detectron2 es una biblioteca de próxima generación que ha sido desarrollada por Facebook AI Research (FAIR) y proporciona algoritmos de detección y segmentación. Este modelo tiene compatibilidad con variados proyectos acerca de visión por ordenador y es el sucesor de los modelos *Detectron* y *Maskrcnn-benchmark* [11].

El modelo consiste en una reescritura desde cero de su antecesor, Detectron. En este modelo se puede encontrar [12]:

- Recetas de entrenamiento para detección de objetos, segmentación semántica, segmentación panóptica, segmentación de instancias, además de detección de puntos clave.
- Más de 80 modelos que han sido previamente entrenados y permiten realizarles ajustes (o entrenarlos de nuevo).

- Soporte para conjuntos de datos populares en visión por computador, como LVIS, ADE20k, Cityscapes, COCO, PASCAL VOC.

1.5.7 Modelo Retinanet

Este es un modelo de detección de objetos que, para afrontar el problema del desequilibrio de clases durante el entrenamiento, utiliza una función de pérdida focal [13]. Está conformada por una red única y unificada que a su vez está compuesta por una red troncal y dos subredes que están dedicadas a tareas específicas [13]. En este modelo, la columna vertebral es una red convolucional externa y es la responsable de calcular un mapa de características convolucionales sobre una imagen de entrada completa. Posterior a esto, la primera subred está encargada de realizar la clasificación de objetos convolucionales a partir de la salida de la red troncal. Luego de esto, en la segunda subred, se realiza una regresión de cuadro delimitador convolucional. Los autores proponen un diseño simple en ambas subredes, este diseño está propuesto específicamente para la detección densa en una sola capa [13].

2. Capítulo 2

2.1 Análisis

2.1.1 *Requerimientos*

Para una correcta obtención de los requerimientos de la solución se realizaron reuniones con el cliente de CIDIS. En estas reuniones, se recopilaron las necesidades de funcionalidades de software que el sistema debe tener, de modo que pueda satisfacer lo que se requiere implementar en el Centro.

El sistema debe tener tres roles de usuario (administrador, custodio, usuario general), con funcionalidades bien definidas para cada uno.

La descripción general de cada rol se detalla a continuación:

- **Administrador:** Es quien tiene control y acceso total a la información registrada en el sistema, puede ver/agregar todos los usuarios, ver/agregar todos los recursos y sus custodios, además de conocer el estado de cada recurso. Este rol también puede ver el estado de todas las solicitudes de préstamo registradas en el sistema.
- **Custodio:** Usuario que tiene a su cargo varios bienes y recursos del CIDIS. Su labor en el sistema se enfoca a tener un control de seguimiento de todos los recursos que estén a su cargo. Además, este usuario es quien decide si un préstamo se autoriza o no.
- **Usuario general:** Este es el rol de usuario más bajo. Sus accesos al sistema se limitan a poder ver la lista de recursos que están disponibles para préstamo (con información limitada para cada recurso) y ver la lista de solicitudes de préstamo que ha realizado y el estado de éstas.

Cada usuario interactuará con la información y los demás usuarios según su rol asignado.

También se recopilaron requerimientos no funcionales, mismos que incluyen los tiempos de

respuesta, tener una interfaz de usuario intuitiva y demás funciones relacionadas a brindar una buena experiencia de usuario.

También se obtuvieron los requerimientos para el modelo de Deep Learning, el cual es un módulo demo que inicialmente funcionará de forma independiente al sistema de inventario, pero que se espera que se integren en un solo sistema en trabajos futuros.

La lista completa de requerimientos de este proyecto se encuentra en el [Apéndice A](#) de este documento.

2.1.2 *Usuarios y beneficiarios de la solución*

Los usuarios y beneficiarios de la solución son todas las personas que forman parte del equipo humano que labora en CIDIS. En determinados casos, personas externas al Centro también podrían usar y beneficiarse esta solución.

2.1.3 *Análisis de confiabilidad y compatibilidad*

Alcance de la solución: Esta solución permitirá registrar códigos de identificación únicos para cada recurso del Centro, de modo que se pueda hacer un seguimiento preciso de los mismos. Facilitará la gestión y acceso rápidos a la información acerca de cada recurso registrado en el sistema, como su custodio, el tipo de recurso, su estado, y demás información relevante acerca del mismo. También permitirá realizar el seguimiento y gestión de las actividades que involucren algún recurso, tales como solicitudes, préstamos, devoluciones.

Limitaciones de la solución: Dependiendo del caso, este tipo de sistema puede llegar a tener costos iniciales significativamente altos debido a adquisición de software y/o hardware,

incluso costos debido a la capacitación de personal; a esto se le debe sumar los costos que puedan surgir debido a su mantenimiento. También es importante considerar que la curva de aprendizaje para el manejo del sistema puede resultar algo compleja para determinados usuarios. Además de esto, un sistema de estas características implica dependencia tecnológica, en caso de alguna falla en las tecnologías utilizadas, esto podría afectar de manera crítica la disponibilidad de la información. Otro punto importante para tener en cuenta es que podrían presentarse desafíos en temas de compatibilidad al momento de querer integrar el sistema con algún otro sistema del Centro.

Riesgos de la solución: Existe el riesgo de errores humanos al momento de ingresar los correspondientes datos, errores de este tipo pueden afectar la precisión del sistema y conllevar a toma de decisiones erróneas. Entre los riesgos también se debe considerar que la tecnología evoluciona rápidamente, esto puede provocar que el sistema se vuelva rápidamente obsoleto y requiera de actualizaciones frecuentes. Otro riesgo importante son las amenazas a la seguridad de los datos, la información que se encuentra almacenada en el sistema podría resultar susceptible a ataques informáticos. Además de esto, también se debe considerar la posibilidad de que exista resistencia por parte del personal, lo que puede ser un obstáculo para el aprendizaje de su manejo y dificultar su adopción en el Centro.

Beneficios de la solución: Esta solución puede permitir conocer de manera óptima y eficaz los recursos con los que cuenta el Centro. Así, se puede evitar tanto la escasez como el exceso, lo que reduce costos. Otro beneficio es contar con información en tiempo real, al tener accesible rápidamente información actualizada acerca de los recursos en inventario, es posible tomar

decisiones de manera informada y estratégica. Además de esto, esta solución puede ayudar a mejorar la eficiencia operativa al permitir acceder rápidamente a la información que se desea conocer de los recursos.

2.1.4 *Análisis de los modelos de IA*

YOLO: es un modelo de detección de objetos en imágenes que tiene gran popularidad en el campo de la visión por computador, entre sus principales ventajas están la eficiencia en el tiempo de detección, las predicciones precisas y la versatilidad para diversas aplicaciones, entre sus desventajas está el tener problemas al detectar objetos pequeños, puede tener problemas al clasificar si en el entrenamiento hay más imágenes de un objeto que de otro, se necesita que el conjunto de datos sea representativo y, además, el entrenamiento puede requerir altos recursos computacionales.

Dectectron 2: Entre las características que presenta, están la amplia variedad de modelos preentrenados, el soporte para múltiples tareas (no solamente la detección de objetos), además de su eficiencia y rendimiento. Entre sus ventajas está tener un buen soporte a través de una comunidad activa y su sencillez de uso, su principal desventaja es que puede llegar a requerir altos recursos computacionales.

Retinanet: Este modelo destaca por la capacidad que posee para la detección de objetos pequeños en imágenes, sus principales ventajas son la alta precisión en las detecciones y ser ampliamente utilizado en aplicaciones prácticas, entre sus desventajas están tener una velocidad de respuesta más tardía en comparación con modelos más ligeros, además de ser altamente sensible a la variación en la elección de los hiperparámetros.

2.1.5 Dataset (modelo de Deep Learning)

Para este proyecto se utilizará un dataset obtenido de internet, las características son en la medida de lo posible similares a lo que se encontraría en las bodegas que almacenan los recursos en CIDIS. El Dataset cuenta con un total de 10011 imágenes de recursos, puede ser obtenido desde el siguiente enlace: [Dataset](#)

Cada uno de los recursos presentes en las imágenes del Dataset está etiquetado con un identificador correspondiente al tipo de recurso al que pertenece, de forma que en el proceso de entrenamiento el modelo asocie la etiqueta y la imagen del recurso detectado.

Las imágenes están divididas en tres grupos, uno para cada fase de desarrollo del modelo, 6980 imágenes para entrenamiento (69.72%), 2026 imágenes para validación (20.24%) y 1005 imágenes para testeo (10.04%). Estas imágenes contienen recursos de 7 clases (tipos de objetos a detectar), CPU, GPU, disco duro HDD, tarjeta madre, fuente de alimentación (PSU), memoria RAM, disco duro SSD.

2.2 Historias de usuario

Las historias de usuario permiten describir las características y requisitos de un software desde el punto de vista del usuario final o cliente [20]. El objetivo de estas historias es ayudar a entender el valor que aporta cada función al cliente y cómo debe funcionar el software [20]. Para su correspondiente realización, primero se realizaron reuniones con el cliente, para así definir los correspondientes requerimientos y objetivos que se desea lograr durante el desarrollo del software. Después se procedió a redactar las historias de forma concisa, clara, sin mencionar soluciones específicas o funcionalidades complejas; y usando un lenguaje no técnico que sea comprensible para todos los involucrados. Las historias de usuario están acompañadas con criterios de

aceptación, que son condiciones específicas y medibles que deben cumplirse para que la historia se considere terminada y satisfactoria [20].

- **Identificador (ID) de la historia:** Clave que identifica de manera única a la historia de usuario.
- **Rol:** Describe el rol de usuario al que corresponde la funcionalidad que describe en la historia.
- **Característica/funcionalidad:** Es la función que el rol necesita realizar en el sistema.
- **Razón/resultado:** Describe lo que el rol desea lograr al realizar la acción. El resultado es desde el punto de vista del rol.
- **Número (#) de escenario:** Número con el que identifica a cada posible escenario que está involucrado en la historia.
- **Criterio de aceptación:** Es el contexto asociado al escenario, en el que se define un comportamiento.
- **Contexto:** Brinda una mayor descripción de las condiciones que desencadenan el escenario.
- **Evento:** Corresponde a la acción que es ejecutada por el usuario, esta acción está bajo el contexto que se define para el escenario.
- **Resultado/comportamiento esperado:** Es el comportamiento que el sistema debe tener luego de haberse ejecutado la acción bajo el contexto definido.

La *Tabla 2.1* muestra cómo luce una historia de usuario de las que se desarrollaron para este proyecto, cada una de ellas cuenta con las siguientes partes:

Identificador (ID) de la historia	Enunciado de la historia			Criterios de aceptación				
	Rol	Característica / Funcionalidad	Razón / resultado	Número (#) de escenario	Criterio de aceptación (Título)	Contexto	Evento	Resultado / Comportamiento esperado
AD001	Como un administrador.	Necesito ver la lista de todos los usuarios registrados en el sistema.	Con la finalidad de conocer quiénes son los usuarios que pueden hacer uso del sistema	1	La lista contiene a los usuarios de tipo Administrador.	En caso de que existan usuarios administradores.	Cuando se ingrese a la lista de usuarios.	Se mostrará la información asociada al usuario, junto con su categoría de Administrador.
				2	La lista contiene a los usuarios de tipo Custodio.	En caso de que existan usuarios custodios.	Cuando se ingrese a la lista de usuarios.	Se mostrará la información asociada al usuario, junto con su categoría de Custodio.
				3	La lista contiene a los usuarios de tipo Usuario General.	En caso de que existan usuarios generales.	Cuando se ingrese a la lista de usuarios.	Se mostrará la información asociada al usuario, junto con su categoría de Usuario General.

Tabla 2.1: *Historia de usuario, para usuario con rol administrador*

La lista completa de historias de usuario para este proyecto se encuentra en el [Apéndice B](#).

2.3 Diseño de la solución (Sistema de inventario)

Utilizaremos el enfoque del Modelo de Vistas de Arquitectura 4+1 de Kruchten [21] para explicar la arquitectura de nuestra solución. Este modelo define cinco perspectivas arquitectónicas que se basan en el uso simultáneo de múltiples vistas. Estas vistas son: Escenarios, Vista Lógica, Vista de Implementación, Vista de Procesos y Vista de Despliegue.

2.3.1 Escenarios

La Figura 2.1 muestra a los diferentes tipos de usuario (Administrador, Custodio, Usuario General), con las correspondientes acciones que pueden realizar en el sistema.

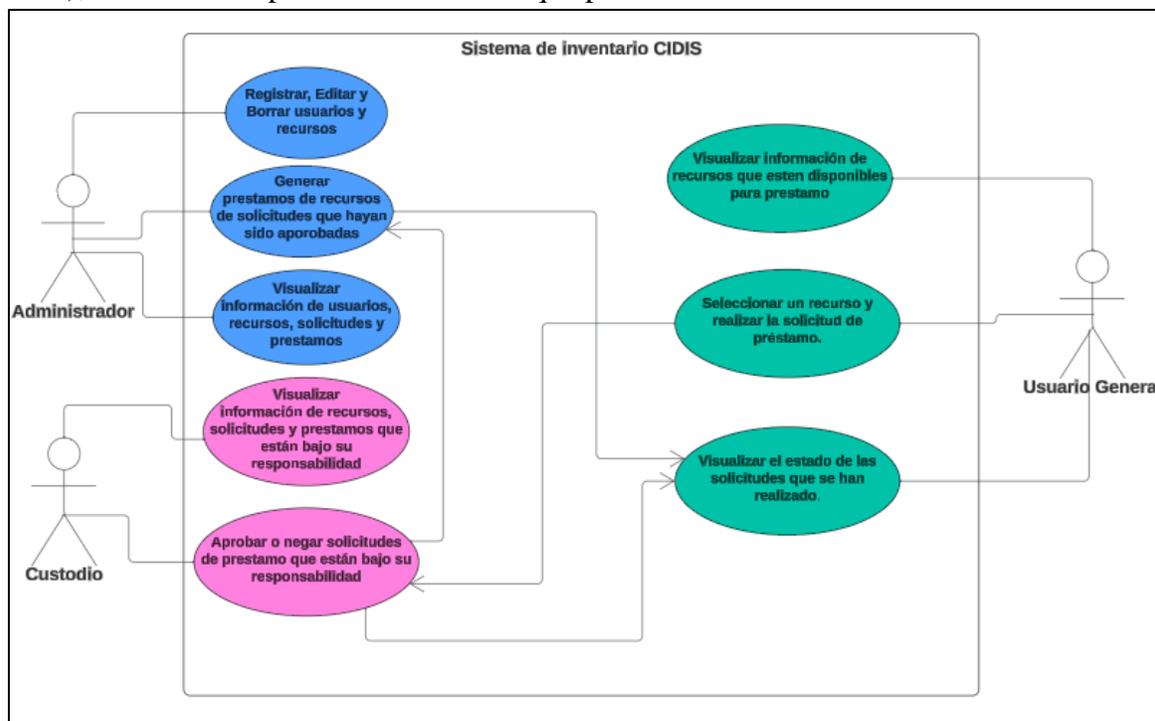


Figura 2.1: Diagrama de casos de uso del sistema de inventario.

2.3.2 Vista Lógica

Como parte de la vista lógica, se tiene la relación entre tablas, según el diseño de la base de datos del sistema.

Por motivos ligados a los requerimientos del proyecto, el diseño de la base de datos se divide en dos secciones, en cada sección las relaciones entre tablas se realizan solo entre tablas de la misma sección. Las secciones nombradas anteriormente corresponden a las siguientes:

- **Sección de usuarios y roles:** Registra la información correspondiente a los usuarios válidos del sistema y sus respectivos roles de usuario.

- **Sección de registros de préstamos y solicitudes:** Almacena la información acerca de los préstamos y solicitudes que se realizan en el sistema.

La *Figura 2.2* muestra las relaciones entre las tablas usuario, rol, recurso, categoria_recurso y estado_recurso. La tabla “rol” almacenará los distintos roles de usuario que el sistema soportará, mientras que en la tabla “usuario” se guardarán los registros de los distintos usuarios válidos del sistema, es posible ver la relación entre ambas tablas mediante la clave foránea “rol” en la tabla “usuario”.

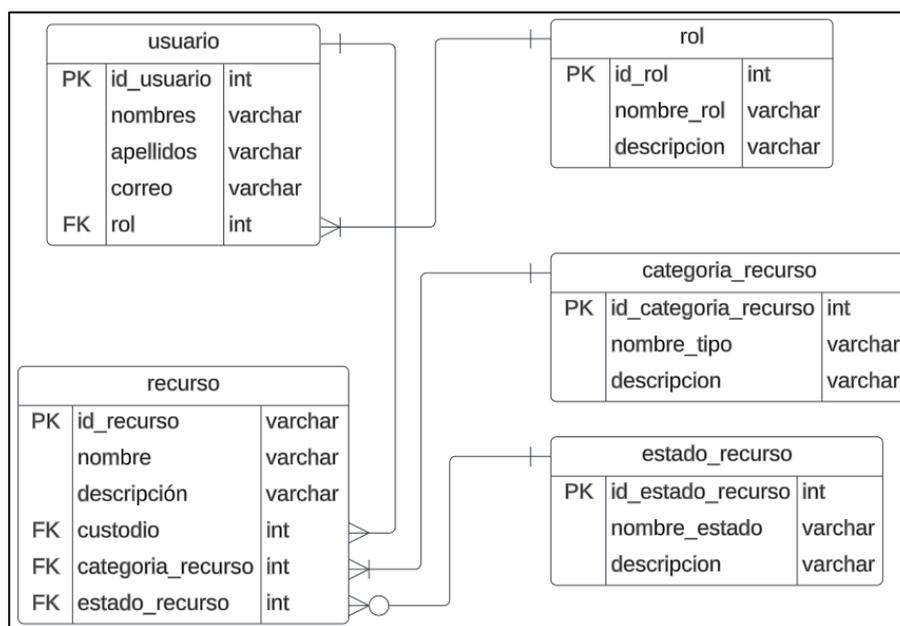


Figura 2.2: Tablas de la sección de usuarios y roles.

La tabla “recurso” almacenará la información más importante para las actividades correspondientes a este sistema, la que concierne a los bienes que posee el Centro. Es posible notar su relación con las tablas usuario, categoria_recurso y estado_recurso, esto debido a que cada recurso posee un custodio (relacionado con la tabla usuario), el cual es un usuario del sistema, además cada recurso pertenece a alguna categoría que el Centro designe (relacionado con la tabla

categoria_recurso), mientras que su estado dependerá de si el bien está en préstamo o disponible en las instalaciones de CIDIS (relación con la tabla estado_recurso).

En la *Figura 2.3* se muestran las tablas que almacenarán la información perteneciente a las solicitudes y préstamos de bienes que se hayan realizado en el sistema. Como se puede apreciar, la tabla “solicitud_o_prestamo” es la que almacena la mayor cantidad de atributos debido a su importancia en el sistema, registros tales como los nombres del custodio, nombres del solicitante, fecha de inicio del período de préstamo, responsable del proyecto en donde se hará uso del recurso, forman parte de la información valiosa y necesaria que se desea registrar de cada solicitud y préstamo que se origine en el sistema.

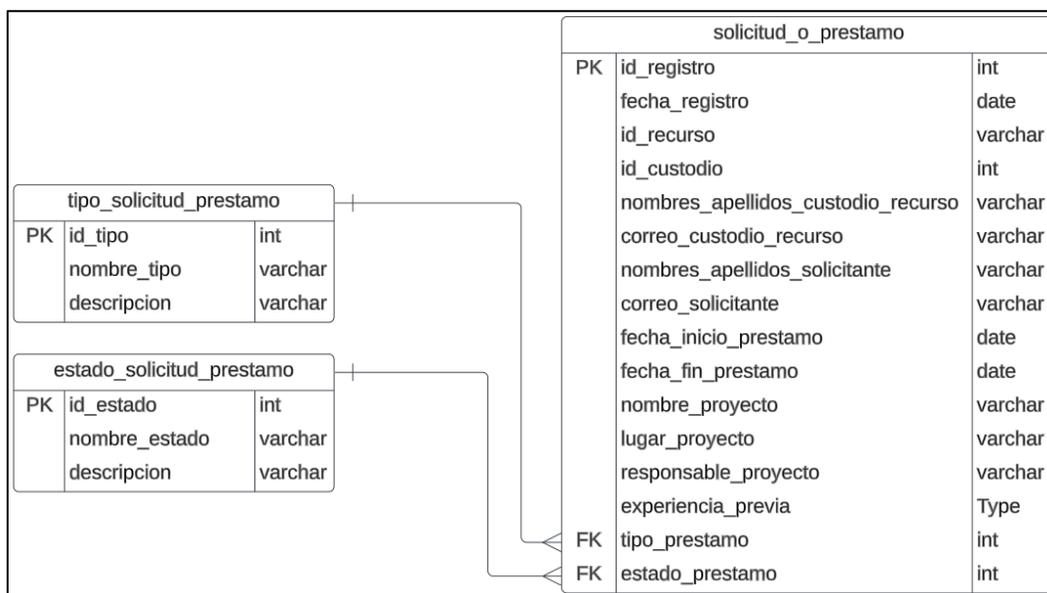


Figura 2.3: Tablas de la sección de registros de préstamos y solicitudes.

Dicha información permitirá a administradores, custodios y usuarios generales, gestionar la información correspondiente a su rol.

2.3.3 Vista de Desarrollo o Implementación

Esta solución de software hará uso de la arquitectura MVC (modelo-vista-controlador) y será implementada mediante una aplicación web compuesta de Frontend y Backend. Se utilizará el framework Angular para el Frontend y el framework Express para el Backend.

La *Figura 2.4* muestra de manera general los módulos que componen el sistema, la descripción de cada módulo se detalla a continuación:

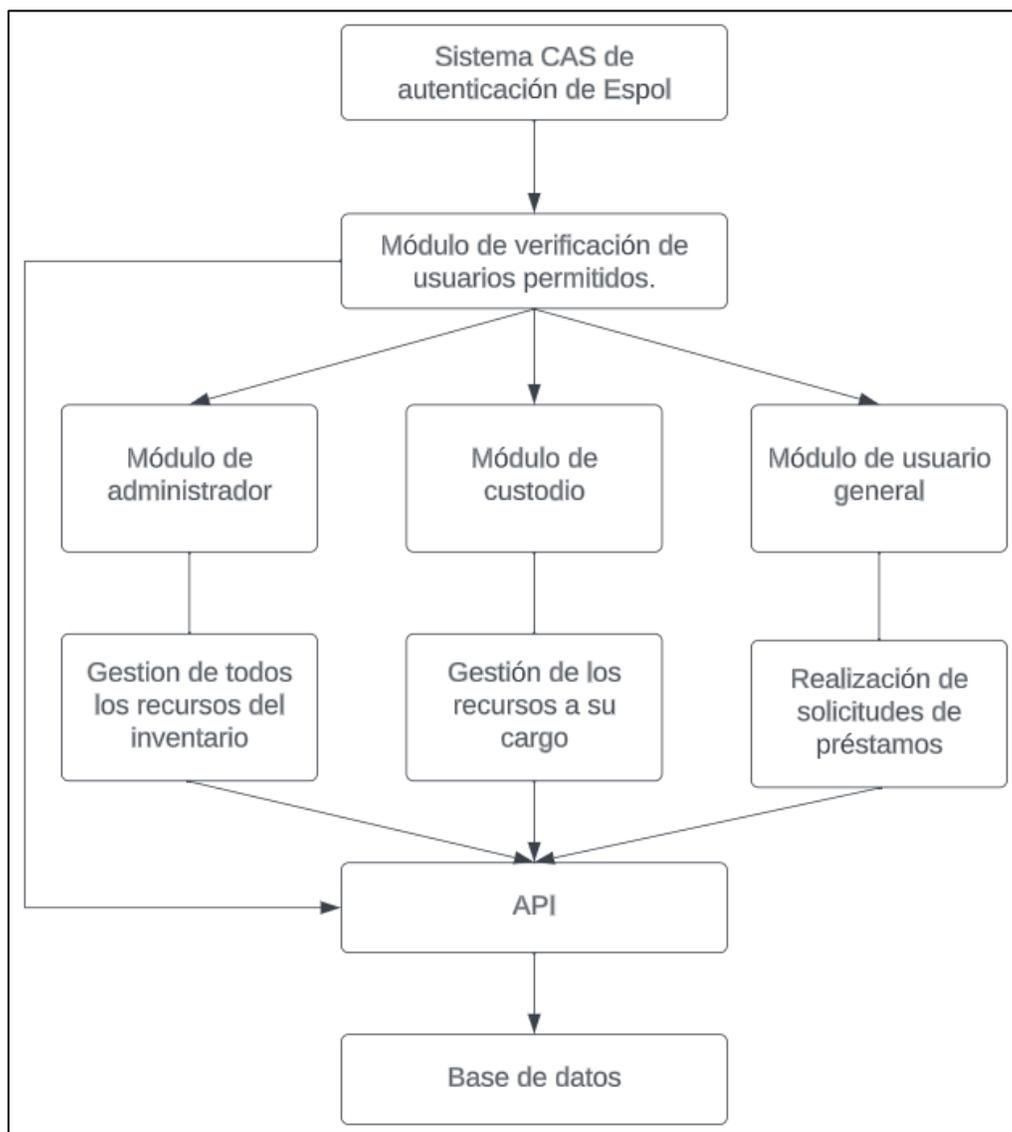


Figura 2.4: Diagrama de bloques general del sistema de inventario.

- **Sistema CAS de autenticación de Espol:** El sistema de autenticación con el que cuenta la universidad, se implementará en la aplicación para validar que los usuarios que inicien sesión pertenezcan a Espol.
- **Módulo de verificación de usuarios permitidos:** Se requiere que los usuarios pertenezcan a Espol, sin embargo, esto es una condición necesaria, más no suficiente. Este módulo se encarga de conectarse a la API para validar que el usuario de Espol que ingresa esté autorizado para hacer uso del sistema. Si el usuario no está autorizado, se le impedirá el acceso al sistema; si es un usuario autorizado, el sistema determinará el rol que posee y lo redirigirá al módulo correspondiente (administrador, custodio o, usuario general).
- **Módulo de administrador:** Aquellos usuarios que posean el rol de administrador harán uso de este módulo para realizar la gestión de todos los recursos de los que dispone el Centro y de los usuarios que tienen permitido hacer uso del sistema. Este módulo se conectará a la API para realizar el registro, actualización y seguimiento del estado e información de todos los recursos del centro, lo mismo para la información de los usuarios.
- **Módulo de custodio:** Similar al módulo de administrador y, de igual forma, se conectará a la API, con la diferencia de que los custodios no realizarán registros ni actualizaciones a la lista de recursos, ni tampoco tendrán acceso a la información de los usuarios del sistema y solamente podrán realizar el seguimiento del estado e información de los recursos que estén bajo su custodia.
- **Módulo de usuario general:** Este es el módulo que mediante conexión con la API permitirá que los usuarios con este rol puedan realizar las solicitudes de préstamo de equipos y realizar el seguimiento de las solicitudes que han realizado.

- **API:** Aquí reside la funcionalidad del Backend de la aplicación. Se encargará de determinar los usuarios autorizados, realizar las peticiones a la base de datos y permitir que los demás módulos realicen las operaciones que les corresponden a su respectivo rol.
- **Base de datos:** Contendrá la información de la que hará uso la aplicación; es decir, información correspondiente a los usuarios autorizados, la información relacionada a cada recurso, los datos de las solicitudes y préstamos que se realicen y demás información que se necesite almacenar en el sistema.

En la *Figura 2.5* se muestra el diagrama de componentes del proyecto. Como se puede apreciar, el sistema cuenta con 4 componentes principales, uno de verificación de permisos de usuario (Login) y tres que corresponden a cada rol de usuario (Administrador, Custodio y Usuario general). Cada uno de estos cuatro componentes cuenta con un componente de Servicios que le proporciona la conexión con la API para que sea posible la realización de sus respectivas funcionalidades.

El componente Login es el encargado de determinar mediante conexión con la API, el permiso y nivel de acceso de cada usuario que intente iniciar sesión en el sistema. Posteriormente, si no es un usuario válido, este componente negará el acceso. Por otra parte, en caso de ser un usuario válido para el sistema, este componente los redirigirá hacia el componente correspondiente de acuerdo con sus permisos de usuario (Administrador, Custodio, Usuario General).

Cada uno de los componentes de rol de usuario están conformados por sus respectivos subcomponentes, que contienen las funciones para la realización de las actividades correspondientes a su rol.

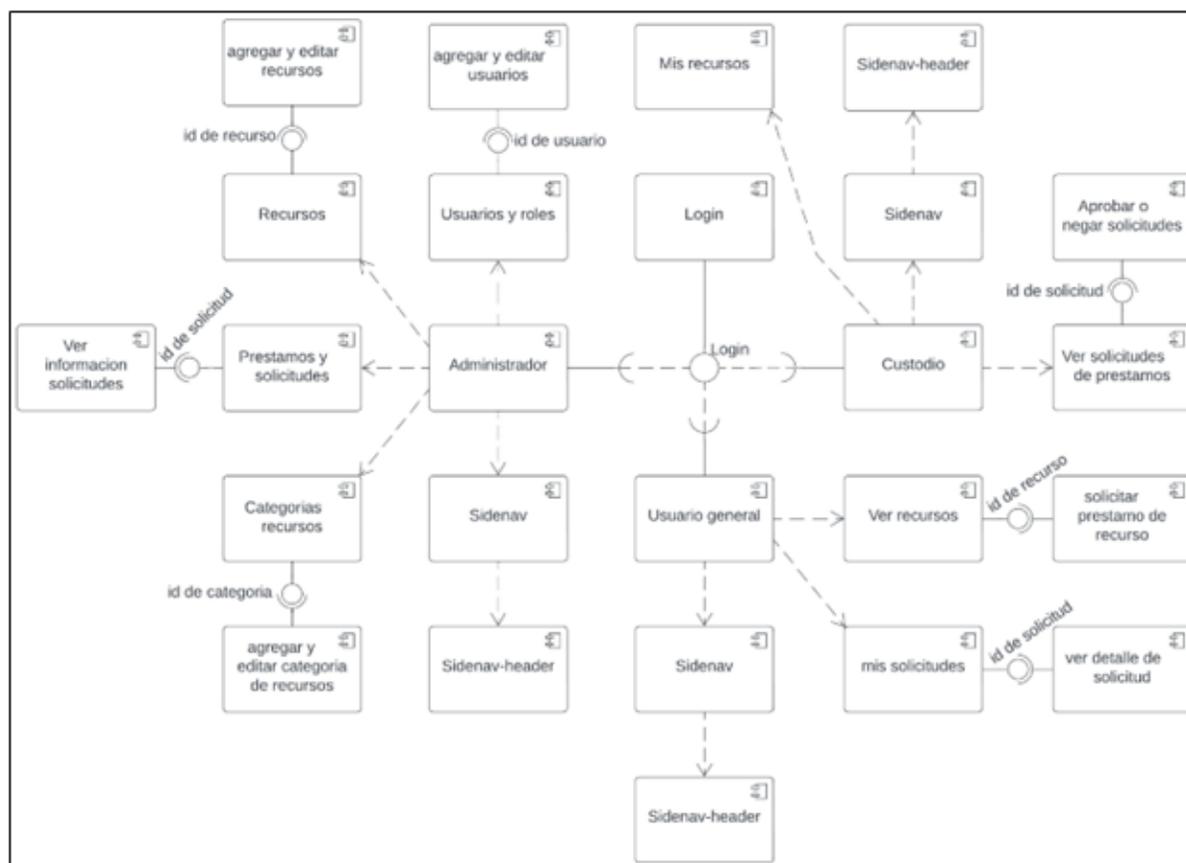


Figura 2.5: Diagrama de componentes del sistema de inventario.

2.3.4 Vista de Procesos

Dado que nuestro proyecto es una aplicación web, contamos con un servidor web que gestiona las interacciones simultáneas de los usuarios y un administrador de base de datos encargado de manejar la concurrencia en la base de datos. Por lo tanto, no hace falta especificar esta vista.

2.3.5 Vista de Despliegue

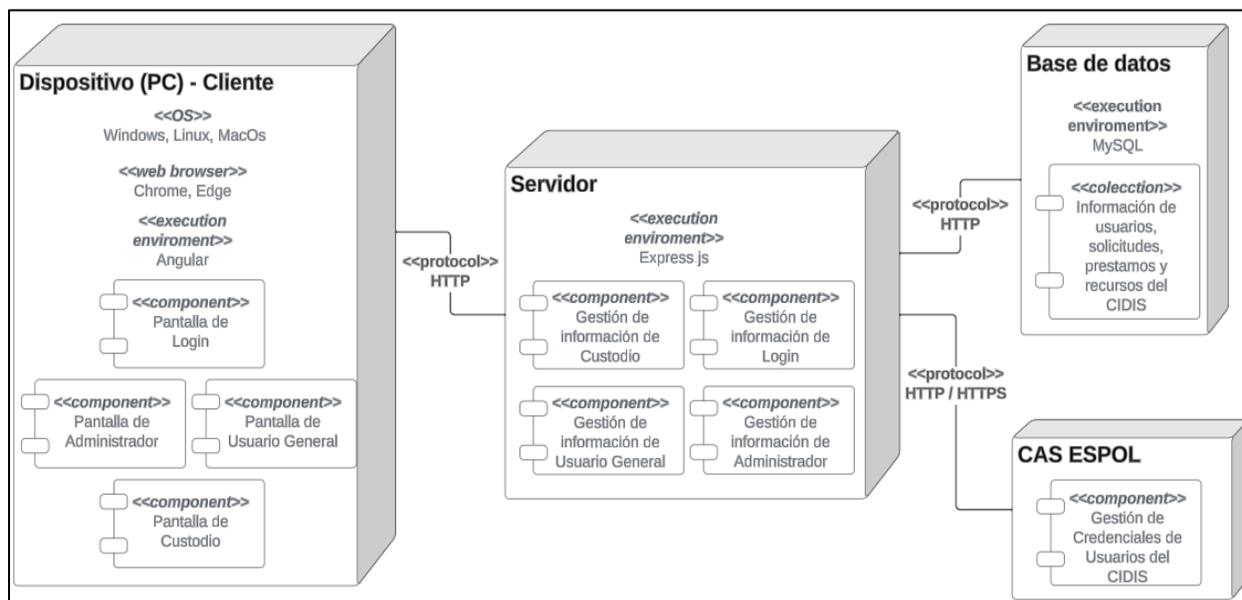


Figura 2.6: Diagrama de despliegue del sistema de inventario.

El diagrama mostrado en la *Figura 2.6* describe los elementos que son necesarios para ejecutar la aplicación. Tal como se muestra en la misma, la ejecución correcta del sistema depende de elementos tanto de hardware como de software. Es necesario contar con un dispositivo (PC) ejecutando un sistema operativo (Windows, Mac, Linux) en modo de interfaz de usuario para poder ejecutar el Cliente, el cual contiene el Frontend que está desarrollado en el framework Angular, luego es necesaria una conexión con el CAS de ESPOL para poder verificar que los usuarios que inicien sesión sean usuarios registrados en ESPOL, posteriormente está la conexión con el servidor cuya lógica está implementada usando el framework Express y permitirá conectarse a la base de datos relacional para poder validar los usuarios permitidos del sistema y que puedan realizar sus respectivas funciones de acuerdo con sus roles.

El sistema está compuesto de Frontend y Backend, para cada uno se utilizaron las siguientes tecnologías:

Frontend

- Angular CLI, versión 16.2.7
- NodeJs, versión 18.18.0
- Node Package Manager (npm), versión 10.2.1

Backend

- Express, versión 4.18.2
- Typescript, versión 5.2.2
- MySQL Community Server, versión 8.1.0

Como recursos de hardware y software se utilizó lo siguiente:

- PC con procesador Intel 10ma generación y 16 GB de memoria RAM.
- Windows 10.
- MySQL Workbench CE, versión 8.0.34.

2.4 Diseño de la solución (Modelo de Deep Learning)

2.4.1 Esquema general de pasos

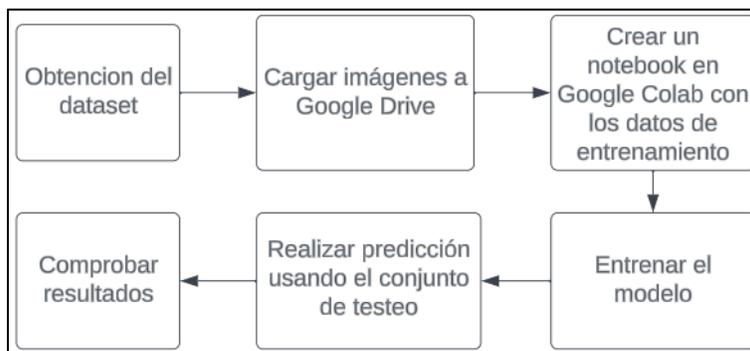


Figura 2.7: Pasos generales del desarrollo del modelo de IA

La *Figura 2.7* muestra los pasos que, a modo general se utilizarán para desarrollar el modelo de detección.

El detalle de cada paso se describe a continuación:

- Obtención del Dataset: Buscar un Dataset con características similares a las que se necesitan y descargarlo desde su fuente de origen.
- Cargar imágenes a Google Drive: Crear una cuenta de Google para el proyecto y una carpeta en Google Drive para cargar el Dataset que se obtuvo.
- Notebook en Google Colab: Crear un archivo con formato Jupyter Notebook (.ipynb) en Google Colab, este archivo contendrá las instrucciones en Python para realizar el entrenamiento accediendo al Dataset que previamente se cargó en Google Drive.
- Entrenar modelo: Consiste en ejecutar las instrucciones descritas en el archivo de Colab y esperar hasta que el entrenamiento finalice.
- Realizar predicción: Una vez que el entrenamiento esté culminado, se procederá a tomar las imágenes del conjunto de testeo y dejar que el modelo entrenado las analice para que reconozca lo que contienen.
- Comprobar resultados: Realizar varias detecciones con el modelo y comprobar que lo que el modelo detecta realmente sea lo que está en las imágenes.

2.4.2 Diagramas de casos de uso

En *Figura 2.8* se visualiza el escenario que ocurre actualmente en las bodegas de CIDIS, mientras que en la *Figura 2.9* se visualiza cómo interviene el modelo de IA en la mejora del control de inventario.

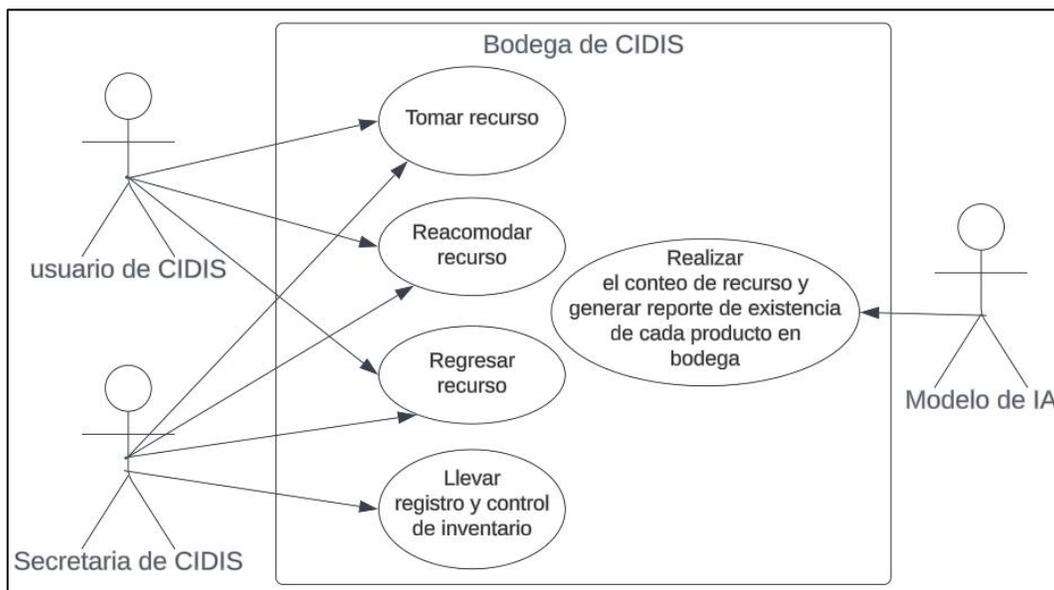


Figura 2.8: Diagrama de casos de uso (situación actual de CIDIS).

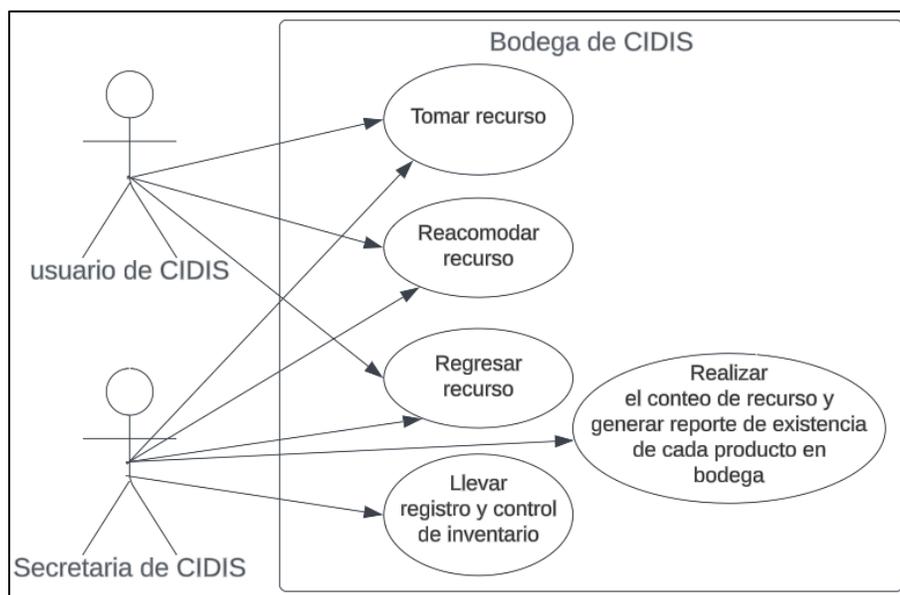


Figura 2.9: Diagrama casos de uso (Solución con modelo de IA).

2.4.3 Entrenamiento

Para el entrenamiento se utilizará el modelo YOLOv8x. Este es un modelo de aprendizaje profundo que ha mostrado gran eficacia en la detección de objetos, tanto en imágenes como en vídeo, al ser el modelo más robusto de YOLOv8 aporta una mejor precisión al momento de reconocer objetos en imágenes [18].

El entorno de entrenamiento estará conformado por el Dataset dividido en los tres grupos (entrenamiento, validación y testeo), código en Python para hacer uso del modelo y entrenarlo, y la plataforma gratuita Google Colab, esta plataforma brinda durante un tiempo limitado acceso gratuito a recursos de hardware como tarjeta gráfica para realizar el entrenamiento del modelo, en caso de que se necesite más tiempo o más potencia computacional de la que se obtiene en Colab, se hará uso de una máquina física provista por CIDIS.

2.5 Plan de Implementación

De acuerdo con los objetivos del proyecto y por la duración de la materia integradora, se estableció un plan de implementación. Para la creación de este, se dividió la implementación del sistema en varias fases con respecto a las historias de usuario. Dicho plan se detalla en la *Tabla 2.2* mostrada a continuación:

Título de Actividad	Fecha de Inicio	Fecha de Fin
Implementación de las HU: - Listar todos los usuarios como usuario administrador. -Agregar, Editar y eliminar usuarios del sistema como usuario administrador.	6/11/2023	8/11/2023
Implementación de las HU: - Visualizar la lista de recursos y su respectiva información como usuario administrador.	9/11/23	10/11/23

Agregar, editar y eliminar recurso del registro como usuario administrador.		
Implementación de las HU: - Visualizar la lista de recursos que están bajo custodia y su estado como usuario custodio. - Visualizar la lista de recursos que tienen disponibilidad de solicitar en préstamo como usuario general	13/11/23	17/11/23
Implementación de las HU: - Seleccionar algún recurso y realizar la solicitud de préstamo como usuario general. Visualizar el estado de las solicitudes que se han realizado como usuario general.	20/11/23	24/11/23
Implementación de las HU: - Visualizar las solicitudes de préstamo que corresponde revisar como usuario custodio. Aprobar o negar las solicitudes de préstamo que corresponden como usuario custodio.	27/11/23	1/12/23
Implementación de las HU: - Visualizar la información de las solicitudes de préstamo y el estado que poseen como usuario administrador.	4/12/23	8/12/23
Implementación de las HU: -Generar préstamos a partir de solicitudes aprobadas como usuario administrador.	11/12/23	15/12/23
Implementación de las HU: - Visualizar lista de recursos en préstamo y el detalle de información del préstamo como usuario administrador.	18/12/23	20/12/23
Implementación de las HU: -Visualizar la información acerca de las solicitudes que se aprueban y que ya generaron un préstamo como usuario custodio.	21/12/23	22/12/23

Pruebas con el cliente	2/1/24	5/1/24
Modificaciones	8/1/24	12/1/24
Pruebas con el cliente final	15/1/24	19/1/24

Tabla 2.2: *Plan de implementación de las historias de usuario.*

3. Capítulo 3

3.1 Desarrollo

Se pudo comprobar que fue posible utilizar Angular y NodeJs de manera satisfactoria para cumplir con los requerimientos de este proyecto, aunque la curva de aprendizaje de Angular puede resultar algo complicada en determinados escenarios, fue posible hacer un correcto aprendizaje de lo necesario para este proyecto. Los módulos de Angular Material fueron una potente herramienta para el diseño de cada una de las ventanas y funcionalidades en el Frontend.

Por otro lado, NodeJs mediante el lenguaje Typescript permitió una sencilla conexión con el motor de base de datos MySQL, además de una eficiente implementación de cada una de las rutas y peticiones (GET, POST, DELETE, UPDATE) para cada uno de los roles de usuario.

Aunque fue posible solventarla, lo que presentó una mayor complejidad fue la conexión desde el Backend con el sistema CAS de Espol, se requirieron de varias pruebas hasta encontrar la configuración correcta de las librerías que se utilizaron.

El desarrollo de este sistema de mediana complejidad requirió de alrededor de tres meses, entre la codificación de las vistas y funcionalidades, y las pruebas con el cliente final. Se designaron tareas para cada miembro del equipo, tal como puede observarse en la *Figura 3.1*.

Nombre	Persona asignada	Fecha límite	Prioridad
✓ Recolección de requerimientos	JA DV	10/20/23	Urgente
✓ Estructura base del FrontEnd	JA	10/24/23	Alta
✓ Estructura base del Backend	JA	10/24/23	Alta
✓ Rutas usuario general (Backend)	DV	10/27/23	
✓ Rutas usuario general (Frontend)	JA	10/27/23	Normal
✓ Rutas custodio (Backend)	DV	10/27/23	Normal
✓ Rutas custodio (Frontend)	JA	10/27/23	Normal
✓ Rutas administrador (Frontend)	JA	10/27/23	Normal
✓ Rutas administrador (Backend)	DV	10/27/23	Normal
✓ Rutas API usuarios y roles (Administrador)	DV	11/3/23	Normal
✓ Vista usuario y roles (Administrador)	JA	11/3/23	Normal

Figura 3.1: Tareas asignadas a cada integrante del equipo de desarrollo.

Para el correcto seguimiento del avance del proyecto y la verificación del cumplimiento de cada una de las tareas, se utilizó la plataforma de manejo de versiones GitHub (ver Figuras 3.2 y 3.3).

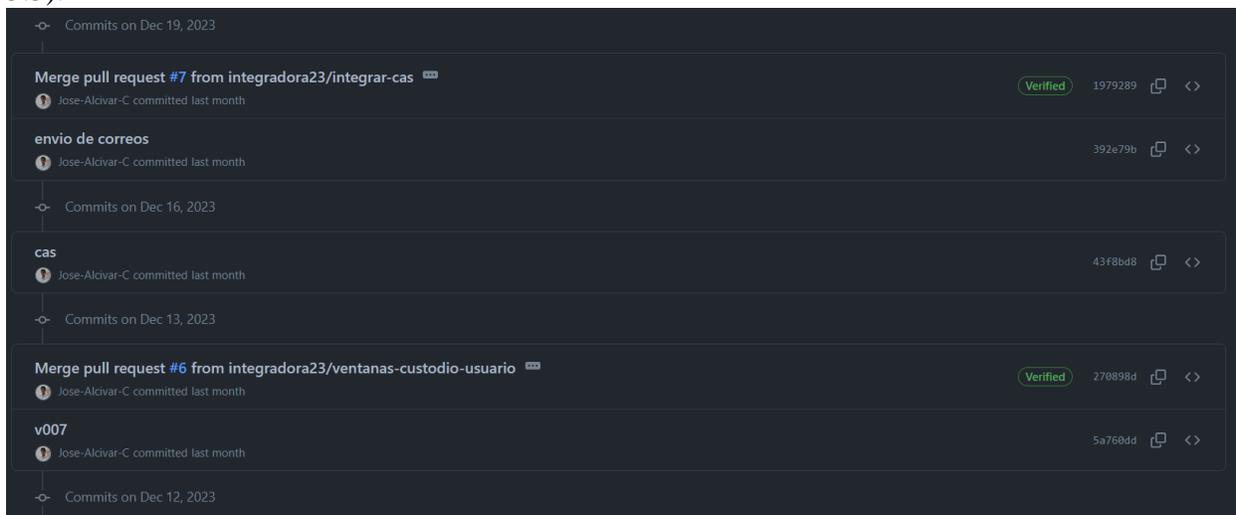


Figura 3.2: Commits en el Frontend.

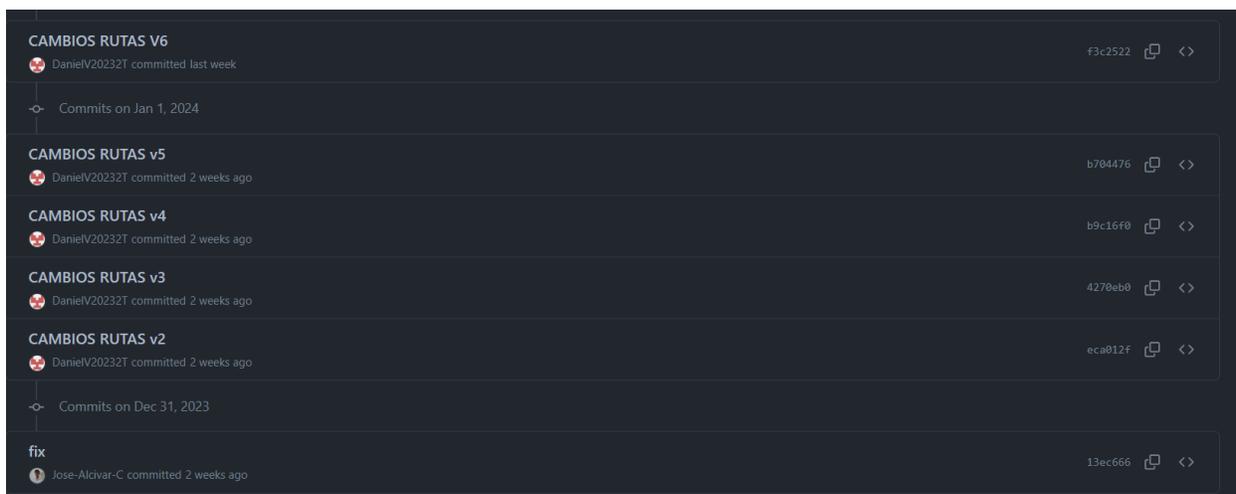


Figura 3.3: Commits en el Backend.

El resultado del desarrollo fue un sistema compuesto de tres módulos principales, cada uno con las funcionalidades y vistas correspondientes a su rol:

- Módulo de administrador.
- Módulo de custodio.
- Módulo de usuario general.

3.2 Resultados y análisis (Sistema de inventario)

A continuación, se presentan las vistas creadas como resultado de una constante retroalimentación a lo largo del proyecto.

3.2.1 Vista de Login

El sistema cuenta con una ventana principal de Login, dicha ventana cumple la función de permitir que un usuario sea redireccionado al sistema CAS de ESPOL para que realice el ingreso de sus correspondientes credenciales de usuario. La *Figura 3.4* muestra la venta de Login.



Figura 3.4: Ventana de Login del sistema.

Al hacer clic en “Iniciar sesión” se redirigirá al sistema CAS de Espol para el respectivo ingreso de credenciales de usuario. Tal como se muestra en la *Figura 3.5*.

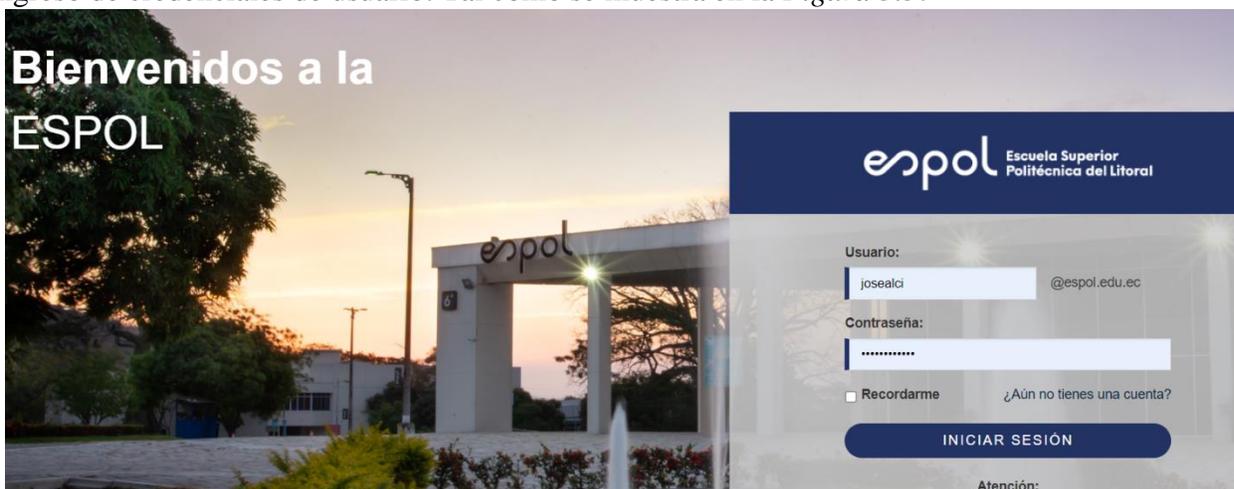


Figura 3.5: CAS de Espol incorporado en el sistema.

En este sistema, es necesario ser un usuario registrado en Espol, sin embargo, no es suficiente, se requiere además ser un usuario registrado (permitido) en el sistema. En la *Figura 3.6* se puede apreciar la ventana que mostrará a un usuario de Espol que NO está registrado en el sistema.



Figura 3.6: Ventana para un usuario no registrado en el sistema.

3.2.2 Vista del administrador

Las vistas y funciones de un administrador le dan un control absoluto sobre los usuarios y sus roles, los recursos y sus estados, y los préstamos que los usuarios realizan mediante el sistema.

The image shows the administrator view of the CIDIS system. The page title is "Sistema de Inventario CIDIS" and the main heading is "Recursos". The page shows a table of resources with columns for ID, Name, Description, Custodian, Category, Status, Location, and Actions. The user "SUAREZ HERRERA STEFANO ISRAEL" is logged in as Administrator. The table contains 8 rows of resource data.

ID del recurso	Nombre	Descripción	Custodio	Categoría de recurso	Estado del recurso	Ubicación	Opciones recurso	Acciones
ID02401	recurso ID02401	descripcion recurso ID02401	Montoya Ortiz Nahim Josue	Categoría 01	No Prestable	Oficina 1		
ID02402	recurso ID02402	descripcion recurso ID02402	Montoya Ortiz Nahim Josue	Categoría 01	No Prestable	Oficina 1		
ID02403	recurso ID02403	descripcion recurso ID02403	Montoya Ortiz Nahim Josue	Categoría 01	No Prestable	Oficina 1		
ID02404	recurso ID02404	descripcion recurso ID02404	Montoya Ortiz Nahim Josue	Categoría 01	No Prestable	Oficina 1		
ID02405	recurso ID02405	descripcion recurso ID02405	Montoya Ortiz Nahim Josue	Categoría 01	No Prestable	Oficina 1		
ID02406	recurso ID02406	descripcion recurso ID02406	Montoya Ortiz Nahim Josue	Categoría 01	No Prestable	Oficina 1		
ID02407	recurso ID02407	descripcion recurso ID02407	Montoya Ortiz Nahim Josue	Categoría 01	No Prestable	Oficina 1		
ID02408	recurso ID02408	descripcion recurso ID02408	Montoya Ortiz Nahim Josue	Categoría 01	No Prestable	Oficina 1		

Recursos mostrados por página: 8 2401 - 2408 of 4800

Figura 3.7: Vista de un administrador

En la *Figura 3.7* se muestra una de las vistas a las que tiene acceso un usuario de tipo administrador, la vista de todos los recursos y los datos asociados a los mismos, datos como su ID, su descripción, su custodio y demás información que le permiten al administrador controlar cada recurso registrado en el sistema. El sidebar de la izquierda contiene los accesos a las demás vistas y funciones que posee un administrador, la vista de usuarios y roles, la vista de las categorías de recursos y, la vista de los préstamos y solicitudes.

3.2.3 Vista del custodio

Los usuarios de tipo custodio tienen control sobre los recursos que están bajo su custodia. Las ventanas que forman parte de las rutas de los custodios les dan acceso a la información completa acerca de los recursos que custodian, información como su estado y su ubicación. Además, el sistema les permite gestionar las solicitudes de préstamo de recursos que los usuarios realizan. Tal como se muestra en la *Figura 3.8*, un custodio también cuenta con un sidebar a la izquierda, el cual les dará acceso a dos vistas con sus respectivas funcionalidades, mis recursos y, préstamos y solicitudes (para gestionar las solicitudes que le corresponde revisar).

ID del recurso	Nombre	Descripción	Custodio	Categoría de recurso	Estado del recurso	Ubicación	Acciones
ID02401	recurso ID02401	descripcion recurso ID02401	Montoya Ortiz Nahim Josue	Categoría 01	No Prestable	Oficina 1	
ID02402	recurso ID02402	descripcion recurso ID02402	Montoya Ortiz Nahim Josue	Categoría 01	No Prestable	Oficina 1	
ID02403	recurso ID02403	descripcion recurso ID02403	Montoya Ortiz Nahim Josue	Categoría 01	No Prestable	Oficina 1	
ID02404	recurso ID02404	descripcion recurso ID02404	Montoya Ortiz Nahim Josue	Categoría 01	No Prestable	Oficina 1	
ID02405	recurso ID02405	descripcion recurso ID02405	Montoya Ortiz Nahim Josue	Categoría 01	No Prestable	Oficina 1	
ID02406	recurso ID02406	descripcion recurso ID02406	Montoya Ortiz Nahim Josue	Categoría 01	No Prestable	Oficina 1	
ID02407	recurso ID02407	descripcion recurso ID02407	Montoya Ortiz Nahim Josue	Categoría 01	No Prestable	Oficina 1	
ID02408	recurso ID02408	descripcion recurso ID02408	Montoya Ortiz Nahim Josue	Categoría 01	No Prestable	Oficina 1	

Figura 3.8: Vista de un custodio.

3.2.4 Vista del usuario general

La *Figura 3.9* muestra una de las vistas correspondiente a un usuario con rol usuario general, en ella se puede notar que tiene acceso a visualizar la lista de recursos que presentan disponibilidad para ser solicitados en préstamo, en caso de requerir realizar una solicitud, cuenta con el correspondiente botón para que llene los campos correspondientes. Un usuario general también cuenta con un sidebar a la izquierda que le permite acceso a cada una de las vistas y funciones correspondientes a su rol, la vista de recursos disponibles y, la vista de préstamos y solicitudes (para ver el estado de las solicitudes que ha realizado).

ID del recurso	Nombre del recurso	Descripción del recurso	Categoría del recurso	
ID00041	recurso ID00041	descripcion recurso ID00041	categoria 01	Solicitar recurso
ID00042	recurso ID00042	descripcion recurso ID00042	categoria 01	Solicitar recurso
ID00043	recurso ID00043	descripcion recurso ID00043	categoria 01	Solicitar recurso
ID00044	recurso ID00044	descripcion recurso ID00044	categoria 01	Solicitar recurso
ID00045	recurso ID00045	descripcion recurso ID00045	categoria 01	Solicitar recurso
ID00046	recurso ID00046	descripcion recurso ID00046	categoria 01	Solicitar recurso
ID00047	recurso ID00047	descripcion recurso ID00047	categoria 01	Solicitar recurso
ID00048	recurso ID00048	descripcion recurso ID00048	categoria 01	Solicitar recurso

Figura 3.9: Vista de un usuario general

3.3 Resultados y análisis (Módulo de IA)

3.3.1 Entrenamiento

El tipo de detección que se realiza en este proyecto es denominado “segmentación de instancias”, para cumplir con el objetivo de entrenar un modelo básico, el entrenamiento fue realizado mediante los recursos gratuitos de la plataforma Google Colab, utilizando el modelo preentrenado YOLOv8x-seg.

El entrenamiento tuvo las siguientes características:

- **Epochs:** 10
- **Learning rate:** 0.001
- **Optimizer:** SGD
- **Batch size:** 16

Las imágenes usadas en entrenamiento, validación y testeo tienen cada una un tamaño de 640 x 640 píxeles. Al final del entrenamiento se obtuvieron resultados muy buenos, la *Figura 3.10* muestra la matriz de confusión del modelo obtenido.

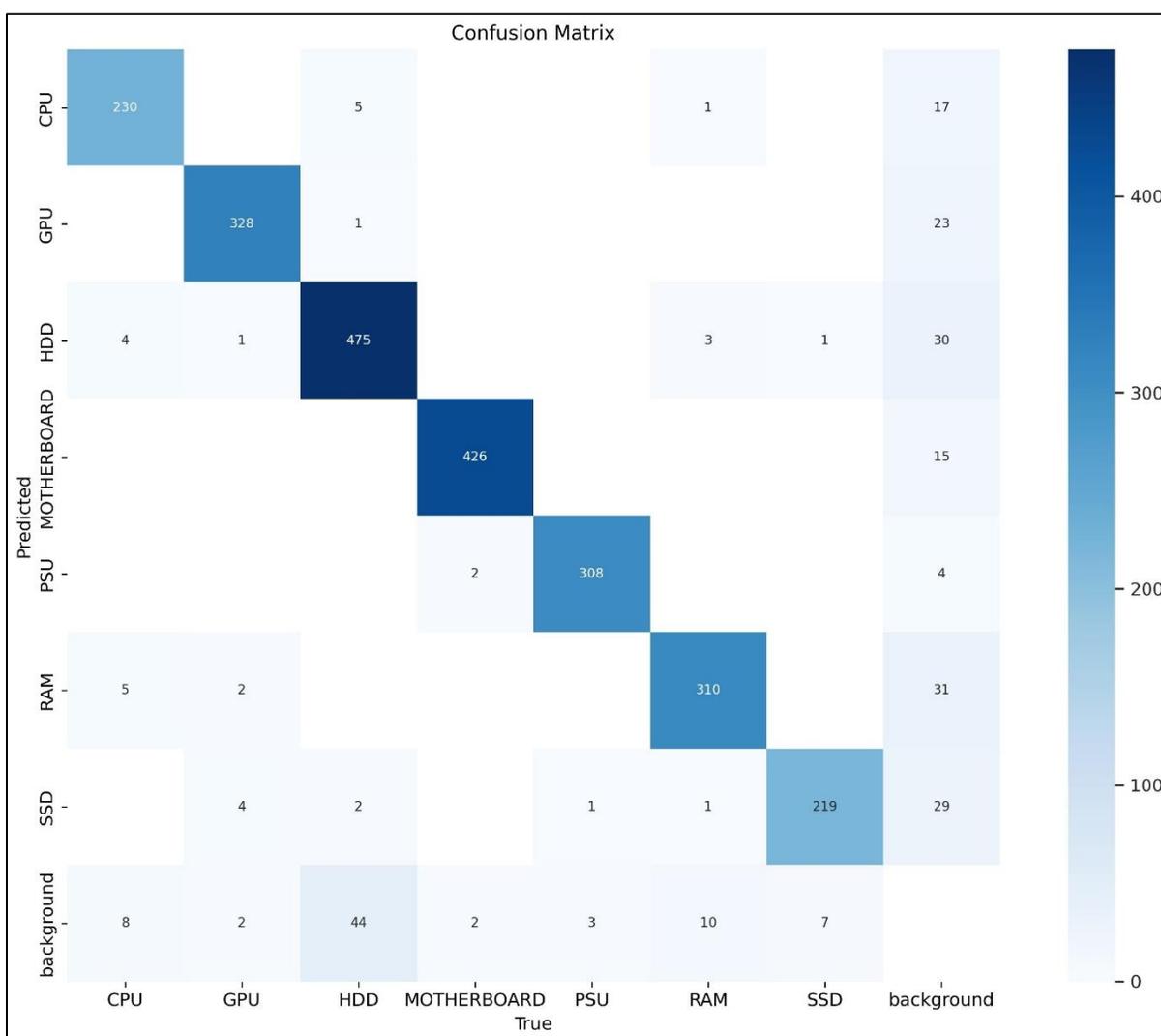


Figura 3.10: Matriz de confusión del modelo de IA.

Lo mostrado en la *Figura 3.10* indica la categoría real a la que pertenece un recurso (eje x) y la categoría que fue detectada por el modelo (eje y). Por ejemplo, en el eje X, para la categoría CPU, 8 recursos fueron detectados como background (detección errónea), 5 fueron detectados como RAM (detección errónea), 4 fueron detectados como HDD (detección errónea) y 230 fueron detectados como lo que realmente son, CPU. La categoría background es una categoría auxiliar usada por el modelo YOLO que indica que un recurso no pudo ser identificado en ninguna de las clases que fueron entrenadas.

La *Figura 3.11* contiene la matriz normalizada del modelo, aquí es posible comprobar una vez más que la gran mayoría de objetos son detectados como la categoría que realmente poseen.

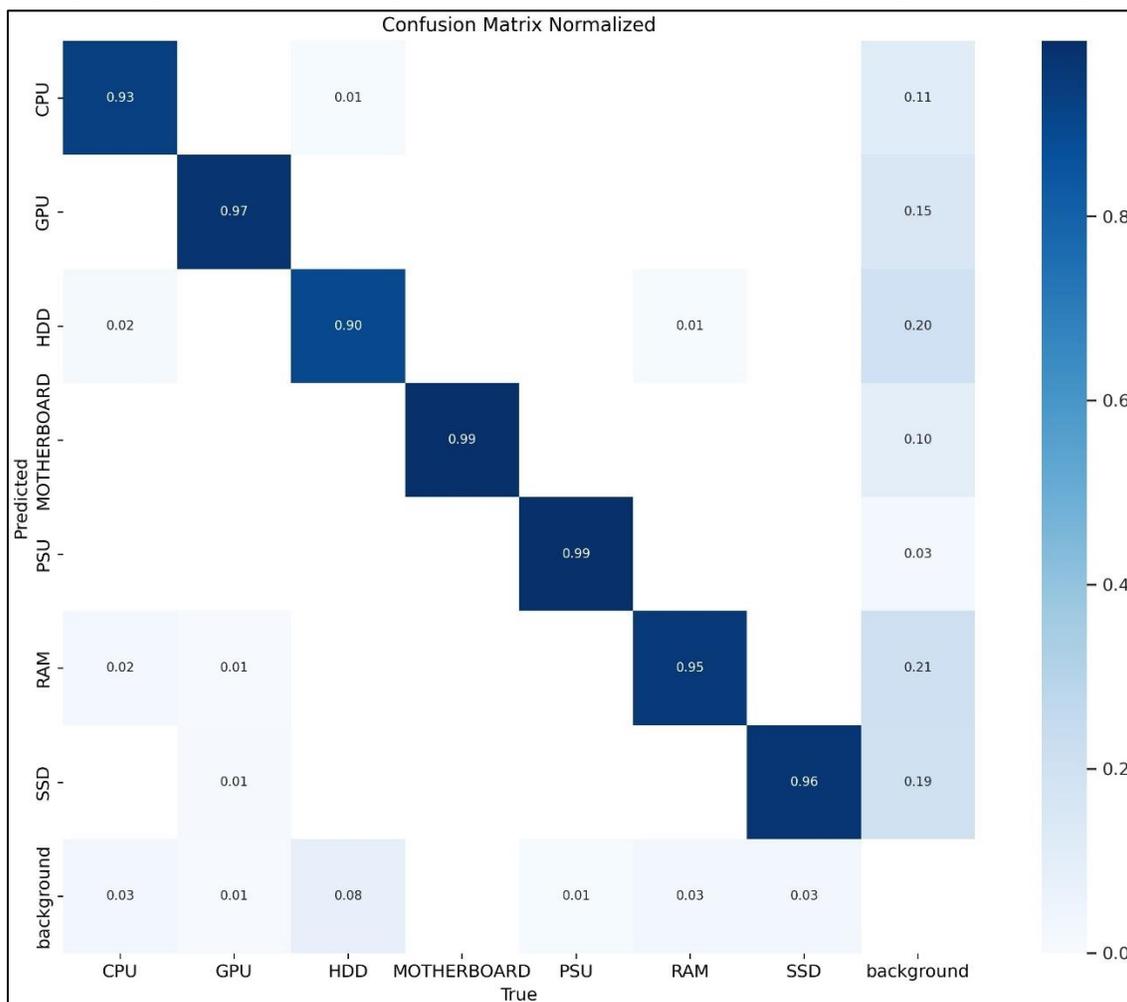


Figura 3.11: Matriz de confusión normalizada.

La figura 3.12 contiene algunas métricas arrojadas por el modelo, una de ellas indica la precisión de que el modelo posee una precisión de entre 0.97 y 0.98, del 97% o 98 %.

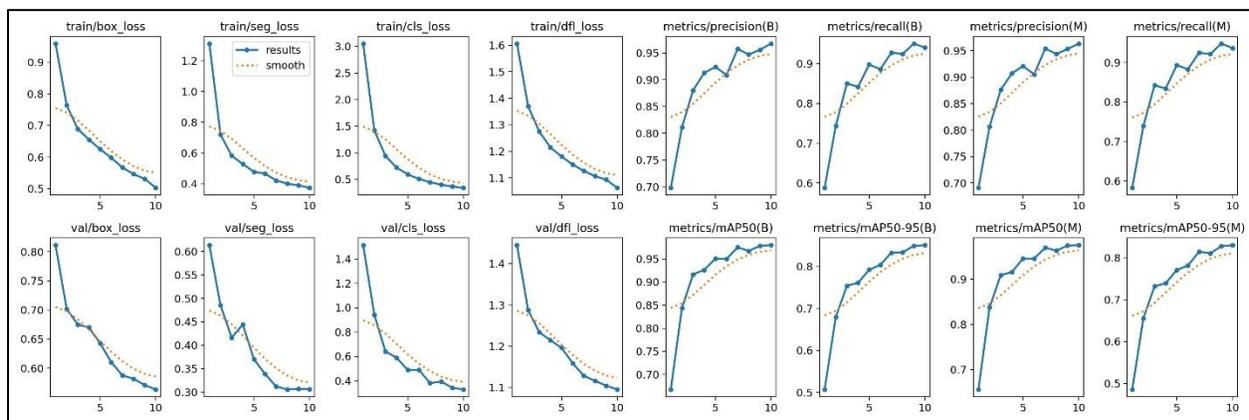


Figura 3.12: Gráficos de métricas del modelo

3.3.2 Detecciones correctas

Al ingresarle distintas imágenes de testeo al modelo, con características similares a lo que se encontraría en CIDIS, se obtuvieron buenos resultados en gran parte de las detecciones de recursos en las imágenes, a continuación, las Figuras 3.13 y 3.14 muestran ejemplos de detecciones correctas realizadas con el modelo de IA.

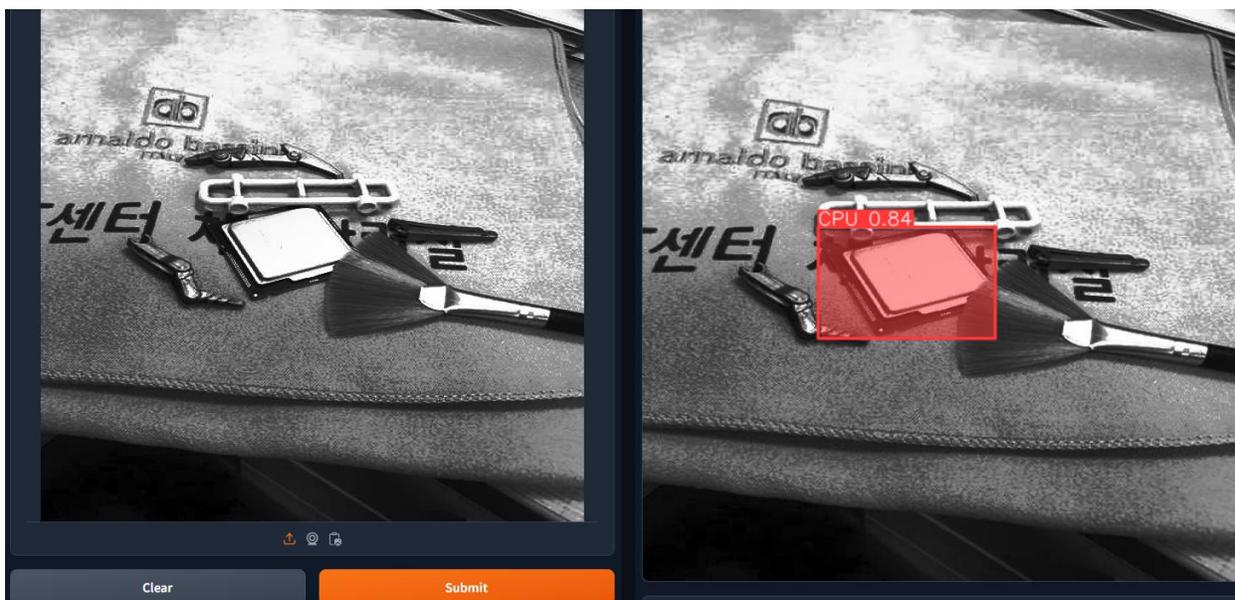


Figura 3.13: Detección correcta de un CPU.

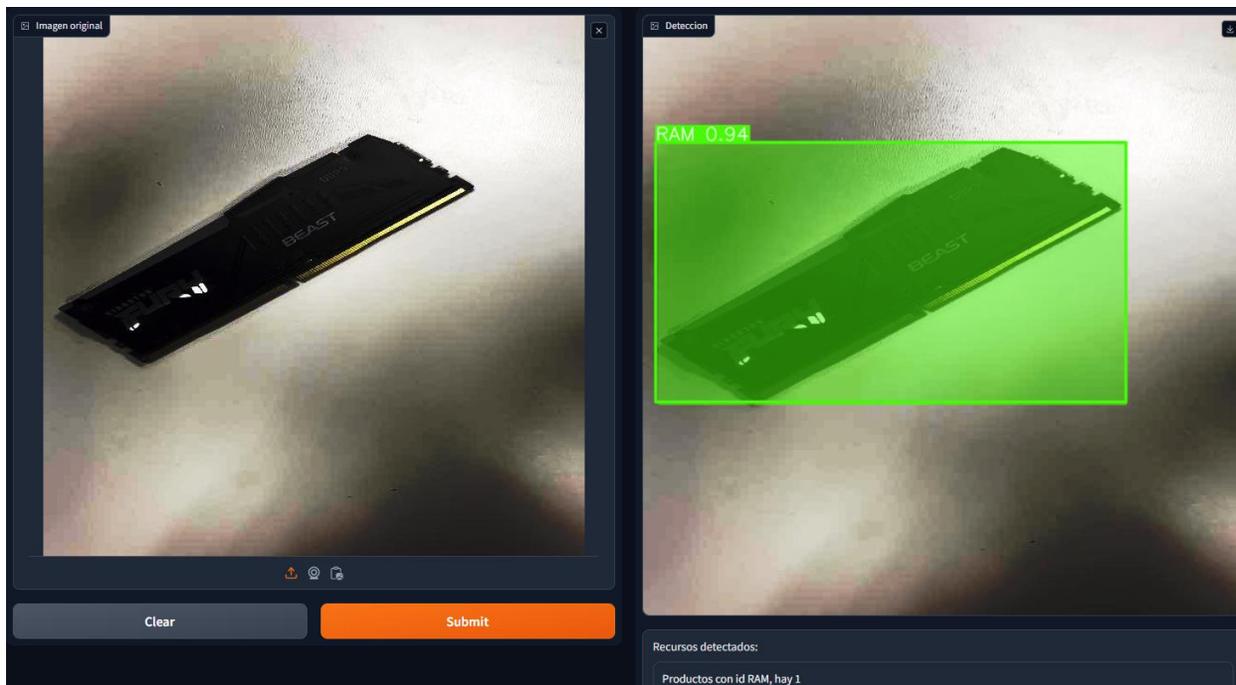


Figura 3.14: Detección correcta de una memoria RAM.

3.3.3 Detecciones incorrectas

El modelo también presentó problemas de detección en distintas imágenes que le fueron ingresadas, las *Figuras 3.15* y *3.16* contienen ejemplos de errores de detección.

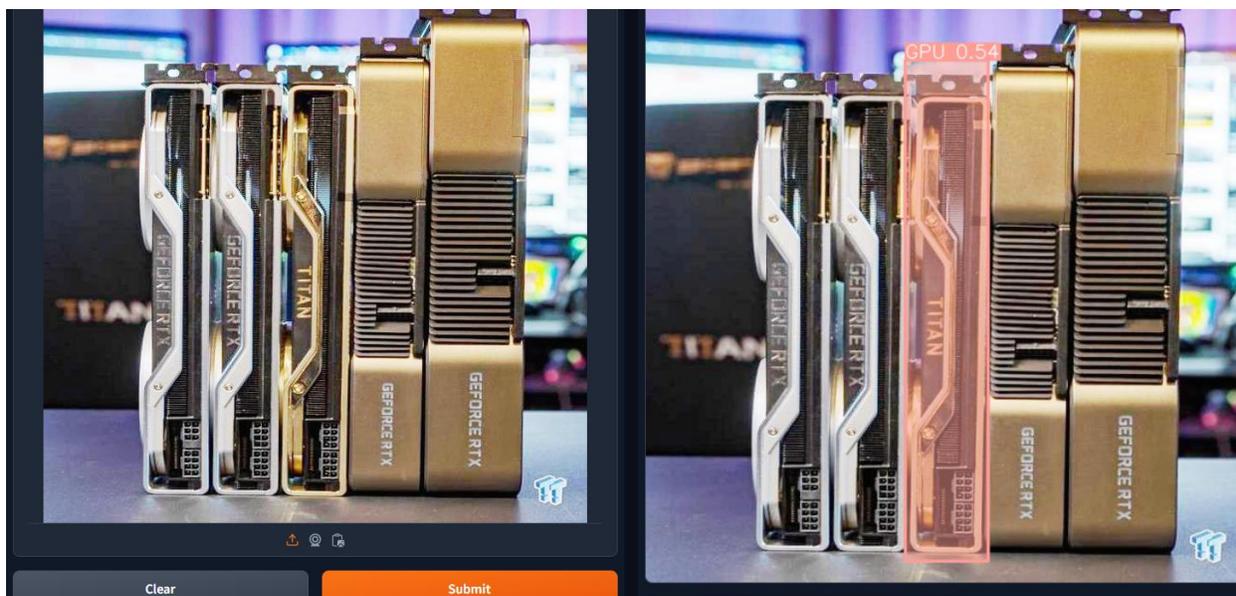


Figura 3.15: Error al detectar GPUs.

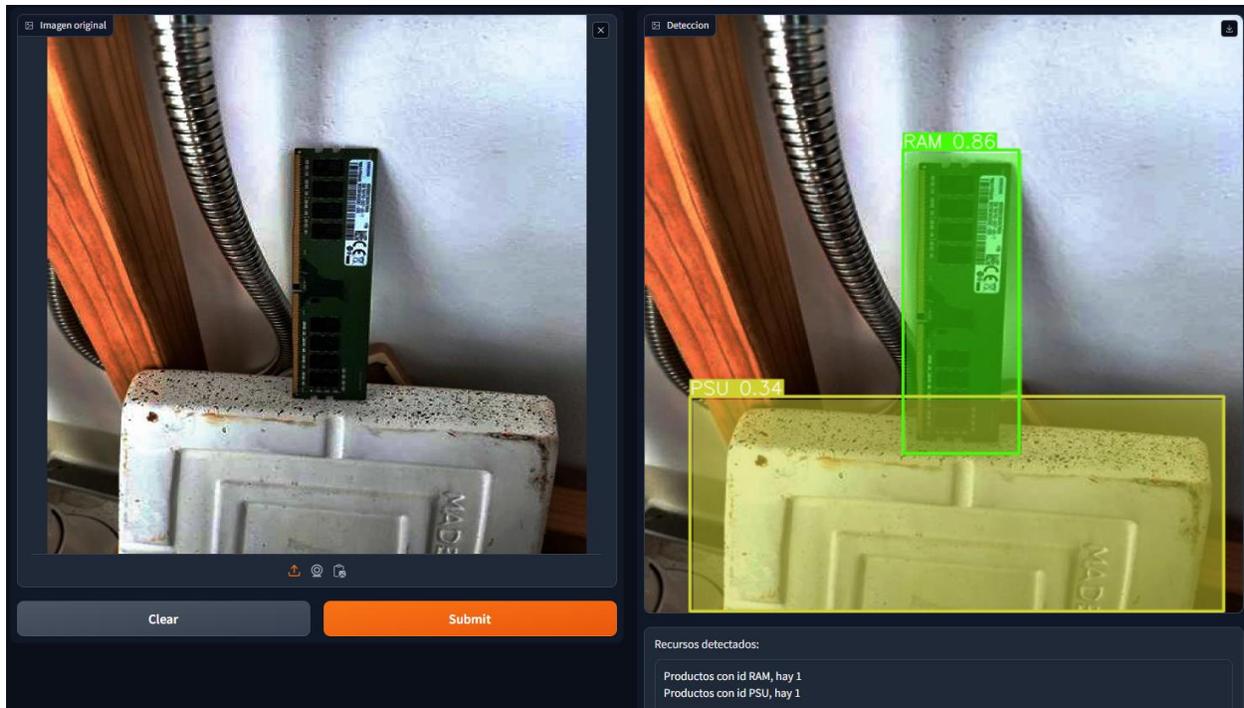


Figura 3.16: Falso positivo para una PSU.

3.3.4 Ambiente de desarrollo

El ambiente de desarrollo del modelo de IA estuvo conformado por dos fases, una de entrenamiento y otra de pruebas de detecciones.

La fase de entrenamiento contó los siguientes recursos:

- Google Drive
- Google Colab (plan gratuito)

Para la fase de prueba de detecciones se usó lo siguiente:

- PC con procesador Intel 10ma generación y 16 GB de memoria RAM.
- Windows 10.
- Python, versión 3.11.6

3.4 Pruebas con el cliente final

Parte importante en el desarrollo de este sistema, fueron las consultas frecuentes hacia el cliente y la retroalimentación provista por el mismo, para este proyecto se hicieron consultas referentes a colores de las vistas, diseño y contenido de las ventanas, flujo de eventos al realizar cada una de las acciones que permite el software y, las funciones permitidas para cada tipo de usuario admitido en el sistema.

Luego de tener todas las historias de usuario implementadas, dando como resultado un software funcional, se tuvieron dos reuniones para comprobar que el sistema cumplía con los requerimientos y expectativas del cliente, para estas reuniones el equipo presentó el sistema al cliente para que realice la interacción completa con el mismo.

En la primera reunión se utilizaron valores de prueba en algunos campos seleccionables, la retroalimentación mostró que, si bien el sistema cumplía con lo que se espera, existían campos que no estaban fácilmente a la vista y que para el cliente sería mejor tener un acceso más directo hacia los mismos. Los colores, flujos, acciones para cada tipo de usuario, y demás requerimientos fueron eficientemente cumplidos. El cliente solicitó una funcionalidad extra que, si bien no formaba parte de los requerimientos, incluirla no representaba un retraso en la entrega, por lo que se procedió a desarrollar dicha funcionalidad.

En la segunda reunión se verificó la incorporación de los campos solicitados en la primera reunión, se acordaron los valores oficiales que reemplazarán a los valores de prueba utilizados en la reunión previa y, se comprobó el correcto funcionamiento de la plataforma en base a los roles de cada usuario, la funcionalidad y la información mostrada en cada una de las vistas es la que es esperada. Como resultado de esta reunión se tiene que el sistema es completamente funcional y cumple con lo solicitado por el cliente para las actividades del CIDIS.

3.5 Análisis de costos

En lo que respecta a los gastos efectivos, el proyecto no incurrió en ningún costo de equipos o software, ya que fue llevado a cabo íntegramente por estudiantes pertenecientes a la carrera de Ingeniería en Computación de la Universidad ESPOL.

Aunque si planteamos el caso de que este proyecto fuese desarrollado por Ingenieros en computación como desarrolladores Junior; se puede realizar un aproximado de costos, basado en el tiempo en el que se trabajó, y la cantidad de personas que estuvieron involucradas.

Usando la información de la *Tabla 2.2*, en donde se encuentran las historias de usuario y el tiempo que demoro cada una; tenemos que la realización de todo el proyecto tomo un total de 54 días, lo que equivale a aproximadamente a 8 semanas.

Tomando como referencia que el sueldo promedio mensual de un desarrollador es de USD 1,500; y que la cantidad de personas encargadas de proyecto serian 2. El mismo tendría un costo aproximado de USD 6,000.

4. Capítulo 4

4.1 Conclusiones

- Se logró implementar un prototipo funcional de un sistema de inventario automatizado que permite el manejo eficiente de los registros de CIDIS, con la finalidad de optimizar las tareas de registro, consulta y seguimiento de los recursos de la organización.
- Se logró construir correctamente, tanto de manera visual, como en la consistencia de información, todo el proceso que conlleva la petición de un recurso del CIDIS. Desde saber si un recurso está disponible, hasta la aprobación o negación de una solicitud de préstamo. Lo que dejó un sentimiento de alta conformidad en el cliente.
- Se logró acoplar correctamente el CAS de ESPOL dentro del sistema de inventario, lo que garantiza que solo podrán usar el sistema aquellos usuarios que tengan sus correspondientes credenciales de ESPOL y sean válidos para el sistema.
- Las reuniones que se tuvieron con el cliente permitieron desarrollar vistas bastante precisas con respecto a la información que se debe mostrar y acciones que se deben habilitar a todos los usuarios que participan en el proceso de solicitud y prestamos de recursos del CIDIS.
- Aunque el dataset utilizado para el proyecto no es propio de CIDIS, los resultados del modelo muestran que un módulo de detección de este tipo puede ser eficientemente aplicado con recursos propios de CIDIS, esto debido a que las características contenidas en las imágenes son similares a lo que habría en CIDIS.

4.2 Recomendaciones

- Para el sistema de inventario, en la medida de lo posible, utilizar las tecnologías con las versiones mencionadas en el documento, tanto de Backend como de Frontend. ya que cambiarlas puede ocasionar fallas en el sistema, como problemas de visualización de

componentes del lado del cliente u errores en gestión de información por parte del servidor. Si se va a actualizar versiones, asegurarse completamente de que la actualización no genere errores.

- No se pueden editar o borrar de los roles arbitrariamente, ya que cada rol tiene su propia lógica definida, tanto en el Backend como en el Frontend. Si por motivos de fuerza mayor, se desea realizar las acciones antes mencionadas, debe previamente conocer los lugares exactos del código en donde debe editar/quitar las funciones.
- Si se desea agregar un nuevo rol, que cuente con determinadas funciones y permisos específicos, se debe conocer detenidamente el código en Frontend y Backend para agregarle donde corresponda los respectivos módulos que sean necesarios.
- Verificar que el servicio de CAS de ESPOL esté activo y sin problemas, ya que es obligatorio si se quiere obtener acceso al sistema.
- Cuando se ponga en producción el Sistema de Inventario y antes de usarlo de manera oficial, se deben realizar pequeñas pruebas para asegurar que el proceso de solicitudes y prestamos esté funcionando correctamente y sin inconsistencia de datos.
- Los usuarios solo pueden tener un rol y solo pueden acceder exclusivamente a las pestañas e información que tenga asignado su correspondiente rol. Por ejemplo, un usuario con rol de administrador no puede ingresar a secciones que estén vinculadas con otros roles.
- Al momento de realizar las pruebas de detección, se contaba con la versión 3.12.1 de Python, sin embargo, esta versión presentó problemas con ciertas librerías que aún no se encontraban disponibles para esta versión, por lo cual se decidió volver a la versión 3.11.6 de Python. Si desea usar versiones más recientes de Python o algún otro lenguaje, asegurarse de que las versiones sean compatibles.

Referencias

- [1] “LOS MÉTODOS DE CONTROL DE INVENTARIOS MÁS USADOS”. Escuela de Negocios - ESERP Digital Business & Law School. Accedido el 16 de octubre de 2023. [En línea]. Disponible: <https://es.eserp.com/articulos/metodo-control-inventarios/#:~:text=Un%20sistema%20de%20inventario%20es,que%20esta%20pueda%20funcionar%20eficazmente.>
- [2] “¿Qué es Deep Learning? | IBM”. IBM in Deutschland, Österreich und der Schweiz | IBM. Accedido el 16 de octubre de 2023. [En línea]. Disponible: <https://www.ibm.com/es-es/topics/deep-learning>
- [3] “Deep Learning o Aprendizaje profundo: ¿qué es?” Formation Data Science | DataScientest.com. Accedido el 16 de octubre de 2023. [En línea]. Disponible: <https://datascientest.com/es/deep-learning-definicion>
- [4] R. Rodríguez Abril. “Detección de objetos I: YOLO”. La Máquina Oráculo. Accedido el 16 de octubre de 2023. [En línea]. Disponible: <https://lamaquinaoraculo.com/deep-learning/deteccion-de-objetos/>
- [5] E. A. "Detección de objetos con YOLO: implementaciones y como usarlas". Medium. [https://medium.com/@enriqueav/detección-de-objetos-con-yolo-implementaciones-y-como-usarlas-c73ca2489246.](https://medium.com/@enriqueav/detección-de-objetos-con-yolo-implementaciones-y-como-usarlas-c73ca2489246)

[6] Vacca, C. (2018). DISEÑO Y DESARROLLO DE SOFTWARE PARA LA OPTIMIZACIÓN DEL INVENTARIO, GESTIÓN DE PRÉSTAMO DE ELEMENTOS TECNOLÓGICOS Y TRAZABILIDAD DE PROYECTOS DE INVESTIGACIÓN DEL CENTRO DE INVESTIGACIÓN (CEINDETEC). Universidad de los Llanos, FACULTAD DE CIENCIAS BASICAS E INGENIERIA. VILLAVICENCIO: Villavicencio: Universidad de los Llanos, 2019. Accedido el 16 de octubre de 2023. [En línea]. Disponible: <https://repositorio.unillanos.edu.co/handle/001/1487>

[7] Loor-Vera, A. T., Moreira-Santos, E. W., & Villao-Loor, M. B. (2021). Sistema de gestión de inventario, compra y venta de productos agrícolas para la toma de decisiones: Vinculación universidad y sociedad. Maestro Y Sociedad, 118-129. Calceta: Escuela Superior Politécnica Agropecuaria de Manabí. Accedido el 18 de octubre de 2023. [En línea]. Disponible: <https://maestroysociedad.uo.edu.cu/index.php/MyS/article/view/5451/5155>

[8] Serrano, A. (2019). SOFTWARE EN LA GESTIÓN PARA LA AUTOMATIZACIÓN DE PROCESOS. UNIVERSIDAD TÉCNICA DE AMBATO, FACULTAD DE CIENCIAS HUMANAS Y DE LA EDUCACIÓN. Ambato: Universidad Técnica de Ambato. Facultad de Ciencias Humanas y de la Educación. Carrera de Docencia en Informática. Accedido el 16 de octubre de 2023. [En línea]. Disponible: <https://repositorio.uta.edu.ec/jspui/handle/123456789/30600>

[9] Durán, Y. (2012). Administración del inventario: elemento clave para la optimización de las utilidades en las empresas. *Visión gerencial*, (1), 55-78. Durán: Universidad de los Andes Venezuela. Accedido el 18 de octubre de 2023. [En línea]. Disponible:

<https://www.redalyc.org/pdf/4655/465545892008.pdf>

[10] Parada, J. (2006). *Sistemas de inventario*. Ediciones PuntoCero. Caracas. Accedido el 18 de octubre de 2023. [En línea]. Disponible: [inventarios-libre.pdf \(d1wqtxts1xzle7.cloudfront.net\)](https://s3.amazonaws.com/d1wqtxts1xzle7.cloudfront.net/inventarios-libre.pdf)

[11] “GitHub - facebookresearch/detectron2: Detectron2 is a platform for object detection, segmentation and other visual recognition tasks.” GitHub. Accedido el 20 de octubre de 2023. [En línea]. Disponible: <https://github.com/facebookresearch/detectron2>

[12] “Papers with Code - Detectron2”. The latest in Machine Learning | Papers With Code. Accedido el 20 de octubre de 2023. [En línea]. Disponible: <https://paperswithcode.com/lib/detectron2>

[13] “Papers with Code - RetinaNet Explained”. The latest in Machine Learning | Papers With Code. Accedido el 20 de octubre de 2023. [En línea]. Disponible: <https://paperswithcode.com/method/retinanet>

[14] “Experiencia del usuario: entiende todo sobre este concepto”. Rock Content - ES. Accedido el 6 de noviembre de 2023. [En línea]. Disponible: <https://rockcontent.com/es/blog/experiencia-del-usuario/>

[15] “8 BENEFICIOS DE UN SISTEMA DE INVENTARIOS - Corposuite | Consultoría del ERP en la nube Oracle NetSuite México”. Corposuite | Consultoría del ERP en la nube Oracle NetSuite México. Accedido el 6 de noviembre de 2023. [En línea]. Disponible: <https://corposuite.com.mx/2020/07/02/8-beneficios-de-un-sistema-de-control-de-inventarios/>

[16] “Bsale Perú”. Bsale Perú. Accedido el 6 de noviembre de 2023. [En línea]. Disponible: <https://www.bsale.com.pe/article/5-beneficios-de-tener-un-sistema-de-control-de-inventarios-stock>

[17] “¿Cuáles son los beneficios de tener un software de inventarios? - Bind ERP”. My Blog. Accedido el 6 de noviembre de 2023. [En línea]. Disponible: <https://bind.com.mx/blog/control-de-inventarios/cuales-son-los-beneficios-de-tener-un-software-de-inventarios>

[18] M. Moin, “Unleashing the power of YOLOv8: A comprehensive guide to the real-time object detection model”, Augmented Startups, 11-ene-2023. [En línea]. Disponible en: <https://medium.com/augmented-startups/unleashing-the-power-of-yolov8-a-comprehensiveguide-to-the-real-time-object-detection-model-af71014a5e7a>.

[19] Torres Sierra, C. (2022). Análisis ABC y su relevancia en la gestión de inventarios: un estudio de revisión. Análisis sistemático de literatura. [En línea]. Disponible en: <https://repository.ucc.edu.co/items/62351e9d-04b9-4b11-a178-46440ed2c0b5>

[20] Izaurre, M. P. (2013). Caracterización de Especificación de Requerimientos en entornos Ágiles: Historias de Usuario. Trabajo de especialidad. [En línea]. Disponible en: [Anteproyecto_Requerimientos_en_Metodologías_Agiles.pdf \(utn.edu.ar\)](#)

[21] Kruchten, P. (1995). Planos Arquitectónicos: El Modelo de 4+ 1 Vistas de la Arquitectura del Software. IEEE software, 12(6), 42-50.

Apéndice A

Requerimientos del proyecto

Sistema de inventario

Requerimientos funcionales:

El sistema debe contar con tres roles de usuario (administrador, custodio, usuario general).

Del administrador:

- El sistema debe permitir agregar, editar y eliminar usuarios permitidos del sistema.
- El sistema debe permitir visualizar la lista de todos los usuarios permitidos del sistema.
- El sistema debe permitir agregar, editar y eliminar los recursos del registro (tanto aquellos que pueden ser solicitados en préstamo, como aquellos que forman parte del equipamiento de CIDIS).
- El sistema debe permitir visualizar la lista de todos los recursos, su información y su respectivo estado (en bodega, solicitado, en préstamo).
- El sistema debe permitir visualizar la información de las solicitudes de préstamo y su estado (pendiente, aprobado, negado).
- Cuando una solicitud sea aprobada por el custodio, el sistema debe permitir generar un préstamo.
- El sistema debe permitir visualizar la lista de recursos en préstamo y el detalle de información del préstamo.

Del custodio:

- El sistema debe permitir visualizar la información de las solicitudes de préstamo que le corresponde revisar.

- El sistema debe permitir aprobar o negar una solicitud de préstamo.
- El sistema debe permitir visualizar la lista de los recursos que están bajo su custodia (tanto aquellos que pueden ser solicitados en préstamo, como aquellos que forman parte del equipamiento de CIDIS) y su estado (en bodega, solicitado, en préstamo).
- El sistema debe permitir visualizar la información de los préstamos que aprobó.

Del usuario general.

- El sistema debe permitir visualizar la lista de recursos que tienen disponibilidad de ser solicitados en préstamo.
- El sistema debe permitir seleccionar el recurso, llenar los correspondientes datos para la solicitud de préstamo y enviar la solicitud.
- El sistema debe permitir visualizar el estado de sus solicitudes realizadas.

Requerimientos no funcionales:

- La interfaz de las diferentes pantallas del sistema debe brindar una buena experiencia de usuario a las personas que lo usarán.
- El sistema debe contar una interfaz intuitiva, fácil de entender su funcionamiento por parte de los usuarios.
- Los tiempos de respuesta a las peticiones que se realicen mediante el sistema deben ser óptimos.

Modelo de Deep Learning

Requerimientos funcionales:

- El modelo entrenado debe ser capaz de detectar y reconocer los recursos de CIDIS presentes en una imagen, además de determinar cuántos hay de cada uno.
- La solución debe generar un reporte con la cantidad de recursos detectados en la imagen.

Requerimientos no funcionales:

- Se requiere que la precisión de la detección sea medianamente aceptable y que se realice en tiempos óptimos.

Apéndice B

Historias de usuario

Historias de usuario para el rol *administrador*.

Identificador (ID) de la historia	Enunciado de la historia			Criterios de aceptación				
	Rol	Característica / Funcionalidad	Razón / resultado	Número (#) de escenario	Criterio de aceptación (Título)	Contexto	Evento	Resultado / Comportamiento esperado
AD001	Como un administrador.	Necesito ver la lista de todos los usuarios registrados en el sistema.	Con la finalidad de conocer quiénes son los usuarios que pueden hacer uso del sistema	1	La lista contiene a los usuarios de tipo Administrador.	En caso de que existan usuarios administradores.	Cuando se ingrese a la lista de usuarios.	Se mostrará la información asociada al usuario, junto con su categoría de Administrador.
				2	La lista contiene a los usuarios de tipo Custodio.	En caso de que existan usuarios custodios.	Cuando se ingrese a la lista de usuarios.	Se mostrará la información asociada al usuario, junto con su categoría de Custodio.
				3	La lista contiene a los usuarios de tipo Usuario General.	En caso de que existan usuarios generales.	Cuando se ingrese a la lista de usuarios.	Se mostrará la información asociada al usuario, junto con su categoría de Usuario General.

Identificador (ID) de la historia	Enunciado de la historia			Criterios de aceptación				
	Rol	Característica / Funcionalidad	Razón / resultado	Número (#) de escenario	Criterio de aceptación (Título)	Contexto	Evento	Resultado / Comportamiento esperado
AD002	Como un administrador.	Necesito visualizar la lista de recursos y su respectiva información.	Con la finalidad de conocer los recursos con los que cuenta el centro, su información y su estado.	1	Es posible ver la lista de recursos del Centro.	N/A	Cuando se ingrese a la lista de recursos.	El sistema mostrará la lista de los recursos del Centro, su información y su estado (En bodega, solicitado, en préstamo).

Identificador (ID) de la historia	Enunciado de la historia			Criterios de aceptación				
	Rol	Característica / Funcionalidad	Razón / resultado	Número (#) de escenario	Criterio de aceptación (Título)	Contexto	Evento	Resultado / Comportamiento esperado
AD003	Como un administrador.	Necesito agregar, editar y eliminar usuarios del sistema.	Con la finalidad de tener un control sobre los usuarios registrados y sus roles.	1	Es posible agregar el usuario.	Se desea registrar un nuevo usuario en el sistema.	Cuando se de click al botón de crear usuario.	Se mostrará el formulario para ingresar los datos del nuevo usuario y, después de agregar la información y hacer click en el botón registrar, el usuario se mostrará en la lista de usuarios registrados.
				2	Es posible editar el usuario.	Se desea actualizar la información de un usuario del sistema.	Cuando se de click al botón de editar usuario.	Se mostrará el formulario con los datos actuales del usuario y, después de actualizar la información y hacer click en el botón actualizar, los nuevos datos se mostrarán en la lista.
				3	Es posible borrar el usuario.	Se desea eliminar un usuario del sistema.	Cuando se de click al botón de borrar usuario.	El usuario se borrará del sistema y ya no se mostrará en la lista de usuarios registrados.

Identificador (ID) de la historia	Enunciado de la historia			Criterios de aceptación				
	Rol	Característica / Funcionalidad	Razón / resultado	Número (#) de escenario	Criterio de aceptación (Título)	Contexto	Evento	Resultado / Comportamiento esperado
AD004	Como un administrador.	Necesito agregar, editar y eliminar recurso del registro.	Con la finalidad de tener control sobre la información correspondiente a los recursos.	1	Es posible agregar un recurso.	Se desea registrar un nuevo recurso en el sistema.	Cuando se de click al botón de agregar recurso.	Se mostrará el formulario para ingresar los datos del nuevo recurso y, después de agregar la información y hacer click en el botón registrar, el recurso se mostrará en la lista de recursos registrados.
				2	Es posible editar un recurso.	Se desea editar la información de un recurso del sistema.	Cuando se de click al botón de editar recurso.	Se mostrará el formulario con los datos actuales del recurso y, después de actualizar la información y hacer click en el botón actualizar, los nuevos datos se mostrarán en la lista.
				3	Es posible eliminar un recurso.	Se desea eliminar del registro un recurso.	Cuando se de click al botón de borrar recurso.	El recurso se borrará del sistema y ya no se mostrará en la lista de recursos registrados.

Identificador (ID) de la historia	Enunciado de la historia			Criterios de aceptación				
	Rol	Característica / Funcionalidad	Razón / resultado	Número (#) de escenario	Criterio de aceptación (Título)	Contexto	Evento	Resultado / Comportamiento esperado
AD005	Como un administrador.	Necesito poder visualizar la información de las solicitudes de préstamo y el estado que poseen.	Con la finalidad de conocer el estado e información de las solicitudes de préstamo.	1	Es posible ver la lista de solicitudes de préstamo de recursos que se han realizado.	N/A	Cuando se ingrese a la lista de solicitudes.	El sistema mostrará la lista de solicitudes realizadas, su información y su estado (pendiente, aprobado, negado).

Identificador (ID) de la historia	Enunciado de la historia			Criterios de aceptación				
	Rol	Característica / Funcionalidad	Razón / resultado	Número (#) de escenario	Criterio de aceptación (Título)	Contexto	Evento	Resultado / Comportamiento esperado
AD006	Como un administrador.	Necesito generar préstamos a partir de solicitudes que fueron aprobadas.	Con la finalidad de tener un control de los préstamos que se generan.	1	Es posible generar el préstamo.	Cuando una solicitud de préstamo tiene el estado "aprobado".	Cuando se presiona el botón generar el préstamo.	Se muestra una ventana con la información de la solicitud de préstamo y un botón de generar préstamo. Al hacer click se genera el préstamo y se registra en la base de datos.

Identificador (ID) de la historia	Enunciado de la historia			Criterios de aceptación				
	Rol	Característica / Funcionalidad	Razón / resultado	Número (#) de escenario	Criterio de aceptación (Título)	Contexto	Evento	Resultado / Comportamiento esperado
AD007	Como un administrador.	Necesito ver la lista de recursos en préstamo y el detalle de información del préstamo.	Con la finalidad de tener un control de los préstamos que se han realizado.	1	Se visualiza la lista de préstamos realizados.	N/A	Cuando se ingrese a la lista de préstamos realizados.	Se muestra la lista de préstamos realizados, junto con sus correspondientes datos de detalle.

Historias de usuario para el rol *custodio*.

Identificador (ID) de la historia	Enunciado de la historia			Criterios de aceptación				
	Rol	Característica / Funcionalidad	Razón / resultado	Número (#) de escenario	Criterio de aceptación (Título)	Contexto	Evento	Resultado / Comportamiento esperado
CU001	Como un custodio.	Necesito visualizar las solicitudes de préstamo que me corresponde revisar.	Para conocer los motivos por los que realizaron la solicitud.	1	Se visualiza la lista de solicitudes que me corresponden (las que involucran recursos que están bajo mi custodia).	N/A	Al ingresar a la lista de solicitudes.	En la lista se muestra breve información acerca de la solicitud, al hacer click en el botón de ver información, se muestra la información completa acerca de dicha solicitud de préstamo.

Identificador (ID) de la historia	Enunciado de la historia			Criterios de aceptación				
	Rol	Característica / Funcionalidad	Razón / resultado	Número (#) de escenario	Criterio de aceptación (Título)	Contexto	Evento	Resultado / Comportamiento esperado
CU002	Como un custodio.	Necesito ver la lista de recursos que están bajo mi custodia y su estado.	Con la finalidad de conocer el estado de los recursos que están a mi cargo.	1	Es posible ver la lista de recursos que están bajo su custodia, su información y su correspondiente estado.	N/A	Cuando se ingrese a la lista de recursos	El sistema mostrará la lista de recursos que están bajo su custodia, su información y su estado (en CIDIS, en bodega, solicitado, en préstamo).

Identificador (ID) de la historia	Enunciado de la historia			Criterios de aceptación				
	Rol	Característica / Funcionalidad	Razón / resultado	Número (#) de escenario	Criterio de aceptación (Título)	Contexto	Evento	Resultado / Comportamiento esperado
CU003	Como un custodio.	Necesito aprobar o negar las solicitudes de préstamo que me corresponden.	Con la finalidad de tener decisión sobre cuáles solicitudes aprobar y cuáles solicitudes negar.	1	Es posible aprobar una solicitud.	En caso de que se desee aprobar la solicitud.	Cuando se de click al botón de ver información completa.	El sistema mostrará la información completa de la solicitud y, después de hacer click en el botón "aprobar", la solicitud cambiará su estado a "aprobado" y se mostrará su nuevo estado en la lista de solicitudes.
				2	Es posible negar una solicitud.	En caso de que se desee negar la solicitud.	Cuando se de click al botón de ver información completa.	El sistema mostrará la información completa de la solicitud y, después de hacer click en el botón "negar", la solicitud cambiará su estado a "negado" y se mostrará su nuevo estado en la lista de solicitudes.

Identificador (ID) de la historia	Enunciado de la historia			Criterios de aceptación				
	Rol	Característica / Funcionalidad	Razón / resultado	Número (#) de escenario	Criterio de aceptación (Título)	Contexto	Evento	Resultado / Comportamiento esperado
CU004	Como un custodio.	Necesito ver la información acerca de las solicitudes que aprobé y que ya generaron un préstamo.	Para tener un historial de los recursos que he dado en préstamo.	1	Se muestra la lista de préstamos generados.	N/A	Cuando se ingrese a la lista de préstamos realizados.	El sistema mostrará la información acerca de los préstamos que se realizaron.

Historias de usuario para el rol *usuario general*.

Identificador (ID) de la historia	Enunciado de la historia			Criterios de aceptación				
	Rol	Característica / Funcionalidad	Razón / resultado	Número (#) de escenario	Criterio de aceptación (Título)	Contexto	Evento	Resultado / Comportamiento esperado
UG001	Como un usuario general.	Necesito ver la lista de recursos que tienen disponibilidad de solicitar en préstamo.	Con la finalidad de conocer cuáles recursos del centro actualmente están disponibles para ser solicitados.	1	Se visualiza la lista de recursos disponibles.	N/A	Al ingresar a la lista de recursos disponibles.	El sistema mostrará la lista de recursos que están disponibles para ser solicitados en préstamo, junto con los datos acerca del recurso.

Identificador (ID) de la historia	Enunciado de la historia			Criterios de aceptación				
	Rol	Característica / Funcionalidad	Razón / resultado	Número (#) de escenario	Criterio de aceptación (Título)	Contexto	Evento	Resultado / Comportamiento esperado
UG002	Como un usuario general.	Necesito seleccionar algún recurso y realizar la solicitud de préstamo.	Con la finalidad de generar una solicitud de préstamo.	1	Es posible solicitar un préstamo de un recurso.	N/A	Al hacer click en el botón "solicitar préstamo".	Se mostrará un formulario con los datos que deben ser llenados para solicitar el préstamo, cuando los datos sean llenados, al presionar el botón "solicitar" se generará una solicitud de préstamo.

Identificador (ID) de la historia	Enunciado de la historia			Criterios de aceptación				
	Rol	Característica / Funcionalidad	Razón / resultado	Número (#) de escenario	Criterio de aceptación (Título)	Contexto	Evento	Resultado / Comportamiento esperado
UG003	Como un usuario general.	Necesito visualizar el estado de las solicitudes que he realizado.	Con la finalidad de conocer el estado de las solicitudes que he realizado.	1	Es posible ver el estado de mis solicitudes.	N/A	Al ingresar a la lista de solicitudes realizadas.	Se mostrará la lista de solicitudes que el usuario realizó, junto con la información de esa solicitud, incluido su estado (pendiente, aprobado, negado).