UNIVERSIDAD LAICA ELOY ALFARO DE MANABÍ EXTENSIÓN EL CARMEN

CARRERA DE INGENIERÉA EN TECNOLOGÍAS DE LA INFORMACIÓN

PROYECTO INTEGRADOR

PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERA EN TECNOLOGÍAS DE LA INFORMACIÓN

SISTEMA INFORMÁTICO CON REDES NEURONALES PARA LA ADMINISTRACIÓN DEL LABORATORIO DE COMPUTACIÓN DE LAS CARRERAS DE TI-SW EN ULEAM EXTENSIÓN EL CARMEN

CARRANZA MOREIRA TIFANI BELÉN
AUTOR

ING. REASCOS PINCHAO RAÚL SAED, MG.
TUTOR

EL CARMEN, AGOSTO 2025

CERTIFICACIÓN DE TUTOR

	NOMBRE DEL DOCUMENTO: CERTIFICADO DE TUTOR	CÓDIGO: PAT-04-F-004
Uleam ELOY ALFARO DE MANABÍ	PROCEDIMIENTO: TITULACIÓN DE ESTUDIANTES DE GRADO BAJO LA UNIDAD DE INTEGRACIÓN CURRICULAR	REVISIÓN: 1
		Página 1 de 1

CERTIFICACIÓN

En calidad de docente tutor de la Extensión El Carmen de la Universidad Laica "Eloy Alfaro" de Manabí, CERTIFICO:

Haber dirigido, revisado y aprobado preliminarmente el Trabajo de Integración Curricular bajo la autoría de la estudiante CARRANZA MOREIRA TIFANI BELEN, legalmente matriculada en la carrera de Ingeniería de Tecnologías de la Información, período académico 2024(2)-2025(1), cumpliendo el total de 384 horas, cuyo tema del proyecto es "SISTEMA INFORMÁTICO CON REDES NEURONALES PARA LA GESTIÓN DEL LABORATORIO DE COMPUTACIÓN DE LAS CARRERAS DE TI-SW EN ULEAM EXTENSIÓN EL CARMEN".

La presente investigación ha sido desarrollada en apego al cumplimiento de los requisitos académicos exigidos por el Reglamento de Régimen Académico y en concordancia con los lineamientos internos de la opción de titulación en mención, reuniendo y cumpliendo con los méritos académicos, científicos y formales, y la originalidad del mismo, requisitos suficientes para ser sometida a la evaluación del tribunal de titulación que designe la autoridad competente.

Particular que certifico para los fines consiguientes, salvo disposición de Ley en contrario.

El Carmen, 15 de agosto de 2025.

Lo certifico,

Ing. Saed Reascos Pinchao, Mg.
Docente Tutor
Área: Tecnologías de la Información

TRIBUNAL DE SUSTENTACIÓN

Universidad Laica Eloy Alfaro de Manabí
Extensión El Carmen
Carrera de Ingeniería en Tecnologías de la Información

TRIBUNAL DE SUSTENTACIÓN

Título del Trabajo de Titulación: Sistema Informático con Redes Neuronales para la Gestión del Laboratorio de Computación de las Carreras de TI-Sw en Uleam Extensión El Carmen

Modalidad: Proyector Integrador

Autora: Carranza Moreira Tifani Belen

Tutor: Ing. Reascos Pinchao Raul Saed, Mg.

Tribunal de Sustentación:

Presidente:
Ing. Mora Marcillo Alex Bladimir, Mg. _____

Miembro:
Ing. Arévalo Hermida Rómulo Danilo, Mg. _____

Miembro:
Ing. Mendoza Villamar Rocio Alexandra, Mg. _____

Fecha de Sustentación: 11 de septiembre de 2025

DECLARACIÓN DE AUDITORÍA

UNIVERSIDAD LAICA "ELOY ALFARO" DE MANABÍ

EXTENSIÓN EN EL CARMEN

DECLARACIÓN DE AUDITORÍA

La responsabilidad del contenido de este Trabajo de titulación, cuyo tema es: SISTEMA INFORMÁTICO CON REDES NEURONALES PARA LA ADMINISTRACIÓN DEL LABORATORIO DE COMPUTACIÓN DE LAS CARRERAS DE TI-SW EN ULEAM EXTENSIÓN EL CARMEN, corresponde exclusivamente a: CARRANZA MOREIRA TIFANI BELÉN con CI. 1314102649, y los derechos patrimoniales de la misma corresponden a la Universidad Laica Eloy Alfaro de Manabí.

Carranza Moreira Tifani Belén
C.I. 1314102649

DEDICATORIA

Dedico este trabajo en primer lugar, a Dios por brindarme la sabiduría, guiarme y fortalecerme en este camino lleno de oportunidades.

A mi padre, mi pilar y apoyo incondicional, quien con su amor y enseñanzas me mostró cómo superar las dificultades y ser fuerte, aunque no esté físicamente a mi lado. A mi madre, por su dedicación, confianza y por creer en mis capacidades, motivándome siempre a alcanzar mis sueños. A ambos les agradezco con todo mi corazón y alma, su esfuerzo y amor han sido fundamentales en mi vida.

Carranza Tifani

AGRADECIMIENTO

Expreso mi más sincero agradecimiento a mis profesores, por compartir generosamente sus conocimientos y experiencias, contribuyendo a mi formación académica y profesional. Y de manera especial, a mi tutor por su guía constante, paciencia y valiosas recomendaciones en cada etapa de este trabajo. Su compromiso y experiencia han sido fundamentales para hacer de este proyecto una experiencia significativa, inspirándome a seguir mejorando cada día.

Carranza Tifani

ÍNDICE DE CONTENIDOS

ÍNDICE TABLAS

ÍNDICE GRÁFICOS E ILUSTRACIONES

ÍNDICE DE ANEXOS

RESUMEN

En la carrera de Tecnología de la información de la Universidad Laica Eloy Alfaro de Manabí Extensión El Carmen, se identificó un problema de gestión y seguridad en el laboratorio de cómputo por el uso indebido de la movilización de los equipos que evidenció deterioró y riesgo de pérdidas. El objetivo de esta investigación fue desarrollar e implementar un sistema informático basado en redes neuronales para la administración del laboratorio a través del reconocimiento de objetos en tiempo real.

En el marco metodológico se aplicó una investigación de tipo bibliográfica, de campo y aplicada. Se seleccionó una muestra de 76 individuos, compuesta por 74 estudiantes encuestados y dos encargados del laboratorio entrevistados, con el objetivo de validar las causas del problema. La propuesta del proyecto desarrolló a través de la metodología ágil Scrum que se estructuró en 9 Sprints incluyendo la recolección del conjunto de datos etiquetados, el entrenamiento del modelo con las redes neuronales convolucionales (CNN) y YOLOv8 con la implementación de la alerta e interfaz web.

Los resultados demostraron que el sistema alcanzó una precisión de 80% en la detección de equipos y la implementación exitosa del registro e inicio de sesión. Se concluye la solución desarrollada constituye un sistema automático y óptimo para la gestión del inventario, lo que fortalece la seguridad y reduce el riesgo de pérdidas.

ABSTRACT

In the Information Technology program at Laica Eloy Alfaro University of Manabí, El Carmen Extension, a management and security problem was identified in the computer lab due to the improper use of equipment, which led to deterioration and risk of loss. The objective of this research was to develop and implement a computer system based on neural networks for laboratory management through real-time object recognition.

The methodological framework involved bibliographic, field, and applied research. A sample of 76 individuals was selected, consisting of 74 students surveyed and two laboratory managers interviewed, with the aim of validating the causes of the problem. The project proposal was developed using the agile Scrum methodology, which was structured in nine sprints, including the collection of the labeled dataset, the training of the model with convolutional neural networks (CNN) and YOLOv8, and the implementation of the alert and web interface.

The results showed that the system achieved 80% accuracy in equipment detection and successful implementation of registration and login. It was concluded that the developed solution constitutes an automatic and optimal system for inventory management, which strengthens security and reduces the risk of losses.

CAPÍTULO I

1 INTRODUCCIÓN

1.1 Introducción

La transformación digital en la actualidad ha exige a las instituciones universitarias optimizar la gestión de sus recursos tecnológicos. En este contexto, la Universidad Laica Eloy Alfaro de Manabí (ULEAM), Extensión El Carmen, enfrenta un desafío significativo en el manejo y supervisión de los equipos de cómputo del laboratorio 01. Actualmente, el aula cuenta con 24 equipos y lo utilizan los estudiantes de la carrera de software, especialmente durante la jornada matutina, donde se centra la mayor demanda de uso académico.

La falta de control adecuado ha provocado riesgos importantes a los equipos como pérdidas, deterioro o daños significativos. El traslado de estos dispositivos no se realiza conforme a la política, lo que ha ocasionado retrasos las actividades académicas. Además, esta situación representa un impacto económico para la institución.

Ante esta situación, surge la necesidad de implementar un sistema informático basado en redes neuronales para reconocer y monitorear constantemente los equipos informático, con el objetivo de fortalecer la administración y seguridad del laboratorio. Además, con esta implementación tecnológica será capaz de presenciar, identificar y detectar movimientos inadecuados en tiempo reales.

El desarrollo de esta propuesta tiene como propósito adaptarse a la metodología ágil Scrum, estructurada en Sprints que incluye las fases de análisis, desarrollo, validación y retroalimentación continua. Se utilizará herramientas de clasificación y detección de los objetos, priorizando aquellos dispositivos con mayor valor económico y necesidad de control.

En consecuencia, este estudio respalda la urgencia de una solución que permita reducir las pérdidas, la administración de control, la optimización del uso disponible y la mejora el entorno académico. La integración de este sistema de vigilancia inteligente tiene como objetivo favorecer un ambiente seguro, eficaz y ordenado para el desarrollo de futuras actividades académicas dando beneficio principalmente a los estudiantes y docentes.

1.2 Presentación del tema

En este proyecto se aborda la problemática de la inseguridad en el laboratorio de computación 01 de la Universidad Laica Eloy Alfaro de Manabí, Extensión El Carmen, específicamente en las carreras de Tecnología de la Información y Comunicación (TIC), Software (SW) y Electromecánica.

La primera prevención de seguridad es reducir el riesgo de pérdidas o daños en el laboratorio causado por el acceso de cualquier usuario. Esta situación ha causado riesgos significativos como consecuencias graves de robos, daños o desorganización de cables de los equipos informáticos.

Para disminuir estos riesgos se propone la implementación de este sistema de seguridad basado en redes neuronales para permitir detectar movimientos inusuales. Este sistema se desarrolla para controlar el riesgo de pérdidas y monitores en tiempo real mostrando cualquier señal detectada en el laboratorio.

1.3 Ubicación y contextualización del problema

El laboratorio de computación 01 se encuentra en el aula 201 de la planta alta de la universidad Laica Eloy Alfaro de Manabí Extensión El Carmen, en la carrera de TI, SW y Electromecánica. Está ubicado en la Av. 3 de Julio, El Carmen, y presenta las siguientes características:

Dimensiones: El laboratorio ocupa un espacio de aproximadamente 6m x 8m, destinado exclusivamente en el área de computación.

Cableado: La instalación de los cables están a través de canaletas lo que permite una conexión segura y organizada de los equipos.

Equipos: El laboratorio cuenta con 22 computadoras, cada uno con sus respectivo monitor, teclado, mouse y regulador de voltaje.

Ventilación: El aula está equipada con un aire acondicionado nuevo que proporciona adecuada ventilación. Además, cuenta con una gran ventada en la parte inferior.

Instalación eléctrica: Cada columna de las filas de los monitores dispone de conectores eléctricos ubicado en el suelo y en las paredes, con reguladores de voltaje para prevenir daños por descargas eléctricas.

Pizarra y proyector: El laboratorio cuenta con estos equipos para facilitar la enseñanza y presentación de contenidos visuales durante las clases.

Piso: En el laboratorio 01 se observa roturas en ciertas áreas del piso.

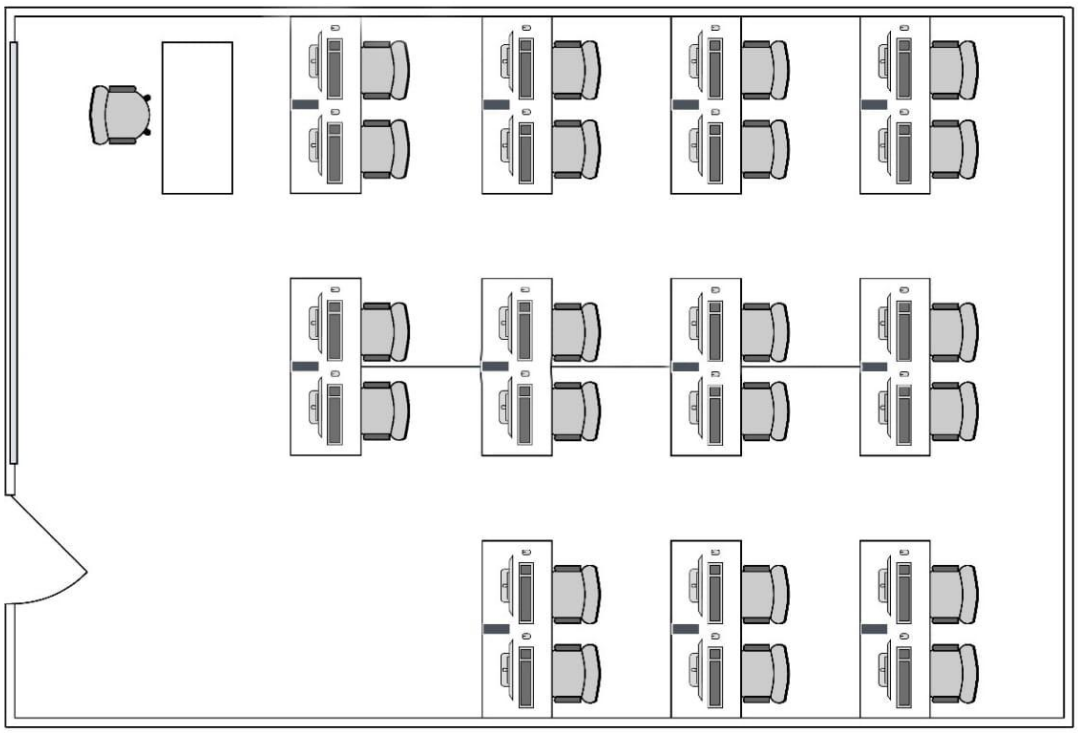


Figura 1 Diagrama del Laboratorio 01

Este laboratorio está equipado con monitores, ratón, teclados y CPU adquiridos entre 2016 y 2022. Este es un registro general del inventario actual del aula.

1.4 Planteamiento del problema

Se ha identificado algunos problemas e impactos originados por la ausencia de seguridad en el laboratorio 01 de la Universidad Laica Eloy Alfaro de Manabí, Extensión El Carmen. Esta circunstancia compromete los equipos en un grado alto de riesgo por robo, pérdidas de trabajo y desorden en el cableado.

Actualmente en el laboratorio no cuenta con medidas de seguridad eficiente lo que hace obligatorio y necesario implementar un sistema de vigilancia constante en el aula para garantizar la protección de los equipos. Lo que ha permitido identificar las causas afectadas por acciones inadecuadas y mostrar un enfoque tecnológico que se ofrece una solución sostenible.

1.4.1 Problematización

¿Qué tipo de inconveniente o desafíos podría surgir en el laboratorio?

¿Cuál es el grado de protección para salvaguardar los equipos del laboratorio?

¿Qué efectos puede generar la pérdida o daños de algún equipo del laboratorio?

1.4.2 Genesis del problema

El problema en el laboratorio 01 surgió debido a la inseguridad que presenta, originada por la inadecuada manipulación de los equipos. Esta situación representa un riesgo tanto operativo como económico para la Universidad. La falta de control adecuado sobre los equipos ha llevado a la desaparición de elementos como teclados y mouse, y a la desconexión de cables.

Anteriormente, el laboratorio contaba con equipos de hardware y software básicos los cuales han sido remplazados por unos más avanzados y adecuados para el aprendizaje, permitiendo ser más eficiente. Actualmente, la cantidad de estudiantes se ha ido aumentando especialmente en las carreras de TIC, SW y Electromecánica, quienes frecuentan más el laboratorio bajo la supervision de un profesor. Una de las causas que ha contribuido al problema de seguridad son los incidentes ocasionados por la ausencia del aula. Si bien estos incidentes no son frecuentes se han reportados pérdidas materiales por parte de los estudiantes.

La principal causa de los incidentes es la ausencia de control de los equipos por pérdidas, daños, desconexión de cables, mouses y teclados. Aunque existe un conserje encargado de abrir y cerrar el laboratorio al inicio y final de las clases, no se cuenta con una vigilancia constante, ya que sus funciones se limitan únicamente a esta tarea y no incluyen monitoreo activo durante las actividades.

1.4.3 Estado actual del problema

El laboratorio 01 cuenta con 22 monitores y CPU en buen estado, lo que se agrava la situación por el aumento de estudiantes debido a la nueva carrera de SW. Aunque el conserje este supervisando no es un monitoreo constante, ya que pueden ocurrir daños o desconexión de cables, estas dificultades interrumpen las actividades académicas de los estudiantes.

La falta del sistema de vigilancia adecuado impide el monitoreo constante lo que dificulta la identificación del responsable ante incidentes como pérdidas, lo que permitirá obtener medidas correctivas en el laboratorio. Esta problemática del laboratorio es suficiente

justificación para fortalecer los mecanismos de control y vigilancia con el fin de gestionar operatividad diaria.

1.5 Diagrama causa efecto

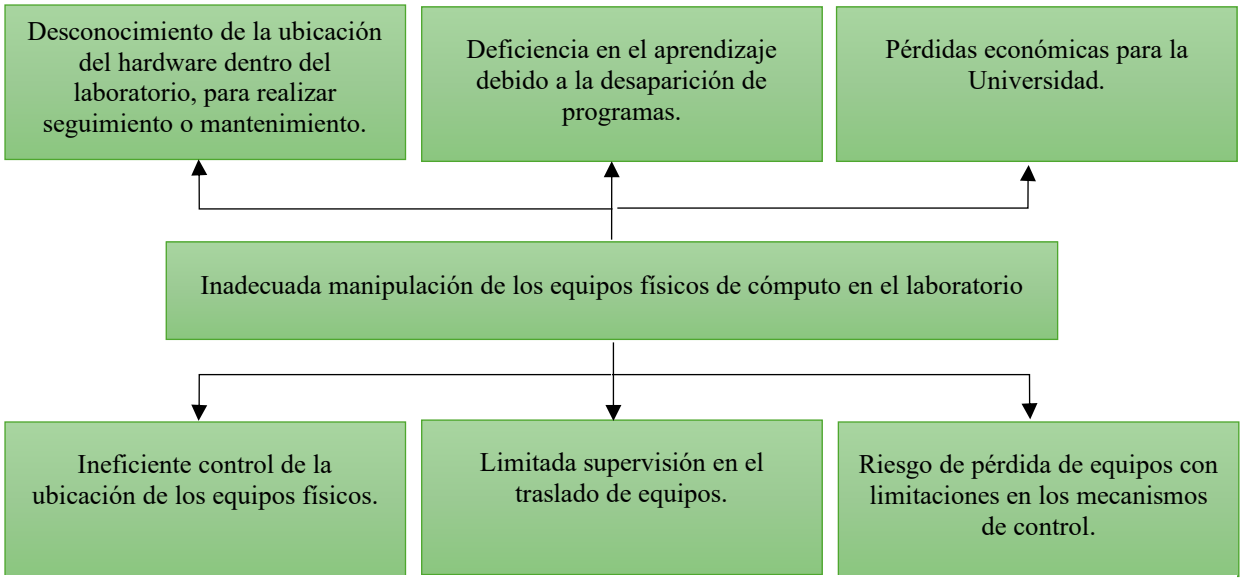


Figura 2 Diagrama causa y efectos

1.6 Objetivos

1.6.1 Objetivo general

Implementar un Sistema Informático con Redes Neuronales para la Administración del Laboratorio de Computación de las carreras de TI-SW en Uleam Extensión El Carmen.

1.6.2 Objetivos específicos

- Planificar el análisis del problema mediante modelos sistémicos para establecer la planificación del presente proyecto.
- Fundamentar teóricamente usando citas APA el desarrollo de un sistema informático que incorpora redes neuronales para mejorar la seguridad del Laboratorio de Cómputo 01 en ULEAM El Carmen.
- Validar la existencia de las causas del problema con métodos de investigación para determinar las áreas de mayor problema a intervenir.
- Desarrollar el programa con uso de algoritmo con redes neuronales para identificar acciones inadecuadas de los equipos del laboratorio.

1.7 Justificación

La presente investigación se justifica por la necesidad de un mecanismo de seguridad y control del laboratorio 01 que ha registrado la desaparición de equipos, una situación que afecta económicamente a la Universidad y los estudiantes por pérdidas de trabajos. Lo que aborda un problema por el entorno de identificar los inconvenientes realizados en el laboratorio, protegiendo el bienestar de la institución y a los estudiantes para su formación.

La propuesta representa un mecanismo optimó para la implementación de un sistema permitiendo reducir los riesgos y facilita un monitoreo constante, garantizando una vigilancia en tiempo real y una eficiente seguridad. La Universidad será beneficiaria económica y académicamente, aumentando la confianza en los estudiantes y profesores mostrando un entorno seguro. Además, promueve el aprendizaje práctico en innovaciones tecnológicas y técnicas de aprendizaje profundo, fortaleciendo la seguridad en el ámbito educativo.

1.8 Impactos esperados

El laboratorio 01 se beneficiará significativamente con esta implementación del sistema dando mejoras de seguridad, con el objetivo de aumentar la vigilancia y reducir los inconvenientes que se realizan a diario.

No solo tiene como objetivo aumentar la seguridad sino minimizar las pérdidas económicas que se han presentado y la protección de trabajos académicos. Al futuro presentará un entorno más seguro y confiado para los universitarios. Estas nuevas tecnologías ayudan a mejorar sistemas de seguridad en todo tipo de ambiente y en especial un monitoreo constante para el laboratorio de la Universidad fortaleciendo sus mecanismos de protección.

1.8.1 Impacto tecnológico

Las tecnologías para desarrollar un software funcional y fiable para este tipo de eventos se basan con las redes neuronales convolucionales que son capaz de entrenar modelos, lo que ha facilitado analizar, optimizar y detectar los reconocimientos de los objetos. Empleando este sistema se incrementa la seguridad y reducción de costos por una supervision más constante para el encargado.

La incorporación de este sistema representa un avance significativo para la Universidad, permitiendo el análisis de patrones y comportamientos malintencionados, y aumentando la capacidad del sistema para adaptarse a nuevas amenazas. Este sistema innovador ayudará a la

Universidad a mantener un control diario más efectivo. A largo plazo, beneficiará a los estudiantes de la carrera, quienes podrán darle mantenimiento y seguir implementándolo en otros laboratorios.

Se espera un aumento de seguridad a través de este sistema con inteligencia artificial para la Universidad, así logrando mantener un efecto positivo en el entorno académico y fomentando a los futuros estudiantes crear e implementar proyectos más avanzados con estas nuevas tecnologías.

1.8.2 Impacto social

La implementación de este sistema es fundamental para la Universidad dando beneficios significativo y proporcionado un espacio de trabajó confiable sin ningún tipo de riesgo. Este entorno es crucial para mejorar su responsabilidad como una segunda casa, lo que favorece un conocimiento y propósito de aumentar avances tecnológicos.

Este proyecto aporta un avance en las nuevas tecnologías dando creatividad y facilidad en la creación e implementación de sistemas de vigilancia, este constante cuidado mejora el tiempo y reduce las dificultades en el laboratorio. Además, da un éxito para prácticas y mejoras del sistema para los estudiantes.

1.8.3 Impacto ecológico

El desarrollo de este sistema contribuirá de manera positiva al medio ambiente al reducir los riesgos de pérdidas, daños y manipulación inadecuado de los equipos. Esto evita la compra innecesaria de nuevos dispositivos. Además, al implementar un control automático y en tiempo real sobre el uso de los equipos, se garantiza una gestión eficiente de los recursos tecnológicos y la protección de los trabajos académicos de los estudiantes.

CAPÍTULO II

2 MARCO TEÓRICO DE INVESTIGACIÓN

En este proceso, se presentará detalladamente la información recopilada de diversas fuentes. Este análisis es fundamental para profundizar el tema central del proyecto, que se enfoca en el desarrollo de un Sistema Informático con Redes Neuronales para la Administración del Laboratorio 01 de Computación.

2.1 Antecedentes históricos

2.1.1 Evolución de los sistemas informáticos

Revisando la documentación en libros, se encontró que la evolución de los sistemas informáticos comenzó en 1837 con la primera máquina mecánica, el motor analítico de Charles Babbage. Aunque esta máquina nunca terminó por problemas económicos en la vida de Babbage, en 1910 s u hijo menor completó parte de la máquina y logró realizar cálculos básicos. Alan Turing, en 1930, con el desarrollo de la máquina electrónica desempeño un rol importante en la teoría de la computación. En 1939, Arthur Scherbius creó la máquina Enigma durante la Segunda Guerra Mundial para que los ejércitos alemanes pudieran comunicarse (Boyle, 2021).

En la década de 1940, los primeros ordenadores empleados con válvulas de vacío se orientaban a aplicaciones científicas y militares. Mark I (1944) fue importante en la construcción por Howard Aiken en colaboración con International Business Machines (IBM), basado en el diseño de la Máquina Analítica de Carles Babbage. Aunque su diseño era más parecido al de una calculadora se le considera el primer ordenador electromecánico. En 1946, se desarrolló la ENIAC (Electronic Numerical Integrator and Computer) el primer ordenador electrónico general, representando un avance importante en el área informática. Sin embargo, se presentaron problemas con el software causado a la contante fusión de los tubos de vacío (Vasconcelos , 2018).

A partir de 1951, con la aparición del UNIVAC I y el UNIVAC II, se dio inicio a la era de los ordenadores comerciales. Durante este periodo también surgieron los primeros conceptos de sistema operativo para la gestión de las máquinas y los primeros lenguajes de programación como el lenguaje ensamblador, Fortran (1957) y LISP (1958), permitiendo que el desarrollo de software sea más flexible. En 1960 y 1965, los ordenadores experimentaron

una transformación significativa al volverse más pequeños, económicos y eficiente al uso de transistores en lugar de válvulas de vacío, los transistores generaban menos calor. Durante esta década surgió el concepto de periférico y el nacimiento de los lenguajes de programación como Algol y Cobal. Además, "el programa se le introduce a la máquina y esta lo traduce en un lenguaje que pueda interpretar, es decir, el lenguaje máquina". En 1966 y 1971 se introduce los circuitos integrados o chips, que contenía transistores miniaturizados, permitiendo la reducción de tamaño y coste de los ordenadores. Estos impulsaron el crecimiento de los ordenadores comerciales aumentado sus ventas. En esta época, también se produjo un avance importante en los lenguajes de programación con la aparición de BASIC, PL/1, C, Pascal y entre otros (Ramos, 2010).

Entre 1971 y 1981, con la integración de componentes electrónicos dio lugar al desarrolló del microprocesador, un procesador implementado en un solo circuito integrado. Este marcó el nacimiento del concepto de ordenador personal (PC), lo que permitió que sean doméstico y profesional. Durante este periodo, surgieron los sistemas operativos como CP/M y Unix, este último precursor del actual Linux. Además, se desarrolló el lenguaje PROLOG, se destacó domo una de las primeras herramientas orientadas a la inteligencia artificial (IA). En 1981, IBM lanzó al mercado la primer PC, lo que marcó en la historia de la informática al establecer un estándar que se fue mejorando y evolucionando con el tiempo. Esta PC utilizaba el sistema operativo MS-DOS, operaba con modo texto y desarrollado por Microsoft. Posteriormente, la interfaz gráfica se popularizó con el lanzamiento de Windows, Microsoft y después otros sistemas operativos como Linux. Se desarrolla la programación orientada a objetos (POO) y lenguajes como Java, C++, C# y Delphi, entre otros (Alegre Ramos, 2019).

2.1.2 Evolución de las redes neuronales

En las redes neuronales se encontró que, como posible inspiración para el concepto de la computación. Sin embargo, los primeros en basarse en la computación neuronal fueron Warren McCulloch, neurofisiólogo, y Walter Pitts, matemático, quienes en 1943 presentaron una teoría sobre las neuronas. Propusieron un modelo simple basado en circuito eléctricos, donde las neuronas eran tratadas como dispositivos binarios con umbrales fijos, siendo este el primer modelo matemático de una neurona. En 1949, Donald Hebb introdujo la idea del aprendizaje no supervisado en su libro The Orgazation of Behavior. Su propuesta sugirió que el aprendizaje ocurriera cuando se activaban ciertos cambios en unas neuronas, intentando relacionar el proceso de aprendizaje con la actividad nerviosa. Estas ideas sentaron las bases

de lo que más tarde sería la Teoría de las Redes Neuronales. Además, en 1959, Karl Lashley, concluyó que la información no se almacenaba de manera centralizada en el cerebro, sino que distribuía a lo largo de diversas áreas de este (Narváez y Vizcaya, 2016).

En 1958, Rosenblatt desarrolló el modelo conocido como Perceptrón, que reavivó el interés en el diseño de redes neuronales. Este modelo consiste en una única capa de procesamiento, encargada de manejar los datos proporcionados por las neuronas para generar una salida de la red neuronal. La red neuronal más antigua que se utiliza hoy en día en aplicaciones como reconocimiento de patrones. En 1960, se desarrolló el modelo Adaline (ADAptive LlNear Elements) por Bernard Widrow y Marcian Hoff una red neuronal orientada solucionar problemas del mundo real, como eliminación de ecos en las líneas telefónicas. Un año después, en 1961, Karl Steinbuch presentó Lernmatrix una red neuronal de diseño sencillo y eficiente para aplicaciones técnicas. En 1969, Minsky y Papert publicaron un estudio en el que destacaron las limitaciones del Perceptrón para resolver problemas complejos, especialmente aquellos que no son linealmente separables. Los autores ampliaron el concepto del Perceptrón de una sola capa hacia las redes neuronales multicapa (Flórez y Fernández, 2008).

En 1974, Paul Werbos presentó la idea básica del algoritmo de aprendizaje de propagación hacia atrás (backpropagation), posteriormente se transformaría en uno de los métodos más importantes para entrenar redes neuronales. Con el paso del tiempo la investigación en redes neuronales continuó avanzando. En 1976, Steve Grossberg y Gail Carpenter presentó la Teoría de la Resonancia Adaptativa (ART), Anderson y Kohonen en 1982, desarrollaron técnicas para el aprendizaje asociativo. En 1984, John Hopfield desarrolló una forma de red neuronal inspirado en conceptos energéticos, conocida como la red de Hopfield. En 1882, Paul Werbos implementó el método de aprendizaje backpropagation, que más adelante se consolidaría como el algoritmo preferido para entrenar redes multicapa. Luego, en 1986, David Rumelhart y Geoffrey Hinton descubrieron y mejoraron este algoritmo, revitalizando el interés en las redes neuronales. En el año 2006, el avance en el entrenamiento de redes neuronales permitió superar los desafíos en el procedimiento de datos, dando lugar al aprendizaje profundo (Deep Learning, DL). En 2012, las redes neuronales experimentaron una transformación importante con la mejorar en el reconocimiento de las imágenes, superando el rendimiento de los métodos anteriores convencionales (López, 2021).

En 2014, se establecieron las Redes Generativas Antagónicas (GAN) como un modelo de aprendizaje profundo con la capacidad de generar imágenes sintéticas. Su estructura se basa de dos redes neuronales profundas que trabajan conjuntamente en un generador y un discriminador. El generador crea nuevas instancias de datos sintéticos, mientras que el discriminador se encarga de distinguir entre los datos 'reales', provenientes del conjunto de entrenamiento y los datos 'falsos'. Ambas redes compiten entre sí, mejorando mutuamente el generador aprende a producir datos más realistas para engañar al discriminador se capacita para ser más exacto al distinguir los datos reales de los sintéticos (López, 2023)

2.2 Antecedentes de investigación

Las investigaciones recientes evidencia que las redes neuronales han adquirido problemas de la vida real han sido amplias con el pasar de los años. Estas van desde la educación, a los negocios e incluso en la rama de la medicina (Coello, 2021). Aborda el avance continuo de las redes neuronales, resaltando la implementación de nuevas formas para simplificar su uso y entendimiento por los individuos, su aplicación mediante la intervención humana demostró cómo las redes neuronales han progresado rápidamente en el desarrollo de sistemas inteligentes en el presente

Esta operación de convolución sobre una imagen puede detectar características diferentes según sean los valores del filtro que se defina. Los valores de este filtro serán los valores que la red neuronal irá aprendiendo poco a poco para poder realizar la tarea de detección (Moreno, 2023). Las redes neuronales convencionales son esenciales para la detección de imágenes, estas realizan una operación matemáticas llamada convolución, en la cual se aplican filtros que permiten identificar características específicas en las imágenes. La red neuronal aprende estos filtros durante el entrenamiento y constituye uno de los primeros pasos fundamentales en la detección.

Las redes convolucionales son las redes neuronales artificiales que se emplean en resolver múltiples problemas prácticos que requieren procesar imágenes (García, 2023). Estas redes tienen la capacidad de procesar las imágenes para detectar e identificar características específicas, también identifica los tipos de objeto en movimiento dentro de la imagen y ayudar a localizarlo con precisión dichos objetos.

Estos proyectos de titulación son parecidos utilizan redes neuronales y se centran en la detección de objetos, enfocándose en la identificación como las imágenes. Las redes neuronales

son esenciales para la identificar patrones y características visuales que cada proyecto que cuenta con un sistema para el desarrollo de aplicación. Sin embargo, las diferencias radican en el tipo de redes neuronales y los objetivos específicos de cada, algunos utilizan redes convolucionales un enfoque en la detección y clasificación de objetos. Un proyecto se centra en el análisis no lineal de estructuras, otro en la prevención de riesgos en el hogar para niños, y otro en la detección y clasificación de residuos. El presente proyecto se enfoca en la implementación de un sistema de seguridad para proteger los equipos en el laboratorio de cómputo 01, un ambiente cerrado destinado principalmente a actividades académicas. Los objetos para proteger dentro del aula son limitados incluyen 22 monitores con sus respectivos teclados, mouse, cables y CPU.

2.3 Definiciones conceptuales

Un sistema informático se define como un conjunto de componentes interrelacionados que permiten capturar, almacenar y procesar datos, basado en los principios fundamentales de la computación. Los principales componentes de un sistema informático incluyen tres partes esenciales como un componente físico (hardware), componente lógico (software) se dividen en dos tipos un software de base y software de aplicación, y un componente humano (Raya et al., 2014).

La red neuronal es "un nuevo sistema para el tratamiento de la información, cuya unidad básica de procesamiento está inspirada en la célula fundamental del sistema nervioso humano: la neurona" (Narváez y Vizcaya, 2016). Las redes neuronales son modelos de computación basados en la estructura del cerebro humano, diseñadas para procesar información y aprender a partir de los datos. Están compuestas por nodos o neuronas, organizas en capas y conectadas entre sí.

2.3.1 Sistemas Informáticos (programación)

El sistema informático (SI) es una estructura compuesta por dispositivos interconectados, tanto física y lógicamente, a través de canales que permiten la comunicación local o remota. Basado en programación, se refiere a un conjunto de hardware y software diseñada para ejecutar tareas específicas. Está gestionado por una unidad central de procesamiento (CPU), que tiene la capacidad de procesar datos y ejecutar algoritmos para cumplir con los requisitos establecidos (Moreno y Gonzáles, 2014).

2.3.1.1 Componentes de un sistema informático

2.3.1.1.1 Sistema informático en hardware

Los elementos físicos del SI, o conocido como hardware incluyen terminales, canales y soportes de la información que se constituyen por dispositivos electrónicos y electromecánicos que permiten la capacidad, procesamiento, almacenamiento y presentación de información. Se encuentra en los dispositivos de sensores, unidades de procesamiento, dispositivos de almacenamiento, monitores y entre otros.

2.3.1.1.2 Sistema informático en software

El software es el soporte lógico y equipamiento de un SI, ayudan al componente no tangible y no físico a realizar tareas específicas en el programa. Para ejecutar las tareas, el hardware se comunica con el software, formando un conjunto de programas de cómputo, procedimientos, reglas, documentación y datos asociados (Moreno y Ramos, 2014).

El ciclo de vida del software se centra en detectar errores lo antes posible, permitiendo que los desarrolladores se enfoquen en la calidad del software en los plazos de implantación y costes asociados, como se ilustra en la siguiente figura:

Figura 3 Ciclo de vida del software

2.3.1.1.3 Componente humano

El componente humano se refiere al personal informático, que incluye tanto a los técnicos encargados de la gestión de los SI, el mantenimiento y la atención a los usuarios. Este personal es responsable del diseño, desarrollo, implantación y operación de los SI, es importante tanto en el proceso de desarrollo de los sistemas como en su uso cotidiano. Este componente esta agrupado junto al hardware y software, como se muestra en la siguiente figura.

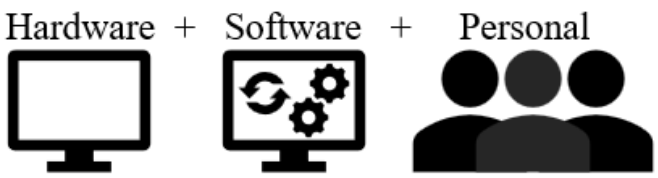


Figura 4 Componente del SI

Los SI están diseñados para realizar acciones destinadas a mejorar su funcionamiento. Entre estas se encuentra la inicialización del sistema, que es una parte fundamental del proceso, la ejecución de instrucciones del programa para cumplir con las tareas programadas, la comunicación del ordenador con el exterior a través de periféricos y el almacenamiento de información, garantizando así la disponibilidad de datos cuando sea necesario (Campos, 2023).

El SI se encargado principalmente del procesamiento de la información y de su almacenamiento cuando es necesario. Es fundamental reconocer que el SI está compuesto por diversos sistemas clave para su correcta ejecución. A continuación, se presentarán:

- Ordenadores personales
- Servidores
- Sistemas embebidos
- Superordenadores

2.3.2 Redes Neuronales

Las redes neuronales son modelos de algoritmos diseñados para realizar tareas cognitivas como el aprendizaje y la optimización. Estos modelos están basados en conceptos amplio y derivados de la investigación sobre la naturaleza del cerebro. Este compuesto por nodos conectados entre sí a través de enlaces (Strickland et al., 1995).

Las redes neuronales son entrenadas a identificar los comportamientos de los datos con la capacidad de reconocer patrones, resultados y otras características. Lo que permite implementar relaciones significativas entre los datos, siendo importante en las capaz de identificar las relaciones de datos.

2.3.2.1 Redes neuronales convolucionales

Las redes neuronales convolucionales, conocidas como CNN (Convolutional Neural Network) o ConNet, son ampliamente utilizadas en diversos problemas de aprendizaje profundo siendo especialmente eficaces en el reconocimiento de objetos, la segmentación, la detección y aplicaciones de visión por computadora por su arquitectura. Estas redes aprenden directamente de los datos de imagen sin necesidad de intervención manual para la extracción de características. Este enfoque ha demostrado un rendimiento sobresaliente en la visión por computadora así destacándose por su capacidad para aprender patrones complejos a partir de imágenes. Desarrolladas para emular el modo en que el cerebro humano procesa visualmente el entorno, las CNN aprovechan el campo receptivo, lo que les permite aumentar su

profundidad y capturar información a gran escala de manera más eficiente. Las CNN presentan tres aspectos importantes como su arquitectura elimina la necesidad de extraer características manuales del aprendizaje de los datos, su creciente popularidad se debe a los resultados que ofrecen en las tareas de reconocimiento de objetos y la flexibilidad permitiendo reentrenarla fácil en múltiples tareas, esencial en diferentes aplicaciones para su construcción de redes neuronales profundas (Millstein, 2018).

2.3.2.2 Capas de las redes neuronales convolucionales

David H. Hubel y Torsten Wiesel evidenciaron la presencia de conjuntos de neuronas en la corteza visual que poseen un campo receptivo local. Este campo se caracteriza por la zona limitada del espacio a la que responden estos grupos de neuronas. Las CNN se utilizan para reconocer patrones de imágenes o videos aplicando diferentes tipos de capas como la convolucional, agrupamiento (pooling) y la capa completamente conectada (full connected), lo que permite realizar las tareas avanzadas de clasificación y reconocimiento visual.

2.3.2.2.1 Capa de entrada

Esta capa de entrada o Input Layer, se encarga de recibir las imágenes como datos de entrada. Para imágenes en escala de grises, como los dígitos escritos a mano normalmente suele tener un tamaño de 32x32x1, en caso de las imágenes de color el tamaño es de 224x224x3, donde los tres canales representan los colores RGB.

2.3.2.2.2 Capa convolucional

La capa convolucional (Convolutional Layer) es fundamental en la construcción de una CNN, por lo que lleva a cabo la mayor parte de los cálculos. A través de las capas el modelo analiza cada imagen para extraer cada píxel de entrada lo que muestra esta neurona la conectividad a una región especifica de la imagen conocida como campo receptivo. La red permite centrarse en pequeños detalles o patrones las cuales se combinan para generar una representación más compleja en las capas siguientes, como se ilustra en la Figura 1.

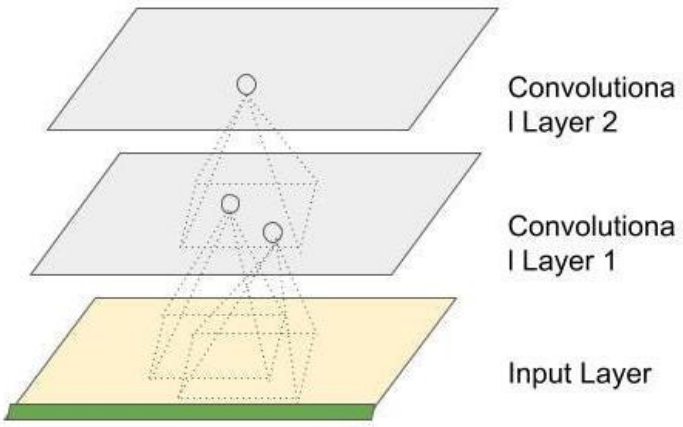


Figura 5 Capas convolucionales. (Manchanda, 2022)

El campo receptivo se desplaza a lo largo de la imagen, aplicando un conjunto de operaciones que capturan información en cada movimiento. Como resultado se genera un único valor por cada región procesada, se aplica una función de activación para transformar estos valores y proporcionar una salida más adecuada para las siguientes capas de la red. La función de activación ReLU (Rectified Linear Unit) se encarga de transformar los valores de entrada de forma que solo los valores positivos se mantengan en cada imagen proporcionada a la red, ReLU devuelve el valor máximo entre el valor de entrada y cero, lo que significa que cualquier valor negativo se convierte en cero, mientras que los valores positivos se conservan. Cuantos mayores sean estos valores, mayor será la similitud con la característica que se busca detectar. (Cicero, 2018). Como se ilustra en la Figura 2.

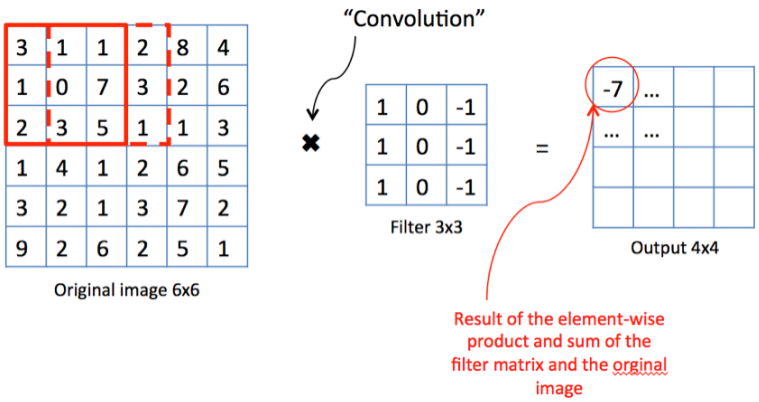


Figura 6 Ejemplo de operación de convolución. (Cavaioni, 2018)

2.3.2.2.3 Capa de Agrupamiento

Esta capa de agrupamiento o Pooling Layer un mecanismo fundamental para reducir la dimensionalidad de los mapas de características generados por las capas convolucionales. Esta capa divide la imagen en subregiones sin superposición y selecciona el valor máximo de cada

una. Su función principal es reducir las dimensiones espaciales, lo que disminuye el número de parámetros y el esfuerzo computacional de la red, mejorando así su eficiencia (Torres, 2020).

2.3.2.2.4 Capa de activación

La capa activación (Activation Layer) se define como la función que determina la salida de un nodo a partir de una o varias entradas. Las funciones de activación más utilizadas incluyen la Unidad Lineal Rectificada (ReLU), la tangente hiperbólica y función sigmoide. En el ámbito del aprendizaje profundo, ReLU es frecuentemente preferida, ya que acelera significativamente el proceso de entrenamiento de la red neuronal (Haque, 2024), se ilustra en la Figura 3.

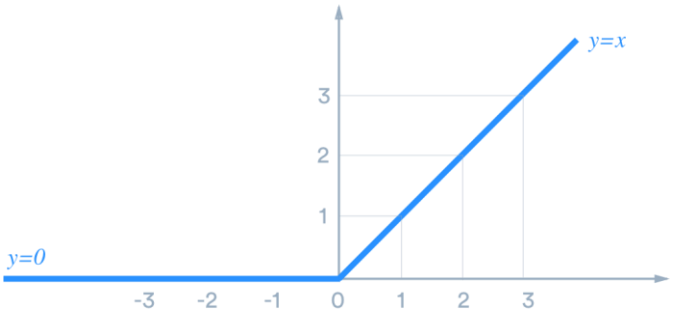


Figura 7 ReLU. (Liu, 2017)

2.3.2.2.5 Capa completamente conectada

La capa completamente conectada (Fully Connected Layer), también conocida como Capa Densa, se encuentra después de varias capas de convolucionales y capas de agrupación en una red neuronal es donde se lleva a cabo el razonamiento de alto nivel. Las neuronas en esta capa están conectadas a todas las activaciones en la capa anterior de manera similar a las redes neuronales artificiales regulares (ANN).

2.3.2.2.6 Capa de Abandono

La capa de abandono (Dropout Layer) elimina algunos nodos de la red neuronal para evitar el sobreajuste, se usa en la mayoría de los tipos de capas generalmente después de la capa completamente conectada.

2.3.2.2.7 Capa de Salida

La capa de salida (Output Layer) es la última en una CNN, su número de salidas varía según la cantidad de clases en la tarea de clasificación. Su función principal es convertir la representación interna generada por capas anteriores en la predicción final (Ramírez, 2023). Como se ilustra en la Figura 4.

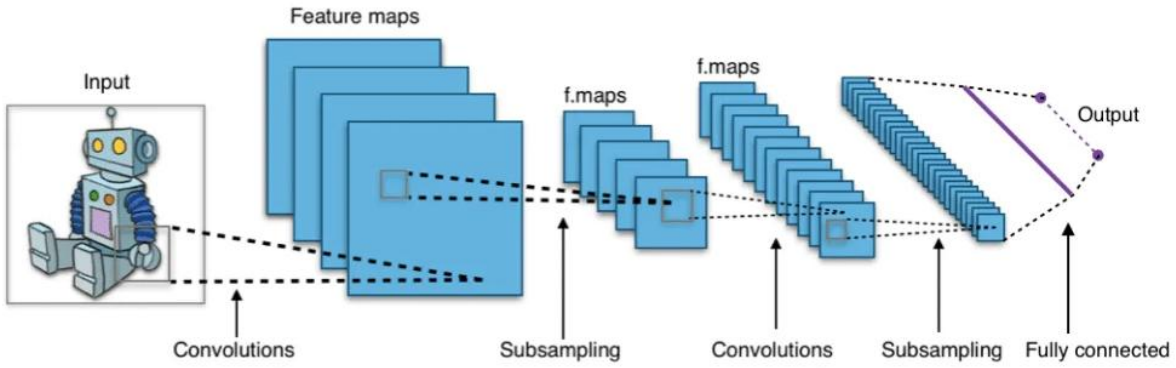


Figura 8 Arquitectura de una CNN. (Costa, 2019)

2.3.3 Administración de Centros de Cómputo

Según Terán (2014), la administración de centros de cómputo debe contar con personal altamente capacitado, especializado en su área de trabajo dentro de los SI, la preparación académica y la experiencia laboral son importantes para un excelente desempeño. De acuerdo con la organización el personal se asigna según sus capacidades y funciones, de manera que casa miembro desempeñe su rol. Por ejemplo, los ingenieros de sistemas de cómputo son expertos en el ámbito informático con conocimiento sólidos, los programadores se especializan en el desarrollo de aplicaciones y programación.

Para una administración adecuada del centro de cómputo es fundamental contar con recursos valiosos proporcionado por el departamento o área específica de trabajo, a continuación, se describen dos aspectos claves:

A) Mantenimiento

- Llevan un registro detallado de fallas, problemas detectados, soluciones aplicadas, acciones correctivas implementadas, respaldo, recuperaciones y trabajos realizados.
- Establecer y cumplir con una fecha de mantenimiento regular para asegurar los equipos.

B) Seguridad

- Implementar y cumplir las normas, reglamentos y procedimientos de seguridad previamente establecidos.
- Garantizar al personal que cumpla con las políticas de seguridad para prevenir incidentes.

2.3.3.1 Mantenimiento del equipo de cómputo

El mantenimiento en los SI y los equipos de cómputo es esencial para asegurarse de que todo funcione correctamente y dure más tiempo, el mantenimiento permite que los sistemas operen sin problemas y mejore su capacidad para ejecutar programas. Se divide en tres tipos principales como el mantenimiento predictivo, preventivo y correctivo. Además, se incluyen otros tipos como el mantenimiento de emergencia y adaptativo.

2.3.3.1.1 Mantenimiento predictivo

El mantenimiento predictivo es una técnica enfocada en anticipar cuándo un componente del sistema podría fallar, permitiendo tomar una decisión correcta ante un problema y se puede verificar el estado del sistema sin detenerlo con herramientas de diagnóstico.

2.3.3.1.2 Mantenimiento preventivo

El mantenimiento preventivo implica una serie de técnicas y procedimientos que se aplican para prevenir y minimizar el riesgo de fallo en el sistema, el mantenimiento es más frecuente e importante tomando medidas prevenciones con los componentes del sistema. Se analiza un análisis detallado y se dedica tiempo para esta evaluación. Además, este mantenimiento consiste en realizar las labores de mantenimiento de manera programada y periódica para evitar imprevistos (Cabello et al., 2002).

2.3.3.1.3 Mantenimiento correctivo

El mantenimiento correctivo se encarga de detectar, analizar y solucionar fallos en el sistema, estos fallos pueden deberse a problema de software, como errores en los drivers, sistemas operativos o configuración incorrecta. Además, también puede haber fallos hardware, como problemas en el disco duro, la placa base y entre otros.

2.3.3.1.4 Mantenimiento de emergencia

El mantenimiento emergencia se lleva a cabo en circunstancias criticas cuando el mantenimiento preventivo no es posible en algunos casos se ejecutan funciones o procesos para preservar el sistema informático. Además, estas representaciones tienen como objetivo corregir los fallos planificados y prevenir fallos inesperados asi evitando que el sistema deteriore o cause un impacto significativo en los equipos.

2.3.3.1.5 Mantenimiento adaptivo

El objetivo de este tipo de mantenimiento es modificar el sistema para ampliar sus funciones o adaptarse a nuevos procesos sin problema. Generalmente, se adaptan en la tecnología que a lo largo del tiempo evoluciona más con las actualizaciones de software y este mantenimiento adaptivo ajusta los sistemas nuevos en su entorno o requisitos establecidos. (Moreno, 2014)

2.3.3.2 Seguridad informática

La seguridad informática asegura los recursos del SI, protegiendo la privacidad de los materiales y programas informáticos. Esto permite una protección integral de los equipos, previniendo problema y ofreciendo soluciones adecuadas. A continuación, se explicará la seguridad física y lógica.

2.3.3.2.1 Seguridad física

La seguridad física "consiste en la aplicación de barreras físicas y procedimientos de control, como medidas de prevención y contramedidas ante amenazas a los recursos e información confidencial" (Costas, 2010, p. 50). Este método se utiliza para proteger el hardware como los datos almacenados, aplicando las medidas de seguridad correspondientes para cada equipo.

Estas amenazas pueden ser provocadas por el ser humano como daños en los discos duros, robos, destrucción de información o equipos del hardware, y sabotajes internos o externos. También incluyen desastres naturales como como los cortes de luz, incendios, tormentas e inundaciones, así como disturbios.

2.3.3.2.2 Seguridad lógica

La seguridad lógica se encarga de proteger el software, especialmente el sistema operativo y las aplicaciones sin perder ningún elemento fundamental. Su objetivo es proteger la información y datos de los usuarios mediante diversos métodos como copia de seguridad, contraseñas, cifrado de datos, permisos, antimalware, actualizaciones o filtros de conexión en aplicaciones de red. Estas medidas ayudan a garantizar una protección más segura para las tareas que se realicen (Costos, 2014).

2.3.3.3 Normas para la seguridad informática

Estas dos normas son importantes para la administración de centro de cómputo y sistemas informáticos que son directrices, reglas o especificaciones técnicas esenciales para garantizar servicios de calidad y productos excelentes. Estas normas facilidad el cumplimiento de las normativas y la gestión eficiente.

2.3.3.3.1 Norma ISO/IEC 27001

Esta norma es un estándar internacional para el Sistema de Gestión de Seguridad de la Información (SGSI), diseñado para proteger la disponibilidad, confidencialidad e integridad de la información. Ayuda a identificar y gestionar riesgos de seguridad de la información como desastres naturales, fraude, error del usuario y fallos en el sistema. A continuación, se presenta los conceptos relacionados con los atributos de un activo de información:

- **Disponibilidad:** Asegura que la información sea accesible y utilizable por los programas de software y los usurarios autorizados en cualquier momento que se necesite.
- **Confidencialidad:** No se revela datos asegurando la confianza a personas autorizadas.
- **Integridad:** Se encarga de proteger la exactitud e integridad de los activos de información (Calder, 2009).

2.3.3.3.2 Norma ISO/IEC 20000_1:2018

Esta normativa internacional dicta los requisitos para un Sistema de Gestión de Servicios (SMS), que se aplica en entidades que ofrecen servicios de TI de calidad y eficiencia. Estas normas son esenciales para un asegurar el cumplimiento, mantenimiento y mejora continua. A continuación, se presentarán unas fases y elementos necesarios de esta norma:

- **Alcance:** Define los servicios de gestión bajo el SMS, garantiza un manteniendo y mejorado continua de los servicios.
- **Referencias normativas:** Esta norma es independiente, aunque puede combinarse con otras normas.
- **Términos y definiciones:** Se agrupa en términos de organización, servicio, incidente, entre otros, asegurando coherentes en los requisitos.
- **Contexto de la organización:** Requiere de un entorno interno y externo de la organización y su capacidad de cumplir las necesidades.

- **Liderazgo:** La habilidad para guiar altos requerimientos por la organización estableciendo roles y responsabilidades definidas para la implementación y administración del sistema.
- **Planificación:** Involucra la planificación del SMS y gestiona los riesgos para establecer los objetivos de servicio y crear el propio SMS.
- **Soporte:** Un conjunto de requisitos que permiten gestionar la comunicación, competencia personal y la documentación del SMS sea adecuada.
- **Operación:** Detalla los procesos necesarios para la entrega y gestión continua de los servicios de TI (Van Der Haven, 2019).

2.4 Conclusiones relacionadas al marco teórico

Este sistema informático identifica los componentes clave para la implementación como es el físico, lógico y humano representando un desarrollo estable y óptimo. Cada uno de ellos desempeña un rol fundamental lo que garantizan la infraestructura, el procesamiento de datos y la administración adecuada del sistema.

Las redes neuronales convolucionales (CNN) son fundamentales para entrenar los modelos de clasificación y reconocimientos de imágenes, diseñado para el manejo de soluciones por el procesamiento de cada capa. Estas capas de salida integran las componentes procesadas en las capas anteriores dando resultado al procesamiento de los datos realizados por las CNN.

Al administrar el centro de cómputo es importante implementar medidas de seguridad como la norma ISO/IEC 27001 esta garantiza la protección de dispositivos, confidencialidad e integridad de la información resguardando el entorno y mejorando el espacio académico. Además, contribuye una mejora continua en el sistema informático asegurando que los requisitos se mantengan actualizados de manera eficiente.

CAPÍTULO III

3 MARCO INVESTIGATIVO (DISEÑO METODOLÓGICO)

3.1 Introducción

En este capítulo se presenta el tipo de investigación, las estrategias y el enfoque metodológico empleado, esta información fue recopilada a través de métodos como la encuesta y la entrevista que se aplicó como parte del proyecto, proporcionando una análisis detallado y estructurado en los resultados.

3.2 Tipo de investigación

3.2.1 Investigación bibliográfica

La información bibliográfica se centra en impulsar y apoyar la docencia e investigación, siendo crucial para respaldar cualquier estudio. La investigación bibliográfica es propedéutica, actúa como un estudio preliminar que prepara y facilita otras investigaciones para cumplir sus objetivos, esta información recopilada es valiosa y ayuda a estructurar de manera más profunda el desarrollo de los estudios o investigaciones (Bernal C. , 2003).

El capítulo 2, se utilizó la investigación bibliográfica para recopilar y organizar información en específico las redes neuronales convolucionales. Esto permitió explorar las redes para identificar los patrones y modelos aplicables a la protección de los equipos del laboratorio.

3.2.2 Investigación de campo

La investigación de campo es un trabajo académico que implica recolectar información de primera mano, ya sea mediante la grabación o documentación de lo que el investigador observa o escucha en un sitio especifico, como comunidades o barrios. Este tipo de investigación tiene como objetivo recopilar material que se analiza y utiliza en investigaciones originales, ensayos, libros, exposiciones o en páginas web (Bartis, 2004).

Las encuestas y entrevistas fueron esenciales en el presente proyecto de investigación cuantitativa para recolectar, procesar y analizar datos. Estas técnicas permitieron obtener información precisa de un entorno determinado, la cual se utilizó para analizar y presentar los

resultados. Además, fue crucial validar y asegurar la fiabilidad de los datos obtenidos, ya que los resultados ayudaron a tomar decisiones para las futuras acciones del proyecto.

3.2.3 Investigación aplicada

La investigación aplicada se centra en implementar y utilizar nuevos conocimientos con el objetivo de alcanzar las metas prácticas y proporcionar soluciones en un entorno problemático. Estos trabajos innovadores no solo resuelven problemas, sino que se encargan de buscar y consolidar el conocimiento para su aplicación práctica (Frascati, 2015).

En este procedimiento, la incorporación de una cámara que permitió un sistema de alertas fue fundamental para el monitoreo representó ventajas cruciales tanto para el administrador del laboratorio como para la Universidad. Esto permitió facilitar la obtención de resultados en tiempo real, simplificando la administración y recolección de datos sobre la manipulación de los equipos. Además, este tipo de investigación no solucionó únicamente un problema inmediato de seguridad, sino optimizó los recursos al permitir una supervisar constante.

3.3 Métodos de investigación

3.3.1 Analítico – Sintético

El método analítico consiste en descomponer todo en sus partes o componentes para estudiar y examinar cada uno de manera individual con el fin de identificar y comprender las relaciones entre ellos. Este proceso permite identificar el comportamiento de cada elemento por separado, descubriendo su funcionamiento interno y las interacciones que estos tiene entre sí. Por otro lado, el método sintético es proceso que une o relaciona hechos y elementos en un principio aislados, para integrarlos en un todo coherente. A través de este método reúne los distintos elementos procesados en la etapa de análisis para crear una visión unificada, permitiendo entender el sistema en su conjunto (Rodríguez, 2005).

Esta metodología facilitó la recopilación de información del sistema identificando patrones y analizar sus componentes, lo que garantiza una seguridad e interacción dentro del laboratorio. Cada estudio realizado de las redes destacaba la estructura y capacidad de desarrollar la detección de objetos. Con el avance la administración, seguridad y gestión incrementaron la confiabilidad del laboratorio.

3.3.2 Inductiva – deductivo

El método inductivo se enfoca en el razonamiento que parte de hechos particulares para llegar a una conclusión general, con el estudio individual de estos hechos para formular principios, normas o teorías de carácter general. Mientras que el método deductivo parte de principios generales o teorías universales para aplicar en casos particulares utilizando leyes, teorías universales o principios ya establecidos para analizar situaciones específicas, comprobada su validez en contextos particulares (Bernal, 2006).

Para el diseño del sistema de seguridad en el laboratorio, se analizaron los componentes específicos como la cámara que captura y los algoritmos de detección de objetos. El reconocimiento de imágenes se realizó con CNN lo que permitió implementar un sistema integral que proteja el laboratorio. En términos generales, la CNN demostraron ser amplias para la detección de objetos. Al aplicar esta teoría en el sistema de seguridad se abordó el monitoreo de la cámara de video, enfocándose en utilizar las CNN con una planificación concreta para mejorar la seguridad y gestión del laboratorio generando una alarma ante cualquier manipulación inadecuada de los equipos.

3.4 Fuentes de información de datos

3.4.1 Fuente primaria

Se recopiló información a través del método de la encuesta sobre el control de inventario, la protección de los equipos, protocolos de seguridad, monitoreo, el acceso restringido y tiempo de uso en el laboratorio. Los resultados son cruciales para realizar un análisis detallado para la medida de seguridad para la protección de los equipos especialmente en los CPU y monitores. Además, se entrevistó al administrador y laboratorista responsable, con el objetivo de conocer en profundidad el estado actual de los equipos, los procesos de gestión y control, los problemas frecuentes y las medidas de seguridad implementadas

3.5 Estrategia operacional para recolectar datos

Es fundamental obtener información precisa y relevante, por lo que usar estas herramientas adecuadas facilita la recopilación de datos estructurados. Este trabajo realizó una encuesta dirigida a estudiantes de software que usan más el laboratorio, para evaluar su nivel de conocimiento sobre el tema y proponer mejoras en el estado del aula. Además, se entrevistaron al administrador y laboratorista encargados del laboratorio, para conocer los

protocolos de seguridad y el funcionamiento del espacio, identificar áreas de mejora en la gestión y uso del laboratorio.

3.5.1 Población – Segmentación – Técnica de muestreo – Tamaño de la muestra

3.5.1.1 Población

La población se define como el conjunto de personas, objetos, animales, muestras biológicas, expedientes, organización y entre otros, que se desea estudiar en un contexto determinado. Esta población constituye al universo del cual se seleccionará una muestra para llevar a cabo el estudio, establece una población definida, limitada y accesible que formará los resultados obtenidos de la muestra permitiendo generar conclusiones y extrapolar información relevante que contribuirá al objetivo de la investigación (Arias et al., 2016).

En el laboratorio 01, se compuso por 199 personas entre estudiantes y profesores que formaban parte del horario más habitual de varias carreras y semestres de TI y Software, mientras los profesores impartían diferentes asignaturas en estas áreas.

3.5.1.2 Segmentación

La segmentación consiste en dividir una población en grupos más pequeños y homogéneos, identificados por características comunes. Este método es útil para estudiar distintos aspectos de una población o aplicar estrategias específicas para cada segmento, los elemento o unidades que forman cada segmento son relevantes y significativos desde el interés de la investigación, permitiendo establecer una unidad de análisis en el estudio (Bautista, 2022).

En el presente trabajo, se recopiló información sobre el horario de los estudiantes y profesores que utilizaban el laboratorio 01. En algunos casos, los estudiantes de semestres inferiores eran reemplazados por niveles superiores, ya que los mismos estudiantes estaban en ambos niveles debido al arrastre de materias. En este caso, se evaluaron todas las materias que utilizaban el laboratorio para centrarse en el área de programación, ya que era la que más utilizaba el laboratorio. A continuación, se presentó la tabla en el periodo 2024(2) en el área de software.

Carreras	Materias y Semestres	N° Estudiante	Profesores
Software	Administración de BDD distribuido (Cuarto)	21	Ing. Danilo Arévalo

Carreras	Materias y Semestres	N° Estudiante	Profesores
Software	Arquitectura Software (Quinto)	20	Ing. Patricio Quiroz
Software	Minería de datos (Quinto)	25	Ing. Saed Reascos
Software	Interación humano computador (Quinto)	24	Ing. Jefferson Arca
	TOTAL DE PERSONAS	**90**	**4**

Tabla 1 Asignación de materias del Laboratorio 01

3.5.1.3 Técnica de muestreo

En el presente proyecto se utilizó el muestreo estratificado, ya que permitía estudiar diversos grupos o subgrupos, y recoger una muestra aleatoria de cada estrato. En este caso, los estudiantes y profesores se dividieron en distintos grupos para identificar patrones específicos dentro de cada uno lo que permitió obtener resultados más precisos y analizar de manera más detallada.

3.5.1.4 Tamaño de la muestra

La fórmula para calcular el tamaño de la muestra incluyó los siguientes elementos: N es el tamaño total de la población, Z representa el nivel de confianza, p es la proporción estimada y he es el margen de error. A continuación, se presenta la fórmula para una población finita, que permite obtener el tamaño de la muestra:

$$n = \frac{N \times Z^2 \times p \times (1-p)}{(N-1) \times e^2 + Z^2 \times p \times (1-p)}$$

En este caso, se utiliza los siguientes datos:

- Tamaño de la población de 94 personas (N=94)
- Nivel de confianza del 95% (Z=1.96)
- Margen de error del 5% (e=0.05)
- Proporción estimada (p=0.5)

Sustituyendo estos valores en la fórmula:

$$\boldsymbol{n} = \frac{94 \times 1.96^2 \times 0.5 \times (1 - 0.5)}{(94 - 1) \times 0.05^2 + 1.96^2 \times 0.5 \times (1 - 0.5)}$$

1) **Calcular Z^2:**

$$1.96^2 = 3.8416$$

2) **Calcular $p \times (1 - p)$:**

$$0.5 \times (1 - 0.5) = 0.25$$

3) **Sustituir los valores:**

$$n = \frac{94 \times 3.8416 \times 0.25}{(94 - 1) \times 0.05^2 + 3.8416 \times 0.25}$$

$$n = \frac{94 \times 0.9604}{93 \times 0.0025 + 0.9604}$$

$$n = \frac{90.2736}{0.2325 + 0.9604}$$

$$n = \frac{90.2736}{1.1929} \approx 75.66$$

El tamaño de la muestra calculado es aproximadamente 76, redondeado al número entero. Se seleccionaron un administrador, un laboratorista para la entrevista y 74 estudiantes de nivel superior para la encuesta.

3.5.2 Análisis de recolección de datos

3.5.2.1 Encuesta – Entrevista – Observación / Otras

En el presente proyecto, se optó utilizar una combinación de entrevistas y encuestas para recopilar información sobre el uso y seguridad de los equipos en el laboratorio de cómputo.

1. *Encuesta*
    - ***Detalle:*** Se aplicó un cuestionario estructurado con preguntas principales cerradas con el objetivo de cuantificar y generalizar las percepciones, hábitos y experiencias de una muestra.
    - ***Población:*** 74 estudiantes de nivel superior debido a su mayor experiencia y percepción acerca de la seguridad del laboratorio.
2. *Entrevista*
    - ***Detalle:*** Se realizaron preguntas abiertas que permitieron profundizar en aspectos cualitativos, obteniendo información detallada y contextualizada.
    - ***Población:*** Se seleccionaron un administrador y laboratorista responsable del área.

3.5.2.2 Estructura de los datos aplicados

Problema radica en la manipulación inadecuada de los equipos en el laboratorio, ya que no existe una supervisión constante. Esto podría derivar en consecuencias negativas, como daño o robos del aula.

Componentes	Preguntas abiertas	Preguntas cerradas
Causa: Falta de control efectivo sobre dónde se encuentran los equipos		
1. Control de inventario 2. Sistema de rastreo 3. Protocolos de registro	¿Cuál es el estado actual de los equipos del laboratorio? ¿Cómo gestiona el traslado de los equipos del laboratorio? ¿Qué procedimiento existe para controlar el uso del laboratorio?	¿Conoce del control de inventario de los equipos del laboratorio? • Si • No ¿Considera que los equipos del laboratorio están adecuadamente protegidos? • Totalmente protegidos • Poco protegidos • Nada protegidos ¿Existe un protocolo documentado para la seguridad en el laboratorio? • Si • No
Causa: Poca supervisión en el traslado de los equipos.		
1. Supervisión activa 2. Autorización de movimientos 3. Regulación de horarios	¿Qué problemas comunes ha encontrado al supervisar los equipos? ¿Cómo evalúa la efectividad de las medidas de	¿Recibe indicaciones específicas sobre la seguridad en el laboratorio? • Siempre • Frecuentemente • Ocasionalmente • Raramente • Nunca

Componentes	Preguntas abiertas	Preguntas cerradas
	seguridad actual? ¿Cómo se registra el uso de los equipos?	¿Con qué frecuencia ha notado movimientos en los equipos del laboratorio? • Muy frecuentemente • A veces • Rara vez • Nunca ¿Se respeta el horario de clases del laboratorio? • Siempre • Frecuentemente • Ocasionalmente • Raramente • Nunca
Causa: Equipos expuestos a posibles daños o robos por falta de seguridad.		
1. Sistema de alarma 2. Accesibilidad restringida 3. Monitoreo constante	¿Cuál es su criterio de un sistema de supervisión implementaría en el laboratorio? ¿Cuál ha sido la frecuencia observada de desconexión de equipos del laboratorio? ¿Qué mejoras implementaría	¿Se activa una alarma cuando ocurre algo con los equipos? • Si • No ¿Existen restricciones de acceso al laboratorio? • Totalmente restringido • Algo restringido • Poco restringido • Nada restringido ¿Cada que tiempo utiliza los equipos del laboratorio? • Siempre • Frecuentemente • Ocasionalmente

Componentes	Preguntas abiertas	Preguntas cerradas
	para optimizar la seguridad de los equipos?	• Raramente • Nunca

Tabla 2 Preguntas para encuestas y entrevistas

3.5.3 Plan de recolección de datos

Una vez establecidas la encuesta y la entrevista como técnicas principales de recopilación de datos, es fundamental garantizar el cumplimiento del proyecto mediante un cronograma. Este cronograma será presentado a la Universidad Laica Eloy Alfaro de Manabí Ext El Carmen. De esta manera, se asegura una recolección de datos estructurada y alineada con el objetivo del sistema propuesto, permitiendo una gestión y optimización eficiente.

CRONOGRAMA OPERATIVO			
Técnica	**Enfocado**	**Responsable**	**Fecha**
Encuesta	Estudiantes	Carranza Tifani	12 al 15 /11/2024
Entrevista	Administrador y Laboratorista.	Carranza Tifani	12/11/2024

Tabla 3 Cronograma de recolección de datos de encuesta y entrevista

3.6 Análisis y presentación de resultados

Para el análisis de estas técnicas aplicadas se implementaron el método estadístico descriptivo y el análisis cualitativo, con el propósito de optimizar las respuestas de los encuestados y entrevistados. Estos métodos fortalecieron los resultados con gráficos y texto facilitando una interacción más clara y precisa de las causas y necesidades del laboratorio 01.

3.6.1 Tabulación y análisis de los datos

3.6.1.1 Entrevista

Esta entrevista se llevó a cabo con el administrador y laboratorista de pasantía, sus respuestas fueron claras, exactas y relevantes brindados datos detallados sobre el estado actual del laboratorio.

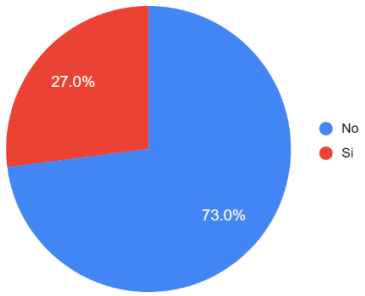
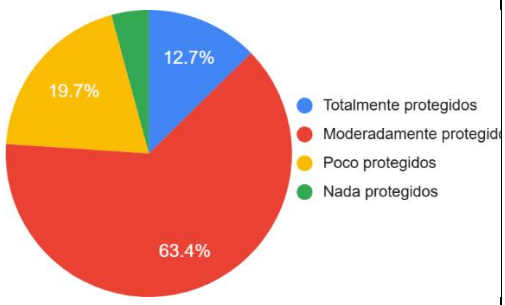
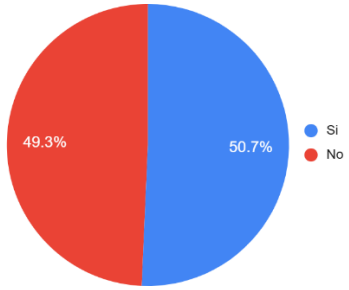
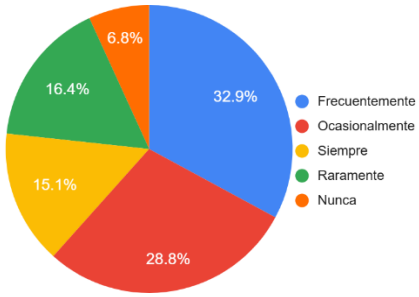
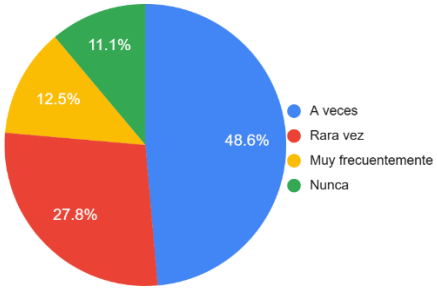
Preguntas	Administrador	Laboratorista	Análisis
1. ¿Cuál es el estado actual de los equipos del laboratorio?	Algunos equipos están en buen estado, pero los teclados y mouse, al ser los más manipulados, son propensos a fallas.	Los equipos están funcionando correctamente en el laboratorio.	En general, los equipos del laboratorio están funcionando correctamente, aunque los periféricos con el uso constante presentan mayor riesgo de fallas.
2. ¿Cómo gestiona el traslado de los equipos del laboratorio?	El encargado del departamento realiza el traslado de los equipos para mantenimientos correctivos e informa si están en revisión o fuera de funcionamiento.	Actualmente, no se permiten mover los equipos. Cuenta con un aviso y está asegurado con correas para impedir cualquier movimiento de cables.	Existe una restricción que impide el traslado de cualquier equipo, aunque el responsable administre el desplazamiento para un mantenimiento.
3. ¿Qué procedimiento existe para controlar el uso del laboratorio?	El control del uso se basa en el horario de clases semestral, sin un procedimiento específico para identificar.	No existe un control establecido para los equipos, por lo que sería necesario implementar uno.	Ambos coinciden que no se presenta un control específico del uso de los equipos.
4. ¿Qué problemas comunes ha encontrado al supervisar los equipos?	El movimiento de mouse, teclados, pantallas o CPU por los estudiantes ante fallas, además de la desconexión frecuente de cables de red.	La acumulación de polvo, cables dañados, periféricos en mal estado, puntos de red defectuosos, procesadores sin pasta térmica y disipadores desconectados.	Además de los problemas técnicos existentes en los equipos, las fallas provocadas por los estudiantes pueden agravar la inestabilidad del sistema.
5. ¿Cómo evalúa la efectividad de las medidas de seguridad actual?	Actualmente, una cámara graba los eventos, pero existe el riesgo de pérdida de información, ya que el	La efectividad es muy baja, ya que no existe un control adecuado en el laboratorio.	Diferentes perspectivas uno enfocado en el riesgo especifico de grabaciones y el siguiente de la

Preguntas	Administrador	Laboratorista	Análisis
	disco duro elimina las grabaciones antiguas.		manera general a la falta de control, coincidieron que requiere mejoras.
6. ¿Cómo se registra el uso de los equipos?	No se tiene un registro específico. En el pasado, se utilizaba una ficha de uso general.	Actualmente, no se lleva ningún registro del uso de los equipos.	Coinciden que no tienen un registro, pero se diferencia en el enfoque temporal.
7. ¿Cuál es su criterio de un sistema de supervisión implementaría en el laboratorio?	Considero que sería muy positivo, ya que permitiría recibir alertas sobre las novedades y realizar una revisión inmediata de los problemas que se presenten.	Sería una medida ideal para el control adecuado, asegurando el mantenimiento y la prevención de problemas de desconexión como de robos.	Ambos tienen la importancia de mejorar el control y la gestión de los equipos.
8. ¿Cuál ha sido la frecuencia observada de desconexión de equipos del laboratorio?	Al principio de este semestre, la desconexión era muy frecuente. Luego, se indicó a los estudiantes, lo que ha mejorado la situación, aunque no de la manera más eficiente.	Se ha observado que el 100% de los equipos han estado desconectados en algún momento.	Abordan el problema de desconexión, pero diferentes enfoques afirmando que todos los equipos han estado desconectados y el primero señala la acción correctiva.
9. ¿Qué mejoras implementaría para optimizar la seguridad de los equipos?	Sería útil implementar un sistema que evite esta situación o que supervise los equipos.	Implementaría un sistema para gestionar y controlar de manera lógica el acceso y uso de los equipos.	Ambos implementan una idea de sistema para mejorar el control con enfoques diferentes.

Tabla 4 Tabulación y análisis de los datos de entrevista

3.6.1.2 Encuesta

La encuesta se dirigió específicamente a 74 estudiantes de la carrera de Software de la Universidad ULEAM Extensión El Carmen, los principales usuarios de los equipos de cómputo.

Pregunta	Gráfica	Análisis
1. ¿Conoce del control de inventario de los equipos del laboratorio?		En su gran mayoría los estudiantes consideran que no conoce del control de inventario, lo que sería importante reforzar la información a todos los estudiantes sobre este procedimiento.
2. ¿Considera que los equipos del laboratorio están adecuadamente protegidos?		La mayoría considera que los equipos son moderadamente y poco protegido.
3. ¿Existe un protocolo documentado para la seguridad en el laboratorio?		La mitad consideran que si existe un protocolo de seguridad.
4. ¿Recibe indicaciones específicas sobre la seguridad en el laboratorio?		La gran mayoría considera que frecuente se recibe indicaciones de seguridad del laboratorio.
5. ¿Con qué frecuencia ha notado movimientos en los equipos del laboratorio?		La gran mayoría considera que a veces han notado movimientos de los equipos.

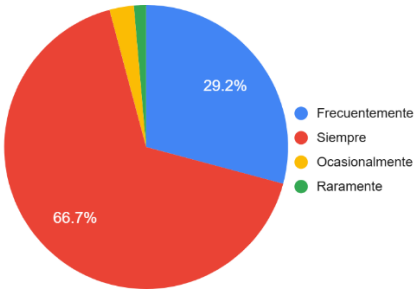
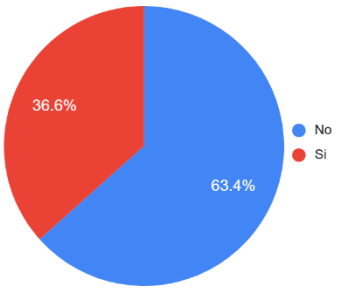
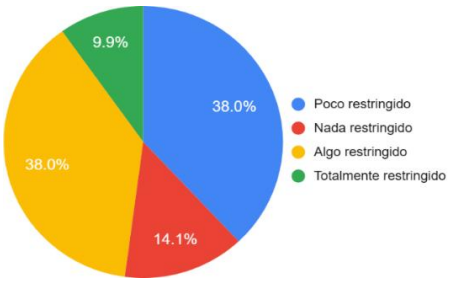
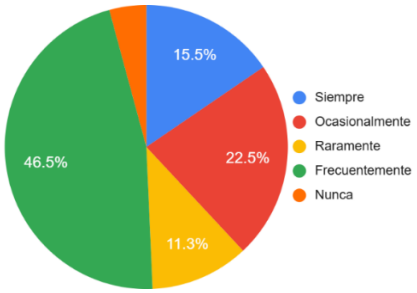
Pregunta	Gráfica	Análisis
6. ¿Se respeta el horario de clases del laboratorio?		La mayoría considera que siempre se respeta el horario de clases.
7. ¿Existe algún mecanismo de aviso cuando ocurre algo con los equipos?		La gran mayoría considera que no se presenta ningún mecanismo cuando ocurre una novedad.
8. ¿Existen restricciones de acceso al laboratorio?		La gran mayoría considera que poco y nada restringido se encuentra el acceso del laboratorio.
9. ¿Cada que tiempo utiliza los equipos del laboratorio?		La mayoría consideran que frecuente utilizan los equipos.

Tabla 5 Tabulación y análisis de los datos de encuesta

3.6.2 Presentación y descripción de los resultados obtenidos

3.6.2.1 Análisis de resultados

- En la pregunta 2 un total de 19.7% se considera que los equipos no están correctamente protegidos.
- La pregunta 5 con un total 61,1% la gran mayoría considera que el movimiento de equipos ocurre a veces y frecuentemente.
- La pregunta 4 con un total de 61,7%, considera que las indicaciones de seguridad en el laboratorio se proporcionan frecuente y ocasionalmente.

3.6.2.2 Interpretación de resultados

- Del análisis realizado se desprende que un porcentaje considerable de estudiantes percibe una falta de protección de los equipos, lo que evidencia la necesidad de implementar el sistema. Además, la entrevista (pregunta 3) refleja la ausencia de un mecanismo de control para la protección de los equipos del laboratorio, ratificando la existencia de la causa: ineficiente control de la ubicación de los equipos físicos.
- Por otro lado, los datos indican que la mayoría de los casos reportan movimientos frecuentes de equipos, una situación de control. Según en la entrevista (pregunta 2), el traslado de los equipos es realizado por el encargado del departamento. Los datos permiten corroborar la existencia de la causa de la limitada supervisión en el traslado de equipos, ya que, aunque hay un encargado, no se refleja un control constante.
- Los participantes en la encuesta hacen referencia frecuente a las indicaciones sobre la seguridad del laboratorio. Como se muestra en la pregunta 1 de la entrevista, se señala que los equipos están en buen estado, pero son más propensos a fallos lo que provoca una necesidad de supervision más constante. Este análisis radica en la presencia de riesgos por pérdida de equipos causado de restricciones de control.

3.6.3 Informe final del análisis de los datos

Según los resultados de la encuesta sobre la protección del laboratorio 01 realizado en la Uleam Ext. El Carmen, el 19.7% de los participantes indicaron que las medidas de seguridad no están adecuadamente resguardadas lo que supone un peligro para el manejo de los equipos de mayor valor. Además, la entrevista se destaca la ausencia de un control adecuado y la urgencia de instalar un control que asegure una mayor protección para el laboratorio.

Se recolectaron datos de la encuesta y el 61,1% de los participantes indican el movimiento constante de los equipos, aunque en ocasiones se mueven de forma de constante. Por otro lado, la entrevista señala que no es posible trasladar los equipos sin la presencia de un encargado del departamento, aunque revela una limitación en la supervisión no es un control constante lo que podría afectar la gestión adecuada de los equipos.

La información proporcionada por los participantes de la encuesta, el 61,7% recibe indicaciones de seguridad con frecuencia y de manera ocasional. Aunque los equipos están bien, se consideran propensos a fallas, esta situación representa perdida de equipos, lo que es necesario mejorar los controles de seguridad.

CAPÍTULO IV

4 MARCO PROPOSITIVO

4.1 Introducción

En este capítulo se presentará la solución al problema de seguridad en el laboratorio de cómputo de la Universidad, se detallará cada aspecto de la iniciativa con el objetivo de fortalecer la protección de los equipos y minimizar los riesgos existentes. La finalidad es implementar medidas responsables y eficientes que garanticen una gestión óptima de la seguridad, asegurando la correcta aplicación de las estrategias planteadas en el proyecto.

4.2 Descripción de la propuesta

La presente propuesta plantea el desarrollo de un sistema de detección de objetos orientados a identificar los CPUs y monitores en tiempo real, con el objetivo de optimizar el control de inventario del laboratorio de cómputo. Para ello, se utilizarán CNN entrenadas con un conjunto de imágenes propias permitiendo reconocer dichos dispositivos de forma precisa.

La implementación de herramientas en el desarrollo del software busca mejorar la funcionalidad y accesibilidad del sistema para la clasificación y detección de objetos en tiempo real. Este ciclo de desarrollo se fortalece con las librerías de TensorFlow y YOLOv8. La gestión del etiquetado se realiza de forma manual para que el entrenamiento generando un modelo preciso y un diseño interactivo para el sistema.

4.3 Determinación de recursos

Este proyecto ha obtenido varios recursos esenciales en cada área lo que otorga una relevancia indispensable durante este proceso de desarrollo. Para una mejor compresión se especifican cada relevancia y operación dentro del proceso de desarrollo del sistema. A continuación, se muestra los medios implementados en el desarrollo del proyecto:

- Se utilizó un ordenador personal para el entrenamiento de los modelos y la implementación.
- Aplicaciones y entornos destinados al desarrollo del sistema.

- Diseño de interfaz web desarrollado para la visualización e interacción con el sistema.
- Materiales físicos necesarios para la implementación práctica del sistema.
- Librerías especializadas integradas para el procesamiento, detección y visualización.
- Horas de trabajo dedicadas a la planificación, ejecución y evaluación técnica.

4.3.1 Humanos

Estos recursos aportan distintas actividades, desde la planificación y recolección de información, hasta la validación y programación del sistema. A continuación, se detalla en la tabla cada especificación de los participantes que forman parte de este proyecto:

Cantidad	Nombre	Rol	Actividad	Tiempo estimado
1	Ing. Bladimir Mora	Coordinador de la carrera.	Participó en la entrevista como informante.	1 hora
1	Ing. Raúl Reascos	Tutor de tesis	Responsable del proceso de revisión.	16 horas
1	Tifani Carranza	Programadora	Encargada de la recolección de datos y del desarrollo del sistema de detección.	384 horas
74	Estudiantes	Encuestados	Se aplicaron encuestas a los estudiantes de niveles superiores de la carrera de software.	1 hora por estudiante
1	Kevin Prado	Laboratorista	Participó en la entrevista.	1 hora

Tabla 6 Determinación de recursos humanos

4.3.2 Tecnológicos

En cuanto al hardware se utilizó una computadora desde la cual el administrador podrá supervisar las alertas enviadas, así como una cámara de seguridad que capturará cada movimiento y permitirá el uso de imágenes en tiempo real.

En el apartado de software se implementó la librería TensorFlow y YOLOv8 a través de Google Colab, una herramienta en la nube que facilita la prueba de código. Además, se empleó Visual Studio Code para el diseño y la visualización del reconocimiento de imágenes en tiempo real. Finalmente, se dispuso de un servidor destinado al almacenamiento de los datos. Se presentó la siguiente figura como la estructura que conforma la arquitectura del sistema:

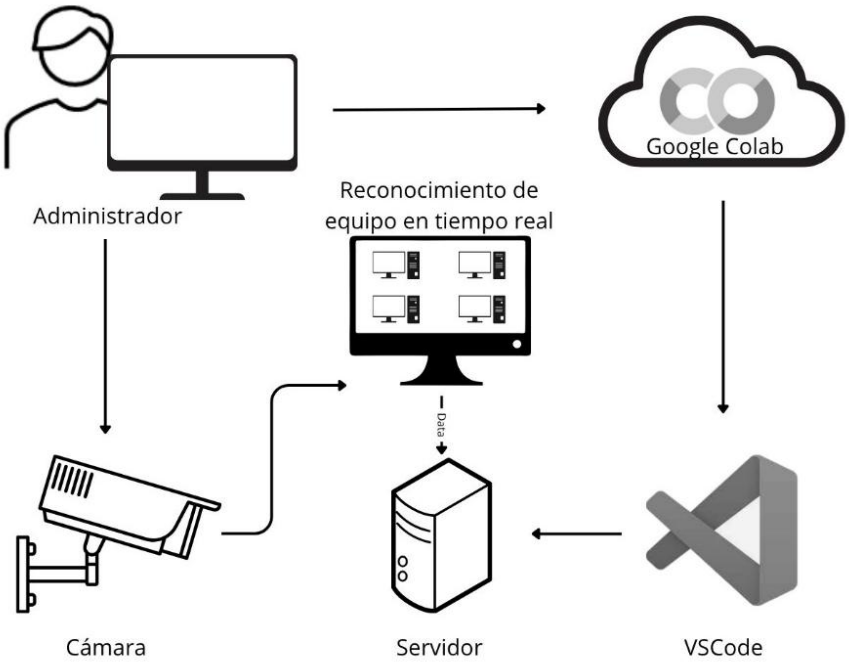


Figura 9 Arquitectura tecnológica del sistema propuesto

4.3.3 Económicos (presupuesto)

A continuación, se detallan los materiales y componentes adquiridos para el desarrollo e implementación del sistema propuesto. Estos elementos abarcan dispositivos de captura, almacenamiento, conectividad y soporte técnico, son fundamentales para garantizar la operatividad y confiabilidad del sistema propuesto.

Cantidad	Descripción	Precio U.	Subtotal
1	Cámara TP-Link VIGI C330I (2.8mm)	$70,00	$70,00

Cantidad	Descripción	Precio U.	Subtotal
1	Cable Ethernet de 30 metros	$0,55	$16,50
4	Batería Seca Recargable (12V 7Ah) First Power	$30,00	$120,00
1	Disco Duro Hp 627114-002 De 300gb Sas Para Servidor 2.5" Sff 6g 15k Hot Plug	$200,00	$204,78
2	Conectores RJ45	$0,50	$1,00
1	Cinta doble faz	$3,75	$3,75
1	Implementación de red local	$100	$100
384	Horas de trabajo (investigación, análisis, diseño, codificación, pruebas e implementación)	$10,00	$3,840
		Total	$519,87

Tabla 7 Determinación de recursos económicos

4.4 Etapas de acción para el desarrollo de la propuesta (software)

Scrum es un marco de trabajo ágil que permite enfrentar problemas complejos y cambiantes mientras se desarrollan soluciones de manera eficiente, creativa y con alto valor. Esta metodología promueve la colaboración entre los integrantes del equipo, promoviendo un entorno laboral activo donde alcanza los resultados de gran relevancia.

En esta etapa se aplicó la metodología ágil Scrum, basada en ciclos de trabajo cortos. El objetivo es asegurar entregas parciales del sistema y del documento mostrando un proceso iterativo de mayor eficiencia. Como se muestra en la siguiente figura, cada etapa debe de completarse permitiendo una integración efectiva y las necesidades del usuario.

Su propósito es asegurar las entregas parciales del sistema para mejorar el proceso de iteración. Como se ilustra en la siguiente figura, cada fase debe finalizarse de forma completa y facilitando una incorporación eficaz en el desarrollo con el propósito de mejorar y optimizar las necesidades de los usuarios.

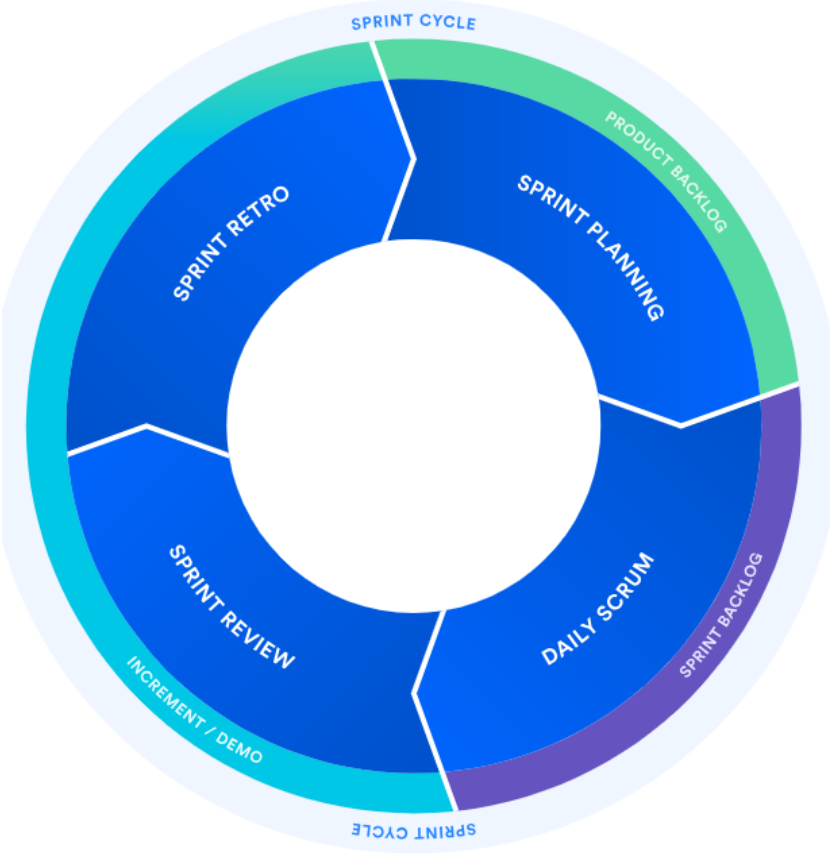


Figura 10 Metodología Scrum https://www.atlassian.com/es/agile/scrum/sprints

Mediante la metodología ágil Scrum se propone un sistema más flexible y enfocada en iterativo, una elección adecuada para proyectos que requieren constante validación como son la inteligencia artificial. Este Scrum permite llevar a cabo entregas progresivas e implementar mejoras a lo largo del proceso, por las respuestas de retroalimentación o la emergencia de problemas técnicos.

Este proyecto basado con las CNN se organiza en fases importantes como la preparación de un dataset para el entrenamiento y validación del modelo. Se deben de llevar a cabo secuencialmente para mantener una perspectiva iterativa y un rendimiento evaluado constantemente para la toma de decisiones sobre posibles ajustes o hallazgos. Esto permite un control más preciso en el progresó y la calidad del sistema desarrollado

A continuación, se presenta una línea de tiempo basado en la metodología de Scrum para visualizar la distribución temporal de los Sprints del proyecto.

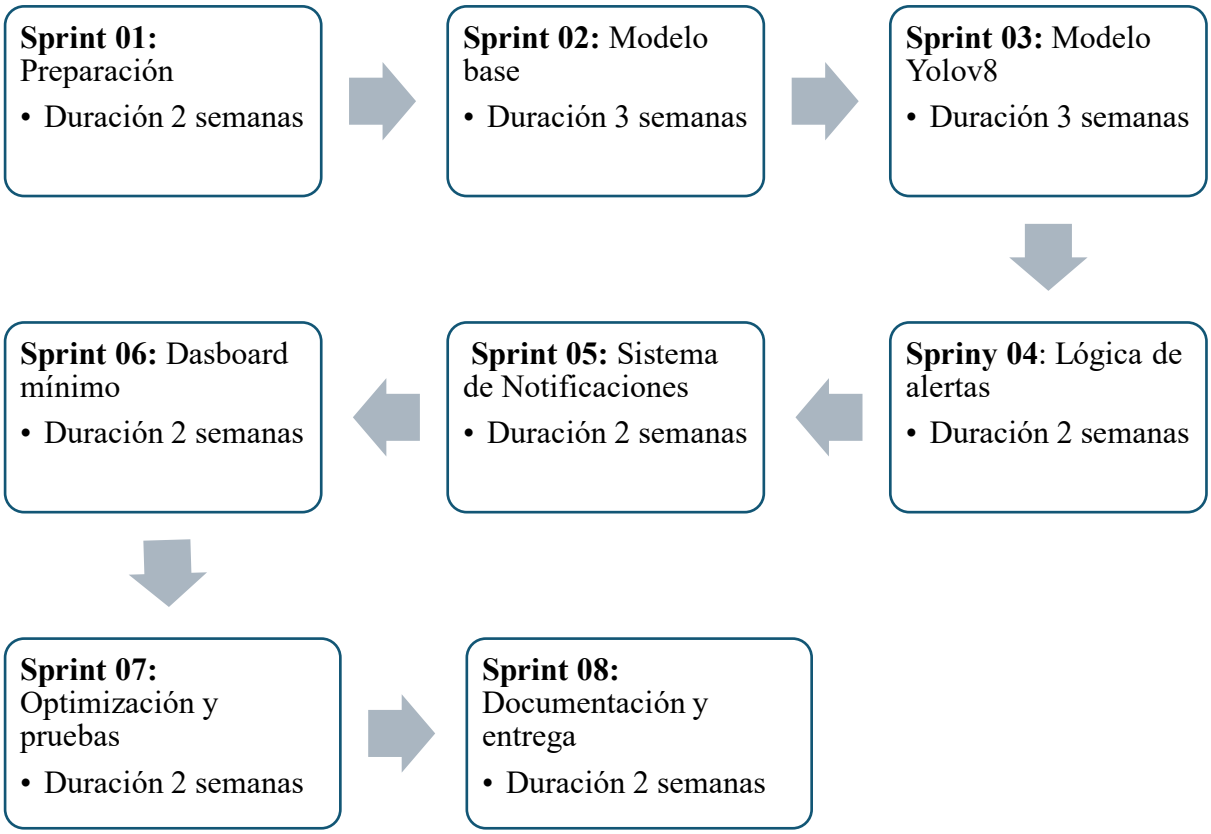


Figura 11 Línea de tiempo del desarrollo del sistema basada en metodología Scrum

4.4.1 Sprint 01 Preparación

Duración: 2 semanas.

Objetivo: Definir los componentes a reconocer, seleccionar hardware\software y planificar la recolección del dataset.

4.4.1.1 Sprint Planning – Roles y comparaciones de herramientas

El objetivo de este Sprint es establecer la base del sistema de detección, definir los roles del equipo de trabajo, seleccionar las herramientas tecnológicas a emplear e identificar los componentes principales del laboratorio de computación que serán considerados en el proceso de reconocimiento visual.

Este Sprint sirve como punto de partida metodológico dentro del marco Scrum, marcando los lineamientos esenciales para la planificación, colaboración y ejecución eficiente del proyecto.:

Rol	Nombre	Función principal
Product Owner (PO)	Tifani Carranza	Define los requisitos, prioridades y visión del sistema.
Scrum Máster (SM)	Ing. Saed Reasco	Guía y supervisor del proyecto.
Equipo de desarrollo	Tifani Carranza	Encargada del diseño, entrenamiento del modelo, pruebas y desarrollo del sistema.

Tabla 8 Roles del Equipo Scrum

A continuación, se expone un análisis detallado de los principales componentes físicos presentes, los cuales forman parte esencial de este trabajo y resultan fundamentales para la operación académica y técnica de los estudiantes. A partir de estas características se evaluará la prioridad a los componentes de mayor costo, por su impacto potencial en el rendimiento general del sistema. La información recopilada será clave para entrenar los modelos de visión artificial y establecer criterios de alerta en caso de detección de anomalías o ausencias.

Componente	Costo	Tamaño	Análisis
CPU	Alto	Grande	Alto valor económico y funcional. Por su tamaño y forma es fácilmente detectable mediante algoritmos de visión artificial.
Monitor	Alto	Mediano	Elemento de alto costo cuya pérdida compromete directamente la operatividad de cada estación de trabajo.
Teclado	Bajo	Pequeño	Riesgo bajo, aunque puede ser monitoreado como parte del conjunto.
Mouse	Bajo	Muy pequeño	Valor económico bajo y alta rotación.
Cables	Mínimo	Variable	Costo muy bajo, pero su desconexión o daño puede afectar el funcionamiento.
Servidor	Alto	Mediano	Alto valor económico para entrenamiento de modelos con GPU.

Tabla 9 Componentes de los equipos de computó

A continuación, se presenta una comparación técnica entre distintas cámaras de vigilancia, analizando sus puntos importantes y evaluación para determinar la opción más adecuada para su integración en el sistema de detección en tiempo real del laboratorio.

Modelo	Características	Costo	Análisis
Cámara Domo Hikvision Turbo Hd 1080p 2mp 2.8mm Varifocal 40m	Resolución (1920x1080). Lente Varifocal de 2.8mm. Alcance infrarrojo de hasta 40 metros. Tecnología Turbo HD (requiere DVR compatible).	$70 -$80	Ofrece buena resolución y alcance nocturno amplio.
TP-Link VIGI C330I	5 MP (2560x1920). Visión nocturna. RTSP.	$55 - $60	Fácil de instalar y compatible con software de visión artificial como OpenCV.
Hikvision DS-2CD1023G0-I	2 MP (1080p). Visión nocturna. RTSP/ONVIF.	$70 - $90	Requiere configuración avanzada, lo que puede dificultar su implementación rápida.
Dahua IPC-HFW1230S	2 MP (1080p). Visión nocturna. RTSP/ONVIF.	$60 - $80	Buen soporte y requiere ajustes técnicos.

Tabla 10 Comparativo de cámaras de vigilancia

La tabla a continuación compara los modelos vinculados para implementar al sistema de CNN, considerando la arquitectura, el nivel de complejidad y utilidad al proyecto. Este análisis es para seleccionar la manera más adecuada para ejecutar el sistema propuesto.

Modelos	Tipo de arquitectura	Complejidad	Análisis
Modelo denso	Red neuronal completamente conectada	Baja	Útil como prueba base.
CNN Básica	2 capas Conv2D + MaxPooling	Media	Reconoce formas básicas y patrones visuales.
El aumento de datos y las CNN con Dropout	Técnica de regulación para las CNN y data augmentation.	Media/Alta	Mayor capacidad de generalización.
YOLOv8	Arquitectura optimizada para detección en tiempo real.	Alta	Especialmente eficaz en tareas de detección en tiempo real de objetos de gran tamaño.
MobileNetV3	CNN ligera optimizada para móviles	Media	Buen balance entre rendimiento y eficiencia.
ResNet-50	Arquitectura profunda.	Alta	Precisión, recurso y tiempo de entrenamiento.

Tabla 11 Comparación de modelos CNN para detección y clasificación de imágenes

Finalmente, se realiza una comparación de las herramientas tecnológicas más relevantes utilizadas durante el desarrollo del proyecto. Se evalúan con el objetivo de justificar su elección en las diferentes etapas del proceso de implementación.

Herramienta	Tipo	Propósito	Ventajas
Google Colab	Entorno de desarrollo en la nube.	Entrenamiento de modelo, pruebas de librerías y visualización de datos.	Gratuito, acceso a GPU, entorno interactivo y colaborativo.
Google Drive	Almacenamiento de la nube.	Almacenamiento y carga del dataset personalizado.	Facilita la organización de los datos de entrada.
Visual Studio Code	Editor de código / IDE	Desarrollo de la interfaz del sistema.	Ligero, multiplataforma, personalizable.

Herramienta	Tipo	Propósito	Ventajas
Replit	Basado en la nube	Un desarrollador de programación desde el navegador sin instalación.	Multilenguaje, colaborativo en tiempo real y accesible.

Tabla 12 Comparativo de herramientas utilizadas

4.4.1.2 Daily Scrum – Pruebas de conexión y herramientas utilizadas

Durante la fase de desarrollo se realizó reuniones diarias con el objetivo de dar seguimiento a las actividades previamente planificadas. Este enfoque permitió mantener una coordinación eficiente, garantizar el cumplimiento de los objetivos propuestos y asegurar una planificación clara y precisa para cada tarea ejecutada. A continuación, se presenta tablas de las actividades realizadas y las herramientas utilizadas durante cinco días consecutivos:

Día	Se hizo	Avance logrado
1	Investigación cámaras IP, características técnicas y protocolos compatibles.	Lista de 3 modelos seleccionados.
2	Prueba e implementación de la cámara TP-Link VIGI C330I usando el modelo OpenCV.	Conexión y codificación en Python.
3	Captura de imágenes desde diferentes ángulos	Dataset inicial listo para entrenamiento
4	Etiquetado de imágenes con LabelImg.	Imágenes etiquetadas.
5	Configuración del entorno en Google Colab con YOLOv8.	Colab listo con entorno funcional.

Tabla 13 Registro del Daily Scrum Sprint 01

4.4.1.3 Sprint Review – Elecciones finales

A partir de las tablas comparativas anteriores, se seleccionaron estos componentes para optimizar la eficiencia y eficacia del proyecto de detección de objetos. Esta elección estratégica responde a criterios de precisión, rendimiento computacional y adaptabilidad, lo que facilitará la implementación de una solución liviana, robusta y funcional para la gestión de los equipos informáticos del laboratorio.

Entrega y decisiones finales:

- Componentes para reconocer solo CPU y monitor.

Hardware:

- Cámara TP-Link VIGI C330I (5MP, RTSP, 30 FPS).

Figura 12 Cámara TP-Link VIGI

Software:

- CNN básica.
- Modelo CNN con YOLOv8.
- OpenCV y Python.
- LabelImg

Herramientas:

- Google Colab.
- Google Drive.
- Visual Studio Code.
- LabelImg

Dataset:

- 1000 imágenes iniciales capturadas (70% train, 20% val, 10% test).

4.4.1.4 Sprint Retro – Conclusión y diagrama de flujo

Las principales ventajas destacadas:

- Enfoque metodológico claro y eficiente, facilitando cada etapa del desarrollo.
- Uso de imágenes reales para el entrenamiento y validación del sistema.

- Integración con Google drive y Colab, mejorando la accesibilidad y colaboración.
- Mejora del proyecto con herramientas y librerías.
- Lograr resultados claros y precisos.

Este Sprint se centró en el inicio y la preparación del dataset para el reconocimiento de los equipos como son los CPUs y monitores, abarcando con la elección de herramientas, librerías y equipos utilizados. A continuación, se presenta un diagrama de flujo resumido en el proceso del desarrollo diferenciando cada etapa realizada para lograr los avances esperados y estructuradas del proyecto.

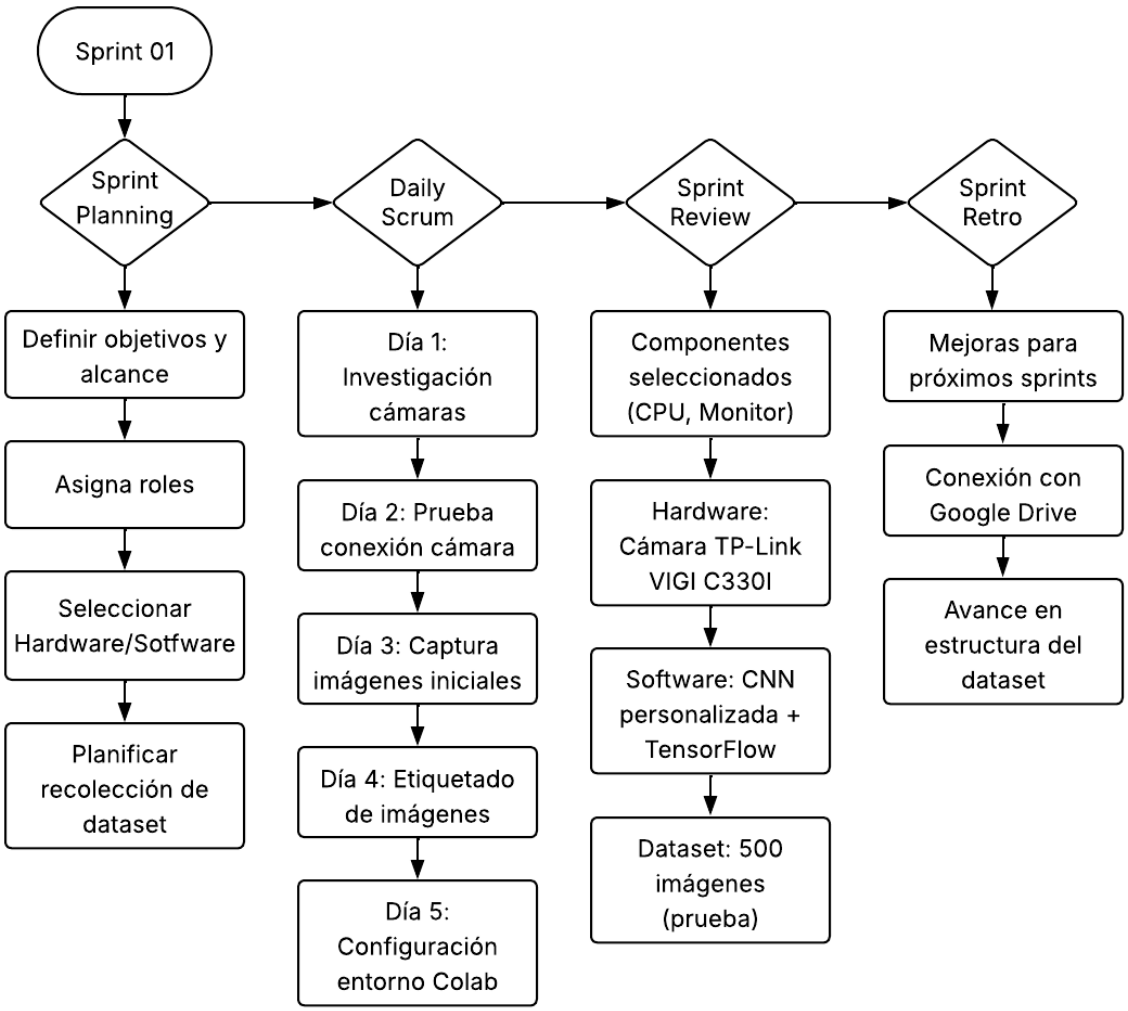


Figura 13 Diagrama de Sprint 01

4.4.2 Sprint 02 Dataset y modelo base del clasificador de imagen

Duración: 2 semanas

Objetivo: Preparar y estructurar un dataset balanceado para la detección de CPUs y monitores, asegurando una base sólida para el entrenamiento de modelos CNN.

4.4.2.1 Sprint Planning – Procesamiento de datos y desarrollo del modelo

El enfoque principal de este proceso fue la captura, organización y preprocesamiento del dataset, permitiendo definir los objetivos de procesamiento de imágenes y ajustar la distribución del trabajo. A continuación, se presentan las tablas con las actividades claves:

Actividad	Detalles	Herramientas
Captura de imágenes	Fotografiar CPU/monitor de diferentes ángulos.	Cámara TP-Link
Etiquetado	Usar LabelImg	LabelImg
Aumentación de datos	Generar variaciones con rotaciones, zoom y cambios de luz.	ImageDataGenerator (Keras)
División train/val/test	Separar aleatoriamente	sklearn.model_selection

Tabla 14 Dataset y preprocesamiento

4.4.2.2 Daily Scrum – Actividades y entrenamiento

Durante el presente sprint, se llevaron a cabo diversas tareas del sistema de clasificación de imágenes, utilizando entornos colaborativos como Google Colab. El trabajo se centró en la implementación, ajuste y evaluación de modelos de aprendizaje profundo aplicados a conjuntos de datos previamente preparados. A continuación, se detallan las principales actividades con sus respectivos fragmentos de código ejecutados.

a) Instalación de dependencias

Para exportar el modelo a formato TensorFlow.js, se instaló este paquete:

```
!pip install tensorflowjs

import tensorflowjs as tfjs
```

Figura 14 Paquete de Tensorflowjs

b) Autenticación y conexión

Se conecta Google Colab con la cuenta de Google para acceder al dataset almacenado:

```
from google.colab import auth
auth.authenticate_user()

from google.colab import drive
drive.mount('/content/drive')
```

Figura 15 Permiso de autenticación para el Drive

c) Carga y preprocesamiento de imágenes

Las imágenes se recorren y se redimensionan a un tamaño uniforme:

```
import matplotlib.pyplot as plt
import cv2
import tensorflow as tf

TAMANO_IMG = 100
plt.figure(figsize=(20, 20))
for i, ruta in enumerate(dataset.take(25)):
    ruta_str = ruta.numpy().decode()  # convertir a string
    imagen = cv2.imread(ruta_str, cv2.IMREAD_GRAYSCALE)
    imagen = cv2.resize(imagen, (TAMANO_IMG, TAMANO_IMG))
    plt.subplot(5, 5, i+1)
    plt.imshow(imagen, cmap='gray')
     plt.title(f"{ruta_str.split('/')[-2]}")  # muestra nombre de
carpeta como clase
    plt.xticks([])
    plt.yticks([])
```

Figura 16 Muestra imágenes del Data con sus etiquetas

d) Procesamiento y aumentación de imágenes

Para mejorar la diversidad del conjunto de datos se utilizó la clase de *ImageDataGenerator* de keras, permite generar nuevas variantes mediante rotaciones, zoom, giros horizontales y cambios de iluminación.

```
datagen = ImageDataGenerator (
    rotation_range=30,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=15,
    zoom_range=[0.7, 1.4],
    horizontal_flip=True,
    vertical_flip=True
)

datagen.fit(X)
plt.figure(figsize=(20,8))
for imagen, etiqueta in datagen.flow(X, y, batch_size=10, shuffle=False):
  for i in range(10):
    plt.subplot(2, 5, i+1)
    plt.xticks([])
    plt.yticks([])
    plt.imshow(imagen[i].reshape(100, 100), cmap="gray")
  break
```

Figura 17 Ejemplo de data augmentation generados con ImageDataGenerator

e) Creación del modelo CNN base y entrenamiento

Se define un modelo secuencial de tipo CNN simple, utilizando capas densamente conectadas para la tarea de clasificación:

```
      Modelo = tf.keras.models.Sequential([
        tf.keras.Input(shape=(100, 100, 1)),
        tf.keras.layers.Flatten(),
        tf.keras.layers.Dense(150, activation='relu'),
        tf.keras.layers.Dense(150, activation='relu'),
        tf.keras.layers.Dense(1, activation='sigmoid')
])
```

Figura 18 Estructura de los modelos Denso, CNN y CNN2

Se compila el modelo para definir la función de pérdida, el optimizador y las métricas de evaluación:

```
modelo.compile(optimizer='adam',
               loss='binary_crossentropy',
               metrics=['accuracy'])
```

Figura 19 Estructura de la compilación de los modelos

Finalmente, se entrena el modelo con los datos cargados:

```
modelo.fit(X, y, batch_size=32,
           validation_split=0.15,
            epochs=100,
            callbacks=[tensorboardDenso])
```

Figura 20 Estructura del proceso de entrenamiento

f) Visualización de curvas de entrenamiento

Gráficas de precisión:

```
plt.figure(figsize=(20, 8))
for i in range(10):
  plt.subplot(2, 5, i+1)
  plt.xticks([])
```

```
  plt.yticks([])
  plt.imshow(X[i].reshape(100, 100), cmap="gray")
```

Figura 21 Precisión del modelo

g) Exportación del modelo a TensorFlow.js

Se exportó el modelo para uso web:

```
!tensorflowjs_converter --input_format keras model.h5 carpeta_salida
```

Figura 22 Exportación del modelo entrenado

Salida del modelo:

Una vez finalizado el entrenamiento, el modelo se exportó al formato TensorFlow.js para permitir su integración en aplicaciones web. El proceso de exportación genera los siguientes archivos:

- group1-shard1of4.bin, group1-shard2of4.bin, group1-shard3of4.bin y group1-shard4of4.bin: Archivos binarios que contienen los pesos entrenados del modelo. Estos se dividen automáticamente por TensorFlow.js para facilitar su carga en navegadores.
- model.json: Contiene las capas, funciones y modelo del sistema.

A continuación, se muestra una tabla específica de las actividades realizadas que reflejan un enfoque más práctico y adaptable al desarrollo del sistema.

Día	Tareas	Código
1-2	Etiquetar imágenes con LabelImg (CPU y monitor).	os.path.basename(os.path.dirname(path)) para organizar según carpetas.
3	Aplicar aumento de datos (rotaciones, zoom, flips).	datagen = ImageDataGenerator(...) y datagen.flow()
4	Balancear clases para asegurar distribución equitativa.	np.unique(y_train, return_counts=True)
5	Entrenar ModeloDenso.	modeloDenso.fit(...) con TensorBoard(log_dir='logs/denso').

Día	Tareas	Código
6-7	Entrenar ModeloCNN_AD.	modeloCNN_AD.fit(...) con steps_per_epoch=... y validation_steps=...
8-9	Optimizar ModeloCNN2_AD con dropout.	Dropout (0.5) en modeloCNN2_AD, y aprendizaje controlado con validación
10	Exportar mejor modelo.	model.save("/content/mi_modelo.h5")

Tabla 15 Adaptaciones en el desarrollo del proyecto

4.4.2.3 Sprint Review – Resultados obtenidos

Dataset preparado:

- 2000 imágenes procesadas y etiquetadas.

Modelos entrenados:

- ModeloDenso (50% precisión).
- ModeloCNN_AD (75% precisión).
- ModeloCNN2_AD (85% precisión).

4.4.2.4 Sprint Retro – Análisis de resultados, dificultades y mejoras en el desarrollo

Con el análisis realizado, confirmó que los datasets y las etiquetas se han implementado correctamente. Sin embargo, la captura de imágenes requirió más tiempo debido a la cantidad de datos a procesar.

El modelo CNN2_AD superó la mejora un 50%, para el proceso de etiquetado de las imágenes se realizó desde la herramienta LabelImg optimizando el modelo. Aunque el volumen de datos sigue siendo limitado el dataset se seguirá ampliando para garantizar una ejecución precisa y sin fallos en las próximas pruebas.

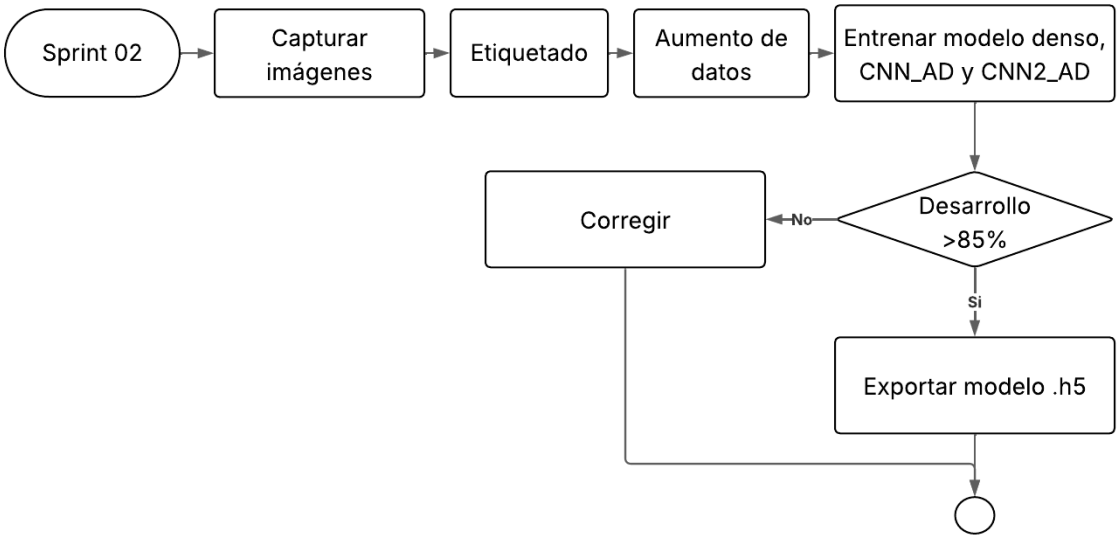


Figura 23 Diagrama del Sprint 02

Ventajas:

- Captura personalizada del dataset, adaptada a los objetivos específicos del sistema.
- Entrenamientos con diversos modelos del CNN y arquitectura.
- Mejora de precisión y validación de técnicas nuevas.

4.4.3 Sprint 03 Modelo avanzado y detección con YOLOv8

Duración: 2 semanas.

Objetivo: Optimizar la precisión del sistema a través de YOLOv8 mediante transfer learning, validación eficiente y documentación de resultados comparativos.

4.4.3.1 Sprint Planning – Implementación del modelo

Para incrementar la precisión en la detección de CPUs y monitores, se implementó YOLOv8 aprovechando su arquitectura optimizada para objetos de tamaño mediano-grande. El proceso incluyó:

- Adaptación del dataset: Conversión a formato YOLO (etiquetas en .txt con coordenadas normalizadas).
- Configuración del entorno: Instalación de YOLOv8 en Visual Studio Code.
- Entrenamiento: Ajuste de hiperparámetros.

Actividad	Detalles	Herramientas
Transfer learning	Uso de un modelo preentrenado YOLOv8	Ultralytics YOLOv8
Optimización	Ajuste de aprendizaje	Optimizar, epochs, etc.
Validación	Evaluación de precisión y perdida	TensotBoard
Documentación	Resultados y registros del modelo	Visualizaciones y métricas

Tabla 16 Proceso de actividades

4.4.3.2 Daily Scrum – Desarrollo técnico

Se detalla las actividades diarias realizada durante el desarrollo, enfocando la implementación, configuración, entrenamiento y validación del modelo. Estas actividades se preparan los datos y resultados, permitiendo una mejora continua y optimización del desempeño del sistema.

Días	Tareas	Código/Herramienta
1-2	Creación de carpeta del dataset	Explorador de archivos (Windows)
2-3	Etiquetado manual con LabelImg en formato YOLO.	LabelImg imagen.jpg Salida: .txt por imagen
4	Verificación manual del etiquetado generado por LabelImg.	Explorador de carpetas + Bloc de notas / Visualización en LabelImg
5-6	Instalación de YOLOv8 y configuración del entorno.	!pip install ultralytics
7-8	Entrenamiento inicial del modelo YOLOv8.	model.train(data='dataset.yaml')
9-10	Evaluación de resultados y métricas.	model.val() Análisis de precisión

Tabla 17 Plan de desarrollo técnico

Día 1-2: Creación de carpeta y organización del dataset

Para iniciar el desarrollo del sistema de detección de objetos con YOLOv8, se creó una carpeta principal directamente en el escritorio del equipo de trabajo. Esta carpeta fue destinada a almacenar imágenes reales capturadas en el laboratorio de computación, representando distintos ángulos y posiciones de los componentes físicos del entorno como son los CPUs y monitores.

Día 2–3: Preparación del dataset y etiquetado con LabelImg

Seguido de esto, se utilizó la herramienta LabelImg configurada en modo YOLO, lo que permitió generar automáticamente los archivos .txt correspondientes a cada imagen. Durante el etiquetado manual se asignaron correctamente las clases de los equipos (CPU y monitor).

Este proceso de etiquetado se genera automáticamente un archivo .txt por cada imagen realizada, lo cual permitió visualizar una estructura organizada de elementos en carpetas y subcarpetas. A continuación, se visualizará la estructura de las carpetas y archivos:

```
data/
├── dataset.yaml
├── train/
│   ├── images/
│   │   ├── cpu1.jpg
│   │   ├── cpu2.jpg
│   └── labels/
│       ├── cpu1.txt
│       ├── cpu2.txt
├── val/
│   ├── images/
│   │   ├── monitor8.jpg
│   │   ├── monitor9.jpg
│   └── labels/
│       ├── monitor8.txt
│       ├── monitor9.txt
```

Figura 24 Estructura del modelo YOLOv8

Día 4: Verificación del etiquetado generado

La herramienta LabelImg permite configurar en formato YOLO lo que garantizó los archivos .txt automáticamente fortaleciendo el modelo. Por ello, no fue necesario realizar ningún proceso de conversión.

Sin embargo, se realizó una revisión manual del etiquetado para verificar la integridad del dataset. Esta revisión consistió en abrir aleatoriamente varias imágenes y comprobar que:

- El archivo .txt asociado a cada imagen estaba correctamente nombrado.
- Las coordenadas y clases etiquetadas correspondían visualmente a los objetos marcados (CPU y monitores).
- No existían archivos vacíos o mal asociados.

A continuación, se muestra esta figura de la interfaz de la herramienta LabelImg, utilizada para el proceso de etiquetado manual de imágenes.

Figura 25 Interfaz de LabelImg para el etiquetado de imágenes

Día 5–6: Configuración del entorno y entrenamiento

Con la herramienta de Google Colab se pudo realizar el entrenamiento del modelo YOLOv8 aprovechando el acceso gratuito a GPU para acelerar el proceso. Para mejorar el modelo se implementó la librería Ultralytics mediante *!pip install ultralytics* y se cargó el dataset personalizado con las etiquetas previamente generadas.

```
!pip install ultralytics

import ultralytics

ultralytics.checks()
```

Figura 26 Instalación de la librería Ultralytics en Google Colab para entrenar YOLOv8

Los resultados de la instalación:

```
Ultralytics 8.3.169 🚀 Python-3.11.13 torch-2.6.0+cu124 CUDA:0 (Tesla T4, 15095MiB)

Setup complete ✅ (2 CPUs, 12.7 GB RAM, 42.2/235.7 GB disk)
```

Figura 27 Resultados de la instalación Ultralytics

Además, para cargar el modelo e implementarlo posteriormente en Visual Studio Code, se utiliza el comando:

```
model = YOLO('yolov8n.yaml')
```

Figura 28 Modelo de YOLO

Para entrenarlo, se especifica la ruta del archivo de configuración del dataset junto con los parámetros de entrenamiento mediante:

```
model.train(data='/content/drive/MyDrive/data/dataset.yaml',  epochs=50, imgsz=640)
```

Figura 29 Ruta del archivo del dataset de YOLO

Este comando indica al modelo que debe entrenarse utilizando los datos especificados en el archivo *dataset.yaml*

Día 7–8: Validación del modelo

Una vez completado el entrenamiento inicial del modelo YOLOv8, se llevó a cabo la fase de validación utilizando el conjunto de datos destinado para este fin. Para ello, se aplicó el método *model.val()* de la librería Ultralytics, el cual permite obtener de forma automática diversas métricas de desempeño relacionadas con la tarea de detección de objetos.

```
resultados = model.val()

print(resultados)
```

Figura 30 Validar el modelo entrenado

Día 9–10: Análisis de resultados finales

Finalmente, entrenado el modelo se analizarán los resultados obtenidos con la función *model.expot()* para convertirlo en formato ONNX, que permite la compatibilidad del Visual Studio Code. Lo que facilitad la interacción con la aplicación web en tiempo real.

```
model.export(format='onnx')
```

Figura 31 Exportar el modelo entrenado a formato ONNX

4.4.3.3 Sprint Review – resultados del modelo YOLOv8

El modelo YOLOv8 alcanzó una precisión del 50% en la fase de validación, con expectativas de mejora continua en la detección. Se busca reducir los errores característicos de los modelos CNN mediante:

- La utilización de imágenes etiquetadas con mayor calidad.
- Una arquitectura optimizada para la detección en tiempo real.

Dataset utilizado:

- Se trabajó con 1400 imágenes etiquetadas para prueba.
- Las imágenes fueron clasificadas mediante valores binarios, donde:
  - 1 representa la presencia de un monitor.
  - 0 indica la presencia de una CPU.

Ejemplos de etiquetas generadas:

- Para una imagen con un monitor.

```
1 0.481879 0.540943 0.630425 0.856771
```

Figura 32 Etiqueta de un monitor en archivo .txt

- Para una imagen con una CPU.

```
0 0.697266 0.557292 0.599392 0.774957
```

Figura 33 Etiqueta de CPU en archivo .txt

4.4.3.4 Sprint Retro – Lecciones aprendidas

- El uso de LabelImg en formato YOLO permitió generar anotaciones compatibles de manera rápida y eficiente.
- Para el entrenamiento y la visualización se recomienda Google Colab y Visual Studio Code.
- Se recomienda documentar todo el proceso con capturas, gráficos de precisión y pérdida, así como métricas comparativas en el Capítulo V, utilizando bases de datos reales para una evaluación más precisa.

A continuación, se muestra un diagrama de flujo sencillo, pero de forma clara y precisa en la estructura del Sprint 03. este proceso implementa y validad el modelo YOLOv8 dentro del proyecto.

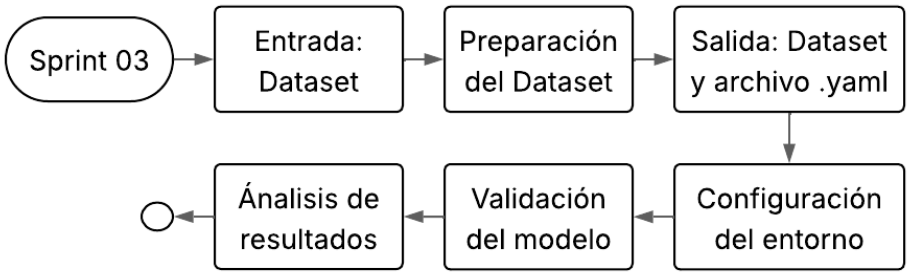


Figura 34 Diagrama del Sprint 03

4.4.4 Sprint 04 Lógica de alerta

Duración: 2 semanas.

Objetivo: Detectar movimientos y activar una alerta cuando alguno de los dos equipos de cómputo no se encuentre en su ubicación habitual.

4.4.4.1 Sprint Planning – Geolocalización y lógica

Se definió un sistema basado en coordenadas de ubicación para monitorear la posición de los equipos en el laboratorio. El objetivo fue identificar movimientos inusuales y generar alertas básicas en tiempo real. A continuación, se presenta un simulador:

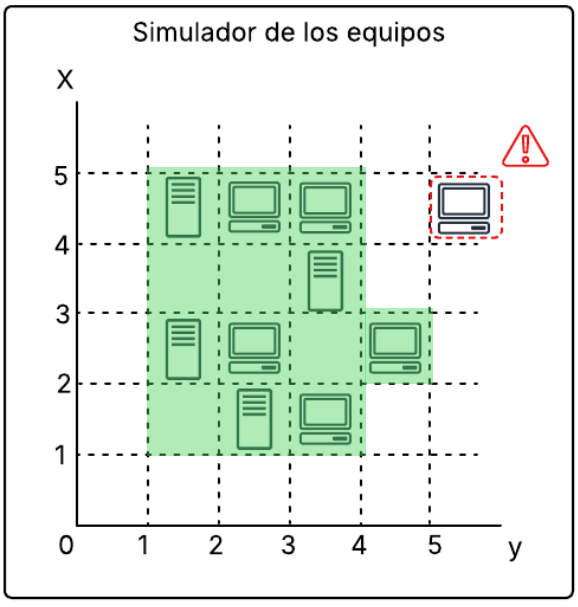


Figura 35 Simulador de los equipos informáticos

4.4.4.2 Daily Scrum – Implementación de reglas de alerta

Se define reglas básicas para la implementación del sistema para la posición mediante la lógica de condiciones y alertas en el HTML.

Actividad	Herramienta
Plano visual del laboratorio	Canva
Asignar coordenadas a los objetos	Python
Alerta si la posición es vacía	Python + YOLOv8
Generar alerta	Python

Tabla 18 implementación del sistema para la alerta

4.4.4.3 Sprint Review – Resultados obtenidos

Se espera que la alerta se activa correctamente en el código cuando se detecta movimiento fuera del área previamente establecida, lo que permite una respuesta inmediata ante situaciones anómalas. La lógica binaria implementada en el código mejora la precisión en la interpretación de los resultados, esta codificación facilita la diferenciación entre:

- Presencia (1): Indica que el equipo ha sido detectado en tiempo real.
- Ausencia (0): Señala que el equipo no está presente en la zona de monitoreo.

Figura 36 Simulador de alerta

4.4.4.4 Sprint Retro – Evaluación del sistema de ubicación

Las coordenadas asignadas a los equipos aún requieren ajustes para garantizar una localización totalmente precisa. Es necesario corregir ciertos detalles técnicos que aseguren un buen control, permitiendo que se desarrolle de manera más eficiente al sistema.

A continuación, se presenta una tabla con las principales dificultades encontradas durante este proceso:

N°	Problemas	Causas principales	Acción correctiva
1	Errores de precisión	Imágenes de entrenamiento muy pequeñas	Se agregaron nuevas imágenes con mejor calidad y resolución
2	Lento el procesamiento del dataset con 2400 imágenes	Volumen elevado de datos	Evaluar reducción de tamaños
3	Coordenadas geolocalizaciones aun no exactas	Configuración manual sin precisión	Implementar pruebas por coordenadas reales y por posiciones

Tabla 19 Errores detectados y mejoras

4.4.5 Sprint 05 Sistema de notificaciones

Duración: 2 semanas

Objetivo: Desarrollar e integrar alertas automáticas en el sistema para detectar los movimientos inadecuados en tiempo real.

4.4.5.1 Sprint Planning – Selección de canal de notificación

Durante esta fase se definió el mecanismo inicial de notificación del sistema. El sistema de aplicación web presenta una ventana flotante para que se visualice la alerta directamente en la pantalla detectando algún equipo ausente de su ubicación habitual.

Como mejoras futuras se implementará un canal de comunicación mediante notificaciones por el correo electrónico mejorando las respuestas de la alerta, así el usuario recibe el aviso, aunque no este activo en el sistema.

4.4.5.2 Daily Scrum – Registro de alertas en la web

Se organizó una secuencia diaria de tareas enfocadas en la implementación del desarrollo del sistema de alertas, con las funciones específicas que facilitan la interacción de las alertas.

Día	Actividad	Herramienta
1-2	Implementar mensaje de alerta en el HTML	HTML + JavaScript
3-4	Capturar fecha y hora del evento	Python + datetime
5-6	Guardar alerta	SQLite o .txt
7-8	Mostrar historial	Tabla en HTML

Tabla 20 Planificación del registro de alertas

4.4.5.3 Sprint Review – Requisitos esperados

Como se muestra en la tabla 20, se espera que el sistema registre correctamente el historial de alertas y muestre los mensajes correspondientes cuando se detecte la ausencia de objetos. Aunque aún existen detalles por corregir en el funcionamiento del sistema, se proyecta que alcance un rendimiento óptimo en su versión final.

Para ello, se ha diseñado la siguiente estructura en SQL que permite almacenar los registros de alertas generadas por el sistema:

```
CREATE TABLE historial_alertas (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    mensaje TEXT NOT NULL,
    fecha_hora DATETIME NOT NULL
);
```

Tabla 21 Estructura de la tabla en SQL

Resultados de la creación de la tabla:

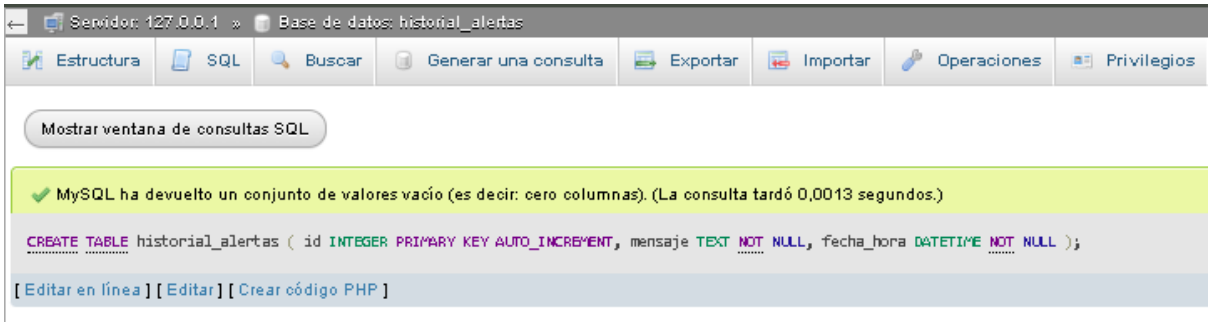


Figura 37 Creación exitosa de la tabla historial_alertas

4.4.5.4 Sprint Retro – Evaluación del sistema de alertas web

Registrar alertas dentro del mismo sistema web facilita el seguimiento inmediato sin depender de herramientas externas. Esta funcionalidad brinda al usuario la capacidad de visualizar eventos en tiempo real y consultar el historial de incidencias de forma rápida y eficiente.

Ventajas:

- Mejora la precisión de incidentes.
- Reducción de tiempo con la alerta.
- La interfaz web facilita el acceso remoto para el responsable de supervisión.

4.4.6 Sprint 06 Monitoreo

Duración: 2 semanas

Objetivo: Diseñar e implementar una interfaz visual con Flask que permita el monitoreo en tiempo real y genere alertas.

4.4.6.1 Sprint Planning – Estructura del módulo

El objetivo de esta fase es facilitar la visualización de eventos y control de monitore, mejorando la experiencia del usuario con una interfaz sólida y fácil de utilizar con respuestas ante situaciones detectadas. Para ello, se definieron los siguientes componentes esenciales:

- Visualización de video en vivo.
- Cuadro de monitoreo de objetos.
- Tabla de alertas dinámicas
- Estilos personalizados.

El desarrollo de la interfaz web fue realizado utilizando el Framework Flask, debido a su flexibilidad al momento de desarrollar en la web y a su capacidad para reducir bastantes líneas de código, implementando adicionalmente modelos como CNN Y YOLOv8, para una mejor eficiencia de visualización en tiempo real y la notificación de alertas.

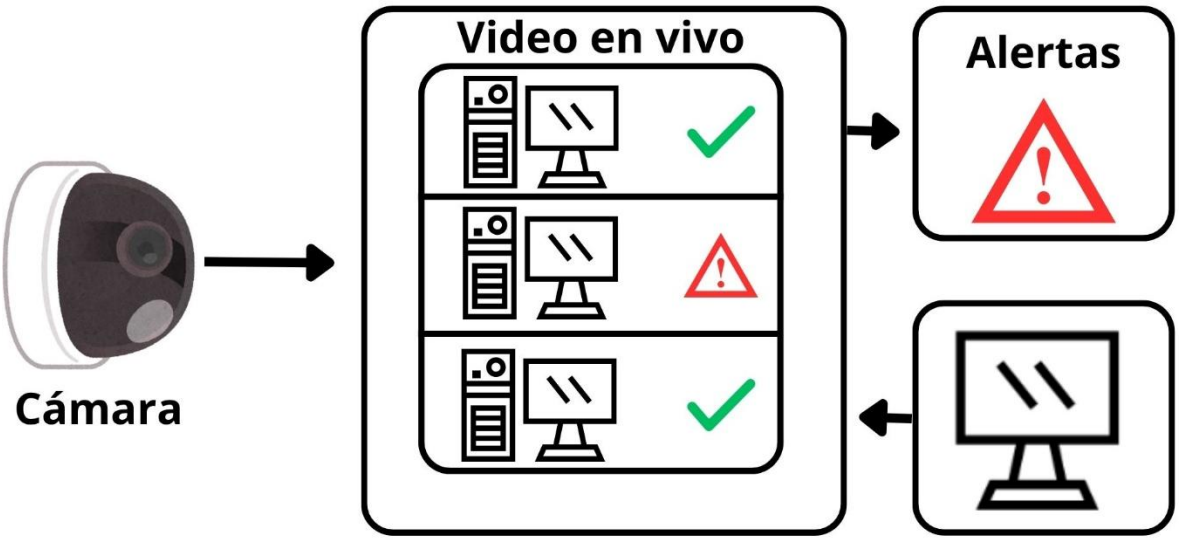


Figura 38 Diagrama del monitoreo en tiempo real del sistema

4.4.6.2 Daily Scrum – Construcción del dashboard

Se presenta las actividades realizadas durante el monitoreo del dashboard. La estructura incluye una interfaz para interactuar en tiempo real la detección, visualizar alertas e implantación de autenticación para el acceso al sistema.

Día	Actividad	Herramienta
1-2	Estructura de la interfaz con HTML/CSS	VSCode + Flask
3-4	Integración de la cámara y detección	OpenCV + YOLO v8
5-6	Registro y visualización de alertas	Python + SQLite + HTML
7-8	Implementación de login básico	Flask

Tabla 22 Construcción del dashboard

4.4.6.3 Sprint Review – Resultados

Se logró implementar una interfaz capaz de mostrar el video en tiempo real con detección de objetos, junto con una tabla de estado que indica la presencia o ausencia de CPU y monitores. A continuación, un ejemplo de la tabla en el HTML:

Ítem	CPU	Monitor
1	1	1
2	1	1
3	1	0

Ítem	CPU	Monitor
4	0	1
5	1	1

Tabla 23 Visualización de la tabla en el HTML del estado actual de los equipos

Esta sección de la lógica fue desarrollada para facilitar la visualización estructurada del estado actual del laboratorio mediante HTML. Cada fila representa una estación individual, mostrando los objetos detectados según su estado operativo. Elementos clave del sistema:

- Monitoreo inteligente del entorno.
- Integra con la base de datos.
- Registro y análisis histórico de eventos.

4.4.6.4 Sprint Retro

El propósito de estas notificaciones es la visualización de valores binarios donde se asigna cada equipo. Por ejemplo, el valor 1 son los dispositivos presentes, mientras que el 0 presenta desaparición. Además, cada evento se registra en una base de datos (BDD) con su respectiva fecha y hora facilitando un seguimiento preciso.

El siguiente diagrama de flujo representa el continuo del sistema de monitoreo, que contempla la captura de imágenes, su procesamiento mediante técnicas de visión computacional y el posterior almacenamiento estructurado en la BDD.

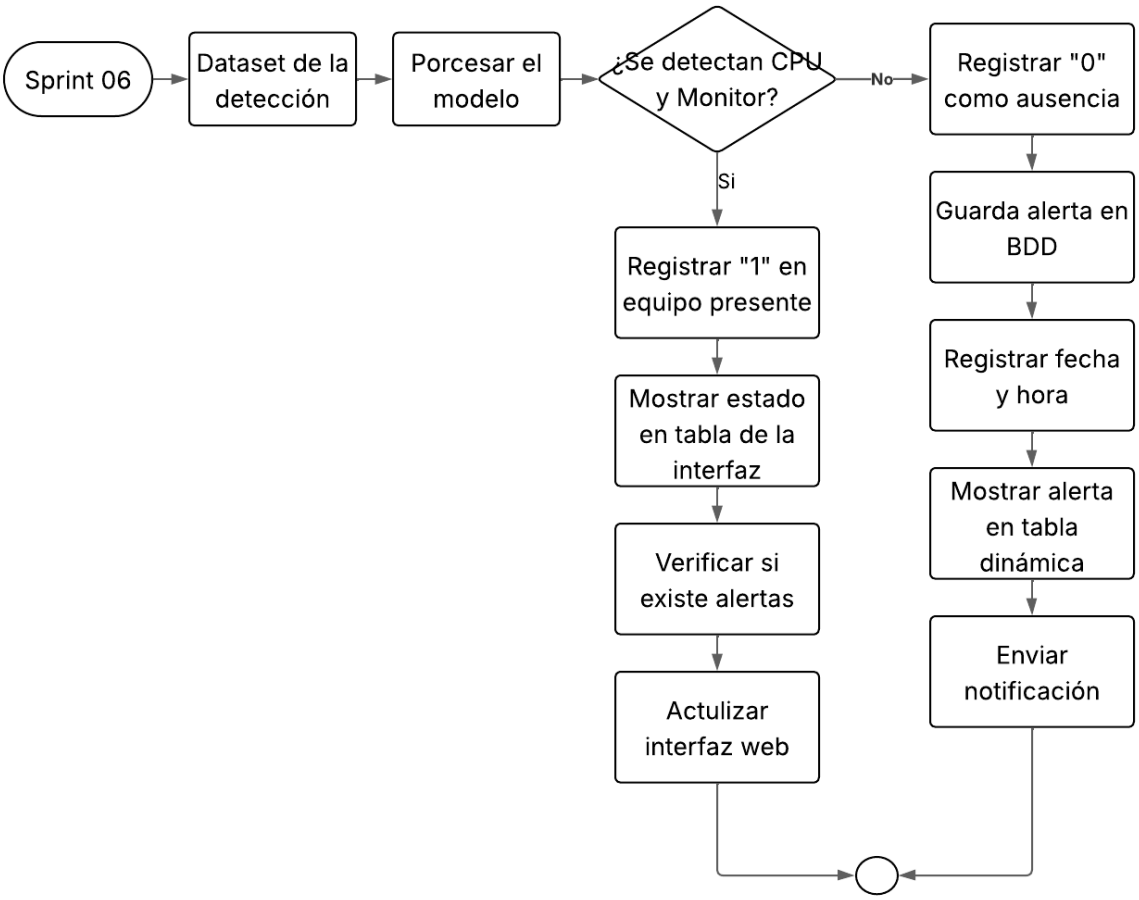


Figura 39 Diagrama de Sprint 06

Observaciones del sistema

- Permite identificación y seguimiento de eventos en tiempo real.
- Facilita la documentación estructurada de los eventos detectados.
- La arquitectura modular del sistema permite su escalabilidad futura.

4.4.7 Sprint 07 – Login básico

Duración: 1 semana

Objetivo: Implementar un prototipo de login simulado para el sistema, con interfaz HTML y lógica de autenticación básica mediante Flask, que permita validar el acceso del administrador encargado del monitoreo.

4.4.7.1 Sprint Planning – Autenticación básica

El propósito de este sprint consiste en desarrollar el mecanismo de autenticación para el sistema, implementando un formulario en HTML conectado a Flask que permita el ingreso

controlado mediante credenciales únicas. La autenticación estará limitada a un usuario administrador, configurado específicamente para realizar pruebas en el entorno de monitoreo.

Requisitos técnicos:

Elemento	Descripción
Infraestructura web	HTML, CSS
Backend	Flask (Python)
Seguridad	Validación de credencial
Usuario	Admin
Contraseña	Admin
Página de destino	En la página principal del sistema
Errores	Mensaje visible en caso de credenciales incorrectas.

Tabla 24 Requisitos del login

4.4.7.2 Daily Scrum – Proceso técnico del login

Este Sprint desarrollo actividades en el diseño e implementación del paso de la autenticación del sistema. se presentarán las tareas ejecutadas en el desarrollo de esta interfaz:

- Diseño de la interfaz de inicio de sesión
- Implementación de credenciales para la seguridad y controles dinámicos.
- Integración de mecanismos de seguridad
- Configuración de la redirección lógica hacia funcionalidades internas del sistema.
- Ejecución de pruebas unitarias y monitoreo continuo de rendimiento

Estas actividades se llevaron a cabo dentro de este sprint, permitiendo una revisión continua del avance, ajustes colaborativos y gestión eficiente:

Día	Actividad	Herramienta
1	Diseño del formulario de login	HTML + CSS
2	Lógica de validación de credenciales	Flask (Python)
3	Redirección la pantalla principal	Index.html

Día	Actividad	Herramienta
4	Registro de acceso	Código estático
5	Pruebas de validación	Navegador

Tabla 25 Actividades planificada del sprint 07

Día 1: Diseño del formulario

Durante el desarrollo del formulario de autenticación, se estructuró la lógica de validación de acceso y se aplicaron estilos personalizados mediante CSS. Como parte del diseño, se incorporó la imagen institucional *Uleam.png* como fondo del formulario, reforzando la identidad visual de la universidad.

El siguiente fragmento de código corresponde a la sección del archivo login.html, donde se define el fondo de la interfaz utilizando la imagen mencionada:

```
body {
            font-family: Arial, sans-serif;
            margin: 0;
            padding: 0;
            height: 100vh;
            display: flex;
            justify-content: center;
            align-items: center;
            background-image:                    url('{{url_for("static",
filename="uleam.png") }}');
            background-size: cover;
            background-position: center;
            background-repeat: no-repeat;
        }
```

Figura 40 Fragmento de código para el fondo

El objetivo es reforzar la identidad visual a la institución dentro de la interfaz de inicio sesión, se integró el logotipo oficial de la universidad mediante la inclusión de un fragmento

de código HTML. Este se colocó estratégicamente después del contenedor principal (<div class>) para asegurar una correcta disposición visual.

```
<img src="{{ url_for('static', filename='logo.jpg') }}" alt="Logo Universidad" class="university-logo">
```

Figura 41 Fragmento de código para la visualización del logo

Día 2: Lógica de validación

La autenticación se implementó en el backend utilizando Flask. El sistema captura los datos del formulario a través del método POST y verifica las credenciales con los siguientes valores definidos para pruebas:

```
VALID_USERNAME = "admin"
VALID_PASSWORD = "admin"
```

Figura 42 Validación de los campos de autenticación

Día 3: Redirección

Se estableció la ruta /index mediante Flask para redirigir al usuario tras una autenticación exitosa:

```
@app.route("/index")
def index():
    return send_file("index.html")
```

Figura 43 Definición de la ruta hacia el módulo principal

Día 4: Registro de acceso

se finalizó exitosamente la lógica funcional del formulario de inicio de sesión, consolidando el flujo completo de validación y redirección del usuario. El módulo implementado incluye:

- Verificación de credenciales
- Gestión de errores e información de retroalimentación al usuario en tiempo real
- Redirección condicional hacia las vistas internas del sistema

El siguiente fragmento de código representa el núcleo funcional del proceso de login, integrando los mecanismos de entrada, verificación y navegación:

```
@app.route("/", methods=["GET", "POST"])
def login():
    error = None
    if request.method == "POST":
        if request.form["username"] == VALID_USERNAME and
request.form["password"] == VALID_PASSWORD:
            return redirect(url_for("index"))  # Redirige a la ruta
        else:
            error = "Credenciales incorrectas"
    return render_template("login.html", error=error)
```

Figura 44 Código que define el inicio de sesión

Día 5: Pruebas de validación

- Se verificó el acceso exitoso al ingresar el usuario y la contraseña correctos.
- Se validó que el sistema no permite el ingreso cuando los campos están vacíos.
- Se comprobó que, en caso de ingresar credenciales incorrectas, el acceso al sistema se bloquee adecuadamente.

4.4.7.3 Sprint Review- Login funcional y navegación

El login muestra una interfaz sencilla y eficiente al proporcionar un acceso básico como el usuario y contraseña, estas credenciales se validan para ingresar correctamente y dirigirse al módulo principal de la ruta index.html.

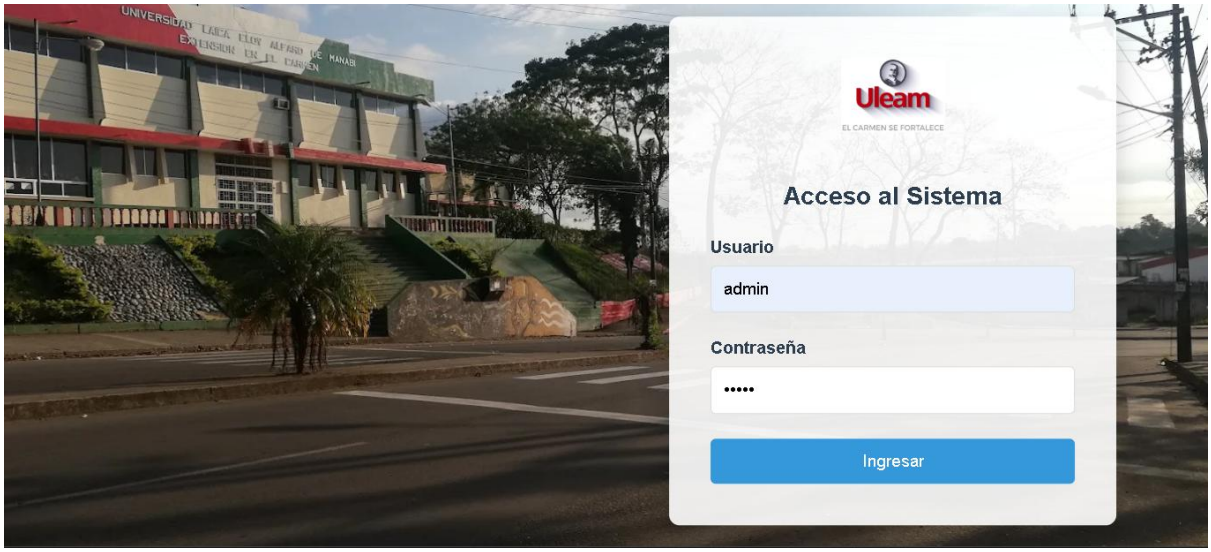


Figura 45 Interfaz de acceso del sistema

4.4.7.4 Sprint Retro – ajustes

- Actualmente, la autenticación es simulada con credenciales estáticas.
- Se recomienda para próximas versiones integrar una base de datos (SQLite o PostgreSQL) y encriptación de contraseñas.
- El login permite controlar el acceso al monitoreo, aunque aún no gestiona sesiones ni usuarios múltiples.

Se incorpora un diagrama de flujo que ilustra de manera estructurada y visual las etapas implementadas durante el sprint, centradas en el desarrollo del módulo de autenticación. Este recurso facilita la comprensión del proceso técnico al mostrar la secuencia lógica de validación de credenciales.

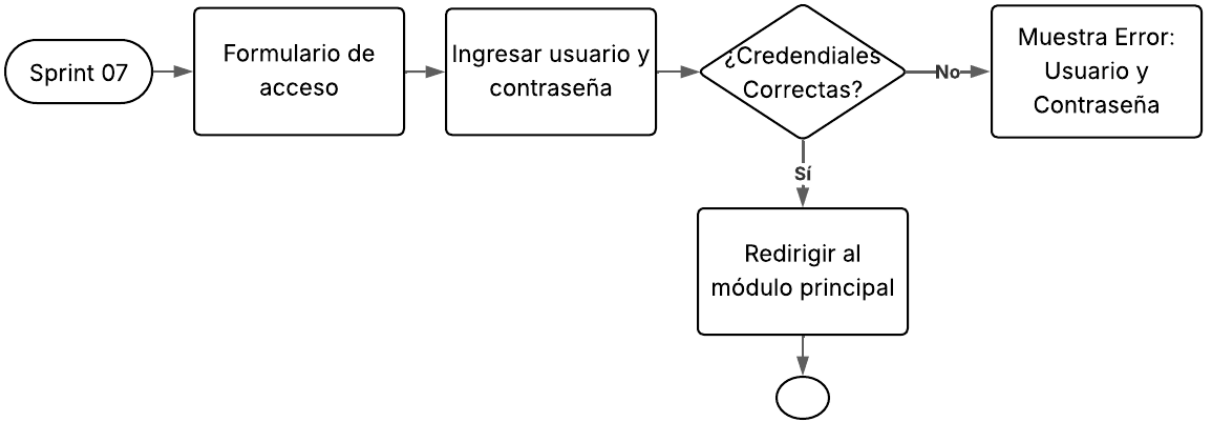


Figura 46 Diagrama del Sprint 07

4.4.8 Sprint 08 – Optimización y pruebas

Duración: 2 semanas

Objetivo: Mejorar la eficiencia del sistema, reduciendo los errores y validando las funciones con la cámara.

4.4.8.1 Sprint Planning – Estrategia de optimización

Las metas es reducir el tiempo de la interacción personal, por la automática asi mejoramos el cuidado de los equipos, se implementa pruebas y las entradas con el video.

Se presenta la tabla para la recopilación de resultados para optimizar y analizar su impacto:

Objetivo	Implementación	Resultados
Reduce tiempo	Activar alertas automáticas	Validación de respuestas rápidas
Progreso de precisión	Ajustar la precisión en YOLOv8 y Python	Reducción de falsos positivos y negativos
Validar y entrenar	Entrenamiento de diversos ángulos	Seguridad del sistema

Tabla 26 Estrategias de optimización

4.4.8.2 Daily Scrum- Ajustes técnicos y validación con YOLO

Para el entrenamiento de detección con YOLO se incluirá los archivos en Visual Studio Code el encargado de implementar la codificación con Python para una visualización y precisión eficiente. Se diseño en tiempo real permitiendo ver los resultados de las pruebas y generando archivos de salida para analizar. A continuación, se detalla las actividades importantes:

a) Cargar modelo ONNX

Se utiliza el archivo entrenado en formato .onnx.

```
session = ort.InferenceSession("best.onnx")
```

Figura 47 Descarga de modelo ONNX

b) Cargar modelo original de Yolo

Se visualiza el modelo original de Yolo en formato *.pt* para visualizar o comparar directamente el modelo.

```
model = YOLO("yolo11n.pt")
```

Figura 48 Cargar modelo YOLO

c) Entrada del modelo

Se obtiene la entrada por el modelo ONNX que es necesario para alimentar los datos preprocesados.

```
input_name = session.get_inputs()[0].name
```

Figura 49 Entrada del modelo ONNX

d) Inicio de la cámara web

Se activa la cámara del sistema para capturar imágenes en tiempo real.

```
cap = cv2.VideoCapture(0)
```

Figura 50 Activación de la cámara web

e) Preprocesamiento de la imagen capturada

Su función es de adaptar la imagen capturada por el formato del modelo ONNX.

```
def preprocess(frame):
    img = cv2.resize(frame, (640, 480))
    img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    img = img.transpose(2, 0, 1)  # HWC a CHW
    img = img / 255.0
    img = img.astype(np.float32)
    img = np.expand_dims(img, axis=0)
    return img
```

Figura 51 Procesamiento de imagen capturada

f) Predicciones obtenidas del modelo

Se extrae resultados del modelo para representar las predicciones de los objetos de CPU y monitor detectando la imagen.

```
predictions = outputs[0][0]
```

Figura 52 Predicciones del modelo

g) Guardar el archivo de Json

Se almacenan las detecciones del archivo *.json* para facilitar el análisis.

```
with open("deteccion.json", "w") as f:
        json.dump(detecciones, f)
```

Figura 53 Archivo Json para el análisis del sistema almacenado

h) Salida del entrenamiento

Se estableció una condición de salida y se finaliza presionando la tecla “q”.

```
if cv2.waitKey(1) & 0xFF == ord('q'):
        break
```

Figura 54 Salida del entrenamiento

4.4.8.3 Sprint Review – Resultados

En esta etapa se presenta los resultados corregidos como el sprint anterior. Estas mejoras buscan optimizar procesos de precisión en la detección de los objetos de cómputo, garantizando la estabilidad y eficiencia del funcionamiento del sistema.

4.4.8.4 Sprint Retro – Evaluación rendimiento final

Se presenta una retroalimentación de este Sprint, se identificaron seguimientos importantes para fortalecer el sistema:

- Aumentar la velocidad del procesamiento.
- Optimatizar el rendimiento.
- Implementar un control para mantener las versiones.
- Reducir el tiempo de espera.
- Realizar pruebas continuas.

Se presenta un diagrama de flujo detallado del proceso de entrenamiento y el despliegue del sistema detección, donde se prepara la exportación del modelo y se ejecuta en el entorno de desarrollo. estos archivos se integran a la página web para visualizar los resultados.

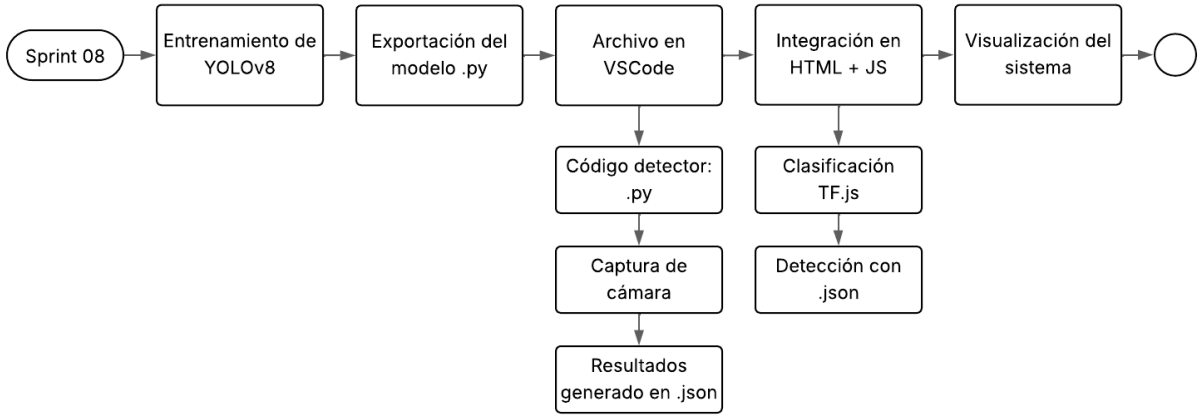


Figura 55 Diagrama del Sprint 08

4.4.9 Sprint 09 Documentación y entrega final

Duración: 1 semanas

Objetivo: Preparar y entregar el sistema ya completado incluyendo las pruebas y evidenciando cada proceso realizado.

4.4.9.1 Sprint Planning – Organización de entrega y presentación

Durante esta etapa se planifico la entrega final, estructurando los elementos del desarrollo y funcionamiento del sistema. Cada fase fue reforzada progresivamente, con el objetivo de consolidar una solución funcional que cumpla con los requisitos establecidos y garantice un nivel básico de calidad.

4.4.9.2 Daily Scrum -Desarrollo de material

En la siguiente tabla se resume las principales actividades desarrolladas en las etapas finales del proyecto, enfocadas en la entrega del sistema.

Ítem	Actividad	Herramienta
1	Documentar el programa	Visual Studio Code
2	Redactar la instalación y uso	Word
3	Diseñar la interfaz final	HTML
4	visualización de tablas y video en tiempo real	Página web
5	Base de datos	Para el registro de fecha y hora

Tabla 27 Actividades finales del sistema

4.4.9.3 Sprint Review

Durante la fase final se mostraron todos los avances obtenidos a lo largo del desarrollo del sistema. Las entregas se organizaron en categorías de documentación, interfaz y materiales.

a) Documentación

Documento	Descripción
Manual técnico	Contiene modelos, configuración y librerías utilizadas.
Manual de usuario	Instrucciones del sistema básico y monitoreo en tiempo real desde la web.
El desarrollo de los Sprint	Detalle del desarrollo técnico por iteración del Scrum.

Tabla 28 Estructura básica de la documentación

b) Estructura del repositorio

Se presenta una estructura que presenta la carpeta del proyecto en visual studio code, que permite una navegación lógica para desarrolladores y usuarios.

```
/CPUMONITOR/
├── /modelos/
│   ├── best.onnx
│   └── yolo11n.pt
│   └── yolov8n.pt
├── index.html
├── style.css
├── script.js
├── detector_yolo.py
└── deteccion.json
```

Figura 56 Estructura de la carpeta del HTML del proyecto

c) Materiales

- Librerías: TensorFlow / Keras, Ultralytics YOLOv8, OpenCV, Flask, SQLite, HTML / CSS / Bootstrap y TensorFlow.js.
- Captura de pantalla del sistema.
- Flujo de datos desde la cámara.
- Visualización de alertas y respuestas del sistema.

4.4.9.4 Sprint Retro

Mediante esta metodología Scrum se estructuró por ocho Sprints para abordar 16 semanas de trabajo, permitiendo organizar, facilitar y asegurarse en realizar cada etapa del trabajo.

A continuación, se detalla los principales avances y las dificultades;

Avances:

- La estructura de los Sprints permitió mantener un flujo de trabajo constante, con entregables claros en cada iteración.
- Para el resultado de validación se clasifico con las CNN para permitir avanzar con los modelos.
- El formato ONNX facilitó la ejecución de tareas en el entorno web.
- Para una mejor visualización se desarrolló esa interfaz clara y visual contribuyó significativamente a la interpretación de los resultados.

Dificultades:

- En el transcurso del proceso de pruebas, cuando se estaba ejecutando el clasificador de imágenes nos arrojó el siguiente error en el código *json: input_layer.js:74 Uncaught (in promise) Error: An Input_Layer should be passed either a batchInputShape or an inputShape.* Debido a que en el modelo TensorFlow, nos solicita que le indiquemos el tamaño de tensores que va a recibir, al no encontrar esa información nos arroja dicho error en la ejecución.
- En LabelImg al etiquetar se demoró por el aumento de imágenes se dificulto un poco la rapidez.
- El formato ONNX presento desafíos se descargó el modelo a través de .pt
- El sistema de detección se requiere resolver entre ambos entornos.

A continuación, se presenta un diagrama que resume el desarrollo general de los Sprints ejecutados durante la planificación e implementación de este capítulo. El cronograma abarcó un total de 16 semanas, en las cuales se estructuraron diversos Sprints enfocados en tareas específicas. La mayoría de ellos tuvo una duración de dos semanas, mientras que dos Sprints

se llevaron a cabo en un solo periodo semanal, según la complejidad de las actividades asignadas.

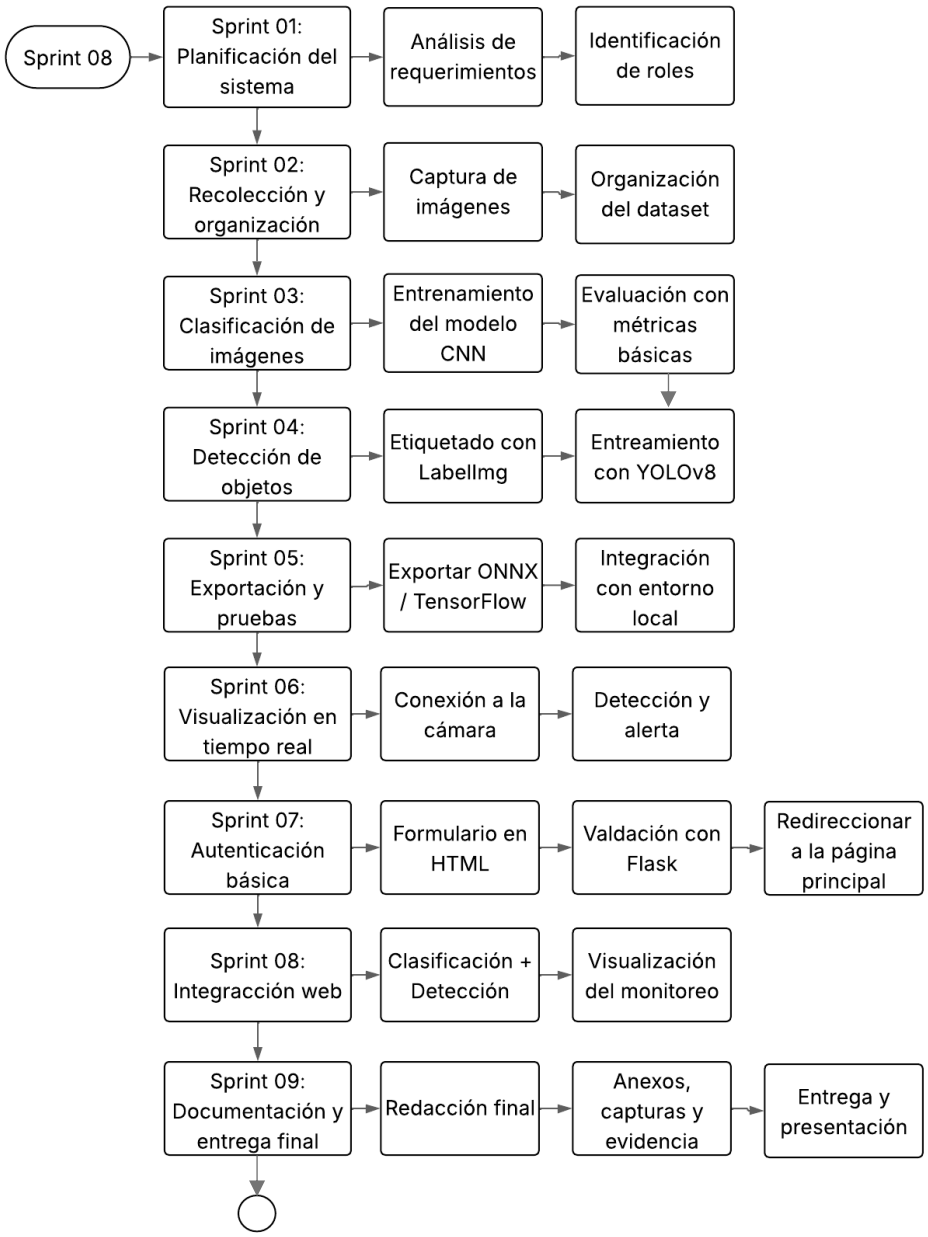


Figura 57 Diagrama general de los Sprint

Capítulo V

5 EVALUACIÓN DE RESULTADOS

5.1 Introducción

Este capítulo tiene como objetivo presentar, analizar e interpretar los resultados obtenidos tras la implementación del sistema de monitoreo inteligente. La solución propuesta combina redes neuronales convolucionales (CNN) para la clasificación de imágenes, junto con el modelo YOLOv8 para la detección en tiempo real de dispositivos informáticos, específicamente CPUs y monitores.

Se realizó un proceso de pruebas simuladas, para así poder validar la productividad del sistema computacional desarrollado. Reproduciendo el entorno del laboratorio mediante un conjunto de datos mostrando imágenes reales y demás datos controlados. Con la intención de hacer una comparación entre el monitoreo manual, frente al monitoreo automatizado.

A través del proceso de pruebas, se dio a cabo con la aplicación de la metodología Ágil Scrum, consiguiendo así dar una mejor estructura a los ensayos, pudiendo así medir con precisión los indicadores técnicos, mejorando los tiempos de respuesta ante los procesos y mejorando a gran escala el margen de error en los resultados. Mostrando en mejores tiempos las alertas que detecta el sistema en un entorno ante la ausencia de equipos en el laboratorio.

5.2 Presentación y monitoreo de resultados

El producto fue procesado mediante evaluaciones en escenarios reales dentro de los laboratorios de cómputo, con la intención de darle experiencia y así poder checar sus capacidades de funcionalidad, en donde se tomaron en cuenta diferentes aspectos indicados a continuación:

- Mediante el modelo YOLOv8 se activó la cámara web, así detectar en tiempo real el entorno del laboratorio donde se ejecutaron las pruebas.
- Tratamiento de archivos como imágenes obtenidas del conjunto de datos, el cual se asignó para hacer las pruebas de rendimiento en el escenario generado.

- Interacción mediante la interfaz web, la cual permitió realizar el inicio de sesión y acceder al panel de monitoreo donde se visualizan los resultados generados por el sistema.
- Registro de las acciones detectadas mediante la aplicación del sistema computacional, en base a la ejecución del monitoreo en el escenario generado para las pruebas realizadas.

A continuación, se presenta una tabla que resume los principales indicadores de rendimiento obtenidos, así como el porcentaje de mejora alcanzado con respecto al modelo base:

Indicador	Sistema propuesto	Mejora
Tiempo de detección	10 segundos	70%
Reconocimiento de los objetos	80% con YOLOv8	30%
Clasificación de los equipos	70% con el modelo CNN	20%
Registro de actividad	Automático	80%
Acceso seguro al sistema	Login validado	100%

Tabla 29 Hay que mencionar que la mejora está en comparación con el proceso manual.

En la búsqueda de verificar la correcta funcionalidad, de los modelos implementados en el sistema computacional para el monitoreo, se desarrolló una comparación tomando los principales indicadores técnicos para basarnos en ellos. Permitiendo así identificar el modelo más adecuado para el monitoreo del entorno generado.

La a siguiente tabla resume los resultados obtenidos durante las pruebas controladas:

Modelo	Precisión	Tiempo de detección	Falsos positivos
CNN básico	70%	120	25
YOLOv8	80%	90	8

Tabla 30 Comparación de desempeño entre modelos de clasificación y detección.

Los resultados que se aprecian dan a conocer que evidentemente existe una mejora instantánea, luego de la implementación del modelo YOLOv8, mejorando la precisión y así consiguiendo que disminuyan los errores de clasificación de objetos.

En el proceso de las pruebas dadas en escenarios con condiciones reales, se permitió obtener los indicadores presentados a continuación:

- *Tasa de éxito:* se refleja un 80% en un escenario que está operativo, dando como resultado una detección de objetos correcta.
- *Tiempo promedio de respuesta:* 2 segundos, lo que demuestra una capacidad eficiente para aplicaciones en tiempo real
- *Falsos positivos:* 5%, principalmente por variaciones en los ángulos y condiciones de iluminación de captura y condiciones de iluminación no controladas.

5.2.1 Detalles y resultados del programa

En esta parte se exponen los datos en base a los resultados que se obtuvieron durante el desarrollo de la investigación, buscando que se facilite la visualización de los procedimientos ejecutados.

a) Detección con YOLOv8

El modelo nos brindó mejorar los tiempos de una manera significativa en el sistema computacional, permitiendo cumplir con los requerimientos necesarios para las necesidades presentadas en los laboratorios de cómputo. En base a su correcta funcionalidad de procesar datos en tiempo real, se facilitó una correcta vigilancia y detección de objetos en el monitoreo realizado. Obteniendo información corroborada en base a los objetos detectados y permitiendo mediante ese proceso detectarlos de forma precisa y correcta. A continuación, se presenta resultados obtenidos de un procedimiento realizado por el sistema computacional.

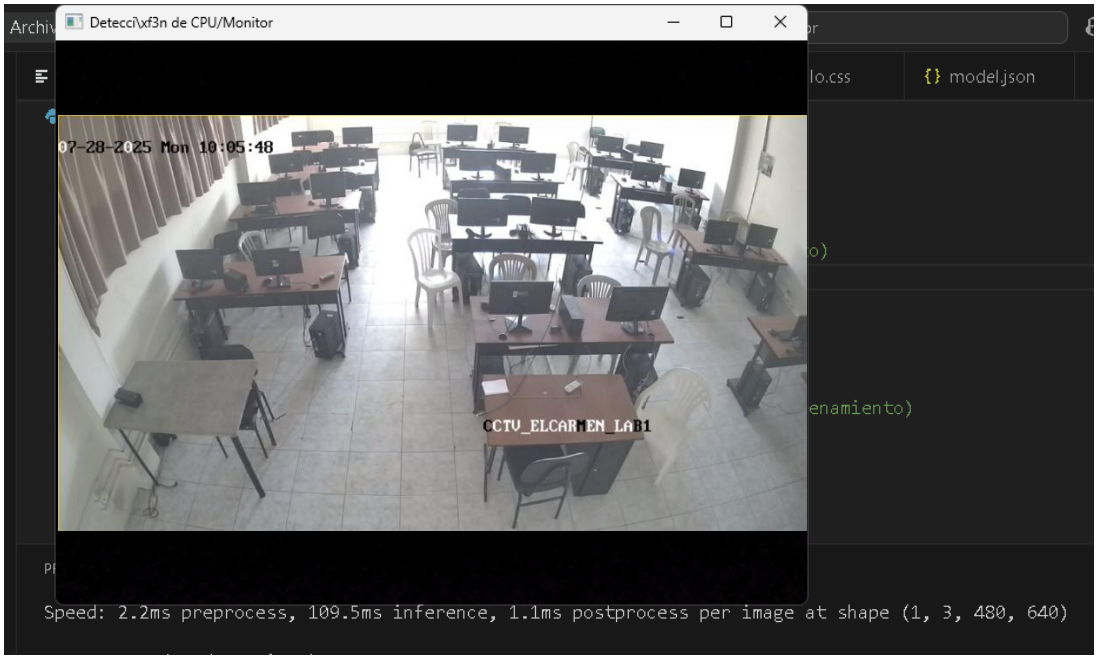


Figura 58 Simulador de la detección de objetos

Complicaciones identificadas durante la ejecución del modelo de detección:

Durante la fase de pruebas del archivo de configuración yolov8n.yaml, se evidenciaron complicaciones relacionadas con el rendimiento del entrenamiento:

- Carga lenta del modelo: El proceso de entrenamiento mostró un tiempo de respuesta.
- Interrupciones no previstas: En situaciones donde el equipo de cómputo se reinicia o entra en suspensión.

b) Clasificador de imágenes

Se ejecutó pruebas en una interfaz de simulación teniendo como resultados, que el sistema detectaba los objetos y los identificaba de manera adecuada, permitiendo clasificarlos. Se desarrolló en un entorno web local en el puerto 8000, para el acceso a la información.

Cabe señalar que esta versión representa únicamente un simulador funcional del sistema de clasificación, orientado a validar su rendimiento y precisión en condiciones controladas.

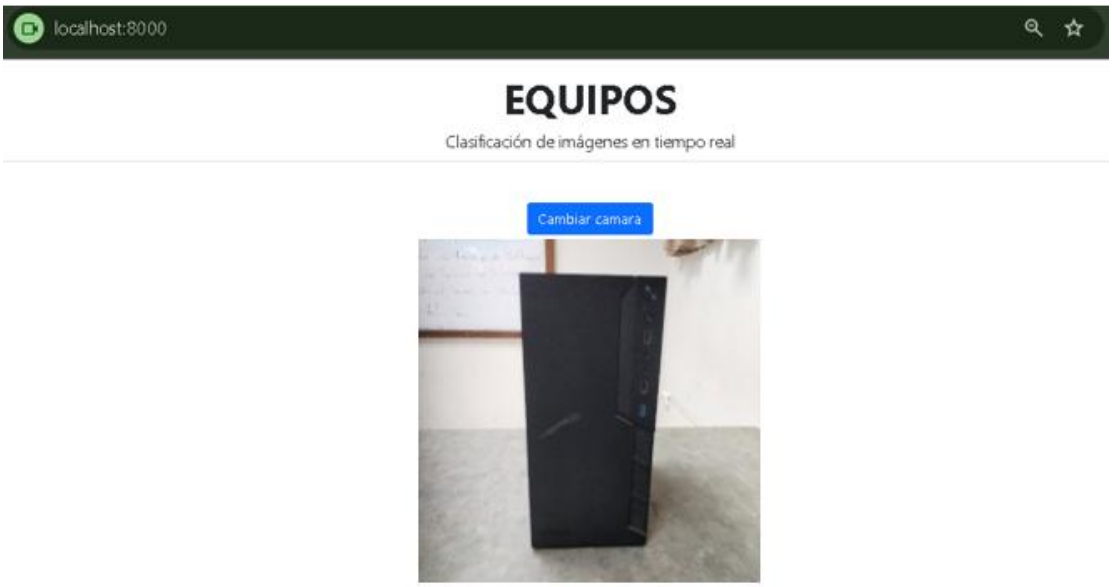


Figura 59 Interfaz de prueba con clasificador de imágenes

Complicaciones en la clasificación de imágenes

A través de diversas pruebas se pudo ir mejorando los resultados para un buen funcionamiento del sistema, al principio se le dificultaba clasificar los objetos, pero en base a las constantes pruebas fue mejorando su capacidad y así obteniendo mejores datos para poder así procesar de mejor manera la información y poder hacer una clasificación adecuada. Tal como se evidencia en una de las versiones presentadas en la figura correspondiente:

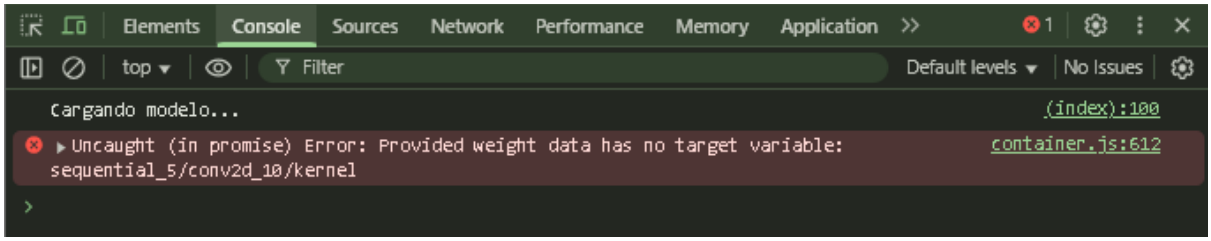


Figura 60 Error en la clasificación de imágenes

c) Acceso al sistema

Se validó correctamente el acceso restringido al sistema mediante este formulario de autenticación. Durante las pruebas funcionales el login cumplió su objetivo de permitir únicamente el ingreso del administrador responsable del monitoreo del laboratorio.

Utilizando el lenguaje HTML, se desarrolló el formulario, el cual es gestionado con el Framework Flask, permitiendo así direccionar en el index.html, el formulario principal y así poder visualizar en tiempo real el funcionamiento del monitoreo.

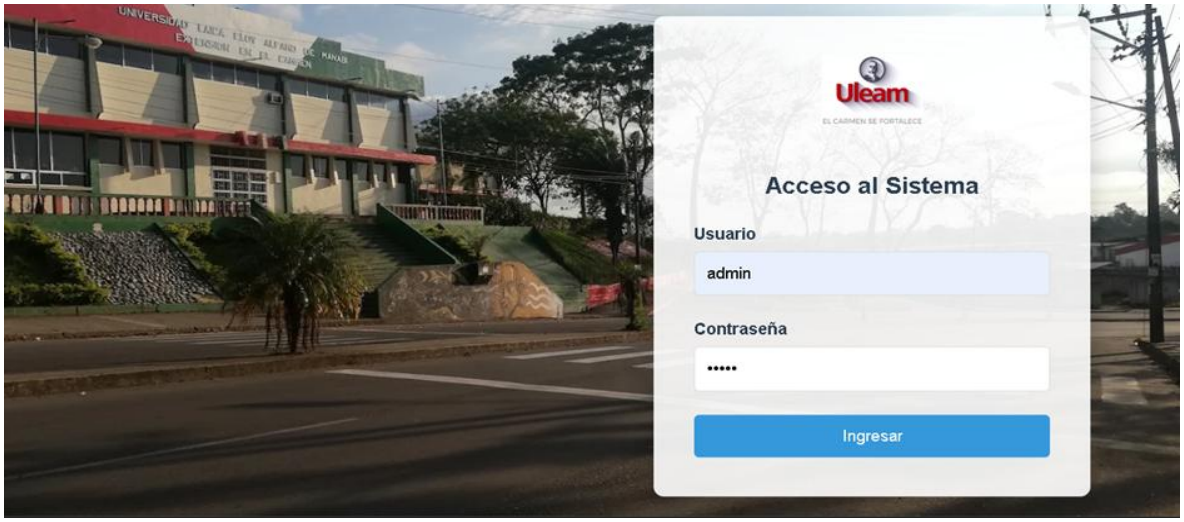


Figura 61 Login funcional validando el acceso al sistema durante las pruebas de monitoreo.

Durante el desarrolló de pruebas de implementación, se logró evidenciar una funcionalidad segura y eficiente, obteniendo así resultados positivos por parte del sistema computacional que permite monitorear. Permitiendo así ejecutarlos en los laboratorios de cómputo para un correcto control y monitoreo.

La figura permite conocer la estructura del sistema de una manera sencilla, mostrando una arquitectura flexible en la que se mantiene una cámara conectada dentro del laboratorio mediante un cable ethernet. Desde el laboratorio hasta el sitio dónde se encuentra el administrador, quien ingresa a la interfaz web utilizando un usuario y clave para login.

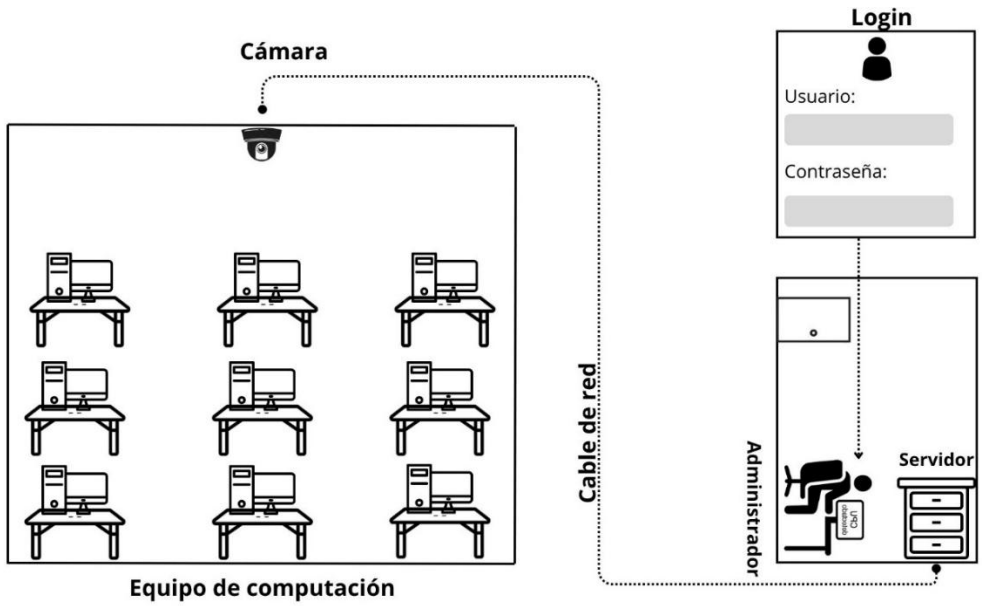


Figura 62 Esquema funcional del sistema de monitore

La figura muestra una interfaz visual simulada correspondiente al sistema desarrollado. Se encuentra estas características simuladas del sistema:

- La ventana superior muestra el video en tiempo real capturado desde el laboratorio.
- Tabla de inventario, es la lista de los equipos registrados como es el CPU y monitor.
- Tabla de estado actual, presenta información actualizada sobre los dispositivos detectados por el sistema.
- Burbuja de alertas, se visualiza los mensajes relacionados con los eventos importantes detectar en un entorno monitoreado.

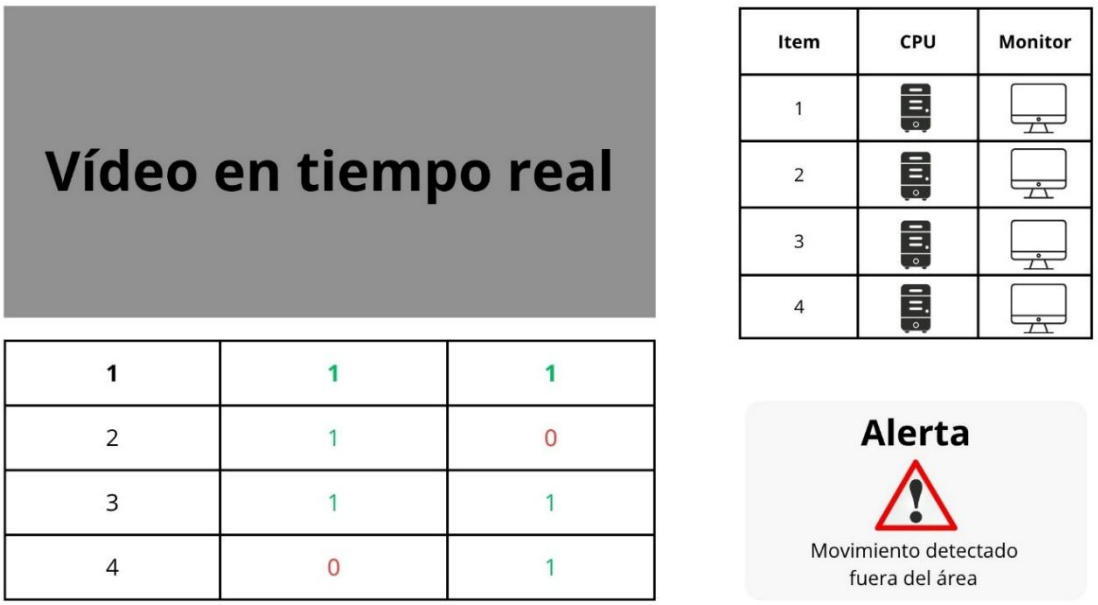


Figura 63 Simulación de la interfaz final del sistema

5.3 Interpretación objetiva

Una vez realizada las pruebas se obtuvieron los resultados correspondientes arrojados por el sistema, entre los aspectos que resaltan principalmente, encontramos los siguientes:

- Una mejora considerable en el tiempo de respuesta, obteniendo resultados con un 90% de mejora en el tiempo de detección de los objetos.
- Gracias a la integración de modelos como YOLOv8 y CNN, se obtuvo resultados positivos con un 80% en detección y un 70% en la clasificación de los equipos que están dentro del entorno de pruebas.

- Registro automático de accesos y actividades, mejorando la trazabilidad operativa frente a sistemas manuales tradicionales.
- La interfaz con login, nos permite brindar seguridad a la información obtenida mediante el sistema, y así no cualquier persona pueda acceder a dicha información.
- Se entreno con un aproximado de 2000 archivos de imágenes, dentro del data set con el fin de mejorar el reconocimiento y la precisión con la cual detecta y hace la clasificación de los objetos.
- Flexibilidad escalabilidad del sistema gracias a los códigos abierto como Flask, tensorflow y ultralytics YOLO.
- Facilidad de mantenimiento y actualización de manera independiente.
- Interfaz intuitiva para la administración de credenciales permitiendo visualizar en tiempo real el laboratorio.

Capítulo VI

6 CONCLUSIONES Y RECOMENDACIONES

6.1 Conclusiones

El establecimiento del sistema computacional utilizando redes neuronales convolucionales, permitió obtener información automatizada en la gestión del monitoreo, control de acceso y registro de eventos que se den en los laboratorios de computación de las carreras de Tecnologías de la información e Ingeniería de Software de la universidad Laica Eloy Alfaro de Manabí, extensión en El Carmen, permitiendo así mejorar la seguridad y mejora operativa de los laboratorios.

A través de la implementación de la metodología ágil Scrum, obtuvimos un desarrollo estructurado del proyecto. Con ello se obtuvo una estrategia más ágil de identificar y dar solución al problema, detectando los recursos a la vista y la información del control y manejo de los laboratorios de cómputo.

Mediante la investigación en la parte del desarrollo se logró obtener y organizar información sofisticada la cual permitió una mejor comprensión de como integrar las redes neuronales al sistema computacional para brindar un monitoreo y control de los laboratorios de computación. Logrando así justificar el uso de herramientas tales como Tensor Flow y YOLOv8 lo que permitió recabar información.

Aplicando el instrumento de encuesta, se pudo identificar la poca eficacia operativa que venían presentando los laboratorios de cómputo al momento de ser utilizados por estudiantes, en donde se encontraron anomalías como el no existir un registro de uso de los equipos, movimientos de los equipos del laboratorio sin una autorización o registro de dicha actividad, ocasionando inconvenientes al momento de hacer uso de los laboratorios.

Mediante el desarrollo del sistema computacional, el cual tiene implementado una red neuronal, con la capacidad de poder detectar objetos a través del uso de una cámara y así gestionar un control y monitoreo adecuado para la seguridad y funcionalidad de los equipos que integran el laboratorio. A través de una interfaz web podemos monitorear en tiempo real el entorno integral del laboratorio.

6.2 Recomendaciones

Se recomienda a los estudiantes de titulación seguir refinando los modelos y incorporando nuevas imágenes en el entorno real. Esto permite reducir los falsos positivos o negativos y mejorando la precisión del reconocimiento visual en diferentes ángulos e iluminación.

Se recomienda al responsable de laboratorios escalar el sistema en otros laboratorios de la universidad, al ser modular y códigos abiertos su adaptación a nuevos entornos solo requiere reentrenamiento de los modelos y ajustes menos en el hardware.

Se sugiere al tutor de titulación incluir en futuras versiones la detección de otros elementos importantes como mouses, teclados, reguladores de energía y proyectores, permitiendo un control más completo de todos los equipos presente en el laboratorio.

Se recomienda generar una base de datos, con la cual se permita realizar consultas a futuro por actividades detectadas y almacenadas desde su implementación hasta su actualidad con la finalidad de poder acceder a esa información en el momento que se requiera. Con ello se podrían generar reportes los cuales podrían ser notificados por diferentes medios de comunicación tales como el correo electrónico o un mensaje de texto hacia el administrador.

BIBLIOGRAFÍA

Alegre Ramos, M. d. (2019). *Sistemas operativos monopuesto.* Paraninfo, S.A.

Arias, J., Villasís, M. Á., y Miranda, M. G. (abril de 2016). El protocolo de investigación III: la población de estudio. *Rev Alerg Méx, III*(2), 201-206.

Bartis, P. (2004). *La tradicion popular y la investigacion de campo.* Library of Congress.

Bautista, N. P. (2022). *Proceso de la investigación cualitativa.* Bogotá, Colombia : El Manual Moderno.

Bernal. (2006). *Metodología de la Investigación para administracion, economía, humanidad y ciencias sociales.* S.A.

Bernal, C. (2003). *Compilación bibliográfica: Tomás de Aquino.* Universidad Santo Tomas.

Boyle, D. (2021). *50 grandes inventos que cambiaron el mundo.* Blume.

Cabello, P., Carbajosa, J. M., Gallego, J. C., Gómez, D., y Blanco, J. (2002). *Montaje y Mantenimiento de Sistemas y Componentes Informáticos.* Editex.

Calder, A. (2009). *Information Security based on ISO 27001/ISO 27002.* Van Haren.

Campos, M. (2023). *SISTEMAS OPERATIVOS, SISTEMAS INFORMÁTICOS Y LENGUAJES DE PROGRAMACIÓN.* Ra-Ma.

Castro, J., y Fitipaldo, J. (21 de Septiembre de 2014). *La encuesta como técnica de investigación, validez y confiabilidad*. UDE: https://ude.edu.uy/la-encuesta-como-tecnica-de-investigacion-validez-y-confiabilidad/

Cavaioni, M. (23 de Febrero de 2018). *Medium*. https://www.hackerrank.com/contests/ai-python/challenges/python-convolution-1

Cicero, E. I. (2018). *UTILIZACIÓN DE REDES NEURONALES CONVOLUCIONALES PARA LA DETECCIÓN DE TIPOS DE IMÁGENES.* ITBA.

Coello, B. X. (2021). *Aplicación de Redes Neuronales Recurrentes para Acelerar el Análisis no Lineal de Estrecuturas.* Tesis, Guayaquil.

Costa, P. (15 de Octubre de 2019). *Pocho Costa*. https://pochocosta.com/podcast/redes-neuronales-convolucionales-explicadas/

Costas, J. (2010). *Seguridad Informática.* Ra-Ma.

Costos, J. (2014). *Mantenimiento de la Seguridad en Sistemas Informáticos.* Ra-Ma.

Duque Domingo, J., Gómez Gacía, J., y Zalama Casanova, E. (2024). *Visión Artificial Componentes de los sistemas de visión y nuevas tendencias en Deep Learning.* Ra-Ma.

Flórez, R., y Fernández, J. (2008). *Las Redes Neuronales Artificiales.* Netbiblo.

Frascati, M. (2015). *Guía para la recopilación y presentación de información sobre la investigación y el desarrollo experimental.* OECD.

García, M. A. (2023). *Uso de inteligencia artificial como herramienta para predicción de accidentes en los hogares para niños menores de 12 años.* Tesis, Cuautitlán Izcalli.

HackerRank. (s.f.). *HackerRank*. HackerRank: https://www.hackerrank.com/contests/ai-python/challenges/python-convolution-1

Haque, K. N. (7 de Julio de 2024). *¿Qué es una red neuronal convolucional? CNN (aprendizaje profundo)*. https://techcommunity.microsoft.com/discussions/azure-ai-foundry-discussions/what-is-convolutional-neural-network-%E2%80%94-cnn-deep-learning/4184725

Liu, D. (30 de Noviembre de 2017). *Medium* . Medium : https://medium.com/@danqing/a-practical-guide-to-relu-b83ca804f1f7

López, J. A. (2021). *Deep Learning Teoría y aplicaciones.* Marcombo, S.L.

López, J. A. (2023). *Deep Learning: teoría y aplicaciones.* Marcombo.

Manchanda, C. (8 de Enero de 2022). *Medium*. Medium: https://manchandachitwan.medium.com/vegetable-image-classification-using-cnn-6f1d1be75cfb

Millstein, F. (2018). *Convolutional Neural Networks In Python.* Copyrigth.

Moreno. (2014). *Operaciones auxiliares de mantenimiento de sistemas MicroInformáticos.* Ra-Ma.

Moreno. (2023). *Sistema de clasificación de residuos mediante redes convolucionales.* Tesis, Catalunya.

Moreno, J. C., y Gonzáles, M. (2014). *Sistemas Informáticos y Redes Locales.* RA-MA.

Moreno, J. C., y Ramos, A. F. (2014). *Administración Hardware de un Sistema Informático.* RA-MA.

Moreno, J. C., y Ramos, A. F. (2014). *Mantenimiento del subsistema lógico de sistema informático.* RA-MA.

Narváez, A. M., y Vizcaya, J. V. (27 de Noviembre de 2016). *un nuevo sistema para el tratamiento de la información, cuya unidad básica de procesamiento está inspirada en la célula fundamental del sistema nervioso humano: la neurona.* https://dialnet.unirioja.es/descarga/articulo/6937134.pdf

Pérez Borrero, I., y Gegúndez Arias, M. E. (2021). *Deep Learning Fundamentos, teoría y aplicación.* UHU.ES.

Ramírez, C. M. (2023). *Programación de Inteligencia Artificial Curso Práctico.* Bogotá, Colombia: Ra-Ma.

Ramos, M. d. (2010). *Sistemas operativos monopuesto.* Paraninfo, S.A.

Raya, J., Raya, L., y Zurdo, J. (2014). *Sistemas Informáticos.* Madrid, Paracuellos de Jarama, España: RA-MA.

Rodríguez, E. (2005). *Metodología de la Investigación.* Univ. Autónoma de Tabasco.

Strickland, Müller, B., Reinhardt, J., y Strickland, M. (1995). *Neural Networks.* Springer.

Terán Pérez, D. M. (2014). *Administración Estratégica de la función Informática.* Alfaomega.

Torres, J. (2020). *Python Deep Learning Introducción práctica con Keras y Tensorflow 2.* Marcombo.

Valdivia Miranda, C. (2017). *Informática Industrial.* Paraninfo.

Van Der Haven, D. (2019). *IT Service Management: IS/IEC 20000.1:2018.* Van Haren.

Vasconcelos , S. J. (2018). *Introducción a la Computación.* Patria Educación. https://doi.org/9786077449539, 6077449539

ANEXOS

Anexo A *Aprobación de tema*

Outlook

DPGA | Titulación | Periodo 2024-2025(2) - Notificación de tutor asignado - TECNOLOGÍAS DE LA INFORMACIÓN 2022 (EL CARMEN)

Desde NOTIFICACIONES TITULACION <notificaciones.titulacion@uleam.edu.ec>
Fecha Sáb 20/07/2024 18:58
Para REASCOS PINCHAO RAUL SAED <raul.reascos@uleam.edu.ec>
CC CARRANZA MOREIRA TIFANI BELEN <e1314102649@live.uleam.edu.ec>; REASCOS PINCHAO RAUL SAED <raul.reascos@uleam.edu.ec>

Uleam
UNIVERSIDAD LAICA
ELOY ALFARO DE MANABÍ

Universidad Laica Eloy Alfaro de Manabí

Periodo 2024-2025(2) - Notificación de tutor asignado - TECNOLOGÍAS DE LA INFORMACIÓN 2022 (EL CARMEN)

Estimad@
Docente y Estudiante
Uleam

En cumplimiento de lo establecido en la Ley, el Reglamento de Régimen Académico y las disposiciones estatutarias de la Uleam, por medio de la presente se oficializa la dirección y tutoría en el desarrollo del Trabajo de Integración curricular / Trabajo de Titulación del siguiente estudiante:

Tema: SISTEMA INFORMÁTICO CON REDES NEURONALES PARA LA GESTIÓN DEL LABORATORIO DE COMPUTACIÓN DE LAS CARRERAS DE TI-SW EN ULEAM EXTENSIÓN EL CARMEN

Estado de aprobación: Aprobado

Tipo de titulación: Trabajo de Integración Curricular

Tipo de proyecto: Trabajo de Integración Curricular / Trabajo de titulación se articula con proyectos y programas de Investigación.

Apellidos y nombres del tutor asginado: REASCOS PINCHAO RAUL SAED

Apellidos y nombres del estudiante: CARRANZA MOREIRA TIFANI BELEN

Carrera: TECNOLOGÍAS DE LA INFORMACIÓN 2022 (EL CARMEN)

Periodo de inducción: Periodo 2024-2025(2)

Anexo B *Instrumento entrevista*

Esta entrevista forma parte del proyecto de titulación titulado Sistema Informático con Redes Neuronales para la Administración del Laboratorio 01 de la Universidad Laica Eloy Alfaro de Manabí Extensión "El Carmen".

Su propósito es recopilar información detallada sobre el entorno operativo del laboratorio, identificar necesidades específicas y detectar posibles limitaciones que impactan la gestión actual.

Preguntas para la entrevista

1. ¿Cuál es el estado actual de los equipos del laboratorio?
2. ¿Cómo gestiona el traslado de los equipos del laboratorio?
3. ¿Qué procedimiento existe para controlar el uso del laboratorio?
4. ¿Qué problemas comunes ha encontrado al supervisar los equipos?
5. ¿Cómo evalúa la efectividad de las medidas de seguridad actual?
6. ¿Cómo se registra el uso de los equipos?
7. ¿Cuál es su criterio de un sistema de supervisión implementaría en el laboratorio?
8. ¿Cuál ha sido la frecuencia observada de desconexión de equipos del laboratorio?
9. ¿Qué mejoras implementaría para optimizar la seguridad de los equipos?

Anexo C Instrumento encuesta

Anexo D *Fotografías*

- Se muestra la Cámara TP-Link VIGI C330I (2.8mm), fue configurada para captar video en tiempo real y conectada al sistema mediante el cable Ethernet.

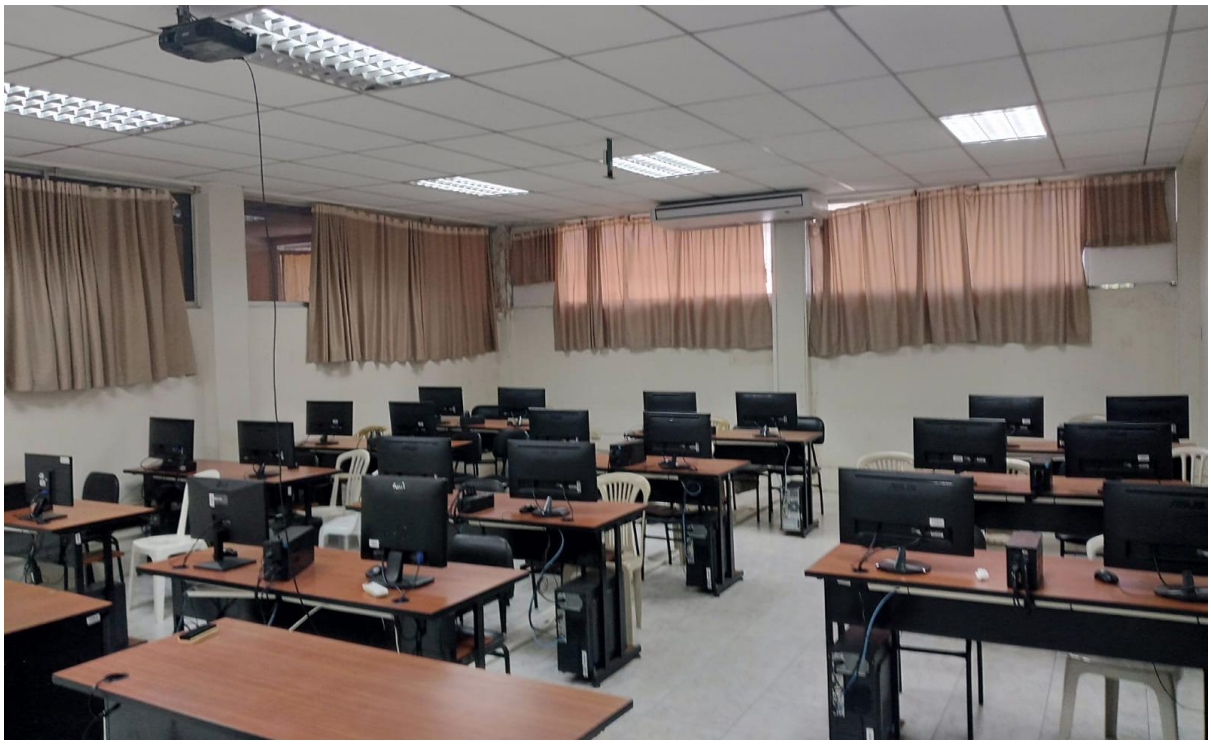
- Disco Duro Hp 627114-002 De 300gb Sas Para Servidor 2.5" Sff 6g 15k Hot Plug

- El Cable Ethernet de 30 metros de longitud, utilizado para conectar la cámara al servidor ubicado en el aula de administración.

- Se observa una Batería PKCELL PK1290 12V 9.0Ah, utilizada para mantener el servidor.

- Dos Conectores RJ45, utilizados para terminaciones de cable Ethernet.

- Se observa el Laboratorio de Cómputo 01 perteneciente a la extensión El Carmen de la ULEAM.



Anexo E *Certificado de coincidencia académica*

CERTIFICADO DE ANÁLISIS
magister

CARRANZA MOREIRA TIFANI BELÉN
Proyecto Final vc3

8% Textos sospechosos

- < 1% **Similitudes**
  < 1% similitudes entre comillas
  0% entre las fuentes mencionadas
- 5% **Idiomas no reconocidos**
- 3% **Textos potencialmente generados por la IA**

Nombre del documento: CARRANZA MOREIRA TIFANI BELÉN Proyecto Final vc3.pdf
ID del documento: 257902dd36daa6d7fe4aa618853b0295bc14c96f
Tamaño del documento original: 6,02 MB

Depositante: RAUL REASCOS PINCHAO
Fecha de depósito: 18/8/2025
Tipo de carga: interface
fecha de fin de análisis: 18/8/2025

Número de palabras: 20.245
Número de caracteres: 139.719

Ubicación de las similitudes en el documento:

Fuentes principales detectadas

N°	Descripciones	Similitudes	Ubicaciones	Datos adicionales
1	upcommons.upc.edu https://upcommons.upc.edu/bitstream/handle/2117/392008/memoria.pdf?sequence=2	< 1%		Palabras idénticas: < 1% (44 palabras)
2	www.ptolomeo.unam.mx http://www.ptolomeo.unam.mx:8080/xmlui/bitstream/handle/132.248.52.100/3735/Tesis.pdf...	< 1%		Palabras idénticas: < 1% (21 palabras)

Fuente con similitudes fortuitas

N°	Descripciones	Similitudes	Ubicaciones	Datos adicionales
1	www.segu-info.com.ar | Seguridad Física | www.segu-info.com.ar https://www.segu-info.com.ar/fisica/#:~:text=Así la Seguridad Física consiste en la "a los recu...	< 1%		Palabras idénticas: < 1% (20 palabras)

Fuentes mencionadas (sin similitudes detectadas)

Estas fuentes han sido citadas en el documento sin encontrar similitudes.

1 https://manchandachitwan.medium.com/vegetable-image-classification
2 https://medium.com/@danqing/a-practical-guide-to-relu-b83ca804f1f7
3 https://en.wikipedia.org/wiki/Convolutional_neural_network
4 https://www.atlassian.com/es/agile/scrum/sprints

GLOSARIO

A

Administración de centros de cómputo: Gestiona y mantiene los recursos tecnológicos en el laboratorio de computación, incluyendo hardware, software y seguridad.

Algoritmo: Conjunto de instrucciones definidas para resolver problemas o tareas especifica en programación.

Análisis de datos: Proceso que implica examinar, depurar y transformar datos con el fin de extraer información útil y tomar decisiones fundamentadas.

API: Interfaz de programación de aplicaciones, es un conjunto protocolo y herramientas para construir software y aplicaciones.

B

Backpropagation: Algoritmo de aprendizaje automático que ajusta los pesos en una red neuronal a través del cálculo.

Base de datos: Sistema organizado que almacena y gestiona datos de forma electrónica, facilitando su acceso y manipulación.

C

Capa convolucional: Componente de las redes neuronales convolucionales que aplica filtros para extraer características relevantes de las imágenes.

Clasificación de imágenes: Proceso mediante el cual se asigna una etiqueta o categoría a una imagen.

Redes Neuronal Convolucional CNN: arquitectura de la red especialmente diseñada para el procesamiento de datos con estructura espacial.

D

Dataset: Conjunto estructurado de datos, utilizado para entrenar, validar y evaluar modelos de aprendizaje automático.

Detección de objetos: Técnica para identificar y localizar objetos dentro de las imágenes o videos.

E

Encuesta: Método de recopilación de datos cuantitativos a través de preguntas estandarizadas, aplicado a una muestra representativa de la población.

Entrevista: Estrategia cualitativa de recolección de información basada en un diálogo estructurado o semiestructurado entre el entrevistador y el entrevistado.

F

Flask: Marco de trabajo ligero para desarrollar aplicaciones web en Python.

Función de activación: Función matemática en una neurona que determina su salida basada en las entradas.

G

Google Colab: Entorno de programación en la nube que permite ejecutar código de Python con acceso a GPU gratuito.

H

Hardware: Componente físico de un sistema informático.

I

Inteligencia artificial (IA), es el campo de la información que busca crear sistemas capaces de realizar tareas.

ISO/IEC 27001: Norma internacional para la gestión de seguridad de la información.

L

LabelImg: Herramienta para etiquetar imágenes para realizar un dataset y utiliza el entrenamiento en modelo Yolo.

M

Machine Learning: subcampo de la IA que permite a los sistemas aprender y mejorar automáticamente los datos.

N

Neurona artificial: Unidad básica de una red neuronal que procesa información mediante entradas, pesos y una función de activación.

O

OpenCV: Biblioteca de código abierto para la visión por computadora y procesamiento de imágenes.

ONNX (Open Neural Network Exchange): Formato estándar para representar modelos de machine learning.

P

Precisión: Métrica que evalúa la exactitud de un modelo en la clasificación o detección de objetos.

Python: Lenguaje de programación utilizado en el desarrollo el sistema.

R

ReLU (Rectified Linear Unit): Función de activación común en redes neuronales.

S

Software: Componentes lógicos de un sistema informático, como programas y aplicaciones.

Sprint: Iteración en Scrum que define ciclos de trabajo con objetivos específicos.

T

TensorFlow: Código abierto para machine learning, desarrollada por Google

Transfer Learning: Técnica que reutiliza un modelo preentrenado para nuevas tareas.

V

Visión por computación: campo de la IA que permite a la computadora interpretar imágenes o videos.

Y

YOLOv8 (You Look Once): Modelo de detección de objetos en tiempo real, conocido por su velocidad y precisión.