**INSTITUTO TÉCNICO NACIONAL DE COMERCIO**

**“FEDERICO ALVAREZ PLATA” NOCTURNO**

**CARRERA: SISTEMAS INFORMÁTICOS**

Diagrama

Descripción generada automáticamente

**SISTEMA INTELIGENTE DE CLASIFICACIÓN DE RESIDUOS BASADO EN REDES NEURONALES Y UN PROTOTIPO DE BASUREROS AUTOMATIZADOS UTILIZANDO COMPONENTES DE ARDUINO, DESTINADO A LA EDUCACIÓN INICIAL: CASO DE ESTUDIO UNIDAD EDUCATIVA NÓRDICO BOLIVIANO”**

**Proyecto de grado para optar al Título de Técnico Superior en Sistemas Informáticos**

**Egr.: Quispe Flores Cinthia**

**Tutor: MBA Lic. Escalera Cruz David**

Cochabamba – Bolivia

Agosto 2024

**DEDICATORIA**

Este proyecto está dedicado a todas las personas que han sido una fuente de inspiración y apoyo. Especialmente, a mi mamá, por su apoyo incondicional; a mis hijos, cuyo futuro me motiva a seguir trabajando por un mundo mejor; y a mi esposo, por su constante apoyo, comprensión y aliento en cada etapa de este proyecto. A mis profesores y mentores, por su guía y sabiduría, sin la cual este proyecto no habría sido posible. Y, finalmente, a las futuras generaciones, con la esperanza de que este esfuerzo contribuya a un futuro más verde y consciente, donde la gestión de residuos sea una práctica habitual y responsable en nuestra sociedad.

**AGRADECIMIENTOS**

Quiero expresar mi más sincero agradecimiento a mis profesores y mentores, cuya guía y sabiduría han sido fundamentales para la culminación de este trabajo. En especial, deseo agradecer al TUTO MBA, Lic. Escalera Cruz David, por su inigualable dedicación y por compartir su vasto conocimiento, colaboración en este proyecto. Sin su contribución, este logro no hubiera sido posible..

**TABLA DE CONTENIDOS**

[CAPÍTULO 1 PLANTEMIENTO DEL PROBLEMA 1](#_Toc180161277)

[1.1. Diagnóstico y justificación 1](#_Toc180161278)

[1.1.1. Diagnostico 1](#_Toc180161279)

[1.1.2. Justificación 2](#_Toc180161280)

[1.2. Planteamiento y formulación del problema técnico/tecnológico 2](#_Toc180161281)

[1.3. Objetivos 4](#_Toc180161282)

[1.3.1. General 4](#_Toc180161283)

[1.3.2. . Alcances 5](#_Toc180161284)

[1.3.3. Límites 7](#_Toc180161285)

[1.4. Enfoque metodológico 7](#_Toc180161286)

[1.4.1. Métodos 7](#_Toc180161287)

[1.4.2. Técnicas 8](#_Toc180161288)

[CAPÍTULO 2 Marco Teórico Conceptual 9](#_Toc180161289)

[2.1 Sistema de Informacion 9](#_Toc180161290)

[2.2 Ingenieria de software 9](#_Toc180161291)

[2.2.1 Metodologias 10](#_Toc180161292)

[2.2.2 RUP 11](#_Toc180161293)

[2.2.3 UML 11](#_Toc180161294)

[2.2.4 Diagrama Estructural 12](#_Toc180161295)

[2.3 Backend y Frontend 12](#_Toc180161296)

[2.4 Frontend 13](#_Toc180161297)

[2.5 Backend 14](#_Toc180161298)

[2.5.1 Python 14](#_Toc180161299)

[2.5.2 Red Neuronal 15](#_Toc180161300)

[2.5.5 Roboflow 16](#_Toc180161301)

[2.5.4 Google Colab 16](#_Toc180161302)

[2.5.4 Pycharm 17](#_Toc180161303)

[2.5.6 Ultralytics 17](#_Toc180161304)

[2.5.7 CUDA 18](#_Toc180161305)

[2.5.7 Deep Learning 18](#_Toc180161306)

[2.5.8 Open CV 19](#_Toc180161307)

[2.5.9 Inteligencia Artificial (IA) 19](#_Toc180161308)

[3 ELEMENTOS ELECTRONICOS PARA EL PROYECTO 19](#_Toc180161309)

[CAPÍTULO 3 Propuesta de Innovación o Solución del Problema 19](#_Toc180161310)

[3.1. Análisis de requerimientos 19](#_Toc180161311)

[3.1.1. Funcionales 20](#_Toc180161312)

[3.1.2 No funcionales 20](#_Toc180161313)

[3.2. Diagramas de casos de uso 22](#_Toc180161314)

[3.2.1 Clasificación automática de residuos 22](#_Toc180161315)

[3.2.2 Interacción con el Prototipo de Hardware 23](#_Toc180161316)

[3.2.3 Apertura automática de contenedores e Hardware 24](#_Toc180161317)

[3.2.4 Captura de imágenes en tiempo real 24](#_Toc180161318)

[3.2.5 Entrenamiento y actualización del modelo de IA 26](#_Toc180161319)

[3.3. Diagrama de secuencias 27](#_Toc180161320)

[3.3.1 Clasificación automática de residuos 27](#_Toc180161321)

[3.3.2 Interacción con el Prototipo de Hardware 28](#_Toc180161322)

[3.3.3 Apertura automática de contenedores e Hardware 29](#_Toc180161323)

[3.3.4 Captura de imágenes en tiempo real 30](#_Toc180161324)

[3.3.5 Entrenamiento y actualización del modelo de IA 30](#_Toc180161325)

[3.4. Diagrama de clases 31](#_Toc180161326)

[3.5. Diagrama Relacional 32](#_Toc180161327)

[3.6. Arquitectura del Sistema 33](#_Toc180161328)

[3.6.1 Base Arquitectónica 33](#_Toc180161329)

[3.6.2 Recursos de Hardware 33](#_Toc180161330)

[3.6.3 Software de codificación 34](#_Toc180161331)

[3.7. Pruebas de calidad 35](#_Toc180161332)

[3.7.1 Introducción 35](#_Toc180161333)

[3.7.2 Pruebas de Software 35](#_Toc180161334)

[3.7.3 Pruebas de hardware 35](#_Toc180161335)

[3.7.4 Pruebas de integración 36](#_Toc180161336)

[3.7. 5 Pruebas de Sistema 36](#_Toc180161337)

[3.8. Estimación de costos 38](#_Toc180161338)

[CAPÍTULO 4 CONCLUSIONES Y RECOMENDACIONES 23](#_Toc180161339)

[4.1. Conclusiones 23](#_Toc180161340)

[4.2. Recomendaciones 24](#_Toc180161341)

**ÍNDICE DE TABLAS**

[Tabla 1 Requerimientos Funcionales 20](#_Toc180535169)

[Tabla 2 Requerimientos no funcionales 21](#_Toc180535170)

**ÍNDICE DE FIGURAS**

[Figura 1Árbol de Problemas 3](#_Toc180535145)

[Figura 2 Capas de Ingeniería 10](#_Toc180535146)

[Figura 3 Arquitectura 13](#_Toc180535147)

[Figura 4 Diagrama de casos de usos clasificación automática 22](#_Toc180535148)

[Figura 5 Diagrama de casos de usos Interacción con el prototipo 23](#_Toc180535149)

[Figura 6 Diagrama de casos de usos apertura automática de contenedores 24](#_Toc180535150)

[Figura 7 Captura de imágenes en tiempo real 25](#_Toc180535151)

[Figura 8 Diagrama de casos de usos Entrenamiento del modelo de IA 26](#_Toc180535152)

[Figura 9 Diagrama de secuencias clasificación automática de residuos 27](#_Toc180535153)

[Figura 10 Diagrama de secuencias Interacción con el prototipo 28](#_Toc180535154)

[Figura 11 Diagrama de secuencias Apertura de contenedores 29](#_Toc180535155)

[Figura 12 Diagrama de secuencias Captura de imágenes en tiempo real 30](#_Toc180535156)

[Figura 13 Diagrama de secuencias Entrenamiento del modelo IA 30](#_Toc180535157)

[Figura 14 Diagrama de clases 31](#_Toc180535158)

[Figura 15 Diagrama relacional 32](#_Toc180535159)

[Figura 16 Base Arquitectónica 33](#_Toc180535160)

[Figura 17 Estimación de costos 1 39](#_Toc180535161)

[Figura 18 Estimación de costos 2 40](#_Toc180535162)

[Figura 19 Estimación de costos 2 41](#_Toc180535163)

[Figura 20 Estimación de costos 3 41](#_Toc180535164)

**RESUMEN**

La gestión eficiente de residuos es esencial en sistemas educativos avanzados, como en Japón, donde la tecnología no solo aporta beneficios económicos, sino también fomenta buenos hábitos en los estudiantes. En contraste, Bolivia enfrenta desafíos significativos en este ámbito. Para abordar esta problemática, se propone desarrollar un sistema web con inteligencia artificial y tecnología Arduino que clasifique visualmente residuos orgánicos, plásticos y papeles, facilitando su depósito adecuado. Este proyecto busca promover buenos hábitos desde la niñez y fomentar una cultura de gestión de residuos a corto, mediano y largo plazo, contribuyendo a la mitigación del impacto ambiental y a un futuro más sostenible.

**INTRODUCCIÓN**

En la actualidad, la gestión de residuos se ha convertido en una prioridad mundial debido a sus consecuencias ecológicas y su impacto en el bienestar público. Con este trasfondo, se propone el desarrollo de un Sistema clasificador de residuos con inteligencia artificial y aplicación de Arduino, con un enfoque de estudio en la Unidad Educativa Nórdico Boliviano. El objetivo principal de este proyecto es educar y concienciar sobre la importancia de la clasificación adecuada de los residuos.

En este proyecto de grado, se aborda la problemática de la gestión de residuos en la educación inicial a través de la implementación de un sistema clasificador de residuos con inteligencia artificial y la integración de Arduino. Este sistema contará con módulos dedicados a la gestión de usuarios, datos, reportes, entre otros aspectos fundamentales. El propósito principal es contribuir a una educación ambiental sólida y consciente.

Implementar un proyecto que clasifique residuos mediante IA y la aplicación de Arduino para la educación inicial en una institución educativa, proporciona una oportunidad para educar a los estudiantes sobre la importancia del reciclaje y la protección del medio ambiente.

Al analizar este caso de estudio, se consideran problemas como la gestión inadecuada de residuos, la escasez de educación ambiental y la falta de innovaciones tecnológicas. La limitada educación ambiental puede contribuir a prácticas inadecuadas en el manejo de residuos. La implementación de un proyecto de esta naturaleza en una institución educativa ofrece la oportunidad de educar a los estudiantes sobre la importancia del reciclaje y la preservación del medio ambiente.

Se pretende mejorar la situación actual en términos de gestión de residuos y conciencia ambiental, así también se sentará las bases para un futuro más sostenible al inspirar a las generaciones futuras a adoptar cultura y prácticas más responsables con el medio ambiente.

Utilizaremos técnicas de programación e inteligencia artificial, como el aprendizaje automático y el procesamiento de imágenes, así como herramientas de desarrollo de software como Python para la programación de algoritmos y la integración con Arduino, y tecnologías para la implementación del sistema en línea. El proceso de ejecución del proyecto se llevará a cabo de manera iterativa, con reuniones regulares de seguimiento y ajustes según sea necesario para garantizar el éxito del proyecto con la metodología Ágil.

**CAPÍTULO I**

**PLANTEAMIENTO DEL PROBLEMA**

# 

# PLANTEMIENTO DEL PROBLEMA

## Diagnóstico y justificación

## Diagnostico

En un contexto global de creciente conciencia ambiental, la clasificación efectiva de residuos se convirtió en una prioridad para instituciones de diversos ámbitos. La necesidad de implementar sistemas que facilitaran la clasificación y el manejo adecuado de residuos se hacía cada vez más evidente en todo el mundo, con el fin de reducir el impacto ambiental y promover prácticas sostenibles.

Para el proyecto "Sistema Web Clasificador de Residuos con Inteligencia Artificial y Aplicación de Arduino, para el caso de estudio en cuestión, se llevó a cabo una entrevista con el director de la U.E. Esta entrevista ha sido diseñada para recopilar información clave sobre las necesidades, percepciones y expectativas relacionadas con la gestión de residuos en la institución. A continuación, se han resumido los puntos más relevantes discutidos durante esta entrevista.

Los estudiantes tienen la tendencia a desechar residuos en cualquier lugar, en vez de utilizar los contenedores designados. Además, cuando estos contenedores suelen estar llenos, los residuos terminan siendo arrojados al suelo, lo que causa incomodidades y promueve la ausencia de buenos hábitos en cuanto a gestión de residuos.

Por otra parte, se indago al director sobre su percepción por el proyecto, demostrando comprender la importancia del reciclaje a tiempo de reconocer los beneficios potenciales de la tecnología avanzada en la gestión de residuos. Además, se mostró dispuesto a recibir capacitación y sugirió la realización de actividades adicionales relacionadas con el medio ambiente en la escuela.

## Justificación

Este proyecto tiene como objetivo principal abordar las deficiencias identificadas mediante la implementación de un sistema innovador. Este sistema integrará inteligencia artificial y tecnología de Arduino para mejorar la clasificación precisa y la gestión efectiva de los residuos en la institución educativa.

Al enseñar a los estudiantes desde una edad temprana la importancia de la clasificación adecuada de residuos, se promueve una cultura de respeto y responsabilidad hacia el medio ambiente desde el inicio de su educación. Este enfoque no solo beneficia a la institución educativa, sino que también tiene un impacto positivo en la comunidad circundante.

El proyecto ofrece la oportunidad de explorar nuevas tecnologías y enfoques en el campo de la clasificación de residuos. Esta exploración puede contribuir al avance del conocimiento en esta área y servir como modelo para futuras iniciativas similares en otras instituciones educativas.

La implementación de este proyecto tiene el potencial de generar una serie de beneficios tanto para la Unidad Educativa Nórdico Boliviano como para la comunidad en general. Contribuirá a la promoción de prácticas más sostenibles y a la formación de ciudadanos responsables con el medio ambiente.

Adicionalmente, este proyecto proporciona una oportunidad única para involucrar a los estudiantes en un aprendizaje práctico al integrar esta tecnología en el plan de estudios. Esto les permitirá adquirir habilidades relevantes en tecnología y gestión ambiental desde una edad temprana, fomentando la conciencia ambiental y las prácticas sostenibles desde una etapa inicial, lo que resultará en un impacto positivo a largo plazo en la comunidad educativa y en la sociedad en general.

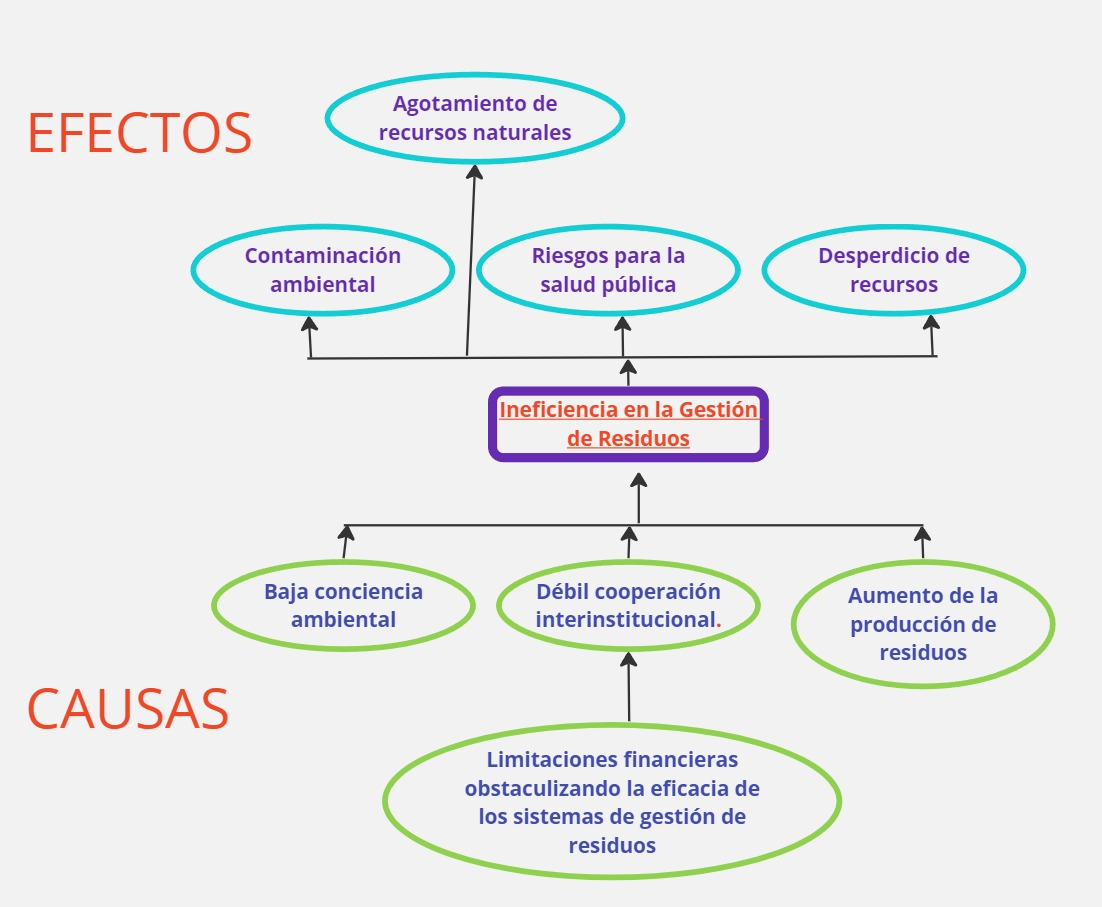
## Planteamiento y formulación del problema técnico/tecnológico

La Unidad Educativa Nórdico Boliviano enfrenta dificultades en la clasificación eficiente de los residuos, lo que resulta en una clasificación inadecuada y un manejo ineficiente de los mismos. Esta situación afecta tanto al entorno escolar como al medio ambiente circundante.

Se requiere un sistema innovador que integre tecnología, inteligencia artificial y aplicación de Arduino para mejorar la clasificación precisa y la gestión adecuada de los residuos en la institución educativa. Actualmente, no existe una solución tecnológica que aborde de manera efectiva estas necesidades específicas.

El éxito del proyecto se medirá por la efectividad del sistema en mejorar la gestión de residuos en la Unidad Educativa Nórdico Boliviano, la satisfacción de los usuarios con la facilidad de uso y funcionalidad del sistema, así como la reducción del impacto ambiental negativo causado por una gestión ineficiente de residuos.

Figura 1Árbol de Problemas



Fuente: Elaboración propia

- La falta de sistemas y prácticas efectivas para la recolección, tratamiento, reciclaje y disposición final de los residuos, lo que resulta en una acumulación excesiva de desechos, contaminación ambiental y riesgos para la salud pública.

- La escasa sensibilización, comprensión o interés respecto a los problemas ambientales y sus efectos, lo que resulta en la adopción de conductas y decisiones que promueven la degradación ambiental y agravan los problemas relacionados con el medio ambiente.

**¿Cómo resolver el problema de la clasificación inadecuada de residuos en la Unidad Educativa Nórdico Boliviano, donde los estudiantes no utilizan los contenedores designados, causando incomodidades y falta de clasificación de los desechos?**

## Objetivos

### General

Desarrollar un sistema inteligente de clasificación de residuos basado en redes neuronales y un prototipo de basureros automatizados utilizando componentes de Arduino, destinado a la educación inicial: Caso de estudio Unidad Educativa Nórdico Boliviano”.

* Desarrollar un módulo de visión artificial que capture imágenes de objetos en tiempo real utilizando una webcam, con la calidad suficiente para el proceso de entrenamiento y desarrollo.
* Implementar un algoritmo de clasificación de objetos basado en técnicas de machine learning, que permita identificar y categorizar los objetos según su tipo (plástico, papel, orgánico).
* Entrenar un modelo IA para estudiar los resultados preliminares en términos de precisión y detección.
* Diseñar y programar un prototipo funcional que utilice Arduino para controlar un actuador (Servomotor) que separe el residuo según la clasificación realizada por el sistema de visión artificial.
* Integrar el sistema de visión artificial con el microcontrolador Arduino para procesar la información en tiempo real y envíe señales correspondientes al clasificador de residuos.
* Realizar pruebas de precisión y eficiencia del sistema para evaluar la exactitud de la identificación de objetos y la clasificación del tipo del residuo, ajustando parámetros del modelo y hardware según sea necesario.

### . Alcances

* **Desarrollar un módulo de visión artificial que capture imágenes de objetos en tiempo real utilizando una webcam, con la calidad suficiente para el proceso de entrenamiento y desarrollo.**
* Se implementará un sistema de captura de imágenes en tiempo real utilizando una cámara web que proporcione la calidad visual necesaria para realizar el entrenamiento del modelo de IA. Este módulo deberá procesar las imágenes capturadas para que sean adecuadas para los algoritmos de clasificación, asegurando que cada objeto sea capturado con suficiente claridad para su reconocimiento y posterior procesamiento. Este sistema será el componente central para la adquisición de datos visuales en el proceso de clasificación de residuos.
* **Implementar un algoritmo de clasificación de objetos basado en técnicas de machine learning, que permita identificar y categorizar los objetos según su tipo (plástico, papel, orgánico).**
* Se desarrollará un algoritmo de clasificación entrenado con técnicas de aprendizaje automático que permita reconocer y categorizar los objetos en las categorías predefinidas (plástico, papel, orgánico). Este algoritmo se alimentará de los datos capturados por el sistema de visión artificial, identificando patrones en las características visuales de los objetos que lo guiarán para asignarles una clasificación adecuada. Esto permitirá automatizar el proceso de separación de residuos en las categorías correspondientes.
* **Entrenar un modelo IA para estudiar los resultados preliminares en términos de precisión y detección.**
* Se llevará a cabo un proceso de entrenamiento de un modelo de inteligencia artificial (IA) utilizando un conjunto de datos de imágenes previamente recopiladas y etiquetadas. El objetivo es mejorar la precisión del modelo en la detección y clasificación de los residuos. Durante este proceso, se evaluarán los resultados preliminares, ajustando los parámetros del modelo según sea necesario, con el fin de optimizar su capacidad de reconocer y clasificar correctamente cada tipo de residuo.
* **Diseñar y programar un prototipo funcional que utilice Arduino para controlar un actuador (Servomotor) que separe el residuo según la clasificación realizada por el sistema de visión artificial.**
* Se creará un prototipo funcional que integre el microcontrolador Arduino con un actuador, como un servomotor, que será responsable de la separación física de los residuos. Este sistema recibirá la información procesada por el módulo de visión artificial y clasificará el residuo identificado. Una vez clasificado, el servomotor moverá el contenedor adecuado para que el residuo se deposite en la categoría correcta, automatizando así el proceso de gestión de residuos.
* **Integrar el sistema de visión artificial con el microcontrolador Arduino para procesar la información en tiempo real y envíe señales correspondientes al clasificador de residuos.**
* Este objetivo se enfocará en la integración del sistema de visión artificial con el microcontrolador Arduino. Se programará la comunicación entre ambos sistemas para que la información procesada por la cámara sea transmitida en tiempo real al Arduino, el cual ejecutará las acciones necesarias para mover el actuador y clasificar los residuos. Este proceso de comunicación permitirá una operación fluida y automática del sistema de clasificación.
* **Realizar pruebas de precisión y eficiencia del sistema para evaluar la exactitud de la identificación de objetos y la clasificación del tipo del residuo, ajustando parámetros del modelo y hardware según sea necesario.**
* Se realizarán una serie de pruebas con el sistema completo para evaluar su rendimiento en términos de precisión y eficiencia en la identificación y clasificación de residuos. Se medirán aspectos como el tiempo de respuesta del sistema, la tasa de éxito en la correcta clasificación de residuos, y el funcionamiento del hardware. En función de los resultados obtenidos, se realizarán los ajustes necesarios tanto en los parámetros del modelo de IA como en la configuración del hardware para mejorar el rendimiento del sistema.

### Límites

A continuación, se establecen los límites del sistema para el proyecto de desarrollo del sistema clasificador de residuos. Estos límites delimitan las funcionalidades y características que no serán consideradas dentro del alcance del sistema. Esto permite al equipo enfocarse en las áreas cruciales para la gestión y clasificación efectiva de residuos en la unidad educativa Nórdico Boliviano.

* El sistema no abordará la gestión logística de la recolección y disposición final de los residuos clasificados.
* No se proporcionará una solución física para la recolección de residuos en la Unidad Educativa Nórdico Boliviano.
* El sistema no garantizará la precisión del proceso de clasificación de residuos en todas las situaciones posibles.

## Enfoque metodológico

### Métodos

Para abordar el desarrollo del sistema clasificador de residuos, se adoptará la metodología ágil Scrum (Simplify, Collaborate, Results, Utilize, Momentum), que es una estructura de trabajo que permite la colaboración efectiva y la entrega incremental de productos en un entorno dinámico y multifuncional. Scrum es ideal para proyectos que requieren flexibilidad y capacidad de respuesta rápida a cambios y la retroalimentación continua, como es el caso de la creación de un sistema que combina tecnología, inteligencia artificial y hardware.

Para el desarrollo del sistema, se aplicarán los siguientes métodos:

• Método Deductivo: Se utilizará el método deductivo para crear un algoritmo de clasificación de residuos, basado en patrones observables, mejorando así la gestión de residuos en la unidad educativa Nórdico Boliviano. La aplicación de principios teóricos adaptados al contexto educativo facilitará la identificación precisa de residuos en los contenedores. Este enfoque promoverá una gestión ambiental eficiente, fomentando prácticas sostenibles en la comunidad educativa.

• Método Analítico: El análisis estadístico de estos datos permitirá identificar patrones y relaciones entre las características de los residuos y su clasificación adecuada. Este enfoque analítico nos ayudará a comprender mejor el proceso de clasificación de residuos y a desarrollar un sistema web clasificador más preciso y eficiente para la unidad educativa Nórdico Boliviano.

• Enfoque cualitativo: El enfoque cualitativo se refiere a un método de investigación centrado en comprender los significados, experiencias y perspectivas de las personas mediante técnicas como entrevistas, grupos focales y observaciones, en lugar de datos numéricos. Este enfoque busca obtener una comprensión profunda y contextualizada de un fenómeno específico, interpretando las experiencias subjetivas de los participantes y adaptándose a los hallazgos emergentes para proporcionar información detallada y rica sobre el tema estudiado.

### Técnicas

Para promover la calidad del sistema web clasificador de residuos, se emplearán las siguientes técnicas para garantizar su funcionamiento óptimo y confiable:

Entrevista: Se aplicará la técnica de entrevista para obtener información relevante de primera mano sobre las necesidades y procesos relacionados con la clasificación de residuos. Se llevará a cabo una entrevista con el director de la unidad educativa para evaluar los desarrollos actuales y los problemas más frecuentes. La entrevista proporcionará datos cualitativos, así como expectativas sobre el nuevo sistema propuesto y sugerencias para su mejora. Estas respuestas orientarán el diseño y desarrollo del sistema, asegurando su alineación con las necesidades y expectativas de los usuarios finales.

Guía de entrevista: Emplearemos una guía diseñada específicamente para dirigir y estructurar las preguntas durante las entrevistas con el director de la unidad educativa. Esta guía facilitará la exploración detallada de los procedimientos actuales de gestión de residuos y posibles áreas de mejora. Las respuestas se registrarán en la guía correspondiente, tomando notas personales durante el proceso de entrevista, y se grabará la entrevista con preguntas abiertas.

**CAPÍTULO II**

**MARCO TEÓRICO CONCEPTUAL**

# Marco Teórico Conceptual

## 2.1 Sistema de Información

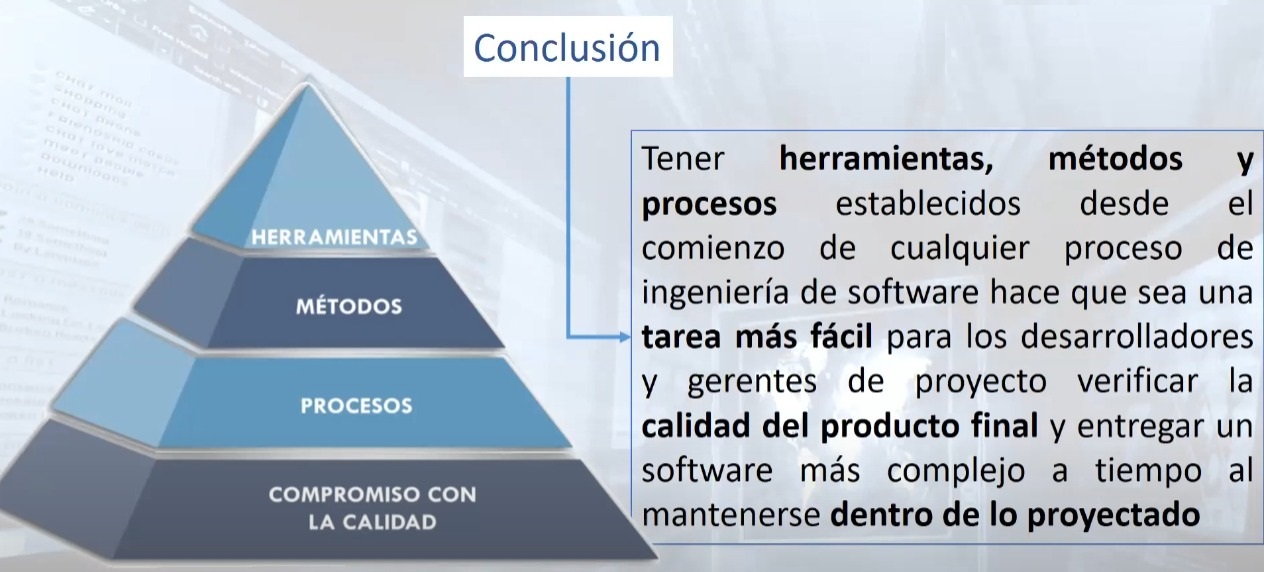
Peter Checkland define un sistema de información como una estructura compuesta por componentes interrelacionados que trabajan juntos para recopilar, procesar, almacenar y distribuir datos, convirtiéndolos en información útil para apoyar las operaciones y decisiones de una organización. Los sistemas de información integran tecnología y aspectos humanos para optimizar procesos, aumentar la productividad y mejorar el rendimiento organizacional. En el contexto educativo, un sistema de información sobre gestión de residuos utiliza inteligencia artificial y hardware interactivo, como Arduino, para enseñar a los estudiantes a clasificar y gestionar residuos, fomentando conciencia ambiental y habilidades tecnológicas. ( Ochoa & Higgings, 2007)

## 2.2 Ingenieria de software

La Ingeniería de Software se define como un proceso que implica analizar los requisitos del usuario y luego diseñar, construir y probar la aplicación de software para satisfacer esos requisitos. En términos más precisos, se refiere a la aplicación de un enfoque sistemático, disciplinado y computable para desarrollar, operar y mantener software.

La Ingeniería de Software, al aplicar un enfoque sistemático y disciplinado, proporciona numerosas ventajas, incluida la confiabilidad y eficiencia del software en las máquinas reales, así como la economía en términos de recursos y costos asociados con su desarrollo. Además, promueve la creación de una documentación adecuada y la aplicación del conocimiento científico en el diseño y construcción de programas de computadora, lo que contribuye a su calidad y fiabilidad a lo largo de su ciclo de vida.

Figura 2 Capas de Ingeniería



Fuente Saul Alexander

La calidad del software abarca varios aspectos fundamentales para su buen funcionamiento y satisfacción del usuario. Esto incluye su funcionalidad, asegurando que cumpla con los requisitos y expectativas; su fiabilidad, para operar de manera consistente y precisa; su usabilidad, facilitando una experiencia amigable para el usuario; su eficiencia, optimizando el uso de recursos del sistema; su mantenibilidad, permitiendo futuras modificaciones y actualizaciones sin complicaciones; su portabilidad, garantizando su compatibilidad en diferentes plataformas; y su seguridad, protegiendo los datos y la privacidad del usuario ante posibles amenazas cibernéticas.

La arquitectura de sistemas juega un papel vital al establecer la estructura general de un proyecto. En el ámbito del software, hay dos tipos principales: el estándar, que es más general y adaptable, y el personalizado, hecho a medida para satisfacer las necesidades particulares de un cliente específico. (Fritz L. & Lines., 2012)

## 2.2.1 Metodologías

• Scrum es una metodología ágil de gestión y desarrollo de proyectos que organiza el trabajo en ciclos cortos y repetitivos llamados sprints, típicamente de dos a cuatro semanas de duración. Se enfoca en la colaboración constante entre los miembros del equipo, la entrega continua de incrementos funcionales del producto, y la capacidad de adaptarse rápidamente a los cambios en los requisitos y prioridades. Los roles clave en Scrum incluyen el Product Owner, el Scrum Master y el equipo de desarrollo, mientras que los eventos fundamentales comprenden la planificación del sprint, las reuniones diarias, las revisiones del sprint y las retrospectivas para la mejora continua (Schwaber & Sutherland, 2017)

## 2.2.2 RUP

Según el autor El Proceso Unificado de Rational (RUP) es un método de desarrollo de software que enfatiza la colaboración entre equipos, la gestión de riesgos y la adaptabilidad a los cambios en los requisitos del cliente. Proporciona un marco detallado que guía a los equipos desde la captura de requisitos hasta la entrega del producto final, centrándose en la calidad, la puntualidad y el cumplimiento del presupuesto. Al fomentar la comunicación y la colaboración, RUP ayuda a asegurar el éxito en proyectos de desarrollo de software en entornos complejos. (Group, 2023)

RUP se basa en cuatro principios clave: la creación de modelos visuales, la verificación continua de la calidad, la gestión de cambios y la gestión de configuración. Estos principios se aplican a través de diferentes fases que van desde la concepción hasta la implementación y despliegue del software. El objetivo de RUP es garantizar la entrega de software de alta calidad que cumpla con los requisitos del cliente, al tiempo que se manejan de manera efectiva el tiempo y el presupuesto del proyecto. Es una herramienta valiosa para las organizaciones que buscan un enfoque estructurado y controlado para el desarrollo de software. (Alpaydin, 2020)

## 2.2.3 UML

(Lenguaje de Modelado Unificado) es un conjunto estándar de herramientas visuales utilizado en ingeniería de software para representar, visualizar y definir componentes, procesos y relaciones de sistemas informáticos, facilitando la especificación precisa y la comprensión de los sistemas de software.

Según el autor UML ofrece una amplia gama de diagramas para modelar diferentes aspectos de un sistema informático, incluyendo el Diagrama de Flujo de Datos para representar la transformación de datos, el Diagrama de Componentes para describir la estructura física del sistema, el Diagrama de Objetos para representar las instancias de clases en tiempo de ejecución, el Diagrama de Paquetes para organizar elementos en grupos lógicos, el Diagrama de Tiempo para modelar el comportamiento temporal, y el Diagrama de Despliegue para describir la configuración del hardware y el software. Cada uno de estos diagramas proporciona una perspectiva única del sistema, lo que permite una comprensión completa y una comunicación efectiva durante el proceso de desarrollo (Goodfellow, Yoshua, & Courville, 2016)

## 2.2.4 Diagrama Estructural

Es una representación visual que muestra la estructura estática de un sistema o parte de un sistema. Se centra en los elementos principales del sistema y sus interacciones en términos de relaciones y dependencias. Estos diagramas ayudan a los diseñadores y desarrolladores a entender la organización y la arquitectura del sistema, así como a comunicar estas estructuras de manera efectiva a otros stakeholders del proyecto. (Flanagan D. , 2020)

## 2.3 Backend y Frontend

* : Backend: Es la parte de una aplicación o sistema que se encarga de la lógica de negocio, procesamiento de datos, y gestión de la base de datos. Está situado en el servidor y no es visible para los usuarios finales. El backend se comunica con el frontend a través de APIs para proporcionar la funcionalidad necesaria.

Figura 3 Arquitectura



Fuente 1 (Juárez, 2008)

* Frontend: Es la parte de una aplicación o sistema que interactúa directamente con los usuarios finales. Está compuesto por la interfaz gráfica de usuario (GUI), que incluye elementos como botones, formularios, y otros elementos visuales. El frontend se ejecuta en el navegador del usuario y se comunica con el backend para obtener y enviar datos. (Freeman, 2017)

## 2.4 Frontend

Frontend El desarrollo de aplicaciones Python se refiere a la parte de la aplicación que interactúa directamente con los usuarios. En el caso de aplicaciones de escritorio, esta interacción se realiza a través de interfaces gráficas (GUIs, por sus siglas en inglés). A diferencia del desarrollo frontend en aplicaciones web, donde se utilizan tecnologías como HTML, CSS y JavaScript, en aplicaciones desarrolladas en Python para interfaces de usuario se emplean bibliotecas como Tkinter, PyQt o Kivy.

Tkinter es una biblioteca estándar de Python utilizada para el desarrollo de interfaces gráficas de usuario. Es una herramienta que permite a los desarrolladores crear aplicaciones con ventanas, botones, menús, formularios y otros elementos de interacción visual. A través de Tkinter, el frontend de una aplicación puede capturar información del usuario, mostrar resultados, y comunicarse con la lógica y el backend de la aplicación. (Haverbeke, 2018)

## 2.5 Backend

Backend se refiere a la parte de una aplicación que maneja la lógica del servidor, bases de datos e integración de sistemas, asegurando que las aplicaciones sean robustas, eficientes y seguras. Utiliza lenguajes como Python, Java, Ruby, PHP y frameworks como Node.js, Django, Ruby on Rails y Spring para procesar y enviar datos entre el frontend y los sistemas de almacenamiento, gestionando la lógica de negocio, autenticación de usuarios y comunicación mediante APIs. (Schwaber & Sutherland, 2017)

## 2.5.1 Python

Según el autor ( Ramalho, 2021)es un lenguaje de programación de alto nivel conocido por su sintaxis clara y legible, que favorece la legibilidad del código y la productividad del programador. Es ampliamente utilizado en diversas áreas como desarrollo web, ciencia de datos, inteligencia artificial, automatización de tareas y más. Python es interpretado y multipropósito, lo que significa que puede ser utilizado para desarrollar desde pequeños scripts hasta aplicaciones complejas y sistemas . robustos, Python soporta la programación orientada a objetos a través de clases y objetos, además de ofrecer módulos para organizar y reutilizar código. También proporciona un manejo de errores robusto mediante excepciones (try y except) y permite la creación concisa de listas y diccionarios con comprehensivos.

## 2.5.2 Red Neuronal

El concepto de red neuronal artificial fue introducido por Warren McCulloch y Walter Pitts en 1943, quienes propusieron el primer modelo matemático de una neurona artificial. Sin embargo, el desarrollo moderno de redes neuronales como las conocemos hoy ha sido impulsado por diversos autores y científicos, como Frank Rosenblatt, quien desarrolló el perceptrón en 1958, y Geoffrey Hinton, considerado uno de los padres del aprendizaje profundo (deep learning), quien popularizó el uso de redes neuronales profundas para aplicaciones de inteligencia artificial en la década de 2000. (Guide & Hinton, 2023)

* **GroundingDINO** es una red neuronal profunda diseñada para abordar el problema de la segmentación y el reconocimiento de objetos en imágenes de manera más precisa y contextual. Combina técnicas de detección de objetos y segmentación semántica para "anclar" (grounding) los objetos detectados en las imágenes con
* Non-locality Optimization). Esto significa que puede procesar tanto la información visual como textual, permitiendo reconocer y segmentar objetos específicos a partir de instrucciones o descripciones textuales.
* La red es útil en aplicaciones donde es necesario identificar y etiquetar con precisión objetos dentro de una imagen, incluso en situaciones donde la definición de los objetos puede ser subjetiva o depender del contexto proporcionado por el lenguaje. Este enfoque puede integrarse en sistemas avanzados de visión por computadora que requieren una comprensión más profunda de las relaciones entre objetos y descripciones textuales, como en sistemas de IA que necesitan entender y describir escenas complejas.ensivos. (Hiton, Hiton, & Bengio, 2015)

## 2.5.5 Roboflow

Según el autor de Roboflow es una plataforma integral de gestión de datos que está diseñada específicamente para proyectos de visión por computadora. La plataforma facilita cada paso del proceso de desarrollo de modelos, desde la creación y etiquetado de datasets hasta la mejora y optimización de esos datos para obtener mejores resultados en la detección de objetos. Una de sus características clave es la capacidad de aumentar automáticamente los datos, lo que significa que puede generar nuevas imágenes a partir de las existentes mediante técnicas como rotaciones, recortes, cambios de iluminación y más, ayudando a evitar problemas de sobreajuste.

Además, Roboflow permite entrenar modelos de visión por computadora en la nube y descargar los modelos entrenados en diferentes formatos listos para ser integrados en aplicaciones de desarrollo. Los usuarios pueden acceder a modelos preentrenados, lo que facilita la creación de soluciones de visión por computadora personalizadas sin necesidad de empezar desde cero. También es compatible con herramientas como YOLO, TensorFlow, PyTorch, entre otras, lo que la convierte en una plataforma flexible para múltiples necesidades. (Roboflow, s.f.)

## 2.5.4 Google Colab

Es una plataforma proporcionada por Google que permite a los usuarios escribir y ejecutar código Python en un entorno de cuaderno en la nube. Facilita el uso de recursos de hardware como GPU y TPU sin necesidad de configuración adicional, lo que lo convierte en una herramienta popular para el desarrollo y la experimentación con modelos de aprendizaje automático y análisis de datos. (Retrieved, 2024)

## 2.5.4 Pycharm

Es un entorno de desarrollo integrado (IDE) para Python desarrollado por JetBrains, que ofrece una serie de herramientas avanzadas para facilitar la programación en este lenguaje. Incluye características como autocompletado de código, depuración, integración con sistemas de control de versiones, y gestión de entornos virtuales, además de soporte para el desarrollo web con frameworks como Django y Flask. PyCharm está diseñado para mejorar la productividad de los desarrolladores de Python y optimizar el flujo de trabajo en proyectos de programación complejos aprendizaje

(August, 2024)

## 2.5.6 Ultralytics

Ultralytics es una compañía de software que se ha especializado en el desarrollo de soluciones avanzadas de visión por computadora, siendo su mayor contribución la serie de modelos YOLO (You Only Look Once), famosos por su capacidad de realizar detección de objetos en tiempo real con alta precisión. Fundada en 2014 por Glenn Jocher, la compañía ha revolucionado el campo de la inteligencia artificial al ofrecer versiones mejoradas y optimizadas de YOLO, como YOLOv5 y YOLOv8, que se caracterizan por su eficiencia, velocidad y facilidad de uso. Ultralytics facilita el desarrollo de aplicaciones de visión por computadora a través de herramientas accesibles y código abierto, apoyando tanto a investigadores como a desarrolladores en la implementación de estas soluciones en proyectos del mundo real. (Jocher, 2020)

## 2.5.7 CUDA

CUDA es una plataforma de computación paralela y modelo de programación desarrollado por NVIDIA que permite a los desarrolladores utilizar la potencia de las GPU (unidades de procesamiento gráfico) para realizar cálculos intensivos de manera más eficiente que con las CPU tradicionales. Introducido en 2007, CUDA está diseñado para aprovechar la arquitectura de las GPU y acelerar aplicaciones complejas en una variedad de campos, incluyendo inteligencia artificial, simulaciones científicas, procesamiento de imágenes y más. CUDA proporciona un entorno de desarrollo compatible con lenguajes como C, C++ y Python, facilitando la creación de algoritmos que ejecuten grandes cantidades de tareas en paralelo, mejorando drásticamente el rendimiento de ciertas aplicaciones, especialmente en deep learning (Corporation., 2007)

## 2.5.7 Deep Learning

Según el autor Deep Learning es una subdisciplina del machine learning que utiliza redes neuronales artificiales, particularmente redes neuronales profundas con múltiples capas, para modelar patrones y aprender representaciones a partir de grandes cantidades de datos. Inspirado en la estructura y funcionamiento del cerebro humano, el deep learning permite a las máquinas aprender de manera autónoma a partir de ejemplos, sin necesidad de reglas explícitas programadas. Esto lo hace particularmente efectivo en tareas complejas como la clasificación de imágenes, reconocimiento de voz, procesamiento de lenguaje natural, y conducción autónoma. Los avances en deep learning han sido posibles gracias a la disponibilidad de grandes volúmenes de datos (big data) y al incremento de la capacidad de procesamiento mediante GPUs y TPUs, herramientas fundamentales para entrenar estas redes. (Y, Y, & G, 2015)

## 2.5.8 Open CV

Según el autor OpenCV, que significa Open Source Computer Vision Library, es una biblioteca de código abierto que se utiliza principalmente para el procesamiento de imágenes y visión por computadora. Fue desarrollada inicialmente por Intel y ahora es mantenida por Willow Garage y Itseez. OpenCV proporciona herramientas para la captura, análisis y manipulación de imágenes y videos, y es ampliamente utilizada en aplicaciones de detección de objetos, reconocimiento facial, realidad aumentada, entre otros. (G & Dobb's, 2000).

## 2.5.9 Inteligencia Artificial (IA)

La Inteligencia Artificial (IA) es un campo de estudio dentro de las ciencias de la computación que se enfoca en la creación de sistemas capaces de realizar tareas que requieren inteligencia humana, tales como el reconocimiento de patrones, la toma de decisiones, la resolución de problemas y el aprendizaje. El objetivo principal de la IA es simular el comportamiento humano mediante algoritmos y modelos matemáticos que permiten a las máquinas aprender de la experiencia y adaptarse a nuevas situaciones.

(S & P, 2016)

## 3 ELEMENTOS ELECTRONICOS PARA EL PROYECTO

Según el artículo La comprensión de los componentes electrónicos es fundamental para el diseño y mantenimiento de sistemas electrónicos. Estos componentes permiten a los ingenieros y técnicos diseñar y solucionar correctamente los problemas de los sistemas electrónicos, mejorar el rendimiento del circuito y diagnosticar y abordar fallas de manera eficiente. (Globalwellpcba, 2024)

* Arduino Board: Un microcontrolador como Arduino Uno, Arduino Mega o Arduino Nano para controlar y programar el prototipo electrónico que abre y cierra los contenedores de residuos.
* Sensores de Visión: Sensores como el sensor de visión de computadora (CMOS) o el sensor de visión de cámara (CCD) para capturar imágenes de los residuos y enviarlas al sistema de clasificación.
* Motor y Actuador: Un motor y un actuador como un servo motor o un motor DC para controlar la apertura y cierre de los contenedores de residuos.
* Componentes de Interfaz: Componentes como botones, LEDs y displays para proporcionar una interfaz de usuario fácil de usar y visualizar los resultados del clasificador de residuos.
* Conectores y Cables: Conectores y cables para conectar los componentes electrónicos y permitir la comunicación entre ellos.
* Alimentación y Protección: Una fuente de alimentación y protección contra sobretensiones y cortocircuitos para garantizar la seguridad y estabilidad del sistema.
* Software de Desarrollo: Software como el IDE de Arduino para programar y desarrollar el sistema electrónico.
* Red neuronal: Una red neuronal como TensorFlow o PyTorch para entrenar y ejecutar el clasificador de residuos con inteligencia artificial.

**CAPÍTULO III**

**PROPUESTA DE INNOVACIÓN O SOLUCIÓN DEL PROBLEMA**

# Propuesta de Innovación o Solución del Problema

## Análisis de requerimientos

De acuerdo con las entrevistas realizadas a interesados de la unidad educativa “Nórdicos Boliviano”, se logró recolectar informaciones relevantes respecto a las necesidades y requerimientos de la U.E. para una mejor comprensión ya análisis con el objeto de cubrir loa aspectos funcionales y no funcionales

## 3.1.1. Funcionales

Tabla 1 Requerimientos Funcionales

|  |  |
| --- | --- |
| REQUERIMIENTOS | DESCRIPCION |
| 1.-Clasificación automática de residuos | El sistema deberá clasificar los residuos en orgánicos, papel y plástico mediante la detección de imágenes utilizando una cámara con IA. |
| 2.-Interacción con el Prototipo de Hardware | El sistema debe enviar una señal al prototipo de basurero automatizado para que abra el contenedor correspondiente según el tipo de residuo clasificado. |
| 3.-Apertura automática de contenedores | Cada contenedor (verde para residuos orgánicos, azul para papel, y amarillo para plástico) deberá abrirse automáticamente cuando la cámara detecte el tipo de residuo correspondiente. |
| 4.-Captura de imágenes en tiempo real | El sistema deberá capturar imágenes en tiempo real a través de una cámara conectada y procesarlas para identificar el tipo de residuo. |
| 5.-Entrenamiento y actualización del modelo de IA | El sistema permitirá entrenar y actualizar el modelo de IA para mejorar la precisión de la clasificación de residuos.   |

Fuente Elaboración propia

## 3.1.2 No funcionales

Tabla 2 Requerimientos no funcionales

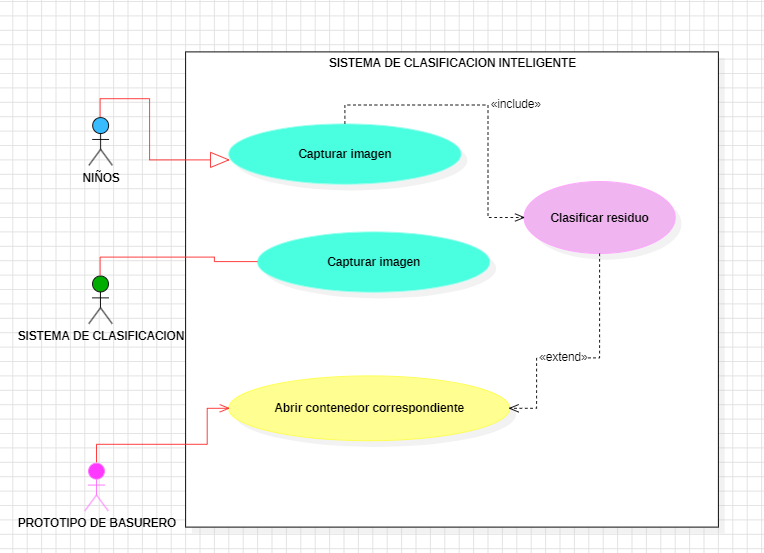
|  |  |
| --- | --- |
| Requerimientos | Descripción |
| * Precisión del Sistema | El sistema debe alcanzar una precisión mínima del 90% en la detección y clasificación de residuos para garantizar su efectividad en el contexto educativo. |
| * Rendimiento | El sistema debe procesar la imagen y emitir una clasificación en tiempo real, con un tiempo de respuesta no mayor a 2 segundos desde la captura de la imagen hasta la apertura del contenedor. |
| * Usabilidad | El sistema debe ser operable por personas sin conocimientos técnicos avanzados. |
| * Portabilidad y Escalabilidad | El sistema debe ser escalable para poder entrenar modelos adicionales que clasifiquen más tipos de residuos en el futuro.  El prototipo de hardware debe ser fácilmente transportable y adaptable a diferentes entornos educativos |
| * Seguridad | El prototipo de basurero debe ser seguro para su uso por niños, evitando cualquier tipo de riesgo físico. |
| * Mantenibilidad | Debe contar con documentación clara y accesible para facilitar el mantenimiento técnico. |
| * Compatibilidad | El sistema debe ser compatible con múltiples tipos de cámaras y plataformas de hardware basadas en Arduino. |

Fuente Elaboración Propia

## Diagramas de casos de uso

## 3.2.1 Clasificación automática de residuos

Figura 4 Diagrama de casos de usos clasificación automática



Fuente Elaboración propia

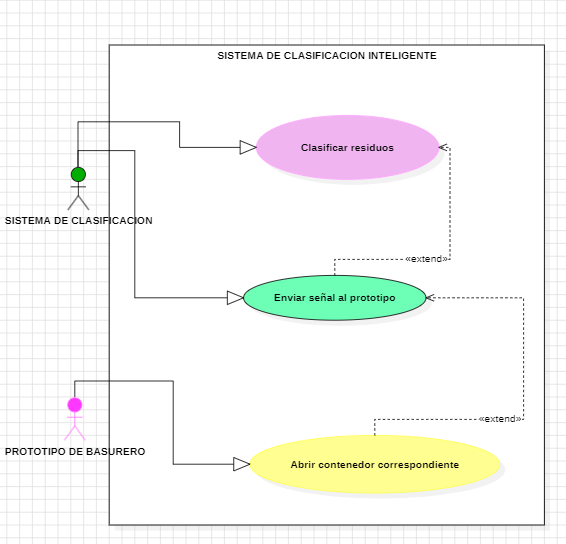
Tabla 3 casos de usos clasificación automática

|  |  |
| --- | --- |
| Campo | Descripción |
| Área | Sistema de Clasificación Inteligente de Residuos |
| Actor(es) | Niños, Sistema de Clasificación, Prototipo de Basurero |
| Interesados | Niños (Usuarios), Administradores del Sistema, Prototipo de Basurero |
| Descripción | Este caso de uso permite capturar la imagen de un residuo, clasificarlo automáticamente, y abrir el contenedor correspondiente donde debe ser depositado. El proceso involucra un prototipo de basurero inteligente que recibe las instrucciones para abrirse una vez que el sistema ha clasificado el residuo. |
| Evento desencadenador | El niño coloca un residuo frente al sistema de clasificación, el cual captura la imagen y activa el proceso de clasificación y apertura del contenedor correspondiente. |

Fuente Elaboración propia

## 3.2.2 Interacción con el Prototipo de Hardware

Figura 5 Diagrama de casos de usos Interacción con el prototipo

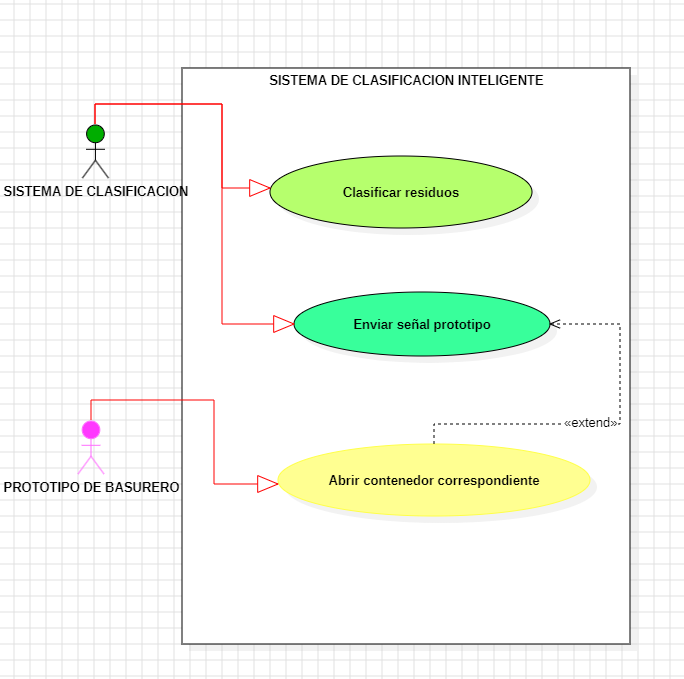


Fuente Elaboración propia

|  |  |
| --- | --- |
| Campo | Descripcion |
| Área | Sistema de Clasificación Inteligente de Residuos |
| Actor(es) | Sistema de Clasificación, Prototipo de Basurero |
| Interesados | Administradores del Sistema, Usuarios del Sistema, Prototipo de Basurero |
| Descripción | Este caso de uso involucra la clasificación de residuos por el sistema inteligente, seguido del envío de una señal al prototipo de basurero, que finalmente abre el contenedor correspondiente. El proceso automatiza la clasificación y manejo de residuos de manera eficiente. |
| Evento desencadenador | El sistema de clasificación identifica el tipo de residuo y envía una señal al prototipo de basurero para abrir el contenedor adecuado, permitiendo la disposición correcta del residuo. |

## 3.2.3 Apertura automática de contenedores e Hardware

Figura 6 Diagrama de casos de usos apertura automática de contenedores



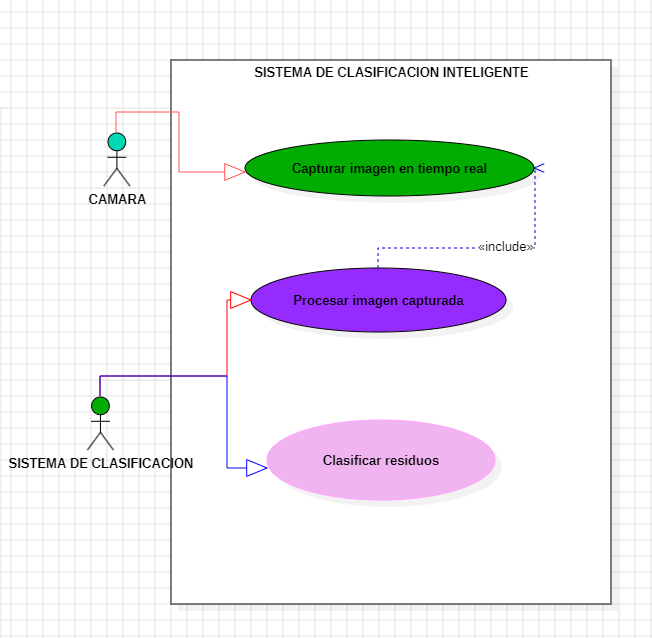
Fuente Elaboración propia

Tabla 4 casos de usos apertura automática de contenedores

|  |  |
| --- | --- |
| Campo | Descripción |
| Área | Sistema de Clasificación Inteligente de Residuos |
| Actor(es) | Sistema de Clasificación, Prototipo de Basurero |
| Interesados | Administradores del Sistema, Usuarios, Prototipo de Basurero |
| Descripción | Este caso de uso describe el proceso de clasificación automática de residuos, seguido del envío de una señal al prototipo de basurero, que abre el contenedor correspondiente para la disposición del residuo. El sistema identifica el tipo de residuo y automatiza su correcta disposición. |
| Evento desencadenador | El sistema de clasificación identifica el residuo, envía una señal al prototipo de basurero y este último abre el contenedor adecuado para el tipo de residuo clasificado. |

## 3.2.4 Captura de imágenes en tiempo real

Figura 7 Captura de imágenes en tiempo real



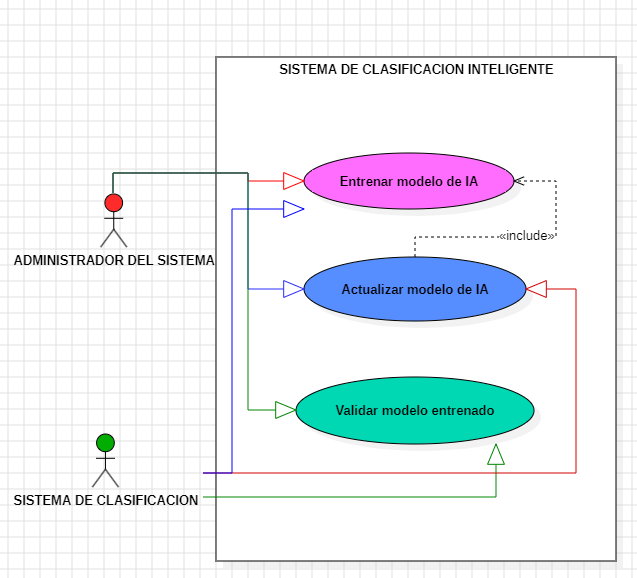
Fuente Elaboración propia

Tabla 5 Captura de imágenes en tiempo real

|  |  |
| --- | --- |
| Campo | Descripción |
| Área | Sistema de Clasificación Inteligente de Residuos |
| Actor(es) | Cámara, Sistema de Clasificación |
| Interesados | Administradores del Sistema, Usuarios |
| Descripción | Este caso de uso permite al sistema capturar una imagen en tiempo real utilizando la cámara. La imagen es procesada para identificar el tipo de residuo, que luego es clasificado por el sistema de clasificación. |
| Evento desencadenador | La cámara captura una imagen del residuo en tiempo real, la cual es procesada para su clasificación automática dentro del sistema. |

## 3.2.5 Entrenamiento y actualización del modelo de IA

Figura 8 Diagrama de casos de usos Entrenamiento del modelo de IA



Fuente Elaboración propia

Tabla 6 de casos de usos Entrenamiento del modelo de IA

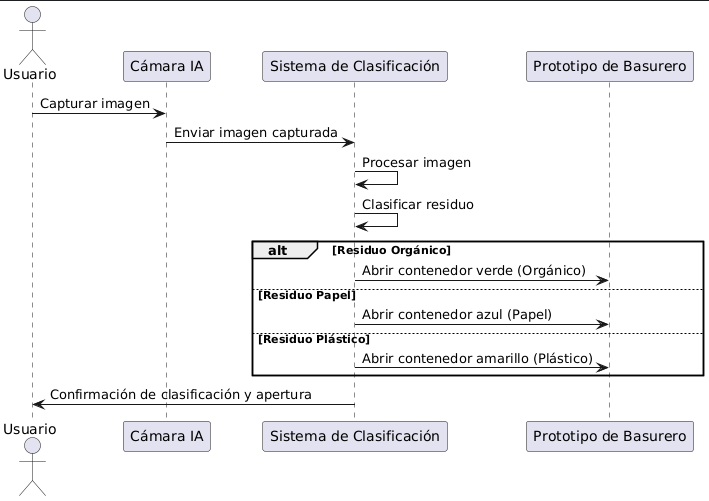
|  |  |
| --- | --- |
| Campo | Descripción |
| Área | Sistema de Clasificación Inteligente de Residuos |
| Actor(es) | Administrador del Sistema, Sistema de Clasificación |
| Interesados | Administradores del Sistema, Desarrolladores de IA |
| Descripción | Este caso de uso permite al administrador del sistema entrenar un modelo de inteligencia artificial (IA), actualizar el modelo existente y validar los resultados del modelo entrenado para asegurar su eficacia en la clasificación de residuos. |
| Evento desencadenador | El administrador inicia el proceso de entrenamiento del modelo de IA, lo actualiza según sea necesario y valida su rendimiento para garantizar una correcta clasificación de residuos. |

Diagrama elaborado en base a los requerimientos funcionales (elaboración propia)

## Diagrama de secuencias

## 3.3.1 Clasificación automática de residuos

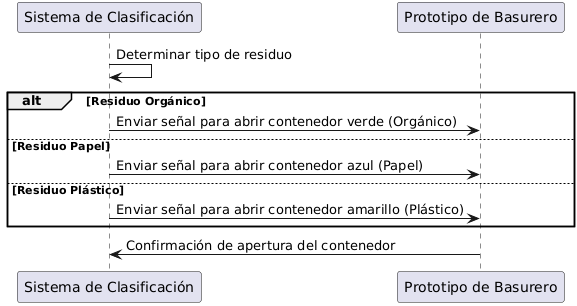
Figura 9 Diagrama de secuencias clasificación automática de residuos



Fuente Elaboración propia

## 3.3.2 Interacción con el Prototipo de Hardware

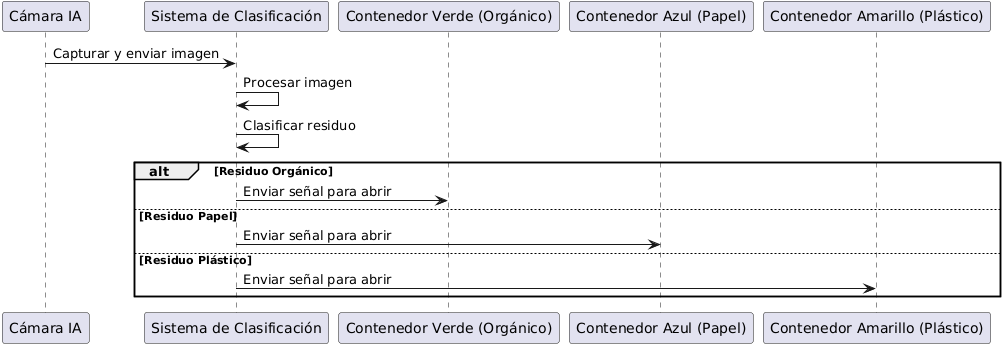
Figura 10 Diagrama de secuencias Interacción con el prototipo



Fuente Elaboración propia

## 3.3.3 Apertura automática de contenedores e Hardware

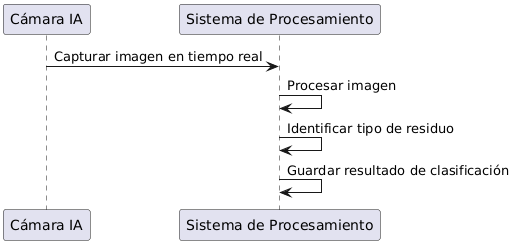
Figura 11 Diagrama de secuencias Apertura de contenedores



Fuente Elaboración propia

## 3.3.4 Captura de imágenes en tiempo real

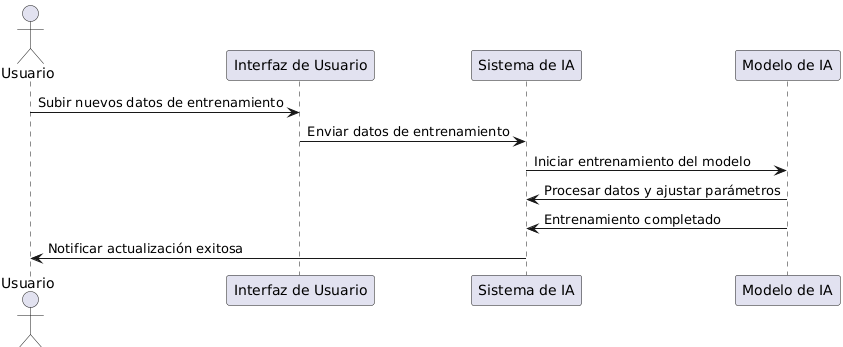
Figura 12 Diagrama de secuencias Captura de imágenes en tiempo real



Fuente Elaboración propia

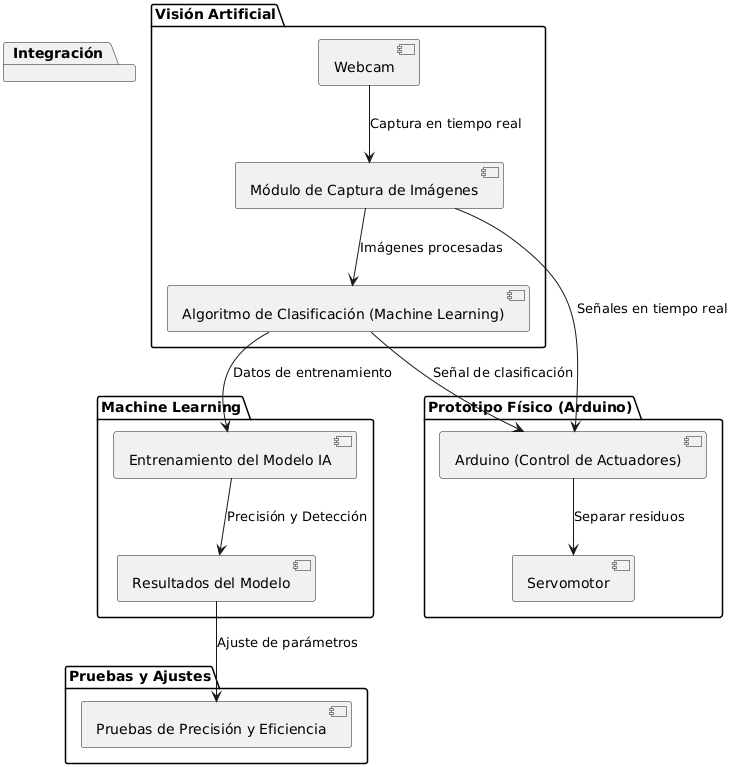
## 3.3.5 Entrenamiento y actualización del modelo de IA

Figura 13 Diagrama de secuencias Entrenamiento del modelo IA



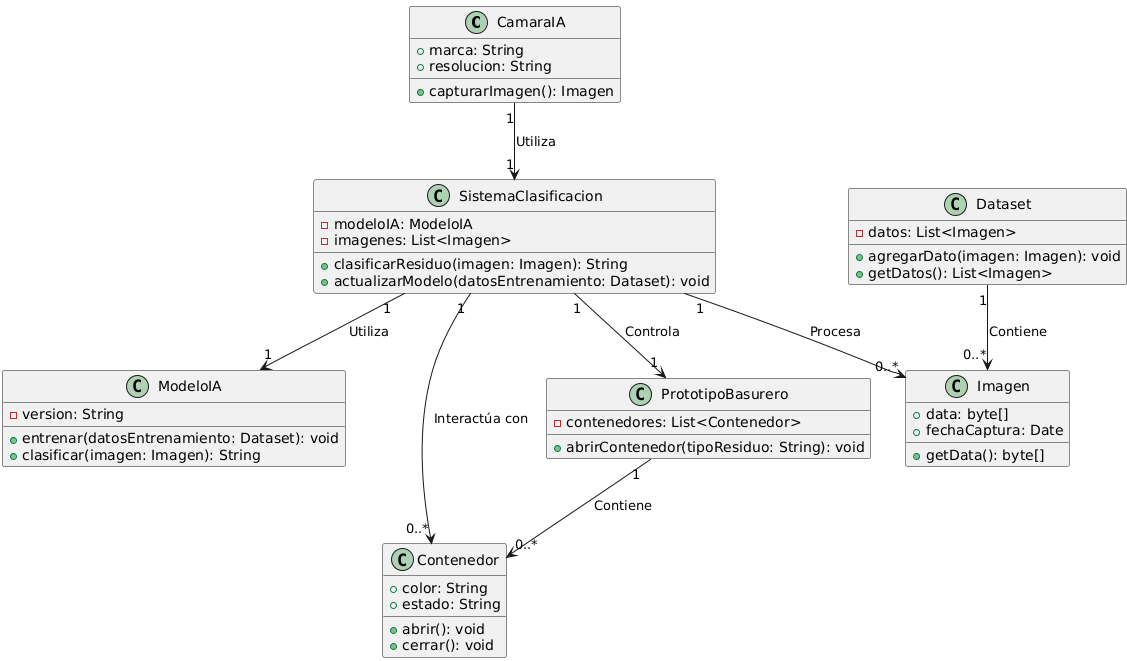
Fuente Elaboración propia

## Diagrama de componentes



## Diagrama de clases

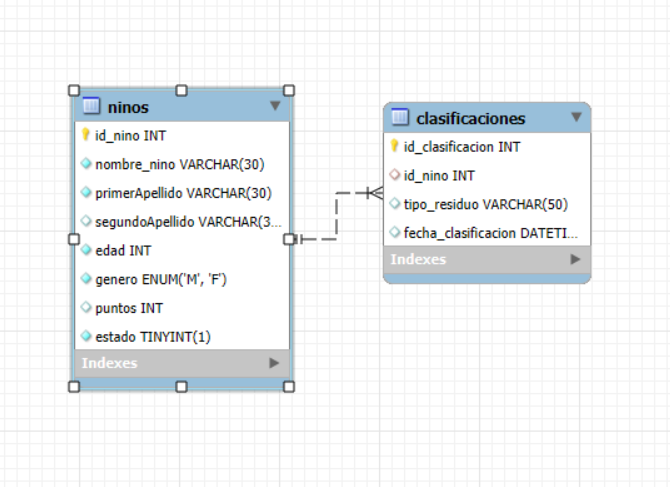
Figura 14 Diagrama de clases



Fuente Elaboración propia

## Diagrama Relacional

Figura 15 Diagrama relacional

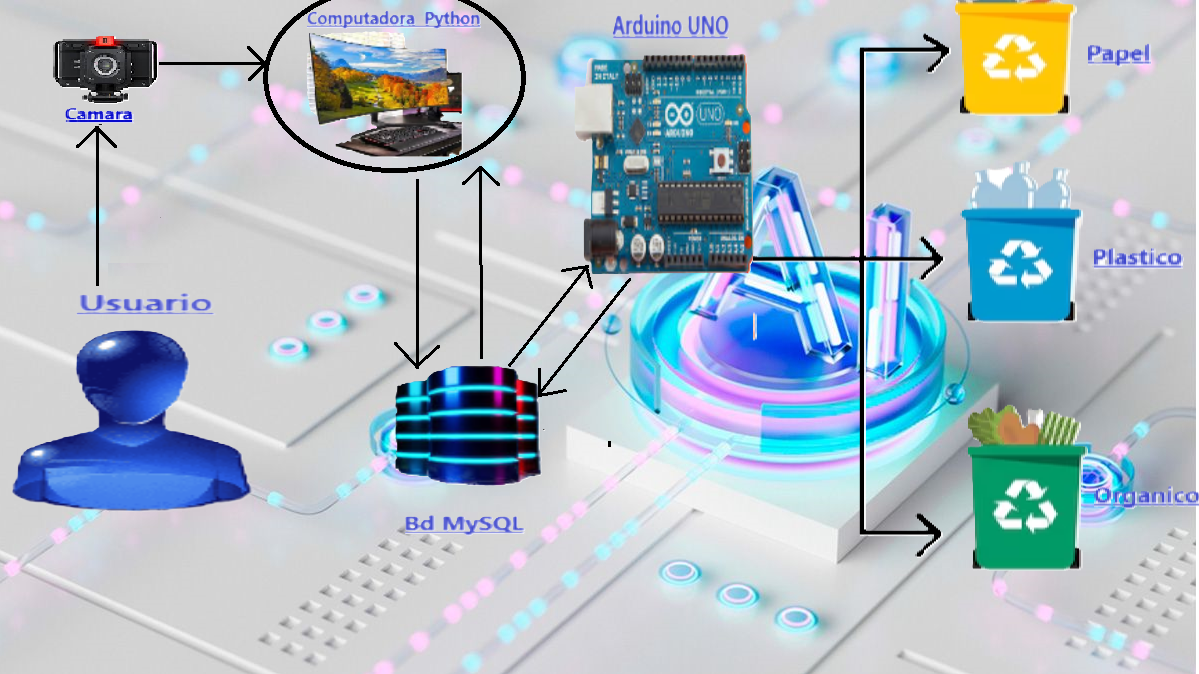


Fuente Elaboración propia

## Arquitectura del Sistema

## Base Arquitectónica

Figura 16 Base Arquitectónica



Fuente Elaboración Propia

La figura 16 muestra a grandes rasgos mediante un esquema plasmado el funcionamiento del proyecto y como estos interactúan entre si().

## Recursos de Hardware

Para el correcto funcionamiento del Sistema, el establecimiento educativo deberá contar con una computadora que será la encargada de ejecutar el sistema de visión artificial y de clasificación. Dicha computadora procesará las imágenes capturadas en tiempo real y enviará las señales correspondientes al sistema de automatización de los contenedores. Asimismo, se requerirá de un microcontrolador Arduino, el cual recibirá las señales del sistema y controlará los actuadores (servomotores) responsables de la apertura y cierre de los compartimientos de residuos.

El sistema necesitará un convertidor de voltaje que proporcionará una salida de 5V para alimentar el microcontrolador y los actuadores, el cual deberá estar conectado a una toma de corriente.

## Software de codificación

* **Interfaz Gráfica**

La interfaz gráfica del sistema fue desarrollada utilizando Tkinter en el entorno de programación Python. Se optó por este enfoque debido a su simplicidad y compatibilidad con múltiples plataformas, lo que facilitó el diseño de una interfaz intuitiva y funcional para la interacción con el sistema de clasificación de residuos. La interfaz permitió a los usuarios visualizar los resultados de la clasificación en tiempo real y acceder a funciones como la generación de reportes y el control de los contenedores automatizados.

* **Componentes electrónicos**

La programación de los componentes electrónicos se llevó a cabo utilizando **Arduino IDE** y el lenguaje de programación **Python**. Estos componentes incluyeron el control de los servomotores encargados de la apertura y cierre de los compartimientos de residuos. Se eligió Arduino debido a su flexibilidad y facilidad de integración con el sistema de clasificación, permitiendo una comunicación fluida entre el software de clasificación y el hardware de automatización.

## Pruebas de calidad

## 3.7.1 Introducción

A continuación, se presentan una serie de pruebas realizadas, demostrando así el correcto funcionamiento del Sistema Inteligente de Clasificación de Residuos y la interacción fluida entre el módulo de visión artificial y los componentes automatizados controlados por Arduino. Estas pruebas permitieron verificar el flujo del proyecto, así como identificar posibles errores o ajustes necesarios que podrían presentarse durante el uso del sistema en tiempo real.

## 3.7.2 Pruebas de Software

Las pruebas de software realizadas en el Sistema Inteligente de Clasificación de Residuos garantizaron el correcto funcionamiento e integración de todos los módulos. Estas pruebas verificaron que la interacción entre el sistema de visión artificial, el modelo de clasificación de residuos y los componentes automatizados controlados por Arduino se desarrollara de manera eficiente.

## 3.7.3 Pruebas de hardware

Las pruebas de hardware realizadas en el Sistema Inteligente de Clasificación de Residuos permitieron comprobar el correcto funcionamiento de los componentes electrónicos y su interacción con el sistema de clasificación. Se realizaron pruebas exhaustivas sobre el Arduino, los servomotores y el convertidor de voltaje, verificando que respondieran adecuadamente a las señales emitidas por el software de clasificación de residuos. Estas pruebas también aseguraron que el hardware pudiera operar de manera continua y precisa en condiciones reales, garantizando la apertura y cierre de los compartimientos de residuos según la clasificación efectuada por el sistema.

## 3.7.4 Pruebas de integración

Con el propósito de asegurar la funcionalidad integral del Sistema Inteligente de Clasificación de Residuos, se llevaron a cabo las pruebas de integración, cuyo objetivo principal fue verificar la correcta comunicación y coordinación entre los diferentes módulos del sistema, tanto de software como de hardware. Durante estas pruebas, se comprobó que el sistema de visión artificial pudiera detectar y clasificar correctamente los residuos, enviando las señales correspondientes al Arduino para el control de los servomotores, encargados de abrir y cerrar los compartimientos de residuos. Asimismo, se validó que la interfaz gráfica funcionara de manera fluida, permitiendo una interacción eficiente con el usuario y garantizando que todos los componentes del sistema trabajaran en conjunto sin interrupciones ni fallos

## 3.7. 5 Pruebas de Sistema

* **Prueba de Seguridad**

Las pruebas de seguridad realizadas en el **Sistema Inteligente de Clasificación de Residuos** tuvieron como propósito garantizar que tanto el sistema como sus componentes no presentaran riesgos para los usuarios durante su operación. Se verificó que los mecanismos automatizados controlados por el **Arduino**, como los servomotores, funcionaran dentro de los parámetros seguros, evitando accidentes durante la apertura y cierre de los compartimientos de residuos.

* **Prueba de Resistencia**

Las pruebas de resistencia se llevaron a cabo con el fin de asegurar la durabilidad de los componentes del sistema bajo condiciones de uso continuo. Se probó el rendimiento de los servomotores, los sensores y el microcontrolador **Arduino** en situaciones de uso prolongado, simulando las condiciones a las que se enfrentará el sistema en un entorno educativo. También se verificó la capacidad del sistema para operar correctamente bajo diferentes condiciones ambientales, como variaciones de temperatura y exposición al polvo.

* **Prueba de validación**

La prueba de validación tuvo como objetivo confirmar que el sistema cumpliera con todos los requisitos funcionales establecidos en el diseño. Durante esta fase, se comprobó que el sistema de visión artificial pudiera identificar y clasificar correctamente los residuos en sus categorías correspondientes. Por ejemplo, al clasificar un plátano, el sistema lo detectó y validó como orgánico, activando el mecanismo adecuado para la apertura del compartimiento correspondiente a residuos orgánicos. Este proceso se repitió con diferentes objetos (como plástico o papel), y en todos los casos, el sistema demostró ser capaz de validar correctamente cada tipo de residuo según su clasificación.

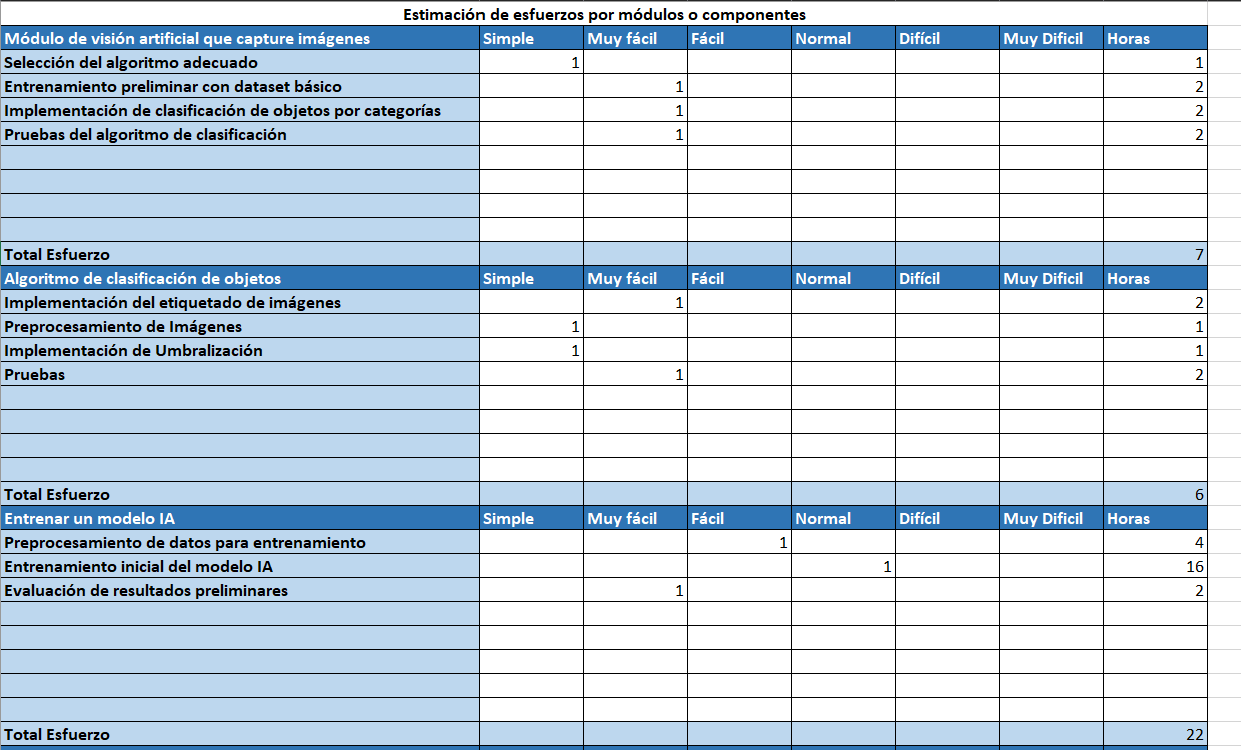
Además, se validó que la interfaz gráfica fuera intuitiva y permitiera al usuario visualizar los resultados de la clasificación en tiempo real, garantizando la correcta interacción entre el software y el hardware. De este modo, el sistema cumplió con las expectativas de los usuarios finales, logrando una clasificación precisa y eficiente.

* **Pruebas con los componentes electrónicos**

Durante las pruebas con los componentes electrónicos del sistema, se evaluó el rendimiento y la compatibilidad de los dispositivos involucrados, como el **Arduino**, los servomotores y los sensores de control. Se comprobó que cada uno de los componentes respondiera correctamente a las señales enviadas por el sistema de clasificación, ejecutando las acciones correspondientes sin errores. Además, se verificó que la conexión entre el microcontrolador y el resto de los elementos electrónicos fuera estable, asegurando una comunicación eficaz y sin interrupciones.

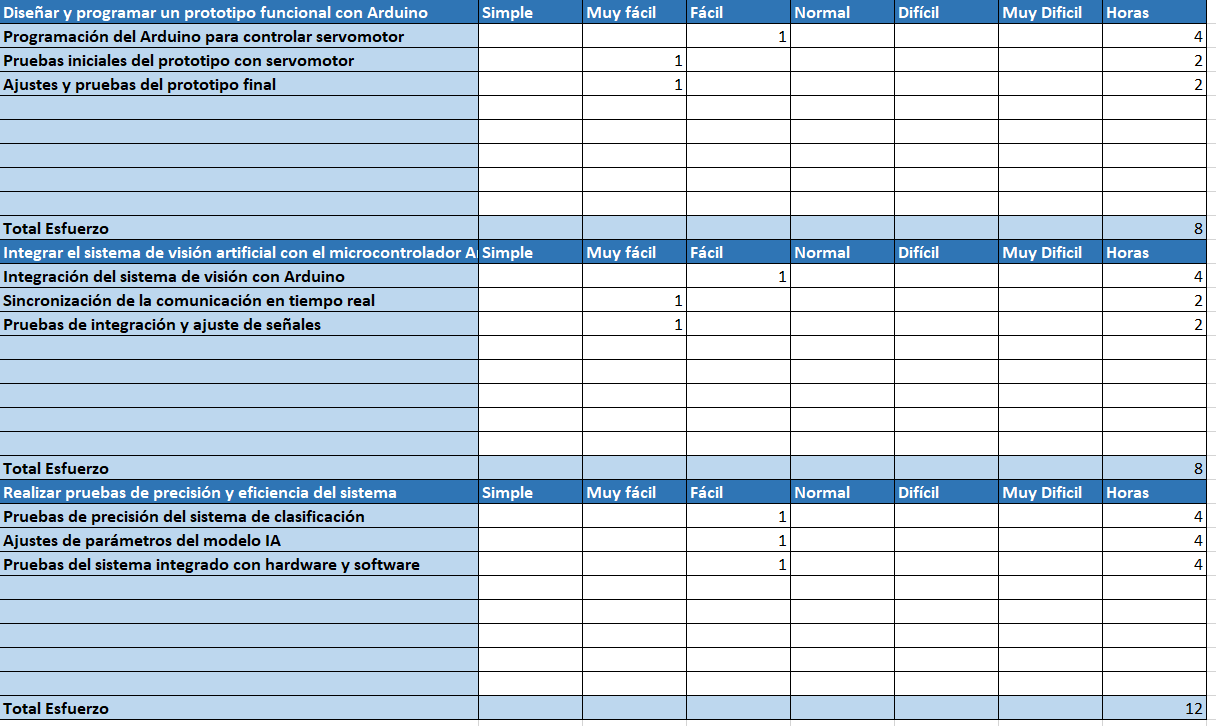
## Estimación de costos

Figura 17 Estimación de costos 1



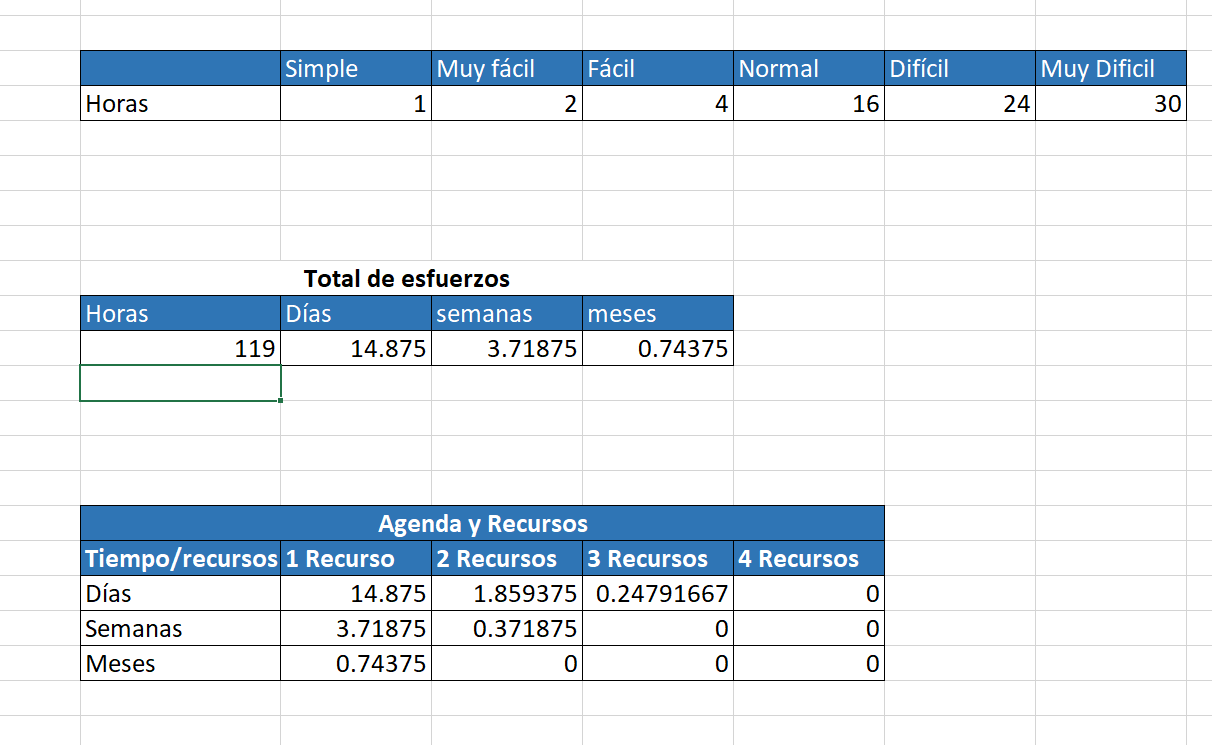
Fuente Elaboración propia

Figura 18 Estimación de costos 2



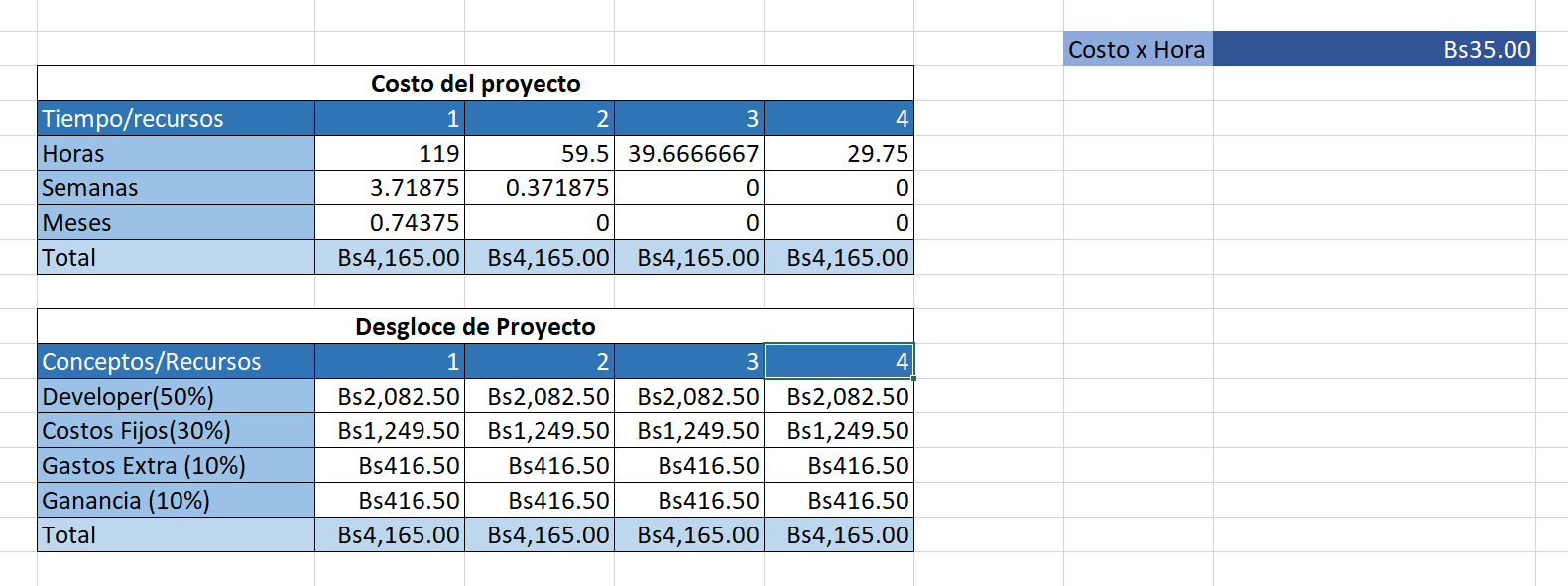
Fuente Elaboración propia

Figura 19 Estimación de costos 2



Fuente Elaboración propia

Figura 20 Estimación de costos 3



Fuente Elaboración propia

3

**CAPÍTULO IV  
CONCLUSIONES Y RECOMENDACIONES**

# CONCLUSIONES Y RECOMENDACIONES

## Conclusiones

En base al análisis y la implementación realizada, se cumplieron los objetivos específicos propuestos a través de los módulos de desarrollo, estableciéndose las siguientes conclusiones:

* Desarrollar un módulo de visión artificial que capture imágenes de objetos en tiempo real utilizando una webcam: Se logró desarrollar el módulo de visión artificial con éxito, implementando una cámara capaz de capturar imágenes de residuos en tiempo real con la calidad suficiente para que el sistema pudiera identificar y clasificar los objetos. Este proceso permitió un entrenamiento adecuado del modelo de inteligencia artificial, optimizando la detección de residuos.
* Implementar un algoritmo de clasificación de objetos basado en técnicas de machine learning: El algoritmo de machine learning fue implementado exitosamente, permitiendo la identificación y clasificación precisa de residuos en las categorías de plástico, papel y orgánico. La integración del modelo con la visión artificial facilitó una clasificación automática y en tiempo real de los objetos, lo que permitió un funcionamiento eficaz del sistema.
* Entrenar un modelo IA para estudiar los resultados preliminares en términos de precisión y detección: Se completó el entrenamiento del modelo de inteligencia artificial con resultados satisfactorios, alcanzando niveles de precisión aceptables en la identificación de residuos. Durante las pruebas iniciales, el sistema demostró una mejora continua en su capacidad de clasificación, tras ajustar los parámetros de entrenamiento.
* Diseñar y programar un prototipo funcional que utilice Arduino para controlar un actuador (Servomotor) que separe el residuo: El prototipo funcional fue diseñado y programado exitosamente, integrando un microcontrolador Arduino con un servomotor que separa los residuos automáticamente según la clasificación realizada por el sistema de visión artificial. Este mecanismo demostró una operatividad eficiente, cumpliendo con los requerimientos del proyecto.
* Integrar el sistema de visión artificial con el microcontrolador Arduino para procesar la información en tiempo real: Se alcanzó una integración efectiva entre el sistema de visión artificial y el microcontrolador Arduino, logrando que las señales generadas por el sistema de clasificación fueran transmitidas correctamente al actuador en tiempo real, permitiendo la separación automatizada de los residuos.
* Realizar pruebas de precisión y eficiencia del sistema: Se realizaron pruebas exhaustivas para evaluar la precisión y eficiencia del sistema, obteniendo resultados favorables en cuanto a la exactitud de la identificación de residuos. Las pruebas permitieron ajustar parámetros tanto del modelo de IA como del hardware, optimizando el rendimiento general del sistema.

## Recomendaciones

* Mejora Continua:
* Entrenamiento del Modelo: Se recomienda continuar mejorando el modelo de IA con datos adicionales y ajustar los parámetros para aumentar la precisión. Es importante realizar pruebas frecuentes para identificar y corregir posibles errores.
* Actualización del Sistema: Se sugiere mantener el sistema actualizado con nuevas funcionalidades y correcciones basadas en la retroalimentación de los usuarios.
* Educación y Capacitación:
* Capacitación del Usuario: Se sugiere proporcionar materiales educativos y sesiones de capacitación para los niños y el personal educativo sobre el uso del sistema y la importancia de la clasificación de residuos.
* Materiales Didácticos: Se recomienda incluir guías visuales y tutoriales en la interfaz para ayudar a los niños a entender el proceso de clasificación.
* Soporte y Mantenimiento:
* Mantenimiento Regular: Se debe establecer un plan de mantenimiento regular para el hardware y el software del sistema para asegurar su funcionamiento continuo.
* Soporte Técnico: Se recomienda proporcionar soporte técnico en caso de problemas con el sistema para resolver cualquier incidencia de manera rápida.

BIBLIOGRAFÍA

Ochoa, L., & Higgings. (2007). Academia. Revista Latinoamericana de Administración. En L. O. Higgings, *Reference list of information systems* (págs. 111-118). Obtenido de Academia. Revista Latinoamericana de Administración.

Ramalho, L. (2021). *Python: Clear, Concise, and Effective Programming.* Sebastopol: O'Reilly Media.

Alpaydin, E. (2020). *Introduccion to Machine Learning.* Cambridge: MIT Press.

August, R. (2024). *PyCharm: The Python IDE for professional developers.* Obtenido de etBrains. (n.d.). PyCharm: https://www.jetbrains.com/pycharm/

Autor, N. d. (2018). *Título del libro.* Cochabamba - Bolivia: ABC.

Banzi, M., & Shiloh, M. (2020). *Getting Started with Arduino: The Open Source Electronics Prototyping Platform (4th ed.). .* Sebastopol, CA: Maker Media.

Corporation., N. (2007). *CUDA: Compute Unified Device Architecture.* Obtenido de https://developer.nvidia.com/cuda-zone

Developers, G. (2019). *Web Fundamentals: Performance*. Obtenido de https://web.dev/articles/web-performance-made-easy

Flanagan, D. (2020). *JavaScript: The Definitive Guide (7th ed.).* Sebastopol, CA: O'Reilly Media.

Flanagan, D. (2020). *JavaScript: The Definitive Guide.* San Francisco: O'Reilly Media.

Freeman, E. (2017). *Head First HTML and CSS: A Learner's Guide to Creating Standards-Based Web Pages".* California: O'Reilly Media.

Fritz L., B., & Lines., M. (2012). *Disciplined Agile Delivery: A Practitioner's Guide to Agile Software Delivery in the Enterprise.* Madrid: IBM Press.

G, B., & Dobb's, J. (2000). *The OpenCV Library.* https://opencv.org.

Globalwellpcba. (2024). Comprensión de los componentes electrónicos.

Goodfellow, I., Yoshua, B., & Courville, A. (2016). *Aprendizaje Profundo.* Estados Unidos: MIT Press.

Group, T. P. (2023). *PHP Manual*. Obtenido de https://www.php.net/manual/en/

Guide, & Hinton, G. (2023). *Deep Learning and Neural Networks.* EEUU: MIT Press.

Haverbeke, M. (2018). *Eloquent JavaScript: A Modern Introduction to Programming.* San Francisco: 3ª edición.

Hiton, Hiton, G., & Bengio, Y. (2015). *Deep learning. Nature, 521(7553).* nature14539.

Jacobson, I., Booch, G., Rumbaugh, J., & Jacobson, I. (2005). *El lenguaje unificado de modelado (UML) guía del usuario.* Madrid: Addison-Wesley.

Jocher, G. (2020). *Ultralytics: Empowering Real-Time Object Detection.* Obtenido de https://github.com/ultralytics

Juárez, E. (2008). *Aplicaciones de escritorio con Tkinter y Arduino [turbocodigo]*. Obtenido de https://www.patreon.com/turbocodigo

Kleppmann, M. (2017). *Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems.* Sebastopol: O'Reilly Media.

Learning Solutions, E. (2021). *Learning Bootstrap 5: A Beginner's Guide to Building Responsive and Modern Websites.* No especificado: No especificado.

Meyer , E., & Lea, D. (2015). *CSS: The Definitive Guide.* Sebastopol: O'Reilly Media.

Molina Garcia, H., D. Ullman , J., & Widom, J. (2020). *Database Systems: The Complete Book.* No especificada: Pearson Education.

Retrieved, A. (2024). *The platform for computer vision data management and model training.* Obtenido de Google. (n.d.). Google Colab.: https://colab.research.google.com

Roboflow. (s.f.). *Build better computer vision models.* Obtenido de https://roboflow.com/

Robson, E., & Freeman, E. (2017). *Head First HTML and CSS: A Learner's Guide to Creating.* Sebastopol, California: O'Reilly Media.

S, R., & P, N. (2016). *Artificial Intelligence: A Modern Approach.* (3ra ed.). Prentice Hall.

Schwaber, K., & Sutherland, J. (2017). *The Scrum Guide.* Boston: Scrum.org.

Schwaber, K., & Sutherland, J. (16 de 06 de 2017). *Scrum.org.* Obtenido de https://www.scrumguides.org/scrum-guide.html

Scott W, A., & Lines, M. (2020). *Disciplined Agile Delivery: A Practitioner's Guide to Agile Software Delivery in the Enterprise.* Madrid: IBM Press.

Sklar, D. (2023). *Packt Publishing.*

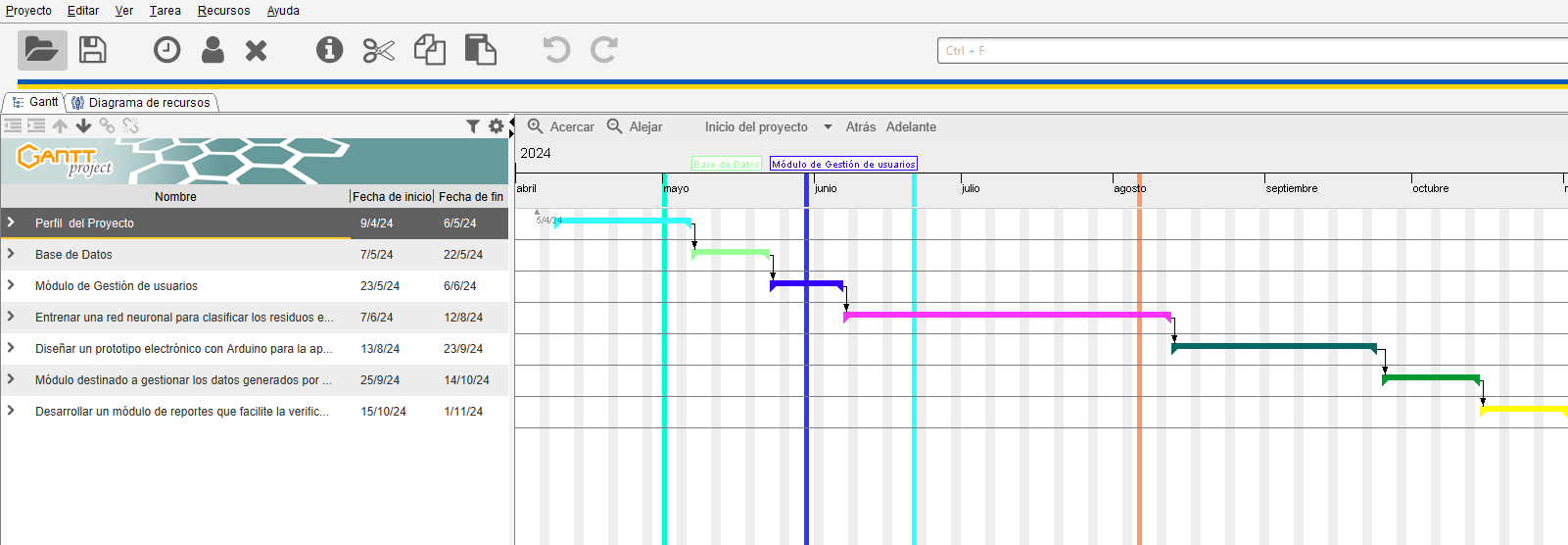
Smith, A., & Wesley, A. (2018). *Elements of Reusable Object-Oriented Software.* California: Desion Patterns.

Tramullas, J. (2010). Gestión de la información y el conocimiento en entornos digitales. *Revista Universidad y Sociedad*, págs. 15-20. Obtenido de Gestión de la información y el conocimiento en entornos digitales.

Y, L., Y, B., & G, H. (2015). *Deep learning. Nature.*

ANEXOS

**Anexo A. Cronograma**



Fuente Elaboración propia

**Anexo B. Guía de entrevista**

Estas preguntas se diseñaron para obtener información relevante y detallada sobre las necesidades, percepciones y expectativas del director con respecto a la gestión de residuos en la unidad educativa.

Preguntas:

1. ¿Qué entiende por reciclaje y usted recicla o separa sus residuos?
2. ¿Podría identificar los diferentes tipos de residuos que existen?
3. ¿Cuál es su percepción sobre la aplicación de inteligencia artificial en la clasificación de residuos?
4. ¿Considera importante la implementación de tecnologías innovadoras en la gestión de residuos dentro del entorno escolar?
5. ¿Cómo cree que esta tecnología puede impactar en la conciencia ambiental de los estudiantes?
6. ¿Cómo visualiza la integración de este sistema web clasificador de residuos en la dinámica diaria de la escuela y cómo cree que puede mejorar la eficiencia en la gestión de residuos?
7. ¿Qué beneficios considera usted que traería la integración de un sistema web clasificador de residuos en su establecimiento educativo?
8. ¿Considera que este enfoque puede fomentar el interés de los estudiantes en la tecnología y la innovación?
9. ¿Qué opinión tiene sobre la posibilidad de recibir capacitación sobre el uso y funcionamiento del sistema clasificador de residuos implementado en nuestra escuela?
10. ¿Tiene alguna sugerencia o comentario adicional sobre el proyecto de gestión de residuos en su U.E. que le gustaría compartir?