

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

Diseño de un sistema automatizado de conteo continuo de varillas
mediante visión artificial.

Previo la obtención del Título de:

Magister en Automatización y Control Industrial

Presentado por:

Vanessa Estefanía Valencia Saá

Alex Fernando Páez Salas

GUAYAQUIL - ECUADOR

Año: 2024

DEDICATORIA

Con inmensa gratitud y cariño dedico este trabajo de titulación a los pilares fundamentales de mi vida, a mi mamá, Adriana Saá, que con su amor incondicional, me ha llenado de fuerza para seguir adelante con mis proyectos y metas, a mi papá, Mauro Valencia, por gran habilidad de aprender y enseñar y su constante apoyo en cada paso que he dado, a mi hermano, Franco Valencia, por su amistad y compañerismo y a mi novio, Diego Galarraga, por su amor absoluto, su comprensión y su constante motivación ha siempre ser mejor. A cada uno de ustedes, mi mayor inspiración, va dedicado este logro, por ser parte importante de mi crecimiento como persona y como profesional.

Con infinito amor.

Ing. Vanessa Estefanía Valencia Saá

DEDICATORIA

En primer lugar, dedicó este título a Dios, por ser mi sabiduría, fortaleza por iluminar mi camino y guiar cada uno de mis pasos.

A mis padres, Sara y Cesar, por su amor incondicional y su apoyo constante. Sus sacrificios y enseñanzas han sido el cimiento sobre el cual he construido este logro.

A mi hermana Pamela, mi confidente y amiga. Tu respaldo y palabras de aliento han sido esenciales en los momentos de duda y dificultad. Gracias por estar siempre a mi lado.

A Isabel, mi novia, por ser mi refugio, apoyo, motivación e inspiración diaria. Su amor y comprensión me han impulsado a seguir adelante y alcanzar mis metas.

A toda mi familia, por su constante fe en mis capacidades. Sus palabras de aliento y apoyo han sido el motor que me ha permitido llegar hasta aquí.

A todos ustedes, dedico este logro con el corazón lleno de gratitud y amor.

Ing. Alex Fernando Páez Salas

AGRADECIMIENTOS

Al culminar esta etapa de mi formación académica, quiero expresar mi más profundo agradecimiento a las personas que me han acompañado a lo largo del camino hacia este logro.

A mi mamá, papá y hermano, Adriana, Mauro y Franco, por su constancia y dedicación, al guiarme, a nunca rendirme ante los desafíos nuevos que se han presentado lo largo de esta etapa.

A mi novio, Diego, quien me ha llenado de valor y fuerza para seguir adelante con mis sueños y por su apoyo, tanto emocional como económico, sin él, hubiera sido mas difícil alcanzar esta meta sola.

A cada uno de mis profesores y en especial a nuestro tutor, MSc. Alexander Prieto por su hábil dirección, apoyo y tiempo en la elaboración de este proyecto.

Gracias por ser y formar parte de una nueva meta más, de las tantas que vendrán.

Ing. Vanessa Estefanía Valencia Saá

AGRADECIMIENTOS

Voy a comenzar agradeciendo en primer lugar, a Dios por su guía y protección en cada paso de este camino.

A mis padres, Sara y Cesar, por su amor incondicional, por estar siempre a mi lado, por su apoyo constante. Gracias por sus sacrificios, por creer siempre en mí y por ser mi fuente de inspiración. A mi hermanita Pamela, por ser mi respaldo, mi ejemplo en todo momento. Que con su luz me ha dado ánimos para no rendirme, por siempre estar ahí cuando la necesito, su apoyo y palabras de aliento han sido fundamentales para superar los desafíos de este camino.

A Isabelita, mi novia, por ser mi motivación e inspiración diaria, por estar conmigo en todo momento, por creer en mi y hacerme ver mis capacidades, incluso cuando yo mismo dudaba de ellas, por levantarme en momentos que no doy más, por todo su amor y comprensión ya que me han impulsado a seguir adelante y alcanzar una meta más. A toda mi familia, por su constante fe en mis capacidades.

Ing. Alex Fernando Páez Salas

DECLARACIÓN EXPRESA

“Los derechos de titularidad y explotación, nos corresponde conforme al reglamento de propiedad intelectual de la institución; *Ing. Vanessa Valencia y Ing. Alex Páez*, damos nuestro consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual”

Ing. Vanessa E. Valencia S.

Ing. Alex F. Páez S.

EVALUADORES

Phd. Efrén V. Herrera Muentes
PROFESOR DE LA MATERIA

MSc. Alexander Prieto León
PROFESOR TUTOR

RESUMEN

El proyecto de titulación se centra en diseñar un sistema automatizado de conteo continuo de varillas mediante visión artificial y sistema SCADA, dirigido a mejorar la precisión y eficiencia en la industria siderúrgica. Se parte de la hipótesis de que integrar visión artificial y SCADA puede reducir errores humanos y optimizar costos operativos en comparación con métodos manuales.

En el desarrollo del proyecto, se utilizaron una cámara para captura de imágenes del proceso de empaquetado de varillas, software Python con OpenCV para procesamiento de imágenes, y técnicas avanzadas de segmentación. El modelo SCADA diseñado facilitó el monitoreo en tiempo real y la gestión eficiente del proceso de conteo de varillas.

Los resultados obtenidos demostraron una mejora significativa en la precisión del conteo y una reducción notable llegando a cero en los errores operativos. Además, se demuestra en base a los resultados que se obtiene la optimización de recursos y se minimizó la intervención manual, lo cual resulta en una operación más eficiente y rentable para las empresas siderúrgicas.

En conclusión, este proyecto no solo cumplió con los objetivos específicos de diseñar e implementar un sistema automatizado de conteo de varillas, sino que también proporcionó una guía práctica para su implementación en el sector siderúrgico, destacando su impacto positivo en la eficiencia operativa y la seguridad.

Palabras Clave: Visión Artificial, SCADA, Automatización, Segmentación.

ABSTRACT

The graduation project focuses on designing an automated system for continuous counting of rods using artificial vision and SCADA system, aimed at enhancing accuracy and efficiency in the steel industry. It starts with the hypothesis that integrating artificial vision and SCADA can reduce human errors and optimize operational costs compared to manual methods.

During the project development, a camera was used to capture images of the rod packaging process, Python software with OpenCV for image processing, and advanced segmentation techniques. The designed SCADA model facilitated real-time monitoring and efficient management of the rod counting process.

The results demonstrated a significant improvement in counting accuracy and a notable reduction, even reaching zero, in operational errors. Furthermore, based on these outcomes, resource optimization was achieved, and manual intervention was minimized, resulting in a more efficient and cost-effective operation for steel companies.

In conclusion, this project not only achieved its specific objectives of designing and implementing an automated rod counting system but also provided a practical guide for its implementation in the steel sector, highlighting its positive impact on operational efficiency and safety.

Keywords: Artificial Vision, SCADA, Automation, Segmentation.

ÍNDICE GENERAL

| | |
|---|-----------|
| DEDICATORIA | 2 |
| AGRADECIMIENTOS | 4 |
| DECLARACIÓN EXPRESA | 6 |
| EVALUADORES..... | 7 |
| RESUMEN | 1 |
| ABSTRACT..... | 2 |
| ABREVIATURAS..... | 5 |
| SIMBOLOGÍA | 6 |
| ÍNDICE DE FIGURAS..... | 7 |
| ÍNDICE DE TABLAS..... | 11 |
| CAPÍTULO 1 | 13 |
| 1. INTRODUCCIÓN..... | 13 |
| 1.1. Descripción del problema | 14 |
| 1.2. Justificación del problema..... | 15 |
| 1.3. Objetivos | 18 |
| 1.3.1. Objetivo General | 18 |
| 1.3.2. Objetivos Específicos | 18 |
| 1.4. Marco teórico..... | 19 |
| 1.4.1. Estado del arte..... | 19 |
| CAPÍTULO 2 | 40 |
| 2. HIPÓTESIS | 40 |
| 2.1. Variables de investigación..... | 40 |
| 2.1.1. Variables Independientes..... | 40 |
| 2.1.2. Variables Dependientes..... | 40 |
| 2.1.3. Parámetros de diseño | 40 |
| 2.1.3.1. Etapa de identificación y adquisición de datos en el tren de carga de varillas. | 41 |

| | | |
|---------------------|--|------------|
| 2.1.3.2. | Etapa de puesta en marcha del sistema SCADA. | 41 |
| 2.1.3.3. | Etapa de análisis de resultados tomados del diseño del sistema automático de Varillas. | 42 |
| 2.1.4. | Alternativa de solución | 43 |
| CAPÍTULO 3 | | 45 |
| 3. | Diseño y Desarrollo | 45 |
| 3.1. | Diseño del sistema automático de conteo de varillas utilizando visión artificial y sistema SCADA. | 45 |
| 3.2. | Diseño de la etapa de identificación y adquisición de datos en el tren de carga de varillas. | 45 |
| 3.3. | Diseño de la etapa de implementación de visión artificial para el conteo de varillas. | 47 |
| 3.3.1. | Captura y preprocesamiento..... | 47 |
| 3.3.2. | Segmentación de imágenes..... | 48 |
| 3.3.3. | Conteo de varillas. | 50 |
| 3.4. | Diseño del sistema SCADA | 51 |
| 3.5. | Selección de componentes | 57 |
| 3.5.1. | Selección del software para visión artificial | 57 |
| 3.5.2. | Selección del software para realizar el sistema SCADA..... | 58 |
| 3.6. | Desarrollo del sistema automatizado de conteo de varillas. | 59 |
| 3.7. | Desarrollo de la visión artificial del Sistema..... | 59 |
| 3.8. | Desarrollo del Sistema SCADA..... | 64 |
| CAPÍTULO 4 | | 75 |
| 4. | Pruebas y Resultados..... | 75 |
| 4.1. | Resultados de ejecución del sistema Scada. | 75 |
| 4.2. | Pruebas de Exactitud del conteo..... | 85 |
| 4.3. | Pruebas de velocidad del conteo | 87 |
| 4.4. | Análisis de Resultados | 89 |
| 4.4.1. | Análisis de resultados de la exactitud del conteo | 89 |
| 4.5. | Análisis de velocidad del conteo. | 93 |
| 4.6. | Análisis de Costos..... | 94 |
| 4.7. | Conclusiones | 97 |
| 4.8. | Recomendaciones | 99 |
| Bibliografía | | 100 |
| APÉNDICES | | 102 |

ABREVIATURAS

| | |
|-------|--|
| ESPOL | Escuela Superior Politécnica del Litoral |
| ASTM | American Society for Testing and Materials |
| NACE | National Association of Corrosion Engineer |
| SSC | Electrodo de Plata Cloruro de Plata |
| CSE | Electrodo de Cobre Sulfato de Cobre |
| HWL | High Water Level |
| LWL | Low Water Level |
| CIS | Inspección pasó a paso, medición de potenciales de encendido |
| MPY | Milésimas de pulgadas por año |

SIMBOLOGÍA

| | |
|-----|------------------------|
| mil | Milésima de pulgada |
| mg | Miligramo |
| pH | Potencial de Hidrógeno |
| m | Metro |
| mV | Milivoltio |
| Cu | Cobre |
| Ni | Níquel |
| C | Carbono |
| Mn | Manganeso |
| P | Fósforo |

ÍNDICE DE FIGURAS

| | |
|--|----|
| <i>Figura 1.1</i> | 19 |
| <i>Resultados del programa de conteo de varillas</i> | 19 |
| <i>Figura 1.2</i> | 20 |
| <i>Etapas del programa de conteo de varillas</i> | 20 |
| <i>Figura 1.3</i> | 21 |
| <i>Fotos con histogramas de objetos</i> | 21 |
| <i>Figura 1.4</i> | 25 |
| <i>Características del sistema SCADA</i> | 25 |
| <i>Figura 1.5</i> | 26 |
| <i>Arquitectura básica para una etapa de procesamiento de imágenes.</i> | 26 |
| <i>Figura 1.6</i> | 29 |
| <i>Implementación de SCADA en indusoft Web Studio Educational</i> | 29 |
| <i>Figura 2.1</i> | 44 |
| <i>Análisis de resultados del diseño del sistema</i> | 44 |
| <i>Figura 3.1</i> | 46 |
| <i>Diagrama de flujo del diseño de la etapa y adquisición de datos</i> | 46 |
| <i>Figura 3.2</i> | 48 |
| <i>Diagrama de flujo de la estructura de programación para captura y procesamiento de imágenes</i> | 48 |
| <i>Diagrama de flujo proceso de segmentación de imágenes</i> | 49 |
| <i>Figura 3.4</i> | 50 |

| | |
|---|-----------|
| <i>Diagrama de flujo de la estructura de programación de conteo de varillas.</i> | <i>50</i> |
| <i>Figura 3.5</i> | <i>51</i> |
| <i>Diseño del sistema SCADA.....</i> | <i>51</i> |
| <i>Figura 3.6</i> | <i>52</i> |
| <i>Distribución interfaz de usuario.....</i> | <i>52</i> |
| <i>Figura 3.7</i> | <i>54</i> |
| <i>Programación orientada a objetos.</i> | <i>54</i> |
| <i>Figura 3.8</i> | <i>55</i> |
| <i>Integración con proceso de visión artificial.</i> | <i>55</i> |
| <i>Figura 3.9</i> | <i>61</i> |
| <i>Figura 3.10</i> | <i>63</i> |
| <i>Imagen del programa con una foto del antes y después del proceso.</i> | <i>63</i> |
| <i>Figura 3.11</i> | <i>64</i> |
| <i>Página del inicio del programa.....</i> | <i>64</i> |
| <i>Figura 3.12</i> | <i>65</i> |
| <i>Pantalla del Historial</i> | <i>65</i> |
| <i>Figura 3.13</i> | <i>65</i> |
| <i>Página de visualización de imágenes.....</i> | <i>65</i> |
| <i>Figura 3.14</i> | <i>66</i> |
| <i>Página de configuración</i> | <i>66</i> |
| <i>Figura 3.15</i> | <i>67</i> |
| <i>Página de ayuda.....</i> | <i>67</i> |
| <i>Figura 3.16</i> | <i>68</i> |
| <i>Ventanas de alertas.....</i> | <i>68</i> |

| | |
|--|-----------|
| Figura 3.17 | 69 |
| Figura 3.18 | 72 |
| Figura 3.19 | 73 |
| Figura 4.1 | 75 |
| Imagen de la Pantalla de seguridad | 75 |
| Figura 4.2 | 76 |
| Imagen de la Pantalla de seguridad | 76 |
| Figura 4.3 | 76 |
| Imagen de la Pantalla de inicio claro | 76 |
| Figura 4.4 | 77 |
| Imagen de la Pantalla de seguridad oscuro. | 77 |
| Figura 4.5 | 78 |
| Imagen de la Pantalla de inicio y procesamiento | 78 |
| Figura 4.6 | 78 |
| Imagen de la Pantalla de inicio y procesamiento | 78 |
| Figura 4.7 | 79 |
| Imagen de la Pantalla de inicio y procesamiento | 79 |
| Figura 4.8 | 80 |
| Imagen de la Pantalla de inicio y procesamiento | 80 |
| Figura 4.9 | 81 |
| Imagen de la Pantalla de inicio y procesamiento | 81 |
| Figura 4.10 | 81 |
| Imagen de la Pantalla del historial | 81 |
| Figura 4.11 | 82 |

| | |
|--|-----------|
| <i>Imagen de la Pantalla de visualización de imágenes.....</i> | <i>82</i> |
| <i>Figura 4.12</i> | <i>83</i> |
| <i>Imagen de la Pantalla de ayuda y documentación.....</i> | <i>83</i> |
| <i>Figura 4.13</i> | <i>84</i> |
| <i>Imagen de la Pantalla de configuración</i> | <i>84</i> |
| <i>Figura 4.14</i> | <i>89</i> |
| <i>Grafica de comparación conteo manual vs conteo de software.....</i> | <i>89</i> |
| <i>Figura 4.15</i> | <i>93</i> |
| <i>Imagen del promedio del error del sistema.....</i> | <i>93</i> |

ÍNDICE DE TABLAS

| | |
|---|----|
| <i>Tabla 1.1</i> | 24 |
| <i>Ventajas y desventajas de utilizar este tipo de sensores</i> | 24 |
| <i>Tabla 1.2</i> | 27 |
| <i>Comparación de softwares para la implementación de visión artificial</i> | 27 |
| <i>Tabla 1.3</i> | 30 |
| <i>Diseño de HMIs</i> | 30 |
| <i>Tabla 1.4</i> | 32 |
| <i>Descripción de softwares importantes para aplicaciones SCADA</i> | 32 |
| <i>Tabla 1.5</i> | 35 |
| <i>Softwares SCADA para utilización en el proyecto</i> | 35 |
| <i>Tabla 1.6</i> | 36 |
| <i>Características principales del software SCADA</i> | 37 |
| <i>Tabla 1.7</i> | 38 |
| <i>Softwares ideales para integración de visión artificial</i> | 38 |
| <i>Tabla 1.8</i> | 38 |
| <i>Métodos para el procesamiento de imágenes</i> | 39 |
| <i>Tabla 3.1</i> | 57 |
| <i>Softwares para visión artificial</i> | 57 |
| <i>Tabla 3.2</i> | 58 |
| <i>Softwares para realizar el sistema SCADA</i> | 58 |
| <i>Tabla 3.3.</i> | 62 |
| <i>Comandos fundamentales</i> | 62 |

| | |
|---|-----------|
| Tabla 3.4 | 70 |
| Tabla de comandas y parámetros | 70 |
| Tabla 3.5 | 74 |
| Tabla de comandas y parámetros | 74 |
| Tabla 4.1 | 86 |
| Tabla de comparación de tomas d imágenes de varillas | 86 |
| Tabla 4.2 | 87 |
| Tabla de valores de velocidad de respuesta manual | 88 |
| Tabla 4.3 | 88 |
| Tabla de valores de velocidad de respuesta con el software | 88 |
| Tabla 4.4 | 91 |
| Tabla de resultados del cálculo de error | 91 |
| Tabla 4.5 | 92 |
| Tabla de resultados del cálculo de erros relativo | 92 |
| Tabla 4.6 | 94 |
| Tabla de análisis de costos | 94 |

CAPÍTULO 1

1. INTRODUCCIÓN

La industria metalúrgica juega un papel fundamental en el desarrollo económico del país. En este sector, la precisión y la eficiencia en los procesos de producción son esenciales para garantizar la calidad del producto final y la competitividad en el mercado.

El conteo de varillas es una tarea crucial en la industria metalúrgica, ya que determina la cantidad de material disponible para la producción y el control de inventario. Tradicionalmente, este proceso se ha realizado de forma manual por operarios, lo que puede ocasionar errores debido a la fatiga, la falta de atención o las condiciones ambientales. Los errores en el conteo de varillas pueden tener un impacto significativo en la industria metalúrgica, incluyendo, pérdidas económicas por el exceso; la falta de varillas puede generar pérdidas de material, lo que afecta directamente los costos de producción, retrasos en la producción debido a un conteo inexacto que puede ocasionar retrasos en la entrega de productos a los clientes, lo que daña la reputación de la empresa, problemas de calidad si el número de varillas no es el adecuado, el producto final puede tener defectos que lo hacen inutilizable.

Para solucionar estos problemas, se propone la implementación de un sistema automatizado de conteo de varillas mediante visión artificial y sistema SCADA.

La visión artificial es una tecnología que permite a las máquinas "ver" y comprender el mundo que las rodea. Se utiliza en una amplia variedad de aplicaciones industriales, incluyendo la inspección de productos, el control de calidad y la robótica.

Con la visión artificial se pueden identificar las varillas, la cámara del sistema captura imágenes de las varillas y un software especializado las identifica y las separa del fondo, el software determina el número de varillas presentes en la imagen y genera un conteo preciso, para luego pasar a verificar la calidad de las varillas

El sistema SCADA (Supervisory Control and Data Acquisition) es un software que se utiliza para la supervisión y el control de procesos industriales. Permite la visualización de datos en tiempo real, la generación de alarmas y la toma de decisiones automatizada.

En el caso del conteo de varillas, el sistema SCADA se puede utilizar para monitorizar el proceso de conteo en tiempo real permitiendo visualizar las imágenes de las varillas y el conteo en tiempo real, el sistema también puede generar alarmas si se detecta un error en el conteo o si se encuentran varillas defectuosas y puede integrarse con el sistema de control de producción para ajustar la cantidad de varillas que se utilizan en cada lote.

La implementación de un sistema automatizado de conteo de varillas mediante visión artificial y sistema SCADA ofrece un conjunto de beneficios que pueden mejorar significativamente la eficiencia, la precisión y la calidad en la industria metalúrgica.

Este trabajo de titulación se centrará en el diseño, la implementación y la evaluación de un sistema de este tipo para una empresa metalúrgica específica. Se analizarán los requisitos del sistema, se seleccionarán las tecnologías adecuadas y se desarrollará un prototipo para su evaluación.

Se espera que la implementación del sistema automatizado de conteo de varillas contribuya a mejorar la competitividad de la empresa metalúrgica en el mercado.

Descripción del problema

El sector siderúrgico ecuatoriano juega un rol fundamental en la economía del país, abasteciendo de materia prima a diversos sectores industriales. Un proceso crucial dentro de la producción siderúrgica es la producción de varillas, la cual se realiza de continua en la mayoría de las empresas.

El conteo de varillas en diversas empresas es de forma manual, esto quiere decir que los operadores son los encargados de realizar el conteo, el cual se desarrolla con un bajo nivel de exactitud y precisión por lo que el margen de error es alto, lo que ocasiona que los lotes no sean iguales entre sí. El problema se da en la mesa

de enfriamiento, cuando los operadores cuentan y separan las varillas para formar el lote, esta es la principal razón para realizar el estudio.

Otro problema que se puede observar es que los operadores pueden estar expuestos a riesgos para la salud debido a la falta de métodos de trabajo estandarizados y equipos de seguridad adecuados para las operaciones.

Esta situación genera serias preocupaciones porque los operadores están expuestos a muchos riesgos potenciales que van desde lesiones físicas hasta enfermedades profesionales.

La falta de procesos de trabajo estandarizados dificulta la implementación de medidas preventivas consistentes y efectivas para proteger la salud y seguridad de los trabajadores.

Además, la falta de equipos de seguridad adecuados agrava este problema porque deja a los operadores vulnerables a accidentes y enfermedades relacionadas con sus actividades laborales.

Por lo tanto, es esencial abordar estas deficiencias implementando prácticas laborales seguras y proporcionando equipos de protección personal adecuados para garantizar un entorno de trabajo saludable y seguro para todas las personas involucradas.

Justificación del problema

Tomando en cuenta un análisis que se realizó en una empresa de producción de varillas en la provincia de Cotopaxi, se ha visto la necesidad de realizar una simulación de un sistema de conteo de varillas. La automatización del conteo de varillas mediante tecnologías como la visión artificial y el aprendizaje automático se presenta como una solución viable para las problemáticas descritas, lo cual mejora el proceso y evita el margen de error en los lotes, también reducirá el número de accidentes laborales en dicha actividad.

Un sistema automatizado permitiría:

- Mejorar la precisión del conteo: Reduciendo significativamente los errores humanos.
- Aumentar la eficiencia: Agilizando el proceso de conteo y liberando tiempo para otras tareas.

- Rapidez: Los sistemas automatizados pueden contar las varillas a una velocidad mucho mayor que un humano.
- Reducir los riesgos de seguridad: Eliminando la manipulación manual de varillas pesadas.
- Optimizar la gestión de inventario: Con información precisa y actualizada sobre la disponibilidad de materiales.
- Mejorar la satisfacción del cliente: Al garantizar la entrega de pedidos completos y precisos.

Vista la necesidad del problema, se plantea incrementar un sistema SCADA para completar el programa de conteo de varillas, dándonos como ventajas:

a) Supervisión y control en tiempo real:

- El sistema SCADA permite la visualización en tiempo real del proceso de conteo de varillas, incluyendo imágenes, datos y estadísticas.
- Permite la intervención manual del operador en caso de errores o situaciones anómalas.
- Ofrece una visión completa del proceso, facilitando la toma de decisiones y la identificación de áreas de mejora.

b) Mayor precisión y eficiencia:

- La automatización del conteo mediante visión artificial reduce significativamente el error humano, aumentando la precisión de los datos.
- El sistema SCADA puede automatizar tareas repetitivas, como la generación de informes y la gestión de alarmas.
- Esto libera tiempo para que los operadores se concentren en tareas más estratégicas.

c) Integración con otros sistemas:

- El sistema SCADA puede integrarse con otros sistemas de la planta, como el sistema de control de producción o el sistema de gestión de inventario.
- Esto permite una mejor coordinación y optimización de los procesos en general.

d) Seguridad y confiabilidad:

- El sistema SCADA puede proteger el acceso a los datos y las funciones del sistema mediante la implementación de diferentes niveles de seguridad.
- El sistema puede registrar eventos y alarmas, facilitando la detección y resolución de problemas.
- La automatización del conteo reduce el riesgo de accidentes y lesiones en el trabajo.

También se ha identificado que el sistema demandaría la adquisición y configuración de equipos especializados, tales como cámaras con alta resolución, sensores de proximidad y software para análisis de imágenes. Además, sería necesario integrar estos componentes en el entorno actual de producción. Esta infraestructura tecnológica no solo aumentaría la precisión y eficacia del conteo de varillas, sino que también contribuiría a disminuir los accidentes laborales al reducir la manipulación manual de materiales pesados. En conclusión, adoptar un sistema automatizado para contar varillas traería beneficios tangibles en términos de seguridad, eficiencia y gestión del inventario para la empresa.

Objetivos

1.1.1. Objetivo General

Diseñar un sistema automatizado de conteo continuo de varillas mediante visión artificial para la industria siderúrgica.

1.1.2. Objetivos Específicos

- Analizar los métodos actuales de conteo de varillas en la empresa siderúrgica con el propósito de identificar áreas de mejora.
- Identificar las necesidades y requisitos para la implementación de un sistema automatizado de conteo.
- Diseñar un prototipo de sistema automatizado de conteo de varillas que integre la visión artificial.
- Diseñar el modelo SCADA para el programa.
- Evaluar la precisión, eficiencia y seguridad del sistema automatizado en un entorno real.
- Elaborar una guía para la implementación del sistema automatizado en las empresas siderúrgicas de Ecuador.

Marco teórico

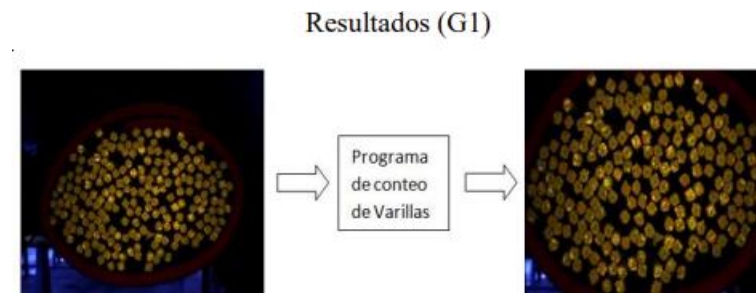
1.1.3. Estado del arte

Conteo de varillas de acero por procesamiento de imágenes (Alejandro & vadillo Vidal, 2013)

Los autores en este proyecto implementan; almacenamiento de imágenes con la presentación de sus respectivos resultados en donde detallan que luego de procesar una imagen se procede a realizar las sumatoria de cuantas varillas están almacenadas por bloque, para una posterior etiqueta de nombres tal como se aprecia en la figura 1.1, el método para facilitar el proceso de revisión o inspección es novedoso ya que otorgan puntos pequeños que detectan si existe una varilla.

Figura 1.1

Resultados del programa de conteo de varillas



Nota: En la siguiente imagen muestra los resultados obtenidos en cuanto a lo propuesto. Tomado de (Alejandro & vadillo Vidal, 2013)

En el proyecto de tesis también se analiza una etapa de adquisición de imágenes, que explica una selección de una cámara digital que cumple los requerimientos para un proceso industrial. Para la adquisición se implementan dos tipos de sensores de imagen: CCD y CMOS el primero ofrece alta sensibilidad a la luz y realiza capturas rápidas mientras tanto el CMOS es un sensor más económico y permite el uso en cámaras pequeñas, aunque uno de los parámetros de influencia es que la calidad de imagen no es tan alta como la que otorga el sensor CCD. La descripción de la imagen que es una parte importante del proyecto es en

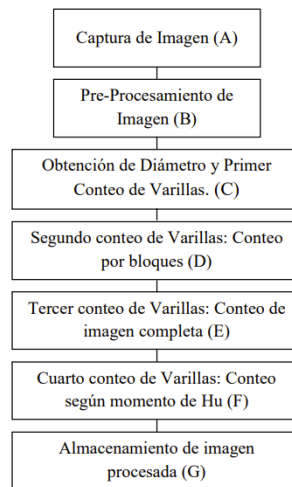
representación de columnas y filas de píxeles. Los autores realizan un análisis entre los sensores lineales y los sensores de área que son las dos arquitecturas principales, cada una con diferentes resoluciones. La elección entre estos factores depende de las necesidades específicas de la aplicación, considerando la sensibilidad a la luz, la calidad de la imagen, el tamaño y el costo. (Alejandro & vadillo Vidal, 2013)

Otro de los factores importantes a analizar es el algoritmo de programa de conteo de varillas, en donde el programa realiza el conteo de las varillas basándose en una imagen binaria, tomando en cuenta que antes se debe realizar una etapa de pre-procesamiento, comenzando con diversos filtros de detección de carillas, observando que estos filtros son más rigurosos conforme pasan. Las etapas de este algoritmo se pueden apreciar en la figura 1.2. (Alejandro & vadillo Vidal, 2013).

Los autores también destacan algunas características de Python en donde definen, su reconocimiento como un lenguaje de programación potente y accesible, destacándose por sus estructuras de datos avanzadas y su enfoque claro hacia la programación orientada a objetos. Su sintaxis elegante y su tipado dinámico, combinados con su capacidad de interpretación, lo convierten en una opción ideal para la creación rápida de aplicaciones y scripts en múltiples ámbitos y plataformas. (RENNARD, AUGUSTO, & JIMENEZ, 2024)

Figura 1.2

Etapas del programa de conteo de varillas



Nota: En la siguiente figura muestra en un algoritmo las etapas del conteo de varillas tomado de (Alejandro & vadillo Vidal, 2013)

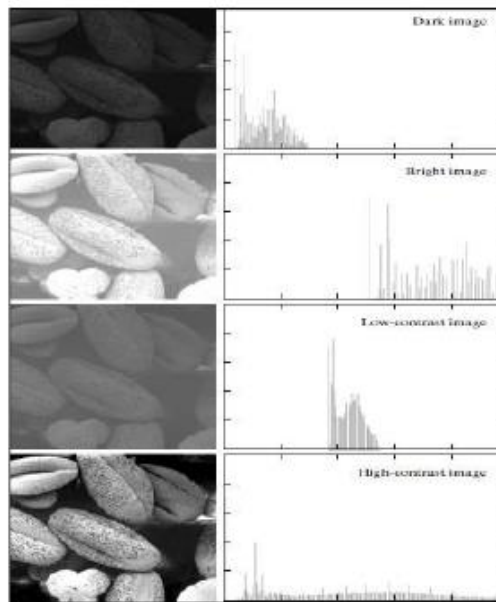
También se puede observar que los autores describen componentes clave en un sistema de adquisición de imágenes, además mencionan los espacios de color en la visualización y representación de las capturas de imagen, concluyendo con resultados de pruebas de confiabilidad otorgando recomendaciones para mejoras de este algoritmo.

Identificación de objetos en tiempo real utilizando técnicas y clasificadores de visión artificial para el reconocimiento de patrones (Sánchez, 2011)

Los autores en el texto readaptan diferentes temas interesantes que apoyarán a realizar el presente proyecto, uno de estos es la técnica de la segmentación para el procesamiento de imágenes, que trata en dividir una imagen en diferentes regiones o elementos que corresponden a objetos o algunas partes de este objeto. Esta técnica tiene como objetivo identificar los límites de los objetos que se requieren analizar en la imagen. En este caso se realiza la segmentación por pixeles que consiste en solo considerar el valor gris de cada píxel para concluir si es o no parte de dicho objeto, se lo determina encontrando el rango de valores de gris característicos del objeto. En este método se aplica un histograma con los valores característicos. En la figura 1.3 se aprecia imágenes con sus respectivos histogramas. (Sánchez, 2011)

Figura 1.3

Fotos con histogramas de objetos



Nota: En la siguiente figura se muestra imágenes con histogramas con los valores respectivos. Tomado de (Sánchez, 2011).

Se implementa también operaciones morfológicas en la que destacamos la dilatación y la erosión, en la primera los autores la utilizan para rellenar pequeños huecos del objeto, conectar objetos que están separados por pequeñas distancias, y para aumentar la dimensión de los objetos logrando que sean más fáciles al momento de la detección. Se aplica un cuadrado estructural de 3x3 que se define como una matriz de pixeles tomando en cuenta que esto se agrega en el borde del objeto original. En cambio, la erosión se la aplica para eliminar los pequeños ruidos de las imágenes, separar objetos que están muy juntos, y la disminución de las dimensiones de los objetos para que sean más fáciles de analizar. (Sánchez, 2011)

En el proyecto también se encuentra la implementación de técnicas de reconocimiento de patrones, las cuales se dividen en dos que son: decisiones teóricas y estructurales en donde la primera trabaja con la longitud, área y textura del objeto, y por otro lado la decisión estructural estudia descriptores cualitativos. En cuanto los momentos menciona al momento de HU y SIFT, el primero lo utiliza para reconocer patrones de imágenes incluyendo: reconocimiento de objetos, registro de imágenes y coincidencia de imágenes; en cuanto a SIFT se implementa para la detección de un punto característico que tiene un objeto en una imagen,

esto se reconoce como la invariancia de escala de rotación, transformaciones afines e iluminación. (Sánchez, 2011)

Otro parámetro fundamental utilizado en el procesamiento de imágenes fue la lógica difusa que tratan de definir diferentes tipos de reglas, teniendo valores intermedios que pueden ser diferentes de 0 o 1 como diferencia de la lógica tradicional, obteniendo un alto rendimiento al momento de la clasificación. (Sánchez, 2011)

Los autores también complementan que la detección de bordes es fundamental en el procesamiento de imágenes para el diagnóstico asistido por ordenador, ya que permite identificar cambios bruscos en la intensidad de los píxeles y obtener información precisa sobre las áreas de interés. Que en este tipo de procedimientos obteniendo métodos para mejorar características y detectar bordes en imágenes mediante el análisis del histograma de distribución de píxeles y el uso del gradiente morfológico. (Niño Rondón, Castellano Carvajal, Castro Casadiego, Medina Delgado, & Guevara Ibarra, 2022)

La correcta interpretación de histogramas es esencial en el procesamiento de imágenes porque permite una detección precisa de bordes, mejora la calidad de las imágenes ajustando los niveles de intensidad de los píxeles, optimiza el rendimiento del procesamiento y facilita la validación y comparación de nuevas técnicas frente a métodos existentes, asegurando mejoras significativas en la precisión y eficiencia del diagnóstico asistido por ordenador. (Niño Rondón, Castellano Carvajal, Castro Casadiego, Medina Delgado, & Guevara Ibarra, 2022).

Diseño de un sistema SCADA para una planta de clasificación con visión artificial (Lojan & Iñiguez, 2009)

En este proyecto los autores utilizan los sensores CCD, lo implementan como sensor de imagen que ofrece una alta sensibilidad a la luz, un amplio rango dinámico y un bajo nivel de ruido. Hay que tomar en cuenta que este tipo de sensores, son más caros, consumen más energía y son más lentos que los sensores CMOS. El funcionamiento de estos sensores se define así:

La luz incide sobre el sensor CCD y es absorbida por los fotodiodos.

Cada fotodiodo convierte la luz en una cantidad de carga eléctrica.

La carga eléctrica de cada fotodiodo es transportada por el registro de desplazamiento a un convertidor analógico-digital.

El convertidor analógico-digital convierte la carga eléctrica en una señal digital que puede ser procesada por el ordenador de la cámara. (Lojan & Iñiguez, 2009)

Tabla 1.1

Ventajas y desventajas de utilizar este tipo de sensores

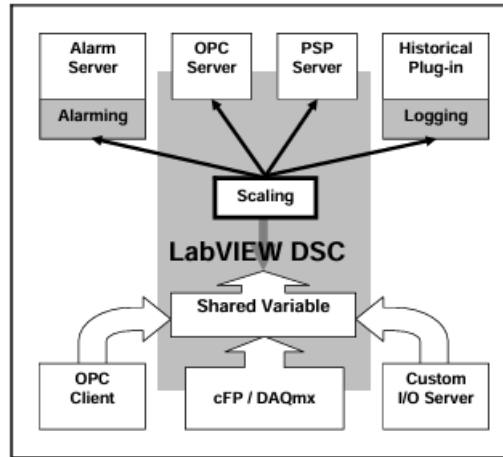
| Ventajas de los sensores CCD | Desventajas de los sensores CCD |
|--|--|
| Alta sensibilidad a la luz: Los sensores CCD son más sensibles a la luz que los sensores CMOS, lo que significa que pueden capturar imágenes con menos ruido en condiciones de poca luz. | Mayor costo: Los sensores CCD son más caros que los sensores CMOS. |
| Mayor rango dinámico: Los sensores CCD tienen un mayor rango dinámico que los sensores CMOS, lo que significa que pueden capturar una mayor variedad de tonos de color. | Mayor consumo de energía: Los sensores CCD consumen más energía que los sensores CMOS. |
| Menos ruido: Los sensores CCD producen menos ruido que los sensores CMOS, lo que da como resultado imágenes más nítidas. | Más lentos: Los sensores CCD son más lentos que los sensores CMOS. |

Nota: Está tabla muestra que ventajas y desventajas se pueden encontrar al ocupar sensores CDD tomado de (Lojan & Iñiguez, 2009)

También en este proyecto se implementa un sistema SCADA con visión artificial, en este caso se realiza con el software LabVIEW para monitorear y controlar un PLC es decir conexión OPC y PLC, también se observa que en el proyecto se obtienen variables en tiempo real y permite a los usuarios interactuar con el sistema por sus componentes de interfaz de usuario, en la figura 1.4 se visualiza las características del sistema SCADA realizado en el software LabVIEW. (Lojan & Iñiguez, 2009)

Figura 1.4

Características del sistema SCADA



Nota: Se visualiza las características del sistema SCADA realizado en el software LabVIEW. Tomado de (Lojan & Iñiguez, 2009)

También los autores mencionan que se puede utilizar este tipo de softwares en investigación que utiliza las capacidades avanzadas de LabVIEW y el protocolo ZigBee para la comunicación inalámbrica, permitiendo un control eficiente y supervisión remota de los procesos de coagulación y floculación en una planta de tratamiento de agua potable (PTAP). Se aplicó la metodología de descomposición del trabajo en fases (WBS) para organizar de manera jerárquica y detallada las tareas. La investigación exploratoria con diseño experimental facilitó una exitosa integración entre hardware y software. (Algarín, Llanos, & Hernández, 2012)

Se pueden obtener resultados mostraron una perfecta sincronización entre las unidades remotas y el programa principal, comunicación inalámbrica eficiente con los módulos ZigBee, tiempos de retardo imperceptibles y alta inmunidad al ruido. Se logró un programa principal robusto, con una interfaz sencilla y amigable, que demanda pocos recursos de cómputo para el control y visualización de datos de temperatura y pH. (Algarín, Llanos, & Hernández, 2012)

Además, se detalla, que se puede desarrollar sistemas específicos para monitorizar y recolectar datos en tiempo real de irradiancia, temperatura, corriente y voltaje de dos paneles solares conectados en paralelo. (Boada Robayo, 2024)

También se puede complementar este tipo de sistemas de monitoreo una transmisión de datos mediante una conexión LAN, asegurando la disponibilidad de la información en tiempo real y proporcionando una herramienta valiosa para el análisis energético y la toma de decisiones basadas en datos. (Boada Robayo, 2024).

Desarrollo e Implementación de un Sistema de Visión Artificial Basado en Lenguajes de Uso Libre para un Sistema Seleccionador de Productos de un Centro Integrado de Manufactura (CIM) (Amaya, Velásquez, & Torres, 2015)

En el proyecto se destaca la implementación de la arquitectura del sistema de identificación de imágenes en donde se destacan puntos como:

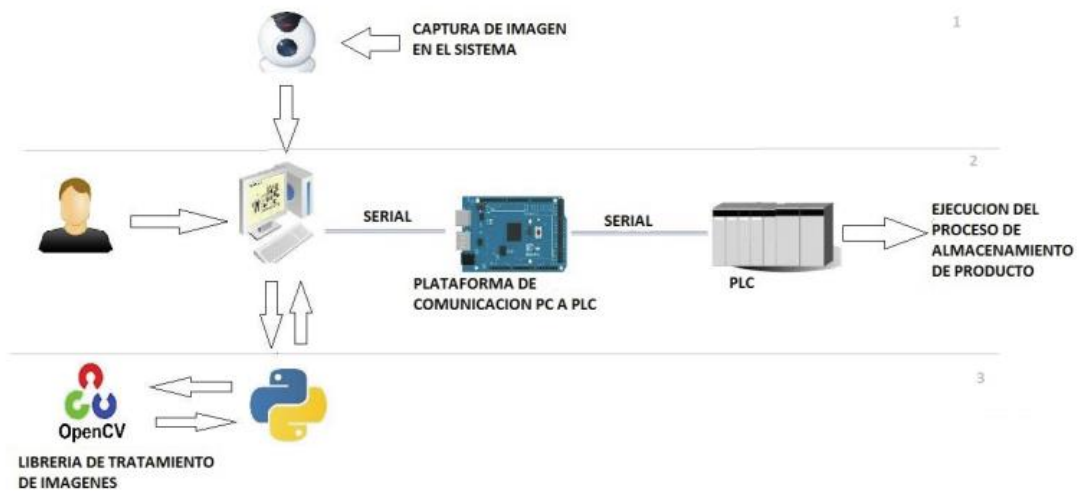
Interfaz de usuario: los autores implementan un sistema con un operador humano, con una interfaz gráfica, donde la característica principal es seleccionar y escoger el objeto basado en su forma y color. (Amaya, Velásquez, & Torres, 2015)

Selección del lenguaje y la librería: Se realiza un análisis de diferentes lenguajes de programación, en donde detalla que Python junto la librería OpenCv, llena todos los requisitos para trabajar con el sistema de visión artificial. Se implementa Open CV como única librería para tratamiento de imágenes. Ya que con Python y su alto nivel de facilidad y modificación se relacionan perfectamente para el tema de procesamiento de imágenes, Además, la disponibilidad de recursos de aprendizaje y la eficiencia en el consumo de recursos son consideraciones clave para elegir este lenguaje. (Amaya, Velásquez, & Torres, 2015)

También detalla las ventajas que el software Python tiene como su naturaleza de código abierto, la abundante información disponible, partiendo desde literatura básica hasta cursos virtuales. (Amaya, Velásquez, & Torres, 2015)

Figura 1.5

Arquitectura básica para una etapa de procesamiento de imágenes.



Nota: se visualiza la arquitectura básica para una etapa de procesamiento de imágenes. Tomado de (Amaya, Velásquez, & Torres, 2015).

Tabla 1.2

Comparación de softwares para la implementación de visión artificial

| Lenguaje | Librerías compatibles | Ventajas | Desventajas |
|----------|---------------------------|---|--|
| C++ | Torch3vision, VLX, openCV | Permite modificar todos los detalles del código, ejecución rápida, compatible con todas las librerías | Complejidad de código alta, difícil de leer y ejecutar en plataformas diferentes |
| Java | OpenCV | Ejecución en cualquier plataforma sin recompilar, escritura y lectura de código simple, integración a múltiples librerías, gran cantidad de bibliografía disponible | Consumo de recursos alto |
| Python | OpenCV | Ejecución en cualquier plataforma sin recompilar, escritura y lectura de código simple, integración a múltiples librerías, gran | Consumo de recursos alto |

Nota: Se visualiza una tabla de comparaciones de softwares para la implementación de visión artificial, detallando sus librerías, ventajas y desventajas que estos poseen. Tomado de (Amaya, Velásquez, & Torres, 2015).

También se detalla que el uso de Python en este tipo de procedimientos es crucial por varias razones. Python es conocido por su simplicidad y legibilidad, lo que facilita el desarrollo y mantenimiento del código, especialmente en proyectos complejos como el procesamiento de imágenes. La extensa biblioteca de Python, como Cv2 (OpenCV), proporciona herramientas poderosas y eficientes para el manejo y procesamiento de imágenes, incluyendo la captura de video y la transformación de imágenes a escala de grises. (Zavaleta, Baquedano, Jhajaira, R., & M., 2022)

Además, Python permite una rápida prototipación y experimentación, lo cual es esencial en proyectos que requieren ajustes continuos y optimizaciones, como los algoritmos de detección de bordes. La compatibilidad de Python con otras herramientas y librerías, como NumPy y SciPy, también mejora las capacidades de procesamiento de datos y análisis, permitiendo implementar algoritmos complejos de manera eficiente. (Zavaleta, Baquedano, Jhajaira, R., & M., 2022)

Elaboración de un sistema SCADA para el monitoreo y control en una línea automatizada de snacks fritos y acceso a tecnología 4.0 (Corella & Reyes, 2022)

El proyecto implementa un sistema SCADA para el monitoreo y control en una línea de automatización industrial, en donde se visualizan procesos de adquisición y almacenamiento de datos, ejecución de acciones de control, representación gráfica de variables de proceso y una arquitectura abierta y flexible, siendo posible conectar con aplicaciones de base de datos. (Corella & Reyes, 2022)

El software que se aplica en la implementación del SCADA es indusoft Web Studio Educational, que tiene una suite de herramientas para la automatización de proyectos que son altamente modulares, para el desarrollo de una gran gama de aplicaciones, se define en interfaces Hombre-Máquina (HMI), en donde se

encuentran sistemas completos de supervisión, adquisición de Datos (SCADA) y control, todo esto implementado en la industria. Como se muestra a continuación en la figura 1.6.

Figura 1.6

Implementación de SCADA en indusoft Web Studio Educational



Nota: Se indica una aplicación creada utilizando el software InduSoft.

También se detallan normas importantes para la realización de sistemas SCADA como se puede apreciar a continuación:

La norma estándar IEC-61131: esta se aplica en el proyecto ya que involucra autómatas programables y sus periféricos, tiene como objetivo definir características para el diseño y selección de PLCs, recomendaciones para la escritura de instrucciones, lenguajes de programación y topologías de red. Su principal finalidad es estandarizar el uso de PLCs en la industria. Incluye la

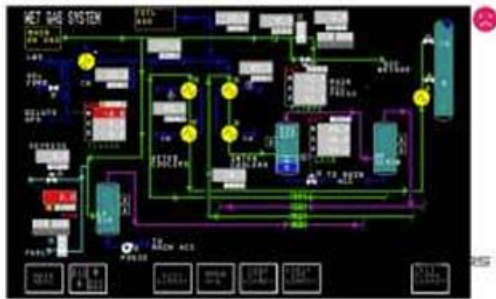
aceptación de cinco lenguajes de programación de PLCs, divididos en gráficos (LD, FBD, SFC) y literales (IL, ST). Los beneficios que se obtiene con la norma es la reducción de costos en recursos humanos, detección de fallos, técnicas de programación, prevención de problemas y comunicación entre departamentos. (Corella & Reyes, 2022)

La normativa ISA 101-High Performance HMI: esta norma implementa directrices que se utilizan para el diseño y desarrollo de HMIs, para la aplicación de monitoreo y control de procesos industriales. Su objetivo es mejorar la productividad y reducir costos, mediante el fácil entendimiento, diseño adecuado de pantalla, y opciones claras para observar alertas y trabajo normal- En la normativa también se encuentra el beneficio de identificar errores comunes en el diseño de HMIs , tal como se puede apreciar a continuación:

Tabla 1.3

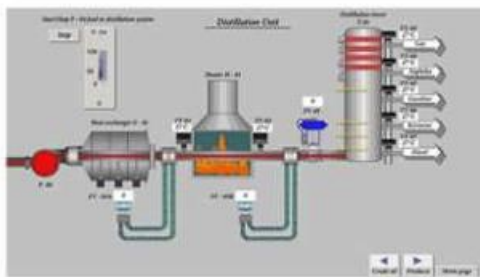
Diseño de HMIs

| Imágenes | Característica | Descripción |
|----------|----------------------------|---|
| | Amplio espectro de colores | El uso de una amplia gama de colores puede ser confuso y abrumador para los usuarios, dificultando la identificación de la información importante. |
| | Gráficos innecesarios | Los gráficos innecesarios pueden distraer la atención de los usuarios de la información esencial y dificultar la comprensión de los datos. |
| | Distracciones visuales | Las distracciones visuales, como animaciones o elementos parpadeantes, pueden desviar la atención de los usuarios de la tarea que están realizando. |



Difícil entendimiento de los datos presentados

La información presentada en la HMI debe ser clara, concisa y fácil de entender. Si los datos son difíciles de entender, los usuarios pueden cometer errores o no tomar las decisiones correctas.



Representación de alarmas incorrectas

La representación incorrecta de las alarmas puede ser peligrosa, ya que los usuarios podrían no ser conscientes de una situación de emergencia.



Inconsistencia en la navegación entre pantallas

La navegación entre pantallas debe ser consistente y lógica. Si la navegación es inconsistente, los usuarios pueden frustrarse y perder tiempo buscando la información que necesitan.

Nota: En la siguiente tabla se muestra características y descripciones sobre HMIs tomado de (Corella & Reyes, 2022)

Mini proyecto automatización industrial (AUTI) (Cabús, Navarrete, & Porras, 2004)

En este proyecto los autores plantean una explicación del concepto de un sistema SCADA, enfocados en la implementación de plantas industriales automatizadas. Destacando características del software para controlar, supervisar y adquirir datos, visualizados desde una pantalla de un ordenador, aplicando interfaz gráfica que configura un usuario. También se detallan características y requerimientos necesarios para el aprovechamiento máximo de un sistema SCADA. (Cabús, Navarrete, & Porras, 2004)

En el proyecto se menciona algunos comentarios de los diferentes softwares analizados en donde se analiza, INTOUCH que muestra ser útil para el entorno educativo e industrial, ofreciendo a los operarios la oportunidad de interactuar con un sistema SCADA real y comprensible. La posibilidad de realizar prácticas demostrativas con una interfaz intuitiva facilita el aprendizaje y comprensión de los conceptos fundamentales de control y automatización. El software también permite una rápida configuración y adaptación a diferentes escenarios y necesidades de los módulos, lo cual es crucial para mantener un laboratorio dinámico y actualizado. (Vaca, 2019).

También se da detalles, CX-Supervisor que puede acceder a funcionalidades de Matlab y Excel mediante scripts en VBScript, aumentando su funcionalidad a través del modelo COM (Component Object Model), que facilita la comunicación entre aplicaciones. (Pastor, 2023); en la tabla 1.4 se puede apreciar un resumen del análisis de diferentes softwares Scada.

Tabla 1.4

Descripción de softwares importantes para aplicaciones SCADA

| Característica | CX-Supervisor | All-Done | InTouch | Vijeo Look | Win CC |
|---------------------------------------|--|--|--|---------------------------------|--------------------|
| Proveedor/Fabricante | Omron/Omron (UK) | Freixas i Ros, S.L/Freixas i Ros, S.L.(E) | Logitec, S.A./Schneider Electric/Wonderware | Siemens/Schneider Electric | Siemens |
| Sistema operativo | Windows 2000/NT 4.0 (Service Pack 5) | Windows 2000/XP | Windows 2000/XP | Windows 95/98 i Windows 2000/XP | Windows NT4.0/2000 |
| Requisitos mínimos del sistema | IBM PC o compatible (200Mhz pentium o superior), Espacio libre de disco duro | Profesional Intel Pentium III/IV, AMD Athlon, Disco duro 20GB, 128MB RAM, Tarjeta SVGA 1024x768 de | Windows 2000/XP Profesional, Intel Pentium III o Celeron procesador: 500Mhz...1.8G | | |

| | | | | | |
|--------------------------------------|--|--|---|--------------------|-----------------------------|
| | 100MB, 128MB RAM, Tarjeta SVGA 1024x768 | resolución, Espacio libre de disco duro 2GB | hz, 128MB RAM, Espacio libre del disco duro 4GB, Resolución 1024 x 768 Espacio disponible: > 500 MB | | |
| Requisitos del sistema | PS4-200 OMRON: Sysmac seric C MOELLER: SIEMENS: simatic S5,S7- 200/300/400 TELEMECANIQ UE: unitelway otros... | ALLEN-BRADLEY SIEMENS MODICON OPTO 22 SQUARE D GE FANUC OMRON MODICON otros... | AEG MODICON MODICON SQUARE D TELEMECANI QUE | | |
| Drivers para PLC's | OMRON: todo los PLC's | | | | |
| Lenguajes de programación | Visual Basic/Java | Visual Basic | propio (basado en C) VBA (Visual Basic for Application) ANSI-C | C | |
| Precio (dólares) | 600 | | | | |
| Control de usuarios | si | Si | si | si | si |
| Nº variables | Ilimitado | ilimitado | 64-64k | 128, 512 0 1024 | 128, 256, 1024 64k |
| Comunicación entre módulos PC | UDP/Ethernet | Ethernet | TCP/IP NetBEUI | | |

| | | | | | |
|---|--|--|--|---|---|
| Estructura | un único PC o varios PC's en multitasca | varios PC's en multitasca | un único PC o varios PC's en multitasca | un único PC | varios PC's en multitasca |
| Gestión de alarmas | si | si | si | si | si |
| Comunicación con PLC | RS-232C UDP sobre PPI/MPI DEVICE-Net Uni-Telway o Modbus sobre RS-232 RS-422 Ethernet Profibus | RS-232C UDP sobre PPI/MPI DEVICE-Net Uni-Telway o Modbus sobre RS-232 RS-422 Ethernet Profibus TCP/IP sobre RS-232, RS-485 | RS-232C UDP sobre PPI/MPI DEVICE-Net Uni-Telway o Modbus sobre RS-232 RS-422 Ethernet Profibus | RS-232C UDP sobre PPI/MPI DEVICE-Net Uni-Telway o Modbus sobre RS-232 RS-422 Ethernet | RS-232C UDP sobre PPI/MPI DEVICE-Net Uni-Telway o Modbus sobre RS-232 RS-422 Ethernet |
| Adquisición de datos | cliente OPC OPC Servidor/Cliente | OPC Servidor/Cliente OPC Factory Server (OFS) Servidor SQL Access 2000 | Servidor/Cliente OPC | Servidor/Cliente OPC | |
| Bases de datos | SQL SERVER MS-Access MDB MDSE | MSDE Sybase SQL ODBC paraORACLE | | | |
| Intercambio de datos en el entorno de automatización | controles ACTIVE X comunicaciones DDE | | | | |

Nota: En la siguiente tabla muestra softwares importantes dentro del sistema SCADA que se implementan en la industrial y se encuentra en el mercado como CXSupervisor, All-Done, Intouch, WinCC y Vijeo Look. Tomado de (Cabús, Navarrete, & Porras, 2004), (Vaca, 2019) y (Pastor, 2023).

Análisis de resultados

Se presenta a continuación, un análisis completo de los distintos datos que se obtuvo en la sección del estado del arte, para obtener los mejores parámetros en el desarrollo del proyecto, en donde con ayuda de tablas se logrará obtener un correcto estudio comparativo de los softwares Scada que se han aplicado en proyectos industriales, programas de visión artificial que hayan obtenido algoritmos de procesamiento de imágenes y un estudio de métodos que se han utilizado para el conteo de objetos aplicando la visión artificial.

El estudio para escoger el mejor programa y método a implementarse en el proyecto, se cuantifico del 1 al 5 en donde 5 es excelente para el proyecto, 4 es muy bueno para el proyecto, 3 es bueno para el proyecto, 2 es regular, 1 que no aporta recursos para el proyecto y 0 que no es posible aplicarlo, se recalca que se toma en cuenta en cada uno de estos parámetros las principales características de estudio. (Cabús, Navarrete, & Porras, 2004)

Comparación de Software SCADA

Se procede a realizar un análisis comparativo, se detallan varios de estos sistemas que aportan a las diferentes industrias al monitoreo y control de sus procesos, se cuantifica mediante una tabla la mejor opción sobre cuál software se adapta mejor a las necesidades del proyecto.

En la tabla 1.5 se puede visualizar un análisis de softwares SCADA que se va a utilizar en el proyecto, en donde se encuentran los más importantes del mercado como son: CX-Supervisor, All-Done, In Touch, Software Python, Win CC e Indusoft Web Studio, se toman en características que aportan con los requerimientos de proyecto. (Cabús, Navarrete, & Porras, 2004)

Tabla 1.5

Softwares SCADA para utilización en el proyecto

| Característica | CX-Supervisor | All-Done | InTouch | Software Python | Win CC | Indusoft Web Studio |
|-----------------------|----------------------|-----------------|----------------|------------------------|---------------|----------------------------|
|-----------------------|----------------------|-----------------|----------------|------------------------|---------------|----------------------------|

| | | | | | | |
|---|----------------|----------------|----------------|-------------|-------------|--------------|
| Compatibilidad con SO | 3 | 3 | 3 | 5 | 2 | 5 |
| Requisitos mínimos del sistema | 2 | 2 | 3 | 5 | 1 | 4 |
| Requisitos recomendados del sistema | 3 | 3 | 4 | 5 | 2 | 5 |
| Drivers para PLC's | 5 | 3 | 3 | 4 | 3 | 5 |
| Lenguajes de programación | 4 | 3 | 2 | 5 | 2 | 5 |
| Precio | 5 | - | - | - | - | 2 |
| Control de usuarios | 5 | 5 | 5 | 5 | 5 | 5 |
| Nº variables | 5 | 5 | 3 | 5 | 4 | 5 |
| Comunicación entre módulos PC | 4 | 5 | 5 | - | - | 5 |
| Estructura | 3 | 5 | 3 | 5 | 3 | 3 |
| Gestión de alarmas | 5 | 5 | 5 | 5 | 5 | 5 |
| Comunicación con PLC | 5 | 5 | 5 | 5 | 5 | 5 |
| Calificación General | 3.8 | 3.3 | 3.7 | 4.8 | 2.7 | 4.3 |
| Recomendación para Visión Artificial | Regular | Regular | Regular | Bajo | Bajo | Bueno |

Nota: La siguiente tabla muestra un análisis de softwares que van a aportar al proyecto.

De la tabla 1.5 se obtiene las siguientes calificaciones tomando en cuenta las características principales, que se visualizan en la tabla 1.6.

Tabla 1.6

Características principales del software SCADA

| Software SCADA | Cuantificación sobre 5 | Detalle |
|-----------------------|-------------------------------|--------------------------|
| CX-Supervisor: | 3.8 | Regular para el proyecto |
| All-Done | 3.3 | Regular para el proyecto |
| InTouch | 3.7 | Bajo para el proyecto |
| Software Python | 4.8 | Bueno para el proyecto |
| Win CC | 2.7 | Bajo para el proyecto |
| Indusoft Web Studio | 4.3 | Bueno para el proyecto |

Nota: En la imagen se puede observar las calificaciones tomando en cuenta las características principales para el proyecto.

Se debe recalcar que uno de los parámetros importantes es la interacción con visión artificial, destacando que la mejor opción es Indusoft Web Studio y Software Python, ya que aparte de tener la mejor cuantificación en las características requeridas también ofrece:

- **Amplia compatibilidad con drivers PLC:** Esto permite una completa conexión con una amplia gama de PLC's, lo que facilita la integración con sistemas industriales.
- **Lenguajes de programación flexibles:** Soporta programas que son frecuentemente utilizados como: Visual Basic .NET, C#, F# y Python, ofreciendo a los autores y desarrolladores una gran variedad de herramientas para la culminación de su proyecto.
- **Requisitos de sistema moderados:** En la actualidad funcionan en sistemas operativos diversos como Windows 7, 8 o 10, haciendo a que se sea accesible para cualquier desarrollador.
- **Estructura escalable:** Otra de las características que se tomó en cuenta es que este software es utilizado en configuraciones de un solo PC o en sistemas multi-PC, adaptándose a las necesidades del proyecto.
- **Comunicación robusta:** Una de las principales características es la comunicación, en donde el software soporta diversos protocolos, incluyendo TCP/IP, UDP/IP, RS-232 y RS-485, lo que garantiza una comunicación confiable con los dispositivos periféricos.

Análisis comparativo de software de Visión Artificial

En la tabla 1.7 se detalla distintos softwares, con sus características principales para una ideal integración de visión artificial asociado con sistemas Scada. De los descritos están Cognex, Matrox, National Instruments, Sick, Keyence y Python.

Tabla 1.7

Softwares ideales para integración de visión artificial

| Característica | National | | | | | |
|--|----------|--------|-------------|------|---------|--------|
| | Cognex | Matrox | Instruments | SICK | Keyence | Python |
| Facilidad de uso | 4 | 5 | 4 | 3 | 5 | 5 |
| Flexibilidad de programación | 5 | 5 | 5 | 4 | 4 | 3 |
| Precisión del conteo | 5 | 5 | 5 | 5 | 5 | 5 |
| Velocidad de procesamiento de imágenes | 5 | 5 | 5 | 5 | 5 | 5 |
| Capacidad de integración con SCADA | 5 | 4 | 3 | 3 | 5 | 5 |
| Soporte técnico | 5 | 5 | 5 | 5 | 5 | 5 |
| Precio | 4 | 4 | 3 | 3 | 4 | 5 |
| Experiencia en aplicaciones de conteo de varillas | 5 | 4 | 3 | 3 | 5 | 5 |

Nota: En la imagen se puede observar las características principales para una ideal integración de visión artificial asociado con sistemas Scada.

Como se aprecia en la tabla 1.7 Python es el software ideal para el proyecto ya que su integración con visión artificial, la velocidad de procesamiento de imágenes, la precisión del conteo, la visualización de datos en tiempo real, el soporte técnico y estabilidad del sistema es muy bueno obteniendo una calificación de 4.97

Análisis del método para el procesamiento de imágenes.

En el apartado de teoría del arte se analiza distintos métodos de procesamiento de imágenes, para la realización del proyecto en donde su principal característica son el conteo de varillas con visión artificial.

Tabla 1.8

Métodos para el procesamiento de imágenes

| Método | Precisión | Robustez | Escalabilidad | Facilidad de implementación | Segmentación de imágenes | Calificación General |
|--|------------------|-----------------|----------------------|------------------------------------|---------------------------------|-----------------------------|
| Regresión lineal: | Media | Media | Media | Alta | No | 2.8 |
| Árboles de decisión: | Media | Media | Media | Alta | No | 2.8 |
| K-Nearest Neighbors (KNN): | Alta | Alta | Media | Media | No | 3.5 |
| Redes neuronales convolucionales (CNN): | Alta | Alta | Alta | Baja | Sí | 4.7 |
| Redes neuronales recurrentes (RNN): | Alta | Alta | Alta | Baja | Sí | 4.7 |

Nota: se aprecia los diversos métodos estudiados, con sus principales características, tomando en cuenta los requerimientos del proyecto.

Como se aprecia los resultados obtenidos en la tabla 1.8 los métodos que se escogen de acuerdo con la calificación son: Redes Neuronales y segmentación de imágenes ya que ofrecen aprender una representación jerárquica de las imágenes.

Por lo tanto, en resumen, el método de segmentación de imágenes y el software Python para realizar la visión artificial y SCADA son los instrumentos adecuados para implementar el sistema de conteo de varillas automatizado

CAPÍTULO 2

2. HIPÓTESIS

El diseño y desarrollo de un sistema automatizado de conteo continuo de varillas mediante visión artificial y sistema SCADA en la industria siderúrgica, utilizando fotos de alta calidad de paquetes de varillas tomadas a 3 metros de distancia al lote de producción mediante cámara, diferentes tipos de varillas y diámetros de varillas, será capaz de mejorar significativamente la precisión del conteo de la cantidad de varillas por lote, reducir los errores humanos y disminuir los costos operativos en comparación con los métodos tradicionales de conteo manual. La optimización de los recursos y la minimización de las pérdidas por errores de conteo reducirán la necesidad de supervisión e intervención manual, lo que se traducirá en ahorros de mano de obra y una operación más eficiente.

2.1. Variables de investigación

2.1.1. Variables Independientes.

- Integración de un sistema SCADA para monitorear y gestionar el sistema automatizado.
- Utilizando algoritmos de segmentación y procesamiento de imágenes.
- Captura de imágenes de grupos de varillas tomadas a 3 metros mediante cámara, para procesamiento de imágenes.
- Utilización de diferentes tipos de Varillas para el conteo automático.
- Variación en los diferentes diámetros de las varillas contadas.

2.1.2. Variables Dependientes.

- Número de varillas contadas en cada lote, medida de precisión del sistema.
- Eficiencia y efectividad del sistema SCADA en el monitoreo y control de los lotes de varillas.

2.1.3. Parámetros de diseño

El sistema de automatizado de conteo de varillas mediante visión artificial debe distribuirse en 4 etapas, las mismas que se describen como: Etapa de

identificación y adquisición de datos en el en el área de producción varillas, etapa de implementación de visión artificial para el conteo de varillas, etapa de puesta en marcha del sistema SCADA y etapa de análisis de resultados tomados del diseño del sistema automático de Varillas. Para que estas distintas etapas cumplan con un funcionamiento eficaz se deben establecer diversos parámetros.

2.1.3.1. Etapa de identificación y adquisición de datos en el tren de carga de varillas.

Esta primera etapa se recolecta todos los datos requeridos para realizar el diseño del conteo de varillas automatizado, aquí se recepta información como el número de lotes, cantidad de varillas producidas en cada lote, tipo o grado de varilla, ubicación de los lotes de varilla, tiempos de conteo del operario de las varillas y diámetro de varillas. En el diseño del sistema se tomarán en cuenta las normas INEN 2167 y ASTM A706, paquetes de 70 a 96 varillas aproximadamente y con diámetros de 8 a 32 mm.

Etapa de implementación de visión artificial para el conteo de varillas.

En la etapa de implementación de visión artificial en el sistema se realizará en el software Python, se considera las imágenes subidas a la base de datos, estas deben ser de alta calidad tomadas a una distancia de 3 metros de los paquetes de varillas, también se corrigen errores de iluminación, reducción de ruido y mejora de contraste, se requiere el uso de bibliotecas de procesamiento de imágenes con OpenCV, algoritmos de segmentación como: Umbralización, Segmentación por regiones, segmentación por contornos, filtros morfológicos, Algoritmo de clustering y máscara de convolución.

Por último, se toma en cuenta que las imágenes deben ser capturadas mínimo con una cámara de 5 megapíxeles para que se pueda realizar todo el procesamiento de imágenes.

2.1.3.2. Etapa de puesta en marcha del sistema SCADA.

El sistema SCADA del conteo de varillas automatizado, deberá cumplir diversos componentes para supervisar, controlar y gestionar de manera eficaz el proceso como; interfaz de usuario amigable, conexión con el sistema de visión artificial tomando en cuenta que se realizará en el mismo software Python; Recopilación

de datos en tiempo real en donde va estar el proceso de conteo, incluyendo el número de varillas contadas, el estado del sistema de visión artificial, y todos los datos relevantes para el proceso de producción, también se toman en cuenta parámetros de control de dispositivos y actuadores, monitoreo de condiciones y alarmas, historial de registro de datos y seguridad y acceso.

El sistema SCADA cumplirá con capacidades de monitoreo y gestión de datos integrales que garantiza el funcionamiento eficiente y confiable del proceso de diseño del conteo de varillas automatizado.

2.1.3.3. Etapa de análisis de resultados tomados del diseño del sistema automático de Varillas.

En la etapa de análisis de resultados se evaluarán varios parámetros clave que garantizan el rendimiento óptimo del sistema. Los parámetros que se analizarán son los siguientes:

Exactitud del conteo: Este parámetro ayuda a evaluar la capacidad del sistema, para proporcionar un conteo correcto y cercano al valor real o esperado del número de varillas. En cuenta el número real de varillas vs el número detectado por el sistema.

Velocidad del conteo: Aquí se evalúa cuanta velocidad le toma al sistema para que pueda contar las varillas, es importante ya que este parámetro garantiza la eficiencia mejorando el proceso de producción.

Eficiencia de detección: En este parámetro se analizará la capacidad que tiene el sistema para la detección correcta de todas las varillas presentes en la línea de producción, incluso cuando existen condiciones variables como el cambio en la posición de las varillas o en la iluminación.

Tasa de falsos positivos/negativos: Evaluar este parámetro es importante ya que se detecta la incidencia de errores, los cuales cuando el sistema identifica incorrectamente una varilla (falso positivo) o cuando no detecta una varilla (falso negativo). La tasa de falsos positivos/ negativos es crucial para la precisión del sistema automatizado de conteo de varillas.

Facilidad de integración con el sistema SCADA: En este parámetro se evalúa la capacidad del sistema al integrarse la parte de visión artificial con el sistema

SCADA, se tomarán en cuenta la transmisión de datos de forma eficiente y la capacidad de monitorear y controlar el proceso que tiene el sistema al momento de contar las varillas desde el sistema SCADA.

2.1.4. Alternativa de solución

Después de analizar el estado del arte y los parámetros requeridos del sistema como técnicas de visión artificial, lectura de base de datos, sistemas SCADA en la industria de siderurgia y diversos aspectos de importancia, se plantea una alternativa de solución tal como se aprecia en la figura 2.1, la misma que indica de manera resumida por medio de un diagrama, el cómo se desarrollará el proyecto. Como se puede visualizar en la figura 2.1, el sistema de conteo de varillas automatizado se dividirá en diferentes etapas. En la parte superior izquierda está la etapa de identificación y adquisición de datos, que serán tomados por el operario de la producción de varillas que la empresa tiene a través del tiempo. También se aprecia que una cámara captura la imagen del paquete de varillas en la línea de producción, esta imagen servirá para enlazar esta primera etapa con la segunda parte del sistema como se observa en la parte superior derecha, donde se encuentra el sistema de visión artificial que realiza la segmentación de los distintos paquetes con varillas que se encuentran en la imagen capturada. La programación esta implementada en el software Python con librerías de OpenCV.

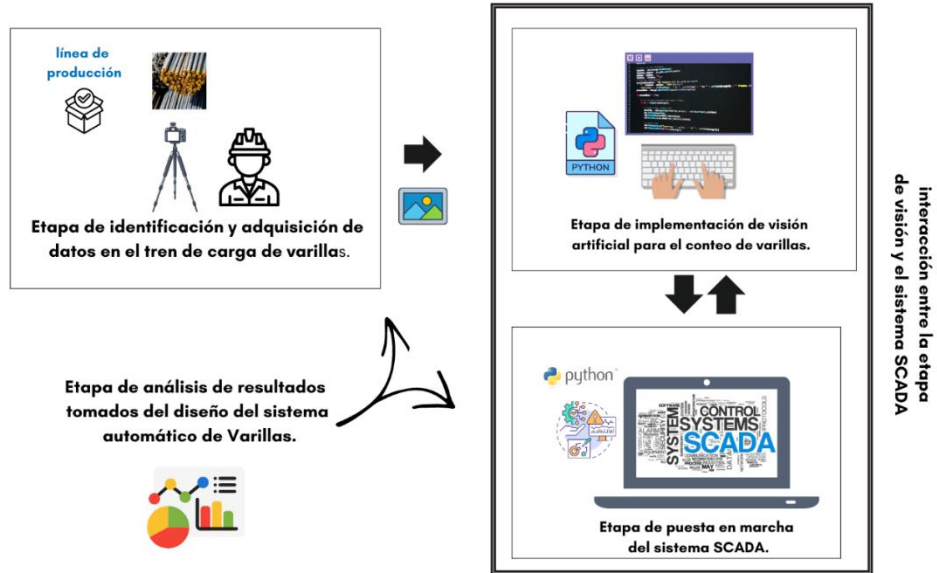
Luego de programar y obtener la base de datos del sistema, se enlaza con el sistema SCADA que funciona en tiempo real, monitoreo de condiciones y alarmas, historial de registro de datos y seguridad y acceso.

Se debe tomar en cuenta que es importante la interacción en conjunto entre la etapa de visión con la etapa del sistema SCADA.

Por último, se visualiza el análisis de resultados del diseño del sistema automatizado de conteo de varillas, verificando la eficiencia y precisión del mismo.

Figura 2.1

Análisis de resultados del diseño del sistema



Nota: Esquema del análisis de resultados del diseño del sistema automatizado de conteo de varillas.

CAPÍTULO 3

3. Diseño y Desarrollo

3.1. Diseño del sistema automático de conteo de varillas utilizando visión artificial y sistema SCADA.

Una vez estudiados los parámetros de diseño y planteada la alternativa de solución que se observa en la figura 2.1, se realiza el respectivo diseño del proyecto, empezando por la etapa de identificación y adquisición de datos en el tren de carga de varillas, seguido del diseño de la programación en visión artificial para el conteo de las varillas, para continuar con el sistema SCADA que visualizará todos los parámetros del conteo automático de varillas.

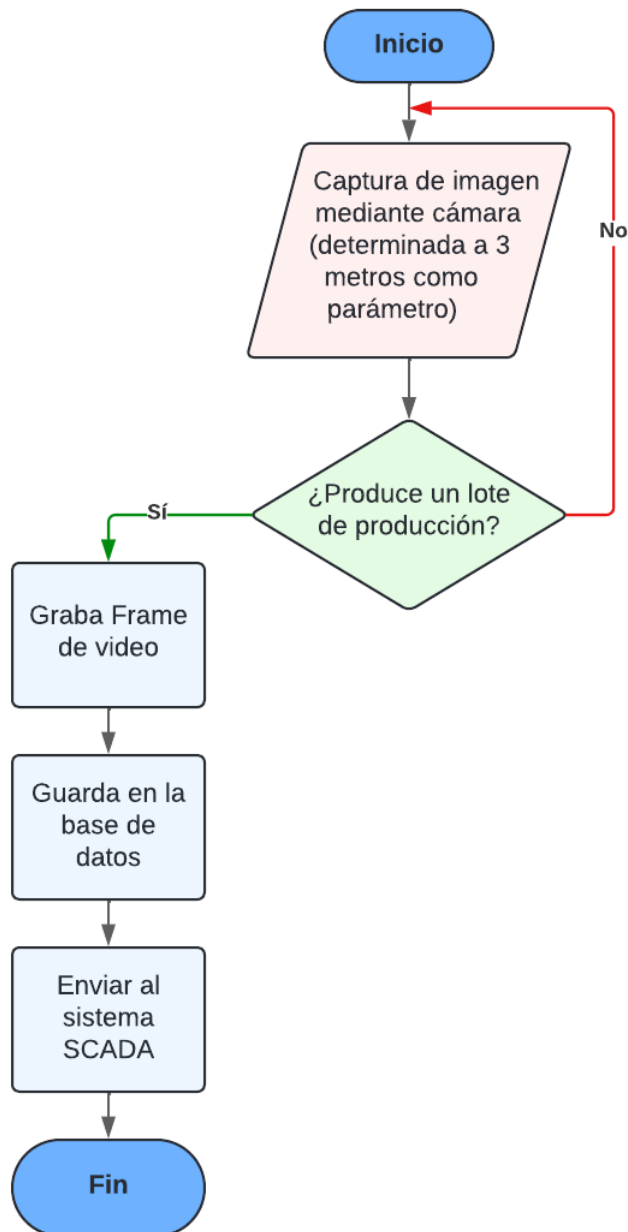
Cabe recalcar que todos estos criterios están acorde a los requerimientos planteados en el capítulo 2, teniendo en cuenta tanto la funcionalidad como la comodidad de uso y el atractivo visual del proyecto.

3.2. Diseño de la etapa de identificación y adquisición de datos en el tren de carga de varillas.

Para que el proyecto se desarrolle eficazmente, debe seguir un orden específico, en donde lo primero que se debe hacer es obtener todos los datos de cómo y cuántas varillas saldrán en cada lote en el tren de carga, tomando en cuenta las normas INEN 2167 y ASTM A706, tal como se menciona en el Ítem 2.1 inciso b del capítulo 2; en la figura 3.1 se puede observar a continuación los pasos en los que esta etapa se va a desarrollar.

Figura 3.1

Diagrama de flujo del diseño de la etapa y adquisición de datos



Nota: Se puede apreciar el proceso mediante diagrama de flujo del diseño de la etapa y adquisición de imágenes

Como se aprecia en la figura 3.1, la etapa de identificación y adquisición de datos en el tren de carga de varillas tiene como parámetros la cantidad de lotes de producción y la cámara para adquirir los frames de video; En este se procede a

determinar la posición de la cámara que será de 3 m de distancia al tren de carga, este parámetro es importante ya que se grabará obteniendo imágenes paquete por paquete en cada línea de manufactura.

Luego se plantea la interrogante si se produce un lote de producción, en donde si la respuesta es afirmativa la cámara procede a grabar, para posteriormente obtener una base de que se enviará al sistema SCADA, caso contrario el proceso regresará antes de la captura de imagen mediante cámara.

3.3. Diseño de la etapa de implementación de visión artificial para el conteo de varillas.

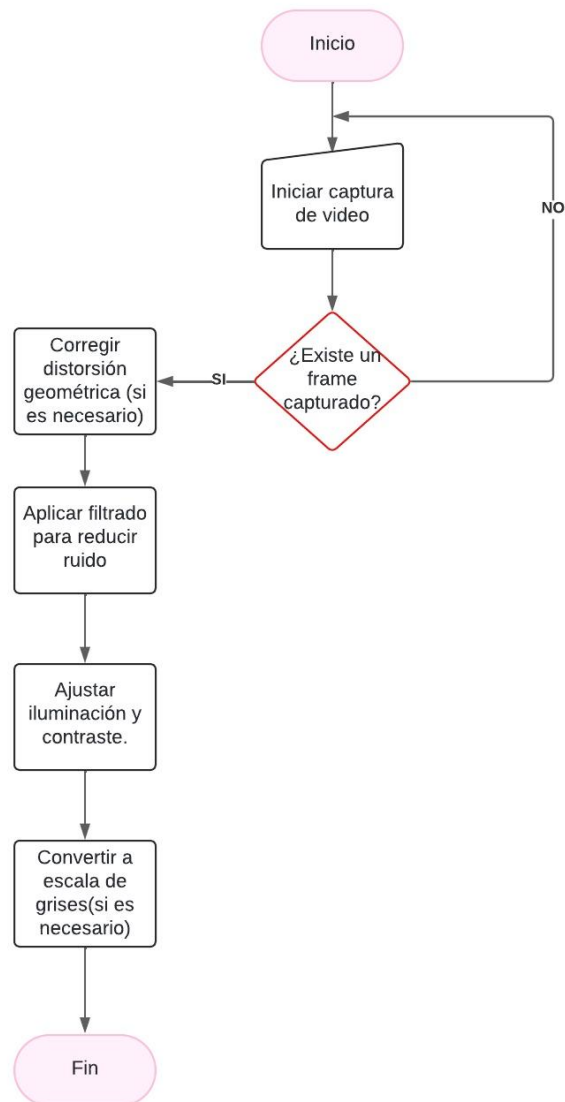
Se procede a diseñar la estructura de programación del sistema de visión artificial para el conteo de las varillas, la programación se dividirá en cuatro partes que son: Captura y preprocesamiento, segmentación de imágenes, conteo de varillas, y la muestra de resultados, cada una de estas subdivisiones se explicarán con su respectivo diagrama de flujo.

3.3.1. Captura y preprocesamiento

En el diagrama de flujo de la figura 3.2 se observa la estructura de programación, iniciando con la captura de video, realizando una interrogante si el frame es capturado, en caso de que la respuesta sea afirmativa, se procede a corregir la distorsión geométrica (si es necesario), aplicar filtrado para reducir ruido, ajustar iluminación y contraste y convertir a escala de grises (si es necesario) y terminar el proceso, caso contrario vuelve a inicializar una nueva captura de video

Figura 3.2

Diagrama de flujo de la estructura de programación para captura y procesamiento de imágenes.



Nota: En la imagen se aprecia la estructura de programación del sistema de visión artificial para el conteo de las varillas, en donde la programación se dividirá en cuatro partes

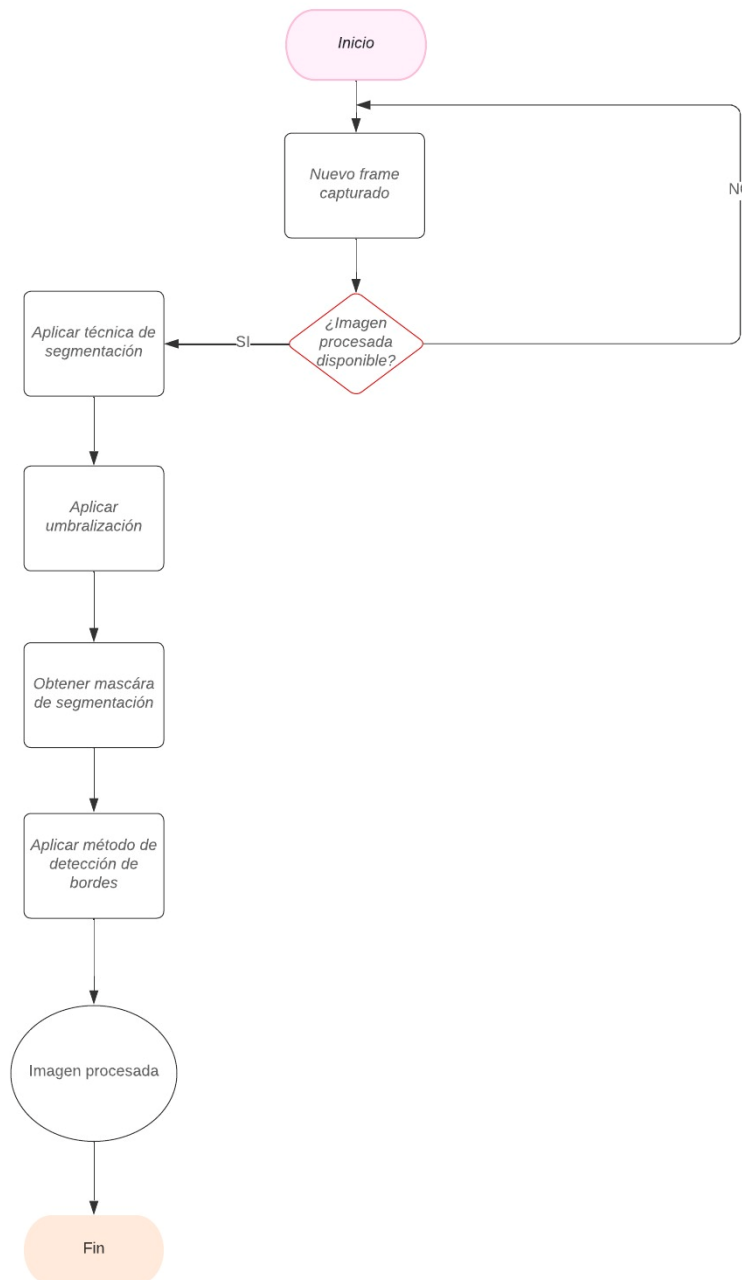
3.3.2. Segmentación de imágenes

En la figura 3.3 se visualiza la estructura que seguirá la parte de segmentación de imágenes, en donde se plantea la interrogante de si la imagen procesada está disponible, para aplicar la técnica de segmentación, que puede contar con umbralización (obtener la máscara de segmentación). También se aplica el

método de detección de bordes el que sirve para encontrar bordes de imagen y máscara de segmentación, en caso de no tener una imagen procesada disponible vuelve a inicializar un nuevo frame.

Figura 3.3

Diagrama de flujo proceso de segmentación de imágenes.



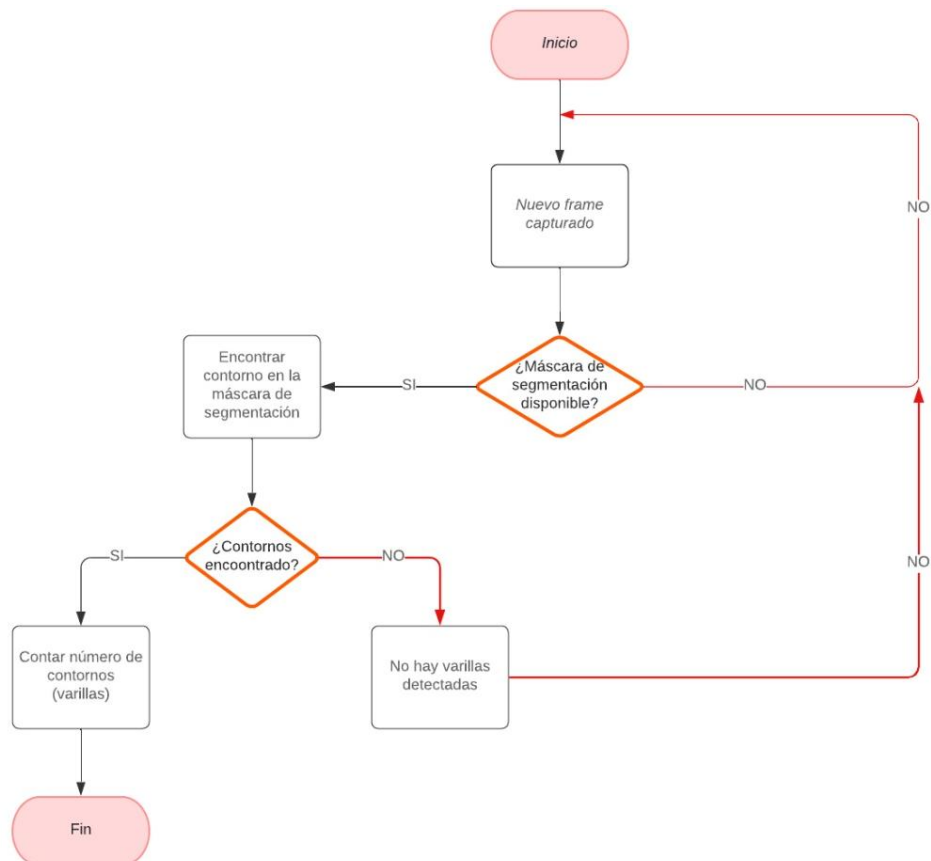
Nota: se visualiza la estructura que seguirá la parte de segmentación de imágenes, en donde se plantea la interrogante de si la imagen procesada está disponible

3.3.3. Conteo de varillas.

Para explicar la parte de la estructura de programación de conteo de varillas, se visualiza el diagrama de flujo que se encuentra en la figura 3.4, en donde inicializa un nuevo frame, para preguntar si se encuentra una máscara de segmentación disponible, para encontrar los contornos en la máscara de segmentación, luego pregunta la existencia de contornos encontrados, para contar el número de contornos (varillas), ya que si no existe estos contornos no hay varillas detectadas. Si todo esto no se encuentra el programa vuelve a preguntar si hay una máscara de segmentación disponible, si no la hay vuelve a inicializar un nuevo frame.

Figura 3.4

Diagrama de flujo de la estructura de programación de conteo de varillas.



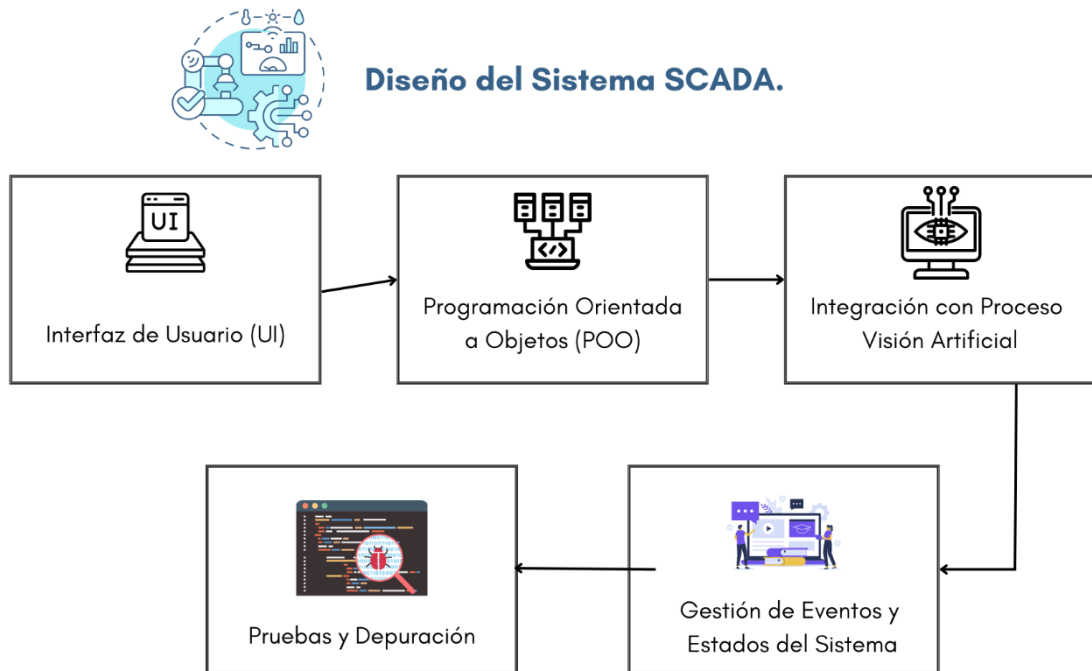
Nota: En la imagen se puede apreciar la inicialización de un nuevo frame, para preguntar si se encuentra una máscara de segmentación disponible.

3.4. Diseño del sistema SCADA

Se procede a realizar el diseño del sistema SCADA, en donde en la figura 3.5 se puede visualizar la secuencia de procesos, para que el esquema del sistema cumpla con los parámetros requeridos por el proyecto tal como se analizó en el capítulo 2, primero se esquematiza la interfaz de usuario, para luego pasar a la programación Orientada a Objetos (POO), después se realizará la integración con el proceso de visión artificial que es la etapa analizada en la sección anterior, terminando con la gestión de eventos y estado del sistema y pruebas.

Figura 3.5

Diseño del sistema SCADA



Nota: Diagrama de todos los procesos del diseño de sistema SCADA distribuido por secciones.

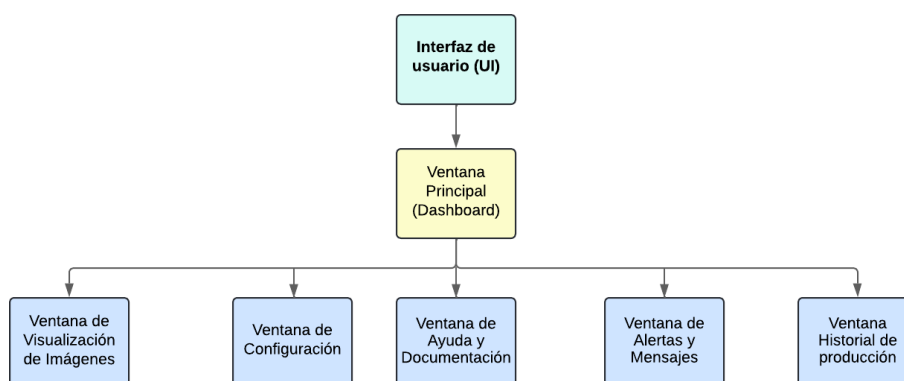
En la figura 3.5 se observa la secuencia de los procesos para el diseño del sistema SCADA en el proyecto del sistema automático del conteo de varillas.

A) Interfaz de usuario (UI)

Se procede a diseñar los procesos de la interfaz de usuario (UI) del sistema SCADA, en la figura 3.6 se observa la distribución de la interfaz; con la finalidad de cumplir los parámetros establecidos en el capítulo II, se plantean seis ventanas que tienen la funcionalidad específica y la experiencia que el operario requiere. Las ventanas están distribuidas en: ventana principal, ventana que indique el historial de cada paquete contado, ventana de visualización de imágenes, ventana de configuración, ventana de ayuda y documentación y ventana de alertas.

Figura 3.6

Distribución interfaz de usuario.



Nota: En la imagen se puede observar la distribución de la interfaz; con la finalidad de cumplir los parámetros establecidos en el capítulo II.

Las ventanas mencionadas en la figura 3.6 tienen una funcionalidad específica que serán escritas a continuación:

- **Ventana Principal (Dashboard):** La ventana principal se obtiene la visión principal del sistema de conteo automático de varillas, en esta ventana se incluye los accesos de las ventanas secundarias, el logo del sistema, gráficos de los parámetros principales del conteo de varillas y el estado del proceso de visión artificial y botones que permitan iniciar, detener o reiniciar el proceso.

- **Ventana de Visualización de Imágenes:** En esta ventana se muestra el video en tiempo real de la producción de las varillas, es decir, cuando estas lleguen al tren de carga, el botón para que el operario capture la imagen que se mostrará en otro cuadro en tiempo real, un cuadro en donde se visualiza la aplicación de la técnica de la visión artificial (segmentación), indicando el resultado del conteo, y por último una parte en donde el operario pueda visualizar el número de varillas contadas.
- **Ventana de Configuración:** En esta ventana se plantea al usuario ajustar los parámetros del sistema del conteo automático del sistema automatizado de varillas, como las configuraciones posibles a la cámara y los algoritmos de procesamiento de imágenes. También se pueden incluir controles deslizantes, casillas de verificación y menús desplegables para ajustar los parámetros según se requiera en el proceso.
- **Ventana de Ayuda y Documentación:** Proporciona información detallada sobre el funcionamiento del sistema, los procedimientos de solución de problemas y la documentación técnica. Es fácilmente accesible desde cualquier parte del sistema para ayudar al usuario en caso de dudas o problemas.
- **Ventana de Alertas y Mensajes:** La ventana de alertas y mensajes debe mostrar las notificaciones y alertas sobre el estado del sistema de conteo, como errores de detección o advertencias de funcionamiento que son anormales en el proceso. Esta ventana también proporciona opciones de reconocimiento, en donde el operario pueda ignorar o resolver según los requerimientos del proceso.

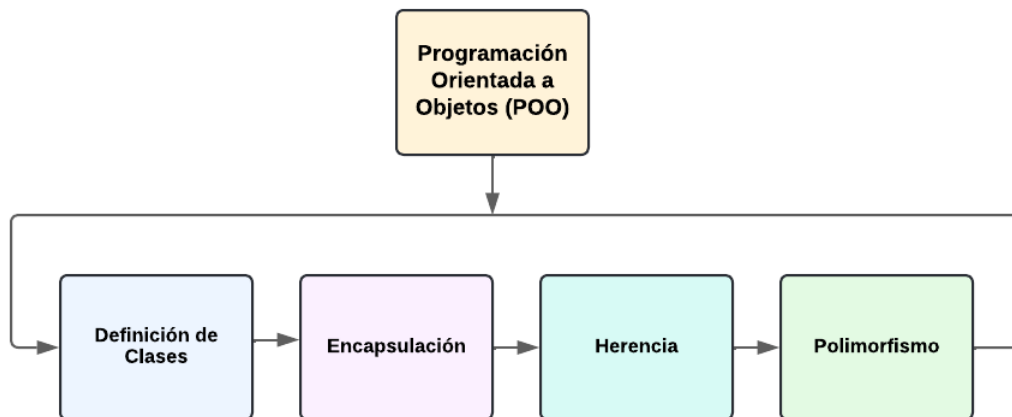
B) Programación Orientada a Objetos (POO)

El diseño de la programación orientada a objetos tiene un factor fundamental para cumplir los parámetros planteados en el capítulo II, ya que permite programar el Sistema SCADA directamente desde el software en donde se desarrolla la visión artificial del proyecto. En la figura 3.7 se puede apreciar la estructura que se va a

seguir en el diseño de esta programación, donde se tiene la definición de clases, la encapsulación, herencia y el polimorfismo.

Figura 3.7

Programación orientada a objetos.



Nota: En la imagen se puede apreciar la estructura que se va a seguir en el diseño de esta programación, donde se tiene la definición de clases, la encapsulación, herencia y el polimorfismo.

Una vez que se visualiza el esquema de la figura 3.7, se define cada uno de estos conceptos que posteriormente se implementarán en el sistema automatizado de conteo de varillas:

- **Definición de Clases:** en esta parte se crean diferentes clases en el software de programación, con la finalidad de representar elementos del sistema, como el proceso de visión artificial, el sistema completo del SCADA y la base de datos de los lotes de producción de las varillas.
- **Encapsulación:** En el sistema automatizado de varillas, se utiliza para exponer al usuario una interfaz pública consistente (dependiendo de los permisos que cada uno tendrá), ocultando los detalles internos de cada objeto, la característica principal de la encapsulación para el presente proyecto es modularizar el código logrando mantener la coherencia y la integridad del sistema.

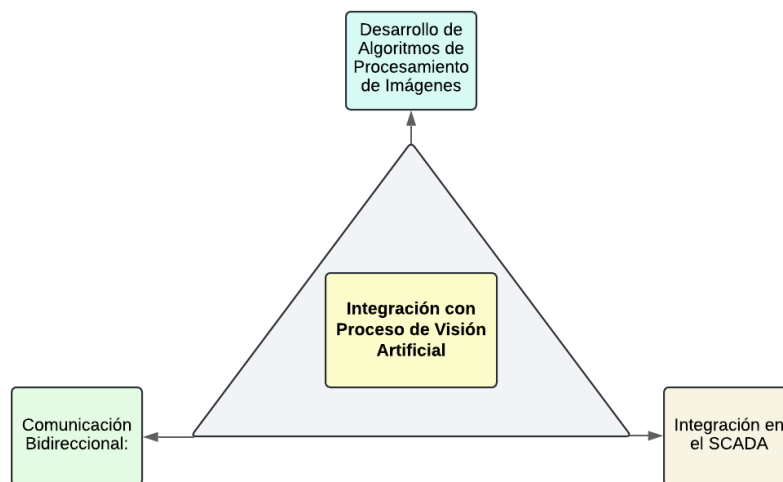
- **Herencia y Polimorfismo:** En el proyecto se implementan la herencia y polimorfismo para modelar relaciones y comportamientos comunes que los diferentes elementos del sistema van a tener. Con esto se obtendrá una clase base para todas las herramientas de detección de varillas en la visión artificial, con subclases específicas para diferentes tipos de herramientas.

C) Integración con Proceso de Visión Artificial

La Integración del SCADA con el proceso de visión artificial debe cumplir diversos factores fundamentales como son: el desarrollo de algoritmos de procesamiento de imágenes, integración de la programación en la interfaz y comunicación Bidireccional. Estos parámetros se pueden apreciar en la figura 3.8, visualizando como se integran entre ellos.

Figura 3.8

Integración con proceso de visión artificial.



Nota: En la imagen se puede apreciar de forma esquemática, el desarrollo de algoritmos de procesamiento de imágenes, integración de la programación en la interfaz y comunicación Bidireccional.

En el desarrollo de Algoritmos de Procesamiento de Imágenes, se utilizará en el proyecto todas las bibliotecas requeridas para que el programa funcione de manera eficaz, también con esto se desarrollan todos los algoritmos de procesamiento de imágenes que detecten y cuenten las varillas en las imágenes capturadas por la cámara.

Para definir completamente la integración en el SCADA, los algoritmos de visión artificial se integran en las clases relevantes del sistema SCADA utilizando la POO tal como se indicó en el Ítem 3.1.1. Permitiendo que el SCADA obtenga el acceso a la funcionalidad de procesamiento de imágenes (varillas en el tren de producción) y reciba datos en tiempo real del proceso de visión artificial.

Y por último la comunicación Bidireccional es establecer datos en tiempo real entre el SCADA y el proceso de visión artificial. Permitiendo que el SCADA envíe comandos al proceso de visión artificial y reciba resultados de detección y conteo de varillas.

D) Gestión de Eventos y Estados del Sistema

En gestión de eventos y estado del sistema en el proyecto se divide en dos parámetros importantes que se describen a continuación:

- **Patrón Observador:** se implementa este parámetro para que el usuario pueda manejar eventos y notificaciones, entre las distintas partes del sistema, ayudando a que exista notificaciones en el SCADA cuando se detecte alguna novedad con respecto al procesamiento de imágenes.
- **Representación de eventos y estados:** con esto el usuario tiene la capacidad para representar los eventos del sistema con ayuda de clases y objetos, en este caso la detección de varillas, gestión de estados del sistema y el estado de ejecución del proceso de visión artificial.

E) Pruebas y Depuración

En esta sección se implementa la depuración de cualquier error o problema que surja durante el desarrollo e implementación del sistema. Esto implica identificar, diagnosticar y corregir errores en la estructura de programación, lógica o diseño en el sistema.

3.5. Selección de componentes

En este proyecto se procede a realizar la selección de componentes o programas el sistema automatizado de conteo de varillas, de acuerdo con los parámetros y alternativa de solución planteadas en el capítulo II, los principales factores a analizar es el proceso de visión artificial y el SCADA. En ambos campos se toman en cuanto características principales tal como se analizó en el capítulo I en la sección de estado de arte.

3.5.1. Selección del software para visión artificial

Se procede a realizar la tabla 3.1 en donde se aprecia los programas más importantes en el mercado con sus principales características, las mismas que se toman en cuenta para que el proyecto se desarrolle de manera eficaz.

Tabla 3.1

Softwares para visión artificial

| Componente | Descripción | Facilidad | | | |
|-------------------|--|-----------|--------------|---------------|--------|
| | | de Uso | Versatilidad | Documentación | Precio |
| OpenCV con Python | Es una biblioteca de visión por computadora en código abierto, ampliamente utilizada para aplicaciones de visión artificial. | Alta | Alta | Muy extensa | Gratis |
| Tensor Flow | Plataforma de código abierto para machine learning desarrollada por Google. | Media | Muy alta | Muy extensa | Gratis |
| PyTorch | Biblioteca de aprendizaje profundo de código abierto desarrollada por Facebook. | Media | Muy alta | Muy extensa | Gratis |

| | | | | | |
|--------------|---|-------|-------|----------|---------|
| scikit-image | Biblioteca de procesamiento de imágenes en Python. | Alta | Media | Extensa | Gratis |
| Dlib | Biblioteca de aprendizaje de máquina y procesamiento de imágenes. | Media | Alta | Extensa | Gratis |
| SimpleCV | Marco de visión por computadora diseñado para simplificar la programación de visión. | Alta | Media | Limitada | Gratis |
| MATLAB | Entorno de programación y plataforma de computación numérica. | Media | Alta | Extensa | De pago |
| Caffe | Marco de aprendizaje profundo de código abierto desarrollado por Berkley AI Research. | Media | Alta | Extensa | Gratis |

Nota: Se aprecia los programas más importantes en el mercado con sus principales características.

Tras haber realizado un análisis con todos los programas y las diferentes características en la tabla 3.1, se escoge el software Python con Open CV para realizar la programación de la visión artificial del sistema de conteo automático de varillas.

3.5.2. Selección del software para realizar el sistema SCADA.

Se plantea en la tabla 3.2 la selección del SCADA con los diferentes softwares del mercado y las características principales, para que el sistema automatizado de conteo de varillas se desarrolle sin inconvenientes y de manera óptima.

Tabla 3.2

Softwares para realizar el sistema SCADA

| Software | Características Principales | Facilidad | | Soporte | |
|-------------------|---|-----------|--------------|-----------|-------|
| SCADA | | de Uso | Flexibilidad | Comunidad | Costo |
| Python (con PyQt) | Desarrollo en Python con una biblioteca de GUI como PyQt. | 4/5 | 5/5 | 4/5 | Libre |

| | | | | | |
|----------|--|-----|-----|-----|-------|
| | Ambiente de desarrollo gráfico, ampliamente utilizado en la industria. | 3/5 | 4/5 | 3/5 | Medio |
| LabVIEW | SCADA basado en web con capacidades de visualización avanzada. | 4/5 | 5/5 | 4/5 | Alto |
| Ignition | Ofrece una amplia gama de herramientas para SCADA y HMI. | 5/5 | 5/5 | 5/5 | Alto |
| Intouch | Plataforma SCADA/HMI que proporciona funcionalidades avanzadas. | 4/5 | 5/5 | 4/5 | Medio |
| InduSoft | | | | | |

Nota: En la imagen se puede apreciar la selección del SCADA con los diferentes softwares del mercado y las características principales.

Tras realizar el análisis con ayuda de la tabla 3.2, se selecciona el software Python, tal como se planteó en el diseño de la interfaz, ya que se realizará con la programación orientada objetos, y por qué cumple con todos los parámetros requeridos para la realización del proyecto, el factor diferencial con el software Indusoft en el costo, por tal motivo se orientó por escoger el software Python.

3.6. Desarrollo del sistema automatizado de conteo de varillas.

Se procede a realizar el desarrollo del proyecto, de acuerdo con el diseño del sistema, analizado en el ítem 3.1, en donde se detalla la estructura y técnica de programación en cuanto a la visión artificial y la estructura del SCADA.

3.7. Desarrollo de la visión artificial del Sistema.

El proceso de desarrollo de la estructura de la visión artificial se realiza en el software Python utilizando librerías de open CV, la estructura de programación se describe a continuación en la siguiente función.

- **Función que permite el procesamiento de las imágenes con visión artificial.**

Esta función realiza todo el proceso del procesamiento de las imágenes en tiempo real de los diferentes paquetes de varillas, extraídas del tren de carga, en la figura 3.9 se aprecia el diagrama de flujo de la programación de la función, el código se puede observar en el Apéndice B.

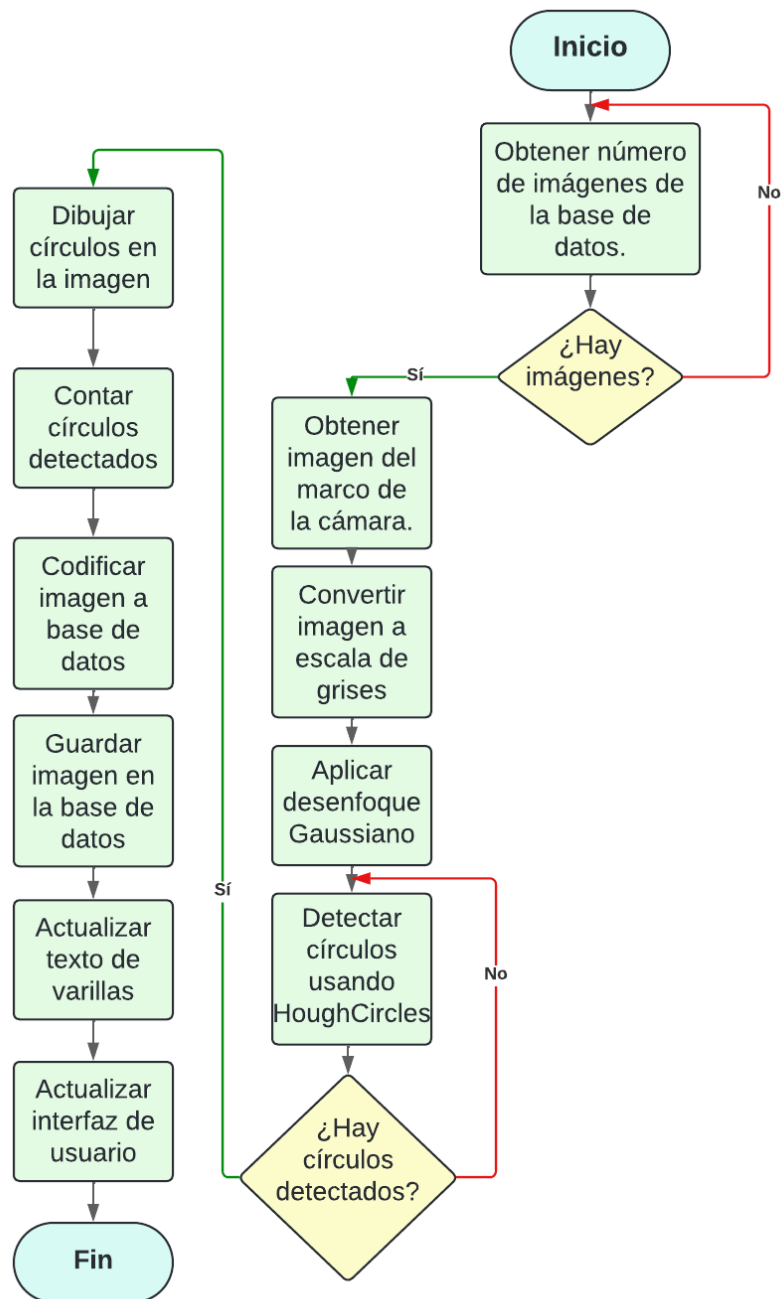
El diagrama de flujo que se encuentra en la figura 3.9 está desarrollado para procesar las imágenes y el contar el total de varillas presentes en cada paquete que se da por lote. Como se aprecia la estructura de esta programación comienza obteniendo el número de imágenes en la base de datos y asigna una nueva ID de imagen.

Luego, captura una imagen en tiempo real del marco de la cámara para realizar el procesamiento en escala de grises y desenfoque gaussiano. También se utiliza la transformada de Hough para la detección de los círculos en la imagen y, si se encuentra, el programa ejecuta el procesamiento y realiza el respectivo conteo, mostrando la imagen procesada y la cantidad de varillas detectadas en la interfaz de usuario.

Finalmente, la imagen procesada se guarda respectivamente en la base de datos con su respectivo nombre y cantidad de varillas detectadas.

Figura 3.9

procesamiento de las imágenes con visión artificial



Nota: En la imagen se puede observar el proceso desarrollado para procesar las imágenes y el contar el total de varillas presentes en cada paquete que se da por lote.

Los comandos fundamentales utilizados en la función, para que se cumplan con lo requerido por el programador, se describen a continuación en la tabla 3.3:

Tabla 3.3.*Comandos fundamentales*

| Captura y Preprocesamiento de la Imagen | |
|--|---|
| Comando | Descripción |
| camera_holder | Captura una imagen |
| cv2.cvtColor(img, cv2.COLOR_BGR2GRAY) | Convierte la imagen del paquete de varillas capturada a escala de grises, lo que es común en el procesamiento de imágenes para simplificar la detección de bordes y formas. |
| cv2.GaussianBlur(gray, (7, 7), 0) | Aplica un desenfoque gaussiano a la imagen en escala de grises para reducir el ruido y mejorar la detección de círculos. |
| Procesamiento de Resultados | |
| cv2.circle | Dibuja cada círculo en la imagen original con un color rojo |
| num_circles | Cuenta el número de círculos detectados. |
| num_varillas | Asigna el número de círculos detectados |
| Codificación y Actualización de la Interfaz | |
| cv2.imencode | Codifica la imagen con los círculos dibujados en formato JPEG y la almacena en un buffer. |
| base64.b64encode(buffer).decode() | Convierte la imagen codificada en una cadena en base64 para su uso en una interfaz web. |

Nota: En la tabla se puede observar los comandos fundamentales utilizados en la función, para que se cumplan con lo requerido por el programador

Detección de Círculos

Para la detección de círculos en el sistema, se utilizó el algoritmo de transformación de Hough, este algoritmo matemático determina la transformación de una línea en un espacio de la imagen a un punto del espacio con parámetros. Para el presente proyecto se utiliza el algoritmo de Hough para círculos, el cual se describe en una forma paramétrica con la siguiente ecuación

$$(x - a)^2 + (y - b)^2 = r^2 \quad (3,1)$$

En donde a y b son coordenadas del centro del círculo y r es el radio.

Hay que tomar en cuenta que, para cada píxel de contorno, existente en la posición (x_0, y_0) se encuentra una familia de circunferencias, las mismas que pasan por este punto y están definidas como:

$$C_x = x_0 + \cos\theta * r \quad (3,2)$$

$$C_y = y_0 + \sen\theta * r \quad (3,3)$$

En la programación se implementa el comando Cv2.HoughCircles que detecta círculos en la imagen desenfocada.

En la figura 3.10 en la parte derecha se aprecia la imagen antes de ser procesada, y en la parte izquierda se encuentra la imagen después del procesamiento, la misma que se encuentra en tiempo real del paquete de varillas, en la parte inferior de la imagen se observa la cantidad de varillas contadas.

Figura 3.10

Imagen del programa con una foto del antes y después del proceso.



Nota: En la figura se puede observar que la parte derecha la imagen antes de ser procesada, y en la parte izquierda se encuentra la imagen después del procesamiento.

3.8. Desarrollo del Sistema SCADA

El desarrollo del SCADA para el sistema automático de varillas, se lo ha segmentado en diferentes funciones y clases, que cumplen con todos los requerimientos del proyecto plantados en el capítulo II, en la figura 3,11 se visualiza la interfaz con todos los parámetros mencionadas anteriormente.

Figura 3.11

Página del inicio del programa.



Nota: En la imagen se visualiza la interfaz con todos los parámetros requeridos para el desarrollo correcto del proyecto.

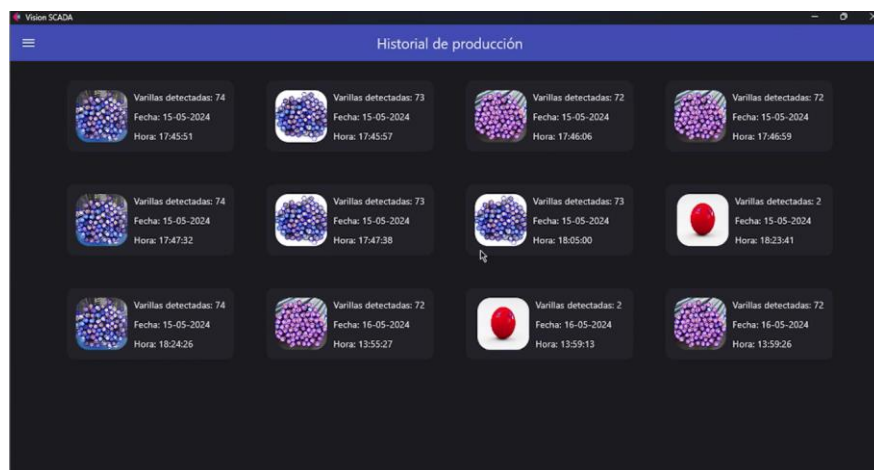
Una vez desarrollada la ventana principal, se procede desarrollar las ventanas secundarias como se indica a continuación:

- **Ventana de Historial de producción:**

En la figura 3.12 se tiene la ventana del historial de producción, tal como se planteó el Ítem 3.1.3, en donde el usuario puede visualizar la imagen del paquete producido, con el número de varillas contadas, la fecha y hora.

Figura 3.12

Pantalla del Historial



Nota: En la figura se puede apreciar la ventana del historial de producción

- **Ventana de Visualización de Imágenes:**

En la figura 3.13 se encuentra la ventana de visualización de imágenes, en donde el usuario tiene todas las imágenes captadas en tiempo real de cada paquete en los distintos lotes de producción.

Figura 3.13

Página de visualización de imágenes



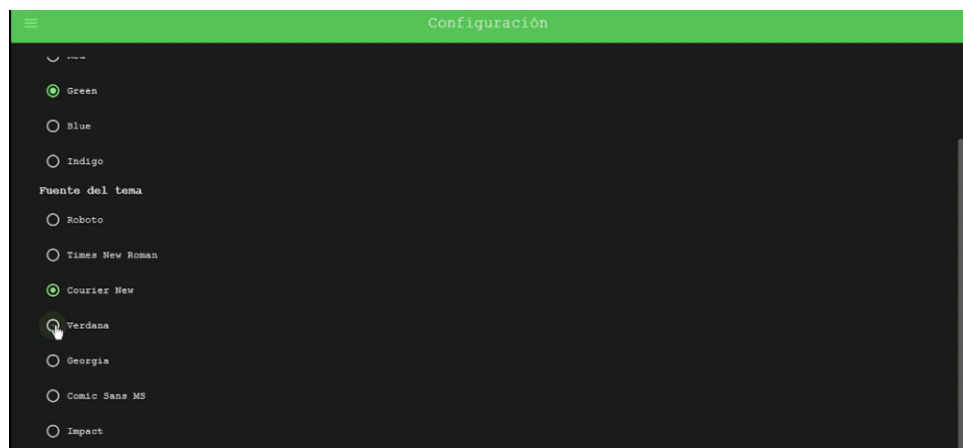
Nota: En la figura se encuentra ventana de visualización de imágenes, en donde el usuario tiene todas las imágenes captadas en tiempo real de cada paquete en los distintos lotes de producción

- **Ventana de Configuración:**

En la ventana de configuración el operario puede variar los parámetros del sistema SCADA como tipo de letra, ambiente de la interfaz, tamaño etc. Tal como se visualiza en la figura 3.14.

Figura 3.14

Página de configuración



Nota: En la imagen se puede apreciar como el usuario parámetros del sistema SCADA como tipo de letra, ambiente de la interfaz, tamaño etc.

- **Ventana de Ayuda y Documentación:**

El sistema Scada de conteo de varillas automático, tiene la ventana de ayuda y documentación, en donde se encuentra toda la información del programa, esto se observa en la figura 3.15.

Figura 3.15

Página de ayuda



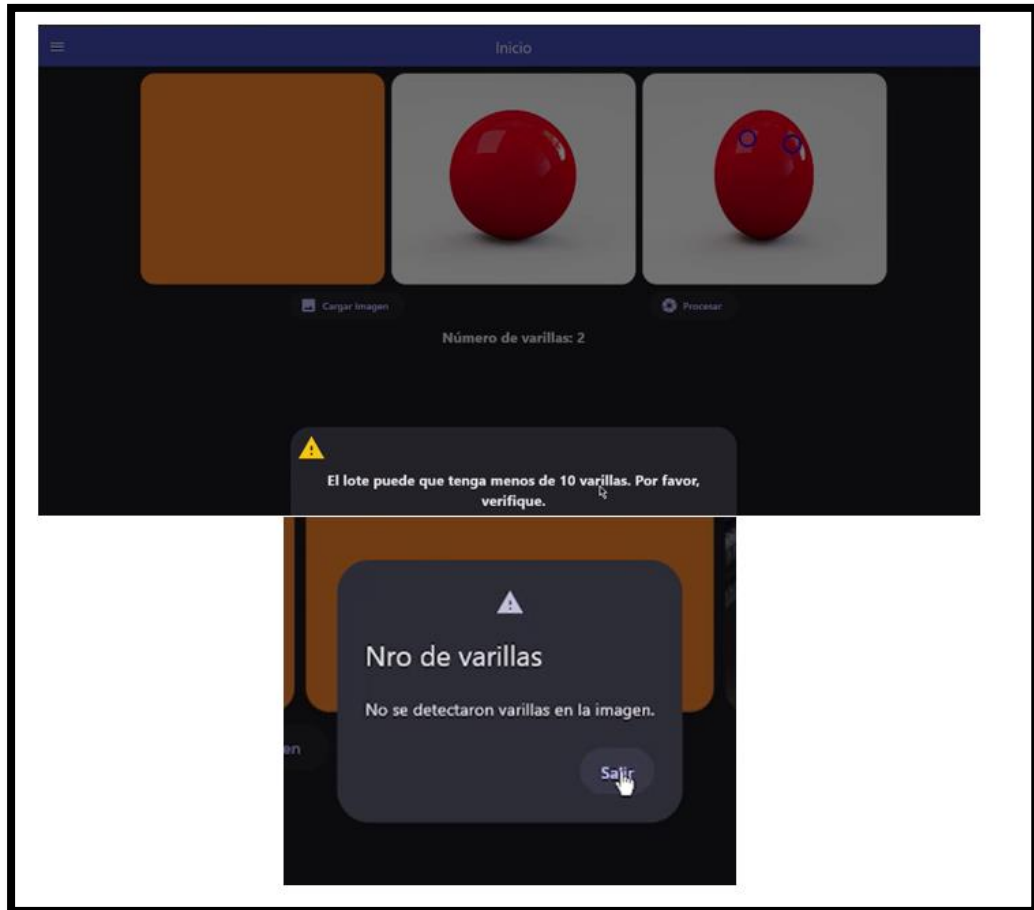
Nota: En la imagen se puede apreciar la ventana de ayuda y documentación, en donde se encuentra toda la información del programa

- **Ventana de Alertas y Mensajes:**

En esta ventana se indica diferentes alertas requeridas en el sistema, como la advertencia si existe o no imágenes a procesar, si el paquete de varillas se encuentra completo o si el sistema tiene algún inconveniente, en la figura 3.16 se puede observar algunas de estas alarmas.

Figura 3.16

Ventanas de alertas



Nota: En esta ventana se indica diferentes alertas requeridas en el sistema, como la advertencia si existe o no imágenes a procesar.

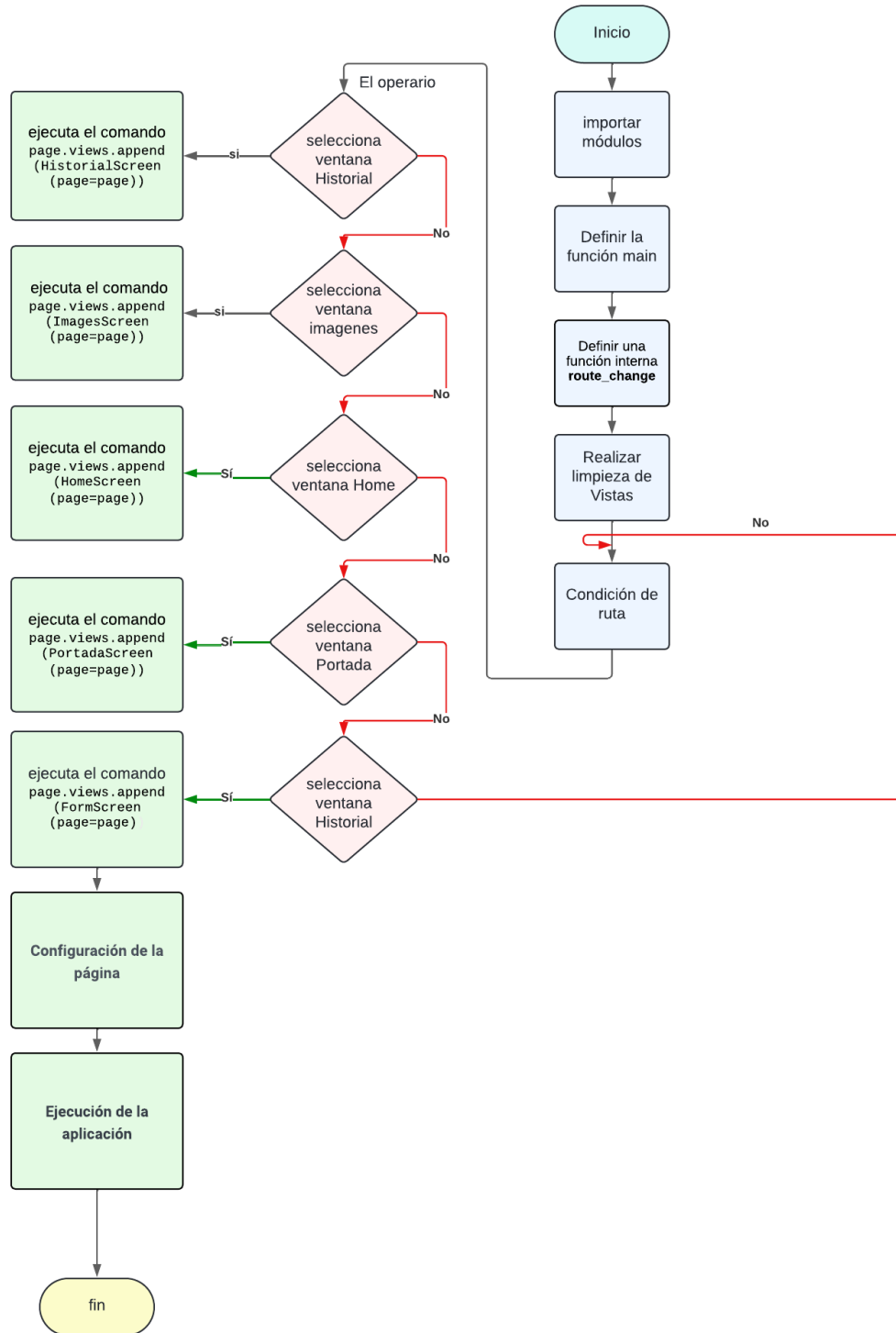
A continuación, mediante diagramas de flujo se procede a explicar la estructura de programación del desarrollo del sistema SCADA, el mismo que esta dividido en funciones y Clases principales.

- **Función principal para construir el programa y navegar sobre las diferentes pantallas.**

En este diagrama de flujo que se encuentra en figura 3.17, explica la estructura de programación del proceso para que el usuario puede navegar sobre las diferentes pantallas.

Figura 3.17

Función principal para construir el programa.



Nota: En la imagen se puede apreciar la estructura de programación del proceso para que el usuario puede navegar sobre las diferentes pantallas.

El diagrama de flujo de la figura 3.17 muestra el proceso de la programación del Apéndice A, en donde importar imágenes en una base de datos. El usuario selecciona la ventana y las imágenes que desea importar, luego se ejecutan las funciones correspondientes para limpiar vistas, importar las imágenes en la base de datos, mostrar la imagen seleccionada o redirigir al usuario a la ventana anterior.

En la tabla 3.4 se puede visualizar los comandos y parámetros fundamentales utilizados en la estructura de programación para desarrollar la navegación entre diferentes pantallas.

Tabla 3.4

Tabla de comandos y parámetros

| Parámetro | Descripción |
|---|---|
| <code>import flet as ft</code> | Importa la biblioteca Flet, que se utiliza para crear aplicaciones web y móviles en Python. |
| <code>from screens.form_screen import FormScreen</code> | Importa la clase FormScreen desde el módulo screens.form_screen. Esto se repite para otras pantallas. |
| <code>def main(page: ft.Page):</code> | Define la función principal main que toma un objeto page de tipo ft.Page, representando la página de la aplicación. |
| <code>def route_change(route):</code> | Define la función route_change que maneja los cambios de ruta en la aplicación. |
| <code>page.views.clear()</code> | Limpia las vistas actuales de la página. |
| <code>page.views.append(...)</code> | Añade una nueva vista (pantalla) a la página. La vista añadida depende de la ruta actual. |
| <code>page.update()</code> | Actualiza la página para reflejar los cambios realizados (como la adición de nuevas vistas). |
| <code>page.title = "Vision SCADA"</code> | Establece el título de la página. |
| <code>page.theme_mode = ft.ThemeMode.LIGHT</code> | Configura el modo de tema de la página en modo claro. |
| <code>page.window_width = 450</code> | Establece el ancho de la ventana de la página. |
| <code>page.window_height = 600</code> | Establece la altura de la ventana de la página. |
| <code>page.on_route_change = route_change</code> | Asigna la función route_change como el manejador de eventos para los cambios de ruta. |

page.go(page.route)

Navega a la ruta actual, lo que desencadenará la función route_change para mostrar la vista inicial.

Nota: En la tabla se aprecia los comandos y parámetros fundamentales utilizados en la estructura de programación para desarrollar la navegación entre diferentes pantallas

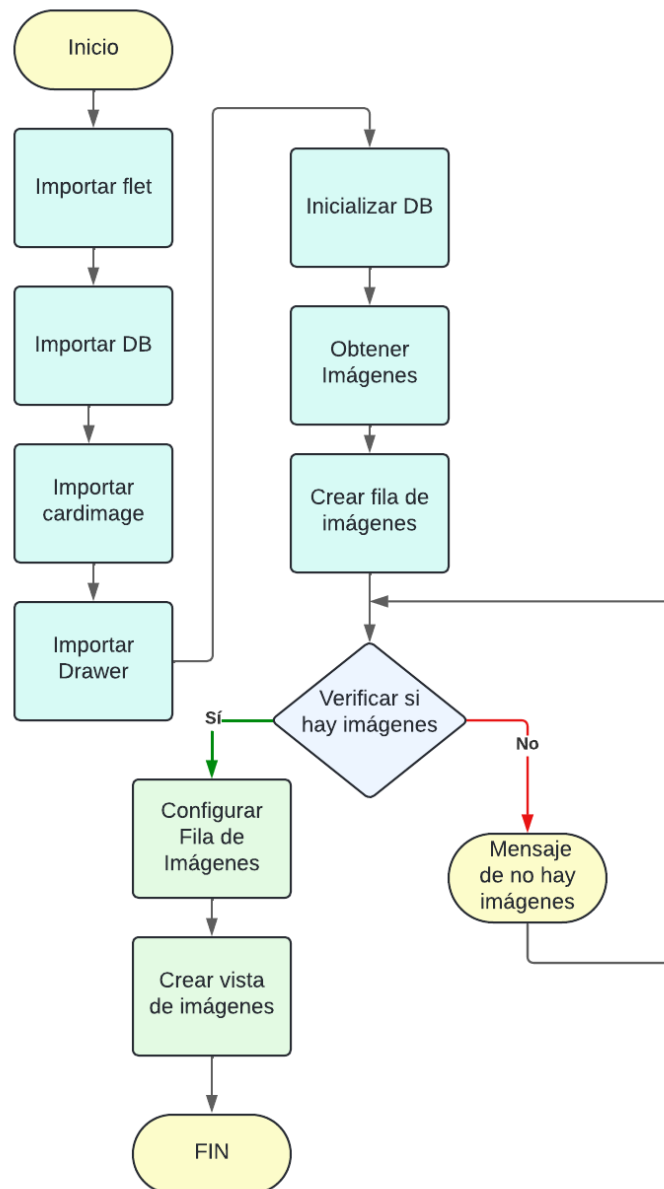
- **Función del proceso de visualización de imágenes.**

En la figura 3.18, se observa la estructura de programación del proceso de visualización de imágenes, que el operario tiene a disposición en tiempo real.

La programación se la encuentra en el Apéndice I de importar imágenes en una base de datos. El usuario importa un archivo de imágenes y una base de datos, luego se extraen las imágenes y se importan en la base de datos. Si no hay imágenes en la base de datos, se muestra un mensaje al usuario. Finalmente, se crea una vista de las imágenes en la base de datos.

Figura 3.18

Función del proceso de visualización de imágenes



Nota: En el diagrama de flujo se observa la estructura de programación del proceso de visualización de imágenes.

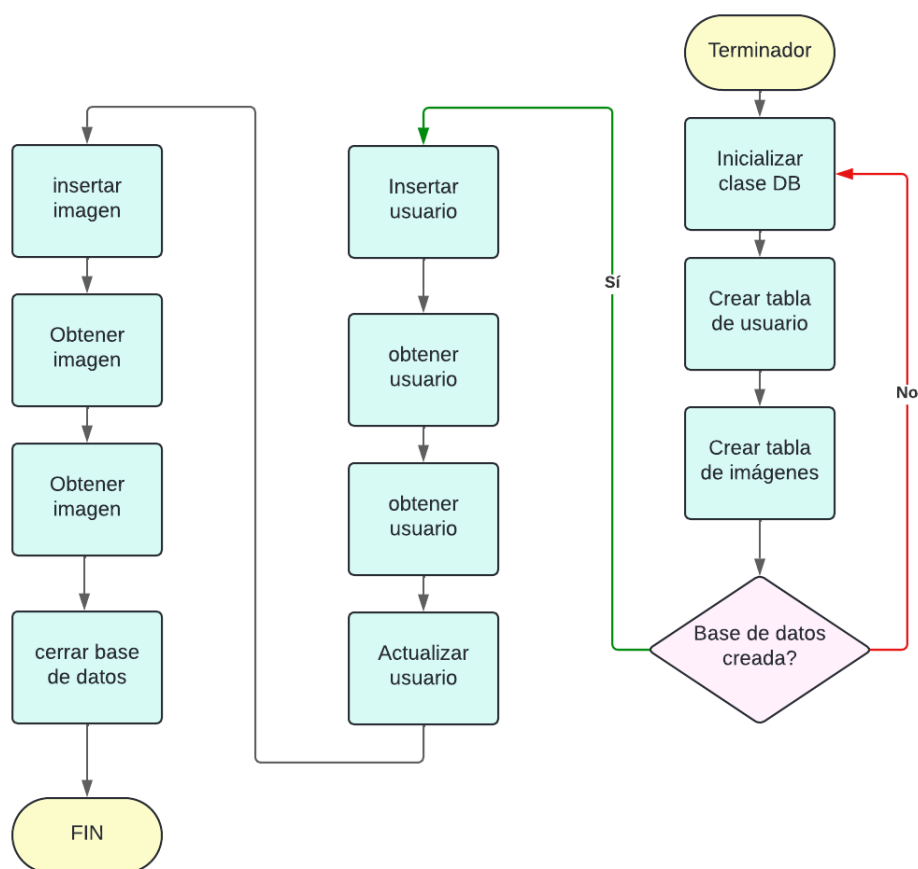
- **Clase DB para administrar la Base de datos.**

Otros de los parámetros importantes es el cómo el sistema interactúa con la base de datos, en la figura 3.19 se puede observar la estructura de programación de estos parámetros que se encuentra en el Apéndice C.

En el diagrama de flujo se explica de manera gráfica, que el usuario define el nombre de la base de datos y las tablas, para luego crear estos parámetros. A continuación, se insertan imágenes y usuarios en la base de datos. Si no hay imágenes o usuarios en la base de datos, se muestra un mensaje al usuario. Finalmente, se cierra la base de datos.

Figura 3.19

Diagrama de flujo clase DB para administrar la Base de datos.



Nota: En el diagrama de flujo se observa la estructura de programación de la clase DB para administrar la base de datos.

En la tabla 3.5 se define los parámetros y comandos principales, con su respectiva descripción. que se utiliza para administrar de forma correcta la base de datos del sistema

Tabla 3.5*Tabla de comandos y parámetros*

| Comando/Concepto | Descripción |
|-------------------------|--|
| sqlite3.connect() | Establece la conexión con la base de datos SQLite. |
| cursor.execute() | Ejecuta un comando SQL. |
| cursor.fetchall() | Obtiene todos los resultados de una consulta SQL. |
| cursor.fetchone() | Obtiene el primer resultado de una consulta SQL. |
| conn.commit() | Confirma (commit) los cambios en la base de datos. |
| conn.close() | Cierra la conexión con la base de datos. |

Nota: En la tabla se encuentra definidos los parámetros y comandos principales que se utiliza para la base d datos.

CAPÍTULO 4

4. Pruebas y Resultados.

En el presente capítulo se procede a realizar las diferentes pruebas del sistema automático de conteo de varillas, tanto del funcionamiento de la parte de visión artificial como del sistema SCADA; con esto se logra obtener los resultados requeridos para el adecuado análisis del proyecto.

4.1. Resultados de ejecución del sistema Scada.

A. Pantalla de seguridad.

Primero el operario introduce su clave de usuario en donde se puede visualizar en la figura 4.1, que tiene parámetros de inicio de sesión, para introducir usuario y contraseña.

Figura 4.1

Imagen de la Pantalla de seguridad



Nota: En la siguiente figura se indica la la pantalla de seguridad del programa.

En caso de que esta sea incorrecta inmediatamente saldrá un mensaje con contraseña incorrecta. Tal como se puede visualizar en la figura 4.2.

Figura 4.2

Imagen de la Pantalla de seguridad



Nota: En la siguiente figura se indica la pantalla de seguridad del programa con el usuario y la contraseña incorrectos.

B. Pantalla de inicio.

Una vez introducido el usuario y la contraseña, el operador visualiza la pantalla de inicio en donde encuentra los parámetros mencionados en el ítem 1.8 del presente documento, como se puede apreciar en la figura 4.3.

Figura 4.3

Imagen de la Pantalla de inicio claro



Nota: En la siguiente figura se indica la pantalla de inicio en color claro.

Para realizar el conteo automatizado, el operador selecciona el botón de Inicio y Procesamiento tal como se aprecia en la figura 4.4.

Figura 4.4

Imagen de la Pantalla de seguridad oscuro.



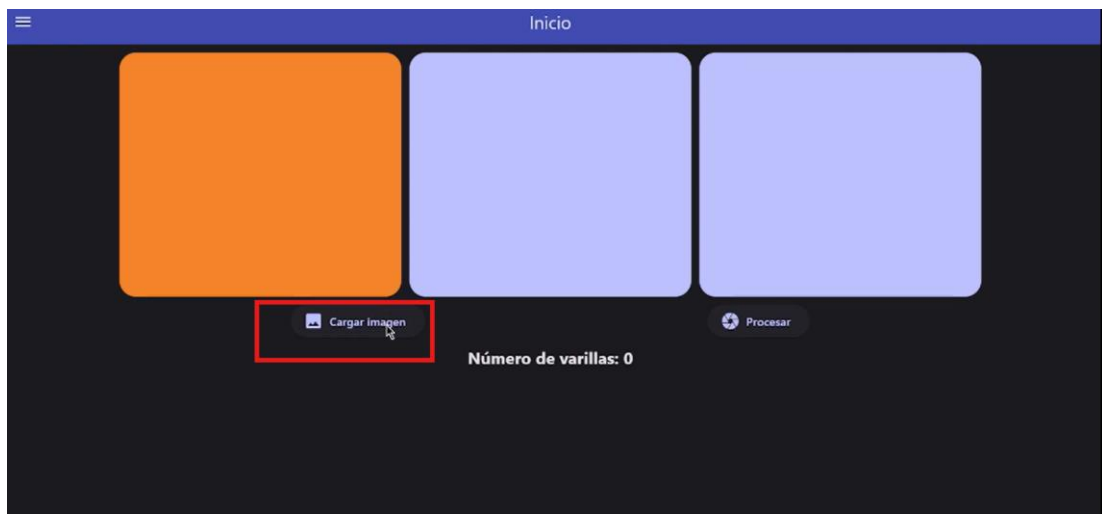
Nota: En la siguiente figura se indica la pantalla de seguridad del programa en oscuro.

C. Pantalla de inicio y procesamiento

Una vez seleccionado el botón de inicio y procesamiento, el operario selecciona el botón de cargar imagen, tal como se indica en la figura 4.5.

Figura 4.5

Imagen de la Pantalla de inicio y procesamiento

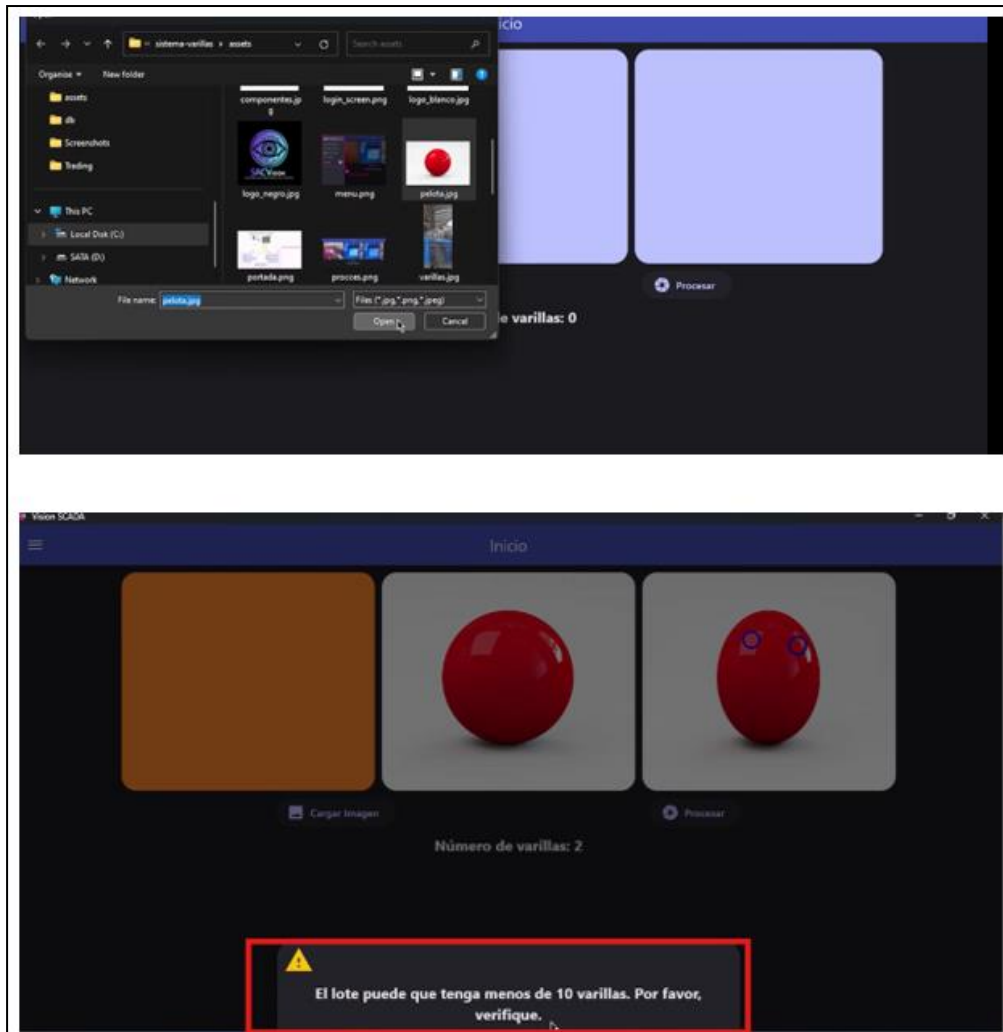


Nota: En la siguiente figura se indica la pantalla de procesamiento del programa.

Después se despliega una ventana en donde se busca la imagen que será analizada, en este caso se selecciona una imagen de una esfera, para demostrar que el sistema presenta una alerta cuando la cantidad de varillas es insuficiente. Tal como se indica en la figura 4.6.

Figura 4.6

Imagen de la Pantalla de inicio y procesamiento

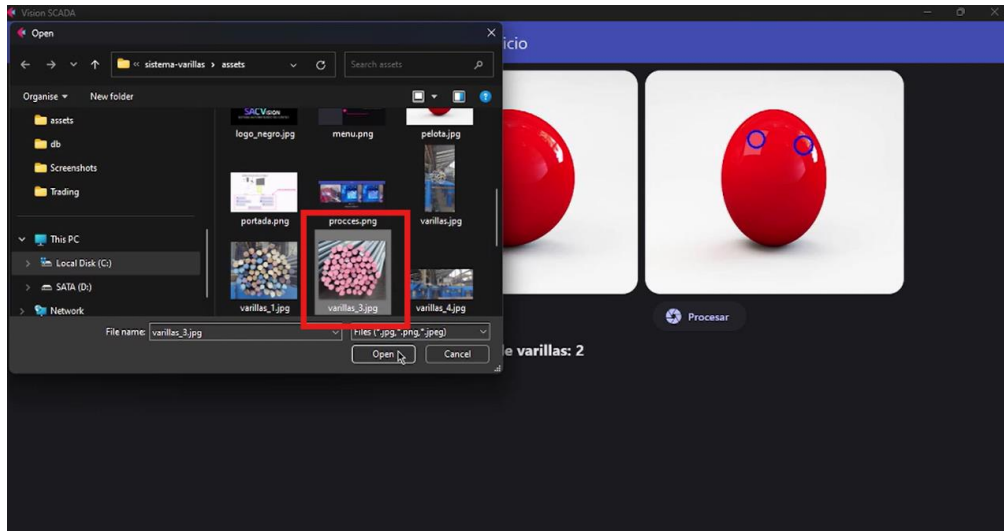


Nota: En la siguiente figura se indica la pantalla de procesamiento y la ventana de alerta del programa.

En la figura 4.7 se puede apreciar que se carga la imagen de prueba de paquete de varillas,

Figura 4.7

Imagen de la Pantalla de inicio y procesamiento



Nota: En la siguiente figura se indica la ventana de donde carga la imagen para el procedimiento de conteo.

Una vez cargada la imagen de prueba se puede observar en la figura 4.8, que se realiza un conteo de varillas de manera automática, en donde el valor total de varillas se indica en la parte inferior de la pantalla en el rectángulo rojo.

Figura 4.8

Imagen de la Pantalla de inicio y procesamiento

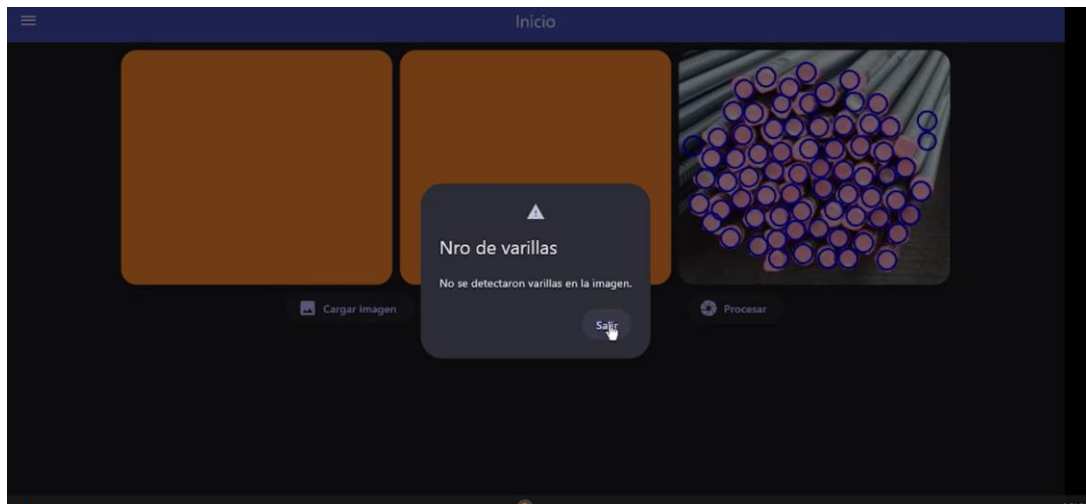


Nota: En la siguiente figura se indica la pantalla del programa con la imagen procesado y el número de varillas contados.

También se cuándo no se carga alguna imagen se proyecta un mensaje que dice que no se detectaron varillas tal como se puede apreciar en la figura 4.9

Figura 4.9

Imagen de la Pantalla de inicio y procesamiento



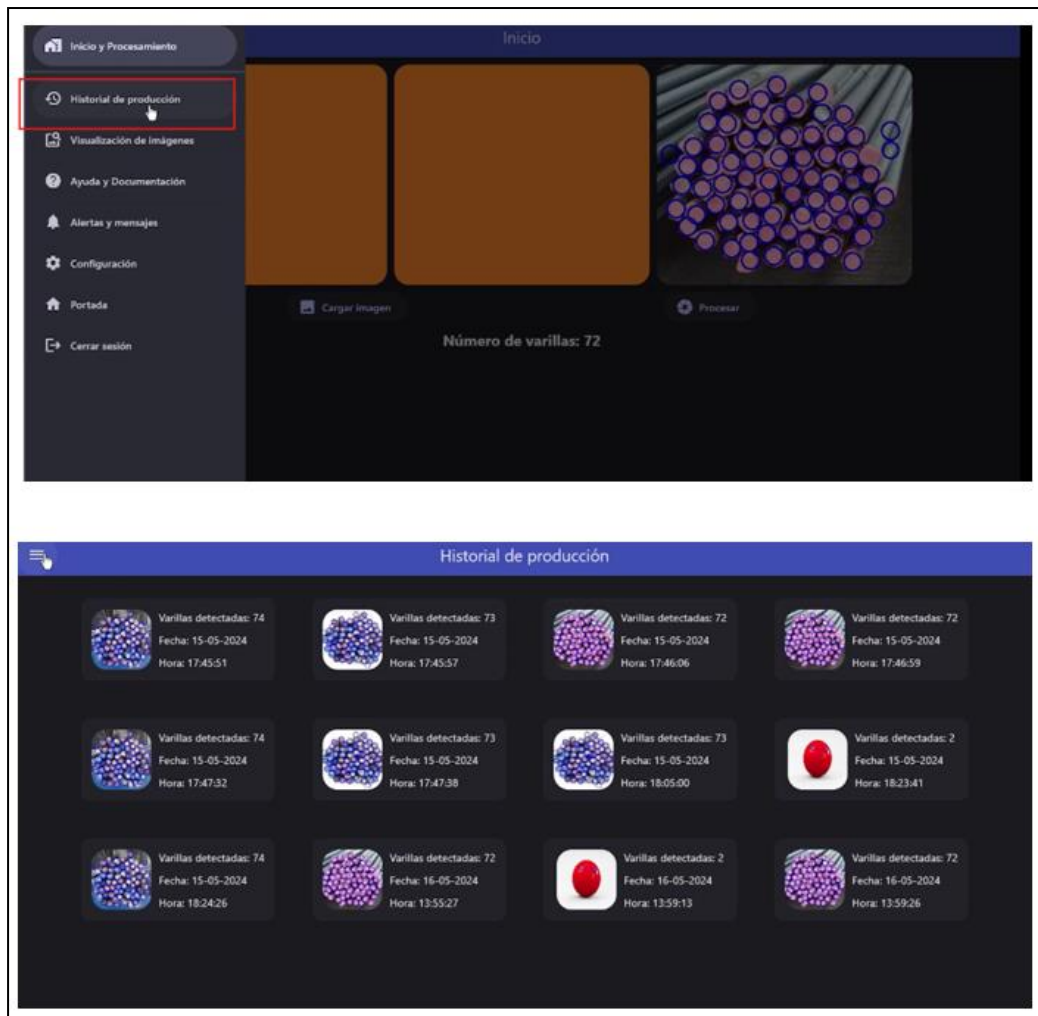
Nota: En la siguiente figura se indica la pantalla de procesamiento con una alerta donde no se detecta varillas.

D. Historial.

También el operario puede observar el historial de producción, seleccionando el su respectivo botón, es decir los paquetes de varillas contados anteriormente tal como se aprecia en la imagen 4.10. Se debe tomar en cuenta que el sistema SCADA también tiene un menú despegable en la parte izquierda de la pantalla para que el operario puede acceder al menú de manera de otra manera y de forma más interactiva

Figura 4.10

Imagen de la Pantalla del historial



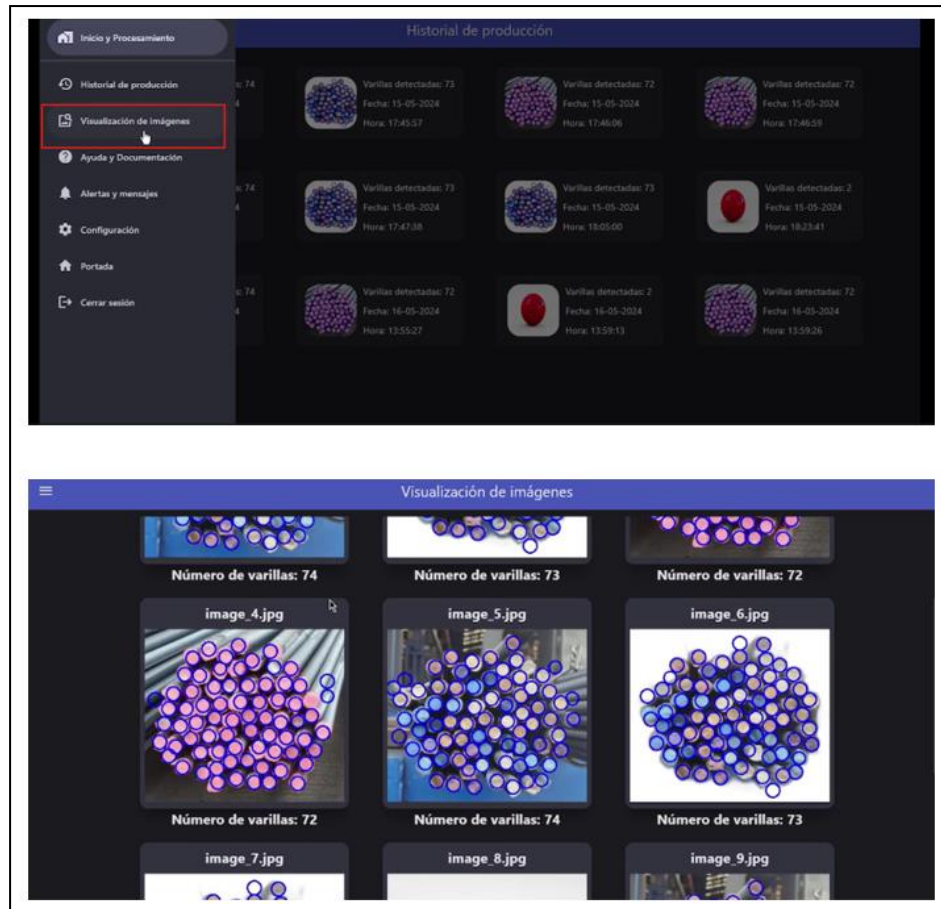
Nota: En la siguiente figura se indica la pantalla del historial de cada proceso de conteo.

E. Visualización de imágenes

En la figura 4.11 se puede apreciar, que el usuario selecciona el botón de visualización de imágenes, el sistema indica una ventana con todos los paquetes anteriormente contados, los mismos que se encuentran etiquetados.

Figura 4.11

Imagen de la Pantalla de visualización de imágenes



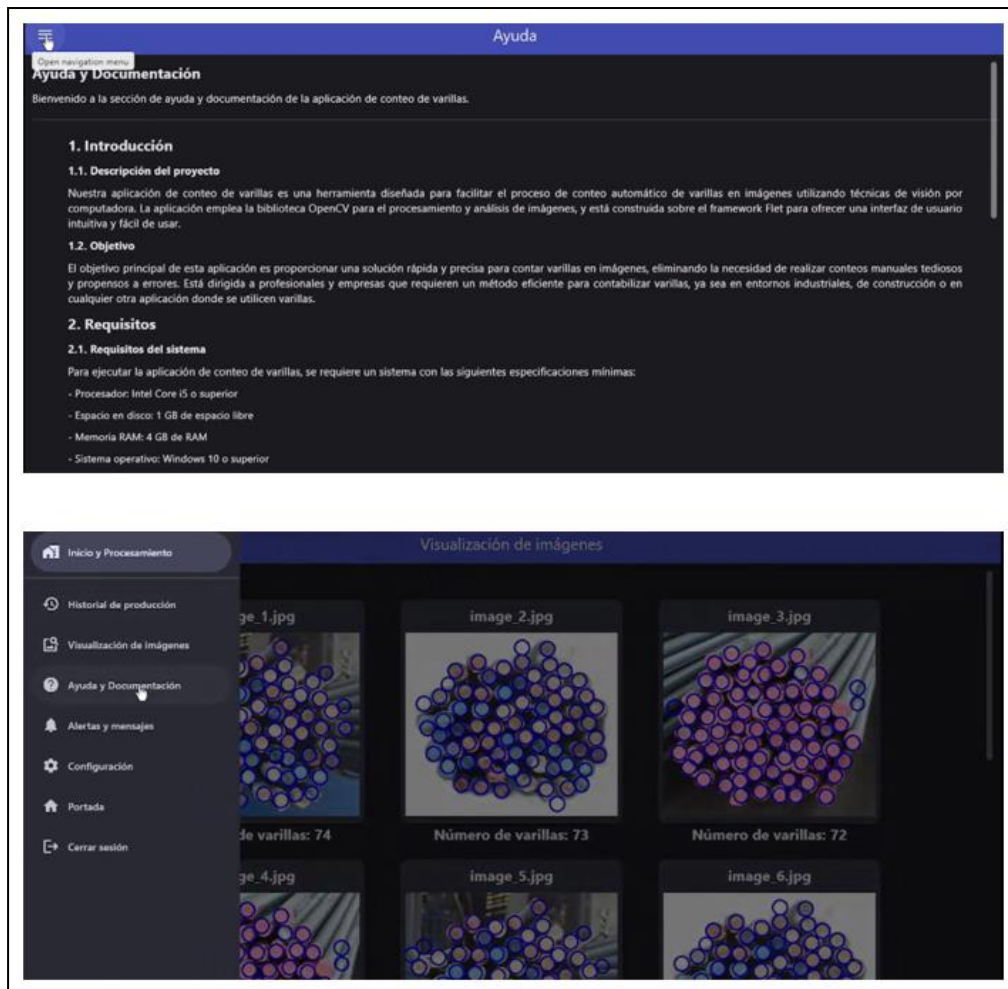
Nota: En la siguiente figura se indica la pantalla del historial de cada imagen procesada.

F. Ayuda y Documentación

Otro factor importante del sistema es su ayuda y soporte el mismo que el usuario puede seleccionar en el menú desplegado en la parte izquierda de la pantalla, tal como se indica en la figura 4.12, en esta ventana se desplegara toda información necesaria del sistema automatizado de conteo de varillas.

Figura 4.12

Imagen de la Pantalla de ayuda y documentación



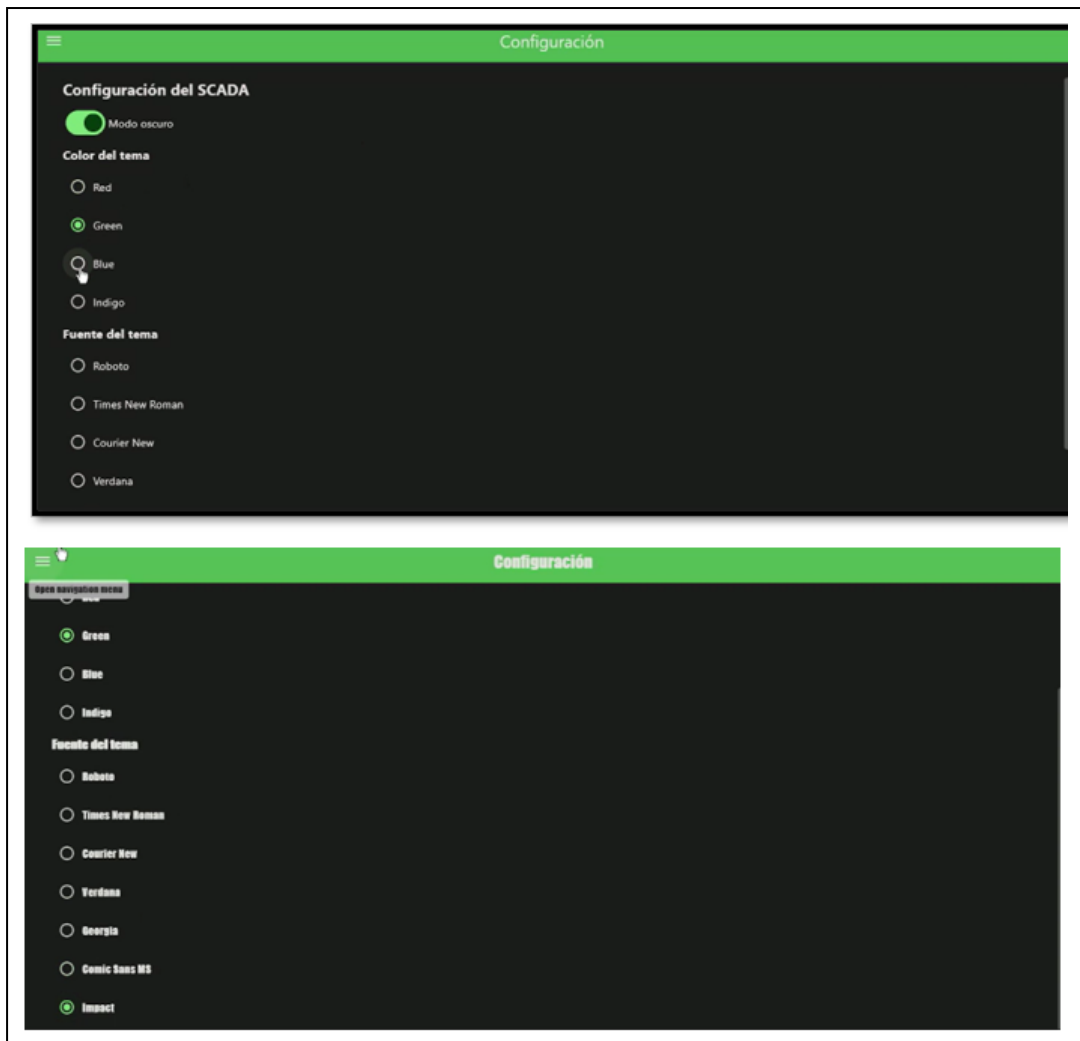
Nota: En la siguiente figura se indica la pantalla de ayuda para el usuario.

G. Configuración

También el operario podrá configurar el sistema en cuanto a su color, tamaño de la ventana y tipo de letra tal como se aprecia en la figura 4.13.

Figura 4.13

Imagen de la Pantalla de configuración



Nota: En la siguiente figura se indica la pantalla de configuración para el programa.

4.2. Pruebas de Exactitud del conteo



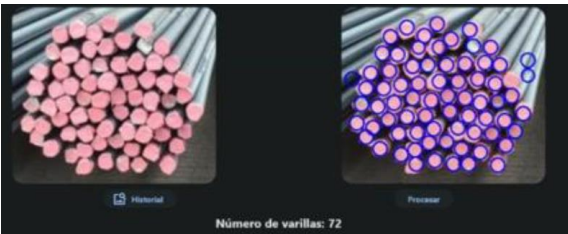

Para realizar las pruebas de exactitud en el conteo de varillas, se utiliza la norma NC-ISO 5725-1, detallada en el Anexo J, en donde indica que, para ejecutar este tipo de procedimientos con visión artificial, se debe obtener diversos ensayos, donde se obtendrá los resultados de exactitud.





Por lo tanto, se desarrolla el procesamiento de 8 diferentes paquetes de varillas, tomadas a diferentes horas del día, en la tabla 4.1 se aprecia las imágenes con los respectivos resultados obtenidos, en donde se tiene columnas de imagen del paquete de varillas con su procesamiento, Número de varillas contadas

manualmente y por software. Se considera que el software se ha ido optimizando hasta disminuir el error de conteo al mínimo.

Tabla 4.1

Tabla de comparación de tomas d imágenes de varillas

| Nro de Prueba | Imagen del paquete de varillas con su respectivo procesamiento | Nro de varillas contadas manualmente | Nro de varillas contadas por software |
|---------------|--|--------------------------------------|---------------------------------------|
| 1 |  | 76 | 74 |
| 2 |  | 74 | 73 |
| 3 |  | 74 | 72 |
| 4 |  | 89 | 90 |

| | | | |
|---|--|----|----|
| 5 |  | 70 | 70 |
| 6 |  | 21 | 21 |
| 7 |  | 6 | 6 |
| 8 |  | 59 | 59 |

Nota: En tabla se aprecia las imágenes con los respectivos resultados obtenidos de la toma de la foto, en dónde se tiene columnas de imagen del paquete de varillas con su procesamiento.

4.3. Pruebas de velocidad del conteo

Para realizar las pruebas de conteo de varillas en los paquetes en cada lote de producción, se procede a realizar una tabla con distintas cantidades de varillas, respecto a la norma ISO/IEC 25010 (norma que evalúa el tiempo de respuesta del software), para obtener diferentes tiempos y poder compararlos con los tiempos en el que el software se demora en contar misma cantidad de varillas. En la tabla 4.2 se aprecia que se realiza 10 pruebas de velocidad con el conteo manual obteniendo un número promedio.

Tabla 4.2

Tabla de valores de velocidad de respuesta manual

| Prueba | 65 varillas | 75 varillas | 85 varillas | 95 varillas | 100 | 110 |
|-----------------|-------------|-------------|-------------|-------------|----------------------|----------------------|
| | (Manual) | (Manual) | (Manual) | (Manual) | varillas (Manual) | varillas (Manual) |
| 1 | 6.3 min | 6.5 min | 6.7 min | 6.9 min | 7.1 min | 7.3 min |
| 2 | 6.4 min | 6.6 min | 6.8 min | 7.0 min | 7.2 min | 7.4 min |
| 3 | 6.5 min | 6.7 min | 6.9 min | 7.1 min | 7.3 min | 7.5 min |
| 4 | 6.6 min | 6.8 min | 7.0 min | 7.2 min | 7.4 min | 7.6 min |
| 5 | 6.7 min | 6.9 min | 7.1 min | 7.3 min | 7.5 min | 7.7 min |
| 6 | 6.8 min | 7.0 min | 7.2 min | 7.4 min | 7.6 min | 7.8 min |
| 7 | 6.9 min | 7.1 min | 7.3 min | 7.5 min | 7.7 min | 7.9 min |
| 8 | 7.0 min | 7.2 min | 7.4 min | 7.6 min | 7.8 min | 8.0 min |
| 9 | 7.1 min | 7.3 min | 7.5 min | 7.7 min | 7.9 min | 8.1 min |
| 10 | 7.2 min | 7.4 min | 7.6 min | 7.8 min | 8.0 min | 8.2 min |
| Promedio | 6.9 min | 7.1 min | 7.3 min | 7.5 min | 7.7 min | 7.9 min |

Nota: En la tabla se encuentran los resultados de las pruebas de velocidad en relación al operador.

En la tabla 4.3 se encuentra las pruebas de velocidad de respuesta con el software de conteo de varillas automático.

Tabla 4.3

Tabla de valores de velocidad de respuesta con el software

| Prueba | 65 varillas | 75 varillas | 85 varillas | 95 varillas | 100 varillas | 110 varillas |
|--------|---------------|---------------|---------------|---------------|---------------|---------------|
| | (Automático) | (Automático) | (Automático) | (Automático) | (Automático) | (Automático) |
| 1 | 0.48 s | 0.51 s | 0.54 s | 0.57 s | 0.60 s | 0.63 s |
| 2 | 0.49 s | 0.52 s | 0.55 s | 0.58 s | 0.61 s | 0.64 s |
| 3 | 0.50 s | 0.53 s | 0.56 s | 0.59 s | 0.62 s | 0.65 s |
| 4 | 0.51 s | 0.54 s | 0.57 s | 0.60 s | 0.63 s | 0.66 s |
| 5 | 0.52 s | 0.55 s | 0.58 s | 0.61 s | 0.64 s | 0.67 s |
| 6 | 0.53 s | 0.56 s | 0.59 s | 0.62 s | 0.65 s | 0.68 s |
| 7 | 0.54 s | 0.57 s | 0.60 s | 0.63 s | 0.66 s | 0.69 s |
| 8 | 0.55 s | 0.58 s | 0.61 s | 0.64 s | 0.67 s | 0.70 s |
| 9 | 0.56 s | 0.59 s | 0.62 s | 0.65 s | 0.68 s | 0.71 s |
| 10 | 0.57 s | 0.60 s | 0.63 s | 0.66 s | 0.69 s | 0.72 s |

| | | | | | | |
|---------|--------|--------|--------|--------|--------|--------|
| Promedi | 0.53 s | 0.56 s | 0.59 s | 0.62 s | 0.65 s | 0.68 s |
| o | | | | | | |

Nota: En la tabla se encuentran los resultados de las pruebas de velocidad en relación al software.

4.4. Análisis de Resultados

Se procede a realizar el análisis de resultados de las pruebas realizadas al sistema, para obtener los datos que ayudaran a concluir el desempeño del sistema automático de detección de varillas.

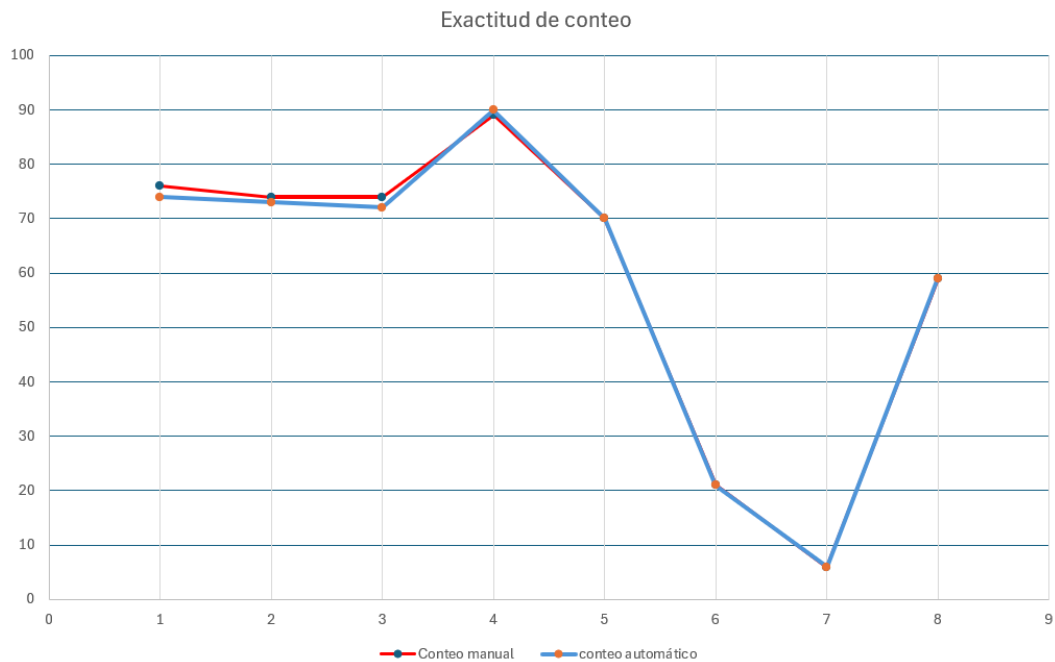
4.4.1. Análisis de resultados de la exactitud del conteo

Se obtiene resultados mediante gráficas que explican como el software ha sido optimizado hasta obtener un error mínimo.

En la figura 4.1 se visualiza una gráfica de dispersión que compara la relación entre el conteo realizado por un operario vs el conteo realizado por un sistema automático de varillas, tomando en cuenta que el parámetro de conteo manual es el valor real del número de varillas; se recalca que estos datos están tomados de las pruebas de exactitud de conteo, precisamente de la tabla 4.14.

Figura 4.14

Gráfica de comparación conteo manual vs conteo de software



Nota: en la siguiente figura se observa la comparación entre los valores medidos de la velocidad del operador vs el software.

En la figura 4.14 se observa que a través que se fueron haciendo las pruebas, el sistema automático de conteo fue mejorando, hasta ya tener un error mínimo, en el eje de las y se indica la cantidad de varillas contadas, en el eje de las x están el número de la prueba, la línea de color azul indica el conteo automático y la línea roja indica el conteo manual (tomando en cuenta que este parámetro se lo toma como valor real).

Cálculo de errores.

Se procede a calcular los errores absolutos y relativos que existen en el presente proyecto, que ayuda entender la exactitud del sistema, para obtener conclusiones e identificar parámetros de mejora.

Error absoluto.

Se calcula el error absoluto entre el conteo de varillas por el operario (valor real), respecto con el conteo de varillas realizado por el software. Los datos se obtienen de la tabla 4.1 de las columnas conteo de varillas.

La ecuación que se utiliza es la siguiente:

$$Ea = |Vm - Va|$$

(4, 1)

Donde:

- Ea=Error Absoluto
- Vm=Valor conteo manual.
- Va=Valor conteo automático

En la tabla 4.4 se indica los resultados del cálculo del error absoluto de cada prueba realizada en el sistema.

Tabla 4.4

Tabla de resultados del cálculo de error

| Nro. de Prueba | Valor Absoluto (número de varillas) |
|----------------|--|
| 1 | $Ea1 = 2$ |
| 2 | $Ea2 = 2$ |
| 3 | $Ea3 = 2$ |
| 4 | $Ea4 = 1$ |
| 5 | $Ea5 = 0$ |
| 6 | $Ea6 = 0$ |
| 7 | $Ea7 = 0$ |
| 8 | $Ea8 = 0$ |

Nota: En la tabla se indican los resultados obtenidos del valor del error absoluto.

Error Relativo.

Se lo realiza para sacar los porcentajes de error de cada prueba para al final realizar el promedio del error total del sistema. La ecuación que se utiliza se describe a continuación:

$$Er = \frac{Ea}{Vm} \times 100\%$$

(4, 2)

Donde:

- Er=Error relativo
- Ea=Error Absoluto
- Vm=Valor conteo manual.

En la tabla 4.5 se encuentra los resultados de los errores relativos por cada prueba realizada al sistema, al final se obtiene el promedio de error explicada mediante gráfica.

Tabla 4.5

Tabla de resultados del cálculo de erros relativo

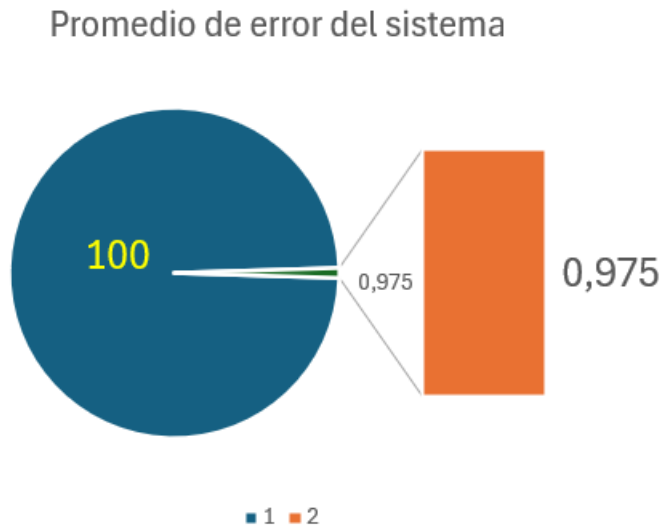
| Nro. de Prueba | Valor Relativo (porcentaje) |
|-----------------------|--|
| 1 | <i>Er1 = 2.63%</i> |
| 2 | <i>Er2 = 1.35%</i> |
| 3 | <i>Er3 = 2.70%</i> |
| 4 | <i>Er4 = 1.2%</i> |
| 5 | <i>Er5 = 0%</i> |
| 6 | <i>Er6 = 0%</i> |
| 7 | <i>Er7 = 0%</i> |
| 8 | <i>Er8 = 0%</i> |
| Promedio: | <i>Epr = 0.975%</i> |

Nota: En la tabla se indican los resultados obtenidos del valor del error relativo.

El error relativo total se lo puede observar en la figura 4.15 obtenida de la tabla, en donde se indica que el valor promedio total de error es 0.975%, en donde este es el promedio de error a través del avance de las diferentes prácticas, demostrando que el software comenzó con un error mínimo hasta lograr eliminar este parámetro por completo. por lo tanto, tomando la norma NC-ISO 5725-1, el sistema tiene una exactitud bastante alta en su medición y conteo.

Figura 4.15

Imagen del promedio del error del sistema



Nota: En la siguiente figura se indica la gráfica en relación del error absoluto vs el error relativo.

4.5. Análisis de velocidad del conteo.

La velocidad de conteo con los dos métodos, el uno manual y el otro automático es significativa, tal como se puede apreciar las tablas 4.2 y 4.3, ya que el tiempo de repuesta del software de conteo es de 0.6 s por paquete de varillas, en cambio el tiempo que demora el operario en contar una varilla es de 5,7 s. Por lo tanto, la eficiencia del software de varillas es significativo.

Explicación del cálculo del tiempo promedio del conteo de varillas manual.

Para calcular el tiempo promedio que tarda un operario en contar una varilla, se deben seguir los siguientes pasos:

1. Sumar los tiempos promedio de todas las cantidades de varillas:

$$6.9 \text{ min} + 7.1 \text{ min} + 7.3 \text{ min} + 7.5 \text{ min} + 7.7 \text{ min} + 7.9 \text{ min} = 44.4 \text{ min}$$

2. Dividir la suma total del tiempo promedio por la cantidad total de varillas contadas:

$$\frac{44.4 \text{ min}}{465 \text{ varillas}} = 0.095 \frac{\text{min}}{\text{varilla}} \quad (4, 3)$$

3. Convertir el tiempo promedio a segundos:

$$0.095 \frac{\text{min}}{\text{varilla}} \times 60 \frac{\text{s}}{\text{min}} = 5.7 \frac{\text{s}}{\text{varilla}} \quad (4, 4)$$

4.6. Análisis de Costos

Se procede a realizar un análisis de costos comparando el sistema de conteo de varillas aplicando visión artificial y SCADA, con respecto a softwares o sistemas industriales que realizan procedimientos similares que se encuentran en el mercado. Esta comparación se puede observar en la tabla 4.6. Se toman en cuenta parámetros importantes que softwares este tipo deben proporcionar a una empresa de siderurgia.

Tabla 4.6

Tabla de análisis de costos

| Criterio | Proyecto (Python +Scada) | Cognex (In- Sight) | (In- Series) | Keyence (IV Series) | (IV Series) | Siemens (Simatic WinCC) | Rockwell Automation (FactoryTalk) |
|-------------------------|--------------------------------|---------------------------|------------------|---------------------------|------------------|-------------------------------|---|
| Costos Iniciales | | | | | | | |
| Hardware | | | | | | | |
| Cámaras | \$300 - \$700 cada una | \$1,000 \$2,000 una | - cada una | \$1,000 \$2,000 una | - cada una | \$1,000 \$2,000 una | - cada una |

| | | | | | | |
|--------------------------------|-------------------------------|----------------------------|----------------------------|---------------------------------|-----------------------------------|----------------------------|
| Computadoras/servidores | \$800 - \$1,500 cada uno | \$1,200 - \$2,500 cada uno | \$1,200 - \$2,500 cada uno | \$1,200 - \$2,500 cada uno | \$1,200 - \$2,500 cada uno | \$1,200 - \$2,500 cada uno |
| Sensores y actuadores | \$100 - \$300 cada uno | \$400 - \$800 cada uno | \$400 - \$800 cada uno | \$400 - \$800 cada uno | \$400 - \$800 cada uno | \$400 - \$800 cada uno |
| Equipos de red | \$100 - \$300 | \$200 - \$600 | \$200 - \$600 | \$200 - \$600 | \$200 - \$600 | \$200 - \$600 |
| Software | | | | | | |
| Licencia de Software | Gratuito | \$3,000 - \$8,000 | \$3,000 - \$8,000 | \$5,000 - \$15,000 | \$5,000 - \$15,000 | \$5,000 - \$15,000 |
| Desarrollo del Sistema | \$0 | \$10,000 - \$20,000 | \$10,000 - \$20,000 | \$10,000 - \$20,000 | \$10,000 - \$20,000 | \$10,000 - \$20,000 |
| Mantenimiento Anual | | | | | | |
| Actualizaciones de Software | \$0 | \$3,000 - \$6,000 | \$3,000 - \$6,000 | \$5,000 - \$10,000 | \$5,000 - \$10,000 | \$5,000 - \$10,000 |
| Soporte Técnico | \$2,000 - \$5,000 | \$4,000 - \$8,000 | \$4,000 - \$8,000 | \$5,000 - \$10,000 | \$5,000 - \$10,000 | \$5,000 - \$10,000 |
| Sustitución de Hardware | \$500 - \$1,500 | \$2,000 - \$4,000 | \$2,000 - \$4,000 | \$2,000 - \$4,000 | \$2,000 - \$4,000 | \$2,000 - \$4,000 |
| Funcionalidades | | | | | | |
| Visión Artificial | Sí (OpenCV) | Sí | Sí | Integración disponible | Integración disponible | |
| SCADA | Sí (PyScada) | Integración disponible | Integración disponible | Sí | Sí | |
| Personalización | Alta (código abierto) | Media (configurable) | Media (configurable) | Baja (dependiente de licencias) | Alta (configurable) | |
| Soporte | Comunidad y autogestión | Profesional (Cognex) | Profesional (Keyence) | Profesional (Siemens) | Profesional (Rockwell Automation) | |
| Facilidad de Implementación | Media (requiere conocimiento) | Alta (solución integrada) | Alta (solución integrada) | Alta (solución integrada) | Media-Alta (flexible) | |
| Total, Costos Iniciales | \$1,800 - \$3,800 | \$15,800 - \$33,900 | \$15,800 - \$33,900 | \$17,800 - \$40,900 | \$17,800 - \$40,900 | \$17,800 - \$40,900 |

| | | | | | | | | | | |
|---------------------|---------|---|----------|---|----------|---|----------|---|----------|---|
| Total, | \$2,500 | - | \$9,000 | - | \$9,000 | - | \$12,000 | - | \$12,000 | - |
| Mantenimient | \$6,500 | | \$18,000 | | \$18,000 | | \$24,000 | | \$24,000 | |
| o Anual | | | | | | | | | | |

Nota: En la tabla se indican el estudio y comparación del análisis de costos del proyecto.

Observando la tabla 4.6 se puede deducir que el proyecto es Ideal para industrias de siderurgia, donde se requiere alta personalización y flexibilidad, con un equipo técnico capaz de desarrollar y mantener el sistema. El sistema automático de conteo de varillas nos da una solución más económica ya que es aproximadamente 8.33 veces menor en términos de software y desarrollo con respecto a los otros softwares utilizados.

4.7. Conclusiones

- Se llevó a cabo un análisis detallado de los métodos de conteo de varillas actuales de la empresa siderúrgica con el objetivo de encontrar áreas de mejora. Se encontró que el proceso actual era manual, propenso a errores y consumía mucho tiempo y recursos; varias áreas importantes de mejora incluyeron la implementación de un sistema de conteo automatizado, la estandarización del proceso y la capacitación de los empleados. Se estima que la implementación de estas mejoras podría reducir significativamente los costos y aumentar la eficiencia.
- Se diseñó y desarrolló un prototipo de sistema automatizado de conteo de varillas que integra la visión artificial; el sistema se basó en el uso de una cámara industrial para capturar imágenes de las varillas, y un software de procesamiento de imágenes para identificar y contar las varillas individuales. El prototipo se probó con éxito en una variedad de condiciones de iluminación y con diferentes tipos de varillas.
- Se diseñó un modelo SCADA para el programa de conteo automatizado de varillas, el modelo SCADA se basó en una arquitectura cliente-servidor, con un servidor central que recopilaba datos de las cámaras industriales y los sensores, y un cliente que proporcionaba una interfaz gráfica de usuario para que los operadores monitorearan y controlaran el sistema. El modelo SCADA se diseñó para ser escalable y modular, para que pudiera adaptarse fácilmente a diferentes necesidades y requisitos.
- Se realizó una evaluación exhaustiva de la precisión, eficiencia y seguridad del sistema automatizado de conteo de varillas en un entorno real, la evaluación reveló que el sistema era altamente preciso y eficiente, y que podía contar las varillas con un alto grado de precisión y en un tiempo significativamente menor que el sistema manual; también se encontró que el sistema era seguro y no presentaba ningún riesgo para los operadores o el entorno.
- Se elaboró una guía completa para la implementación del sistema automatizado de conteo de varillas, la guía proporcionó instrucciones

detalladas sobre todos los aspectos de la implementación, desde la planificación e instalación del sistema hasta la capacitación de los empleados y el mantenimiento del sistema.

4.8. Recomendaciones

- El análisis ha demostrado que la automatización del proceso de conteo de varillas ofrece importantes beneficios, como la reducción de errores, el aumento de la eficiencia y la disminución de costos. Se recomienda invertir en la implementación del prototipo de sistema automatizado desarrollado, o en un sistema comercial similar, para optimizar el proceso de conteo.
- La transición a un sistema automatizado y la estandarización del proceso requerirá que los empleados adquieran nuevas habilidades y conocimientos. Se recomienda desarrollar un programa de capacitación integral que familiarice a los empleados con el nuevo sistema, los procedimientos estandarizados y las mejores prácticas para el conteo de varillas.
- La evaluación ha demostrado que el modelo SCADA diseñado es robusto, escalable y efectivo para el monitoreo y control del sistema automatizado de conteo de varillas. Se recomienda implementar este modelo en la operación real para optimizar la gestión y supervisión del sistema.
- Para maximizar el valor del sistema de conteo automatizado, se recomienda integrarlo con otros sistemas de la empresa, como el sistema de gestión de inventario o el sistema de control de producción. Esto permitirá una mejor comunicación y flujo de datos entre los diferentes sistemas, lo que puede conducir a una mayor eficiencia y optimización general de las operaciones.
- Para garantizar el funcionamiento continuo y confiable del sistema automatizado de conteo de varillas, se recomienda implementar un programa de mantenimiento preventivo. Este programa debe incluir inspecciones regulares, limpieza y calibración de los componentes del sistema, así como actualizaciones periódicas del software. Un programa de mantenimiento adecuado ayudará a prevenir fallas y prolongar la vida útil del sistema.

Bibliografía

- Alejandro, D. R., & vadillo Vidal, C. (2013). *Conteo de varillas de acero por procesamiento de imágenes*. Lima: Universidad Peruana de Ciencias Aplicadas .
- Algarín, C. R., Llanos, R. C., & Hernández, A. S. (Enero de 2012). Diseño de un sistema Scada basado en labview, implementando el protocolo de comunicación inalámbrica zigbee. *Dialnet*, págs. 44-52.
- Amaya, S., Velásquez, D., & Torres, I. (2015). *Desarrollo e Implementación de un Sistema de Visión Artificial Basado en Lenguajes de Uso Libre para un Sistema Seleccionador de Productos de un Centro Integrado de Manufactura (CIM)*. Medellín.
- Boada Robayo, E. (2024). *Análisis y diseño de un sistema SCADA del sistema de microrred no convencional del edificio Julio Mario Santo Domingo de la Universidad de los Andes*. Bogotá: Universidad de los Andes.
- Cabús, J., Navarrete, D., & Porras, R. (2004). *Sistemas SCADA*. Cataluña: Universidad Politécnica de Cataluña.
- Corella, J., & Reyes, J. (2022). *Elaboración de un sistema SCADA para el monitoreo y control en una línea automatizada de snacks fritos y acceso a tecnología 4.0*. Guayaquil: ESPOL.
- Lojan, E., & Iñiguez, D. (2009). *Diseño de un sistema HMI/SCADA para una planta de clasificación con visión artificial* . Cuenca: Universidad del Azuay.
- Niño Rondón, C. V., Castellano Carvajal, D. A., Castro Casadiego, S. A., Medina Delgado, B., & Guevara Ibarra, D. (2022). Una aproximación a la detección de bordes en imágenes médicas mediante análisis de histograma y gradiente morfológico. *Ingeniería y competitividad: revista científica y tecnológica*, págs. 1-18.
- Pastor, A. A. (2023). *Diseño de un sistema de supervisión y control remoto de una mesa XY* . Barcelona : EPSEVG, Ingeniería Superior de Automática y Electrónica Industrial .
- RENNARD, F. L., AUGUSTO, O. A., & JIMENEZ, C. (2024). *AUTOMATIZACIÓN DEL PROCESO DE SELECCIÓN DE MANGOS UTILIZANDO PROCESAMIENTO DE*

IMÁGENES Y REDES NEURONALES. Perú: UNIVERSIDAD NACIONAL DEL CALLAO .

Sánchez, J. (2011). *Identificación de objetos en tiempo real utilizando técnicas y clasificadores de visión artificial para el reconocimiento de patrones* . Cuenca : Universidad Politécnica Salesiana .

Vaca, E. (2019). *IMPLEMENTACIÓN DE UN SISTEMA SCADA MEDIANTE EL SOFTWARE INTOUCH PARA EL CONTROL Y VISUALIZACIÓN DE PROCESOS INDUSTRIALES*. Ibarra: UNIVERSIDAD TÉCNICA DEL NORTE.

Zavaleta, B., Baquedano, U., Jhajaira, A. A., R., R. M., & M., J. S. (2 de Agosto de 2022). Desarrollo de un sistema de visión artificial para detección de abolladuras en latas de atún, Trujillo. *INGnosis.*, pág. 62.

APÉNDICES

Apéndice A

Nombre: Función principal para construir el programa y navegar sobre las diferentes pantallas

```
import flet as ft

from screens.form_screen import FormScreen
from screens.historial_screen import HistorialScreen
from screens.home_screen import HomeScreen
from screens.images_screen import ImagesScreen
from screens.portada_screen import PortadaScreen

def main(page: ft.Page):

    def route_change(route):
        page.views.clear()
        if page.route == "/historial":
            page.views.append(
                HistorialScreen(page=page)
            )
        elif page.route == "/images":
            page.views.append(
                ImagesScreen(page=page)
            )
        elif page.route == "/home":
            page.views.append(
                HomeScreen(page=page)
            )
        elif page.route == "/portada":
            page.views.append(
```

```

        PortadaScreen(page=page)
    )
    elif page.route == "/":
        page.views.append(
            FormScreen(page=page)
        )

    page.update()

    page.title = "Vision SCADA"
    page.theme_mode = ft.ThemeMode.LIGHT
    page.window_width = 450
    page.window_height = 600
    page.on_route_change = route_change
    page.go(page.route)

ft.app(target=main

```

Apéndice B

Nombre: Función que permite el procesamiento de la imagen con visión artificial y realizar el conteo de las varillas.

```

def proccess_and_count(e):
    nonlocal count, id_img
    num_imgs = len(db.get_images())
    if num_imgs > 0:
        id_img = num_imgs + 1
    else:

```

```

        id_img = 1
        count = id_img
        img = camera_holder.get_image_from_frame()
        gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
        blurred = cv2.GaussianBlur(gray, (7, 7), 0)
        circles = cv2.HoughCircles(blurred, cv2.HOUGH_GRADIENT, 1,
                                   minDist=20, param1=100, param2=7,
minRadius=11, maxRadius=13)
        if circles is not None:
            circles = circles.round().astype(int)
            for (x, y, r) in circles[0]:
                cv2.circle(img, (x, y), r, (255, 0, 0), 2)
            num_circles = len(circles[0])
            num_varillas = num_circles

            _, buffer = cv2.imencode('.jpg', img)
            img_base64 = base64.b64encode(buffer).decode()
            image_process_holder.content = ft.Image(
                src_base64=img_base64,
                fit=ft.ImageFit.COVER,
            )
            image_process_holder.bgcolor = None
            image_process_holder.content.visible = True
            text_varillas.value = f"Número de varillas: {num_varillas}"
            save_image_to_db(base64_str=img_base64,
                            name=f"image_{count}.jpg",
count=num_varillas, e=e)

            text_varillas.update()
            image_process_holder.update()

```

Apéndice C

Nombre: Clase que permite administrar la base de datos

```
import sqlite3
from typing import List, Dict, Any

class DB:

    def __init__(self, db_path: str):
        self.db_path = db_path
        self.conn = sqlite3.connect(database=self.db_path,
check_same_thread=False, timeout=10)
        self.cursor = self.conn.cursor()
        self.create_table_users()
        self.create_table_images()
        print("Base de datos creada")

    def create_table_users(self):
        self.cursor.execute("""
            CREATE TABLE IF NOT EXISTS users (
                id INTEGER PRIMARY KEY AUTOINCREMENT,
                username TEXT,
                password TEXT
            )
        """)
        self.conn.commit()

    def create_table_images(self):
        self.cursor.execute("""
            CREATE TABLE IF NOT EXISTS images (
                id INTEGER PRIMARY KEY AUTOINCREMENT,
                name TEXT,
                base64_str TEXT,
```



```

        count INTEGER
    )
    """
    self.conn.commit()

def insert_user(self, username: str, password: str):

    get_user = self.get_user(username)
    if get_user:
        print("Usuario ya existe")
    else:
        self.cursor.execute("""
            INSERT INTO users (username, password)
            VALUES (?, ?)
        """, (username, password))
        self.conn.commit()

def get_user(self, username: str) -> Dict[str, Any]:
    self.cursor.execute("""
        SELECT * FROM users WHERE username = ?
    """, (username,))
    user = self.cursor.fetchone()
    if user:
        return {
            "id": user[0],
            "username": user[1],
            "password": user[2],
        }
    return None

def get_users(self) -> List[Dict[str, Any]]:
    self.cursor.execute("""
        SELECT * FROM users
    """)
    users = self.cursor.fetchall()

```

```

return [
    {
        "id": user[0],
        "username": user[1],
        "password": user[2],
    }
    for user in users
]

def delete_user(self, username: str):
    self.cursor.execute("""
        DELETE FROM users WHERE username = ?
    """, (username,))
    self.conn.commit()

def update_user(self, username: str, password: str):
    self.cursor.execute("""
        UPDATE users SET password = ? WHERE username = ?
    """, (password, username))
    self.conn.commit()

def close(self):
    self.conn.close()
# Path: src/services/db/images_db.py

def insert_image(self, name: str, base64_str: str, count: int):
    self.cursor.execute("""
        INSERT INTO images (name, base64_str, count)
        VALUES (?, ?, ?)
    """, (name, base64_str, count))
    self.conn.commit()

def get_image(self, name: str) -> Dict[str, Any]:
    self.cursor.execute("""

```

```

        SELECT * FROM images WHERE name = ?
        """ , (name,))
    image = self.cursor.fetchone()
    if image:
        return {
            "id": image[0],
            "name": image[1],
            "base64_str": image[2],
            "count": image[3]
        }
    return None

def get_images(self) -> List[Dict[str, Any]]:
    self.cursor.execute("""
        SELECT * FROM images
    """)
    images = self.cursor.fetchall()
    return [
        {
            "id": image[0],
            "name": image[1],
            "base64_str": image[2],
            "count": image[3]
        }
        for image in images
    ]

```

Apéndice D

Nombre: Clase que permite mostrar la cámara dentro del programa

```
import flet as ft
import base64
import numpy as np
import cv2
num_camera = 0
class Camera(ft.UserControl):

    def __init__(self, cap=cv2.VideoCapture(num_camera)):
        super().__init__()
        self.cap = cap
        self.image = None
        self.image_base64 = None
        self.width = 350
        self.height = 300

    def did_mount(self):
        self.update_timer()

    def get_cap(self):
        return self.cap

    def update_timer(self):

        while True:
            ret, frame = self.cap.read()
            self.image = frame
            if ret:
                frame = cv2.resize(frame, (self.width,
self.height))
                _, buffer = cv2.imencode('.jpg', frame)
                self.image_base64 = base64.b64encode(buffer)
```

```
        self.img.src_base64 = self.image_base64.decode('utf-8')
        self.update()
    else:
        print("Error al leer la cámara")

        self.update()

def get_image_from_frame(self):
    return self.image

def build(self):
    self.img = ft.Image(
        src="assets/white_image.png",
        border_radius=ft.border_radius.all(20)
    )
    return self.img
```

Apéndice F

Nombre: función que construye la portada del sistema luego del inicio de sesión

```
import flet as ft

def PortadaScreen(page: ft.Page):

    return ft.View(
        route="/portada",
        horizontal_alignment=ft.CrossAxisAlignment.CENTER,
        vertical_alignment=ft.MainAxisAlignment.CENTER,
        controls=[
            ft.Text(value="SISTEMA SCADA DE CONTEO DE VARILLAS",
                    style=ft.TextStyle(size=20,
weight=ft.FontWeight.BOLD)),
            ft.Image(src="./assets/componentes.jpg", width=400,
                    height=300, fit=ft.ImageFit.COVER),
            ft.Container(
                width=600,

                content=ft.Column(
                    horizontal_alignment=ft.CrossAxisAlignment.CENTER,
                    alignment=ft.MainAxisAlignment.CENTER,
                    controls=[
                        ft.Row(
                            alignment=ft.MainAxisAlignment.SPACE_BETWEEN,
                            controls=[
                                ft.ElevatedButton(
                                    text="Configuración",
                                    on_click=lambda e: page.go("/home"),
                                    elevation=10,
                                    icon=ft.icons.SETTINGS,
                                ),
                                ft.ElevatedButton(
                                    text="Historial de producción",
```

```

        on_click=lambda e: page.go("/historial"),
        elevation=10,
        icon=ft.icons.HISTORY,
    ),
]
),
ft.Row(
    alignment=ft.MainAxisAlignment.SPACE_BETWEEN,
    controls=[
        ft.ElevatedButton(
            text="Visualización de imágenes",
            on_click=lambda e: page.go("/images"),
            elevation=10,
            icon=ft.icons.IMAGE,
        ),
        ft.ElevatedButton(
            text="Ayuda y Documentación",
            on_click=None,
            elevation=10,
            icon=ft.icons.HELP,
        ),
    ]
),
ft.Row(
    alignment=ft.MainAxisAlignment.SPACE_BETWEEN,
    controls=[
        ft.ElevatedButton(
            text="Cerrar sesión",
            on_click=lambda e: page.go("/"),
            elevation=10,
            icon=ft.icons.LOGOUT,
        ),
        ft.ElevatedButton(
            text="Alertas y mensajes",
            on_click=None,

```

```

        elevation=10,
        icon=ft.icons.NOTIFICATIONS,
    ),
]
),
ft.ElevatedButton(
    text="Inicio y Procesamiento",
    on_click=lambda e: page.go("/home"),
    elevation=10,
    icon=ft.icons.HOME,
),
]
),
),
]
)

```

Apéndice G

Nombre: Función que construye la portada del sistema luego del inicio de sesión

```

import flet as ft

def PortadaScreen(page: ft.Page):

    return ft.View(
        route="/portada",

```



```

horizontal_alignment=ft.CrossAxisAlignment.CENTER,
vertical_alignment=ft.MainAxisAlignment.CENTER,
controls=[
    ft.Text(value="SISTEMA SCADA DE CONTEO DE VARILLAS",
             style=ft.TextStyle(size=20,
weight=ft.FontWeight.BOLD)),
    ft.Image(src="./assets/componentes.jpg", width=400,
             height=300, fit=ft.ImageFit.COVER),
    ft.Container(
        width=600,

        content=ft.Column(
            horizontal_alignment=ft.CrossAxisAlignment.CENTER,
            alignment=ft.MainAxisAlignment.CENTER,
            controls=[
                ft.Row(
                    alignment=ft.MainAxisAlignment.SPACE_BETWEEN,
                    controls=[
                        ft.ElevatedButton(
                            text="Configuración",
                            on_click=lambda e: page.go("/home"),
                            elevation=10,
                            icon=ft.icons.SETTINGS,
                        ),
                        ft.ElevatedButton(
                            text="Historial de producción",
                            on_click=lambda e: page.go("/historial"),
                            elevation=10,
                            icon=ft.icons.HISTORY,
                        ),
                    ],
                ),
                ft.Row(
                    alignment=ft.MainAxisAlignment.SPACE_BETWEEN,
                    controls=[

```

```

        ft.ElevatedButton(
            text="Visualización de imágenes",
            on_click=lambda e: page.go("/images"),
            elevation=10,
            icon=ft.icons.IMAGE,
        ),
        ft.ElevatedButton(
            text="Ayuda y Documentación",
            on_click=None,
            elevation=10,
            icon=ft.icons.HELP,
        ),
    ]
),
ft.Row(
    alignment=ft.MainAxisAlignment.SPACE_BETWEEN,
    controls=[
        ft.ElevatedButton(
            text="Cerrar sesión",
            on_click=lambda e: page.go("/"),
            elevation=10,
            icon=ft.icons.LOGOUT,
        ),
        ft.ElevatedButton(
            text="Alertas y mensajes",
            on_click=None,
            elevation=10,
            icon=ft.icons.NOTIFICATIONS,
        ),
    ]
),
ft.ElevatedButton(
    text="Inicio y Procesamiento",
    on_click=lambda e: page.go("/home"),
    elevation=10,

```

```
        icon=ft.icons.HOME,
    ),
    ],
),
),
]
)
```

Apéndice H

Nombre: Función que permite construir la pantalla del historial

```
import flet as ft

from services.db.database import DB
from widgets.card_historial import CardHistorial
from widgets.drawer import Drawer

def HistorialScreen(page: ft.Page):

    db =DB("./src/services/db/users.db")
    images = db.get_images()
    row_card = ft.Row()
```

```

if len(images) == 0:
    row_card.controls.append(
        ft.Text("No hay imágenes guardadas")
    )
for img in images:
    str_base64 = img["base64_str"]
    count = img["count"]
    row_card.controls.append(
        CardHistorial(src_base64=str_base64, count=count, date="10-05-
2024")
    )
row_card.wrap = True
row_card.alignment = ft.MainAxisAlignment.CENTER

return ft.View(
    route="/historial",
    drawer=Drawer(page=page),
    scroll=ft.ScrollMode.ADAPTIVE,
    controls=[
        ft.AppBar(
            title=ft.Text("Historial"),
            center_title=True,
            bgcolor=ft.colors.SURFACE_VARIANT,
        ),
        ft.Column(
            controls=[
                ft.Text("Historial de procesos",
style=ft.TextStyle(size=20, weight=ft.FontWeight.BOLD)),
                ft.Text("Aquí se mostrarán los procesos realizados
para el sistema SCADA", style=ft.TextStyle(size=15)),
            ]
        ),
        ft.Divider(height=20),
        row_card

```



Apéndice I

Nombre: Pantalla que permite cargar las imágenes de la base de datos y mostrarlas

```
import flet as ft

from services.db.database import DB
from widgets.card_image import CardImage
from widgets.drawer import Drawer

def ImagesScreen(page: ft.Page):

    db = DB("./src/services/db/users.db")
    images = db.get_images()
    row_images_card = ft.Row()
    if len(images) == 0:
        row_images_card.controls.append(
            ft.Text("No hay imágenes guardadas")
        )
    for img in images:
        str_base64 = img["base64_str"]
        name = img["name"]
        count = img["count"]
        row_images_card.controls.append(
            CardImage(src_base64=str_base64, name=name, count=count)
        )
    row_images_card.wrap = True
```

```
return ft.View(  
    route="/images",  
    drawer=Drawer(page=page),  
    scroll=ft.ScrollMode.ADAPTIVE,  
    horizontal_alignment=ft.CrossAxisAlignment.CENTER,  
    controls=[  
        ft.AppBar(  
            title=ft.Text("Imágenes Cargadas"),  
            center_title=True,  
            bgcolor=ft.colors.SURFACE_VARIANT,  
        ),  
        ft.Container(  
            padding=ft.padding.all(20),  
            content=row_images_card  
        )  
    ]  
)
```