
Documento de Especificaciones y Requisitos de Producto [DEP] para el desarrollo de productos mecatrónicos

Proyecto: KiSS
Revisión 1.0



Abril 2025

Instrucciones para el uso de este formato

Este formato es una plantilla tipo para documentos de requisitos de producto para su desarrollo.

Está basado y es conforme con el estándar IEEE Std 830-1998 y ha sido modificada para su uso en un ambiente de desarrollo mecatrónico simplificado.

El uso de este documento permite capturar la información relevante para desarrollar un producto o algunas de sus partes, sean electrónicas, mecánicas, de software y funcionales.

Las secciones que no se consideren aplicables al sistema descrito podrán de forma justificada indicarse como no aplicables (NA).

Notas:

Los textos en color azul son indicaciones que deben eliminarse y, en su caso, sustituirse por los contenidos descritos en cada apartado.

Los textos entre corchetes del tipo “[Inserte aquí el texto]” permiten la inclusión directa de texto con el color y estilo adecuado a la sección, al pulsar sobre ellos con el puntero del ratón.

Los títulos y subtítulos de cada apartado están definidos como estilos de MS Word, de forma que su numeración consecutiva se genera automáticamente según se trate de estilos “Titulo1, Titulo2 y Titulo3”.

La sangría de los textos dentro de cada apartado se genera automáticamente al pulsar Intro al final de la línea de título. (Estilos Normal indentado1, Normal indentado 2 y Normal indentado 3).

El índice del documento es una tabla de contenido que MS Word actualiza tomando como criterio los títulos del documento.

Una vez terminada su redacción debe indicarse a Word que actualice todo su contenido para reflejar el contenido definitivo.

Ficha del documento

Fecha	Revisión	Autor	Verificado dep. calidad.
21-03-2025	1	Melvin Jr. Núñez Ramos	Mecavisión=Melvin Núñez
21-03-2025	1	Ismenia Rocío Beltré H.	Mecavisión=Ismenia Beltré

Documento validado por las partes en fecha: [\[Fecha\]](#)

Por el cliente	Por la empresa suministradora
Fdo. D./ Dña cardinal healt	Fdo. D./Dña Mecavisión



Contenido

FICHA DEL DOCUMENTO	3
CONTENIDO	4
1 INTRODUCCIÓN	6
1.1 Propósito	6
1.2 Alcance	6
1.3 Personal involucrado	7
1.4 Definiciones, acrónimos y abreviaturas	7
1.5 Referencias	8
1.6 Resumen	9
2 DESCRIPCIÓN GENERAL	9
2.1 Perspectiva del producto	9
2.2 Funcionalidad del producto	9
2.3 Características de los usuarios	¡Error! Marcador no definido.
2.4 Restricciones	10
2.5 Suposiciones y dependencias	11
2.6 Evolución previsible del sistema	12
3 REQUISITOS ESPECÍFICOS	13
3.1 Requisitos comunes de los interfaces	16
3.1.1 Interfaces de usuario	16
3.1.2 Interfaces de hardware	16
3.1.3 Interfaces de software	17
3.1.4 Interfaces de comunicación	18
3.2 Requisitos funcionales	18
3.2.1 Requisito funcional 1	¡Error! Marcador no definido.
3.2.2 Requisito funcional 2	¡Error! Marcador no definido.
3.2.3 Requisito funcional 3	¡Error! Marcador no definido.
3.2.4 Requisito funcional n	¡Error! Marcador no definido.
3.3 Requisitos no funcionales	19
3.3.1 Requisitos de rendimiento	19
3.3.2 Seguridad	20
3.3.3 Fiabilidad	20
3.3.4 Disponibilidad	20



3.3.5	Mantenibilidad	21
3.3.6	Portabilidad	21
3.4	Otros requisitos	¡Error! Marcador no definido.
4	APÉNDICES	22



1 Introducción

En la industria de dispositivos médicos, el control de calidad es crucial para garantizar la seguridad y eficacia de los productos. Los procesos manuales de verificación son propensos a errores y pueden resultar en costos adicionales debido a productos defectuosos o retirados del mercado. Este proyecto propone el diseño e implementación de un sistema automatizado de control de calidad que mejora la eficiencia operativa y reduce los errores asociados con la lectura incorrecta de códigos y datos impresos en los productos.

1.1 Propósito

El propósito de este documento es detallar el diseño, los servicios y la implementación de una empresa especializada en sistemas automatizados de control de calidad para la industria de dispositivos médicos. El enfoque principal está en la detección eficiente de errores mediante lectura de códigos y trazabilidad en tiempo real durante el proceso de producción.

Este documento incluye la descripción de los servicios ofrecidos, el mercado objetivo, las ventajas competitivas y el plan estratégico inicial para posicionar la empresa en un sector altamente regulado y exigente. Asimismo, servirá como una guía para la toma de decisiones estratégicas, operativas y comerciales, estableciendo los lineamientos para el lanzamiento, consolidación y expansión del negocio dentro del sector médico-tecnológico.

1.2 Alcance

El presente proyecto abarca el diseño, desarrollo e implementación de un sistema automatizado de control de calidad orientado específicamente a la verificación de empaques de sábanas médicas en una línea de producción. El sistema está enfocado en la **detección, decodificación y validación en tiempo real** de códigos QR, DataMatrix y texto impreso utilizando técnicas de visión por computadora y reconocimiento óptico de caracteres (OCR). Con esta solución se busca garantizar la integridad de la información crítica impresa en cada empaque, como el número de lote, la fecha de caducidad y los identificadores únicos del producto, esenciales para la trazabilidad y cumplimiento normativo.

El alcance del sistema incluye la integración de hardware (cámaras industriales de alta resolución) y software especializado (OpenCV, EasyOCR, Flask) que permita la inspección visual sin intervención humana directa. Esta solución también contempla la implementación de una interfaz web para la supervisión en tiempo real del proceso de verificación, así como alertas automáticas y generación de reportes de calidad. Además, se incluye la capacidad de comunicación con las máquinas de producción existentes, de modo que el sistema pueda **detener automáticamente el proceso** si se detecta un error, evitando así la propagación de productos defectuosos.



Desde el punto de vista organizacional, el alcance contempla la **reducción de la carga operativa humana**, eliminando la necesidad de operadores dedicados exclusivamente a tareas de inspección visual, y promoviendo una mejora en la eficiencia del personal al permitir su reasignación a funciones de mayor valor. Con ello, se estima un ahorro operativo anual de \$14,000 y una mejora significativa en la **consistencia y precisión del control de calidad**, disminuyendo el margen de error humano.

1.3 Personal involucrado

Nombre	Ismenia Beltré
Rol	Especialista en Soporte Técnico
Categoría profesional	Ingeniera
Responsabilidades	Brindar asistencia técnica remota y presencial a los clientes.
Información de contacto	809-750-9332

Nombre	Melvin Jr. Núñez
Rol	Desarrollador
Categoría profesional	Ingeniero
Responsabilidades	Desarrollar el proceso web y automatización de este proyecto
Información de contacto	849-658-9704

1.4 Definiciones, acrónimos y abreviaturas

Definiciones

- **Reconocimiento óptico de caracteres (OCR):** Tecnología que permite extraer texto desde imágenes, facilitando la digitalización de datos impresos o escritos a mano.
- **Código QR:** Tipo de código de barras bidimensional que almacena información codificada, como enlaces o números de identificación, y puede ser escaneado por cámaras.
- **Código DataMatrix:** Código de barras en 2D muy compacto, utilizado comúnmente en el etiquetado de productos para rastreo y control de calidad.
- **Cámara de inspección:** Dispositivo encargado de capturar imágenes en tiempo real del área de producción para analizar la información visual mediante software.
- **Control de calidad automatizado:** Proceso asistido por software y hardware que permite verificar si un producto cumple con los estándares definidos, sin intervención manual constante.
- **Monitoreo en tiempo real:** Supervisión continua de procesos o datos a medida que ocurren, permitiendo detección inmediata de errores.
- **Interfaz de usuario (UI):** Plataforma visual o panel desde donde los usuarios pueden interactuar con el sistema, observar datos y tomar decisiones.
- **Procesamiento de imágenes:** Técnica informática que analiza y transforma imágenes digitales con el fin de extraer información útil.
- **Captura de video continua:** Flujo ininterrumpido de imágenes obtenidas por la cámara, procesadas en tiempo real para análisis.

Acrónimos y Abreviaturas

- **OCR (Optical Character Recognition):** Reconocimiento óptico de caracteres.
- **QR (Quick Response Code):** Código de respuesta rápida.
- **GTIN (Global Trade Item Number):** Número global de artículo comercial.



- **UI (User Interface):** Interfaz de usuario.
- **API (Application Programming Interface):** Interfaz de programación de aplicaciones.
- **IP (Internet Protocol):** Protocolo de Internet.
- **AI (Artificial Intelligence):** Inteligencia artificial.
- **NVR (Network Video Recorder):** Grabador de video en red.
- **HD (High Definition):** Alta definición.
- **LAN (Local Area Network):** Red de área local.
- **UPS (Uninterruptible Power Supply):** Fuente de alimentación ininterrumpida.
- **CV (Computer Vision):** Visión por computadora.
- **FPS (Frames Per Second):** Cuadros por segundo.
- **GS1:** Organización internacional para el desarrollo de estándares de identificación de productos.
- **LOT:** Número de lote, usado para la trazabilidad de productos en manufactura.

1.5 Referencias

Referencia	Titulo	Ruta	Fecha	Autor
ISO/IEC 18004	Information technology — Automatic identification and data capture techniques	[Ruta]	2015	Joint Technical Committee ISO/IEC JTC 1, Subcommittee SC 31:
ISO/IEC 27001:2013	Norma ISO/IEC 27001 - Gestión de Seguridad de la Información		2013	ISO
ISO/IE15416	Petroleum and natural gas industries — Materials for use in H2S-containing environments in oil and gas production		2020	Comité Técnico Conjunto ISO/IEC JTC 1, Subcomité SC 31
ISO / CEI TR 29158 (AIM DPM)	Information technology — Automatic identification and data capture techniques — Direct Part Mark (DPM) quality guideline		2011	International Organization for Standardization (ISO) y la International Electrotechnical Commission (IEC),
ISO / IEC 15415.	Information technology — Automatic identification and data capture techniques — Bar code symbol print quality test specification — Two-dimensional symbols		2011-04 (abril de 2011)	International Organization for Standardization (ISO) y la International Electrotechnical Commission (IEC)



1.6 Resumen

Este documento presenta el desarrollo de un sistema automatizado de control de calidad enfocado en la detección, decodificación y evaluación en tiempo real de códigos QR y DataMatrix, además de la extracción de texto mediante OCR (Reconocimiento Óptico de Caracteres). La implementación de esta solución tiene como objetivo optimizar los procesos de control de calidad y trazabilidad en la producción de dispositivos médicos, específicamente en el empaque de sábanas médicas. Al automatizar este proceso, se estima un ahorro anual de \$14,000 al eliminar la necesidad de dos operadores dedicados a tareas de control de calidad ineficientes.

2 Descripción general

2.1 Perspectiva del producto

El sistema automatizado de control de calidad es un componente dentro de la línea de producción de sábanas médicas. Su función principal es verificar, mediante visión artificial y lectura de códigos, que la información impresa en los empaques sea correcta antes de su distribución.

Este sistema se integra con las máquinas de empaque y los sistemas de gestión de producción, generando alertas en caso de errores y evitando que productos defectuosos lleguen al mercado. Además, permite registrar los datos de inspección para fines de trazabilidad y auditorías.

Su diseño modular lo hace adaptable a otras líneas de producción, mejorando la eficiencia, reduciendo errores humanos y garantizando el cumplimiento de estándares de calidad en la industria médica.

2.2 Funcionalidad del producto

a. Detección y Decodificación en Tiempo Real

El sistema utiliza cámaras de alta resolución para capturar imágenes de los empaques mientras se imprimen. Empleando algoritmos de visión por computadora, detecta y decodifica códigos QR y DataMatrix al instante.

b. Reconocimiento Óptico de Caracteres (OCR)

Se aplica OCR para extraer y validar el texto impreso alrededor de los códigos, asegurando que datos como el número de lote y la fecha de caducidad sean correctos y legibles.

c. Integración con la Línea de Producción

El sistema está diseñado para comunicarse con las máquinas de producción. Si se detecta un defecto, envía una señal para detener la producción inmediatamente, evitando que productos defectuosos sean empaquetados y distribuidos.



d. Interfaz de Usuario

Se proporciona una interfaz web intuitiva donde los operadores pueden visualizar en tiempo real el proceso de detección y decodificación, recibir alertas y acceder a reportes de calidad.

Implementación Técnica

1. Arquitectura del Sistema

El sistema se compone de módulos interconectados que manejan la captura de video, procesamiento de imágenes, decodificación de códigos, OCR y comunicación con las máquinas de producción.

2. Procesamiento de Imágenes

Preprocesamiento: Se aplica filtrado y ajuste de contraste para mejorar la calidad de las imágenes antes de la decodificación y OCR.

Detección de Códigos: Utilizando OpenCV y librerías especializadas como pyzbar y pylibdmtx para códigos QR y DataMatrix respectivamente.

Extracción de Texto: EasyOCR procesa las imágenes preprocesadas para extraer texto, que luego se compara con los datos esperados.

3. Algoritmo de Decisión

El sistema compara los datos decodificados y extraídos con los valores esperados. Si hay discrepancias, se activa una alerta y se envía una señal para detener la máquina de producción.

4. Comunicación con la Máquina de Producción

Se implementa una API o protocolo de comunicación que permite al sistema enviar comandos a la maquinaria existente, integrándose sin interrumpir el flujo de trabajo actual.

Tipo de usuario	Técnicos de Soporte y Operadores
Formación	Formación técnica en electrónica
Habilidades	Instalación y mantenimiento de equipos
Actividades	Instalación de equipos de seguridad

Tipo de usuario	Desarrollador web
Formación	Ingeniero en Mecatrónica
Habilidades	Desarrollar páginas web y automatización
Actividades	Automatización del proceso

2.3 Restricciones

El desarrollo del sistema automatizado de control de calidad para la línea de producción de sábanas médicas estará sujeto a las siguientes restricciones:

Compatibilidad de hardware: El sistema debe adaptarse al espacio físico y a la maquinaria existente en la línea de producción, sin necesidad de modificarla de forma significativa.



Velocidad de procesamiento: El sistema debe analizar cada empaque en tiempo real, sin generar retrasos en el proceso de empaclado.

Lenguaje de programación: Se usará preferiblemente Python por su compatibilidad con bibliotecas de visión artificial (como OpenCV) y facilidad de integración con hardware.

Sistema operativo: El sistema debe correr en una distribución estable de Linux o Windows, según los requerimientos del entorno de fábrica.

Normativas del sector salud: El sistema debe cumplir con los estándares de calidad y trazabilidad exigidos en la industria médica, incluyendo regulaciones sobre control de lote y registro.

Interfaz de usuario: Debe ser intuitiva y adecuada para operadores sin formación técnica avanzada.

Presupuesto limitado: Se utilizarán sensores y cámaras de visión industrial de costo medio, evitando componentes innecesariamente costosos.

2. Lenguaje de programación

Para el desarrollo del sistema automatizado de control de calidad se emplearán los siguientes lenguajes de programación, seleccionados por su versatilidad, eficiencia y compatibilidad con bibliotecas de procesamiento de imágenes y hardware:

Python: Será el lenguaje principal utilizado para el procesamiento de imágenes, control del sistema y lógica de detección de errores en los empaques. Se emplearán bibliotecas como OpenCV, NumPy y posiblemente TensorFlow para tareas de visión e inteligencia artificiales.

C/C++: Se utilizará en caso de requerirse control directo de hardware o procesamiento a bajo nivel para mejorar la velocidad de respuesta del sistema en tiempo real.

HTML/CSS/JavaScript (opcional): En caso de desarrollar una interfaz gráfica o panel de monitoreo accesible vía web, estos lenguajes se utilizarán para la parte visual y de interacción.

La elección de estos lenguajes permite desarrollar un sistema robusto, adaptable a futuros requerimientos y optimizado para operar en ambientes industriales exigentes.

2.4 Suposiciones y dependencias

El desarrollo y correcto funcionamiento del sistema automatizado de control de calidad depende de ciertas suposiciones y condiciones externas. Si alguno de estos factores cambia, podrían requerirse ajustes en los requisitos o en la arquitectura del sistema. Entre las principales suposiciones y dependencias se encuentran:

Se asume que el entorno industrial donde se instalará el sistema contará con una fuente de energía eléctrica estable y protección contra sobrecargas.

Se asume la disponibilidad de cámaras de alta resolución compatibles con las bibliotecas de visión por computadora utilizadas (como OpenCV). El sistema dependerá de la instalación de un sistema operativo compatible, como Linux o Windows, según la plataforma elegida para el procesamiento.

Se supone que habrá acceso constante a una red local (LAN) para la transmisión de datos y posible integración con otros sistemas de la planta. El hardware empleado (como procesadores, microcontroladores o GPU) debe cumplir con los requisitos mínimos para ejecutar los algoritmos de procesamiento de imágenes en tiempo real.

Se asume que el personal responsable del monitoreo tendrá conocimientos básicos para operar la interfaz del sistema y reportar incidencias.



Estas condiciones deben mantenerse estables para garantizar la operatividad del sistema y evitar modificaciones costosas o retrasos en el desarrollo.

2.5 del sistema

1 Visión por Computadora y Procesamiento de Imágenes

OpenCV: Biblioteca de código abierto que permite el procesamiento en tiempo real de imágenes y videos. Se utiliza para la detección y decodificación de códigos QR y DataMatrix, así como para preprocesar imágenes antes de aplicar OCR.

2. Reconocimiento Óptico de Caracteres (OCR)

EasyOCR: Framework de OCR que soporta múltiples idiomas y es eficiente en la extracción de texto de imágenes complejas. Permite validar que el texto impreso coincide con la información esperada.

3. Desarrollo Web y APIs

Flask: Microframework de Python utilizado para crear una interfaz web que permite interactuar con el sistema, visualizar el video en tiempo real y mostrar resultados de decodificación y OCR.

JavaScript y jQuery: Para manejar interacciones en el frontend, actualizaciones en tiempo real y comunicación asíncrona con el backend.

4. Multithreading y Procesamiento en Tiempo Real

Threading en Python: Para manejar la captura y procesamiento de video en tiempo real sin bloquear la interfaz de usuario.

Queues: Utilizadas para gestionar eficientemente los frames capturados y procesados, asegurando que el sistema pueda manejar la carga en tiempo real.

Descripción del Sistema

1. Contexto Operativo

En la línea de producción de sábanas médicas, se imprime información crítica en cada empaque, incluyendo el número de lote, fecha de caducidad y códigos GTIN. Esta información es esencial para la trazabilidad y cumplimiento de regulaciones en la industria médica.

Datos Impresos:

LOT 24KGR005
2029-10-01
Use By



(01)00192253020741
(17)291001
(10)24KGR005

Número de Lote (LOT): Ejemplo, LOT 24KGR005.

Fecha de Caducidad: Ejemplo, 2029-10-01 (1 de octubre de 2029).

GTIN y Códigos GS1:

- (01)00192253020741: Identificador global único del producto.
- (17)291001: Fecha de caducidad en formato YYMMDD (29 significa 2029, 1001 es 1 de octubre).

Implementación Técnica

1. Arquitectura del Sistema



El sistema se compone de módulos interconectados que manejan la captura de video, procesamiento de imágenes, decodificación de códigos, OCR y comunicación con las máquinas de producción.

2. Procesamiento de Imágenes

Preprocesamiento: Se aplica filtrado y ajuste de contraste para mejorar la calidad de las imágenes antes de la decodificación y OCR.

Detección de Códigos: Utilizando OpenCV y librerías especializadas como pyzbar y pylibdmtx para códigos QR y DataMatrix respectivamente.

Extracción de Texto: EasyOCR procesa las imágenes preprocesadas para extraer texto, que luego se compara con los datos esperados.

3. Algoritmo de Decisión

El sistema compara los datos decodificados y extraídos con los valores esperados. Si hay discrepancias, se activa una alerta y se envía una señal para detener la máquina de producción.

4. Comunicación con la Máquina de Producción

Se implementa una API o protocolo de comunicación que permite al sistema enviar comandos a la maquinaria existente, integrándose sin interrumpir el flujo de trabajo actual.

3 Requisitos específicos

3.1.1 3.1 Requisitos Funcionales

RF1. El sistema debe capturar imágenes en tiempo real de los empaques que pasan por la línea de producción.

RF2. El sistema debe detectar errores visuales en los empaques, como doblez, mal sellado, etiquetas mal colocadas o faltantes.

RF3. El sistema debe clasificar los empaques como "Correcto" o "Defectuoso" según los criterios configurados.

RF4. El sistema debe mostrar los resultados en una interfaz gráfica amigable al usuario.

RF5. El sistema debe emitir una señal (por ejemplo, mediante una salida digital) cuando se detecte un empaque defectuoso para activar una acción correctiva (como retirar el producto).

RF6. El sistema debe almacenar un registro con la fecha, hora e imagen del empaque defectuoso para fines de trazabilidad.

RF7. El sistema debe permitir a los operadores configurar los parámetros de detección (umbrales, tolerancias, tipo de error a detectar).

RF8. El sistema debe poder operar de forma continua durante la jornada laboral sin necesidad de reinicios constantes.

3.1.2 3.2 Requisitos No Funcionales

RNF1. El sistema debe ser capaz de procesar cada imagen en menos de 1 segundo para evitar retrasos en la línea de producción.

RNF2. La interfaz de usuario debe estar en idioma español y ser comprensible para operarios con conocimientos técnicos básicos.

RNF3. El sistema debe ser compatible con sistemas operativos Windows 10 y/o Linux.

RNF4. El sistema debe poder funcionar con cámaras industriales estándar de al menos 5 MP de resolución.

RNF5. La base de datos de imágenes y registros debe poder almacenar información por al menos 30 días antes de requerir limpieza o respaldo.

RNF6. El sistema debe tener una tasa mínima de precisión del 95% en la detección de errores para considerarse eficiente.



3.1.3 3.3 Restricciones Técnicas

Lenguaje de programación principal: Python.

Uso de bibliotecas de visión por computadora como OpenCV.

Hardware requerido: cámara industrial, microcomputadora (por ejemplo, Raspberry Pi o equivalente), pantalla de visualización.

Beneficios y Ahorros

1. Ahorro de Costos

Reducción de Personal: Al automatizar el proceso, se eliminan dos posiciones de operadores, ahorrando aproximadamente \$14,000 al año.

Disminución de Desperdicios: Al detectar errores en tiempo real, se evita la producción de lotes defectuosos, reduciendo costos por materiales y reprocesos.

2. Mejora en la Eficiencia

Velocidad de Detección: El sistema opera en tiempo real, permitiendo respuestas inmediatas ante cualquier error.

Consistencia: La automatización asegura una inspección constante y libre de fatiga, a diferencia del control manual.

3. Cumplimiento Normativo

Trazabilidad Mejorada: Al asegurar la exactitud de los códigos y datos impresos, se mejora la trazabilidad del producto, cumpliendo con regulaciones estrictas en la industria médica.

Documentación Automatizada: Se generan reportes automáticos de control de calidad, facilitando auditorías y cumplimiento regulatorio.

Aplicaciones en la Industria de Dispositivos Médicos

Control de Calidad en Empaque: Asegura que los productos cumplen con los estándares antes de ser distribuidos.

Prevención de Retiradas de Productos: Al minimizar errores en la información impresa, se reduce el riesgo de retiradas costosas y dañinas para la reputación.

Integración con Sistemas de Gestión: Los datos recopilados pueden integrarse con sistemas ERP y de gestión de inventario, mejorando la planificación y logística.

Consideraciones de Implementación

1. Compatibilidad y Flexibilidad

El sistema está diseñado para adaptarse a diferentes tipos de maquinaria y líneas de producción, facilitando su implementación sin necesidad de cambios significativos en la infraestructura existente.

2. Escalabilidad

La arquitectura modular permite expandir el sistema a otras líneas de producción o plantas, compartiendo datos y aprendiendo de múltiples fuentes.



3. Seguridad y Confiabilidad

Se garantiza que el sistema cumple con los estándares de seguridad informática, protegiendo tanto la información como los procesos operativos de la empresa.

Tecnologías Utilizadas

1. Visión por Computadora y Procesamiento de Imágenes

OpenCV: Biblioteca de código abierto que permite el procesamiento en tiempo real de imágenes y videos. Se utiliza para la detección y decodificación de códigos QR y DataMatrix, así como para preprocesar imágenes antes de aplicar OCR.

2. Reconocimiento Óptico de Caracteres (OCR)

EasyOCR: Framework de OCR que soporta múltiples idiomas y es eficiente en la extracción de texto de imágenes complejas. Permite validar que el texto impreso coincide con la información esperada.

3. Desarrollo Web y APIs

Flask: Microframework de Python utilizado para crear una interfaz web que permite interactuar con el sistema, visualizar el video en tiempo real y mostrar resultados de decodificación y OCR.

JavaScript y jQuery: Para manejar interacciones en el frontend, actualizaciones en tiempo real y comunicación asíncrona con el backend.

4. Multithreading y Procesamiento en Tiempo Real

Threading en Python: Para manejar la captura y procesamiento de video en tiempo real sin bloquear la interfaz de usuario.

Queues: Utilizadas para gestionar eficientemente los frames capturados y procesados, asegurando que el sistema pueda manejar la carga en tiempo real.

Objetivos

- **Desarrollar un sistema capaz de detectar y decodificar códigos QR y DataMatrix en tiempo real durante el proceso de producción.**
- **Implementar OCR para extraer y verificar el texto circundante a los códigos, asegurando la exactitud de la información crítica del producto.**
- **Automatizar el proceso de control de calidad, reduciendo la dependencia de operadores humanos y minimizando errores.**
- **Integrar el sistema con las máquinas de producción para detener automáticamente el proceso en caso de detectar defectos, evitando desperdicios y costos adicionales.**



3.2 Requisitos comunes de las interfaces

3.2.1 Interfaces de usuario

El sistema contará con una interfaz gráfica intuitiva accesible desde un monitor táctil o computadora conectada a la red local. Los requisitos clave son:

Diseño responsivo: adaptable a diferentes resoluciones de pantalla.

Lenguaje del sistema: interfaz en español, con opción a inglés.

Estilo visual: uso de colores neutros (grises y azules) que no distraigan al operador.

Pantallas principales:

- **Panel de monitoreo en tiempo real:** muestra la imagen en vivo de los empaques procesados.
- **Panel de estadísticas:** muestras métricas de detección, tasas de error y desempeño del sistema.
- **Historial de errores:** acceso a registros de empaques defectuosos detectados.
- **Opciones de configuración:** menú para calibrar parámetros de detección, umbrales de error, y zonas de análisis.

Permisos de usuario: diferenciación entre operadores, técnicos y administradores, con niveles de acceso definidos.

3.2.2 Interfaces de hardware

El sistema se conectará con varios componentes físicos que permiten la detección y procesamiento. Estas interfaces incluyen:

Cámara de visión artificial (alta resolución, mínimo 1080p, con enfoque automático).

- Protocolo de conexión: USB 3.0 o GigE.
- Montura ajustable para calibración de ángulos de captura.

Iluminación LED: uniforme, sin parpadeos, para garantizar detección precisa.

Sensor de proximidad/posición: detecta el paso de un empaque para sincronizar la captura.

Microcontrolador (Arduino o similar): para control de actuadores (expulsión de empaques defectuosos).

Actuadores: mecanismo para retirar productos no conformes de la línea de producción.

UPS (fuente de alimentación ininterrumpida): garantiza funcionamiento continuo ante cortes de energía.

El sistema automatizado de control de calidad mediante visión por computadora utiliza principalmente el **protocolo de comunicación serial por USB** para establecer la conexión entre el procesador (como un PC o una Raspberry Pi) y el microcontrolador (como un Arduino o ESP32). Esta comunicación permite enviar instrucciones en tiempo real al mecanismo encargado de rechazar los empaques defectuosos detectados por la cámara.

Adicionalmente, se emplea el **protocolo TCP/IP** para transmitir datos hacia la interfaz de usuario o una base de datos, permitiendo el monitoreo remoto, el almacenamiento de resultados y la generación de reportes. Esta combinación de protocolos garantiza una comunicación confiable, rápida y adecuada tanto para el control físico del sistema como para la visualización y análisis de los datos obtenidos.



El sistema automatizado de control de calidad cuenta con varias características de configuración esenciales para garantizar su funcionamiento eficiente y adaptable a distintos entornos de producción:

Parámetros ajustables del sistema de visión:

Se pueden configurar aspectos como la resolución de la cámara, el enfoque, el brillo, la sensibilidad y los umbrales de detección de errores en los empaques. Esto permite adaptar el sistema a diferentes condiciones de iluminación y tipos de producto.

Velocidad de procesamiento:

El software permite ajustar la frecuencia de captura de imágenes (FPS) y el tiempo de respuesta para actuar sobre empaques defectuosos, lo cual es crucial para líneas de producción con distintas velocidades.

Conexión y puertos de comunicación:

Se pueden configurar los puertos COM utilizados para la comunicación entre el sistema de visión (PC/Raspberry Pi) y el microcontrolador (Arduino o ESP32). También se configuran parámetros como el baud rate (9600, 115200, etc.) para la comunicación serial.

Dirección IP y parámetros de red:

Para el envío de datos a través de la red, el sistema permite configurar direcciones IP, máscaras de subred y puertas de enlace, facilitando la integración del sistema en una red local o en la nube para monitoreo remoto.

Zonas de detección personalizables:

La interfaz gráfica del software permite al usuario definir zonas específicas en las imágenes donde se debe aplicar la inspección, adaptándose a diferentes tamaños y formas de empaques.

Sistema de notificaciones y alertas:

El sistema puede configurarse para emitir notificaciones visuales, sonoras o digitales (por ejemplo, vía correo electrónico o mensajes en pantalla) en caso de detectar empaques defectuosos repetidamente o fallas en el sistema.

Modo de operación manual o automático:

El sistema puede operar de forma automática (procesamiento continuo) o manual (activación de prueba), lo que facilita tareas de calibración, mantenimiento o pruebas de funcionamiento.

3.2.3 Interfaces de software

Página web <https://mecavision.b12sites.com/>

El sistema debe integrarse con otras soluciones de software que faciliten la operación y el análisis de datos :

OpenCV : biblioteca de visión por computadora utilizada para la detección de errores.

Propósito del interfaz : procesamiento de imágenes en tiempo real.

Formato : entrada en tiempo real de imágenes RAW o JPEG, salida en forma de alertas, coordenadas y anotaciones en la imagen.

Base de datos (MySQL o SQLite) :

Propósito : almacenamiento de registros históricos de inspecciones.



Definición : cada registro contiene fecha, hora, ID del producto, resultado de inspección y causa del rechazo (si aplica).

Panel web (opcional) : para visualizar estadísticas de producción desde dispositivos móviles o PC conectados a la red.

3.2.4 Interfaces de comunicación

El sistema requiere comunicación interna y externa, cumpliendo los siguientes protocolos:

Comunicación entre módulos del sistema:

Protocolo: MQTT o serial USB para comunicación entre el módulo de procesamiento y el microcontrolador.

Tiempo de respuesta bajo (latencia < 100 ms).

Red de datos local (LAN):

Permite la transmisión de imágenes y resultados de análisis hacia el servidor o panel web.

Protocolo: TCP/IP.

Acceso remoto (si aplica):

Protocolo: HTTP/HTTPS para consultar reportes desde otras terminales conectadas a la misma red.

Seguridad: autenticación con usuario y contraseña, y posible cifrado de datos mediante SSL.

Exportación de datos:

Formatos compatibles: CSV, JSON, PDF (para reportes automáticos).

Vía: conexión directa o sincronización por FTP a un servidor.

3.3 Requisitos funcionales

Los requisitos funcionales de este sistema definen las acciones esenciales que debe realizar el software y hardware involucrado, desde la adquisición de datos hasta la generación de resultados, garantizando la detección precisa de errores en empaques.

3.3.1 Requisito funcional 1:

Captura y procesamiento de imágenes

El sistema debe capturar imágenes en tiempo real desde la cámara de visión conectada. Las imágenes deben procesarse utilizando técnicas de visión por computadora (OpenCV o similar) para detectar defectos visibles (como etiquetas mal colocadas, empaques abiertos, mal alineados, etc.).

El sistema deberá validar que las entradas (imágenes) tengan el formato y resolución adecuados antes de ser procesadas.

En caso de entrada inválida (imagen dañada o fuera de foco), el sistema debe descartar la imagen y registrar el error en un log.



3.3.2 Requisito funcional 2:

Clasificación de empaques

El sistema debe determinar si un empaque es **válido** o **defectuoso** según parámetros configurables.

Las condiciones de validación serán establecidas previamente por el operador (posición de la etiqueta, simetría, integridad del empaque).

Una vez procesada la imagen, el resultado debe clasificarse automáticamente.

3.3.3 Requisito funcional 3:

Activación del actuador

Si se detecta un empaque defectuoso, el sistema debe enviar una señal al microcontrolador para que active un actuador (por ejemplo, un brazo robótico o pistón neumático) que retire el empaque defectuoso de la línea.

En caso de fallo en la comunicación con el actuador, el sistema debe emitir una alerta visual y sonora.

3.3.4 Requisito funcional 4:

Almacenamiento de resultados

El sistema debe almacenar en una base de datos local o en la nube los resultados del análisis de cada empaque:

- Fecha y hora
- Imagen capturada (si es necesario)
- Resultado de la inspección (válido/defectuoso)
- Tipo de defecto identificado

Esta base de datos será utilizada para generar reportes de desempeño del sistema.

3.3.5 Requisito funcional 5:

Interfaz gráfica de usuario

El sistema debe proporcionar una interfaz visual que muestre:

- Imágenes en tiempo real
- Resultados de la inspección
- Contadores de empaques procesados y defectuosos
- Alertas o fallas

La interfaz debe permitir cambiar parámetros del sistema (umbral de validación, sensibilidad, configuración del hardware, etc.).

3.3.6 Requisito funcional 6:

Manejo de errores

El sistema debe manejar errores como:

- Fallo de conexión con la cámara
- Desbordamiento de búfer de imágenes
- Caídas de comunicación serial

Debe intentar reconectar automáticamente o notificar al operador si el problema persiste.

3.4 Requisitos no funcionales

3.4.1 Requisitos de rendimiento

Velocidad de Detección: El sistema opera en tiempo real, permitiendo respuestas inmediatas ante cualquier error.



Consistencia: La automatización asegura una inspección constante y libre de fatiga, a diferencia del control manual.

Capacidad de Dispositivos: Soporta hasta 500 terminales conectados simultáneamente (cámaras, sensores, drones).

Usuarios Simultáneos: Soporta al menos 200 usuarios conectados al mismo tiempo en plataformas web y móviles sin degradación de rendimiento.

Medición: El rendimiento se evalúa en función de la fluidez del sistema y la ausencia de pérdidas o demoras.

3.4.2 Seguridad

Se garantiza que el sistema cumple con los estándares de seguridad informática, protegiendo tanto la información como los procesos operativos de la empresa.

El sistema de Mecavision debe ser capaz de soportar una carga adecuada y ofrecer un rendimiento eficiente en condiciones normales y durante picos de uso. Los requisitos relacionados con la carga deben estar bien definidos para garantizar una experiencia de usuario fluida, sin demoras, y con una alta disponibilidad del sistema.

Requisito: El sistema debe ser capaz de manejar hasta 500 terminales (dispositivos como cámaras, sensores y drones) conectados simultáneamente.

Descripción: Estos terminales pueden incluir cámaras IP, sensores de movimiento, sensores de temperatura, drones y otros dispositivos conectados al sistema. La comunicación con estos dispositivos debe ser continua, garantizando la sincronización y respuesta en tiempo real.

Medición: Se debe medir la capacidad del sistema para gestionar la comunicación con estos dispositivos sin pérdidas de datos ni demoras significativas.

3.4.3 Fiabilidad

La fiabilidad del sistema es un aspecto fundamental para garantizar que Mecavision funcione de manera continua y sin fallos, minimizando el tiempo de inactividad y asegurando que se puedan acceder a las funcionalidades críticas de manera consistente. La fiabilidad se medirá en términos de tiempo entre fallos y el total de incidentes permisibles, y se establecerán umbrales específicos para asegurar que el sistema se mantenga operativo con el mínimo de interrupciones.

- **MTBF (Tiempo Medio Entre Fallos):** Mínimo de 1000 horas de operación continua sin errores.
- **MTTR (Tiempo Medio de Recuperación):** Tiempo máximo de recuperación ante fallos menor a 2 horas.

3.4.4 Disponibilidad

La disponibilidad del sistema es un factor crítico para Mecavision, ya que el sistema debe estar operando de manera continua, minimizando el tiempo de inactividad y asegurando que puedan acceder a las funcionalidades esenciales del sistema en todo momento. La disponibilidad se expresa generalmente en un porcentaje del tiempo total en el que el software debe estar operativo.

1. Disponibilidad General del Sistema



Requisito 1.1: Disponibilidad mínima del 99.9%.

Descripción: El sistema debe estar operativo el 99.9% del tiempo, lo que equivale a un máximo de 43 minutos de inactividad al mes o 8.76 horas de inactividad al año.

Medición: Se debe monitorizar continuamente el tiempo de inactividad para asegurar que no supere este umbral, incluyendo las interrupciones no planificadas y los mantenimientos programados.

3.4.5 Mantenibilidad

El mantenimiento del sistema de Mecavisión es fundamental para asegurar su correcto funcionamiento a largo plazo, evitando fallos y garantizando la continuidad del servicio. A continuación, se detallan los tipos de mantenimiento necesarios, quién debe realizarlos, y cuándo deben llevarse a cabo.

1. Mantenimiento Correctivo

Descripción: Este tipo de mantenimiento se realiza para corregir errores o fallos en el sistema que afectan su funcionalidad o rendimiento. Es necesario cuando un incidente ocurre o se identifica un fallo en el sistema que requiere ser resuelto.

Responsable: Equipo de soporte técnico y desarrolladores del sistema.

Frecuencia: Este mantenimiento se realiza de manera ad hoc en respuesta a incidentes o fallos no previstos. Debe ejecutarse tan pronto como se identifique un problema.

Actividades Típicas:

- Diagnóstico y resolución de fallos del sistema (errores en la base de datos, fallos en la red, problemas con los sensores o drones).
- Corrección de errores de software que afectan la experiencia del usuario.
- Restablecimiento de la funcionalidad tras un fallo de hardware o software.

2. Mantenimiento Preventivo

Descripción: Este mantenimiento busca prevenir problemas y fallos en el futuro mediante la revisión periódica del sistema, actualización de componentes, y limpieza de recursos.

Responsable: Administradores de sistema y personal de infraestructura.

Frecuencia: Realización mensual o trimestral dependiendo de la criticidad de los componentes. Este mantenimiento debe estar programado para evitar interrupciones y minimizar el impacto en los usuarios.

Actividades Típicas:

Actualización de sistemas operativos, parches de seguridad y bibliotecas de software.
Revisión y optimización de la base de datos.
Reemplazo de hardware obsoleto o con alta probabilidad de fallo.
Realización de pruebas de rendimiento y capacidad.

3.4.6 Portabilidad

El sistema Mecavision debe ser diseñado para facilitar su migración o adaptación a diferentes plataformas o entornos. Esto implica que el código y los componentes del sistema deben ser lo menos dependientes posible de plataformas específicas



para permitir su uso en distintos servidores y sistemas operativos. Los puntos clave incluyen:

Dependencia del servidor: El sistema debe ser independiente de un servidor específico, lo que facilita su despliegue en servidores de diferentes proveedores y configuraciones.

Código independiente de la plataforma: Se deben utilizar lenguajes de programación que favorezcan la portabilidad, como Java o Python, que son ampliamente compatibles con diferentes plataformas.

Uso de compiladores y plataformas universales: Mecavision debe ser desarrollado utilizando compiladores y herramientas de desarrollo que sean compatibles con múltiples plataformas y sistemas operativos, asegurando que el sistema pueda ser portado sin dificultades a distintos entornos.

Independencia del sistema operativo: El sistema debe funcionar de manera eficiente en los sistemas operativos más utilizados (Windows, Linux, macOS), minimizando las dependencias específicas del SO.

De esta forma, el sistema Mecavision garantiza su capacidad de ser implementado en diferentes infraestructuras tecnológicas sin problemas.

Conclusiones

La implementación de este sistema automatizado de control de calidad ofrece una solución integral que aborda los desafíos actuales en la producción de dispositivos médicos. Al combinar tecnologías avanzadas de visión por computadora, OCR y automatización, se logra una mejora significativa en la eficiencia operativa, reducción de costos y aseguramiento de la calidad.

Esta iniciativa no solo proporciona beneficios económicos tangibles, como el ahorro anual de \$14,000, sino que también fortalece la posición competitiva de la empresa en el mercado al ofrecer productos de mayor calidad y confiabilidad.

1. Recomendaciones y Próximos Pasos

2. **Validación en Entorno Controlado:** Realizar pruebas piloto en una línea de producción para ajustar y optimizar el sistema antes de una implementación completa.
3. **Capacitación al Personal:** Involucrar a los op
4. **Monitoreo y Mejora Continua:** Establecer indicadores clave de desempeño (KPI) para monitorear la eficacia del sistema y realizar mejoras continuas.
5. **Exploración de Nuevas Tecnologías:** Considerar la integración de inteligencia artificial y aprendizaje automático para mejorar la precisión y adaptabilidad del sistema.

6. Referencias

- **Regulaciones de la Industria Médica:** Normativas de la FDA y estándares ISO aplicables al control de calidad y trazabilidad en dispositivos médicos.
- **Tecnologías Utilizadas:**



- **OpenCV Documentation:** <https://opencv.org/>
- **EasyOCR GitHub:** <https://github.com/JaiedAI/EasyOCR>
- **Flask Framework:** <https://flask.palletsprojects.com/>

4 Apéndices

Este apartado incluye información adicional relevante para la comprensión y desarrollo del sistema Mecavision, pero que no forma parte directa del Documento de Especificación de Requisitos del Software (DEP). Se presentan aquí elementos complementarios que forman parte de un ecosistema más amplio dentro del cual se integra Mecavision.

4.1 Integración con otros sistemas

Mecavision puede formar parte de un sistema mayor de automatización industrial o de monitoreo de calidad en procesos de manufactura. Entre los posibles sistemas asociados se incluyen:

- Sistemas SCADA (Supervisory Control and Data Acquisition).
- ERP (Enterprise Resource Planning) para sincronización con gestión de producción.
- Bases de datos centralizadas para almacenamiento histórico de fallos detectados.

4.2 Herramientas utilizadas durante el desarrollo

- Lenguaje de programación: Python y JavaScript.
- Frameworks: TensorFlow para visión por computadora, Django para backend, React para frontend.
- Bases de datos: PostgreSQL.
- Plataforma de despliegue: Docker sobre servidores Linux.

4.3 Estándares y normativas de referencia

- ISO/IEC 25010: Modelo de calidad del software.
- ISO 9241: Ergonomía de la interacción persona-sistema.
- Normativas de seguridad industrial aplicables al entorno donde se instalará Mecavision.

4.4 Glosario de términos

- MTBF: Tiempo medio entre fallos.
- MTTR: Tiempo medio para recuperación.
- IA: Inteligencia Artificial.
- Sistema crítico: Componente cuya falla compromete el funcionamiento general del sistema.



GitHub: <https://github.com/MacroVirux/MecaVision>

Página web: <https://mecavision.b12sites.com/>