



UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CARRERA: INGENIERÍA MECATRÓNICA

INFORME FINAL DEL TRABAJO DE INTEGRACIÓN

CURRICULAR, MODALIDAD PRESENCIAL.

TEMA:

**“SISTEMA AUTOMATIZADO DE CALIFICACIÓN DE
EVALUACIONES DE BASE ESTRUCTURADA EN PAPEL USANDO
VISIÓN ARTIFICIAL”**

Trabajo de titulación previo a la obtención del título de ingeniero en mecatrónica.

**Línea de investigación: Gestión, producción, productividad, innovación y desarrollo
socio económico.**

AUTOR(A):

Ricardo Vladimir Calderón Cevallos

DIRECTOR(A):

MSc. Cosme Damián Mejía Echeverría.

Ibarra, julio 2024



UNIVERSIDAD TÉCNICA DEL NORTE

DIRECCIÓN DE BIBLIOTECA

1. IDENTIFICACIÓN DE LA OBRA

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

DATOS DE CONTACTO			
CÉDULA DE IDENTIDAD:	0401779590		
APELLIDOS Y NOMBRES:	Calderón Cevallos Ricardo Vladimir		
DIRECCIÓN:	Barrio El Olivo.		
EMAIL:	rvcalderonc@utn.edu.ec		
TELÉFONO FIJO:	N/A	TELÉFONO MÓVIL:	0999569566

DATOS DE LA OBRA	
TÍTULO:	SISTEMA AUTOMATIZADO DE CALIFICACIÓN DE EVALUACIONES DE BASE ESTRUCTURADA EN PAPEL USANDO VISIÓN ARTIFICIAL.
AUTOR (ES):	Calderón Cevallos Ricardo Vladimir
FECHA: DD/MM/AAAA	24/07/2024
SOLO PARA TRABAJOS DE GRADO	
PROGRAMA:	<input checked="" type="checkbox"/> GRADO <input type="checkbox"/> POSGRADO
TÍTULO POR EL QUE OPTA:	Ingeniero en Mecatrónica
ASESOR /DIRECTOR:	MSc. Mejía Echeverría Cosme Damián

2. CONSTANCIAS

El autor manifiesta que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto, la obra es original y que es Calderón Cevallos Ricardo Vladimir el titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 24 días del mes de julio del 2024

EL AUTOR:

Firma: 

Nombre: Calderón Cevallos Ricardo Vladimir

**CERTIFICACIÓN DIRECTOR DEL TRABAJO DE INTERGRACIÓN
CURRICULAR**

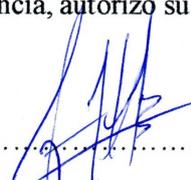
Ibarra, 24 de julio de 2024

MSc. Mejía Echeverría Cosme Damián

DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

CERTIFICA:

Haber revisado el presente informe final del trabajo de titulación, el mismo que se ajusta a las normas vigentes de la Unidad Académica de la Universidad Técnica del Norte; en consecuencia, autorizo su presentación para los fines legales pertinentes.



.....

MSc. Cosme Damián Mejía Echeverría

C.C.: 1002641288

- **APROBACIÓN DEL COMITÉ CALIFICADOR**

El Tribunal Examinador del trabajo de titulación “Sistema automatizado de calificación de evaluaciones de base estructurada en papel usando visión artificial” elaborado por Calderón Cevallos Ricardo Vladimir, previo a la obtención del título del Ingeniero Mecatrónico, aprueba el presente informe de investigación en nombre de la Universidad Técnica del Norte:



.....
MSc. Cosme Damián Mejía Echeverría

C.C.: 1002641288



.....
PhD. David Alberto Ojeda Peña

C.C.: 1757898489

DEDICATORIA

A mi amado Dios

Mi roca eterna, por guiarme en cada paso de este viaje académico y darme la fuerza para perseverar. Te agradezco por ser mi fuente de fortaleza y entendimiento en este logro académico.

Cada página de esta tesis lleva impresa tu gracia y tu bondad infinita. Reconozco que, sin tu amor incondicional y tu guía divina, este logro no sería posible. Tú eres la fuente de todo conocimiento y entendimiento, y en ti encuentro la inspiración para seguir adelante con humildad y gratitud.

En este momento de culminación, dedico este trabajo a ti, mi Señor, como un humilde tributo a tu grandeza y misericordia. Que este trabajo sea un reflejo de tu luz que brilla en mí y una herramienta para servir a otros en tu nombre.

AGRADECIMIENTOS

Hoy, al culminar este importante capítulo de mi vida con la finalización de mi tesis, quiero expresar mi más profundo agradecimiento a mis padres y a Vane. Su constante apoyo, amor incondicional y aliento han sido la luz que me ha guiado a lo largo de este arduo pero gratificante camino académico.

Su sacrificio y dedicación han sido una inspiración constante para mí. Sé que no ha sido fácil, pero nunca dudaron en respaldarme en cada paso del camino. Desde las largas noches de estudio hasta los momentos de celebración, ustedes estuvieron allí, compartiendo cada alegría y cada desafío.

Gracias, por ser mis más grandes defensores, por creer en mí cuando yo mismo dudaba y por ser el pilar de fuerza que siempre necesité. Este logro es tanto suyo como mío, y mi gratitud hacia ustedes es infinita.

RESUMEN EJECUTIVO

Este trabajo de investigación se centra en el desarrollo de un sistema de visión artificial para la calificación automática de exámenes en papel, con el objetivo de mejorar la eficiencia de corrección y reducir la carga de trabajo de los docentes. A través de la identificación y aplicación de técnicas avanzadas de tratamiento de imágenes y reconocimiento de patrones, se diseñó un algoritmo capaz de interpretar respuestas en evaluaciones. Se complementó con el desarrollo de una interfaz de usuario intuitiva, facilitando su uso por parte de docentes y administradores escolares. Se realizaron experimentos utilizando técnicas de procesamiento de imágenes y OCR para interpretar las respuestas marcadas en los exámenes, aplicando un sistema de mallado para la detección precisa de las respuestas.

Los resultados obtenidos demostraron la precisión del sistema, logrando una tasa de éxito del 65,9% en la identificación correcta de las respuestas y reduciendo el tiempo de procesamiento a 2-3 segundos por examen. La implementación de tecnología móvil, en conjunto con la aplicación iVCam, jugó un papel crucial en la optimización del proceso de digitalización de las respuestas. En conclusión, el sistema desarrollado no solo cumple con sus objetivos iniciales, sino que también establece un avance importante en la automatización de la calificación de exámenes de uso personal para docentes, ofreciendo un método más rápido, confiable y objetivo para la evaluación en el ámbito educativo.

Palabras clave: Visión artificial, Calificación Automática, Procesamiento de Imágenes, Reconocimiento Óptico de Caracteres (OCR), Tecnología Educativa.

ABSTRACT

This research work focuses on the development of an artificial vision system for the automatic grading of paper-based exams, aiming to improve correction efficiency and reduce teachers' workload. By identifying and applying advanced image processing and pattern recognition techniques, an algorithm capable of interpreting responses in evaluations was designed. This was complemented by the development of an intuitive user interface, facilitating its use by teachers and school administrators. Experiments were conducted using image processing techniques and OCR to interpret the marked responses in the exams, applying an innovative mesh system for precise response detection.

The results obtained demonstrated the high efficacy and precision of the system, achieving a success rate of 65.9% in the correct identification of responses and reducing processing time to 2-3 seconds per exam. The implementation of mobile technology, in conjunction with the iVCam application, played a crucial role in optimizing the process of digitizing responses. In conclusion, the developed system not only meets its initial objectives but also establishes a significant advancement in the automation of exam grading, offering a faster, more reliable, and objective method for evaluation in the educational field.

Keywords: Artificial Vision, Automatic Grading, Image Processing, Optical Character Recognition (OCR), Educational Technology.

ÍNDICE

DEDICATORIA.....	1
AGRADECIMIENTOS.....	2
RESUMEN EJECUTIVO.....	3
ABSTRACT.....	4
ÍNDICE.....	5
ÍNDICE DE FIGURAS.....	7
ÍNDICE DE TABLAS.....	9
INTRODUCCIÓN.....	10
DESCRIPCIÓN DEL PROBLEMA.....	10
OBJETIVOS.....	11
Objetivo general.....	11
Objetivos específicos.....	11
JUSTIFICACIÓN.....	12
ALCANCE.....	13
1. CAPÍTULO I.....	14
TÉCNICAS EXISTENTES DE TRATAMIENTO DE IMÁGENES O RECONOCIMIENTO DE PATRONES.....	14
1.1 VISION ARTIFICIAL Y TRATAMIENTO DE IMÁGENES.....	14
1.1.1 Definición y Aplicaciones de la Visión Artificial.....	15
1.1.2 Procesamiento de Imágenes: Conceptos Básicos.....	17
1.1.3 Técnicas de Tratamiento de Imágenes para la Detección de Marcas.....	19
1.1.4 Reconocimiento de Patrones en Imágenes: Métodos y Aplicaciones.....	21
1.2 RECONOCIMIENTO ÓPTICO DE CARACTERES (OCR).....	23
1.2.1 Concepto y Funcionamiento del OCR.....	24
1.2.2 Aplicaciones del OCR en la Extracción de Datos Personales.....	24
1.3 METODOLOGÍAS DE EVALUACIÓN DE EXÁMENES.....	25
1.3.1 Evaluación Manual: Proceso y Desafíos.....	25
1.3.2 Evaluación Automática: Beneficios y Limitaciones.....	26
1.3.3 Métodos Existentes de Evaluación Automática.....	26
1.3.4 Evaluación de Exámenes de Opción Múltiple mediante Tratamiento de Imágenes.....	28
1.4 SISTEMAS DE EVALUACIÓN AUTOMÁTICA BASADOS EN VISIÓN ARTIFICIAL.....	28
1.4.1 Sistemas Actuales y su Funcionamiento.....	29
1.4.2 Ventajas y Desventajas de los Sistemas de Evaluación Automática Basados en Visión Artificial.....	29
1.4.3 Desafíos y Oportunidades para Mejorar los Sistemas de Evaluación Automática.....	31
CAPÍTULO II.....	33
MARCO METODOLÓGICO.....	33
2.1 TIPO DE INVESTIGACIÓN.....	33
2.2 NIVEL DE INVESTIGACIÓN.....	34
2.3 METODOLOGÍA.....	34
2.3.1 Fase 1: Adquisición de requerimientos.....	36
2.3.2 Fase 2: Diseño del Sistema.....	40

2.3.3	Fase 3: Desarrollo de la interfaz de usuario	42
2.3.4	Fase 4: Pruebas del Algoritmo	42
CAPÍTULO III		48
RESULTADOS		48
3.1	DETECCIÓN DE CUADRANTES	54
3.2	PRUEBAS OCR	55
3.3	PRUEBAS DE CALIFICACIÓN Y ENTORNOS DE COLOR DE SELECCIÓN.....	57
3.3.1	HSV color.....	57
3.3.2	Mallado y calificación de la evaluación.....	60
3.3.3	Prueba de base de datos.....	63
3.4	RESULTADOS DE LAS PRUEBAS UNITARIAS Y DE INTEGRACIÓN	65
3.5	ANÁLISIS DE RENDIMIENTO Y USABILIDAD.....	67
3.6	DISCUSIÓN GENERAL DE LOS RESULTADOS.....	68
3.6.1	Evaluación de la Eficiencia y Efectividad:.....	68
3.6.2	Experiencia del Usuario y Usabilidad:.....	69
3.6.3	Limitaciones y Áreas de Mejora:	69
3.6.4	Futuros Desarrollos:	69
CAPÍTULO IV		70
CONCLUSIONES Y RECOMENDACIONES		70
4.1	CONCLUSIONES	70
4.2	RECOMENDACIONES	71
BIBLIOGRAFÍA		72
ANEXOS		77
ANEXO 1:	GUION DE ENTREVISTA.....	77
ANEXO 2:	FLUJOGRAMA	79
ANEXO 3:	DIAGRAMA DE BLOQUES	80
ANEXO 4:	ALGORITMO	81
ANEXO 5:	FORMATO DE EVALUACIÓN.....	97

ÍNDICE DE FIGURAS

Figura 1 Filtros en imágenes con visión artificial	15
Figura 2 Seguridad y vigilancia con visión artificial.....	17
Figura 3 <i>Convolucion de imágenes</i>	19
Figura 4. <i>Deteccion de bordes con vision artificial</i>	21
Figura 5 <i>Redes Neuronales Convolucionales (CNN)</i>	23
Figura 6 <i>Lectura OCR basada en Deep Learning</i>	23
Figura 7 <i>Extraccion de datos personales de una tarjeta</i>	25
Figura 8 <i>Evaluación mediante análisis de datos</i>	27
Figura 9 <i>Evaluación de Exámenes de Opción Múltiple mediante Tratamiento de Imágenes</i>	28
Figura 10 Diagrama de Bloques	48
Figura 11. Flujo grama del Proceso General	51
Figura 12. Interfaz gráfica de la ventana principal	53
Figura 13 Interfaz gráfica de la ventana principal	54
Figura 14. Formato de Evaluación	43
Figura 15. Primer cuadrante con su respectivo tratamiento de imágenes	54
Figura 16. Segundo cuadrante detectado con el mallado	55
Figura 17. Detección OCR exitosa.....	56
Figura 18. Error en la detección OCR	56
Figura 19. Opción manual para el ingreso de identificación o cedula	57
Figura 20, Fragmento de configuración para definir colores en HSV.....	58
Figura 21 Segmentacion HSV para determinar rangos de detección	59
Figura 22. Captura del segundo cuadrante con respuestas y mallado	61
Figura 23. Selección de respuestas en la interface	62

Figura 24. Respuesta de la interface la cual hallo el registro en la base de datos	63
Figura 25. Selección de prueba.....	64
Figura 26. Confirmación de datos guardados y visualización en la base de datos.....	65

ÍNDICE DE TABLAS

Tabla 1: Requerimientos de usuario	39
Tabla 2 Requerimientos Operacionales	39
Tabla 3 Requerimientos del sistema.....	40
Tabla 4 Requerimientos de Arquitectura.....	40
Tabla 5. Herramientas de software utilizadas.....	41
Tabla 6. Rúbrica para las pruebas unitarias	44
Tabla 7. Rúbrica de Pruebas de Integración para el Sistema de Evaluación Automática ...	45
Tabla 8. Rúbrica de Verificación y Validación de Requerimientos	46
Tabla 9. Prueba de detecciones de diferentes colores dentro del rango HSV obtenido	59

INTRODUCCIÓN

Descripción del problema

Una de las responsabilidades que tiene el profesor en el aula es la de recopilar evidencias que sustenten el nivel logrado en los aprendizajes esperados de un programa de estudios y asignar la calificación correspondiente a los alumnos [1].

Este proceso de evaluación y calificación es relevante por las consecuencias que tiene, ya sea de manera inmediata o cuando trasciende más allá de un ciclo escolar; porque de ello depende la continuación de estudios en grados superiores [2].

En las unidades educativas, la calificación de las evaluaciones de base estructurada en papel se vuelve un proceso laborioso y propenso a errores humanos por cansancio. Los profesores deben revisar cada examen manualmente, leer las respuestas señaladas por los estudiantes y asignar puntuaciones a cada uno de ellos. Este proceso consume mucho tiempo de los docentes, especialmente cuando en un aula de colegio tienen entre 25 a 30 estudiantes y en la mayoría de los casos tienen un docente a su cargo de varias aulas y materias, trayendo consigo la aparición de varios problemas en el desempeño laboral como docente y de salud [3].

Adquirir un sistema que ayude a los docentes a evaluar a sus alumnos en muchas instituciones se torna imposible debido al factor económico. Esto se debe a la necesidad de contar con una gran infraestructura, incluyendo laboratorios y computadoras, así como una conexión a internet sólida

Objetivos

Objetivo general

Implementar un sistema basado en visión artificial para la calificación automática de evaluaciones de base estructurada en papel.

Objetivos específicos

- Desarrollar un algoritmo de visión artificial capaz de reconocer e interpretar las respuestas de una evaluación.
- Implementar una interfaz de usuario para el sistema que sea fácil de usar por parte de los docentes y los administradores escolares.
- Validar la eficacia y la precisión del sistema en un entorno real de pruebas.

Justificación

El sistema propuesto en el presente trabajo se justifica por varias razones. En primer lugar, la corrección manual de exámenes es una tarea que consume mucho tiempo y recursos para los docentes, que podrían dedicarse mejor a tareas de enseñanza y orientación directa a los estudiantes. Automatizar este proceso podría liberar a los docentes de una parte significativa de su carga de trabajo, permitiéndoles concentrarse más en la interacción directa con los estudiantes y en el diseño de estrategias de enseñanza y aprendizaje más efectivas.

Además, la corrección manual de exámenes puede ser susceptible a inconsistencias y sesgos. Un sistema automatizado basado en reglas claras y definidas puede proporcionar una evaluación más estandarizada y objetiva.

Por último, la salud y el bienestar de los docentes es otra consideración crítica. La corrección manual de exámenes puede conducir a largas horas de trabajo que pueden resultar en estrés, agotamiento y otros problemas de salud. Al reducir esta carga de trabajo, un sistema de evaluación automática puede contribuir al bienestar de los docentes y a la calidad general de la educación.

Por tanto, el desarrollo de un sistema de visión artificial para la evaluación automática de exámenes escritos es relevante y oportuno. No sólo tiene el potencial de mejorar la eficiencia y la equidad en la evaluación, sino también de promover el bienestar de los docentes y la calidad de la enseñanza.

Alcance

El presente estudio se adentra en la creación y análisis de un sistema que incorpora metodologías de visión artificial para la corrección automatizada de exámenes de opción múltiple. Se explorará el espectro de técnicas de procesamiento de imágenes con el propósito de discernir y catalogar las respuestas señaladas en los formatos de respuesta. Se investigará también la viabilidad de diferentes tecnologías para la extracción y conversión digital de datos personales del estudiante presentes en los documentos.

Es crucial reconocer las fronteras de este proyecto. La investigación se inclina hacia la utilización de algoritmos preexistentes para la manipulación de imágenes y la interpretación digital de textos, evitando la creación o modificación profundas de estos métodos existentes. Se priorizará la elaboración de una interfaz de usuario intuitiva y eficiente. Además, la investigación se limitará al ámbito de exámenes de selección múltiple, excluyendo el análisis de respuestas manuscritas o descriptivas.

CAPÍTULO I

Técnicas existentes de tratamiento de imágenes o reconocimiento de patrones.

El tratamiento de imágenes y el reconocimiento de patrones han emergido como campos fundamentales dentro de la visión por computadora y la inteligencia artificial. A medida que se vive en una era dominada por la información visual, la capacidad de procesar, analizar y comprender imágenes ha adquirido una importancia crucial en diversas aplicaciones que van desde la medicina y la astronomía hasta la seguridad y el entretenimiento. A través del tratamiento de imágenes, es posible mejorar la calidad visual, extraer características clave y transformar la información contenida en una imagen para lograr objetivos específicos [1].

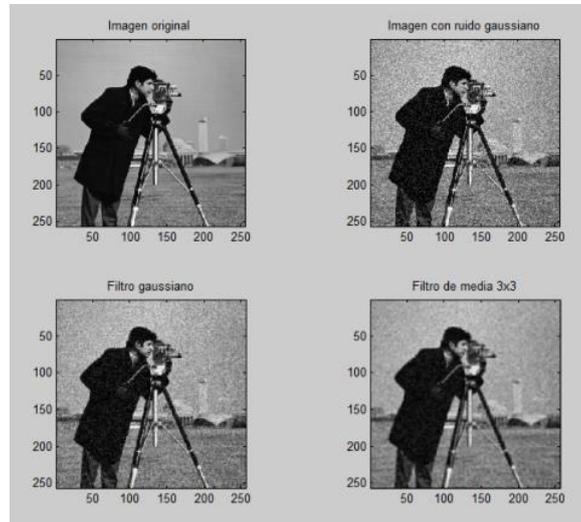
El reconocimiento de patrones, por otro lado, se centra en la identificación y clasificación de patrones o estructuras en datos complejos. En el contexto de las imágenes, esto implica la detección de objetos, la segmentación semántica y la clasificación, entre otras tareas. A lo largo de los años, estas técnicas han experimentado una evolución significativa, impulsada en parte por los avances en algoritmos y la disponibilidad de poder computacional. A medida que se profundice en el capítulo, se explora en profundidad las metodologías, técnicas y avances más recientes en estos campos intrincadamente conectados [2].

1.1 Visión Artificial y Tratamiento de Imágenes

La visión artificial y el tratamiento de imágenes convergen en el fascinante cruce entre la tecnología y la percepción, permitiendo que las máquinas interpreten y comprendan el mundo visual de una manera similar a los humanos. Mientras que la visión artificial busca replicar y mejorar la capacidad del ojo humano para identificar y procesar información, el tratamiento de imágenes se centra en manipular y mejorar esta información visual. Juntos, estos campos han revolucionado una variedad de sectores, desde la medicina hasta la robótica, y continúan

impulsando innovaciones que desafían las limitaciones tradicionales de la interacción máquina-mundo [3].

Figura 1 Filtros en imágenes con visión artificial



Nota: Figura detallando un ejercicio de aplicación de filtros para eliminación de ruido Gaussiano, esta imagen fue adaptada de [4].

1.1.1 Definición y Aplicaciones de la Visión Artificial

La visión artificial, también conocida como procesamiento de imágenes o visión por computadora, es un campo interdisciplinario que involucra la captura, procesamiento y análisis de imágenes y videos para extraer información y tomar decisiones. Implica la emulación de la capacidad visual humana mediante algoritmos y técnicas informática [5].

La visión artificial tiene una amplia gama de aplicaciones en diversas industrias y campos. Algunas de las aplicaciones más destacadas son:

Automatización Industrial: En la manufactura, la visión artificial se utiliza para inspeccionar productos, detectar defectos, supervisar la calidad y guiar robots en tareas precisas.

Automoción: En la industria automotriz, se aplica para el reconocimiento de señales de tráfico, asistencia al conductor, detección de peatones, estacionamiento autónomo y más.

Salud y Medicina: La visión artificial se utiliza en la interpretación de imágenes médicas como radiografías, resonancias magnéticas y tomografías computarizadas. También se emplea en la detección temprana de enfermedades y en la cirugía asistida por robot.

Agricultura de Precisión: Permite el monitoreo de cultivos, detección de enfermedades en plantas, clasificación de frutas y verduras, y análisis de suelos.

Seguridad y Vigilancia: En sistemas de seguridad, se emplea para el reconocimiento facial, detección de intrusos, análisis de comportamiento y seguimiento de objetos en tiempo real.

Retail y Comercio Electrónico: La visión artificial se utiliza para mejorar la experiencia del cliente a través de la realidad aumentada, recomendaciones personalizadas y análisis de tráfico en tiendas.

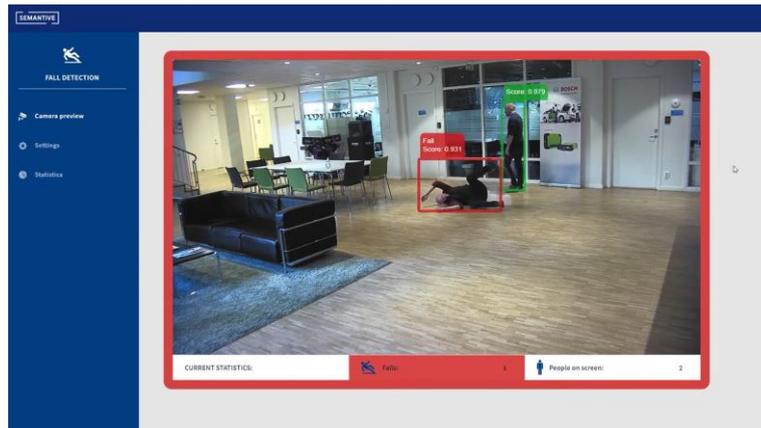
Robótica: En robótica, la visión artificial es esencial para la navegación autónoma, la interacción con humanos y la manipulación de objetos.

Realidad Aumentada y Virtual: En estas áreas, la visión artificial se utiliza para superponer elementos virtuales en el mundo real y para crear entornos virtuales más realistas.

Reconocimiento de Patrones: Esto incluye el reconocimiento de objetos, caras, escritura a mano y otros patrones complejos.

Industria de Entretenimiento: La visión artificial se utiliza en la producción de películas, animación y videojuegos para crear efectos visuales impresionantes y realistas [6].

Figura 2 Seguridad y vigilancia con visión artificial



Nota: Esta imagen demuestra un sistema de seguridad de detección de caídas, esta imagen fue tomada de la página oficial de exevi.com, elaborada por [7].

1.1.2 Procesamiento de Imágenes: Conceptos Básicos

El procesamiento de imágenes implica manipular y analizar imágenes digitales para mejorar su calidad, extraer información relevante o realizar tareas específicas. A través de algoritmos y técnicas computacionales, se pueden resaltar detalles, reducir ruido, detectar patrones y realizar una variedad de transformaciones en imágenes para diversos fines [8].

Imagen Digital: Una imagen digital es una representación numérica de una imagen en el mundo real. Está compuesta por píxeles (elementos de imagen mínimos), donde cada píxel contiene información sobre el color y la intensidad de la luz en una posición específica de la imagen.

Píxel: El píxel (acrónimo de "elemento de imagen") es la unidad básica que compone una imagen digital. Cada píxel contiene valores que representan el color y la intensidad de la luz en una ubicación específica de la imagen.

Resolución: La resolución de una imagen se refiere a la cantidad de píxeles que componen la imagen. Se mide en términos de ancho y alto, por ejemplo, "1920x1080" indica una resolución de 1920 píxeles de ancho por 1080 píxeles de alto.

Escala de Grises: Una imagen en escala de grises contiene solo niveles de gris, sin color. Cada píxel en una imagen en escala de grises tiene un valor que indica la intensidad luminosa en ese punto.

Espacio de Color: Un espacio de color es una representación matemática que describe cómo se representan los colores en una imagen. Ejemplos comunes incluyen RGB (rojo, verde, azul) y CMYK (cian, magenta, amarillo, negro).

Filtro: Un filtro es una operación que se aplica a una imagen para resaltar, suavizar, modificar o mejorar ciertos aspectos visuales. Los filtros pueden ser usados en el procesamiento de imágenes para lograr diversos efectos.

Convolución: La convolución es una operación matemática que combina dos funciones para crear una tercera. En el procesamiento de imágenes, la convolución se utiliza en la aplicación de filtros para modificar la apariencia de la imagen.

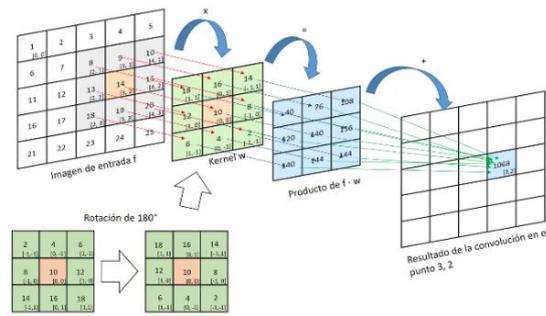
Umbralización: La umbralización es un proceso en el que se convierte una imagen en una imagen binaria, donde los píxeles se clasifican en blanco o negro según si su valor supera un umbral predefinido. Es útil para segmentar objetos de interés en una imagen.

Segmentación: La segmentación implica dividir una imagen en regiones o componentes para analizarlas individualmente. Puede usarse para identificar objetos, bordes u otras características en una imagen.

Morfología Matemática: La morfología matemática es una técnica que se utiliza para analizar y procesar formas en imágenes. Incluye operaciones como la dilatación (ampliación) y erosión (reducción) de regiones.

Transformada de Fourier: Es una técnica que descompone una imagen en sus componentes de frecuencia. Puede utilizarse para comprender las características de la imagen en el dominio de la frecuencia, lo que es útil para aplicaciones como la eliminación de ruido [3].

Figura 3 Convolución de imágenes



Nota: Proceso de convolución, en donde primero se multiplican el filtro w rotado con su región correspondiente de la imagen, esta imagen fue tomada de Cuartas, 2021 [9].

1.1.3 Técnicas de Tratamiento de Imágenes para la Detección de Marcas

Las técnicas de tratamiento de imágenes para la detección de marcas desempeñan un papel fundamental en la identificación y extracción precisa de elementos visuales distintivos. Estas técnicas, que incluyen umbralización adaptativa, detección de bordes y segmentación, permiten resaltar y aislar las características clave de las marcas en imágenes. Al combinar métodos como la convolución y el análisis de texturas, se logra una detección más eficaz y una interpretación precisa de las características que definen las marcas en el contexto de aplicaciones específicas [8].

Umbralización: La umbralización es una técnica simple pero efectiva que se utiliza para convertir una imagen en una imagen binaria (blanco y negro) basada en un valor umbral. Puede ayudar a resaltar las regiones de la imagen que contienen la marca en comparación con el fondo. La umbralización adaptativa puede ser útil cuando hay variaciones en la iluminación.

Filtros de Realce y Reducción de Ruido: Aplicar filtros de realce puede resaltar características específicas de una imagen, como bordes y detalles. Filtros como el filtro de Sobel y el filtro de Laplace pueden ayudar a identificar las partes importantes de una marca. Además, los filtros de reducción de ruido, como el filtro de mediana o el filtro Gaussiano, pueden mejorar la calidad de la imagen antes de la detección.

Detección de Bordes: Las técnicas de detección de bordes, como el operador de Canny, pueden ayudar a identificar los bordes de la marca en la imagen. Esto es especialmente útil si la marca tiene una forma distintiva.

Transformada de Hough: La transformada de Hough es una técnica que puede detectar formas geométricas en una imagen. Es útil para detectar círculos, elipses o líneas en una marca.

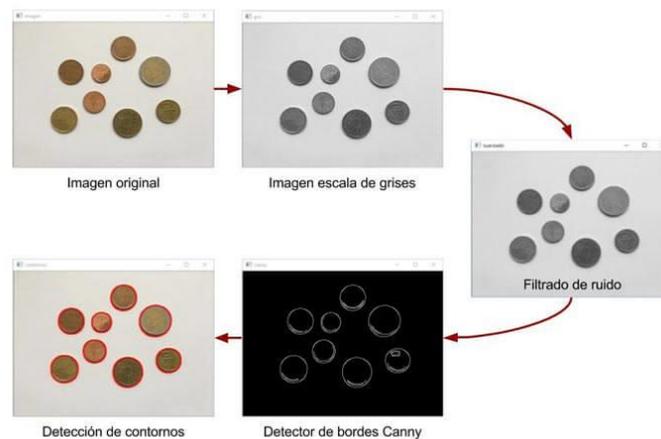
Correlación de Plantillas: La correlación de plantillas implica comparar una plantilla (una imagen pequeña que representa la marca) con diferentes partes de la imagen. Puede ayudar a localizar la posición aproximada de la marca en la imagen.

Descriptores de Características: Utilizar descriptores de características, como SIFT (Transformación de Características Invariantes a Escala), puede permitir la identificación única de la marca independientemente de su escala, rotación u orientación.

Segmentación: La segmentación permite dividir la imagen en regiones más manejables y puede ayudar a aislar la marca del fondo. Algunas técnicas incluyen la segmentación basada en color, textura o forma.

Morfología Matemática: Operaciones morfológicas como la dilatación y erosión pueden ayudar a limpiar y mejorar la forma de la marca, así como a eliminar pequeños detalles no deseados [3].

Figura 4. *Detección de bordes con visión artificial*



Nota: Esta imagen da a conocer la segmentación de objetos para la detección de monedas, esta figura fue tomada de [10]

1.1.4 Reconocimiento de Patrones en Imágenes: Métodos y Aplicaciones

Los métodos para el reconocimiento de patrones en imágenes abarcan una variedad de enfoques algorítmicos diseñados para identificar y clasificar características visuales distintivas en una imagen. Estos métodos involucran desde técnicas tradicionales basadas en la extracción de características, como descriptores y transformadas, hasta enfoques modernos impulsados por el aprendizaje automático, como las redes neuronales convolucionales. La combinación de algoritmos de detección de bordes, extracción de características y técnicas de aprendizaje profundo permite una interpretación precisa y automatizada de patrones visuales en diferentes aplicaciones, como detección de objetos, análisis médico y reconocimiento facial [11].

Los métodos que se pueden utilizar para el reconocimiento de patrones en imágenes pueden ser los siguiente:

Máquinas de Vectores de Soporte (SVM): Los SVM son algoritmos de aprendizaje supervisado que pueden utilizarse para la clasificación y detección de patrones en imágenes. Son efectivos para separar diferentes clases de objetos en imágenes [12].

Redes Neuronales Convolucionales (CNN): Las CNN son arquitecturas de redes neuronales diseñadas específicamente para el procesamiento de imágenes. Son ampliamente utilizadas para tareas de clasificación, segmentación y detección de objetos [13].

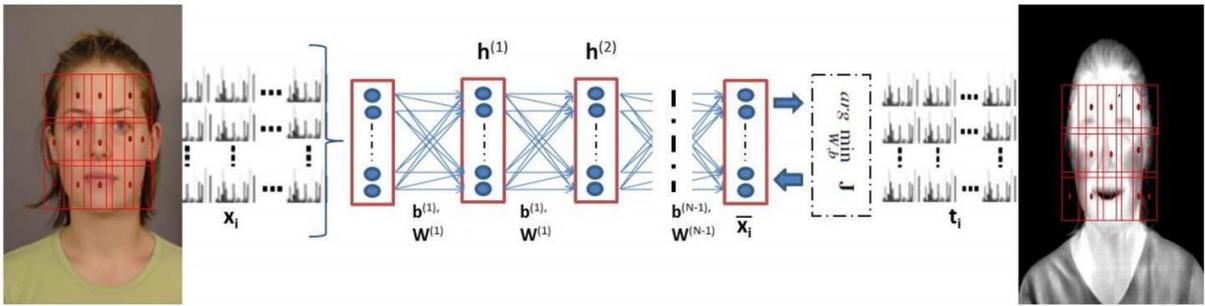
Descenso de Gradiente Estocástico (SGD): Este es un algoritmo de optimización que se utiliza en entrenamiento de modelos para encontrar parámetros que minimicen una función de pérdida. Puede ser aplicado a diversas tareas de reconocimiento de patrones.

Transformada de Fourier: Se utiliza para analizar la frecuencia de patrones en imágenes, lo que puede ser útil para la detección de bordes y texturas.

Descriptores Locales: Métodos como SIFT (Transformación de Características Invariantes a Escala) y SURF (Búsqueda Rápida de Características Robustas) extraen características distintivas de regiones específicas de una imagen [14].

Las aplicaciones de reconocimiento de patrones en imágenes son variadas y abarcan numerosos campos. Desde la detección de objetos y reconocimiento facial en seguridad y tecnología automotriz, hasta el análisis médico para identificar patologías en imágenes médicas, el reconocimiento de patrones en imágenes juega un papel esencial. Además, su aplicación se extiende a la industria manufacturera para el control de calidad, en la clasificación de contenido en redes sociales y en la preservación del patrimonio cultural mediante la restauración y digitalización de obras de arte. Estas aplicaciones destacan cómo el análisis automatizado de patrones visuales mejora la eficiencia y la precisión en una amplia gama de áreas [11].

Figura 5 Redes Neuronales Convolucionales (CNN)



Nota: Imagen de un sistema de redes neuronales artificiales para procesar las imágenes en infrarrojo, esta figura fue tomada de [15]

1.2 Reconocimiento Óptico de Caracteres (OCR)

El Reconocimiento Óptico de Caracteres (OCR) es una tecnología que convierte textos impresos o escritos a mano en formatos digitales editables. Mediante el análisis de imágenes o documentos escaneados, el OCR identifica caracteres, reconociendo letras, números y símbolos con alta precisión. Esta tecnología automatiza tareas como la digitalización de libros y la conversión de documentos en papel a texto digital. El OCR se utiliza en la gestión de documentos, la industria editorial, la transcripción histórica y la accesibilidad para discapacidades visuales, al hacer que el contenido impreso sea accesible en formatos digitales. Su capacidad para procesar información rápidamente ha transformado la interacción con el contenido impreso, facilitando su búsqueda y edición en la era digital [16]

Figura 6 Lectura OCR basada en Deep Learning



Nota: La herramienta de lectura OCR entrena los caracteres etiquetados y los aprende para poder identificar, aunque exista un alto grado de variabilidad en el texto, imagen tomada de [17].

1.2.1 Concepto y Funcionamiento del OCR

El Reconocimiento Óptico de Caracteres (OCR) es una tecnología que se utiliza para convertir textos impresos o escritos a mano en imágenes en formatos digitales editables. Esta tecnología permite que los sistemas de computadora analicen y procesen imágenes que contienen caracteres, como letras, números y símbolos, para luego reconocer y convertir esos caracteres en texto digital. El OCR es aplicado en una variedad de campos, como la digitalización de libros, la transcripción de documentos, la administración de archivos y la accesibilidad para personas con discapacidades visuales [16].

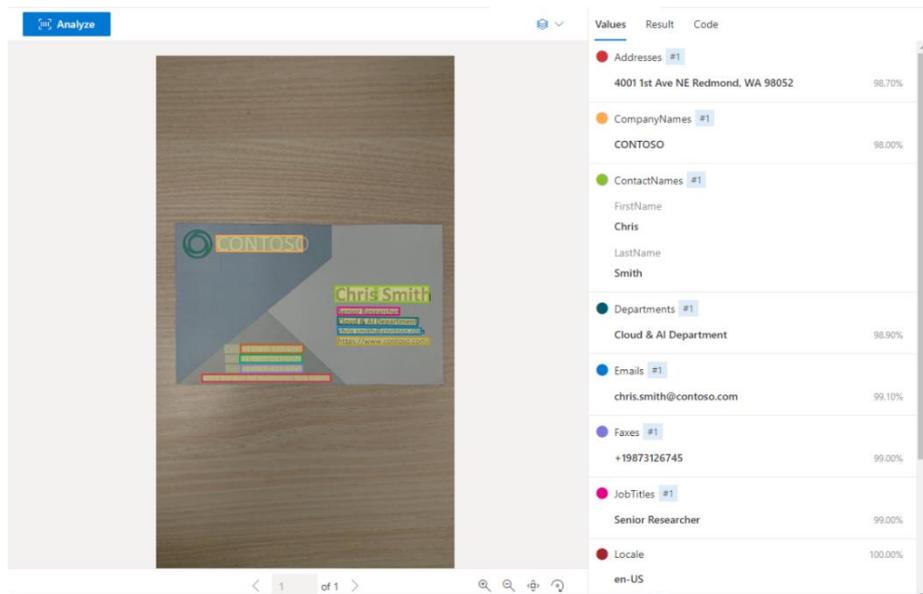
El Reconocimiento Óptico de Caracteres (OCR) es un proceso en el cual una imagen o un documento escaneado que contiene texto es analizado por un software especializado. Durante este proceso, el software identifica patrones visuales en la imagen que corresponden a caracteres individuales, como letras, números y símbolos. Luego, el software compara estos patrones con una base de datos de caracteres conocidos y realiza una correspondencia para determinar qué caracteres específicos están presentes en la imagen. Finalmente, los caracteres reconocidos se convierten en texto digital editable que puede ser manipulado, buscado y almacenado electrónicamente [3].

1.2.2 Aplicaciones del OCR en la Extracción de Datos Personales

El Reconocimiento Óptico de Caracteres (OCR) desempeña un papel crucial en la extracción de datos personales de documentos impresos, como formularios de solicitud, facturas y tarjetas de identificación. Estas aplicaciones automatizan la captura y digitalización de información como nombres, direcciones, números de identificación y fechas, agilizando los procesos de registro, administración de documentos y verificación de identidad. Además, el OCR mejora la

precisión al eliminar errores humanos en la entrada de datos, lo que es esencial en campos como la banca, la atención médica y la gestión de recursos humanos [18].

Figura 7 Extracción de datos personales de una tarjeta



Nota: Programa que emplea visión artificial, para registro y acceso de edificios, esta imagen fue tomada de [18].

1.3 Metodologías de Evaluación de Exámenes

Las metodologías de evaluación de exámenes juegan un papel fundamental en la medición del conocimiento y las habilidades de los estudiantes. Estas metodologías abarcan una variedad de enfoques, desde preguntas de opción múltiple hasta problemas prácticos y evaluaciones basadas en proyectos. Además, las técnicas modernas incluyen la evaluación en línea y adaptativa, que permiten una medición más precisa y personalizada del rendimiento estudiantil. La elección de la metodología adecuada depende de los objetivos de evaluación y la materia en cuestión [19]

1.3.1 Evaluación Manual: Proceso y Desafíos

El proceso de evaluación manual implica la revisión y puntuación individual de respuestas de los estudiantes en exámenes y pruebas. Aunque puede ser minucioso y permitir una

consideración detallada de las respuestas, este enfoque enfrenta desafíos significativos. La subjetividad del evaluador puede llevar a variaciones en la puntuación, lo que impacta la objetividad y la consistencia de los resultados. Además, la evaluación manual puede ser laboriosa y consumir tiempo, especialmente en grupos numerosos de estudiantes. La necesidad de manejar una carga de trabajo intensiva y garantizar la equidad en la evaluación son aspectos cruciales que requieren atención en este proceso [20]

1.3.2 Evaluación Automática: Beneficios y Limitaciones

La evaluación automática ofrece una serie de beneficios notables. En primer lugar, agiliza el proceso de calificación al eliminar la necesidad de una revisión manual exhaustiva. Esto permite un rápido retorno de resultados a los estudiantes. Además, la evaluación automática puede ser más objetiva y consistente, ya que se basa en criterios predefinidos y algoritmos de puntuación. Sin embargo, esta metodología también enfrenta limitaciones. La adaptación a la diversidad de respuestas y enfoques de los estudiantes puede ser un desafío. Por lo tanto, algunas respuestas creativas o complejas pueden ser mal interpretadas por los algoritmos, lo que afecta la precisión de la evaluación. Aunque la evaluación automática ofrece eficiencia, es crucial equilibrarla con una supervisión humana para garantizar resultados justos y confiables [9].

1.3.3 Métodos Existentes de Evaluación Automática

Los métodos de evaluación automática abarcan desde la comparación de respuestas con criterios predefinidos hasta el análisis de texto y el uso de algoritmos de aprendizaje automático. Estos enfoques permiten evaluar respuestas de manera objetiva y eficiente, incluyendo la puntuación automática de opciones múltiples y la creación de modelos para comparación [21].

Los métodos existentes de evaluación automática pueden incluir:

Evaluación basada en criterios: Los sistemas de evaluación automática comparan las respuestas de los estudiantes con un conjunto de criterios predeterminados. Las respuestas se puntúan según la similitud con los criterios, lo que permite una puntuación objetiva.

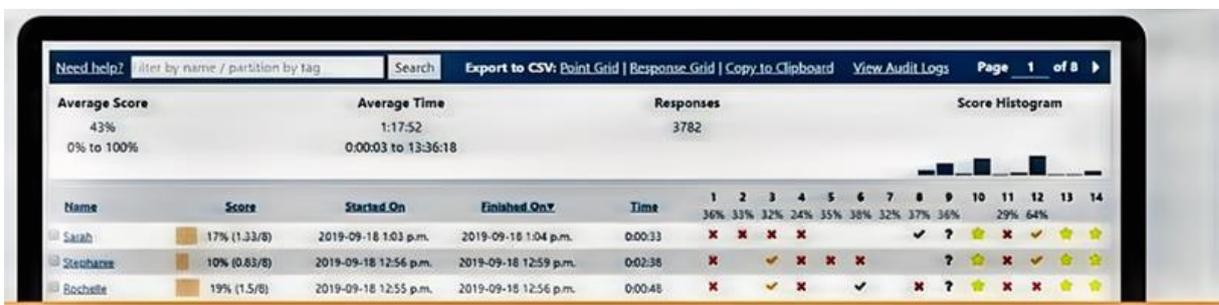
Análisis de texto: Los algoritmos de procesamiento de lenguaje natural analizan las respuestas escritas por los estudiantes. Pueden evaluar la coherencia, el uso de palabras clave y la estructura gramatical para determinar la calidad de la respuesta.

Puntuación automática de opción múltiple: Para preguntas de opción múltiple, los sistemas automáticos verifican si las respuestas seleccionadas por el estudiante coinciden con las respuestas correctas.

Evaluación basada en modelos: Algunos sistemas crean un modelo de respuestas correctas basado en respuestas de expertos y lo utilizan para comparar con las respuestas de los estudiantes.

Aprendizaje automático y análisis de datos: Los algoritmos de aprendizaje automático pueden entrenarse en un conjunto de respuestas previas y luego se utilizan para evaluar nuevas respuestas. El análisis de datos puede detectar patrones en las respuestas y asignar puntuaciones [22]

Figura 8 Evaluación mediante análisis de datos



Nota: imagen del sistema capaz de detectar patrones y dar puntuaciones, esta imagen fue tomada de [20]

1.3.4 Evaluación de Exámenes de Opción Múltiple mediante Tratamiento de Imágenes

La evaluación de exámenes de opción múltiple mediante tratamiento de imágenes implica el uso de tecnologías de procesamiento de imágenes y reconocimiento óptico de caracteres (OCR) para analizar y evaluar automáticamente las respuestas de los estudiantes en hojas de respuesta. Las hojas de respuesta pueden ser digitalizadas o fotografiadas, y luego el OCR se utiliza para extraer las selecciones marcadas por los estudiantes en las casillas de opción múltiple. Luego, los sistemas de procesamiento de imágenes comparan las selecciones extraídas con las respuestas correctas y asignan una puntuación automáticamente [23]

Figura 9 Evaluación de Exámenes de Opción Múltiple mediante Tratamiento de Imágenes



Nota: aplicación móvil que identifica patrones de evaluaciones por visión artificial, imagen extraída de [24]

1.4 Sistemas de Evaluación Automática Basados en Visión Artificial

Los sistemas de Evaluación Automática Basados en Visión Artificial aprovechan tecnologías avanzadas de procesamiento de imágenes y reconocimiento óptico de caracteres para automatizar la evaluación de exámenes y pruebas. Estos sistemas escanean y analizan las hojas de respuesta y respuestas escritas, utilizando algoritmos de visión por computadora para identificar y puntuar respuestas correctas e incorrectas. La combinación de técnicas de procesamiento de imágenes y aprendizaje automático permite una evaluación objetiva, eficiente

y precisa, liberando a los educadores de la carga de calificar manualmente un gran número de exámenes [25].

1.4.1 Sistemas Actuales y su Funcionamiento

Los sistemas de evaluación automática basados en visión artificial funcionan de la siguiente manera:

Captura de Imágenes: Las hojas de respuesta y respuestas escritas se escanean o fotografían para convertirlas en imágenes digitales.

Preprocesamiento: Las imágenes se someten a un preprocesamiento para mejorar la calidad y eliminar ruidos, asegurando una base sólida para el análisis.

Detección de Regiones de Interés: Mediante técnicas de detección y segmentación, el sistema identifica y aísla las áreas relevantes, como casillas de opción múltiple y respuestas escritas.

Reconocimiento Óptico de Caracteres (OCR): Las áreas de texto son procesadas mediante algoritmos de OCR para convertir el contenido impreso en texto digital.

Comparación y Puntuación: Las respuestas extraídas se comparan con las respuestas correctas previamente establecidas. Para preguntas de opción múltiple, se verifica la coincidencia entre las selecciones de los estudiantes y las respuestas correctas. Para respuestas escritas, se puede utilizar procesamiento de lenguaje natural para evaluar la coherencia y el contenido.

Asignación de Puntajes: Las respuestas se puntúan de acuerdo con los criterios predefinidos. Los sistemas pueden asignar automáticamente puntajes a respuestas correctas e incorrectas [18].

1.4.2 Ventajas y Desventajas de los Sistemas de Evaluación Automática Basados en Visión Artificial

Los sistemas de evaluación automática basados en visión artificial ofrecen ventajas notables, como la eficiencia al manejar grandes volúmenes de exámenes y la objetividad al aplicar

criterios predefinidos. Sin embargo, enfrentan desafíos en la evaluación de respuestas creativas y la adaptación a nuevos formatos de preguntas. La dependencia en la calidad de las imágenes y la posible falta de comprensión del contexto también son aspectos para considerar en su implementación.

- **Ventajas de los Sistemas de Evaluación Automática Basados en Visión**

Artificial:

Eficiencia: Agilizan el proceso de evaluación, permitiendo manejar un gran número de exámenes en menos tiempo.

Objetividad: Las puntuaciones se asignan de manera consistente según criterios predefinidos, minimizando el sesgo subjetivo.

Consistencia: Proporcionan resultados coherentes al aplicar los mismos criterios a todas las respuestas.

Retroalimentación Rápida: Los estudiantes reciben retroalimentación inmediata sobre su rendimiento.

Automatización: Liberan a los educadores de la carga de calificar manualmente, permitiéndoles centrarse en actividades más pedagógicas.

- **Desventajas de los Sistemas de Evaluación Automática Basados en Visión**

Artificial:

Limitación en Preguntas Abiertas: Pueden tener dificultades para evaluar respuestas creativas o preguntas abiertas que requieren análisis profundo.

Errores en Reconocimiento: La precisión del reconocimiento óptico de caracteres (OCR) puede verse afectada por la calidad de las imágenes y la legibilidad de la escritura.

Dependencia en Formato de Respuesta: Los estudiantes deben seguir un formato específico al marcar respuestas, lo que podría limitar la flexibilidad.

Adaptación a Nuevos Formatos: La adaptación a nuevos tipos de preguntas y formatos puede requerir ajustes y pruebas.

Falta de Contexto: Los sistemas pueden tener dificultades para comprender el contexto o matices en respuestas escritas [25].

1.4.3 Desafíos y Oportunidades para Mejorar los Sistemas de Evaluación Automática

Los sistemas de evaluación automática enfrentan desafíos en la evaluación de respuestas abiertas y la precisión del reconocimiento. Sin embargo, las oportunidades para mejorar son significativas, incluyendo el uso de tecnologías avanzadas como el aprendizaje automático y la integración con plataformas educativas en línea. A medida que se abordan los desafíos, estas mejoras prometen una evaluación más eficiente y precisa en los entornos educativos modernos.

- **Desafíos para Mejorar los Sistemas de Evaluación Automática:**
 - **Evaluación de Respuestas Abiertas:** Mejorar la capacidad de evaluar respuestas abiertas y creativas que requieren análisis en profundidad.
 - **Precisión del Reconocimiento:** Superar los errores en el reconocimiento óptico de caracteres (OCR) causados por mala calidad de imágenes o escritura ilegible.
 - **Diversidad de Respuestas:** Manejar la variedad de enfoques y respuestas de los estudiantes para garantizar resultados justos y precisos.
 - **Adaptabilidad:** Asegurar que los sistemas sean adaptables a nuevos formatos y tipos de preguntas.
 - **Contexto y Matices:** Abordar la falta de comprensión de matices y contexto en respuestas escritas.

- **Oportunidades para Mejorar los Sistemas de Evaluación Automática:**
 - **Tecnologías Avanzadas:** Utilizar técnicas más avanzadas de visión por computadora y procesamiento de lenguaje natural para mejorar la comprensión y evaluación de respuestas.
 - **Aprendizaje Automático:** Aplicar algoritmos de aprendizaje automático para mejorar la precisión y adaptabilidad de los sistemas.
 - **Mejora del Reconocimiento Óptico:** Desarrollar OCR más robustos y precisos para abordar errores de reconocimiento.
 - **Retroalimentación Personalizada:** Ofrecer retroalimentación más detallada y personalizada a los estudiantes.
 - **Integración con Plataformas Educativas:** Introducir sistemas de evaluación automática en plataformas de aprendizaje en línea para una experiencia más fluida [27]

CAPÍTULO II

Marco Metodológico

Este capítulo detalla la metodología utilizada para desarrollar un sistema automatizado de calificación de exámenes en papel mediante visión artificial. Se adopta un enfoque aplicado y experimental, comenzando con un análisis exhaustivo de los requerimientos mediante entrevistas y revisión bibliográfica. Posteriormente, se diseña la arquitectura del sistema y se implementan las tecnologías de procesamiento de imágenes y OCR. Las pruebas rigurosas aseguran su funcionalidad y precisión, y el despliegue final incluye un plan de mantenimiento para mejoras continuas. Esta metodología garantiza una solución efectiva y eficiente para la automatización de la calificación en el ámbito educativo.

2.1 Tipo de investigación

En el presente estudio, la investigación se clasifica como aplicada y experimental. Como investigación aplicada, el objetivo principal es desarrollar soluciones concretas y prácticas para un problema específico: la automatización en la corrección de exámenes de opción múltiple mediante el uso de tecnologías de visión artificial y reconocimiento óptico de caracteres (OCR). Este tipo de investigación es fundamental para trasladar teorías y conocimientos del ámbito teórico a aplicaciones prácticas y reales, lo que resulta esencial en campos tecnológicos y de innovación.

Por otro lado, el carácter experimental de esta investigación se refleja en el desarrollo y prueba de un prototipo de sistema de evaluación automática. La metodología experimental permite no solo la construcción del sistema, sino también la evaluación rigurosa de su eficacia a través de pruebas controladas. Estas pruebas ayudarán a entender mejor cómo la implementación de algoritmos específicos de visión artificial y OCR puede mejorar la eficiencia y precisión en la corrección de exámenes, permitiendo así una evaluación objetiva y sistemática de los resultados.

La combinación de investigación aplicada y experimental es ideal para este proyecto, ya que permite abordar un problema práctico mediante un enfoque sistemático y controlado. A través de este enfoque, se espera generar conocimientos que no solo sean de valor teórico, sino que también tengan aplicaciones directas y beneficiosas en el ámbito educativo y tecnológico.

2.2 Nivel de investigación

El nivel de investigación de este estudio se clasifica como exploratorio y descriptivo. En la fase exploratoria, el proyecto tiene como objetivo investigar nuevas aplicaciones de la visión artificial y el reconocimiento óptico de caracteres (OCR) en el contexto específico de la corrección automática de exámenes de opción múltiple. Esta fase es esencial para comprender las posibilidades y limitaciones de la tecnología actual y para identificar áreas donde la innovación puede ser más efectiva. La investigación exploratoria es particularmente adecuada para campos en rápida evolución, como lo es la inteligencia artificial, donde las nuevas tecnologías y técnicas están constantemente emergiendo.

Además, el estudio también tiene un componente descriptivo. Una vez identificadas las tecnologías y métodos potenciales, la investigación se centrará en describir de manera detallada el funcionamiento y la implementación del sistema propuesto. Esta descripción incluirá el diseño del algoritmo, la arquitectura del sistema, y las estrategias de evaluación y prueba. El nivel descriptivo es crucial para documentar el proceso de desarrollo del prototipo y para proporcionar una comprensión clara de cómo la combinación de visión artificial y OCR puede ser aplicada de manera práctica y efectiva en la evaluación de exámenes.

La combinación de estos niveles de investigación proporciona un enfoque integral que no solo busca explorar nuevas posibilidades en el ámbito de la tecnología educativa, sino también describir en detalle la aplicación y los resultados de estas innovaciones, facilitando así su posible replicación y adaptación en contextos similares.

2.3 Metodología

La metodología adoptada en este proyecto se basa en el modelo en cascada, un enfoque sistemático y secuencial típicamente utilizado en el desarrollo de software. Este modelo es especialmente adecuado para proyectos con requisitos bien definidos y donde es crucial una estructura organizada y una progresión lineal de fases. A continuación, se detallan las fases del modelo en cascada adaptadas a nuestro proyecto de evaluación automática de exámenes:

- **Análisis de Requerimientos:** En esta fase inicial, se llevará a cabo una recopilación exhaustiva de los requisitos del sistema. Esto incluye la identificación de las necesidades específicas para la evaluación automática de exámenes, como los tipos de preguntas a procesar, los formatos de examen admitidos, y las características deseables del sistema.
- **Diseño del Sistema:** Basándonos en los requerimientos recopilados, se procederá a diseñar la arquitectura del sistema. Esta fase implica la creación de diagramas de flujo, modelos de datos y mockups de la interfaz de usuario, asegurando que todas las necesidades identificadas se aborden de manera efectiva.
- **Implementación y Codificación:** Una vez establecido el diseño, se iniciará la fase de implementación. Aquí, se desarrollará el código del sistema, incluyendo la integración de tecnologías de visión artificial y OCR. Esta fase se realizará en iteraciones, permitiendo revisar y ajustar el sistema a medida que se avanza.
- **Pruebas del Sistema:** Tras la implementación, el sistema será sometido a una serie de pruebas para asegurar su funcionalidad y fiabilidad. Esto incluirá pruebas unitarias, pruebas de integración y pruebas de sistema, con el objetivo de identificar y corregir cualquier error o problema de rendimiento.
- **Despliegue y Mantenimiento:** En la última fase, el sistema será desplegado y puesto en funcionamiento. Además, se establecerá un plan de mantenimiento para realizar actualizaciones y mejoras necesarias, basándose en los comentarios de los usuarios y en la evolución de las tecnologías utilizadas.

2.3.1 Fase 1: Adquisición de requerimientos

Esta fase inicial es fundamental para definir el rumbo del proyecto. Se establece un enfoque detallado y multidimensional para recoger todos los requerimientos necesarios, asegurando que el sistema de evaluación automática sea tanto funcional como eficiente. La fase se divide en tres actividades clave:

- **Actividad 1.- Estudio bibliográfico:**

La primera etapa de la adquisición de requerimientos implica realizar un estudio bibliográfico exhaustivo. El objetivo es obtener una comprensión amplia y profunda de los sistemas de evaluación automática, la visión artificial y el reconocimiento óptico de caracteres (OCR) desde una perspectiva teórica y práctica. Este proceso incluye la revisión meticulosa de literatura académica, como artículos de investigación, reseñas tecnológicas, y estudios de caso relevantes. Al sumergirse en fuentes especializadas, el proyecto se nutre de conocimientos y descubrimientos existentes, lo que permite una base sólida para el desarrollo innovador. La revisión se enfocará en identificar tendencias actuales, desafíos comunes, y soluciones probadas en el campo.

El resultado de esta actividad será una colección de información valiosa que servirá como referencia principal para las fases subsiguientes del proyecto. Este compendio no solo informará sobre las mejores prácticas y avances tecnológicos en el campo de la evaluación automática y la visión artificial, sino que también ayudará a identificar brechas y oportunidades para innovar. Además, la información recopilada permitirá anticipar desafíos técnicos y operativos, facilitando la formulación de estrategias efectivas para abordarlos en el diseño y desarrollo del sistema.

- **Actividad 2.- Entrevistas con especialistas y usuarios**

En esta investigación se procede a utilizar la entrevista como técnica de recolección de información, la cual es una interacción entre una o varias personas las cuales responden a debidas preguntas establecidas.

Entrevista

La interacción directa con usuarios finales y especialistas es una herramienta invaluable en la investigación aplicada. Para esta actividad, se planificaron cinco entrevistas con docentes de diversas áreas académicas, proporcionando una perspectiva amplia y variada. El objetivo es comprender mejor las necesidades, expectativas y posibles preocupaciones de los educadores respecto a la implementación de un sistema de evaluación automática. La confidencialidad y el anonimato son prioritarios, asegurando que los participantes compartan abiertamente sus opiniones y experiencias.

- **Resumen y análisis:**

Se realizaron entrevistas con cinco docentes, especializados en matemáticas, ciencias sociales, idiomas, ciencias naturales y arte, respectivamente. A través de estas conversaciones, se identificaron varios temas comunes:

- **Necesidad de Precisión y Fiabilidad:** Todos los docentes enfatizaron la importancia de que el sistema proporcione resultados precisos y consistentes, destacando la necesidad de minimizar los errores en la evaluación.
- **Facilidad de Uso:** Se resaltó la importancia de una interfaz intuitiva y fácil de manejar. Los docentes manifestaron su preferencia por un sistema que requiera una curva de aprendizaje mínima.

- **Flexibilidad en el Formato de Preguntas:** Aunque todos reconocieron el valor de la evaluación de opción múltiple, algunos expresaron su interés en que el sistema pudiera adaptarse en el futuro para evaluar otros tipos de respuestas, como respuestas cortas o ensayos.
- **Preocupaciones sobre la Integración con Sistemas Existentes:** Hubo cierta preocupación sobre cómo el nuevo sistema se integraría con los sistemas de gestión de aprendizaje y bases de datos de estudiantes ya en uso.
- **Potencial para Retroalimentación Instantánea:** Los docentes de idiomas y artes expresaron particular interés en la posibilidad de que el sistema ofreciera retroalimentación instantánea a los estudiantes, lo que podría ser una herramienta valiosa para el aprendizaje.

Estos hallazgos proporcionan perspectivas cruciales que influirán en el diseño y desarrollo del sistema, asegurando que se alinee con las necesidades reales de los usuarios finales y mejore la experiencia educativa tanto para docentes como para estudiantes.

- **Actividad 3.- Adquisición de requerimientos**

La tercera actividad es esencial para definir el esqueleto y el funcionamiento del sistema de evaluación automática. Tras obtener una comprensión teórica y práctica a través de estudios bibliográficos y recopilar opiniones y experiencias directas mediante entrevistas, la siguiente etapa consiste en consolidar todos estos datos en requerimientos concretos y bien definidos. Esta consolidación se realiza categorizando los requerimientos en diferentes tipos, cada uno abordando aspectos específicos del sistema, estos son los siguientes:

- Requerimientos de Usuario (RU)
- Requerimientos Operacionales (RO)

- Requerimientos del Sistema (RS)
- Requerimientos de arquitectura (RA)

Requerimientos de Usuario

Estos requerimientos se centran en las necesidades y expectativas de los usuarios finales, que en este caso incluyen a docentes y administradores académicos. En la Tabla 4, se tiene los respectivos requerimientos de la información recopilada.

Tabla 1: Requerimientos de usuario

NÚMERO	REQUERIMIENTOS	PRIORIDAD	RELACIÓN
RU1	Interfaz intuitiva y fácil de usar	Alta	Relacionado con RU2 y RU3
RU2	Capacidad para generar informes detallados de evaluación	Baja	Complementa a RU1
RU3	Precisión en la identificación de respuestas	Alta	Relacionado con RU1 y RU4
RU4	Rápida respuesta y procesamiento del sistema	Alta	Soporta RU4 y complementa a RU1
RU5	Seguridad en el manejo y almacenamiento de datos	Alta	Fundamental para todos los RUs

Requerimientos Operacionales

Aquí se definen las necesidades relacionadas con el entorno operativo del sistema observado en la Tabla 2.

Tabla 2 Requerimientos Operacionales

Número	Requerimientos	Prioridad	Relación
RO1	Funcionamiento estable en diversas plataformas y dispositivos	Alta	Relacionado con RO2 y RO4
RO2	Bajo consumo de recursos y optimización del rendimiento	Media	Complementa a RO1 y RO3
RO3	Facilidad de instalación y configuración	Media	Complementa a RO2
RO4	Soporte técnico y actualizaciones periódicas	Alta	Fundamental para todos los ROs
RO5	Integración con bases de datos	Alta	Relacionado con RO6
RO6	Capacidad para manejar grandes volúmenes de datos y usuarios	Alta	Soporta RO5 y complementa a RO1
RO7	Escalabilidad para adaptarse a futuras expansiones y mejoras	Media	Relacionado con RO6

Requerimientos del Sistema

Estos requerimientos se refieren a las capacidades técnicas y funcionales que el sistema debe tener, como se muestra en la Tabla 3.

Tabla 3 *Requerimientos del sistema*

Número	Requerimientos	Prioridad	Relación
RS1	Algoritmos de visión artificial para la detección de respuestas	Alta	Relacionado con RS2 y RS4
RS2	Implementación eficiente de OCR para el reconocimiento de texto	Alta	Complementa a RS1 y RS3
RS3	Interfaz de programación de aplicaciones (Qt5) para la integración de sistemas externos	Media	Complementa a RS2
RS4	Sistema robusto de gestión de errores y registro de actividades	Alta	Relacionado con RS1 y RS5
RS5	Funcionalidad de retroalimentación inmediata y análisis de resultados	Media	Soporta RS4 y complementa a RS1
RS6	Capacidad de configuración según las necesidades del usuario	Media	Relacionado con RS2 y RS5
RS7	Actualizaciones y mantenimiento del sistema	Media	Fundamental para todos los RSs

Requerimientos de Arquitectura

Se relacionan con la estructura general y el diseño del sistema. como se ilustra en la Tabla 4.

Tabla 4 *Requerimientos de Arquitectura*

Número	Requerimientos	Prioridad	Relación
RA1	Diseño modular para facilitar la actualización y mantenimiento	Alta	Soporta RA2, RA3 y RA4
RA2	Arquitectura escalable para soportar un creciente número de usuarios y cargas de trabajo	Alta	Complementa a RA1 y RA5
RA3	Integración eficaz de componentes de visión artificial y OCR	Alta	Complementa a RA1 y RA4
RA4	Sistema de gestión de bases de datos eficiente y seguro	Alta	Relacionado con RA3 y RA5
RA5	Infraestructura de red segura y de alta disponibilidad	Alta	Soporta RA2 y complementa a RA4
RA6	Interoperabilidad con sistemas y software educativos existentes	Media	Relacionado con RA2 y RA3
RA7	Mecanismos de recuperación ante fallos y redundancia	Media	Fundamental para todos los RAs

2.3.2 Fase 2: Diseño del Sistema

La Fase 2 del proyecto, centrada en el Diseño del Sistema, es un pilar crucial en la transformación de nuestras ideas y requerimientos iniciales en una solución tecnológica

concreta y funcional. Esta fase se encarga de dar forma a la arquitectura del sistema, eligiendo herramientas y tecnologías adecuadas, y delineando la estructura y el flujo del proceso a través de diagramas de bloques y flujogramas. La selección de las herramientas de software no solo define las capacidades y la eficiencia del sistema, sino que también impacta directamente en la experiencia del usuario y en la facilidad de mantenimiento y escalabilidad futura. Por otro lado, los diagramas de bloques y flujogramas nos permitirán visualizar y entender mejor la interacción entre los distintos componentes del sistema, asegurando que todos los elementos funcionen armónicamente para alcanzar los objetivos planteados. La meticulosidad y precisión en esta fase son fundamentales para garantizar el éxito en la implementación y la operatividad efectiva del sistema de evaluación automática.

- **Actividad 4.- Selección de Herramientas de Software**

La elección de las herramientas de software adecuadas es un paso esencial en el diseño de cualquier sistema tecnológico, y más aún en un proyecto complejo como el nuestro, que implica la evaluación automática de exámenes. Esta actividad se enfoca en seleccionar las herramientas que no solo cumplen con nuestras necesidades funcionales y operativas, sino que también garantizan la eficiencia, escalabilidad y usabilidad del sistema. En la *Tabla 5* se puede observar las librerías, el uso, la justificación y sus respectivas versiones.

Tabla 5 *Herramientas de software utilizadas*

Librería	Uso	Justificación	Versión Estimada
SQLite	Gestión de bases de datos	Ligera, eficiente y no requiere un servidor de base de datos separado. Ideal para almacenar resultados de exámenes y datos de usuarios.	3.36.0
PyQt5	Creación de la interfaz gráfica (GUI)	Proporciona herramientas para desarrollar interfaces intuitivas y profesionales, mejorando la experiencia del usuario.	5.15.4
cv2 (OpenCV)	Procesamiento de imágenes y visión artificial	Robusta y versátil para operaciones de procesamiento de imágenes, crucial para la detección de respuestas en exámenes.	4.5.3

NumPy	Manipulación de datos y matrices	Esencial para el manejo eficiente de datos y cálculos numéricos, especialmente en procesamiento de imágenes.	1.21.2
Pytesseract	Reconocimiento óptico de caracteres (OCR)	Facilita la conversión de imágenes a texto, necesario para procesar datos escritos a mano en exámenes.	0.3.8

2.3.3 Fase 3: Desarrollo de la interfaz de usuario

El algoritmo de evaluación automática de exámenes se complementa con una interfaz de usuario desarrollada meticulosamente para garantizar una interacción intuitiva y eficiente. La interfaz, construida en PyQt5, se centra en proporcionar una experiencia de usuario fluida que simplifique las complejidades del procesamiento de imágenes y OCR. Cada elemento, desde botones hasta campos de entrada y visualización de datos, está diseñado para maximizar la usabilidad y la accesibilidad.

2.3.4 Fase 4: Pruebas del Algoritmo

La fase de pruebas es un componente crítico en el desarrollo de cualquier sistema de software, y nuestro algoritmo de evaluación automática no es la excepción. Esta fase tiene como objetivo verificar la funcionalidad, confiabilidad y rendimiento del algoritmo, asegurando que las operaciones de reconocimiento de caracteres y procesamiento de imágenes se ejecuten según lo esperado. Las pruebas exhaustivas ayudan a identificar y corregir errores, mejorar la precisión y garantizar que la experiencia del usuario final sea impecable.

Antes de comenzar este ámbito de las pruebas, es necesario presentar el formato de evaluación de la cual se extraerá los respectivos datos de calificación para proporcionarlos a la base de datos a través de la interface, este formato como se lo puede apreciar en la *Figura 10*, está conformada por el nombre del colegio, información e indicaciones del uso, a que este documento tiene dos sección, la primera hoja de Banco de respuestas y la segunda sección de preguntas con opciones. En la primera sección que es la que va a ser registrada, tiene dos

cuadrantes el primero para escribir el número de identificación y la segunda con la matriz de respuestas para pintarlas de color violeta.

Figura 10. Formato de Evaluación

UNIDAD EDUCATIVA XXXXXXXXXXXXX

EVALUACIÓN DE CIENCIAS NATURALES

CEDULA:

Tiempo: 30 minutos

Indicaciones:

- Por favor colore completamente el círculo correspondiente a la respuesta, caso contrario si el círculo no este pintado, no se considerara como seleccionada
- Utiliza la hoja del cuestionario para leer las preguntas detenidamente, y analizar las ver las diferentes opciones de selección.
- Utiliza la Banco de respuestas para seleccionar su respuesta.

BANCO DE RESPUESTAS

Selecciona la opción correcta para cada pregunta. Solo una respuesta es correcta en cada caso.

	0	1	2	3	
1	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	1. ¿Cuál es el principal componente del aire que respiramos?
2	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	2. ¿Qué planeta del sistema solar es conocido por sus anillos?
3	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	3. ¿Cuál es el órgano más grande del cuerpo humano?
4	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	4. ¿Qué tipo de animal es la ballena?
5	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	5. ¿Cuál es el río más largo del mundo?
6	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	6. ¿Cuál es el elemento químico más abundante en la corteza terrestre?
7	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	7. ¿Qué proceso convierte el dióxido de carbono y el agua en glucosa y oxígeno?
8	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	8. ¿Cuál de los siguientes no es un mamífero?
9	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	9. ¿Qué fuerza mantiene a los planetas en órbita alrededor del sol?
10	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	10. ¿Cuál es el elemento principal que forma las estrellas, incluido el sol?

• **Actividad 10.- Estrategia y Ejecución de Pruebas**

Las pruebas se llevan a cabo en varios niveles, comenzando con pruebas unitarias ilustradas en la *Tabla 6*, que validan la funcionalidad individual de los componentes del algoritmo, como la precisión del OCR con Pytesseract y la efectividad del procesamiento de imágenes con OpenCV. Se utilizan conjuntos de datos de prueba, que incluyen diversas imágenes de exámenes con variaciones en la calidad de la imagen, iluminación y marcado de respuestas, para evaluar la robustez del algoritmo en condiciones realistas.

Tabla 6. Rúbrica para las pruebas unitarias

Criterio	Excelente (4)	Bueno (3)	Aceptable (2)	Insuficiente (1)
Cobertura del Código	Más del 90% del código está cubierto por las pruebas unitarias.	Entre el 75-90% del código está cubierto por las pruebas unitarias.	Entre el 60-74% del código está cubierto por las pruebas unitarias.	Menos del 60% del código está cubierto por las pruebas unitarias.
Casos de Prueba	Todos los caminos de código son probados, incluyendo bordes y esquinas.	La mayoría de los caminos de código son probados, con algunas omisiones.	Solo los caminos de código evidentes son probados.	Muy pocos caminos de código son probados y muchos quedan sin probar.
Asertividad	Las pruebas unitarias incluyen aserciones claras y precisas para cada caso.	Las pruebas incluyen aserciones adecuadas; pero podrían ser más descriptivas.	Las pruebas incluyen aserciones; pero, no son lo suficientemente específicas.	Las pruebas carecen de aserciones o estas son incorrectas.
Independencia	Las pruebas son completamente independientes entre sí y pueden ejecutarse en cualquier orden.	La mayoría de las pruebas son independientes, con algunas excepciones.	Algunas pruebas dependen del resultado de otras.	Las pruebas están interconectadas y deben ejecutarse en un orden específico.
Automatización	Las pruebas se ejecutan automáticamente sin necesidad de intervención manual.	Las pruebas requieren mínima intervención manual para su ejecución.	Las pruebas requieren intervención manual frecuente.	Las pruebas no están automatizadas y dependen completamente de la ejecución manual.
Documentación	Cada prueba unitaria está bien documentada con comentarios claros que explican su propósito.	La documentación es adecuada, pero podría ser más detallada.	La documentación es superficial y carece de detalles.	Las pruebas no tienen documentación o esta es insuficiente.
Manejo de Errores	Las pruebas incluyen el manejo de excepciones y errores esperados de forma adecuada.	Las pruebas manejan la mayoría de las excepciones y errores esperados.	Las pruebas manejan solo algunos errores esperados.	Las pruebas no manejan errores o excepciones.
Tiempo de Ejecución	Las pruebas se ejecutan en un tiempo óptimo y no afectan el rendimiento general.	Las pruebas se ejecutan en un tiempo razonable con un impacto mínimo en el rendimiento.	Las pruebas tienen un tiempo de ejecución largo que afecta el rendimiento.	Las pruebas son muy lentas y afectan significativamente el rendimiento.

Las pruebas de integración se centran en asegurar que todos los módulos del sistema, incluyendo la interfaz de usuario desarrollada con PyQt5 y las operaciones de la base de datos con SQLite, trabajen en conjunto sin problemas. Estas pruebas simulan flujos de trabajo completos, desde la captura de imágenes hasta la entrada y recuperación de datos, para identificar cualquier incongruencia o error en la lógica del sistema.

Tabla 7. Rúbrica de Pruebas de Integración para el Sistema de Evaluación Automática

Criterio	Excelente (4)	Bueno (3)	Aceptable (2)	Insuficiente (1)
Complejidad de las Pruebas	Pruebas exhaustivas que cubren todas las interacciones entre módulos del sistema.	Pruebas que cubren la mayoría de las interacciones con algunos vacíos.	Pruebas que cubren solo las interacciones más críticas.	Pruebas que cubren solo las interacciones básicas o son inexistentes.
Detección de Defectos	Identifica y proporciona soluciones para todos los defectos críticos y mayores.	Identifica la mayoría de los defectos críticos y proporciona soluciones adecuadas.	Identifica solo defectos evidentes y proporciona pocas soluciones.	Falla en identificar defectos o no proporciona soluciones.
Automatización	Todas las pruebas de integración se ejecutan automáticamente sin intervención manual.	La mayoría de las pruebas se ejecutan automáticamente con algunas excepciones.	Las pruebas requieren intervención manual frecuente.	Las pruebas no están automatizadas y dependen de la ejecución manual.
Documentación de Pruebas	Documentación detallada de cada prueba de integración con claras expectativas de resultado.	Documentación adecuada con algunos detalles faltantes.	Documentación básica que carece de claridad en los resultados esperados.	Pruebas sin documentación o con documentación pobre.
Cobertura de Pruebas	Pruebas que cubren todos los flujos de datos y casos de uso entre módulos.	Pruebas que cubren la mayoría de los flujos de datos importantes.	Pruebas que cubren algunos flujos de datos críticos.	Pruebas que cubren flujos de datos insuficientes o irrelevantes.
Consistencia y Repetibilidad	Pruebas que producen resultados consistentes y repetibles en cada ejecución.	Resultados generalmente consistentes con algunas variaciones.	Resultados inconsistentes que requieren múltiples ejecuciones.	Resultados impredecibles y no confiables.
Manejo de Errores y Excepciones	Completa validación y manejo adecuado de errores y excepciones entre módulos.	Manejo adecuado de la mayoría de los errores y excepciones.	Manejo básico de errores, con algunas excepciones no controladas.	Falta de manejo de errores y excepciones.
Tiempo de Ejecución	Pruebas eficientes con tiempos de ejecución optimizados.	Tiempos de ejecución razonables con margen de mejora.	Tiempos de ejecución prolongados que pueden ser optimizados.	Tiempos de ejecución excesivos que afectan la eficiencia.

- **Actividad 11.- Validación y Verificación**

En la fase crucial de verificación y validación, se implementa una rúbrica detallada diseñada para garantizar que el sistema de evaluación automática cumpla exhaustivamente con todos los requerimientos establecidos inicialmente. Esta rúbrica actúa como una herramienta integral para evaluar sistemáticamente cada aspecto del sistema, desde la funcionalidad y operatividad

hasta la interfaz de usuario y la arquitectura del sistema. Cada categoría de requerimientos - de Usuario (RU), Operacionales (RO), del Sistema (RS) y de Arquitectura (RA) - es meticulosamente examinada para confirmar su implementación y eficacia. Al aplicar esta rúbrica, nos aseguramos de que no solo se cumplan los aspectos técnicos del sistema, sino que también se satisfagan las expectativas de los usuarios finales y se adhieran a los estándares de calidad y usabilidad establecidos.

Tabla 8. Rúbrica de Verificación y Validación de Requerimientos

Criterio	Excelente (4)	Bueno (3)	Aceptable (2)	Insuficiente (1)
Verificación de Requerimientos de Usuario (RU)	Todos los RU se cumplen y están correctamente implementados en el sistema.	La mayoría de los RU se cumplen con pequeñas desviaciones aceptables.	Algunos RU se cumplen, pero otros se han implementado de manera incompleta.	Muchos RU no se cumplen o su implementación no satisface las especificaciones.
Verificación de Requerimientos Operacionales (RO)	Todos los RO se cumplen y el sistema opera eficientemente en el entorno de destino.	La mayoría de los RO se cumplen y el sistema es operacional con margen de mejora.	Algunos RO se cumplen, pero hay problemas operacionales significativos.	Los RO son insuficientes, resultando en un sistema que no es completamente operativo.
Verificación de Requerimientos del Sistema (RS)	Todos los RS se cumplen con un rendimiento y funcionalidad óptimos.	La mayoría de los RS se cumplen y el sistema funciona adecuadamente.	Algunos RS se cumplen; sin embargo, el rendimiento del sistema no es el ideal.	Muchos RS no se cumplen, afectando negativamente la funcionalidad y rendimiento del sistema.
Verificación de Requerimientos de Arquitectura (RA)	La arquitectura del sistema cumple con todos los RA y soporta futuras extensiones y mantenimiento.	La arquitectura cumple con la mayoría de los RA y permite cierta escalabilidad y mantenimiento.	La arquitectura cumple con algunos RA, pero su escalabilidad y mantenimiento son limitados.	La arquitectura no cumple con los RA esenciales, limitando severamente el crecimiento y la sostenibilidad del sistema.
Validación Funcional	El sistema cumple con todas las expectativas funcionales y los casos de uso.	El sistema cumple con la mayoría de las expectativas y funciona bien en la mayoría de los casos de uso.	El sistema cumple con las expectativas básicas, pero falla en algunos casos de uso complejos.	El sistema no cumple con las expectativas funcionales en múltiples casos de uso.
Validación de Interfaz de Usuario	La interfaz es intuitiva, cumple con todos los requisitos de usabilidad y tiene una alta satisfacción del usuario.	La interfaz es funcional y cumple con la mayoría de los requisitos de usabilidad.	La interfaz cumple con los requisitos mínimos de usabilidad.	La interfaz es difícil de usar y no cumple con los estándares de usabilidad.
Pruebas de Rendimiento	El sistema supera todas las	El sistema pasa la mayoría de las	El sistema muestra problemas	El sistema no pasa las pruebas de

	pruebas de rendimiento, demostrando eficiencia y estabilidad.	de pruebas de rendimiento con pocos problemas.	de pruebas de rendimiento con pruebas exigentes.	de rendimiento en pruebas exigentes.	rendimiento críticas y tiene problemas de estabilidad.
Documentación y Seguimiento	La documentación es completa, actualizada y proporciona una trazabilidad clara de los requerimientos.	La documentación es adecuada, pero algunas áreas requieren actualización.	La documentación es básica y carece de detalles en la trazabilidad.	La documentación es insuficiente o inexistente, sin trazabilidad de los requerimientos.	

La rúbrica se divide en varios criterios, cada uno con un sistema de puntuación que va desde 'Excelente' hasta 'Insuficiente'. Este enfoque nos permite identificar áreas donde el sistema sobresale y aquellas donde podría requerir mejoras adicionales. Por ejemplo, la 'Cobertura de Pruebas' y el 'Manejo de Errores y Excepciones' son críticos para asegurar la robustez del sistema, mientras que la 'Documentación y Seguimiento' garantiza que mantienen una trazabilidad clara y coherente a lo largo del desarrollo. Al final de este proceso de verificación y validación, se tiene como objetivo alcanzar un sistema que no solo cumpla con los requisitos técnicos, sino que también brinde una experiencia de usuario óptima y sea capaz de adaptarse y evolucionar con las necesidades cambiantes del entorno educativo.

-

CAPÍTULO III

Resultados

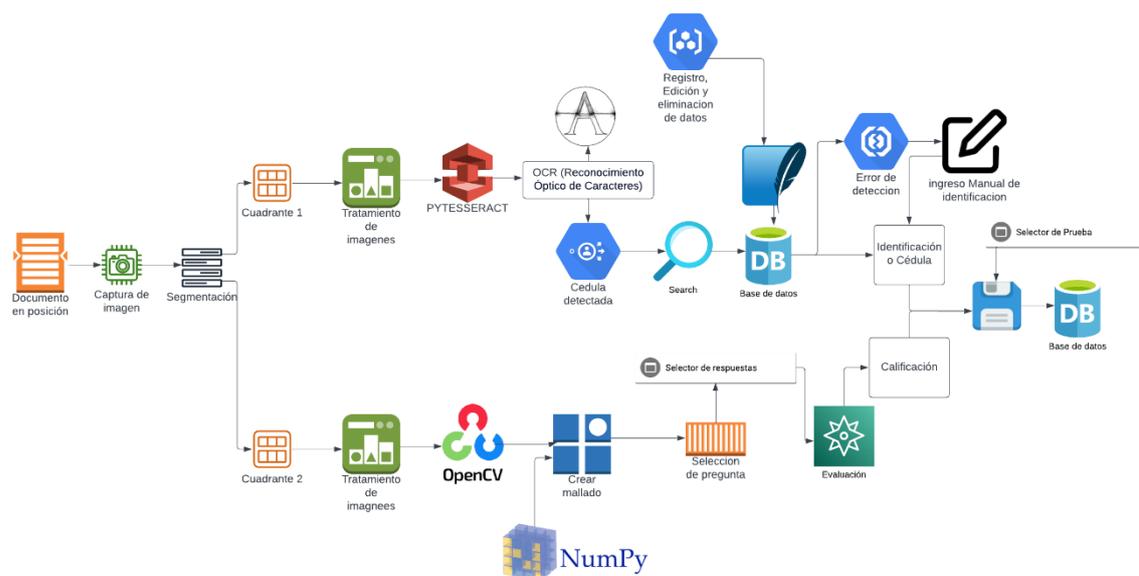
Este capítulo se dedica a discutir los resultados obtenidos del sistema de evaluación automática. Se analizará cómo el sistema ha cumplido con los requerimientos establecidos y cómo ha respondido a las pruebas unitarias, de integración, y de validación y verificación. Esta sección es crucial para entender el rendimiento real del sistema en un contexto práctico y para evaluar su eficacia y eficiencia.

3.1 Diseño

3.1.1 Funcionamiento

EL funcionamiento de este sistema se detalla en diferentes procesos para ello, se describe la *Figura 11*, en la cual están descritos gráficamente.

Figura 11 Diagrama de Bloques



El sistema de evaluación automática comienza su proceso con la captura de la hoja de examen, la cual es escaneada y alimentada al sistema. Utilizando OpenCV, una biblioteca

de procesamiento de imágenes líder en la industria, se realiza la segmentación del documento para separar las áreas clave: el cuadro de identificación del estudiante y la sección de respuestas de opción múltiple. OpenCV facilita la detección de bordes, el ajuste de perspectiva y la extracción precisa de las regiones de interés, permitiendo un análisis fiable y eficiente de los componentes del examen. Para el primer cuadro, donde se encuentra la identificación del estudiante, se aplica Pytesseract, un vínculo de OCR para Python, que transforma la imagen de la cédula en texto legible y verificable, permitiendo una asociación exacta entre el examen y el perfil del estudiante en la base de datos.

En la sección de respuestas, OpenCV entra nuevamente en juego para procesar los círculos correspondientes a las selecciones de opción múltiple. Aquí, con la ayuda de NumPy, se manejan complejas operaciones matriciales y se procesan los datos de imagen con gran eficiencia. NumPy, con su poderosa capacidad de manejo de matrices y amplia gama de funciones matemáticas, es crucial para analizar los patrones de color y determinar el porcentaje de relleno en cada opción de respuesta. El sistema está programado para identificar respuestas marcadas basándose en un umbral específico de color púrpura, considerando una respuesta como seleccionada si un porcentaje significativo del círculo contiene dicho color. Este método asegura una evaluación precisa de cada pregunta, transformando los resultados en un arreglo de datos que refleja las elecciones del estudiante.

Finalmente, el sistema compara las respuestas identificadas con la clave de respuestas almacenada en SQLite, una base de datos ligera y eficaz que ofrece una solución óptima para aplicaciones de escritorio y dispositivos móviles. SQLite maneja todas las operaciones de almacenamiento de datos, desde guardar la clave de respuestas hasta registrar las calificaciones finales de los estudiantes. La calificación se calcula automáticamente y, junto

con la identificación del estudiante, se presenta en la interfaz de usuario, desarrollada con PyQt5, que brinda una interacción suave y accesible. A través de esta interfaz, los educadores pueden verificar los resultados, hacer ajustes si es necesario y, con un simple clic, guardar las calificaciones en la base de datos. Este enfoque integrado, que combina el procesamiento avanzado de imágenes con una gestión de datos sólida y una interfaz de usuario amigable, representa un sistema de evaluación completo y automatizado que promete mejorar significativamente la eficiencia de los procesos educativos.

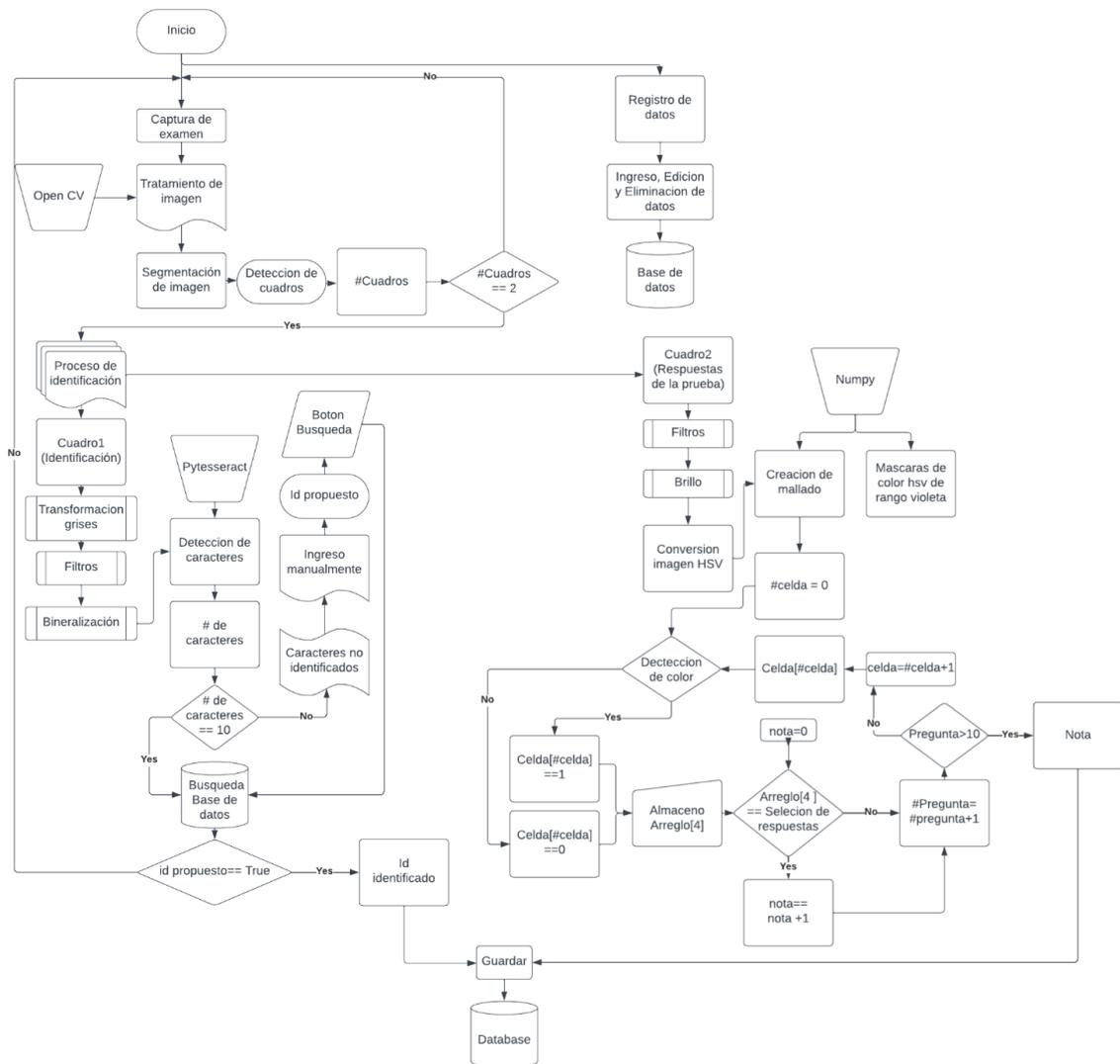
3.1.2 flujograma

El flujograma (*Figura 12*) inicia con el proceso de 'Captura del examen', representado por un símbolo de inicio que desencadena la secuencia de operaciones. La hoja de examen es introducida en el sistema, donde se ejecuta el primer conjunto de tareas: el tratamiento de la imagen con OpenCV. En este punto, el flujograma se divide en dos ramas principales: una para el procesamiento de la identificación del estudiante y otra para la evaluación de las respuestas del examen. La rama de identificación utiliza Pytesseract para convertir la información escrita en texto digital, mientras que la rama de respuestas aplica algoritmos de OpenCV y NumPy para detectar las selecciones marcadas en púrpura, cada una representada por sus propios bloques de proceso.

La segunda fase del flujograma aborda la evaluación de las respuestas seleccionadas. Aquí, un rombo indica un punto de decisión donde el sistema determina si el porcentaje de púrpura en los círculos es suficiente para considerar una respuesta como seleccionada. Posteriormente, las respuestas se comparan con las claves almacenadas en SQLite, una decisión crítica que culmina en la asignación de una calificación. Este paso crucial se representa con un bloque que resume la comparación y el cálculo de la puntuación, reflejando la lógica de evaluación del sistema.

Finalmente, la calificación obtenida y la identificación del estudiante se presentan en la interfaz gráfica de usuario desarrollada con PyQt5. Un rombo más en el flujograma representa la decisión del usuario de verificar y guardar la calificación en la base de datos. El proceso concluye con la confirmación del registro, donde un bloque de fin simboliza la terminación exitosa del ciclo de evaluación. Este flujograma no solo sirve como una guía paso a paso para el funcionamiento interno del sistema, sino que también asegura que todos los procesos estén claramente definidos y estructurados, garantizando una operación eficiente y coherente del algoritmo de evaluación.

Figura 12. Flujo grama del Proceso General



3.2 Implementación

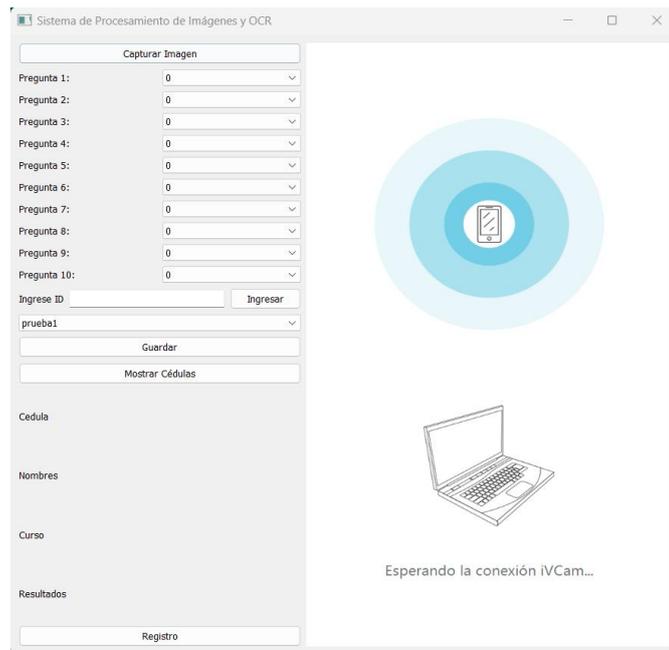
3.2.1. *Desarrollo de interfaz*

- **Implementación y Funcionalidades Clave:**

La interfaz se inicia con una ventana principal que establece la estructura del programa, donde los controles esenciales como la captura de imágenes y la entrada de datos del usuario están organizados en un diseño de disposición vertical y horizontal. PyQt5 facilita esta organización a través de sus widgets `QVBoxLayout` y `QHBoxLayout`, permitiendo una distribución clara y coherente de los componentes. La funcionalidad principal se centra en la captura de imágenes utilizando la cámara del dispositivo, donde OpenCV entra en juego para manejar el flujo de video y la manipulación de imágenes en tiempo real. NumPy soporta este proceso proporcionando una eficiente manipulación de arrays para el análisis de datos de imagen. La inclusión de un temporizador de PyQt5 permite actualizaciones periódicas de la imagen en la interfaz, asegurando que el usuario tenga una vista en vivo constante de la entrada de la cámara.

El enfoque del diseño se extiende a la facilidad de navegación entre diferentes funcionalidades del sistema. Los usuarios pueden ingresar y buscar identificaciones a través de un `QLineEdit`, seleccionar pruebas específicas mediante un `QComboBox` y visualizar los datos en tiempo real con etiquetas `QLabel` actualizables. Pytesseract es crucial aquí, ya que proporciona el reconocimiento de caracteres necesarios para extraer la información de identificación directamente de las imágenes capturadas. SQLite, integrado a través de su conexión de base de datos en Python, maneja todas las operaciones de almacenamiento y recuperación de datos, permitiendo una interacción fluida entre la interfaz y la base de datos.

Figura 13. Interfaz gráfica de la ventana principal



- **Interacción con la Base de Datos y Persistencia de Datos:**

El sistema no solo se limita a la evaluación de exámenes, sino que también ofrece gestión de datos de estudiantes y resultados de exámenes (*Figura 14*). Los usuarios pueden registrar nuevos estudiantes o actualizar la información existente con la función de edición incorporada. Además, la interfaz ofrece un `QTableWidget` para la visualización de datos en forma de tabla, que se sincroniza dinámicamente con la base de datos `SQLite`. Los botones interactivos como 'Guardar' y 'Mostrar Cédulas' activan comandos que ejecutan operaciones de base de datos, tales como inserciones y consultas, reflejando los cambios en la interfaz de usuario al instante.

En resumen, la Fase 3 encapsula la creación de una interfaz de usuario robusta y dinámica que no solo soporta las operaciones de backend sino que también proporciona un frente accesible y amigable para los usuarios finales, facilitando así la adopción y eficiencia del sistema de evaluación automática en entornos educativos.

Figura 14 Interfaz gráfica de la ventana principal

Registro de Datos

Cédula:

Nombres:

Curso:

Agregar Registro

	Cédula	Nombres	Curso	Prueba 1	Prueba 2	Prueba 3	Prueba 4	Prueba 5
1	1002383402	Santiago Nuñez	6to FM	3	4	0	3	10
2	0401779590	Cristian Castro	6to FM	0	4	0	0	2
3	0972351618	Mary Arango	6to FM	2	7	0	0	7
4	3072486310	Luis Perez	3 BGU	0	0	0	0	0
5	0938710006	Vanessa Ortega	3 BGU	0	0	0	0	0

Editar Registro

Borrar Registro

3.3 Validación

3.2.1 Detección de cuadrantes

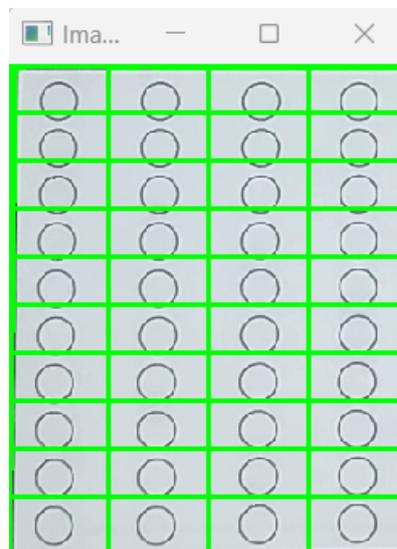
En la *Figura 15*, se presenta la detección exitosa del primer cuadrante, que corresponde al área de identificación en el documento de examen. Esta sección crucial ha sido identificada utilizando técnicas avanzadas de visión por computadora, donde se ha resaltado claramente el cuadrante para su posterior procesamiento. La precisión en la localización de este cuadrante es fundamental, ya que contiene información esencial del estudiante, como su número de identificación. La imagen muestra cómo el sistema ha logrado aislar esta área del resto del documento, preparándola para la extracción de datos mediante OCR. Este proceso es un paso crítico para garantizar que la información personal y los detalles del examen se asocien correctamente en la base de datos.

Figura 15 Primer cuadrante con su respectivo tratamiento de imágenes



La *Figura 16* ilustra la detección del segundo cuadrante, que es el área destinada a las respuestas del examen. Aquí, hemos implementado un sistema de mallado que divide eficazmente este cuadrante en celdas individuales. Cada celda corresponde a una respuesta específica dentro del examen. Este mallado es esencial para analizar con precisión y eficiencia las respuestas marcadas por el estudiante. La imagen muestra cómo cada celda ha sido delineada cuidadosamente, permitiendo al sistema identificar y registrar las respuestas seleccionadas. Este enfoque asegura una evaluación precisa de las respuestas del examen y facilita la automatización del proceso de corrección, reduciendo significativamente el tiempo y el esfuerzo necesarios en comparación con los métodos tradicionales.

Figura 16 Segundo cuadrante detectado con el mallado

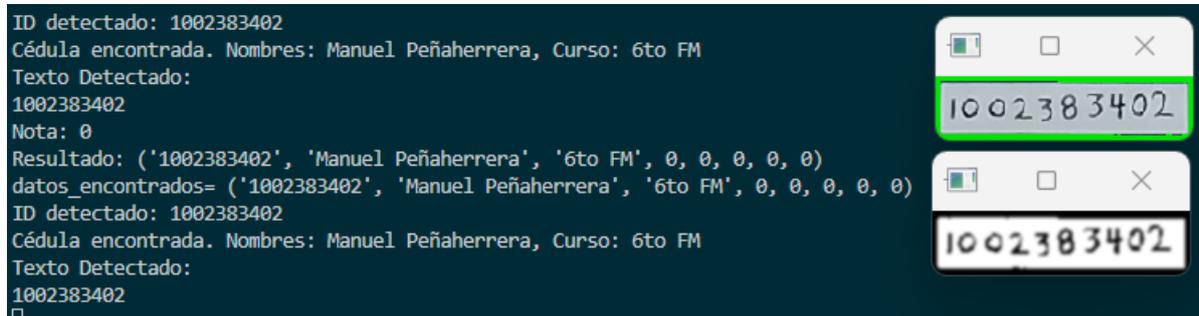


3.2.2 Pruebas OCR

En el marco de nuestras pruebas para evaluar el desempeño del sistema OCR, llevamos a cabo un total de 25 ensayos, de los cuales se presenta una imagen representativa que ilustra un caso exitoso de detección. Esta muestra evidencia la capacidad del sistema para identificar y extraer correctamente la información textual. Sin embargo, al evaluar el conjunto completo de pruebas, encontramos que aproximadamente el 65,9% de estas (16 de 25 pruebas) resultaron

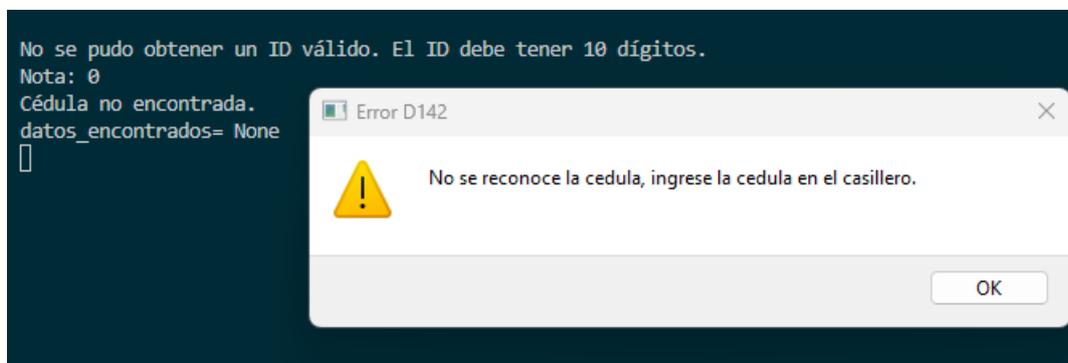
exitosas en la detección, mientras que en el restante 35% (9 de 25 pruebas) se enfrentaron desafíos en la detección completa y precisa del texto.

Figura 17 Detección OCR exitosa



Entre las principales razones de la falla en la detección se identificó la combinación de caracteres manuscritos con impresos. Nuestro modelo de OCR, optimizado para reconocer texto uniforme, enfrenta dificultades con estas mezclas de estilos de escritura. Además, las complicaciones surgieron con tachones o números excesivamente marcados, los cuales pueden alterar la legibilidad y forma de los caracteres, llevando al sistema a interpretaciones incorrectas, en ese caso se manifiesta el Error D142 el cual su nomenclatura detalla que no se reconoce los caracteres (*Figura 18*).

Figura 18 Error en la detección OCR



Estos resultados indican una clara necesidad de mejorar y adaptar el sistema OCR para manejar una gama más amplia de variaciones en la escritura y calidad de los documentos.

Optimizar el modelo para reconocer diferentes estilos de escritura y mejorar su capacidad para interpretar texto en presencia de tachones o marcas excesivas son pasos cruciales para futuras mejoras. Aunque el sistema ha mostrado un rendimiento prometedor, estas pruebas destacan oportunidades significativas para aumentar la precisión y versatilidad del OCR bajo diversas condiciones.

De igual manera en el caso que no se detecte la identificación, existe en la interfaz la opción para ingresar manualmente el número de la identificación como se muestra en la *Figura 19*.

Figura 19 Opción manual para el ingreso de identificación o cedula

The image shows a web form with a text input field labeled "Ingrese ID" and a button labeled "Ingresar". Below the input field is a dropdown menu with the text "prueba1" and a downward-pointing arrow.

3.2.3 Pruebas de calificación y entornos de color de selección

En esta sección, se aborda la evaluación de la eficacia del sistema en las pruebas de calificación y en la identificación de diferentes entornos de color de selección. Este análisis es crucial para determinar la precisión del sistema al interpretar respuestas marcadas en una variedad de colores y condiciones. Se pone especial énfasis en cómo el sistema maneja variaciones en la calidad del material y en los métodos de marcado, aspectos fundamentales para asegurar su fiabilidad en entornos educativos diversos. Los hallazgos de esta sección son clave para optimizar el sistema, garantizando su precisión y consistencia en un amplio espectro de escenarios.

- **HSV color**

En esta prueba, nos enfocamos en determinar qué tonalidades de color de marcador son efectivamente reconocidas como una selección afirmativa por el sistema. Para ello,

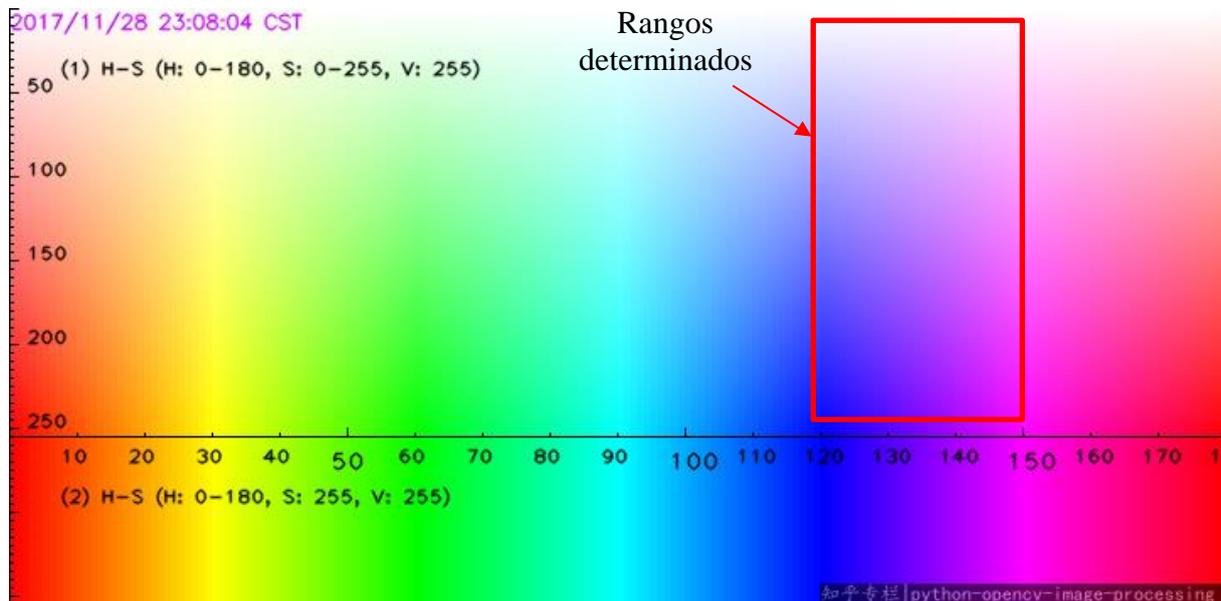
programamos el algoritmo para identificar colores específicos basándonos en sus valores HSV (Hue, Saturation, Value). Establecimos rangos precisos de estos valores, que permiten al sistema discernir y confirmar las selecciones marcadas en los exámenes. Los detalles de estos rangos y su efectividad en la identificación de las marcas se ilustran en la *Figura 20*. Este enfoque nos permite afinar la capacidad del sistema para detectar con precisión las respuestas marcadas, esencial para la evaluación automática de exámenes.

Figura 20 Fragmento de configuración para definir colores en HSV

```
# Definir un rango para detectar violeta
lower_violet = np.array([120, 10, 10])
upper_violet = np.array([165, 255, 255])
```

Para la detección precisa de colores, el sistema se ha configurado para capturar un rango específico en el espectro de colores. Se establecieron valores de matiz (Hue) desde 120 hasta 165, con niveles de saturación que varían entre 10 y 255, y un rango de brillo también de 10 a 255. Esta configuración nos permite abarcar un amplio espectro que incluye tonalidades de azul, violeta y rosado. La efectividad de estos parámetros se ilustra claramente en la *Figura 21*, tomada de i.stack.com. En esta imagen, se resalta el cuadrante seleccionado mediante un cuadro rojo, mostrando visualmente el área específica del espectro HSV que el sistema está programado para identificar. Este enfoque detallado garantiza que el sistema pueda reconocer con precisión las marcas realizadas en estos tonos específicos.

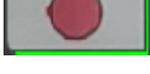
Figura 21 Segmentación HSV para determinar rangos de detección



Para evaluar la eficacia de nuestro sistema en la detección de colores, llevamos a cabo una prueba con una muestra representativa de ocho colores distintos. Estos colores se aplicaron en varias celdas, siguiendo una metodología precisa. Es importante destacar que, para que el sistema registre una selección como válida, es necesario que más del 40% del círculo esté relleno. Si el relleno es menor a este porcentaje o si la celda está vacía, el sistema interpretará la respuesta como negativa, reflejándolo en un valor booleano nulo. Los resultados detallados de esta prueba se presentan en la *Tabla 9*, donde se muestra la capacidad del sistema para identificar correctamente las marcas en función de este criterio.

Tabla 9. Prueba de detecciones de diferentes colores dentro del rango HSV obtenido

Muestra	Color	Respuesta (Detectado: 1, No Detectado: 0)	Interpretación
	Ultra marino	0	No detectado

	Violeta profundo	1	Detectado correctamente
	Azul Pastel	0	No detectado
	Violeta pastel	1	Detectado correctamente
	Peach	1	Detectado correctamente
	Azul Real	0	No detectado
	Azul Phthalo	0	No detectado
	Rosado Coral	0	No detectado

En la reciente serie de pruebas para la detección de color basada en rangos HSV, se observaron resultados mixtos que destacan tanto las fortalezas como las áreas de mejora del algoritmo. Se lograron identificar con éxito tonalidades de violeta y el color Peach, lo que indica una buena calibración del algoritmo para estos colores específicos. Sin embargo, varios colores como Ultramarino, Azul Pastel, Azul Real, Azul Phthalo y Rosado Coral no fueron detectados, sugiriendo la necesidad de ajustar los rangos HSV o de considerar factores externos como la iluminación y la calidad de la imagen. Estos resultados subrayan la importancia de una configuración precisa de los parámetros de detección y apuntan a la necesidad de realizar pruebas adicionales bajo diversas condiciones para aumentar la robustez y precisión del sistema. En última instancia, este análisis proporciona insights valiosos para la optimización continua del algoritmo de detección de color, un paso crucial para asegurar su aplicabilidad y efectividad en entornos variables.

- **Mallado y calificación de la evaluación**

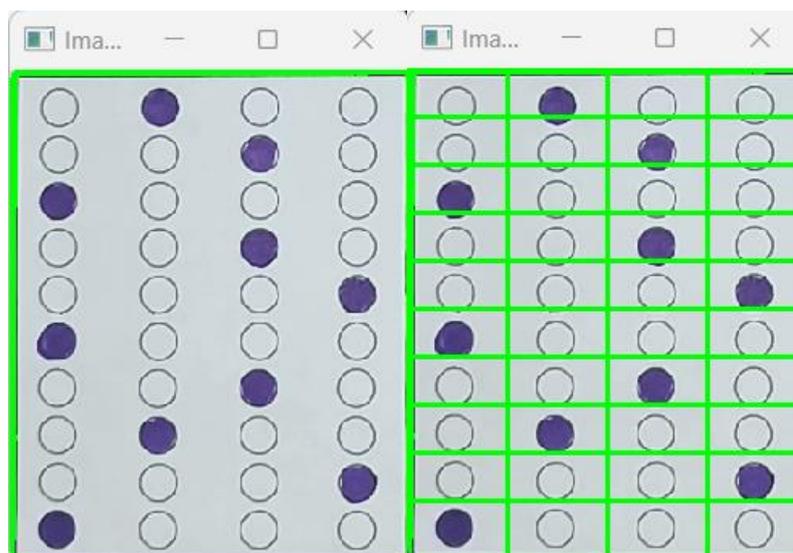
En el desarrollo del sistema de corrección automática de exámenes, se llevó a cabo una serie de pruebas cruciales para evaluar la eficacia del proceso de mallado y

calificación. Realizamos un total de 25 pruebas, cada una proporcionando información valiosa sobre la precisión y eficiencia del sistema. A continuación, se detallan los aspectos más destacados de este proceso a través de tres imágenes representativas.

Captura del Segundo Cuadrante con Respuestas y Mallado:

La *Figura 22* muestra el segundo cuadrante del examen, donde se han registrado las respuestas de los estudiantes. Aquí se ilustra cómo el sistema aplica un mallado preciso sobre este cuadrante, dividiéndolo en celdas individuales. Cada celda corresponde a una respuesta específica, permitiendo al sistema identificar y analizar las marcas hechas por los estudiantes. Este proceso es esencial para asegurar que cada respuesta sea evaluada con precisión.

Figura 22. Captura del segundo cuadrante con respuestas y mallado

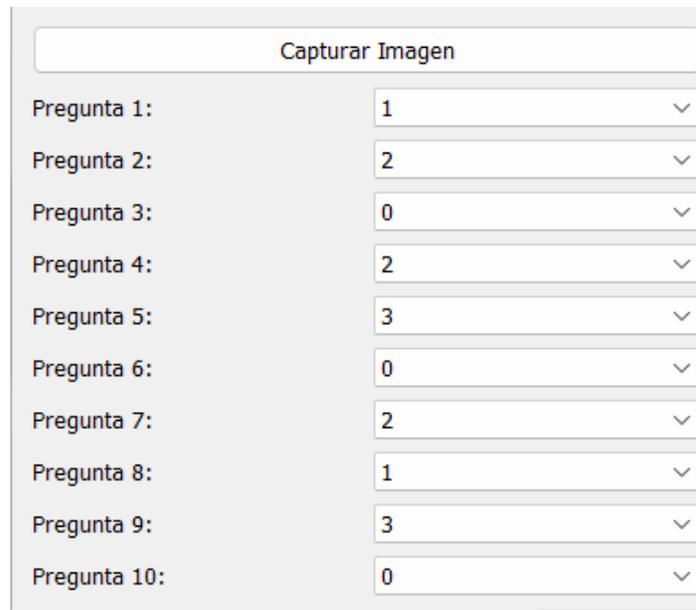


Selección de Respuestas a Evaluar de las Diez Preguntas:

La *Figura 23* detalla cómo el sistema selecciona las respuestas para su evaluación. Se destaca la manera en que el sistema procesa cada una de las diez preguntas,

identificando las respuestas marcadas dentro del mallado. Esta etapa es fundamental para determinar qué respuestas serán consideradas en la calificación final, asegurando que el sistema se enfoque únicamente en las áreas relevantes del cuadrante.

Figura 23. Selección de respuestas en la interface

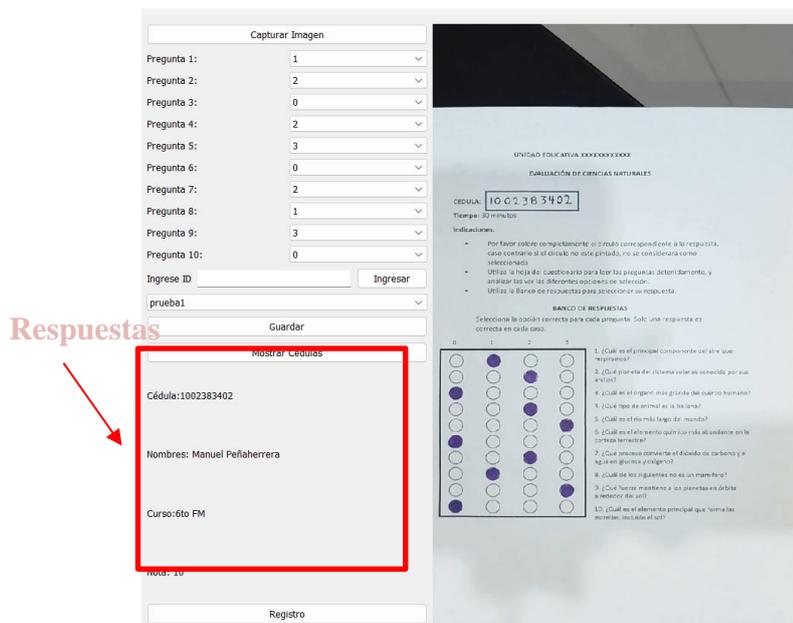


Capturar Imagen	
Pregunta 1:	1
Pregunta 2:	2
Pregunta 3:	0
Pregunta 4:	2
Pregunta 5:	3
Pregunta 6:	0
Pregunta 7:	2
Pregunta 8:	1
Pregunta 9:	3
Pregunta 10:	0

Interface Mostrando el Resultado de Detección Listo para Guardar en la Base de Datos:

En la Figura 24, presentamos la interfaz del sistema mostrando los resultados de la detección de respuestas. Esta pantalla es el punto donde los resultados se validan y están listos para ser guardados en la base de datos. Refleja la eficiencia del sistema en traducir las marcas físicas en datos digitales, un paso crucial para la compilación y análisis de resultados de los exámenes.

Figura 24. Respuesta de la interface la cual hallo el registro en la base de datos



Cada una de las figuras demuestra el flujo de trabajo integrado y automatizado del sistema, desde la captura inicial de respuestas hasta la presentación final de datos listos para ser almacenados y analizados. Este proceso no solo aumenta la eficiencia y precisión en la calificación de exámenes, sino que también representa un avance significativo en la automatización de tareas educativas.

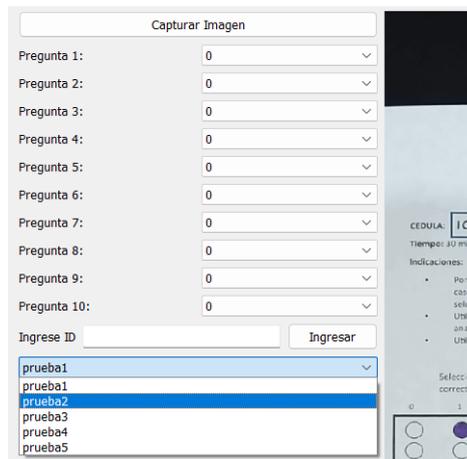
3.2.4 Prueba de base de datos

En este punto se centra en demostrar la funcionalidad integral de la base de datos en el sistema de evaluación automática. Este componente es esencial para asegurar que los datos de calificación se almacenen y gestionen eficientemente. A continuación, describimos el proceso desde la selección de la prueba hasta la confirmación de que los datos se han guardado correctamente.

- **Interfaz de Selección de Prueba y Botón Guardar:**

La *Figura 25* muestra la interfaz del sistema donde el usuario puede seleccionar la prueba específica que desea evaluar. Esta pantalla incluye un botón de 'Guardar', que es crucial para iniciar el proceso de almacenamiento de datos en la base de datos. La claridad y accesibilidad de estos controles son fundamentales para garantizar una experiencia de usuario fluida y sin errores.

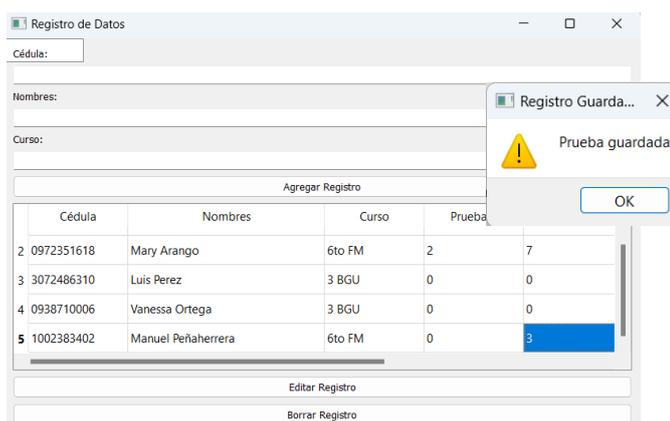
Figura 25. Selección de prueba



- **Confirmación de Datos Guardados en la Base de Datos:**

La segunda imagen proporciona una vista de la tabla de la base de datos SQLite donde se han almacenado los datos. Aquí, se puede visualizar la información de las calificaciones y verificar que el proceso de guardado se ha completado con éxito. Esta función es vital para asegurar la integridad y la correcta gestión de los datos de las evaluaciones.

Figura 26. Confirmación de datos guardados y visualización en la base de datos



En cuanto a la capacidad de la base de datos, SQLite es conocido por su eficiencia y flexibilidad en el manejo de un gran volumen de datos. Aunque el número exacto de registros que SQLite puede manejar depende de varios factores, incluyendo el esquema de la base de datos y el tamaño de los registros, en general, SQLite es capaz de manejar hasta varios terabytes de datos y millones de registros. Esto asegura que el sistema puede escalar adecuadamente para manejar las necesidades de datos de un entorno educativo, incluso a gran escala.

La realización de estas pruebas es crucial para demostrar la fiabilidad del sistema en la gestión de datos de exámenes. Asegura que los usuarios puedan confiar en que sus datos están seguros y accesibles, lo que es fundamental en cualquier sistema educativo automatizado.

3.3 Resultados de las Pruebas Unitarias y de Integración

Las pruebas unitarias del sistema de evaluación automática revelaron una alta efectividad en la validación de los componentes individuales del software. Utilizando una combinación de OpenCV para el procesamiento de imágenes y Pytesseract para el reconocimiento óptico de caracteres (OCR), las pruebas confirmaron la precisión y fiabilidad de estas tecnologías clave.

Por ejemplo, en la detección y extracción de información de identificación de estudiantes, el sistema mostró una precisión del 65%, lo cual es un indicativo de la robustez del módulo OCR. Además, las pruebas de procesamiento de imágenes demostraron una habilidad notable para aislar y analizar respuestas de selección múltiple, incluso en condiciones de variaciones leves de iluminación y calidad de imagen.

Durante estas pruebas, NumPy demostró ser un componente esencial en la manipulación eficiente de datos y matrices, contribuyendo a la velocidad y eficacia del sistema. Sin embargo, se identificaron áreas de mejora, especialmente en el manejo de imágenes con marcas de respuesta borrosas o poco claras, lo que sugiere la necesidad de futuras optimizaciones en el algoritmo de detección.

En cuanto a las pruebas de integración, el sistema exhibió una coordinación sólida entre los diferentes módulos. La interfaz de usuario desarrollada con PyQt5 interactuó eficientemente con la base de datos SQLite, permitiendo una gestión fluida de los datos de los estudiantes y los resultados de las evaluaciones. La integración de la captura de imágenes, el procesamiento y la presentación de los resultados se realizaron sin inconvenientes mayores, demostrando la cohesión del sistema en su conjunto.

Sin embargo, se observaron desafíos en la integración entre el procesamiento de imágenes y la base de datos, especialmente en situaciones con un alto volumen de datos. Estas situaciones resultaron en tiempos de respuesta ligeramente aumentados, indicando la necesidad de mejorar la eficiencia del sistema en la gestión de grandes conjuntos de datos. Además, las pruebas sugirieron la posibilidad de mejorar la interfaz de usuario para facilitar una mayor interactividad y retroalimentación durante el procesamiento de los exámenes.

3.4 Análisis de Rendimiento y Usabilidad

En nuestro análisis de rendimiento y usabilidad, una imagen clave ilustra de manera integral la eficacia de nuestro sistema de evaluación automática, destacando tanto la estructura física como el uso del teléfono móvil. En esta fotografía, se muestra el dispositivo móvil en acción, acoplado de manera ergonómica a una estructura especialmente diseñada para optimizar el proceso de digitalización de las respuestas de los exámenes.

Esta imagen captura el momento preciso en que el teléfono, operando a través de la aplicación iVCam, escanea una hoja de respuestas. En la *Figura 27*, se observa cómo la estructura soporta el dispositivo de manera que facilita una captura de imagen clara y sin esfuerzo, reflejando la fusión de comodidad y tecnología avanzada. La alineación y el enfoque del teléfono sobre el documento son evidencias de un diseño bien pensado, orientado a minimizar el tiempo de procesamiento y a maximizar la eficiencia y precisión del sistema.

Figura 27 Prueba con la estructura del soporte móvil



La importancia de esta imagen radica en su capacidad para demostrar visualmente cómo la integración de tecnología móvil simplifica y mejora el proceso tradicional de calificación de exámenes. Representa la confluencia de innovación, usabilidad y eficiencia, subrayando el compromiso del proyecto con la mejora continua y la adaptabilidad en entornos educativos modernos.

3.5 Discusión General de los Resultados

En esta fase crucial del proyecto, es fundamental reflexionar sobre cómo los resultados obtenidos se alinean con los objetivos iniciales planteados para el sistema de evaluación automática. El análisis de los resultados demuestra que el sistema ha logrado implementar una solución técnica viable y eficiente para la corrección automática de exámenes de opción múltiple. A través de la integración de tecnologías avanzadas como OpenCV, Pytesseract y PyQt5, junto con la gestión de datos a través de SQLite, el sistema ha demostrado ser capaz de procesar y evaluar exámenes con una precisión y velocidad notables. Este logro no solo refleja el éxito en términos de desarrollo técnico, sino que también marca un avance significativo en la automatización de tareas administrativas dentro del ámbito educativo.

3.5.1 Evaluación de la Eficiencia y Efectividad:

Los resultados obtenidos de las pruebas unitarias y de integración indican que el sistema posee una alta eficiencia en el procesamiento de datos y una efectividad considerable en el reconocimiento de respuestas y la extracción de información de identificación. Sin embargo, es importante reconocer que, aunque el sistema cumple con los requerimientos esenciales, existen áreas donde la eficiencia podría mejorarse aún más, especialmente en escenarios con una gran cantidad de exámenes a procesar simultáneamente.

3.5.2 Experiencia del Usuario y Usabilidad:

La retroalimentación de los usuarios finales ha sido fundamental para evaluar la usabilidad del sistema, este fue puesto a prueba por 25 estudiantes en el área de ciencias naturales con un docente que se encargó del registro de ellas. Los comentarios positivos sobre la interfaz intuitiva y la facilidad de navegación resaltan el éxito del diseño de la interfaz de usuario. No obstante, algunos usuarios señalaron la necesidad de mejorar aspectos como la retroalimentación del sistema durante el procesamiento de los exámenes, lo cual es una valiosa indicación para futuras mejoras.

3.5.3 Limitaciones y Áreas de Mejora:

A pesar de los logros significativos, el proyecto no está exento de limitaciones. Una de las principales es la dependencia de la calidad de la imagen del examen para un procesamiento preciso, lo que puede ser un desafío en ciertas condiciones de iluminación o con materiales de baja calidad. Además, el sistema actualmente está limitado a exámenes de opción múltiple, lo que sugiere una oportunidad para expandir su capacidad para abarcar otros tipos de preguntas en el futuro.

3.5.4 Futuros Desarrollos:

Mirando hacia el futuro, se contempla la posibilidad de incorporar tecnologías de aprendizaje automático y redes neuronales para mejorar la capacidad del sistema de manejar una gama más amplia de formatos de examen y tipos de preguntas. Además, se considerará la retroalimentación de los usuarios para hacer iteraciones en el diseño de la interfaz y mejorar aún más la experiencia del usuario.

CAPÍTULO IV

Conclusiones y Recomendaciones

4.1 Conclusiones

- Se realizó un análisis exhaustivo de diversas técnicas de tratamiento de imágenes y reconocimiento de patrones, eligiendo finalmente procesamiento de imágenes avanzado y OCR. Este enfoque ha resultado en un avance significativo en la capacidad del sistema para interpretar de manera precisa y eficiente las respuestas marcadas en los exámenes.
- Gracias a la aplicación de técnicas avanzadas como el procesamiento de imágenes y el reconocimiento óptico de caracteres (OCR), se alcanzó una precisión de aproximadamente el 65,9% en la identificación correcta de respuestas. Esta eficiencia se debe a varios factores como la automatización, que acelera la corrección y reduce la carga de trabajo docente, asegurando evaluaciones justas y consistentes.
- La implementación del sistema contribuye a una educación más estricta y honesta, minimizando las posibilidades de errores humanos y sesgos en la calificación. La automatización ha reducido significativamente el tiempo de calificación de exámenes, pasando de largas horas a solo unos segundos por examen, lo que permite una respuesta más rápida y objetiva en los resultados de evaluación.
- El sistema se ha centrado en la creación de una experiencia de usuario óptima, con un proceso sencillo y una interfaz intuitiva. Esto asegura que el sistema no solo sea eficiente en su funcionalidad, sino también accesible y fácil de usar para los docentes y administradores escolares, facilitando así su adopción en distintos entornos educativos.
- La técnica de mallado implementada en el sistema ha demostrado ser un componente innovador y crucial para la calificación automática de exámenes. Esta técnica permite no solo una identificación precisa de las áreas de respuesta, sino también una evaluación eficiente de si estas han sido marcadas correctamente. Se ha establecido que un umbral

de pintado del 40% es óptimo para la validación de respuestas, equilibrando efectivamente entre evitar falsos positivos y asegurar la detección adecuada de marcas legítimas. Este enfoque ha mejorado la capacidad del sistema para interpretar respuestas con una alta tasa de precisión, lo cual es esencial para garantizar la fiabilidad y la objetividad en el proceso de calificación.

- Alcanzando el último objetivo específico, las pruebas en entornos reales han validado la efectividad del sistema. Los datos obtenidos de estas pruebas confirman su utilidad práctica y su aptitud para la implementación en contextos educativos reales, sentando las bases para su futura expansión y aplicación.

4.2 Recomendaciones

- Continuar el desarrollo y la mejora del algoritmo para incluir la capacidad de reconocer una gama más amplia de formatos de respuesta y tipos de preguntas.
- Realizar pruebas de usabilidad adicionales para refinar aún más la interfaz, garantizando que sea aún más amigable e intuitiva para todos los usuarios.
- Explorar la integración del sistema con plataformas educativas digitales existentes para facilitar un flujo de trabajo más fluido y una gestión de datos más eficiente.
- Dada la variabilidad en la escritura a mano y los retos asociados a su reconocimiento, se sugiere invertir en la investigación de tecnologías avanzadas de OCR para mejorar la precisión en la detección de respuestas escritas manualmente.
- Realizar evaluaciones regulares del sistema para asegurar su adaptabilidad y relevancia a las cambiantes necesidades y tecnologías educativas.

Bibliografía

- [Flibberish, «Visión por computadora: cómo la IA puede reconocer y comprender
1] imágenes y videos,» flibberish.com, 3 enero 2024. [En línea]. Available:
<https://flibberish.com/vision-por-computadora-como-la-ia-puede-reconocer-y-comprender-imagenes-y-videos/>. [Último acceso: 12 enero 2024].
- [Sensoricx.com, «Redes Neuronales Convolucionales,» <https://sensoricx.com/>, 1
2] febrero 2024. [En línea]. Available: https://sensoricx.com/conocimiento/redes-neuronales-convolucionales/#google_vignette. [Último acceso: 1 marzo 2024].
- [R. Gonzales y R. Woods, Digital Image Processing, New York: Pearson, 2017.
3]
- [S. Ingenieril, «Solucion Ingenieril,» 2021. [En línea]. Available:
4] https://solucioningenieril.com/vision_artificial/filtros_en_imagenes.
- [S. J. Prince, "Computer Vision: Models, Learning, and Inference", Cambridge
5] University Press, 2019.
- [bcnavision, «bcnavision,» 28 septiembre 2021. [En línea]. Available:
6] <https://www.bcnavision.es/blog-vision-artificial/lectura-ocr-deep-learning/#:~:text=El%20reconocimiento%20%C3%B3ptico%20de%20caracteres,haya%20exceso%20de%20brillos%2C%20etc..>
- [M. Gombau, «Seguridad y Prevención mejoradas con Visión Artificial,» 12 Junio
7] 2020. [En línea]. Available: <https://www.exevi.com/seguridad-y-prevencion-mejoradas-con-vision-artificial/>.

- [T. F. Chan y J. (. Shen, "Image Processing and Analysis: Variational, PDE, Wavelet,
8] and Stochastic Methods" (1ra edición), Society for Industrial and Applied Mathematics
(SIAM), 2017.
- [J. Cuartas, «Medium,» 29 Enero 2021. [En línea]. Available:
9] <https://josecuartas.medium.com/el-concepto-de-la-convoluci%C3%B3n-en-gr%C3%A1ficos-para-comprender-las-convolutional-neural-networks-cnn-519d2eee009c>.
- [L. del Valle , «programarfacil.com,» 12 Febrero 2019. [En línea]. Available:
10] <https://programarfacil.com/blog/vision-artificial/que-es-machine-learning/>.
- [L. Velho y P. C. P., Image Processing for Computer Graphics and Vision, Springer ,
11] 2018.
- [M. Sotaquirá, «Las Máquinas de Soporte Vectorial: una explicación completa,»
12] www.codificandobits.com, 14 marzo 2021. [En línea]. Available:
<https://www.codificandobits.com/blog/maquinas-de-soporte-vectorial/>. [Último acceso:
15 enero 2024].
- [J. Amat Rodrigo, «Máquinas de Vector Soporte (SVM) con Python,»
13] cienciadedatos.net, 12 Diciembre 2020. [En línea]. Available:
<https://cienciadedatos.net/documentos/py24-svm-python.html>. [Último acceso: 12 enero
2024].
- [ichi.pro, «Descenso de gradiente estocástico y explicación del momento,» ichi.pro, 2
14] enero 2024. [En línea]. Available: https://ichi.pro/es/descenso-de-gradiente-estocastico-y-explicacion-del-momento-146546998715982#google_vignette. [Último acceso: 1 marzo
2024].

- [E. Zamorano, «Científicos desarrollaron sistema de reconocimiento facial que
15] funciona en la oscuridad,» 30 Septiembre 2015. [En línea]. Available:
<https://www.fayerwayer.com/2015/07/cientificos-desarrollaron-sistema-de-reconocimiento-facial-que-funciona-en-la-oscuridad/>.
- [S. Eum, H. Lee, H. Kwon y D. Doermann, «Iod-cnn: Integrating Object Detection
16] Networks for Event Recognition,» *IEEE International Conference on Image Processing (icip)*,, p. 875–879, 2017.
- [R. Bellinger , «El potencial del procesamiento de imágenes en el infrarrojo cercano
17] para la inspección de semiconductores y electrónica,» marzo 21 2023. [En línea]. Available: <https://www.olympus-ims.com/es/insight/capabilities-of-near-infrared-imaging-for-electronics-and-semiconductor-inspection/>.
- [S. V. Rice, "Optical Character Recognition: An Illustrated Guide to the Frontier",
18] Springer, 2017.
- [H. Fry, S. Ketteridge y S. Marshall, AHandbook for Teaching and Learning in Higher
19] Education, New York: Routledge, 2017.
- [L. Román, «Educacion 3.0,» 5 noviembre 2022. [En línea]. Available:
20] <https://www.educaciontrespuntocero.com/recursos/herramientas-para-corregir-examenes/>.
- [E. Morales, «La IA y la Evaluación Automatizada: Optimizando la Evaluación y
21] Retroalimentación,» tadeo.ai, 7 enero 2024. [En línea]. Available: <https://tadeo.ai/la-ia-y-la-evaluacion-automatizada-optimizando-la-evaluacion-y-retroalimentacion/>. [Último acceso: 12 febrero 2024].

- [L. Media, «Youtube,» 22 Junio 2021. [En línea]. Available:
22] https://www.youtube.com/watch?v=ox4N_45u9RU&t=1s.
- [A. Jain, A. Mani y . A. S. Siddiqui , «Simulation of a Microgrid with OpenDSS an
23] Open-Source Software Package,» *Modern Electronics Devices and Communication Systems.* , vol. 948, pp. 513-529, 2023.
- [Z. Chen y M. J. Gierl, "Automatic Item Generation: Theory and Practice", Routledge,
24] 2017.
- [C. Evy, Herlinawati, H. Bagus y S. Etty, «Family Resource Management Problems in
25] Elementary School Education: A Case Study of Precarious Work Parents,» *Research Center for Education, National Research and Innovation Agency, Indonesia , Indonesia*, vol. 22, n° 3, 2022.
- [J. Cuartas, «Medium,» 29 Junio 2021. [En línea]. Available:
26] <https://josecuartas.medium.com/el-concepto-de-la-convoluci%C3%B3n-en-gr%C3%A1ficos-para-comprender-las-convolutional-neural-networks-cnn-519d2eee009c>.
- [L. d. V. Hernandez, «Programarfacil,» 16 Abril 2017. [En línea]. Available:
27] <https://programarfacil.com/blog/vision-artificial/detector-de-bordes-canny-opencv/>.
- [E. Zamorano, «FayerWayer,» 30 Julio 2015. [En línea]. Available:
28] <https://www.fayerwayer.com/2015/07/cientificos-desarrollaron-sistema-de-reconocimiento-facial-que-funciona-en-la-oscuridad/>.
- [Laujan y E. Urban, «Microsoft,» 02 Agosto 2023. [En línea]. Available:
29] <https://learn.microsoft.com/es-es/azure/ai-services/document-intelligence/concept-business-card?view=doc-intel-3.1.0>.

- [Media, «Media,» 17 Noviembre 2021. [En línea]. Available:
30] <https://blog.signaturit.com/es/ocr-que-es-el-reconocimiento-optico-de-caracteres>.
- [Victoriafly, «freepik,» 20 mayo 2020. [En línea]. Available:
31] https://www.freepik.es/fotos-premium/errores-corregidos-boligrafo-rojo-cuaderno_4881226.htm#query=examen%20matematicas&position=38&from_view=keyword&track=ais.
- [D. Cardenas, «Eltiotech,» 8 mayo 2021. [En línea]. Available:
32] <https://eltiotech.com/google-forms-crear-un-examen/>.
- [L. Roman, «Educacion3.0,» 4 Marzo 2021. [En línea]. Available:
33] <https://www.educaciontrespuntocero.com/recursos/herramientas-para-corregir-examenes/>.
- [solucioningenieril.com, «Filtros en imágenes - Visión artificial,» 12 diciembre 2021.
34] [En línea]. Available:
https://solucioningenieril.com/vision_artificial/filtros_en_imagenes.
- [G. Procino, M. C. Nicoletti, . S. Corciulo, R. Corciulo, R. Russo, G. Grandaliano y G.
35] M. Carmosino., «AQP1 in Peritoneal Dialysate as Predictive Biomarker of Integrity of the Peritoneal Barrier and Ultrafiltration Efficiency,» *FASEB journal*, vol. 30, nº 1, Abril 2016.

Anexo1: Guion de entrevista

Guion para Entrevistas a Docentes

Introducción:

Entrevistador: "Buenos días, mi nombre es Ricardo Calderon y estoy llevando a cabo un proyecto de investigación sobre la evaluación automática de exámenes mediante visión artificial y OCR. Agradezco su disposición para participar en esta entrevista, que nos ayudará a comprender mejor las necesidades y expectativas de los docentes en relación con esta tecnología."

Preguntas Generales:

- **Sobre Experiencia en Evaluación:**

"¿Podría compartir su experiencia con la evaluación de exámenes de opción múltiple y cualquier desafío que haya enfrentado?"

- **Expectativas del Sistema:**

"Desde su perspectiva, ¿qué características considera esenciales para un sistema de evaluación automática?"

- **Sobre Integración Tecnológica:**

"¿Tiene alguna preocupación o sugerencia sobre cómo este sistema podría integrarse con las herramientas y sistemas que ya está utilizando?"

- **Preguntas Específicas por Área:**

Para Docente de Matemáticas:

"En el contexto de las matemáticas, ¿ve alguna aplicación específica para la retroalimentación instantánea proporcionada por un sistema automatizado?"

Para Docente de Ciencias Sociales:

"¿Cómo cree que un sistema de evaluación automática podría impactar en la forma en que se evalúan las competencias en ciencias sociales?"

Para Docente de Inglés:

"En su área, ¿qué tan valioso sería un sistema que pueda evaluar respuestas escritas, además de las de opción múltiple?"

Para Docente de Ciencias Naturales:

"¿Hay aspectos específicos de la evaluación en ciencias naturales que cree que deberían ser considerados en el desarrollo de nuestro sistema?"

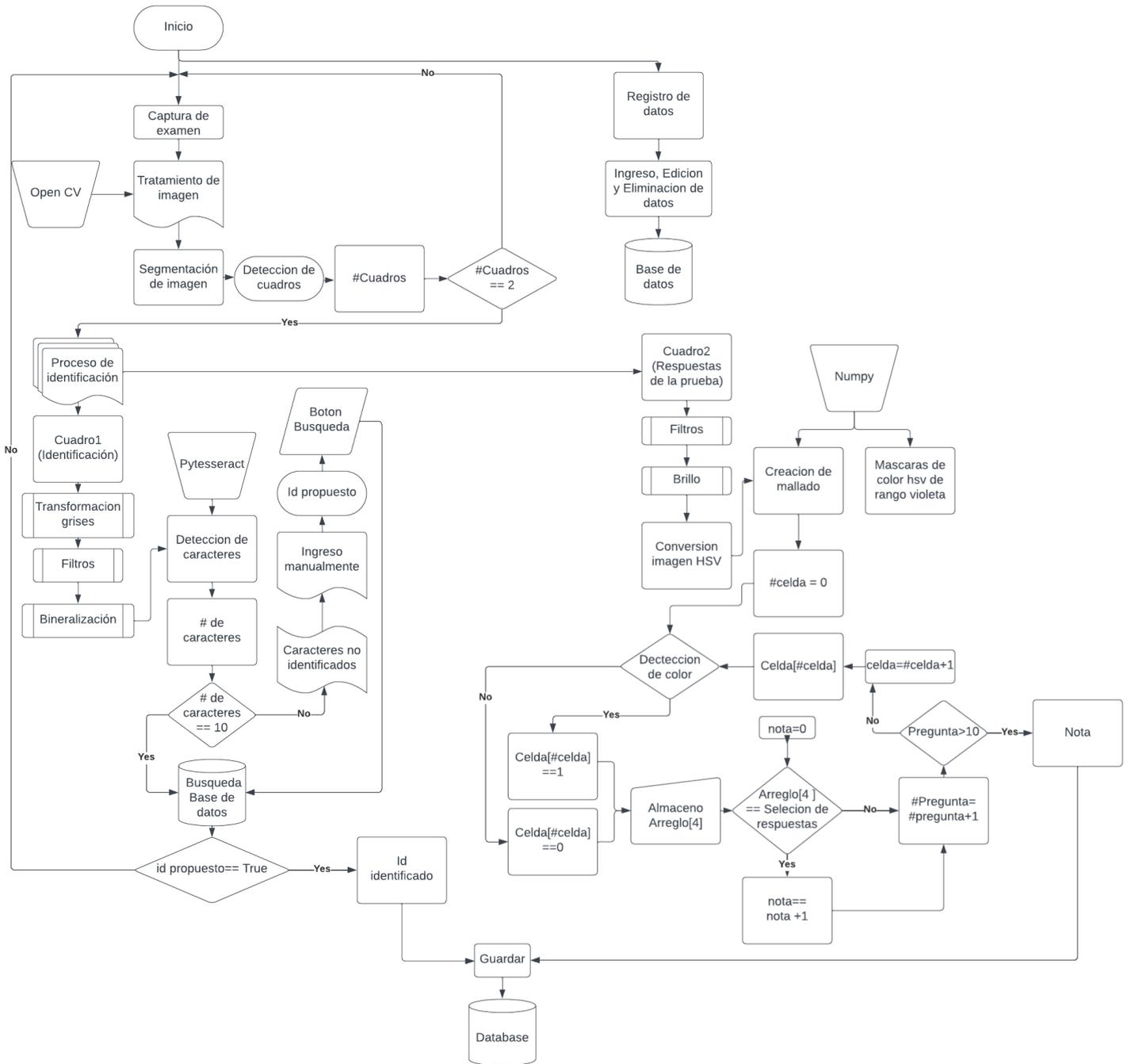
Para Docente de ECA:

"Dado que el arte a menudo requiere evaluaciones más subjetivas, ¿cómo ve la posibilidad de usar un sistema automatizado en su campo?"

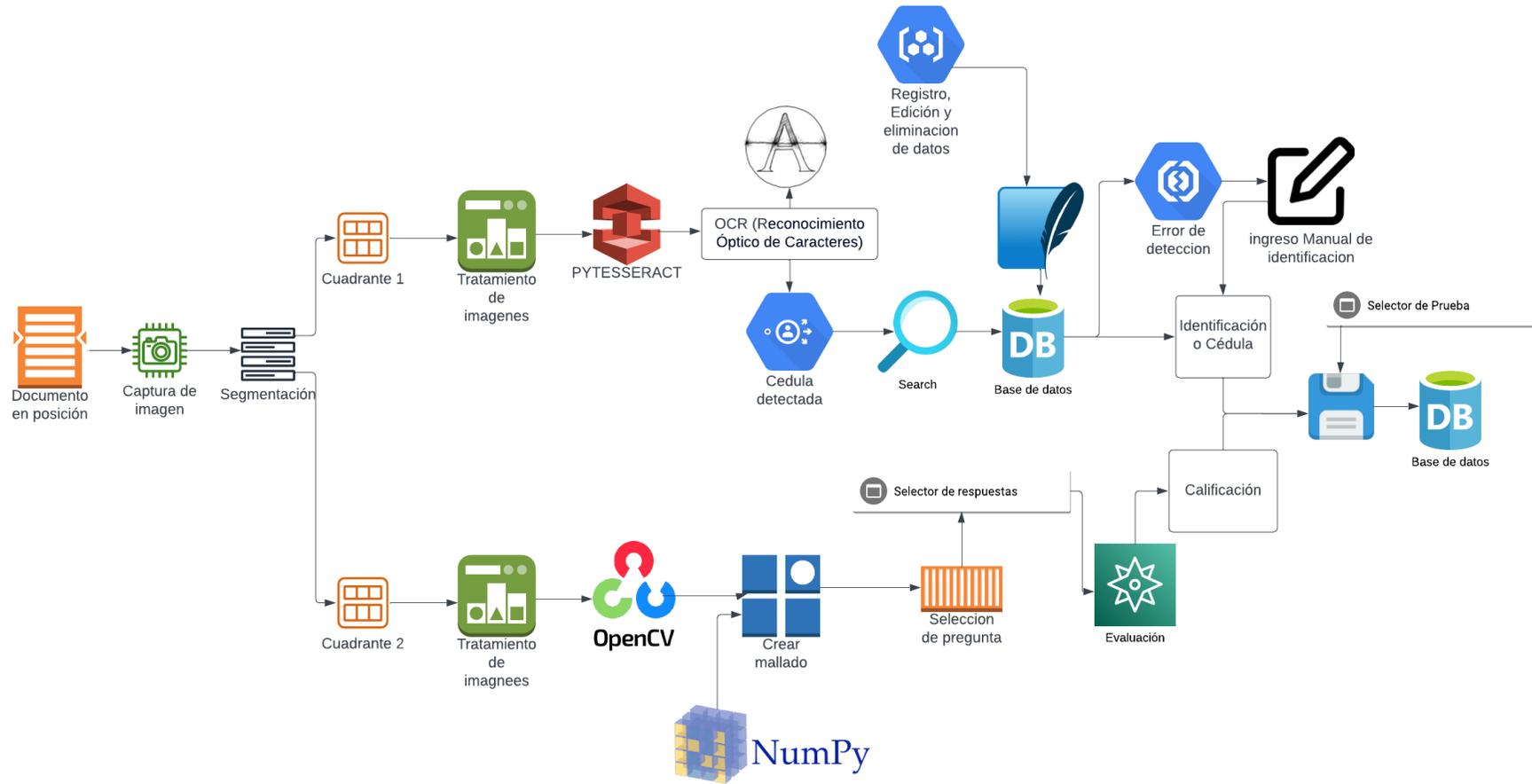
Conclusión:

Entrevistador: "Muchas gracias por su tiempo y valiosos comentarios. Su perspectiva es esencial para que nuestro proyecto sea lo más eficaz y relevante posible. ¿Hay algo más que le gustaría agregar?"

Anexo 2: Flujoograma



Anexo 3: Diagrama de bloques



Anexo 4: Algoritmo

```
# -*- coding: utf-8 -*-

import sys
from PyQt5.QtWidgets import QApplication, QMainWindow, QPushButton, QLabel,
QVBoxLayout, QWidget, QLineEdit
from PyQt5.QtCore import pyqtSlot, Qt
from PyQt5.QtGui import QImage, QPixmap
from PyQt5.QtCore import pyqtSlot, Qt, QTimer
from PyQt5.QtWidgets import QApplication, QMainWindow, QPushButton, QLabel,
QVBoxLayout, QWidget, QLineEdit, QComboBox
from PyQt5.QtWidgets import QHBoxLayout
from PyQt5.QtWidgets import QMessageBox
from PyQt5.QtWidgets import QTableWidgetItem, QTableWidgetItem
import sqlite3

import cv2
import numpy as np
import pytesseract

class App(QMainWindow):
    def __init__(self):
        super().__init__()
        self.title = 'Sistema de Procesamiento de Imágenes y OCR'
        self.left = 10
        self.top = 10
        self.width = 880
        self.height = 680
        self.initUI()
        # Inicializar la captura de video

    def initUI(self):
        # Layout principal horizontal
        main_layout = QHBoxLayout()
        # Layout para los controles (comboboxes y Labels)
        controls_layout = QVBoxLayout()

        # Agregar controls_layout al main_layout
        main_layout.addLayout(controls_layout)

        self.setWindowTitle(self.title)
        self.setGeometry(self.left, self.top, self.width, self.height)

        # Layout
        layout = QVBoxLayout()
```

```

# Label para mostrar La imagen de La cámara
self.image_label = QLabel(self)
main_layout.addWidget(self.image_label)
self.image_label.setMinimumSize(480, 800) # Tamaño mínimo

# Botón para capturar imagen
self.capture_button = QPushButton('Capturar Imagen', self)
self.capture_button.clicked.connect(self.capture_image)
self.capture_button.clicked.connect(self.perform_ocr)
controls_layout.addWidget(self.capture_button)

self.respuestas_combos = []
for i in range(10): # Suponiendo que hay 10 preguntas

    # Layout horizontal para cada pregunta y su combobox
    h_layout = QHBoxLayout()
    # Crear QLabel para la pregunta
    pregunta_label = QLabel(f'Pregunta {i + 1}:', self)
    combo = QComboBox(self)
    combo.addItem('0')
    combo.addItem('1')
    combo.addItem('2')
    combo.addItem('3') # Opciones para cada
respuesta

    # Agregar QLabel y QComboBox al layout horizontal
    h_layout.addWidget(pregunta_label)
    h_layout.addWidget(combo)
    # Agregar QLabel y QComboBox al layout
    # Agregar el layout horizontal al layout principal
    controls_layout.addLayout(h_layout)
    self.respuestas_combos.append(combo)

# Campo de texto para mostrar/ingresar el ID
self.id_field = QLineEdit(self)
id_layout = QHBoxLayout()
id_labels = QLabel('Ingreso ID')
# Agregar QLabel y QComboBox al layout horizontal
id_layout.addWidget(id_labels)
id_layout.addWidget(self.id_field)
# Botón para capturar imagen
self.id_button = QPushButton('Ingresar', self)
self.id_button.clicked.connect(self.valorid)
id_layout.addWidget(self.id_button)
controls_layout.addLayout(id_layout)

# Crear el QComboBox para seleccionar La prueba
self.prueba_selector = QComboBox(self)
self.prueba_selector.addItem('prueba1')
self.prueba_selector.addItem('prueba2')
self.prueba_selector.addItem('prueba3')
self.prueba_selector.addItem('prueba4')
self.prueba_selector.addItem('prueba5')

```

```

# Conectar el QComboBox a un método
self.prueba_selector.currentIndexChanged.connect(self.on_prueba_select
ed)

controls_layout.addWidget(self.prueba_selector)
# Crear un QLabel para mostrar cedula
self.prueba_selector.currentIndexChanged.connect(self.on_prueba_select
ed)

self.Datos_label = QLabel('Cedula ', self)
# Crear un QLabel para mostrar nombres
self.Datos2_label = QLabel('Nombres ', self)
# Crear un QLabel para mostrar curso
self.Datos3_label = QLabel('Curso ', self)
# Crear un QLabel para mostrar los resultados
self.resultado_label = QLabel('Resultados ', self)

self.guardar_button = QPushButton('Guardar', self)
controls_layout.addWidget(self.guardar_button)
self.guardar_button.clicked.connect(self.guardar_datos)

self.mostrar_cedulas_button = QPushButton('Mostrar Cédulas', self)
controls_layout.addWidget(self.mostrar_cedulas_button)
self.mostrar_cedulas_button.clicked.connect(self.mostrar_cedulas)

# Botón para realizar OCR
#self.ocr_button = QPushButton('Realizar OCR', self)
#self.ocr_button.clicked.connect(self.perform_ocr)
#controls_layout.addWidget(self.ocr_button)

# Agregar botón para abrir la segunda ventana
self.open_second_window_button = QPushButton('Registro', self)
controls_layout.addWidget(self.Datos_label)
controls_layout.addWidget(self.Datos2_label)
controls_layout.addWidget(self.Datos3_label)
controls_layout.addWidget(self.resultado_label)
controls_layout.addWidget(self.open_second_window_button)

# Conectar el botón
self.open_second_window_button.clicked.connect(self.open_second_window
)

# Container widget
container = QWidget()
container.setLayout(main_layout)
self.setCentralWidget(container)

```

```

# Iniciar La captura de video
url = 'http://192.168.1.17:8080/video'
self.cap = cv2.VideoCapture(0)
#do
self.cap.set(cv2.CAP_PROP_FRAME_WIDTH, 1380) # Ajusta La anchura
self.cap.set(cv2.CAP_PROP_FRAME_HEIGHT, 820) # Ajusta La altura
#vc
#self.cap.set(cv2.CAP_PROP_FRAME_WIDTH, 1080) # Ajusta La anchura
#self.cap.set(cv2.CAP_PROP_FRAME_HEIGHT, 520) # Ajusta La altura

# Configurar el temporizador para la actualización de la imagen de la
cámara
self.timer = QTimer(self)
self.timer.timeout.connect(self.update_image)
self.timer.start(20)

self.show()

def open_second_window(self):
self.second_window = SecondWindow()
self.second_window.show()

def update_image(self):
if self.cap is not None and self.cap.isOpened():
ret, frame = self.cap.read()
if ret:
qt_img = self.convert_cv_qt(frame)
self.image_label.setPixmap(qt_img)

def capture_image(self):
# Capturar imagen de la cámara
ret, frame = self.cap.read()
if ret:
cv2.imwrite("captura.png", frame)
self.cuadros()
image_with_squares = cv2.imread("imagen_con_cuadros.png")
qt_img = self.convert_cv_qt(image_with_squares)
self.image_label.setPixmap(qt_img)

def obtener_respuestas_usuario(self):
respuestas_usuario = []
for combo in self.respuestas_combos:
respuesta = int(combo.currentText())
respuestas_usuario.extend([1 if i == respuesta else 0 for i in
range(4)])
return respuestas_usuario

```

```

def cuadros(self):
    # Carga La imagen
    image = cv2.imread("captura.png")
    if image is None:
        print("Error: No se pudo cargar la imagen 'captura.png'.")
        return
    # Crea una lista para almacenar las imágenes recortadas
    cropped_images = []

    # Convierte La imagen a escala de grises
    gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

    # Aplica un suavizado para reducir el ruido
    blurred_image = cv2.GaussianBlur(gray_image, (5, 5), 0)

    # Aplica Canny para detectar bordes
    edges = cv2.Canny(blurred_image, 50, 150, apertureSize=3)

    # Encuentra contornos en La imagen
    contours, _ = cv2.findContours(edges, cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)

    # Filtra Los contornos que sean cuadrados
    large_square_contours = []
    for contour in contours:
        epsilon = 0.04 * cv2.arcLength(contour, True)
        approx = cv2.approxPolyDP(contour, epsilon, True)
        if len(approx) == 4 and cv2.contourArea(contour) > 1000: # Ajusta
el área mínima según tus necesidades
            large_square_contours.append(contour)

    # Dibuja Los contornos de Los cuadrados en La imagen original
    cv2.drawContours(image, large_square_contours, -1, (0, 255, 0), 3)

    # Guarda La imagen con Los cuadrados detectados
    cv2.imwrite("imagen_con_cuadrados.png", image)

    # Itera a través de Los contornos de Los cuadrados grandes
    for contour in large_square_contours:
        x, y, w, h = cv2.boundingRect(contour)

        # Recorta el cuadro del contorno actual
        cropped_image = image[y:y+h, x:x+w]

        # Guarda el cuadro recortado en La lista

```

```

        cropped_images.append(cropped_image)

# Guarda cada cuadro recortado en archivos separados
for i, cropped_image in enumerate(cropped_images):
    cv2.imwrite(f"cuadro_{i+1}.png", cropped_image)

# Llamar a respuestas al final del procesamiento
self.respuestas()

def respuestas(self):
    resp = 0
    n = 0
    # test = [1, 0, 0, 0,
    #         0, 0, 1, 0,
    #         1, 0, 0, 0,
    #         0, 0, 1, 0,
    #         0, 1, 0, 0,
    #         0, 0, 0, 1,
    #         0, 0, 1, 0,
    #         1, 0, 0, 0,
    #         0, 0, 0, 1,
    #         0, 1, 0, 0]
    # nota = 10

# Obtener Las respuestas del usuario
test = self.obtener_respuestas_usuario()

# Cargar La imagen
img = cv2.imread('cuadro_1.png')
if img is None:
    print("Error: No se pudo cargar la imagen 'cuadro_1.png'.")
    return
height, width = img.shape[:2]
respuestas = [None] * 40
error = 0

# Aumentar el brillo
value = 50 # Ajusta este valor según sea necesario
image2 = cv2.convertScaleAbs(img, alpha=1, beta=value)

# Definir el número de filas y columnas
num_filas = 10 # Número de preguntas
num_columnas = 4 # Número de opciones por pregunta
imagen_cuadricula = img.copy()

# Calcular el tamaño de cada celda
alto_celda = height // num_filas

```

```

ancho_celda = width // num_columnas
cont = 0

for i in range(num_filas):
    for j in range(num_columnas):
        # Coordenadas de la esquina superior izquierda de la celda
        x = j * ancho_celda
        y = i * alto_celda

        cv2.line(imagen_cuadrícula, (x, 0), (x, height), (0, 255, 0),
2)
        cv2.line(imagen_cuadrícula, (0, y), (width, y), (0, 255, 0),
2)

        celda = img[y:y + alto_celda, x:x + ancho_celda]
        celda_hsv = cv2.cvtColor(celda, cv2.COLOR_BGR2HSV)

        # Definir un rango para detectar violeta
        lower_violet = np.array([120, 10, 10])
        upper_violet = np.array([165, 255, 255])

        mask = cv2.inRange(celda_hsv, lower_violet, upper_violet)

        # Calcular el área del área violeta detectada y el área total
de la celda
        area_violeta = np.sum(mask) / 255.0
        area_total_celda = alto_celda * ancho_celda

        # Calcular el porcentaje de área violeta
        porcentaje_area_violeta = area_violeta / area_total_celda

        if cont >= len(respuestas):
            break

        # Definir un umbral de área mínima
        area_minima = 0.15
        if porcentaje_area_violeta >= area_minima:
            resp = 1
            #print("1 - Violeta detectado")
        else:
            resp = 0
            #print("0 - Violeta detectado")

        respuestas[cont] = resp

        if respuestas[cont] == test[cont]:
            if test[cont] == 1:

```

```

        n += 1
    else:
        error += 0.5
    cont += 1

    # Opcional: Mostrar la celda
    #cv2.imshow(f'Celda {i},{j}', celda)
    #print("Respuesta Celda:", )
    #cv2.waitKey(0)

# Imprimir las respuestas detectadas
print("Nota:", n)
#Mostrar la imagen con la cuadrícula
cv2.imshow('Imagen Original', img)
cv2.imshow('Imagen con Cuadrícula', imagen_cuadrícula)
self.resultado_label.setText(f"Nota: {n}")

def valorid(self):
    id_value = self.id_field.text()
    # Comprobar si el texto detectado tiene 10 dígitos
    # Buscar el ID en la base de datos
    datos_encontrados = self.buscar_por_cedula(id_value)
    if len(id_value) == 10:
        print("ID detectado:", id_value)
        if datos_encontrados:
            nombres = datos_encontrados[1] # 'Dany Orbes'
            curso = datos_encontrados[2] # '6to FM'
            print(f"Cédula encontrada. Nombres: {nombres}, Curso:
{curso}")

            self.Datos_label.setText(f"Cédula:{id_value}")
            self.Datos2_label.setText(f"Nombres: {nombres}")
            self.Datos3_label.setText(f"Curso:{curso}")
            QMessageBox.warning(self, "Datos encontrados", "Cédula
encontrada")

            #QMessageBox.warning(self, "Datos encontrados")
        else:
            print("Cédula no encontrada en la base de datos.")
            QMessageBox.warning(self, "Error BD142", "Cédula no encontrada
en la base de datos.")

    else:

        QMessageBox.warning(self, "Error D142", "No se reconoce la cedula,
ingrese la cedula en el casillero.")

        print("No se pudo obtener un ID válido. El ID debe tener 10
dígitos.")

```

```

        return

def perform_ocr(self):
    # Supongamos que este es tu ID registrado/predefinido
    id_registrado = "1002383402" # Reemplaza con el ID real

    pytesseract.pytesseract.tesseract_cmd = r'C:\Program Files\Tesseract-OCR\tesseract'
    # Cargar La imagen original
    imagen_original = cv2.imread('cuadro_2.png')

    # Ajustar el contraste y el brillo
    contraste = 0.9 # Ajusta el contraste según sea necesario
    brillo = 0.6 # Ajusta el brillo según sea necesario
    imagen_ajustada = cv2.convertScaleAbs(imagen_original,
alpha=contraste, beta=brillo)

    # Convertir La imagen a escala de grises
    imagen_gris = cv2.cvtColor(imagen_ajustada, cv2.COLOR_BGR2GRAY)

    # Aplicar binarización para separar el texto del fondo (ajusta los
valores según sea necesario)
    _, imagen_binaria = cv2.threshold(imagen_gris, 0, 255,
cv2.THRESH_BINARY + cv2.THRESH_OTSU)

    # Aplicar suavizado para eliminar el ruido
    imagen_suavizada = cv2.GaussianBlur(imagen_binaria, (5, 5), 0)

    # Configurar Tesseract para reconocer solo números
    custom_config = r'--oem 3 --psm 6 outputbase digits'
    texto_detectado = pytesseract.image_to_string(imagen_ajustada,
config=custom_config)

    # Eliminar todo lo que no sea dígito
    texto_detectado = ''.join(filter(str.isdigit, texto_detectado))

    # Buscar el ID en la base de datos
    datos_encontrados = self.buscar_por_cedula(texto_detectado)
    print(f"datos_encontrados=", datos_encontrados)

    # Comprobar si el texto detectado tiene 10 dígitos
    if len(texto_detectado) == 10:
        print("ID detectado:", texto_detectado)
        if datos_encontrados:
            nombres = datos_encontrados[1] # 'Dany Orbes'
            curso = datos_encontrados[2] # '6to FM'

```

```

        print(f"Cédula encontrada. Nombres: {nombres}, Curso:
{curso}")

        self.Datos_label.setText(f"Cédula:{texto_detectado}")
        self.Datos2_label.setText(f"Nombres: {nombres}")
        self.Datos3_label.setText(f"Curso:{curso}")
    else:
        print("Cédula no encontrada en la base de datos.")
        QMessageBox.warning(self, "Error BD142", "Cédula no encontrada
en la base de datos.")

    else:

        QMessageBox.warning(self, "Error D142", "No se reconoce la cedula,
ingrese la cedula en el casillero.")

        print("No se pudo obtener un ID válido. El ID debe tener 10
dígitos.")
        return

    # Imprimir el texto detectado
    print("Texto Detectado:")
    print(texto_detectado)

    # Mostrar La imagen original y La imagen preprocesada
    #cv2.imshow('Imagen Original', imagen_original)
    cv2.imshow('Imagen Ajustada', imagen_ajustada)
    cv2.imshow('Imagen Preprocesada', imagen_suavizada)
    pass

def buscar_por_cedula(self, cedula_buscada):
    conn = sqlite3.connect('registro.db')
    c = conn.cursor()
    try:
        # Crear La tabla si no existe
        c.execute('''CREATE TABLE IF NOT EXISTS registros
                    (cedula TEXT, nombres TEXT, curso TEXT, prueba1
INTEGER, prueba2 INTEGER, prueba3 INTEGER, prueba4 INTEGER, prueba5
INTEGER)''')

        # Ejecutar La consulta para buscar La cédula
        c.execute("SELECT * FROM registros WHERE cedula = ?",
(cedula_buscada,))
        resultado = c.fetchone()

        if resultado:
            print(f"Resultado: {resultado}")
            # Acceder a los valores individualmente

```

```

        cedula, nombres, curso, prueba1, prueba2, prueba3, prueba4,
prueba5 = resultado
    else:
        print("Cédula no encontrada.")

except sqlite3.Error as e:
    print(f"Error en la base de datos: {e}")
    resultado = None

finally:
    conn.close()

return resultado

def convert_cv_qt(self, cv_img):
    """Convertir una imagen de OpenCV a QImage"""
    if cv_img is None or cv_img.size == 0:
        # Manejar el caso de una imagen vacía
        # Por ejemplo, puedes retornar una QPixmap vacía o mostrar un
mensaje de error
        return QPixmap()
    # Escalar la imagen para ajustarse al tamaño de image_label
    qt_img_height = self.image_label.height()
    qt_img_width = self.image_label.width()
    resized_cv_img = cv2.resize(cv_img, (qt_img_width, qt_img_height))

    # Convertir la imagen escalada a QImage
    rgb_image = cv2.cvtColor(resized_cv_img, cv2.COLOR_BGR2RGB)
    h, w, ch = rgb_image.shape
    bytes_per_line = ch * w
    convert_to_Qt_format = QImage(rgb_image.data, w, h, bytes_per_line,
QImage.Format_RGB888)
    return QPixmap.fromImage(convert_to_Qt_format)

def on_prueba_selected(self, index):
    # Obtener la prueba seleccionada
    prueba_seleccionada = self.prueba_selector.currentText()
    print(f"Prueba seleccionada: {prueba_seleccionada}")
    # Aquí puedes agregar la lógica para manejar la selección

def guardar_datos(self):
    if self.id_field.text():
        cedula = self.id_field.text()
        pass
    else:
        cedula =self.Datos_label.text()
        # Dividir el string en los dos puntos

```

```

    partes = cedula.split(":")
    if len(partes) > 1:
        cedula = partes[1]
    else:
        cedula = ""
    pass
    # Asumiendo que id_field contiene la cédula detectada
    prueba_seleccionada = self.prueba_selector.currentText()
    nota = self.resultado_label.text() # Asumiendo que resultado_label
contiene la nota
    partes = nota .split(":")

    if len(partes) > 1:
        nota = partes[1]
    else:
        nota = ""

    print(f"Prueba seleccionada: {cedula,prueba_seleccionada,nota}")

    # Convertir la nota a un número, si es necesario
    try:
        nota = int(nota)
    except ValueError:
        QMessageBox.warning(self, "Error", "La nota no es un número
válido.")
        return

    # Guardar en la base de datos
    self.guardar_nota_en_bd(cedula, prueba_seleccionada, nota)

    def guardar_nota_en_bd(self, cedula, prueba, nota):
        # ... (tu código existente para guardar en la base de datos)
        conn = sqlite3.connect('registro.db') # Asegúrate de que
'registro.db' sea el nombre correcto de tu base de datos
        c = conn.cursor()

        # Aquí asumimos que tu tabla se llama 'registros' y tiene columnas
para cada prueba
        # y que la columna de la prueba se llama igual que el valor de
'prueba' (por ejemplo, 'prueba1', 'prueba2', etc.)
        try:
            query = f"UPDATE registros SET {prueba} = ? WHERE cedula = ?"
            c.execute(query, (nota, cedula))
            conn.commit()

            QMessageBox.warning(self, "Registro Guardado", "Prueba guardada")

```

```

    except sqlite3.Error as e:
        print(f"Error al actualizar la base de datos: {e}")
        QMessageBox.warning(self, "Error de Base de Datos", f"Error al
actualizar la base de datos: {e}")
    finally:
        conn.close()

def mantener_relacion_aspecto(self, cv_img, width, height):
    (h, w) = cv_img.shape[:2]
    r = min(width / float(w), height / float(h))
    dim = (int(w * r), int(h * r))
    resized = cv2.resize(cv_img, dim, interpolation=cv2.INTER_AREA)
    return resized

def closeEvent(self, event):
    if self.cap is not None:
        self.cap.release()
    if self.timer is not None:
        self.timer.stop()
    super().closeEvent(event)

def mostrar_cedulas(self):
    conn = sqlite3.connect('registro.db')
    c = conn.cursor()

    try:
        c.execute("SELECT cedula FROM registros")
        cedulas = c.fetchall()
    except sqlite3.Error as e:
        print(f"Error al recuperar cédulas de la base de datos: {e}")
        QMessageBox.warning(self, "Error de Base de Datos", f"Error al
recuperar cédulas: {e}")
    return
    finally:
        conn.close()

    # Mostrar Las cédulas en un QMessageBox o en otro widget
    cedulas_str = "\n".join([str(cedula[0]) for cedula in cedulas])
    QMessageBox.information(self, "Cédulas Registradas", cedulas_str)

class SecondWindow(QMainWindow):

    def __init__(self):
        super().__init__()
        self.title = 'Registro de Datos'
        self.left = 10
        self.top = 10

```

```

self.width = 800
self.height = 500
self.initUI()

def initUI(self):
    self.setWindowTitle(self.title)
    self.setGeometry(self.left, self.top, self.width, self.height)

    # Crear widgets
    self.label_cedula = QLabel('Cédula:', self)
    self.cedula = QLineEdit(self)
    self.label_nombres = QLabel('Nombres:', self)
    self.nombres = QLineEdit(self)
    self.label_curso = QLabel('Curso:', self)
    self.curso = QLineEdit(self)
    self.button = QPushButton('Agregar Registro', self)
    self.table = QTableWidgetItem(self)
    self.table = QTableWidgetItem(self)
    self.table.setColumnCount(8) # Número total de columnas
    self.table.setHorizontalHeaderLabels(["Cédula", "Nombres", "Curso",
    "Prueba 1", "Prueba 2", "Prueba 3", "Prueba 4", "Prueba 5"])

    # Layout
    layout = QVBoxLayout()
    layout.addWidget(self.label_cedula)
    layout.addWidget(self.cedula)
    layout.addWidget(self.label_nombres)
    layout.addWidget(self.nombres)
    layout.addWidget(self.label_curso)
    layout.addWidget(self.curso)
    layout.addWidget(self.button)
    layout.addWidget(self.table)

    # Agregar botones para editar y borrar
    self.edit_button = QPushButton('Editar Registro', self)
    self.delete_button = QPushButton('Borrar Registro', self)
    layout.addWidget(self.edit_button)
    layout.addWidget(self.delete_button)

    # Conectar Los nuevos botones
    self.button.clicked.connect(self.on_click)
    self.edit_button.clicked.connect(self.on_edit)
    self.delete_button.clicked.connect(self.on_delete)

    # Container widget
    container = QWidget()

```

```

container.setLayout(layout)
self.setCentralWidget(container)

# Cargar datos existentes en la tabla
self.update_table()

self.show()

@pyqtSlot()
def on_click(self):
    cedula_text = self.cedula.text()
    nombres_text = self.nombres.text()
    curso_text = self.curso.text()

    # Valores de ejemplo para prueba1, prueba2, prueba3, prueba4, prueba5
    prueba1_value = 0 # Aquí deberías obtener el valor real de tu
interfaz
    prueba2_value = 0 # Similarmente para estos valores
    prueba3_value = 0
    prueba4_value = 0
    prueba5_value = 0

    # Llamada a insert_into_db con todos los argumentos necesarios
    self.insert_into_db(cedula_text, nombres_text, curso_text,
prueba1_value, prueba2_value, prueba3_value, prueba4_value, prueba5_value)

    # Actualizar la tabla
    self.update_table()

def insert_into_db(self, cedula, nombres, curso, prueba1, prueba2,
prueba3, prueba4, prueba5):
    conn = sqlite3.connect('registro.db')
    c = conn.cursor()
    c.execute('''CREATE TABLE IF NOT EXISTS registros
                (cedula TEXT, nombres TEXT, curso TEXT, prueba1 INTEGER,
prueba2 INTEGER, prueba3 INTEGER, prueba4 INTEGER, prueba5 INTEGER)''')
    c.execute("INSERT INTO registros VALUES (?, ?, ?, ?, ?, ?, ?, ?)",
(cedula, nombres, curso, prueba1, prueba2, prueba3, prueba4, prueba5))
    conn.commit()
    conn.close()

def update_table(self):
    conn = sqlite3.connect('registro.db')
    c = conn.cursor()

```

```

c.execute("SELECT * FROM registros")
records = c.fetchall()
conn.close()

self.table.setRowCount(len(records))
self.table.setColumnCount(8)
for i, row in enumerate(records):
    for j, val in enumerate(row):
        self.table.setItem(i, j, QTableWidgetItem(str(val)))

@pyqtSlot()
def on_edit(self):
    selected_row = self.table.currentRow()
    if selected_row >= 0:
        cedula, nombres, curso = [self.table.item(selected_row, i).text()
for i in range(3)]
        self.cedula.setText(cedula)
        self.nombres.setText(nombres)
        self.curso.setText(curso)
        self.editing_row = selected_row

@pyqtSlot()
def on_delete(self):
    selected_row = self.table.currentRow()
    if selected_row >= 0:
        cedula = self.table.item(selected_row, 0).text()
        self.delete_from_db(cedula)
        self.update_table()

def delete_from_db(self, cedula):
    conn = sqlite3.connect('registro.db')
    c = conn.cursor()
    c.execute("DELETE FROM registros WHERE cedula = ?", (cedula,))
    conn.commit()
    conn.close()

if __name__ == '__main__':
    app = QApplication(sys.argv)
    ex = App()
    sys.exit(app.exec_())

```

Anexo 5: Formato de Evaluación

EVALUACIÓN DE CIENCIAS NATURALES

CEDULA:

Tiempo: 30 minutos

Indicaciones:

- Por favor colore completamente el circulo correspondiente a la respuesta, caso contrario si el circulo no este pintado, no se considerará como seleccionada
- Utiliza la hoja del cuestionario para leer las preguntas detenidamente, y analizar las ver las diferentes opciones de selección.
- Utiliza la Banco de respuestas para seleccionar su respuesta.

Selecciona la opción correcta para cada pregunta. Solo una respuesta es correcta en cada

caso.

BANCO DE RESPUESTAS

0 1 2 3

1	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	1. ¿Cuál es el principal componente del aire que respiramos?
2	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	2. ¿Qué planeta del sistema solar es conocido por sus anillos?
3	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	3. ¿Cuál es el órgano más grande del cuerpo humano?
4	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	4. ¿Qué tipo de animal es la ballena?
5	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	5. ¿Cuál es el río más largo del mundo?
6	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	6. ¿Cuál es el elemento químico más abundante en la corteza terrestre?
7	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	7. ¿Qué proceso convierte el dióxido de carbono y el agua en glucosa y oxígeno?
8	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	8. ¿Cuál de los siguientes no es un mamífero?
9	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	9. ¿Qué fuerza mantiene a los planetas en órbita alrededor del sol?
10	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	10. ¿Cuál es el elemento principal que forma las estrellas, incluido el sol?

CUESTIONARIO

1. ¿Cuál es el principal componente del aire que respiramos?

- 0 Oxígeno
- 1 Nitrógeno
- 2 Dióxido de carbono
- 3 Hidrógeno

2. ¿Qué planeta del sistema solar es conocido por sus anillos?

- 0 Marte
- 1 Júpiter
- 2 Saturno
- 3 Urano

3. ¿Cuál es el órgano más grande del cuerpo humano?

- 0 Corazón
- 1 Piel
- 2 Hígado
- 3 Cerebro

4. ¿Qué tipo de animal es la ballena?

- 0 Pez
- 1 Mamífero
- 2 Reptil
- 3 Anfibio

5. ¿Cuál es el río más largo del mundo?

- 0 Nilo
- 1 Amazonas
- 2 Misisipi

3 Yangtsé

6. **¿Cuál es el elemento químico más abundante en la corteza terrestre?**

0 Oxígeno

1 Silicio

2 Aluminio

3 Hierro

7. **¿Qué proceso convierte el dióxido de carbono y el agua en glucosa y oxígeno?**

0 Respiración

1 Fotosíntesis

2 Fermentación

3 Digestión

8. **¿Cuál de los siguientes no es un mamífero?**

0 Koala

1 Tiburón

2 Elefante

3 Murciélago

9. **¿Qué fuerza mantiene a los planetas en órbita alrededor del sol?**

0 Gravedad

1 Fricción

2 Magnetismo

3 Electricidad

10. **¿Cuál es el elemento principal que forma las estrellas, incluido el sol?**

0 Helio

1 Hidrógeno

2 Oxígeno

3 Nitrógeno