



Instituto Tecnológico de Pabellón de Arteaga



“Desarrollo de base de datos y diseño de interfaz para un sistema automatizado de captura de imágenes adquiridas a través de la experimentación microbiológica”

TITULACIÓN INTEGRAL:

TESIS

Para Obtener el Grado de:

Ingeniero en Tecnologías de la Información y las Comunicaciones

PRESENTA:

Karla Paola Martínez Medina

DIRIGIDA POR:

Dra. Pamela Romo Rodríguez

CO DIRIGIDO POR:

Ing. Oscar Lenin Espinoza Álvarez

CO DIRIGIDO POR:

Dr. Diego Alberto Mercado Ravell

Pabellón de Arteaga, Ags., noviembre 2024.



TECNOLÓGICO
NACIONAL DE MÉXICO



Instituto Tecnológico de Pabellón de Arteaga
Subdirección Académica
Ingenierías

Pabellón de Arteaga, Aguascalientes, 21/octubre/2024

Oficio No. DIN-0155/2024

Asunto: Tesis-titulación

KARLA PAOLA MARTÍNEZ MEDINA
ESTUDIANTE DE LA CARRERA DE INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN
NO. DE CONTROL 191050172

PRESENTE

Por medio de este documento me permito comunicar a Usted que habiendo recibido los votos aprobatorios de los(las) revisores(as) de su trabajo de Tesis titulado: **“DESARROLLO DE BASE DE DATOS Y DISEÑO DE INTERFAZ PARA UN SISTEMA AUTOMATIZADO DE CAPTURA DE IMÁGENES ADQUIRIDAS A TRAVÉS DE LA EXPERIMENTACIÓN MICROBIOLÓGICA”** realizado en el **“Laboratorio de Biotecnología Fúngica del ITPA,** hago de su conocimiento que puede imprimir dicho documento y continuar con los trámites para la presentación de su acto protocolario de titulación integral por Tesis, así mismo fungiendo como **Presidenta S.B. Pamela Romo Rodríguez, Secretario Oscar Lenin Espinoza Álvarez y Vocal Diego Alberto Mercado Ravell.**

Sin otro particular, aprovecho la ocasión para enviarle un cordial saludo, quedo de Usted.

ATENTAMENTE

*Excelencia en Educación Tecnológica-
“Tierra Siempre Fértil”*

EDINGUER VÁZQUEZ AYALA
JEFE DEL DEPARTAMENTO DE INGENIERÍAS



Carretera a la Estación de Rincón Km. 1 C.P. 20670 Pabellón de Arteaga, Aguascalientes
Tel. 465 958-2482 Ext. 118 e-mail: ing_parteaga@tecnm.mx tecnm.mx | pabellon.tecnm.mx



Pabellón de Arteaga, Aguascalientes, **21/octubre/2024**
Oficio No. DIN-0156/2024
Asunto: Tesis-Licenciatura

EDINGUER VÁZQUEZ AYALA
JEFE DEL DEPARTAMENTO DE INGENIERÍAS

PRESENTE

Por medio del presente doy el visto bueno a la Tesis de Licenciatura titulada **“DESARROLLO DE BASE DE DATOS Y DISEÑO DE INTERFAZ PARA UN SISTEMA AUTOMATIZADO DE CAPTURA DE IMÁGENES ADQUIRIDAS A TRAVÉS DE LA EXPERIMENTACIÓN MICROBIOLÓGICA”** realizado en el **“Laboratorio de Biotecnología Fúngica del ITPA”** de la estudiante **KARLA PAOLA MARTÍNEZ MEDINA** con número de control 191050172 de la carrera Ingeniería en Tecnologías de la Información. Dicho trabajo ya fue revisado por cada uno de los miembros del comité titular y el estudiante ya realizó los cambios sugeridos, por lo que autorizamos su impresión.

Sin otro particular, aprovecho la ocasión para enviarle un cordial saludo, quedo de Usted.

ATENTAMENTE
Excelencia en Educación Tecnológica
“Tierra Siempre Fértil”TM

S.B. PAMELA ROMO RODRÍGUEZ
MIEMBRO DE COMITÉ TUTORIAL

OSCAR LENIN ESPINOZA ÁLVAREZ
MIEMBRO DE COMITÉ TUTORIAL

DIEGO ALBERTO MERCADO RAVELL
MIEMBRO DE COMITÉ TUTORIAL

AGRADECIMIENTOS

Quisiera expresar mi más sincero agradecimiento a todas las personas que han contribuido a lo largo de mi camino académico y en la realización de este proyecto de tesis, cuyo apoyo fue invaluable para su desarrollo.

En primer lugar, extendiendo mi profundo agradecimiento a la doctora Pamela. Su generosidad al abrirme las puertas para emprender este proyecto y la confianza depositada en mi capacidad para enfrentarme a nuevos retos me permitieron explorar y crecer en áreas fuera de mi zona de confort. Su apoyo constante, así como su valiosa orientación, fueron una guía esencial para mantenerme firme en la búsqueda de mis objetivos.

Asimismo, agradezco al ingeniero Lenin, quien me acompañó no solo como docente, sino también como asesor en este proyecto. Su compromiso y dedicación en cada paso del proceso, al brindarme consejo y compartir su conocimiento, fortalecieron mi comprensión y habilidades en el desarrollo de este trabajo. Su paciencia y sus enseñanzas dejaron una huella significativa en mi trayectoria, dándome las herramientas necesarias para enfrentar los desafíos con seguridad.

A mis amigos, mi gratitud más profunda. A Valeria, mi mejor amiga, por su compañía constante y su apoyo incondicional. A Diego, quien siempre estuvo dispuesto a ayudarme a estudiar para los exámenes y me brindó su tiempo y conocimiento. A Víctor, Omar y Cristian, quienes no dudaron en brindarme su apoyo incluso en sus momentos libres. Su amistad y generosidad me recordaron que nunca debieron rendirme y me motivaron a perseverar incluso en los momentos difíciles.

Quiero también expresar mi gratitud a mis demás docentes a lo largo de la carrera, quienes, con sus consejos y enseñanzas, me impulsaron a dar siempre lo mejor de mí. Sus palabras de aliento y su dedicación fueron una fuente constante de inspiración en este camino académico.

Finalmente, a mi familia, mi reconocimiento más profundo. Ellos han sido el pilar sobre el cual se sostiene este logro, brindándome su apoyo incondicional y alentándome en cada etapa. Su presencia y confianza fueron fundamentales para sobrellevar los momentos de dificultad, y cada palabra de aliento me impulsó a seguir adelante.

A cada uno de ustedes, mi gratitud más sincera por ser parte de este camino y por su contribución invaluable en la culminación de este proyecto.

RESUMEN

El presente trabajo de tesis desarrolla un sistema de adquisición de imágenes para experimentación microbiológica, capaz de capturar, procesar y almacenar imágenes de manera eficiente y accesible. A través de una interfaz intuitiva, se facilita la interacción del usuario, garantizando un flujo de trabajo óptimo y minimizando errores durante la captura de imágenes en tiempo real. Se implementó un almacenamiento local en la Raspberry Pi, lo que asegura la integridad de los datos y el acceso directo a las imágenes sin requerir conexión externa. Además, el sistema incluye herramientas avanzadas de procesamiento de imágenes que permiten analizar visualmente la morfología y evolución de las colonias microbiológicas. Los resultados evidencian la efectividad y adaptabilidad del sistema para su uso en investigaciones experimentales en microbiología, abriendo posibilidades para futuras mejoras y aplicaciones en entornos de investigación más complejos.

INDICE

Agradecimientos	2
Resumen	3
1.Introduccion	7
2.Planteamiento del problema	11
3.Objetivos	12
3.1 Objetivo general	12
3.2 Objetivos específicos	12
4.Justificación	13
5.Hipótesis	14
6.Alcances y limitaciones	15
6.1 Alcances	15
6.2 Limitaciones	15
7.Marco teórico	16
7.1 Sistemas de adquisición de imágenes	16
7.2 Impacto de los sistemas de adquisición de imágenes	16
7.3 Evolución de adquisición de imágenes	17

7.4 Herramientas para la adquisición y procesamiento de imágenes	17
7.4.1 Visual studio code	17
7.4.2 Python	17
7.4.3 Librería Opencv	18
7.4.4 Librería Tkinter	18
7.4.5 PyInstaller	18
7.4.6 RaspBerryPi3	18
7.4.7 Protocol FTP	19
7.4.8 FileZilla	19
7.4.9 Metodología Scrum	19
8. Metodología	20
8.1 Requerimientos del Sistema	20
8.2 Metodología scrum	20
8.3 Planificación de desarrollo	22
8.4 Herramientas tecnologías	23
8.5 Algoritmo de captura	24
8.6 Diseño de interfaces	25
8.7 Almacenamiento de imágenes adquiridas	27

8.8 Propuesta de actualización para el sistema de adquisición de imágenes	28
9.Resultados y Conclusiones	30
9.1 Rendimiento de las interfaces	30
9.2 Ventana de inicio de session	30
9.3 Ventana de captura de imágenes	31
9.4 Ventana de opciones	33
9.5 Ventana de edición de imágenes	33
9.6 Ventana de detección de contornos	34
9.7 Lógica del sistema de captura	36
9.8 Flujo general del Sistema	37
9.9 Almacenamiento de imágenes	37
9.10 Archivo ejecutable	38
9.11 Impacto de propuesta en el sistema de adquisición de imágenes	38
10.Conclusiones	40
11.Referencias bibliográficas.	41

1. INTRODUCCIÓN

La Microbiología es la ciencia que estudia los microorganismos, organismos no visibles al ojo humano, así como su modo de vida y sus interacciones, tanto benéficas como patogénicas. Como ciencia, está relacionada con muchas disciplinas: la Salud Humana, Animal y Vegetal, la Producción Agrícola, la Alimentación, la Industria, la Biotecnología, el Medio Ambiente, y la Ecología. Por esta razón, la Microbiología es una ciencia interdisciplinaria que tiene un impacto fundamental en varios aspectos de nuestra vida cotidiana, que no solo se limitan a la salud humana (Madigan Martinko, Bender Buckley y Sthal, 2015). La importancia de esta ciencia radica en su capacidad para contribuir al entendimiento y solución de problemas esenciales en diversos campos, desde la producción sostenible de alimentos hasta la prevención del tratamiento de enfermedades infecciosas. Esta capacidad no solo se apoya en los avances biológicos, sino también en la sinergia con las Tecnologías de la Información y las Comunicaciones (TIC), las cuales facilitan la recolección, análisis y difusión de datos biológicos a gran escala. En esta área el empleo de las (TIC) ocupa también un papel determinante, con amplia evidencia en la literatura científica. Ello se debe a que la gestión de información, así como el análisis de datos que se producen con el empleo de herramientas tecnológicas favorecen la adecuada ejecución de los procesos que son llevados a cabo por el microbiólogo clínico. La integración de las TIC en la Microbiología no solo mejora la eficiencia en el diagnóstico y tratamiento de enfermedades, sino que también facilita la investigación y el desarrollo de nuevas soluciones para problemas microbiológicos complejos. Asimismo, mejoran la toma de decisiones, hacen más oportunos y precisos los diagnósticos emitidos, y optimizan los análisis que son realizados con los datos (Madigan Martinko, Bender Buckley y Sthal, 2015). Los sistemas de información de laboratorio, basados en el empleo de las TIC deben ser capaces de almacenar y preparar estos datos para su posterior análisis. Cómo en la mayoría de los servicios hospitalarios, un aspecto importante de la actividad del laboratorio de microbiología es la docencia y la investigación. (Ana Maria Villa C. y Fredy Gavilanes Sagnay, (2021)

Por otro lado, el Internet de las cosas (IOT) es una de las tecnologías más mencionadas en la actualidad debido a su capacidad para conectar todo tipo de dispositivos al Internet. Si a las potencialidades de IOT se le adiciona otra tecnología de alto impacto como la Visión Artificial, se abre un amplio campo de aplicaciones innovadoras, El procesamiento de imágenes y video en tiempo real permite la visualización de grandes cantidades de datos. Las principales aplicaciones que se desarrollan con IOT y Visión Artificial pueden ser implementadas en educación, medicina, edificios inteligentes, sistemas de vigilancia de personas y vehículos, entre otros. Este tipo de aplicaciones mejoran la calidad de vida de los usuarios, sin embargo, para su desarrollo se requiere una infraestructura que permita la convergencia de diferentes protocolos y dispositivos, pero de manera especial que puedan manejar las diferentes fases de la adquisición de imágenes. El término Urban IOT se entiende como la tecnología de Internet de las Cosas aplicado a las actividades cotidianas de los ciudadanos y su interacción con los servicios y recursos propios de cada ciudad. (Alvear Puertas, Rosero-Montalvo, P., Peluffo-Ordóñez, D., &

Pijal-Rojas, J. (2017). Con el transcurso del tiempo el concepto de IOT se ha ido cambiando y transformando. En el 2005, la Unión Internacional de Telecomunicaciones (UIT) publicó su primer informe sobre el IOT (ITU,2005). Internet de las Cosas (IOT) ha evolucionado con la finalidad de unir a un gran número de elementos con diferentes capacidades de conexión y la computación a una gran y única red, siendo estos capaces de enviar información sobre la Internet, detectar el estado de un entorno, procesar los datos y enviar los resultados. (Alvear Puertas et al., Rosero-Montalvo, P., Peluffo-Ordóñez, D., & Pijal-Rojas, J. (2017)

La visión artificial o como la llaman otros autores visión por computador, se la define como la ciencia de programar un computador para procesar imágenes o videos e incluso entenderlos (Culjak y Abram, 2012). La transformación de datos desde un fotograma o video cámara en lo que puede ser una decisión o una nueva representación (Bradski y Kaebler, 2008). Por ejemplo, los sistemas de detección facial han tomado fuerza en los últimos años debido a que son ampliamente utilizados en sistemas biométricos, el objetivo de estos sistemas es identificar a un individuo en una imagen digital de forma automática. La detección facial automática tuvo sus inicios en los años 60, el primer sistema semiautomático necesitaba que el administrador localice rasgos como ojos, cejas, nariz y boca en las imágenes para calcular distancias a determinados puntos de referencia que después serían analizados con datos almacenados en el sistema. Los próximos sistemas de detección usaron distintos marcadores y técnicas para mejorar el proceso de detección. En la actualidad existen sofisticadas herramientas que son aplicadas en distintas áreas sociales como entretenimiento, tarjetas inteligentes (pasaportes, licencias de conducir), seguridad de la información, aplicación de la ley y vigilancia, entre otros.

La visión artificial, que se basa en el uso de cámaras y algoritmos especializados, permite el análisis de imágenes y videos con una precisión notable. Este enfoque tecnológico ha sido clave en el desarrollo de sistemas que pueden interpretar visualmente el entorno, detectando cambios y patrones con una eficiencia que antes era impensable. Estos sistemas no solo analizan imágenes en tiempo real, sino que también pueden procesar y reaccionar ante información visual de manera autónoma, marcando un avance significativo en la interacción entre las máquinas y la obtención de información de su entorno (Ansari, A. N., Sedky, M., Sedky, M., & Tyagi, A., 2015).

El sistema se basará en el uso de un script, el cual es un conjunto de instrucciones escritas en un lenguaje de programación que se ejecutan de manera secuencial, el script fue realizado en el lenguaje de programación de Python. Python es un lenguaje de programación moderno creado por Guido van Rossum a inicios de los años noventa. Se llevo en conjunto con el desarrollo del script la configuración de un sistema externo llamado FileZilla, FileZilla es un cliente FTP, (FTP son las siglas de “File Transfer Protocol” o en español “Protocolo de transferencia de archivos”, que es un protocolo para la transferencia de archivos entre sistemas conectados a una red), multiplataforma de código abierto y software libre, licenciado bajo la Licencia MIT. Soporta los protocolos FTP, SFTP y FTP sobre SSL/TLS (FTPS), la configuración implementada fue de forma específica para que

fungiera como un almacenamiento para la recopilación de información gráfica en un servidor FTP externo. (Tizne Mena, G. (2015). Todo parte desde la base del procesamiento de imágenes que es indispensable, de tal manera que la interpretación de una imagen o la captura de videos puedan ser procesados para mejor desempeño de un algoritmo. Aquí es donde entra en juego el uso de bases de datos en los sistemas de adquisición de imágenes. Una base de datos es una colección de archivos relacionados que permite el manejo de la información de una compañía. Cada archivo puede ser visto como una colección de registros, y cada registro está compuesto por una colección de campos. Los campos permiten llevar información de algún atributo de una entidad del mundo real. Un archivo de una base de datos también puede ser pensado como una tabla con renglones y columnas, cada renglón correspondiendo a un registro y cada columna a un campo. Los sistemas manejadores de bases de datos conocidos también como bases de datos electrónicas se usan ampliamente para: organizar y manipular grandes volúmenes de datos de las empresas. Un sistema manejador de base de datos no es más que un sistema computarizado para llevar registros (Marco Antonio Cruz Chávez, 2011).

En primer lugar, proporcionan un método estructurado para almacenar información, lo que facilita la recuperación y el análisis de datos. Además, permiten la integración de datos de diferentes fuentes, asegurando que la información esté centralizada y sea coherente. Las bases de datos también soportan múltiples usuarios y aplicaciones simultáneamente, garantizando la integridad y consistencia de los datos a través de mecanismos de control de acceso y gestión de transacciones (Gomez, 2012).

Una de las tecnologías que puede aprovechar las bases de datos para optimizar sus funciones es el FTP. Desde un equipo cliente, se puede conectar a un servidor para descargar archivos desde él o para enviarle archivos, independientemente del sistema operativo utilizado en cada equipo. Este protocolo es esencial para la subida y bajada de archivos, facilitando la gestión y transferencia de grandes volúmenes de datos (Cardenal Gardoki, 2010). Un servidor FTP también ofrece varias características útiles para la gestión de datos en sistemas de adquisición de imágenes. Por ejemplo, permite la configuración de permisos y accesos diferenciados, asegurando que solo usuarios autorizados puedan subir o descargar archivos específicos. Además, puede manejar múltiples conexiones simultáneas, lo que es crucial en entornos donde se requiere la transferencia de datos en tiempo real o casi en tiempo real. El uso de FTP en combinación con bases de datos proporciona una infraestructura robusta para almacenar, organizar y acceder a imágenes y otros tipos de datos, mejorando la eficiencia y fiabilidad de los sistemas de procesamiento de imágenes. Asimismo, FTP puede integrarse con otros protocolos y servicios, como el cifrado de datos mediante FTPS (FTP Secure) o SFTP (SSH File Transfer Protocol), aumentando así la seguridad durante la transferencia de información sensible.

La integración de FTP con bases de datos permite la automatización de muchas tareas administrativas, como la actualización de registros en tiempo real a medida que se reciben nuevas imágenes o datos. Esto es particularmente útil en aplicaciones que requieren monitoreo constante, como sistemas de vigilancia,

telemedicina, y análisis científicos. Por ejemplo, en un sistema de monitoreo ambiental, las imágenes capturadas por cámaras remotas pueden ser transferidas automáticamente a un servidor central, donde se almacenan en una base de datos y se analizan para detectar cambios en el entorno. Además, la capacidad de FTP para gestionar grandes volúmenes de datos es crucial en proyectos de *Big data*, donde el almacenamiento y la transferencia eficiente de datos son esenciales para el análisis a gran escala. Con el soporte adecuado, un servidor FTP puede manejar terabytes de datos sin comprometer la velocidad ni la integridad de los archivos transferidos. Esta capacidad de gestión de grandes datos es indispensable para proyectos que involucran aprendizaje automático e inteligencia artificial, donde se necesitan grandes conjuntos de datos para entrenar modelos precisos y efectivos.

En resumen, FTP es una herramienta versátil y poderosa que, cuando se combina con bases de datos, puede mejorar significativamente la gestión, transferencia y análisis de datos en una variedad de aplicaciones tecnológicas avanzadas (MIT, Massachusetts Institute of Technology).

En la utilización de dispositivos externos surge la Raspberry Pi, una placa computadora (SBC) de bajo costo que se considera un ordenador de tamaño reducido, similar al tamaño de una tarjeta de crédito. Desarrollada en el Reino Unido por la Fundación Raspberry Pi (Universidad de Cambridge) en 2011, su objetivo principal es estimular la enseñanza de la informática en las escuelas, aunque su comercialización no comenzó hasta el año 2012. El concepto detrás de la Raspberry Pi es el de un ordenador desnudo de todos los accesorios que se pueden eliminar sin que afecte al funcionamiento básico. Está formada por una placa que soporta varios componentes necesarios en un ordenador común y es capaz de comportarse como tal. La Raspberry Pi ha sido definida como una maravilla en miniatura, que guarda en su interior un importante poder de cómputo en un tamaño muy reducido (Axel Giovanni Arellano García, 2021).

Para que la Raspberry Pi sea plenamente funcional, utiliza Raspbian, un conjunto de programas básicos y utilitarios (programas de soporte) que permiten que la Raspberry sea útil. Raspbian no es solo un sistema operativo convencional; es una distribución optimizada específicamente para la Raspberry Pi. Raspbian proporciona un entorno de escritorio completo, con todas las herramientas que un usuario pueda necesitar para la programación, la administración del sistema y el uso general. Esto incluye lenguajes de programación como Python, Scratch y C++, así como una gama de utilidades y herramientas de desarrollo que son esenciales para cualquier proyecto de informática o electrónica. La distribución de Raspbian se mantiene en constante desarrollo con el objetivo principal de mejorar el desempeño y la estabilidad de los paquetes originales de la distribución de Linux/Debian, esto significa que los usuarios de Raspberry Pi tienen acceso a las últimas actualizaciones y mejoras, asegurando que su sistema operativo sea siempre eficiente y seguro (Axel Giovanni Arellano García, 2021, p.23).

2. PLANTEAMIENTO DEL PROBLEMA

En el ámbito de la experimentación microbiológica, la captura y gestión de imágenes presentaba varios desafíos. Anteriormente los métodos manuales que se empleaban resultaban lentos y propensos a errores humanos, lo que consumía una cantidad considerable de tiempo del personal del laboratorio. Esta problemática afectaba directamente la eficiencia y precisión del trabajo en investigaciones microbiológicas. La falta de automatización en el proceso de captura de imágenes impactaba negativamente en el laboratorio de biotecnología fúngica, dentro del Instituto Tecnológico de Pabellón de Arteaga, particularmente en los investigadores. Los resultados científicos y la comprensión de los microorganismos se veían comprometidos debido a la ineficiencia y la posible inexactitud de los datos recopilados manualmente. Para abordar esta problemática, resultaba crucial implementar un sistema automatizado de captura de imágenes que incluyera una interfaz de usuario; este sistema permitiría almacenar las imágenes en paquetes dentro de carpetas etiquetadas con la fecha de captura de forma automática, mejorando significativamente la adquisición y gestión de las capturas durante la experimentación microbiológica. Además, se había planteado desarrollar un servicio FTP como base de datos para el almacenamiento de los paquetes de fotos capturadas. La automatización del proceso no solo aceleraría la recolección de datos y el análisis de imágenes, sino que también reduciría la posibilidad de errores humanos, incrementando así la calidad y fiabilidad de los resultados científicos.

3. OBJETIVOS

3.1 Objetivo general

Desarrollar un sistema automatizado de captura de imágenes, con una interfaz sencilla y una base de datos para el almacenamiento de las imágenes obtenidas a través de la experimentación microbiológica.

3.2 Objetivos Específicos

- Desarrollar una interfaz con el sistema de captura de imágenes:

Desarrollar una interfaz comprensible y fácil de usar para el usuario en conjunto con las herramientas que proporciona el sistema para capturar imágenes en tiempo real.

- Desarrollar Base de datos:

Se implementará el desarrollo de un servicio FTP para almacenar y gestionar los archivos de imágenes obtenidas de la experimentación microbiológica. Este servicio funcionará como base de datos permitiendo la organización de paquetes con las capturas en tiempo real de la experimentación microbiológica.

- Propuesta de procesamiento de imágenes

Diseñar una propuesta para el procesamiento de imágenes a través de herramientas que proporcionen una calidad de mejora en la visualización de las imágenes capturadas en tiempo real de la experimentación microbiológica.

4. JUSTIFICACIÓN

En el ámbito de la microbiología experimental, la captura y el análisis de imágenes de muestras microbiológicas eran esenciales tanto para la investigación como para el diagnóstico. La realización de estas experimentaciones generaba una gran cantidad de datos visuales (imágenes) que debían ser analizados y almacenados de manera eficiente. Tradicionalmente, este proceso se llevaba a cabo manualmente, lo que lo hacía propenso a errores, tales como el control de calidad y el ángulo de captura de las imágenes.

La necesidad de automatización en este campo surgía con el objetivo de aumentar la precisión y la eficiencia, así como de mejorar la obtención y el análisis de datos a partir de las imágenes capturadas. La implementación de técnicas automatizadas no solo minimiza los errores humanos, sino que también optimiza el tiempo y los recursos necesarios para llevar a cabo estas tareas, permitiendo un manejo de datos más fiable. Por lo tanto, la automatización en la captura y análisis de imágenes microbiológicas resulta fundamental para avanzar en la precisión y la efectividad de la investigación y el diagnóstico microbiológico.

Para abordar esta problemática, fue crucial implementar un sistema automatizado de captura de imágenes que incluyera una interfaz de usuario. Este sistema permite almacenar las imágenes en paquetes dentro de carpetas etiquetadas con la fecha de captura de forma automática, mejorando significativamente la adquisición y gestión de las capturas durante la experimentación microbiológica. Además, se desarrolló un servicio FTP a manera de base de datos para el almacenamiento de los paquetes de fotos capturadas. La automatización del proceso no solo acelera la recolección de datos y el análisis de imágenes, sino que también reduce la posibilidad de errores humanos, incrementando así la calidad y fiabilidad de los resultados científicos.

5. HIPOTESIS

El desarrollo de una base de datos y una interfaz para un sistema automatizado de captura de imágenes en experimentación microbiológica optimiza el almacenamiento y la gestión de datos, mejora la precisión y repetibilidad de los experimentos, y facilita la colaboración entre investigadores. La automatización del sistema y el uso de un servicio FTP incrementan la eficiencia en el acceso y organización de la información, asegurando la integridad y disponibilidad de los datos para un análisis más concreto.

6. ALCANCES Y LIMITACIONES

6.1 Alcances

- Se desarrolló una interfaz intuitiva y fácil de usar que permitió a los usuarios interactuar con el sistema de manera efectiva.
- Se desarrolló el sistema con un algoritmo para la captura de fotografías en tiempo real.
- Se implementaron mecanismos que simplificaron la gestión de las imágenes capturadas, como el etiquetado automático con la fecha de toma de cada foto.
- Se integró la interfaz de usuario con el sistema de capturas en tiempo real, logrando una interacción fluida y eficiente entre la interfaz y el sistema, asegurando un funcionamiento correcto y cohesivo.
- Se desarrollaron funcionalidades que permitieron la visualización de los paquetes de fotografías capturadas y funcionalidades para tener una mejor visualización de las imágenes capturadas como ajuste de brillo, saturación, encontrar contornos.
- Se aseguró que el sistema permitiera un acceso rápido y eficiente a los datos almacenados.

6.2 Limitaciones

- Compatibilidad limitada con servidores FTP: El sistema no es completamente compatible con programas de gestión de archivos como FileZilla, lo cual dificulta la sincronización automática y el almacenamiento remoto de datos.
- Dependencia de recursos adicionales para procesamiento avanzado: La eficiencia en el procesamiento de imágenes, especialmente para segmentación, está limitada por la disponibilidad de recursos y requiere tiempos de procesamiento más largos, lo que impide el análisis en tiempo real.
- Requiere conexiones estables y dispositivos compatibles: El sistema depende de un ambiente controlado con conexiones estables y hardware compatible, restringiendo su adaptabilidad en entornos con configuraciones variables.

7. MARCO TEORICO

Aunque algunas de las referencias utilizadas en este marco teórico eran de años anteriores, su contenido seguía siendo altamente relevante y crucial para la comprensión de los conceptos y tecnologías fundamentales en la adquisición de imágenes. Estos estudios pioneros habían establecido las bases teóricas y metodológicas que sustentaban los avances actuales en este campo. Por ejemplo, las definiciones y principios básicos del lenguaje de programación Python, desarrollados por (Van Rossum y De Boer en 1991), seguían siendo la piedra angular de su uso y evolución en la actualidad. Del mismo modo, las metodologías desarrolladas para el enfoque automático en sistemas de microscopía por (Prabhat et al., Ram, S., Ward, ES y Ober, RJ 2004) continuaban siendo relevantes para la mejora continua de estas tecnologías. Por lo tanto, era esencial considerar estas fuentes históricas para entender completamente los desarrollos contemporáneos y sus aplicaciones.

7.1 Sistemas de adquisición de imágenes

La adquisición de imágenes constituía una tarea fundamental en la transformación del mundo real al mundo digital, siendo crucial para garantizar la calidad necesaria en el posterior análisis del objeto de estudio (Kozubek, M. 2005). Un aspecto crítico en esta fase era la obtención de un plano focal óptimo, especialmente en el contexto de microscopios diseñados para capturar imágenes de organismos que podían moverse rápidamente en un solo plano focal. Aunque existían dispositivos que permitían el enfoque automático mediante hardware, estos sistemas solían ser relativamente lentos, requiriendo aproximadamente 10 milisegundos para ajustar el plano focal (Prabhat et al., Ram, S., Ward, ES y Ober, RJ 2004). Esta velocidad limitada podía resultar en la pérdida de eventos significativos en las imágenes adquiridas.

7.2 Impacto de los sistemas de adquisición de imágenes

Durante las últimas décadas, se habían desarrollado nuevos sensores de adquisición de imágenes, lo que incrementó la disponibilidad de datos con características nuevas: imágenes de alta resolución espacial, temporal y espectral, así como modelos digitales de elevación más precisos y globales. A la par, esto ocasionó una disminución en los costos (Mas & Morales, 2011).

7.3 Evolución de adquisición de imágenes

El Internet de las cosas (IOT, por sus siglas en inglés) fue una de las tecnologías más mencionadas en la actualidad debido a su capacidad prevista para conectar todo tipo de dispositivos a Internet. Si a las potencialidades del IOT se le añadía otra tecnología de alto impacto, como la Visión Artificial, se obtenía un amplio campo de aplicaciones innovadoras, donde el procesamiento de imágenes y video en tiempo real permitía la visualización de grandes cantidades de datos. Este tipo de aplicaciones mejoraba la calidad de vida de los usuarios; sin embargo, para su desarrollo se requería una infraestructura que permitiera la convergencia de diferentes protocolos y dispositivos, especialmente aquellos que pudieran manejar las diferentes fases de la adquisición de imágenes (Alvear-Puertas, Rosero-Montalvo, Peluffo-Ordóñez y Pijal-Rojas, 2017)

7.4 Herramientas para la adquisición y procesamiento de imágenes

Para el desarrollo de este proyecto, se emplearon diversas herramientas tecnológicas que facilitaron la adquisición y procesamiento de imágenes, esenciales para alcanzar los objetivos planteados.

7.4.1 Visual studio code

Visual Studio Code es un IDE dedicado al desarrollo de cualquier software. La herramienta es ligera y se asemeja a la funcionalidad de Visual Studio, aunque presenta utilidades integrales de revisión y compilación empaquetadas en una interfaz de aspecto moderno. Una de las características notables de Visual Studio Code es su capacidad de control de Git (Git es un sistema de control de versiones distribuido) que ofrece el programa. Facilita las pruebas, la construcción, el empaquetado e incluso la implementación de varios tipos de software. (Code,2019)

7.4.2 Python

Lenguaje de desarrollo interpretado y versátil, Originalmente pensado como una alternativa para el tipo de trabajos que tradicionalmente se resuelven mediante una mezcla de scripts de Shell manualmente. Python ha evolucionado hasta convertirse en un lenguaje de creación de prototipos. Uno de los puntos fuertes es la capacidad que tiene el usuario de escribir código y ejecutarlo inmediatamente; el rendimiento se ve mejorado por la sintaxis clara y concisa, su alto nivel con los tipos de datos y su falta de declaraciones que se compensa mediante la verificación de datos en tiempo de ejecución. Python pretende ser una herramienta para programadores de sistemas profesionales (Van Rossum, G.y De Boer, J. 1991).

7.4.3 Librería Opencv

Opencv (Open Source Computer Vision Library) es una librería orientada al procesamiento y análisis de imágenes. Desarrollada inicialmente por Intel en 1999, siendo una de las librerías más empleadas en el desarrollo de aplicaciones de visión artificial. Sus ámbitos de aplicación son variados donde se destacan:

- Reconocimiento facial
- Identificación de objetos o personas
- Inspección y vigilancia
- Análisis de imágenes

(Tomás Domínguez Mínguez,2021)

7.4.4 Librería Tkinter

La GUI (por sus siglas en inglés: interfaz gráfica de usuario) en Python se implementa mediante diversas librerías, siendo Tkinter una de las más utilizadas. Tkinter es una librería orientada a objetos que permite crear interfaces gráficas fácilmente utilizando el lenguaje de programación Python. (Axel Giovanni Arellano García,2021)

7.4.5 PyInstaller

PyInstaller es un programa que convierte los programas de Python en ejecutables autónomos para Windows, Linux, Mac OS X, etc. Crea ejecutables pequeños y utiliza el soporte del sistema operativo para cargar las bibliotecas dinámicas, garantizando así la compatibilidad con el sistema operativo. (Adrián López Fernández, (2017)

7.4.6 RaspBerry Pi3

La Rasperry Pi 3 es una microcomputadora. Esta placa fue diseñada para proporcionar un entorno accesible para aprender informática y programación, ofreciendo capacidades de un ordenador básico a un precio extremadamente asequible. Lanzada en 2011 por la Fundación Rasperry Pi en el Reino Unido, la Rasperry Pi 3 es un dispositivo compacto que permite realizar diversas tareas computacionales, desde proyectos educativos hasta aplicaciones avanzadas de desarrollo (Cristina Secada Carral,2019)

7.4.7 Protocolo FTP

El protocolo FTP (Protocolo de transferencia de archivos) como su nombre lo indica, es un protocolo para transferir archivos. La implementación del FTP se remonta a 1971 cuando se desarrolló un sistema de transferencia de archivos entre equipos del Instituto Tecnológico de Massachusetts.

El protocolo FTP define la manera en que los datos deben ser transferidos a través de una red por sus siglas en inglés TCP/IP (Transmission Control Protocol/Internet Protocol). (MIT, Massachusetts Institute of Technology).

7.4.8 FileZilla

"FileZilla es un programa gratuito que actúa como cliente FTP (Protocolo de Transferencia de Archivos), lo que significa que permite a los usuarios transferir archivos entre su computadora y un servidor remoto, como si fuera un intermediario para subir y descargar archivos. Su principal ventaja es que facilita este proceso de manera sencilla y rápida, incluso para personas sin experiencia técnica. Aunque existen otros programas con más funciones avanzadas, FileZilla es conocido por ser uno de los más confiables y fáciles de usar, lo que lo convierte en una opción ideal. (Ali Teodardo Martínez Cervantes, (2016)

7.4.9 Metodología Scrum

Scrum es una metodología ágil que se utiliza para gestionar y desarrollar proyectos de manera eficiente, especialmente en entornos complejos o donde los requisitos pueden cambiar con frecuencia. En lugar de seguir un plan rígido, Scrum divide el trabajo en ciclos cortos llamados 'sprints'.

Esta metodología es fácil de entender para cualquier persona, ya que se basa en el trabajo en equipo, la transparencia y la flexibilidad, lo que permite que los proyectos avancen de manera constante y con mayor eficiencia. (Rodríguez, C., & Vicente, R. D. (2015)

8. METODOLOGÍA

"En esta sección se describe la metodología empleada para llevar a cabo el proyecto, detallando cada paso del proceso y los materiales utilizados."

8.1 Requerimientos del sistema

Durante esta etapa se identificaron los objetivos principales del proyecto y se desarrolló un plan detallado para alcanzarlos. Se elaboró un cronograma que incluía reuniones para asegurar el progreso y ajustar el plan según fuera necesario. Fue crucial detallar los requerimientos al inicio del proyecto para asegurar el cumplimiento de los objetivos y funcionalidades requeridas del sistema de adquisición de imágenes.

Se comenzó con el análisis y la definición de los requerimientos del proyecto, los cuales proporcionaron una guía clara sobre lo que se necesitaba lograr. Los requerimientos detallados permitieron una planificación más precisa y efectiva, priorizando de manera más clara las tareas, el tiempo estimado y los recursos necesarios de manera más exacta. Se llevaron a cabo reuniones para recopilar información fundamental sobre los requerimientos del proyecto. Estos requerimientos se dividieron en cuatro categorías:

- **Requerimientos de negocio:** Este requerimiento especificaba los objetivos y metas que el proyecto debía alcanzar, como la optimización de la captura de imágenes microbiológicas para su análisis detallado y automatizado.
- **Requerimientos de usuario:** Este requerimiento describía las necesidades y expectativas de los usuarios del sistema de adquisición de imágenes. Se llevaron a cabo reuniones detalladas con los usuarios finales para entender sus necesidades específicas en términos de captura y almacenamiento de imágenes.
- **Requerimientos funcionales y Requerimientos no Funcionales:** Estos requerimientos definían los criterios de desempeño y la calidad que el sistema debía cumplir, así como las funcionalidades específicas que debía tener para satisfacer las expectativas de los usuarios. Se definieron de manera detallada las funcionalidades solicitadas por los usuarios, así como las expectativas de visualización y almacenamiento del sistema de adquisición de imágenes.

8.2 Metodología Scrum

Al tener los requerimientos del sistema ya detallados, se procedió con la selección de una metodología para establecer un plan de desarrollo del sistema dentro del tiempo estimado. Para el desarrollo del sistema se optó por la metodología ágil Scrum debido a su flexibilidad y enfoque iterativo, lo que facilitó la adaptación a cambios y

la mejora continua a lo largo del desarrollo. La metodología Scrum se basó en un ciclo de desarrollo iterativo e incremental, permitiendo adaptarse a los cambios y la mejora continua. Se estructuró en torno a eventos, artefactos y roles que aseguraron una colaboración eficaz y la entrega regular de valor.

Para la planificación del desarrollo del sistema, se utilizaron herramientas de la metodología Scrum, como los ciclos de sprint, para fijar la duración del trabajo. Este trabajo se representó en forma de cronograma con actividades específicas asignadas a cada día. Cada día se tenía una actividad específica en la que se trabajaba de manera continua durante toda la semana, como lo podemos ver representado en la figura 1. Estos ciclos, llamados sprints, tuvieron una duración de dos semanas.

	A	B	C	D	E	F	G	H
	ACTIVIDAD		ENERO	FEBRERO			ACTIVIDAD	MARZO
1							Desarrollo de login para el proyecto	
2	Reunion de introduccion al proyecto						Prueba de funcionamiento de Raspberrry	
3	Adquisicion de objetivos y requerimientos del proyecto						pruebas de conexión de raspberrry y camara	
4	Realizacion de un cronograma de actividades						Instalacion de python en raspberrry	
5	Analisis e instalacion de herramientas que seran utilizadas						Diseño de interfaces del sistema	
6	Reunion de introduccion para el conocimiento de opencv						Desarrollo de primeras interfaces	
7	Recabar informacion sobre la nueva herramienta de opencv						Reunion de aprobacion de interfaces	
8	Recabar informacion sobre los materiales externos para el proyecto						Desarrollo de funcionalidades de primeras interfaces	
9	Investigacion sobre el almacenamiento de la informacion						Desarrollo de practicas de tutoriales opencv	
10	Desarrollo de diagrama de flujo para el funcionamiento del proyecto						Desarrollo de interfaces faltantes para el sistema	
11	Desarrollo de un diagrama de flujo para el almacenamiento						Desarrollo de servidor ftp	
12	Lectura e introduccion a los tutoriales de opencv						Desarrollo de funcionales de captura con camara	
13	Investigacion de entornos graficos para el lenguaje de python						Desarrollo de creacion de paquetes	
14	Investigacion sobre la funcionalidad de filezilla						Desarrollo de funcionalidad de combinacion de imágenes	
15	Investigacion sobre la realizacion de conexiones de filezilla y python						Division de interfaces del proyecto por ventanas	
16	Realizacion de practica de primeros tutoriales de opencv						Desarrollo de una conexión con el sistema para visualizacion	
17	Realizacion de pruebas de conexión de python con camara web						de los paquetes de fotografias	
18	Desarrollo de primeras interfaces con python						Reunion para evaluacion de avances	
19	Investigacion sobre el desarrollo de un servidor ftp							

Figura 1. Cronograma de actividades del proyecto. El diagrama de Gantt mostrado fue elaborado en excel y abarca los primeros meses de desarrollo del proyecto de tesis. En él se detallan las principales actividades planificadas durante esta fase inicial, Cada actividad está representada por una barra que indica su duración estimada, permitiendo una visualización clara del progreso y el tiempo asignado para cada etapa del proyecto. Este cronograma sirve como una guía para asegurar el cumplimiento de los plazos y una organización eficiente del trabajo.

Al final de cada sprint, el último día de la segunda semana, se realizó una reunión de revisión para evaluar los avances y recibir retroalimentación. Esta estructura permitió una planificación detallada y una evaluación constante del progreso, asegurando que el desarrollo del sistema se mantuviera enfocado y alineado con los objetivos establecidos, como lo podemos observar en la Figura 2.

I	J	K	L	M	N
		ACTIVIDAD	ABRIL	Marzo	
		Desarrollo de conexión con todas las interfaces			
		Pruebas de funcionalidad entre interfaces			
		Puebas de intrerfaz y conexión con camara web			
		Desarrollo de funcionalidad para la descarga de imagen			
		Puebas de ruta de descarga de paquetes y cantidades			
		Investigacion sobre que es pyinstaller			
		Investigar sobre mas funcionalidades de Tkinter			
		Asesoría de funcionalidades con python			
		Pruebas de funcionalidad entre interfaces			
		Realización de practicas de opencv			
		Pruebas de conexión con el servidor ftp			
		Obtener el prototipo de raspberry con camara web			
		formatear raspebrry			
		instalar otro sitema operativo			
		instalar paquetes necesarios para el sistema operativo			
		instalar driver para el reconocimiento de camara web			
		instalar lenguaje de python			
		creacion de un entorno virtual con python			
		instalacion de dependencias necesarias de python			
		Instalacion de opencv			
		Montaje del sistema de captura			
		Pruebas exhaustivas del sistema y la creacion de paquetes			
		Reunion para la presentacion del sistema			

Figura 2. Cronograma de la fase final del proyecto. En esta parte del diagrama de Gantt, se muestran las actividades correspondientes a la última fase de desarrollo del sistema, donde se incluyen las pruebas finales aplicadas al sistema y la presentación del proyecto. Las barras del cronograma reflejan el tiempo dedicado a la corrección de errores, optimización del sistema y validación de su funcionalidad

8.3 Planificación de desarrollo

Dentro del trabajo de los ciclos de sprint, se llevó a cabo la realización de un modelado de procesos para lograr una mejor comprensión de cómo debían fluir los datos del sistema al momento de ejecutar la adquisición de las imágenes. Este modelado permitió visualizar y documentar el flujo de información y las interacciones entre diferentes componentes del sistema. Se realizó un diagrama de caso de uso el sistema con las funcionalidades que debe cumplir al estar en interacción con el usuario, como lo podemos ver en la Figura 3.

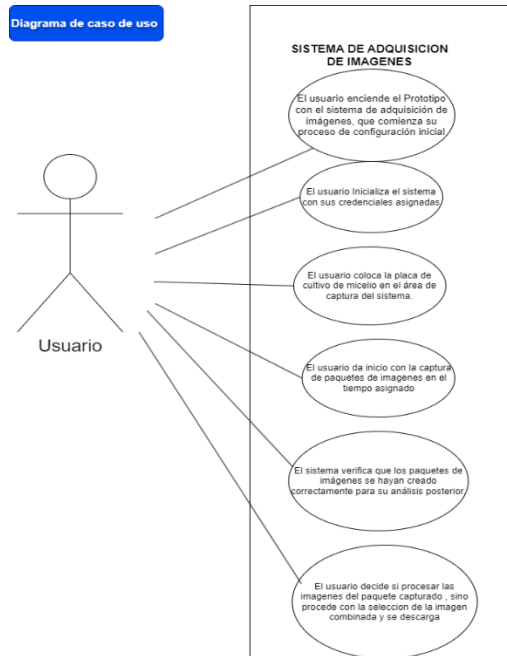


Figura 3. Diagrama de casos de uso del sistema. Se representa cómo el usuario interactúa con el sistema de adquisición de fotos y el proceso que este realiza. El diagrama de casos de uso muestra las acciones clave que el usuario puede llevar a cabo, como la captura de imágenes y cómo el sistema procesa estas solicitudes, desde la toma de las fotografías hasta su almacenamiento y posterior visualización o descarga.

8.4 Herramientas tecnológicas

Se utilizó una laptop con el sistema operativo Windows 10 como apoyo para todo el desarrollo del sistema de adquisición de imágenes. Este sistema operativo contaba con la instalación previa de Visual Studio Code, que fue la herramienta principal para la codificación.

Para el desarrollo, se eligió Python como el lenguaje de programación, dado que era el más adecuado para los objetivos planteados en los requerimientos del sistema. Python se descargó desde el sitio web oficial y se procedió a su instalación en el sistema operativo Windows 10.

Con el lenguaje de programación establecido y el entorno de codificación configurado, se procedió con la instalación de la librería Opencv, la cual proporciona múltiples funcionalidades relacionadas con el procesamiento de imágenes. Al igual que con el sistema operativo, la librería se descargó del sitio web oficial y se instaló en el sistema operativo Windows 10.

Con todas las herramientas implementadas y listas para su uso, se procedió a realizar varias pruebas de verificación del funcionamiento. Utilizando la librería Opencv, se creó un archivo. Py para verificar la funcionalidad de un dispositivo de cámara en la laptop. Esta verificación resultó crucial para el desarrollo del sistema, ya que aseguró que el equipo pudiera capturar imágenes de manera efectiva

8.5 Algoritmo de Captura

Basándose en el diagrama de flujo del sistema, se diseñaron una serie de algoritmos funcionales con distintos niveles de complejidad lógica, como lo podemos ver en la Figura 4. El proceso inició con una prueba sencilla para verificar la correcta conexión con la cámara y la captura básica de imágenes.

A medida que el sistema avanzó, se desarrollaron versiones más avanzadas de los algoritmos, que permitieron no solo la captura de un número específico de imágenes en intervalos definidos, sino también la implementación de funciones como la edición de imágenes, la creación de paquetes de fotos y la detección de contornos. Estas características mejoraron la funcionalidad del sistema, ofreciendo al usuario más control sobre el procesamiento y manipulación de las imágenes capturadas, siguiendo el flujo detallado en el diagrama.

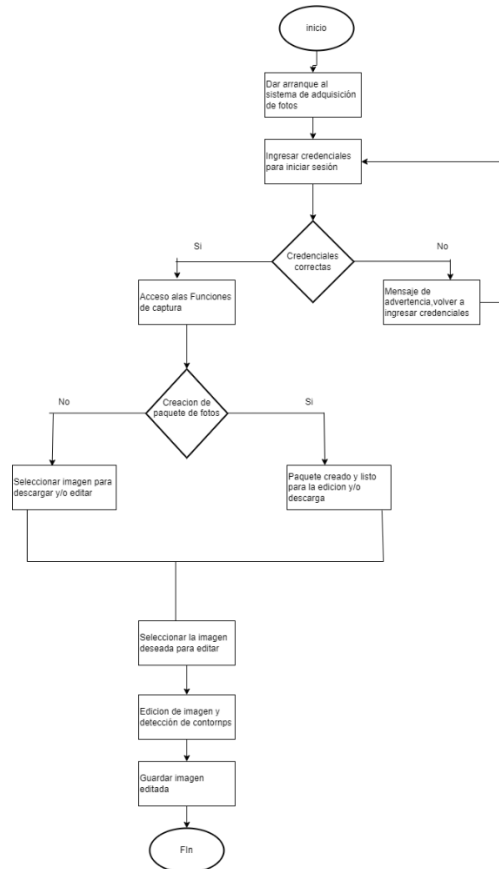


Figura 4. Diagrama de flujo del funcionamiento del sistema. Esta imagen muestra el diagrama de flujo que describe el funcionamiento completo del sistema. Se ilustran los diferentes pasos y decisiones que el sistema sigue desde el inicio hasta la finalización del proceso. El flujo incluye las fases de captura de imágenes, procesamiento, almacenamiento, y las posibles acciones del usuario, como visualizar o descargar las fotos.

8.6 Diseño de interfaces

Al asegurar que la lógica algorítmica funcionara correctamente, se proporcionó una base sólida sobre la cual construir las interfaces de usuario. Esto minimizó el riesgo de tener que rediseñar o ajustar las interfaces debido a problemas de funcionalidad básica no resueltos previamente.

Una vez que se estableció una base lógica sólida, esta se complementó con la codificación para el desarrollo de interfaces visuales para las funciones algorítmicas ya diseñadas. Para el lenguaje de programación Python, la opción más ideal para el desarrollo de diseño gráfico fue la utilización de las herramientas que proporcionaba la librería Tkinter, la cual era totalmente compatible con el lenguaje.

Tkinter fue una biblioteca estándar de Python que se utilizó para crear interfaces gráficas de usuario (GUI). Esta librería fue altamente valorada por su simplicidad y compatibilidad con el lenguaje Python, lo que la convirtió en una opción ideal para el desarrollo de aplicaciones visuales. Tkinter proporciona una amplia gama de widgets y herramientas que permiten diseñar y gestionar ventanas, botones, etiquetas, campos de entrada, lienzos y pestañas, entre otros elementos gráficos.

Se realizó una investigación para conocer las funcionalidades básicas necesarias para crear elementos gráficos como contenedores, botones, pestañas, ventanas, entre otras características. Durante esta investigación, se exploraron funciones de Tkinter. Estas funciones permitieron experimentar y crear distintos prototipos de entornos gráficos visuales para el usuario, asegurando así el desarrollo de interfaces de usuario eficientes y funcionales.

Se diseñaron prototipos en los cuales se probaron las funciones en distintos aspectos, como la creación de una sola ventana con múltiples funcionalidades integradas. También se realizaron pruebas con ventanas que contenían distintas pestañas, así como pruebas de funcionalidades específicas dentro de estas pestañas. Estas pruebas permitieron evaluar y ajustar el comportamiento de las interfaces gráficas para asegurar su eficiencia y usabilidad, como se pudo observar en la Figura 5.

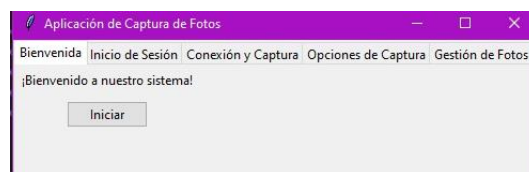


Figura 5. Primer prototipo de la interfaz del sistema utilizando Tkinter. En la imagen se muestra una ventana con varias pestañas, cada una de ellas con una funcionalidad específica integrada. Este diseño inicial permitió evaluar y probar las diferentes funciones de la interfaz gráfica, sirviendo como base para el desarrollo posterior del sistema.

Para validar estos prototipos, se llevaron a cabo sesiones de prueba con usuarios finales. Se recopilaron sus comentarios y se realizaron ajustes en función de sus necesidades y sugerencias, lo que garantizó que las interfaces fueran intuitivas y fáciles de usar. Además, se trabajó en la integración de estos prototipos gráficos con la lógica algorítmica previamente desarrollada, asegurando que todas las funciones del sistema de adquisición de imágenes se ejecutaran correctamente desde la interfaz de usuario.

Para mantener la sobriedad y enfoque del sistema, las interfaces gráficas se diseñaron utilizando una paleta de colores en escala de grises. Esta decisión estética aseguró que la influencia de los colores no afectara la renderización del sistema, ni la influencia para la utilización y que el entorno visual se mantuviera profesional y objetivo.

Inicialmente, las pestañas del prototipo se diseñaron para cumplir con los requerimientos establecidos, asegurando acceso directo a todas las funcionalidades de manera sencilla y objetiva, sin añadir características innecesarias que afectaran los objetivos del proyecto. Sin embargo, conforme se fueron puliendo los detalles de la interfaz, se optó por reemplazar este enfoque con un sistema de ventanas emergentes, donde cada ventana permite acceder de forma más clara e independiente a las funcionalidades clave, como la apertura y cierre de la cámara, la captura de imágenes y la gestión de tiempos de captura, utilizando Tkinter y Opencv.

Al finalizar el proceso de desarrollo, se crearon varios prototipos como parte de los resultados. Estos prototipos no solo evidenciaron la funcionalidad del sistema, sino que también sirvieron como referencia para futuras mejoras y expansiones del proyecto, como podemos ver en la Figura 6, estos prototipos ayudarían a la definición detallada de las interfaces necesarias para el sistema en su versión final.

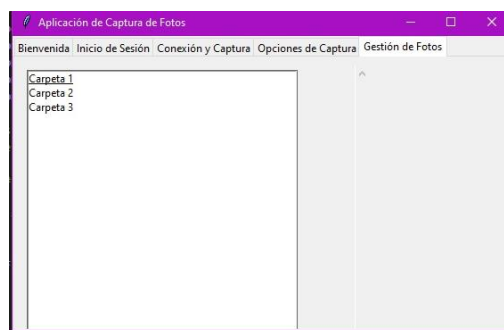


Figura 6. Segundo prototipo de la interfaz del sistema utilizando Tkinter. Con estas adiciones, el desarrollo del sistema no solo se enfocó en la funcionalidad interna, sino también en proporcionar una experiencia de usuario optimizada y coherente con los objetivos del proyecto.

8.7 Almacenamiento de Imágenes adquiridas

Una vez establecidos y validados los prototipos de las interfaces gráficas y las funcionalidades algorítmicas del sistema de adquisición de imágenes, se abordó la siguiente fase del proyecto: la implementación de un servidor FTP. La integración de un servidor FTP es importante para el almacenamiento y transferencia eficiente de las imágenes capturadas.

Haciendo una investigación profunda y llevando a cabo un análisis de la experiencia previa en la utilización de bases de datos, se optó por la utilización de un servicio FTP con ayuda del sistema de FileZilla. Este enfoque se consideró más adecuado para el proyecto debido a su simplicidad y efectividad en la transferencia y almacenamiento de archivos. La decisión se tomó teniendo en cuenta la necesidad de gestionar grandes volúmenes de datos de imágenes de manera eficiente y segura.

En un principio, se seleccionó un software de servidor FTP adecuado, como FileZilla, reconocido por su estabilidad y seguridad. Sin embargo, al analizar las configuraciones necesarias para mantener la funcionalidad del servicio FTP dentro del sistema operativo Raspbian, se encontraron varios obstáculos.

Durante el proceso de implementación, se llevaron a cabo varias etapas de configuración y pruebas. Inicialmente, se optó por el software de servidor FTP debido a su reputación de estabilidad y seguridad. Sin embargo, surgieron dificultades al intentar configurar este servidor en el sistema operativo Raspbian. Uno de los problemas principales fue la falta de paquetes esenciales en los repositorios de Raspbian, lo cual complicó la instalación y generó errores durante el proceso de configuración. A pesar de los intentos de encontrar repositorios alternativos o versiones compatibles de los paquetes necesarios, la ausencia de elementos críticos representó un desafío significativo.

Además, se enfrentaron problemas de incompatibilidad de versiones entre los paquetes disponibles en los repositorios y la versión específica de Raspbian utilizada. Estos conflictos resultaron en errores durante la instalación y configuración del servidor FTP. Los intentos de ajustar o revertir versiones de paquetes no siempre resolvieron los problemas, ya que las versiones compatibles a menudo generaban nuevas incompatibilidades con otros componentes del sistema, como lo podemos observar en la Figura 7, donde el sistema reconoce y envía un mensaje de error al usuario sobre la conexión al servidor ftp.

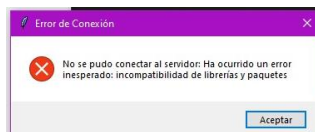


Figura 7. En esta imagen se muestra un mensaje de error del sistema de adquisición de imágenes que informa al usuario sobre la imposibilidad de establecer conexión con el servidor FTP.

Estas dificultades con las dependencias añadieron complejidad al proceso. La necesidad de versiones específicas de paquetes y la gestión de sus dependencias se volvieron difíciles de manejar. Aunque se consideró la instalación manual desde el código fuente como una solución alternativa, esta opción también presentó desafíos debido a la complejidad de resolver todas las dependencias de manera manual.

Las dificultades no se limitaron solo a la instalación de paquetes, sino que también incluyeron la configuración del servidor FTP en sí. Por ejemplo, configurar adecuadamente los permisos de acceso para garantizar la seguridad y la gestión de archivos solo por usuarios autorizados fue un proceso complejo, exacerbado por las inconsistencias en las versiones de software. Asimismo, la configuración de medidas de seguridad como FTPS requirió ajustes detallados y pruebas para asegurar la protección de los datos transferidos.

A estas dificultades se suma una nueva limitación: el tiempo disponible para la realización del proyecto. Esta restricción temporal representa un impedimento significativo, ya que reduce la capacidad para abordar los errores iniciales del servidor FTP de manera adecuada y oportuna. La presión del tiempo dificulta la posibilidad de realizar pruebas exhaustivas y de ajustar la configuración necesaria para garantizar el correcto funcionamiento del sistema.

8.8 Propuesta de actualización para el sistema de adquisición de imágenes

Tras asegurar la operatividad del sistema de adquisición de imágenes con funcionalidad estable, surgió la visión de expandir sus capacidades con funcionalidades avanzadas de procesamiento de imágenes. Se tenía previsto integrar herramientas de OpenCV para llevar a cabo operaciones sencillas como ajustes de brillo o contraste para las imágenes capturadas para posteriormente encontrar contornos para lograr dar una perspectiva más adecuada al momento de analizar el crecimiento de las setas en la experimentación microbiológica.

Para preparar el terreno hacia esta expansión, se inició una investigación exhaustiva sobre las capacidades y aplicaciones de OpenCV. Esto incluyó la exploración de tutoriales detallados para comprender a fondo las técnicas y algoritmos necesarios para el procesamiento de imágenes en tiempo real. Estos esfuerzos estaban orientados a adquirir el conocimiento práctico necesario para implementar de manera efectiva estas funcionalidades en el sistema existente.

Esta biblioteca ofrecía algoritmos robustos para tareas como detección de objetos, reconocimiento facial, corrección de color y mucho más, lo cual era fundamental para apoyar al microbiólogo experto en el monitoreo del crecimiento de los hongos y utilidad del sistema de adquisición de imágenes.

Al integrar estas capacidades, se esperaba no solo mejorar la calidad y precisión del análisis de imágenes microbiológicas, sino también facilitar la automatización de procesos críticos en el contexto del análisis microbiológico. Un aspecto fundamental de esta integración es la accesibilidad de las herramientas para los usuarios, incluso aquellos sin un conocimiento técnico profundo. Se han implementado funciones sencillas que permiten a los usuarios realizar ajustes básicos en las imágenes capturadas por el sistema, como el brillo, la saturación y el contraste. Estas opciones se presentan de manera intuitiva, lo que permite a los investigadores mejorar la visualización de las muestras sin necesidad de complicados procedimientos técnicos.

Por ejemplo, un usuario puede fácilmente ajustar el brillo de una imagen para mejorar la iluminación de las muestras microbiológicas, o modificar la saturación para resaltar características específicas de los organismos presentes. Además, se ofrece la opción de guardar estas imágenes modificadas, lo que permite a los investigadores conservar un registro visual de sus hallazgos y realizar comparaciones a lo largo del tiempo.

Estas herramientas no solo simplifican la manipulación de imágenes, sino que también son cruciales para el análisis del crecimiento de setas en las experimentaciones microbiológicas. Los usuarios pueden aplicar funciones de detección de contornos a las imágenes procesadas, facilitando la identificación y el seguimiento del desarrollo de los cultivos.

Esto permite un análisis más eficiente y preciso de los resultados experimentales, brindando a los investigadores una valiosa ayuda en la interpretación de sus datos. La integración de estas capacidades no solo optimiza la calidad del análisis, sino que también transforma la experiencia del usuario, haciendo que el análisis microbiológico sea más accesible y efectivo.

9. RESULTADOS Y CONCLUSIONES

Los resultados presentados demuestran la efectividad del sistema de adquisición de imágenes en tiempo real para experimentación microbiológica. Se logró capturar imágenes y crear paquetes de 10 fotos, generando una imagen final que combina estas 10 en una sola. Además, se implementaron funcionalidades para el procesamiento de imágenes, permitiendo ajustes de brillo y contraste, así como la detección de contornos en los cultivos. Aunque se decidió mantener el sistema a nivel local sin la integración de un servidor FTP.

9.1 Rendimiento de las interfaces del sistema de adquisición de imágenes

Las interfaces desarrolladas no solo facilitan la captura de imágenes microbiológicas, sino que también optimizan de manera notable la interacción del usuario con el sistema. Esto se tradujo en una mejora significativa de la usabilidad y la eficiencia del sistema de adquisición de imágenes. A través de estas interfaces, los usuarios pudieron interactuar de manera intuitiva con las diversas funciones del sistema, lo que les permitió ajustar parámetros específicos en función de las necesidades experimentales y realizar un seguimiento detallado del proceso de captura en tiempo real de imágenes, como los ajustes de brillo, contraste y la detección de contornos, se integraron de forma efectiva, mejorando la capacidad de análisis de las imágenes obtenidas. En conjunto, estas mejoras contribuyen a una experiencia de usuario más fluida y eficiente, facilitando un control más preciso sobre cada etapa del proceso sin necesidad de intervenciones complejas o conocimientos técnicos avanzados.

Este enfoque centrado en el usuario aseguró que tanto investigadores como técnicos pudieran utilizar el sistema con facilidad y precisión, minimizando la curva de aprendizaje y maximizando la productividad durante las operaciones de laboratorio. Permitió una gestión más eficiente del flujo de trabajo en entornos experimentales.

Este capítulo profundizó en el rendimiento específico de las interfaces, detallando las funcionalidades de cada una y cómo el usuario interactuó con ellas. Se analizó el uso de las distintas funcionalidades y la forma en que estas interfaces se integraron en la operación general del sistema, evaluando su impacto en el desempeño del mismo, además, se destacaron las áreas de mejora que fueron identificadas durante las pruebas y evaluaciones realizadas.

9.2 Ventana de inicio de sesión

La ventana de inicio de sesión se implementó en respuesta a un requisito clave discutido durante las reuniones de avance del proyecto. Este requisito establecía la necesidad de restringir el acceso al sistema de adquisición de imágenes a solo usuarios autorizados, asegurando que únicamente ellos pudieran utilizar las funcionalidades del sistema. Para cumplir con esta exigencia, se diseñó un formulario de inicio de sesión sencillo y seguro, que requería la autenticación mediante nombre de usuario y contraseña. Este enfoque no solo cumplió con los

estándares de seguridad definidos, sino que también, garantizó que solo los usuarios verificados podrían acceder y operar herramientas críticas como la captura, visualización y configuración de imágenes microbiológicas. De este modo, se aseguró tanto la protección de los datos sensibles como el cumplimiento del requisito de seguridad acordado, en la Figura 8 queda presentado la implementación en el sistema.

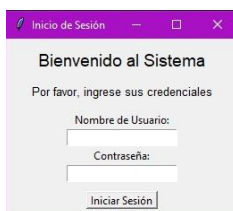


Figura 8. Ventana de inicio de sesión del sistema de adquisición de imágenes. Ventana de inicio de sesión diseñada para restringir el acceso al sistema únicamente a usuarios autorizados, cumpliendo con el requisito de seguridad. La interfaz permite la autenticación de tres usuarios específicos, garantizando un uso controlado y seguro de las funcionalidades del sistema.

Durante el proceso de inicio de sesión, se implementaron mensajes claros para indicar el estado de la autenticación. Cuando los usuarios ingresaban credenciales válidas, el sistema les mostraba un mensaje de 'Acceso correcto', confirmando que las credenciales eran válidas y permitiendo el acceso a las funcionalidades del sistema, por otro lado, si las credenciales ingresadas no coincidían con los datos autorizados, se generaba un mensaje de 'Acceso rechazado', informando al usuario que los datos proporcionados eran incorrectos y negando el acceso. los usuarios con credenciales válidas podrían interactuar con el sistema, protegiendo la seguridad de la información y las herramientas críticas.

9.3 Ventana de captura de imágenes

El objetivo de tener el paquete de 10 fotografías fue obtener una imagen final que reuniría todas las fotos capturadas durante el proceso. Esta foto final consistía en una única imagen que mostraba las demás, lo que permitía observar de manera clara y efectiva el crecimiento de la experimentación microbiológica.

A través de esta imagen, los investigadores pudieron visualizar los diferentes estados de desarrollo de las setas a lo largo del tiempo. Además, esta representación gráfica resultó ser una herramienta valiosa para llevar a cabo un análisis concreto de la experimentación microbiológica, facilitando la identificación de patrones y tendencias en el crecimiento y desarrollo de los microorganismos involucrados. De este modo, la imagen no solo sirvió como un registro visual, sino también como un recurso fundamental para la interpretación de los resultados.

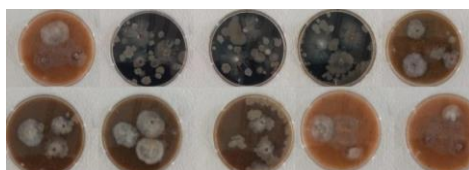


Figura 9. Imagen combinada. muestra el resultado del paquete de captura de fotos, donde se han integrado 10 imágenes individuales en una sola. Esta visualización permite un análisis más fácil y eficiente de los cultivos microbiológicos, destacando los detalles esenciales capturados durante el proceso de experimentación. Cada sección de la imagen representa un momento específico en el tiempo, proporcionando un registro visual claro y ordenado de las condiciones experimentales.

La interfaz de usuario del sistema de adquisición de imágenes fue diseñada cuidadosamente para proporcionar una experiencia intuitiva, facilitando una operación eficiente y minimizando los errores por parte del usuario. La interfaz se dividió en dos secciones principales, representadas por dos botones claves que desempeñan funciones críticas en el flujo operativo del sistema.

La primera sección corresponde al botón de "Capturar Paquetes de Fotos". Cuando el usuario hace clic en este botón, el sistema inicia la captura de imágenes en la experimentación microbiológica. El proceso de captura sigue un intervalo de tiempo predeterminado, configurado previamente por el usuario en el código, que controla el tiempo de espera entre cada foto. Las imágenes se capturan a lo largo de un período para registrar distintas etapas de crecimiento del hongo, permitiendo observar cambios progresivos en su desarrollo.

Cada imagen capturada se guarda con un nombre basado en la fecha y hora en la que se toma, lo que garantiza un registro adecuado y evita la creación de duplicados. Al finalizar la captura, el sistema genera un paquete de 10 fotos, que representan varias etapas en el tiempo, y muestra un mensaje de "Paquete realizado correctamente". Dentro de este paquete, se incluye una "foto objetivo", que combina las 10 fotos anteriores en una sola imagen, facilitando su visualización y análisis.

Este diseño optimizado no solo mejoró la eficiencia operativa del sistema, sino que también aseguró una gestión clara y ordenada de las fotos y los paquetes capturados, como lo representa la Figura 10.

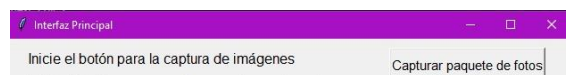


Figura 10. Primer botón de la interfaz de captura de fotos. Esta imagen muestra el botón principal utilizado para iniciar la captura de imágenes en la interfaz del sistema. Al presionar este botón, el usuario activa la cámara para capturar las fotografías que documentan el proceso experimental.

La segunda sección de la interfaz está conformada por el botón de "Opciones de Captura". Este botón, como su nombre lo indica, se divide en varias subsecciones y permite al usuario acceder a funciones adicionales del sistema. Estas funciones son esenciales para la edición de las imágenes capturadas, así como para su gestión y almacenamiento. A través de este botón, el usuario puede descargar imágenes que forman parte del paquete, editarlas según sea necesario y luego guardarlas, ya sea dentro del mismo paquete o en una ruta diferente que el usuario especifique.

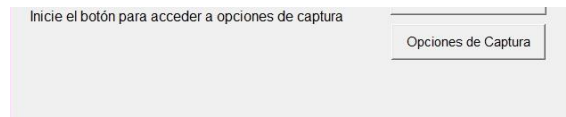


Figura 11. Segundo botón de opciones de captura. El botón de 'Opciones de Captura' permite al usuario acceder a funcionalidades adicionales del sistema, como la descarga, edición y guardado de imágenes capturadas.

9.4 Ventana de opciones

Esta ventana presenta dos botones principales, brindando al usuario la libertad de elegir cuál utilizar según las funcionalidades requeridas. El primer botón, denominado "Descarga de Fotografía en Particular", permite al usuario seleccionar una imagen específica dentro del paquete de fotos capturadas y proceder a descargarla. Esta funcionalidad es especialmente útil, ya que permite al usuario trabajar con una copia de la imagen sin comprometer la original en el paquete. De este modo, el usuario puede utilizar la imagen descargada para otras tareas, como análisis externos o reportes, manteniendo intacta la organización y registro de las imágenes originales dentro del sistema.

El segundo botón, denominado "Procesar Detalles de la Imagen", ofrece la opción de seleccionar cualquier imagen del paquete para realizar ajustes y ediciones. Esta funcionalidad proporciona al usuario la capacidad de modificar parámetros de la imagen, como el brillo, contraste o la nitidez, permitiendo una personalización más detallada según los requerimientos específicos de la experimentación. Las ediciones realizadas no afectan la imagen original, ofreciendo la posibilidad de guardar la versión editada en el paquete o en una ubicación diferente especificada por el usuario, garantizando flexibilidad y control en el manejo de los archivos.

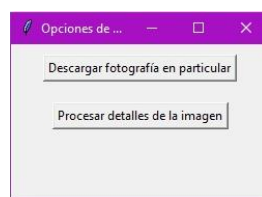


Figura 12. Ventana con opciones de descarga y procesamiento de imágenes. Ventana de la interfaz que presenta dos botones principales: el primero permite la descarga de una fotografía específica del paquete de imágenes capturadas, mientras que el segundo ofrece la opción de procesar y editar los detalles de cualquier imagen seleccionada

9.5 Ventana de edición de imágenes

La ventana de edición de imágenes está diseñada para proporcionar una experiencia interactiva y eficiente, permitiendo al usuario realizar ajustes visuales de manera precisa. La interfaz se divide en dos secciones principales: en el lado izquierdo, se muestra la imagen seleccionada que el usuario desea editar, mientras que en el lado derecho se encuentran varias barras deslizantes que ofrecen diferentes opciones de ajuste. Estas barras

permiten al usuario modificar en tiempo real el brillo, la saturación y la nitidez de la imagen, con los cambios reflejados de inmediato en la vista previa de la imagen situada en el lado izquierdo de la ventana.

El ajuste de brillo permite aumentar o reducir la luminosidad general de la imagen, facilitando la visualización de detalles en áreas oscuras o excesivamente iluminadas. La barra de saturación ofrece la posibilidad de intensificar o reducir los colores de la imagen, haciéndolos más vibrantes o, en su defecto, más tenues según las necesidades del usuario. Por su parte, el ajuste de nitidez permite aclarar los bordes y detalles de los elementos dentro de la imagen, ayudando a resaltar las características importantes para el análisis.

Además, la interfaz cuenta con dos botones adicionales para complementar el proceso de edición. El primero, denominado "Guardar Imagen", permite al usuario almacenar la imagen editada una vez que está satisfecho con los ajustes realizados. Esta opción asegura que la versión modificada quede guardada sin afectar la imagen original, ofreciendo flexibilidad para trabajar con diferentes versiones de la misma imagen.

El segundo botón, denominado "Seleccionar Nueva Imagen", permite al usuario elegir una nueva imagen del paquete de fotos para iniciar un nuevo proceso de edición.

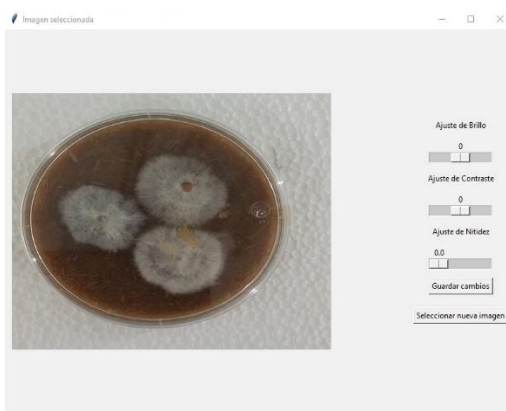


Figura 13. Ventana de edición de imágenes con opciones de guardado y selección. Ventana de edición de imágenes que muestra los controles principales para ajustar brillo, saturación y nitidez mediante barras deslizantes. A la izquierda se observa la imagen seleccionada, mientras que a la derecha se encuentran las herramientas de ajuste. Los botones adicionales permiten al usuario guardar la imagen editada o seleccionar una nueva imagen para comenzar un nuevo proceso de edición.

9.6 Ventana de detección de contornos

En esta nueva ventana, la imagen seleccionada aparece en el lado izquierdo de la pantalla, permitiendo al usuario visualizar claramente el objeto de estudio. Al lado derecho, se presenta una única función, representada por un botón que, al ser presionado, activa el proceso de detección de contornos. Este botón, etiquetado como "Detectar Contornos", ofrece una forma sencilla de iniciar el análisis sin distracciones adicionales.

Cuando el usuario hace clic en este botón, el sistema procesa la imagen y proporciona, dentro de la interfaz, la misma imagen, pero con un cambio significativo: se añaden líneas de un color notable para que el usuario pueda identificar fácilmente las áreas de crecimiento. Esta representación visual es esencial para la evaluación de los cultivos microbiológicos, ya que facilita al usuario observar la morfología y características de los organismos presentes.

Además, la interfaz cuenta con un botón adicional que permite al usuario guardar la nueva imagen procesada, ya sea dentro del paquete de fotos o en otra ruta que el usuario necesite. Este diseño funcional garantiza que los investigadores puedan realizar un seguimiento eficiente de los cultivos microbiológicos y contar con datos visuales valiosos para la interpretación de los resultados experimentales.

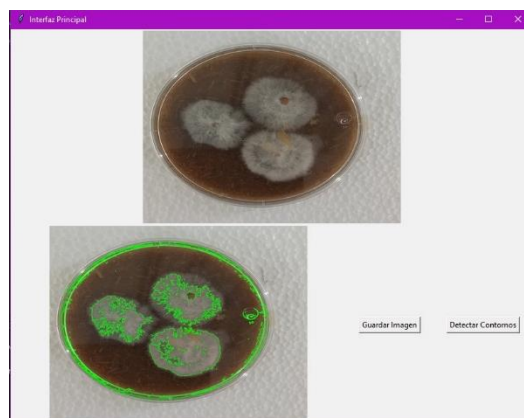


Figura 14. Ventana de detección de contornos. muestra la imagen seleccionada del lado izquierdo y el botón 'Detectar Contornos' del lado derecho. Al procesar la imagen, se añaden líneas de un color notable para resaltar las áreas de crecimiento en la placa de experimentación microbiológica. La interfaz también incluye un botón adicional para guardar la imagen procesada, permitiendo al usuario conservar el resultado en el paquete de fotos o en una ubicación personalizada.

En conclusión, la integración de estas nuevas funcionalidades en el sistema de adquisición y procesamiento de imágenes mejoró notablemente la capacidad de los usuarios para analizar y comprender los datos microbiológicos. La combinación de edición de imágenes y detección de contornos permitió obtener una representación más rica y detallada de los cultivos, asegurando que los investigadores tuvieran acceso a herramientas eficaces para respaldar sus estudios y conclusiones en el ámbito de la microbiología.

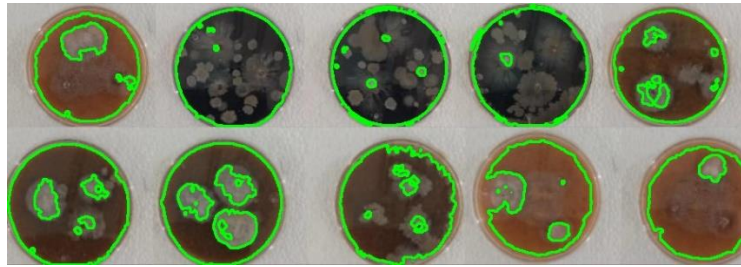


Figura 15. Imagen final de experimentación microbiológica. Imagen final que muestra el paquete de fotos capturadas en una sola representación visual. Esta imagen destaca las diferentes etapas de crecimiento de los cultivos microbiológicos, resaltando los contornos de manera clara y efectiva. El procesamiento de la imagen asegura una visualización estable y editada, permitiendo a los investigadores analizar y documentar de forma precisa los resultados de la experimentación.

9.7 Lógica del sistema de captura de imágenes en tiempo real

En esta sección se describe la lógica del sistema de captura de imágenes en tiempo real, detallando cada paso necesario para el funcionamiento óptimo del proceso de adquisición de imágenes microbiológicas. Se desglosaron los componentes clave del código, explicando cómo cada elemento contribuyó al desempeño general del sistema y garantizó una captura eficiente y organizada de las imágenes.

El proceso comenzó con la inicialización de la cámara, en la cual se establecieron las conexiones necesarias entre el dispositivo y el sistema de control. Esto permitió asegurar que la cámara estuviera lista para comenzar a capturar imágenes de manera inmediata cuando el usuario lo solicitara. Durante esta fase, se configuraron parámetros críticos, como la resolución de la imagen y el formato de salida, adaptándolos a los requisitos específicos.

la lógica del sistema incluyó la configuración de los parámetros de captura, como el intervalo de tiempo entre cada fotografía. Esta configuración fue fundamental para asegurar que el sistema tomara imágenes a intervalos regulares, permitiendo el seguimiento preciso del desarrollo de los cultivos microbiológicos. El usuario podía definir estos intervalos desde el código, adaptando la frecuencia de las capturas según las necesidades de cada etapa de la experimentación.

Finalmente, el sistema manejó el almacenamiento organizado de las imágenes capturadas. Cada imagen se guardó con un nombre único, basado en la fecha y la hora de la captura, lo que permitió un registro cronológico de los resultados y evitó la duplicación de archivos. Este enfoque no solo facilitó la gestión de los datos, sino que también aseguró que el usuario pudiera acceder rápidamente a las imágenes correspondientes a cada etapa de crecimiento.

En conjunto, la lógica del sistema de captura de imágenes en tiempo real garantizó una captura eficiente y ordenada, proporcionando un flujo de trabajo adaptable y confiable para los investigadores en el seguimiento de

los cultivos microbiológicos. Esta estructura lógica permitió integrar funciones avanzadas, como la detección de contornos y la edición de imágenes, optimizando el uso del sistema para obtener datos precisos y visualizaciones claras de cada fase de la experimentación.

9.8 Flujo general del sistema

El sistema de captura de imágenes en tiempo real se basó en un algoritmo diseñado para permitir la adquisición continua de imágenes desde una cámara conectada, garantizando un monitoreo constante y preciso de los cultivos microbiológicos. El flujo general del sistema abarcó una serie de pasos clave que aseguraron la correcta operación y manejo de las imágenes capturadas. Estos pasos fueron los siguientes:

- **Inicialización de la cámara:** El proceso comenzó con la configuración e inicio de la cámara, estableciendo una conexión estable entre el dispositivo de captura y el sistema. Durante esta fase, se configuraron parámetros esenciales, como la resolución de la imagen y el formato de salida, asegurando que la cámara esté lista para iniciar la adquisición de imágenes en las condiciones óptimas para cada experimento.
- **Apertura de Imágenes:** Una vez que la cámara estuvo lista, el sistema procedió a la adquisición de imágenes a intervalos regulares, tal como fue programado por el usuario. Estos intervalos permitieron capturar una serie de fotografías que documentaron de manera precisa cada etapa del desarrollo microbiológico. Este enfoque fue fundamental para obtener un registro detallado del crecimiento y evolución de los cultivos a lo largo del tiempo.
- **Almacenamiento de imágenes:** Cada imagen capturada fue guardada en el sistema de archivos con un nombre único, basado en la fecha y hora exactas de la captura. Esto permitió mantener un registro ordenado y cronológico de las imágenes, evitando duplicados y facilitando el acceso rápido a las fotografías de cada fase del proceso. El almacenamiento organizado fue clave para asegurar la integridad de los datos y su disponibilidad para análisis posteriores.

9.9 Almacenamiento de imágenes

El proceso de almacenamiento de imágenes comenzó con la configuración e inicialización de la cámara, seguido de la captura de las imágenes de acuerdo con los parámetros establecidos. Cada imagen fue almacenada de forma ordenada en una carpeta designada, lo que permitió una fácil organización y acceso a los archivos, además, el sistema permitió combinar todas las imágenes de un conjunto en una única imagen más grande, que también se almacenó en la misma carpeta para mantener un registro consolidado de las capturas.

Para facilitar el manejo de los datos, el sistema identificaba automáticamente la carpeta más reciente, donde se almacenaban las imágenes, permitiendo a los usuarios acceder de manera rápida a las fotografías más recientes

sin necesidad de buscar manualmente entre varios directorios. Esto optimizó el flujo de trabajo y garantizó un acceso eficiente a los resultados experimentales.

```
def guardar_fotos(self, fotos, carpeta):
    os.makedirs(carpeta, exist_ok=True)
    [cv2.imwrite(f'{carpeta}/foto_{i}.jpg', foto) for i, foto in enumerate(fotos)]

def combinar_fotos(self, carpeta_base=None):
    if not carpeta_base:
        carpeta_base = self.obtener_ultima_carpeta()
    if not carpeta_base:
        print("No se encontraron carpetas de fotos.")
        return
    imagenes = [cv2.imread(os.path.join(carpeta_base, f'foto_{i}.jpg')) for i in range(5)]
    combined_image = np.vstack([np.hstack(imagenes[:5]), np.hstack(imagenes[5:])])
    output_path = os.path.join(carpeta_base, 'foto_combinada.jpg')
    cv2.imwrite(output_path, combined_image)
    print(f'Fotos combinadas y guardadas como {output_path}')
    return carpeta_base
```

Figura 16. Proceso de almacenamiento de imágenes en el sistema de captura. La imagen muestra la estructura de almacenamiento de las fotografías capturadas, organizadas en carpetas específicas y combinadas en una única imagen consolidada.

9.10 Archivo ejecutable

Una vez finalizado el desarrollo del sistema de adquisición de imágenes, se procedió a la creación de un archivo ejecutable para facilitar su acceso y uso por parte de los usuarios. Esta decisión se tomó con el objetivo de simplificar el proceso de inicio del sistema, asegurando que la interacción con el software fuera intuitiva y eficiente.

La creación del archivo ejecutable permitió que el sistema pudiera ser utilizado de manera más sencilla, ya que los usuarios solo necesitaban hacer clic en un icono para iniciar la aplicación. Esto eliminó la necesidad de ejecutar comandos complejos o configurar manualmente el entorno de trabajo, lo que resultó en una experiencia de usuario más amigable, especialmente para aquellos que no contaban con conocimientos avanzados en programación o manejo de sistemas.

Con esta solución, el sistema de adquisición de imágenes se volvió más accesible, reduciendo las barreras de uso y garantizando que las funcionalidades del software estuvieran disponibles de forma rápida y sin complicaciones. Esto no solo mejoró la operatividad del sistema, sino que también aumentó la adopción y el uso efectivo de la herramienta en los procesos de experimentación microbiológica.

9.11 Impacto de propuesta en el sistema de adquisición de imágenes

Como resultado de la propuesta planteada en esta tesis y en línea con los objetivos específicos, se implementaron las funcionalidades de edición de imágenes y detección de contornos en el sistema de adquisición de imágenes.

Estas mejoras permitieron ampliar las capacidades del sistema, proporcionando herramientas adicionales para el análisis de las fotografías capturadas durante la experimentación microbiológica.

La integración de la funcionalidad de edición de imágenes permitió a los usuarios realizar ajustes de brillo, saturación y nitidez directamente desde la interfaz del sistema. Esto facilitó la optimización de la visualización de las imágenes de acuerdo con las necesidades específicas de cada experimento, mejorando así la calidad y la precisión de las observaciones.

Por otro lado, la funcionalidad de detección de contornos se implementó para identificar de manera precisa los bordes y áreas de crecimiento en las muestras microbiológicas. El sistema aplica algoritmos de procesamiento de imágenes que detectaban los contornos de las colonias de manera automática, resaltando las áreas de interés con líneas destacadas que facilitaban la visualización y el análisis de las etapas de crecimiento de las colonias. Esto resultó ser una herramienta fundamental para la caracterización y cuantificación de las colonias en cada fase del experimento. El algoritmo de detección de contornos aplicado en el sistema es capaz de procesar las imágenes de las muestras, mostrando los resultados antes y después del análisis. Esta visualización permitió a los usuarios identificar de manera clara las áreas de crecimiento dentro de las placas de cultivo, ofreciendo un análisis visual directo y efectivo.

En conclusión, la implementación de estas funcionalidades cumplió con los objetivos propuestos, enriqueciendo el sistema de adquisición de imágenes y aportando un valor significativo al proceso de experimentación. Además, estas herramientas sentaron una base sólida para futuras mejoras del sistema, permitiendo la posibilidad de incorporar nuevas técnicas de procesamiento de imágenes que contribuyan al análisis microbiológico de manera más detallada.

10. CONCLUSIONES

- **Desarrollo de la interfaz con el sistema de captura de imágenes:** Durante este proyecto, se desarrolló una interfaz intuitiva y funcional que facilitó la captura de imágenes en tiempo real. La interfaz permitió a los usuarios interactuar de manera eficiente con el sistema de adquisición de imágenes, brindando una experiencia de usuario mejorada. Esto no solo optimizó la usabilidad del sistema, sino que también minimizó la probabilidad de errores durante la captura de imágenes, garantizando la precisión y la calidad de los datos obtenidos.
- **Almacenamiento de imágenes en la Raspberry Pi:** Si bien inicialmente se planteó la integración de un servicio FTP para gestionar el almacenamiento de las imágenes capturadas, las dificultades técnicas y problemas de compatibilidad llevaron a reconsiderar esta opción. Finalmente, se optó por utilizar el almacenamiento interno de la Raspberry Pi como alternativa. Esta decisión permitió mantener las imágenes organizadas y accesibles de manera local, asegurando la integridad de los datos y una fácil gestión de los archivos sin la necesidad de una conexión a un servidor externo, lo que resultó en una solución más práctica para las necesidades del proyecto.
- **Propuesta de procesamiento de imágenes:** Se implementaron funcionalidades de procesamiento de imágenes dentro del sistema, como la edición de brillo, contraste y la detección de contornos, específicamente adaptadas para el análisis de experimentaciones microbiológicas. Esto permitió identificar y resaltar los contornos de crecimiento en las muestras, brindando información precisa sobre las diferentes etapas de crecimiento de las colonias. A través del uso de la biblioteca OpenCV y la integración de estas herramientas, se sentaron las bases para un análisis más detallado de las imágenes capturadas y se abrió la posibilidad de futuras mejoras en el sistema.

11. REFERENCIAS BIBLIOGRÁFICAS

- Montanez, A., Trasante, T., Silva, C., & Imbert, D. (2021). Aprendizaje por indagación en la enseñanza de la Microbiología de suelos: diseño participativo de herramientas para la experimentación. *Revista de Educación en Biología*, 24(1), 102-118.
- Villa, A. M. C., & Sagnay, F. G. (2021). Gestión de información y análisis de datos clínicos en los laboratorios de microbiología, una revisión de la literatura. *Serie Científica de la Universidad de las Ciencias Informáticas*, 14(10), 119-130.
- Alvear-Puertas, V., Rosero-Montalvo, P., Peluffo-Ordóñez, D., & Pijal-Rojas, J. (2017). Internet de las Cosas y Visión Artificial, Funcionamiento y Aplicaciones: Revisión de Literatura. *Enfoque UTE*, 8, 244-256.
- Culjak, I., & Abram, D. (2012). A brief introduction to OpenCV. *MIPRO*, 1725-1730.
- Bradski, G., & Kaebler, A. (2008). *Learning OpenCV*. Cambridge: O'Reilly Media, Inc.
- Ansari, A. N., Sedky, M., Sharma, N., & Tyagi, A. (2015, January). An Internet of things approach for motion detection using Raspberry Pi. In *Proceedings of 2015 International Conference on Intelligent Computing and Internet of Things* (pp. 131-134). IEEE.
- Tizne Mena, G. (2015). *Escáner 3D autónomo con RaspberryPi* (Doctoral dissertation, Universitat Politècnica de València).
- Marco Antonio Cruz Chávez, 2011, Bases de Datos, Conceptos y sus Características, Centro de Investigaciones en Ingeniería y Ciencias Aplicadas, Universidad Autónoma del Estado de Morelos.
- Arellano García, A. G. (2021). Diseño e implementación de una Raspberry Pi 3 Model B como sistema de monitoreo y visualización de una línea de producción de bebidas con PLC CONTROLLOGIX 1756.
- Van Rossum, G., & De Boer, J. (1991). Interactively testing remote servers using the Python programming language. *CWI quarterly*, 4(4), 283-303.
- Prabhat, P., Ram, S., Ward, E. S., & Ober, R. J. (2004). Simultaneous imaging of different focal planes in fluorescence microscopy for the study of cellular dynamics in three dimensions. *IEEE transactions on nanobioscience*, 3(4), 237-242.
- Morales, j. Una revisión de las tendencias recientes en el análisis de datos de percepción remota.
- Alvear-Puertas, V., Rosero-Montalvo, P., Peluffo-Ordóñez, D., & Pijal-Rojas, J. (2017). Internet de las Cosas y Visión Artificial, Funcionamiento y Aplicaciones: Revisión de Literatura. *Enfoque UTE*, 8, 244-256
- Code, V. S. (2019). Visual studio code. Recuperado el Octubre de.
- Mínguez, T. D. (2021). *Visión artificial: aplicaciones prácticas con OpenCV-Python*. Marcombo.
- Van Rossum, G., & De Boer, J. (1991). Interactively testing remote servers using the Python programming language. *CWI quarterly*, 4(4), 283-303.

- Arellano García, A. G. (2021). Diseño e implementación de una Raspberry Pi 3 Model B como sistema de monitoreo y visualización de una línea de producción de bebidas con PLC CONTROLLOGIX 1756 (Doctoral dissertation, Universidad de San Carlos de Guatemala).
- López Fernández, A. (2017). Establecimiento de la procedencia de muestras arqueológicas de obsidiana de Oriente Próximo mediante el programa estadístico Source Seeker 2.1 (Bachelor's thesis, Universitat Politècnica de Catalunya).
- Secada Carral, C. (2019). Plataforma portátil de pentesting basada en Raspberry Pi.
- Cafiero, A. J. (2018). La comunicación digital de derechos y gestión de Propiedad Intelectual en universidades: lecciones del Massachusetts Institute of Technology (MIT).
- Martínez Cervantes, A. T. (2016). Proyecto de estadía profesional. Ingeniería en Informática. Repositorio remeri.
- Rodríguez, C., & Vicente, R. D. (2015). ¿Por qué implementar Scrum? Revista Ontare, 3(1), 125-144.