0

Informe Proyecto Final

1 Luis Montoya C.C. 1007286308,

2 Isabella Vergara C.C.1001143472,

3 Isabella Valencia C.C. 1000414584 ,

4 Luisa Hernández C.C. 1086418571

*Bioingeniería, Facultad de Ingeniería, Universidad de Antioquia Sede Medellín*

Junio 04, 2024

**Introducción**

En el campo de la bioingeniería, los avances en el análisis y la recopilación de datos médicos de los pacientes están transformando la atención médica. Este proyecto se centra en el desarrollo de un sistema integral que registra y analiza datos críticos de pacientes, como la frecuencia cardíaca, la temperatura corporal y las imágenes médicas. Este sistema puede funcionar como un prediagnóstico fácil de usar tanto para pacientes como para el personal de salud, facilitando las consultas médicas y permitiendo el monitoreo continuo de la salud de los pacientes.

**Justificación**

La falta de un sistema de prediagnóstico en la bioingeniería presenta desafíos que afectan tanto a los pacientes como al sistema de salud en general. La ausencia de herramientas de prediagnóstico puede resultar en diagnósticos tardíos, errores médicos y tratamientos no personalizados, comprometiendo así la eficacia del cuidado médico y el bienestar del paciente. Además, la falta de un monitoreo continuo dificulta la detección oportuna de cambios críticos en la salud del paciente, lo que puede tener consecuencias graves.

La eficiencia en el uso de recursos médicos también se ve afectada, ya que la ausencia de datos en tiempo real impide una gestión adecuada y la priorización de la atención. Esto es muy problemático en situaciones de alta demanda médica, como emergencias. Además, la limitación en las capacidades de telemedicina impide el acceso a la atención médica a distancia, especialmente en áreas rurales o desatendidas.

La investigación médica y la educación de los profesionales de la salud también se ven restringidas sin un sistema robusto de recopilación de datos. La falta de datos limita la capacidad de los investigadores para desarrollar nuevos tratamientos y mejorar las prácticas clínicas, lo que ralentiza el progreso en la comprensión y tratamiento de diversas enfermedades.

En resumen, la implementación de un sistema de prediagnóstico es crucial para abordar estos desafíos y avanzar en la bioingeniería. Este sistema proporcionaría una solución integral que mejoraría la calidad, eficiencia y seguridad del cuidado del paciente, al tiempo que impulsaría la investigación y la educación en el campo de la salud.

**Metodología**

El desarrollo del sistema integral para el registro y análisis de datos críticos de pacientes se ha hecho con una metodología basada en la arquitectura Modelo-Vista-Controlador (MVC), que garantiza la separación de responsabilidades y facilita el mantenimiento y la escalabilidad del sistema. La arquitectura MVC se divide en tres componentes principales: el Modelo, que gestiona los datos y la lógica de negocio; la Vista, que se encarga de la interfaz de usuario y la presentación de datos; y el Controlador, que actúa como intermediario entre la vista y el modelo, gestionando la lógica de la aplicación y la interacción del usuario. Este enfoque modular permite que cada componente del sistema sea desarrollado y probado de manera independiente, asegurando un funcionamiento robusto y coherente.

Para el desarrollo del sistema se utilizó el lenguaje de programación Python debido a su versatilidad y amplio ecosistema de bibliotecas. La interfaz de usuario se implementó con PyQt5, una biblioteca que proporciona herramientas para crear interfaces gráficas complejas y altamente interactivas. Se diseñaron varias interfaces, incluyendo una ventana de inicio para el login ver *Fig.1*, una ventana de menú principal con opciones para agregar pacientes, consultar datos y analizar estudios, así como componentes visuales como botones, campos de texto y etiquetas para facilitar la interacción del usuario. La elección de PyQt5 permitió crear una experiencia de usuario intuitiva y agradable, esencial para el manejo de datos sensibles en un entorno médico.

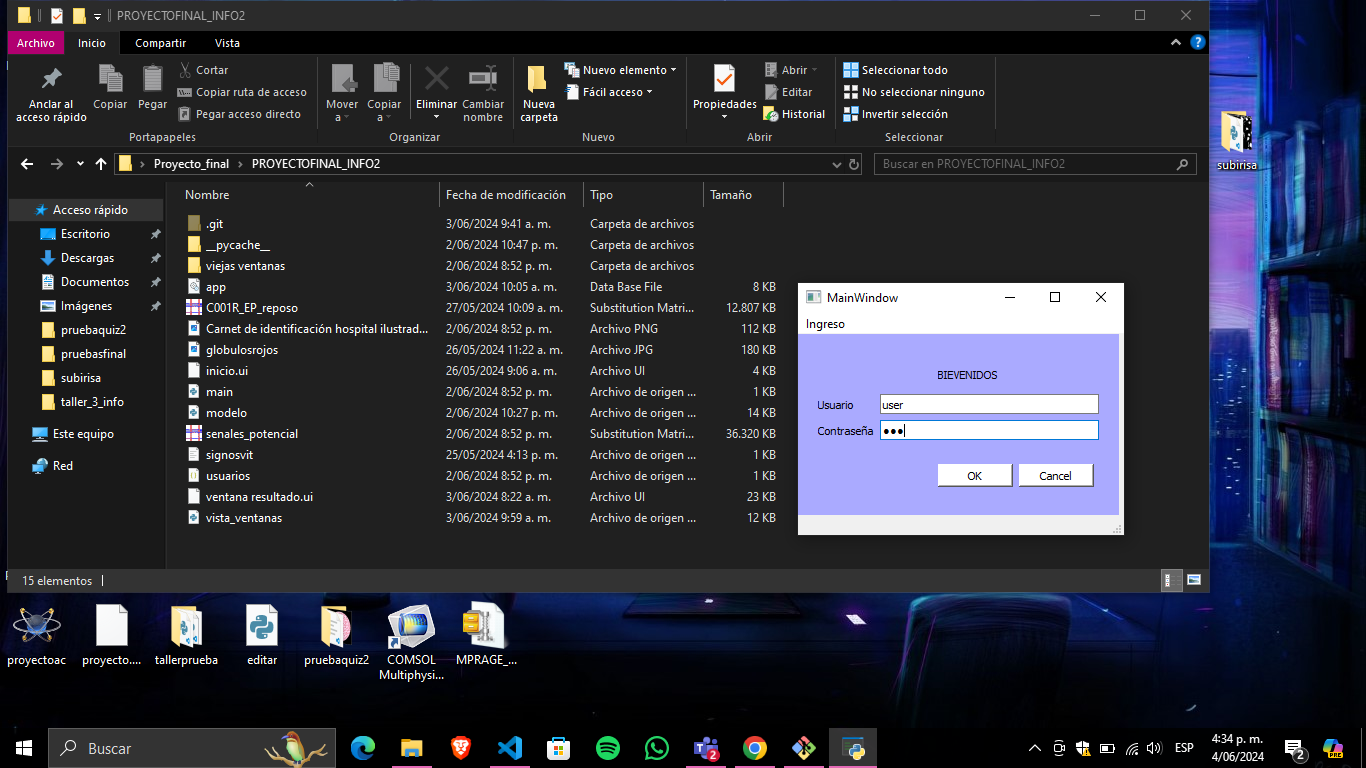


Fig 1. Interfaz de usuario, ventana de inicio de sesión.

La interfase de Fig 1. se usa line edit para que luego ser extraída la información e ingresarla

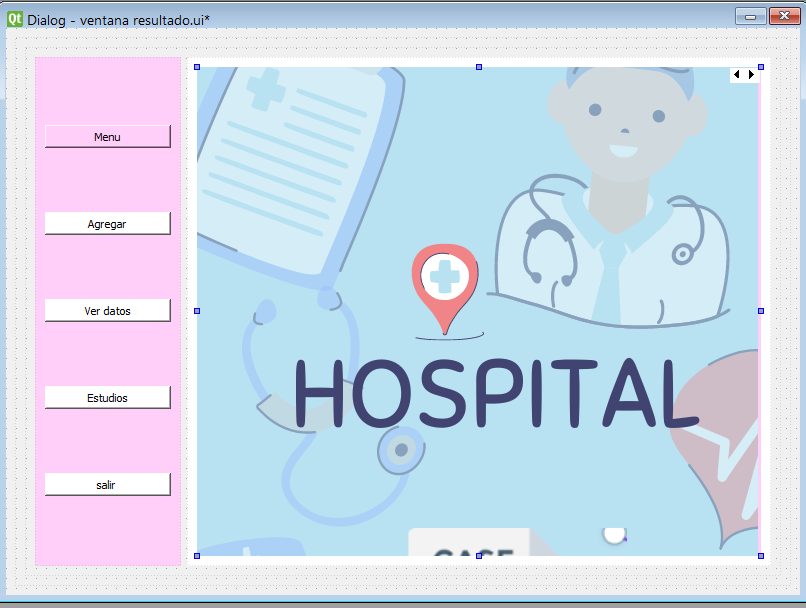


Fig 2. Interfaz de usuario, ventana de inicio de sesión.

Se crearon solo dos interfases con el objetivo de evitar la creación de varios controladores y tener que hacer múltiples conexiones entre las ventanas, se usó herramientas como stacked witged para este fin, también se usaron botones, para asociarlos desde el código con cada una de las ventanas del widget.

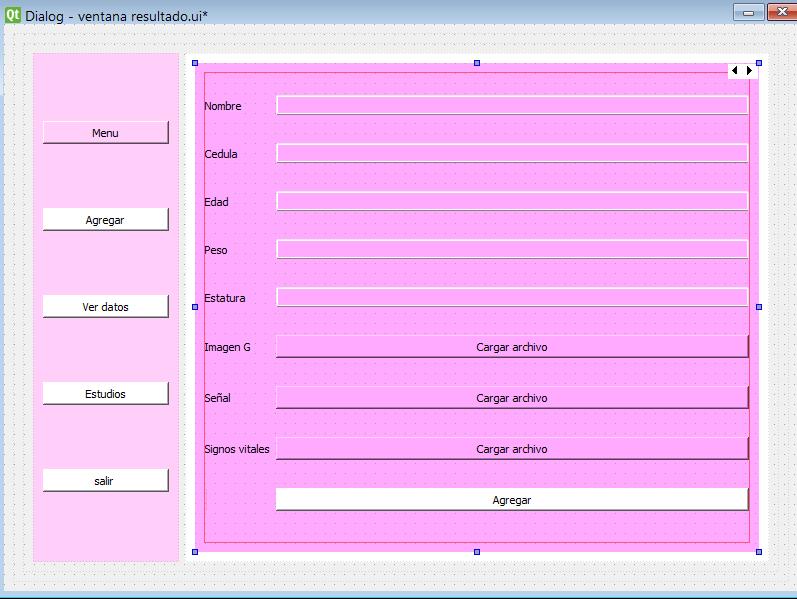


Fig 3. Pagina agregar de ventana menu

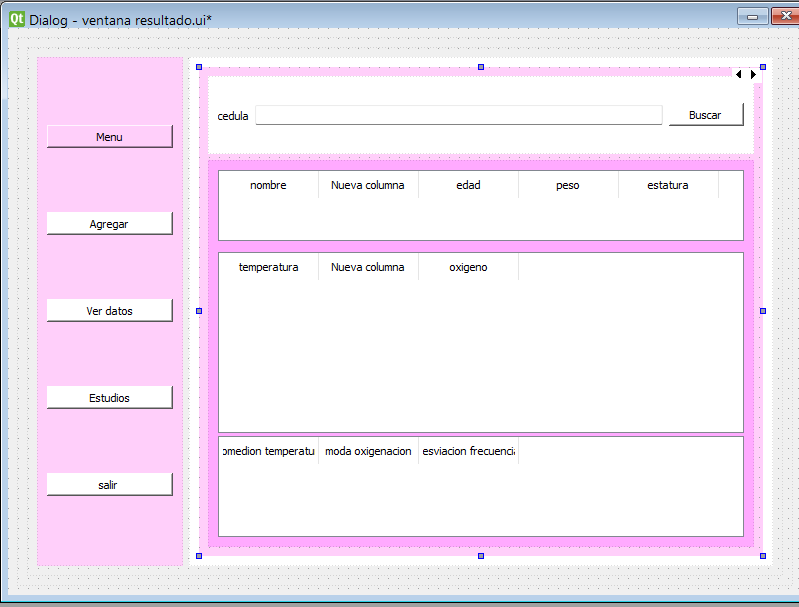


Fig 4. Pagina ver datos de ventana menú

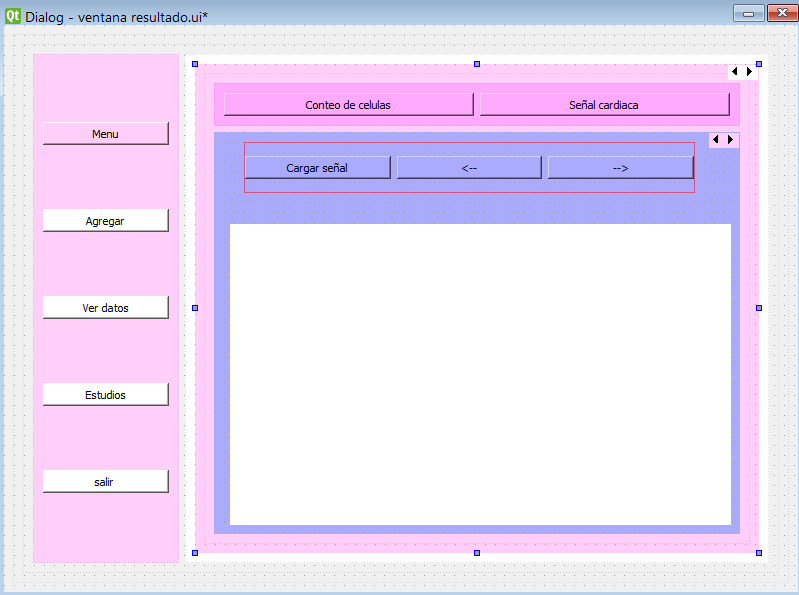
Se usan tablas con el objetivo de hacer ver más organizada la extracción de los datos del paciente y la información de los archivos CSV

Fig 5. Pagina estudios de ventana menú( función graficar señal)

En la ventana de la también hay stacked widget que permiten compactar dos funciones relacionadas a un solo botón del menú, también se agregaron frames para organizar mejor la distribución de las imágenes y señales graficar

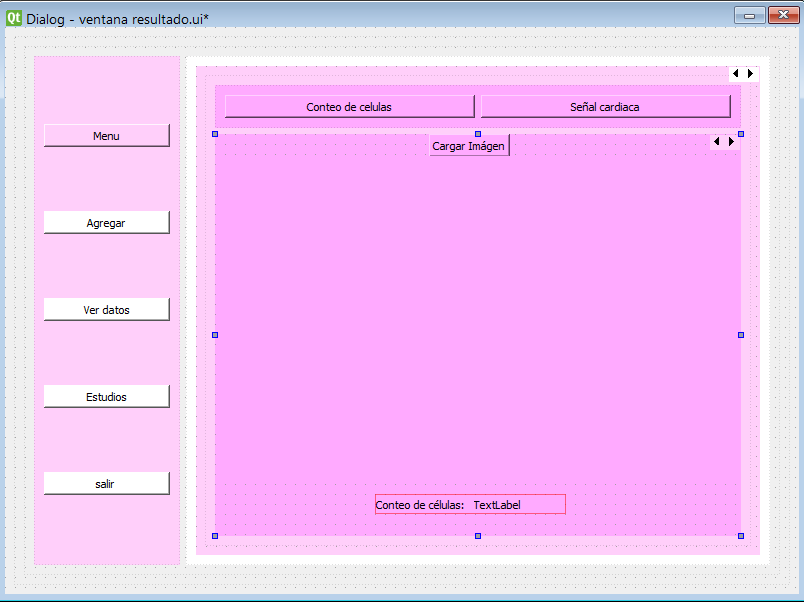


Fig 6. Pagina estudios de ventana menú (función muestra imagen y conteo de célula)

Las páginas agregadas optimizan le código, se crearon menor cantidad de clases y las conexión se hacen directamente desde una sola clase.

El sistema utiliza SQLite como base de datos para el almacenamiento de información de pacientes. SQLite es una opción adecuada debido a su simplicidad, su naturaleza de base de datos embebida y su compatibilidad con múltiples plataformas. Se implementaron tablas en SQLite para almacenar datos esenciales de los pacientes, como nombre, cédula, edad, peso, estatura y URLs de imágenes y señales médicas. Las operaciones de inserción, actualización y consulta de datos se gestionaron mediante comandos SQL ejecutados a través de la biblioteca sqlite3 de Python, asegurando que los datos se manipulen de manera eficiente y segura.

Para el procesamiento y análisis de imágenes médicas se utilizó OpenCV, una biblioteca de visión por computadora que ofrece numerosas funcionalidades para la manipulación de imágenes. OpenCV permitió realizar tareas como la conversión de imágenes a escala de grises, la detección de bordes y la visualización de imágenes procesadas. Estas capacidades son cruciales para el análisis detallado de imágenes médicas, facilitando la identificación de características importantes en los estudios de los pacientes. Además, se integraron funcionalidades para leer y analizar señales médicas utilizando SciPy y NumPy, bibliotecas especializadas en el manejo y análisis de datos científicos y numéricos. Estas bibliotecas permitieron la manipulación eficiente de señales en formato matricial y la realización de cálculos estadísticos para extraer información relevante, como la frecuencia cardíaca media.

La implementación del sistema incluyó también la capacidad de manejar archivos CSV y convertirlos a formato Excel, utilizando la biblioteca panda. Esto facilitó la gestión de grandes volúmenes de datos tabulares y su integración en la base de datos, mejorando la accesibilidad y el análisis de la información. La integración de estas diversas tecnologías y bibliotecas no solo mejoró la funcionalidad del sistema, sino que también garantizó una alta precisión y eficiencia en el procesamiento de datos críticos de los pacientes.

La metodología aplicada en el desarrollo de este sistema integral combina la arquitectura MVC con un conjunto robusto de tecnologías y bibliotecas de Python, lo que permite una gestión eficiente y análisis detallado de datos críticos de pacientes. La utilización de PyQt5, SQLite, OpenCV, SciPy, NumPy y pandas asegura que el sistema sea modular, escalable y capaz de manejar las complejidades inherentes al análisis de datos médicos. Esta metodología facilita el desarrollo y mantenimiento del sistema y proporciona una base sólida para futuras expansiones y mejoras, garantizando que los profesionales de la salud tengan una herramienta confiable y eficaz para gestionar la información de sus pacientes.

**Resultados**

La arquitectura propuesta ofrece una solución integral para la gestión de datos médicos y el análisis de información clínica. Comenzando con la ventana de inicio de sesión, que cuenta con campos de entrada para el nombre de usuario y la contraseña *ver Fig. 1*, junto con dos botones de acción claramente etiquetados para aceptar o cancelar. Estos elementos de interfaz, como los lineEdits para el nombre de usuario y la contraseña, junto con los pushButtons para las acciones de aceptar o cancelar, proporcionan una experiencia de usuario intuitiva y segura. Esta interfaz no solo garantiza la confidencialidad y la integridad de los datos médicos al restringir el acceso solo a usuarios autorizados, los cuales están previamente guardados en un archivo Json.

Al ingresar al sistema, la ventana principal presenta una serie de botones intuitivos que facilitan la interacción del usuario *ver Fig.2*. Al seleccionar el botón para ingresar un nuevo paciente, se despliega un formulario donde se pueden introducir los datos del paciente *ver Fig. 7*, los cuales se almacenan de manera segura en la base de datos. Aquí, los lineEdits permiten la entrada de datos, mientras que los pushButtons activan las acciones correspondientes, como la inserción de datos del paciente en la base de datos, cuando el paciente ha sido agregado correctamente notifica al usuario *ver Fig.8.* Esta funcionalidad simplifica el proceso de registro de pacientes y garantiza la disponibilidad de información completa y actualizada para su posterior análisis.

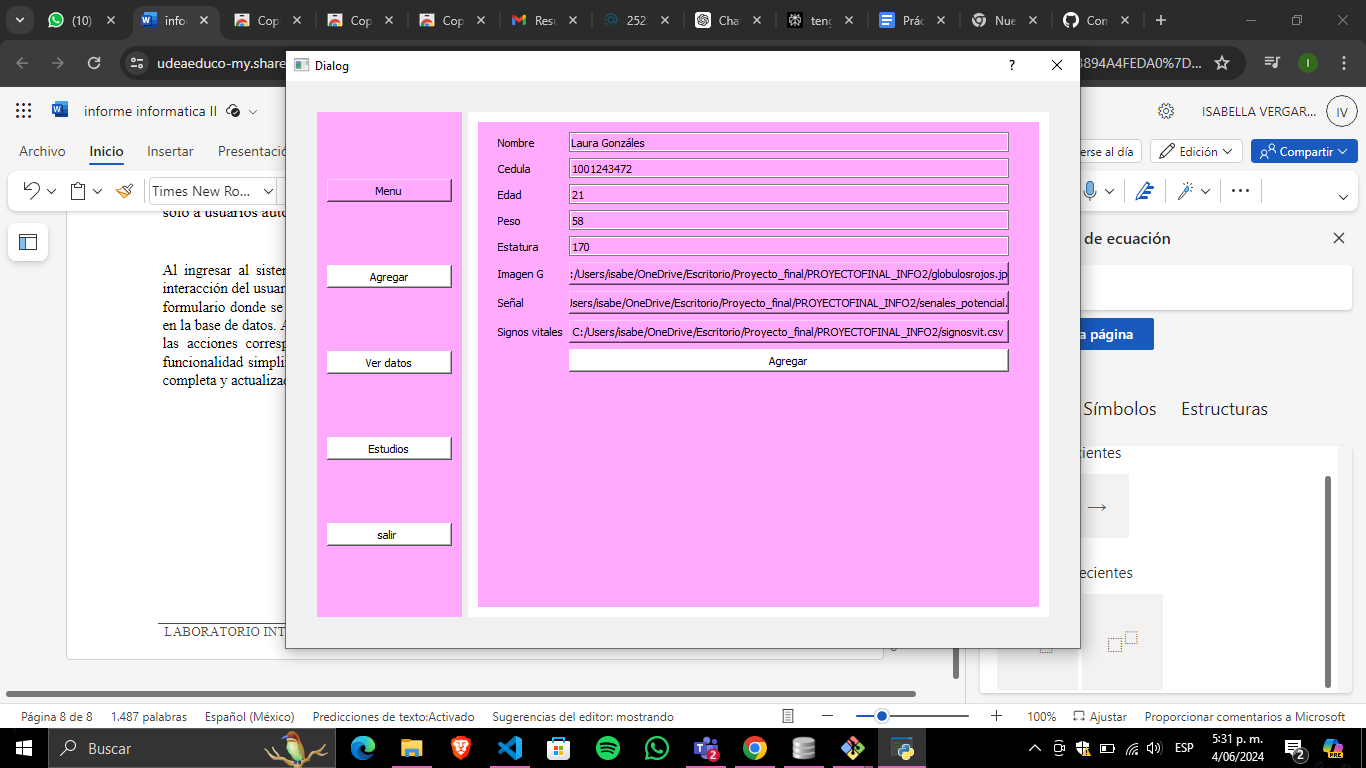


Fig.7 Interacción del usuario con el botón Agregar paciente del menú principal

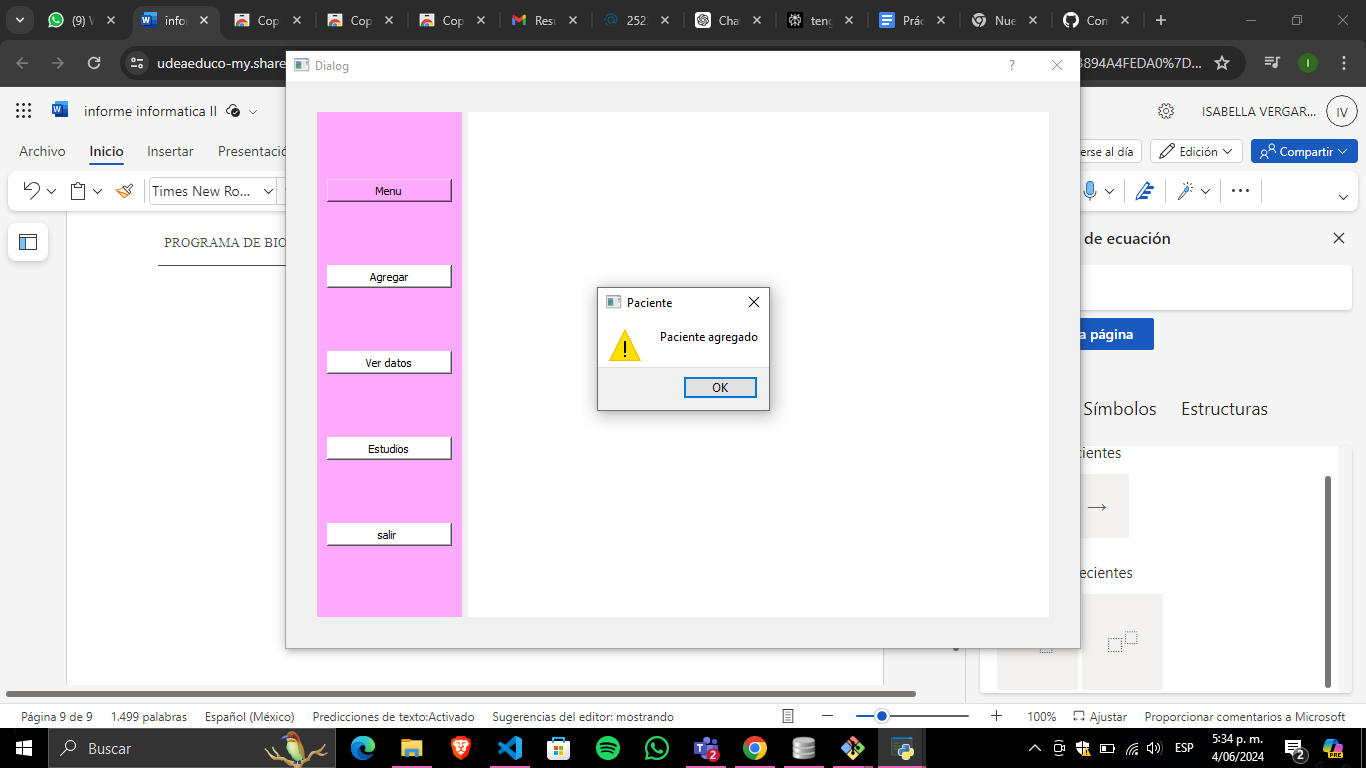


Fig.8 Interacción del usuario con el botón Agregar paciente del menú principal

La capacidad de cargar archivos, como imágenes para análisis con OpenCV, archivos .mat para visualizar señales con Matplotlib y archivos CSV para realizar estadísticas, añade una capa adicional de versatilidad al sistema. Los lineEdits permiten especificar los datos de los pacientes, mientras que los pushButtons activan la carga y procesamiento de los archivos. Esto permite la integración de diferentes tipos de datos médicos y facilita la realización de análisis avanzados para obtener información clínica relevante y precisa.

El menú principal ofrece acceso rápido a diversas funcionalidades, como la búsqueda de datos de pacientes por medio de la cédula de identidad en el botón del menú principal Ver datos. Al procesar los datos del paciente con su respectivo archivo CSV, el sistema muestra automáticamente estadísticas clave, como el promedio de temperaturas registradas, la moda de la oxigenación en sangre y la desviación de la frecuencia cardíaca *ver Fig. 9*. Aquí, los lineEdits pueden utilizarse para ingresar la cédula de identidad del paciente, mientras que los pushButtons activan la búsqueda y visualización de datos. Esta capacidad de generar estadísticas de manera automática agiliza el proceso de análisis y proporciona información valiosa para la toma de decisiones clínicas.

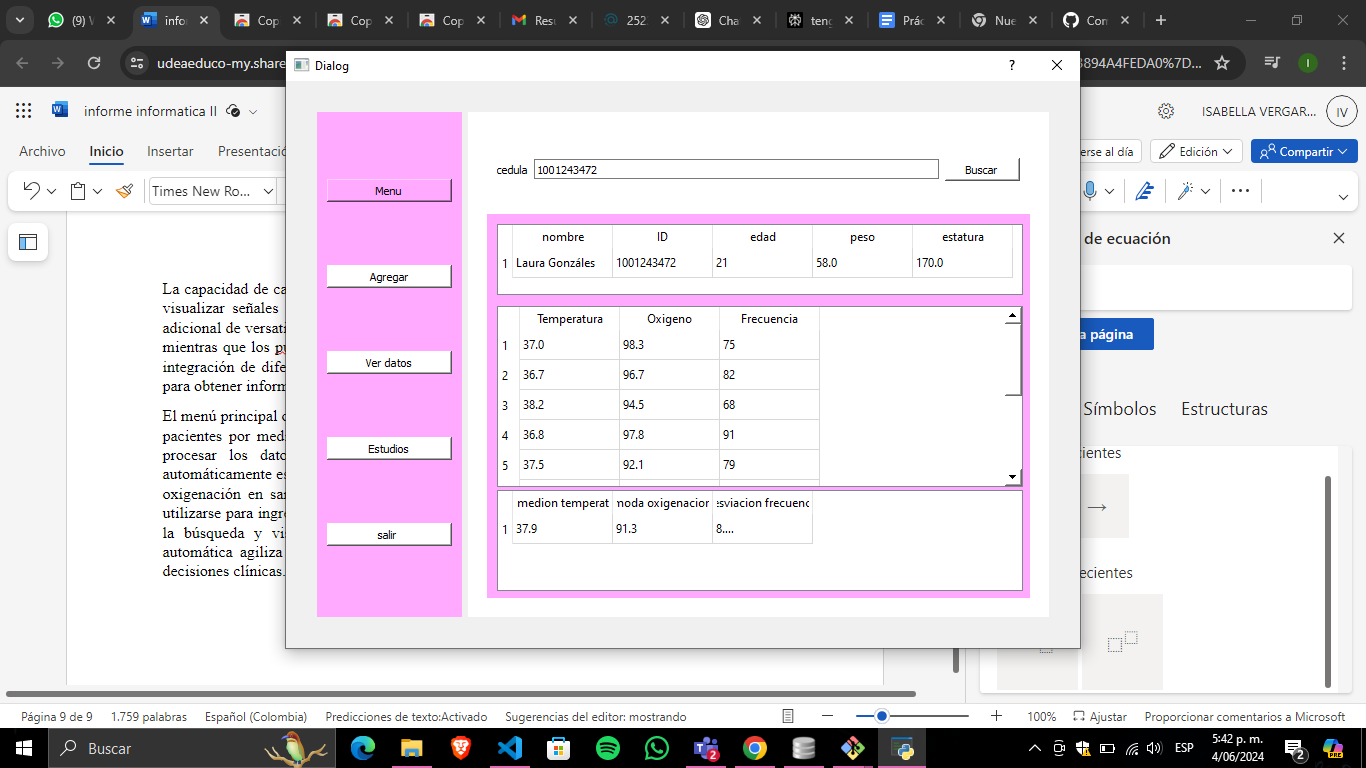


Fig.9 Interacción del usuario con el botón Ver datos del menú principal

En la sección de Estudios, los botones ofrecen una amplia gama de funcionalidades adicionales diseñadas para mejorar la capacidad de análisis y diagnóstico de los profesionales médicos. Además de cargar imágenes y visualizar gráficas de señales cardíacas, el sistema también integra herramientas específicas para realizar el conteo de células *ver Fig. 10*, una tarea fundamental en muchos análisis clínicos.

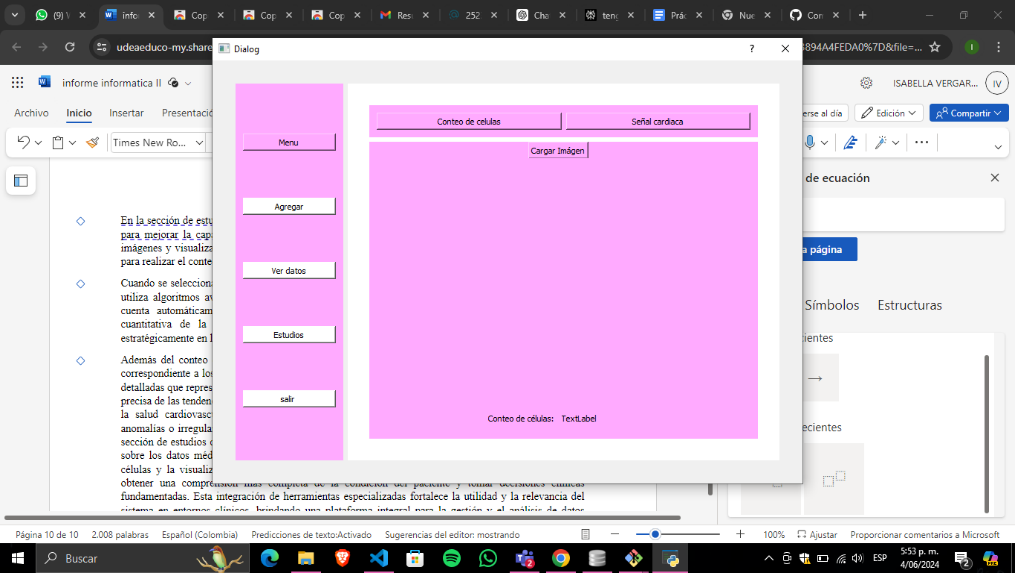


Fig.10 Interacción del usuario con el botón Estudios del menú principal

Cuando se selecciona el botón para realizar el conteo de células, se activa una función especializada que utiliza algoritmos avanzados para procesar la imagen cargada previamente. Esta función identifica y cuenta automáticamente las células presentes en la imagen, proporcionando así una evaluación cuantitativa de la muestra, además muestra la imagen transformada con las células etiquetadas. La información resultante del conteo de células se muestra en una text label, ubicada estratégicamente en la interfaz para facilitar la visualización y el análisis por parte del usuario *ver Fiq. 11*.

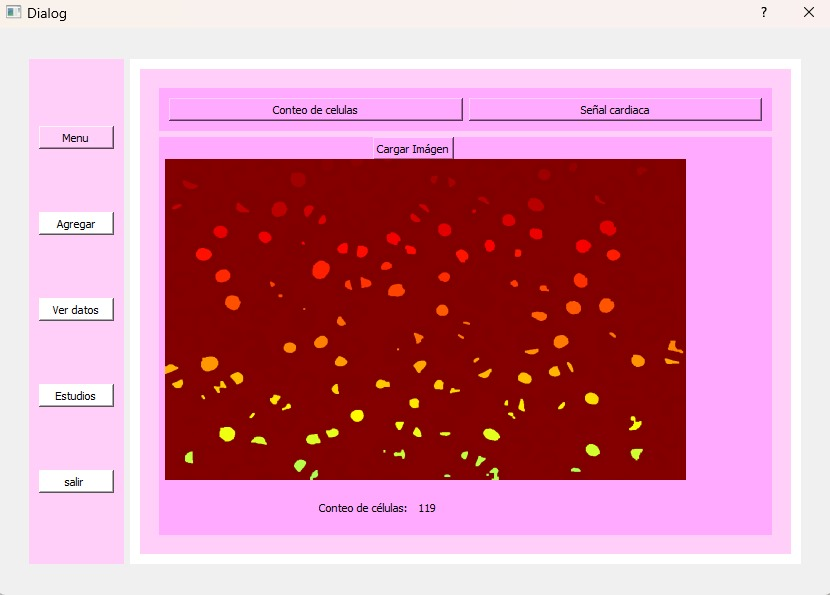


Fig.11 Prueba de interacción del usuario con el botón Conteo de células incluido en Estudios

Además del conteo de células, el sistema también permite a los usuarios visualizar la señal graficada correspondiente a los datos médicos del paciente en el archivo.mat . Al seleccionar el botón respectivo, se generan gráficas detalladas que representan la actividad cardíaca del paciente *ver Fig. 12*, lo que proporciona una visualización clara y precisa de las tendencias y patrones relevantes. Estas gráficas son herramientas valiosas para el análisis de la salud cardiovascular del paciente, permitiendo a los profesionales médicos identificar posibles anomalías o irregularidades en la función cardíaca. En conjunto, estas funcionalidades adicionales en la sección de estudios complementan la capacidad del sistema para realizar análisis detallados y avanzados sobre los datos médicos de los pacientes. Al proporcionar herramientas específicas para el conteo de células y la visualización de señales cardíacas, el sistema capacita a los profesionales médicos para obtener una comprensión más completa de la condición del paciente y tomar decisiones clínicas fundamentadas. Esta integración de herramientas especializadas fortalece la utilidad y la relevancia del sistema en entornos clínicos, brindando una plataforma integral para la gestión y el análisis de datos médicos.

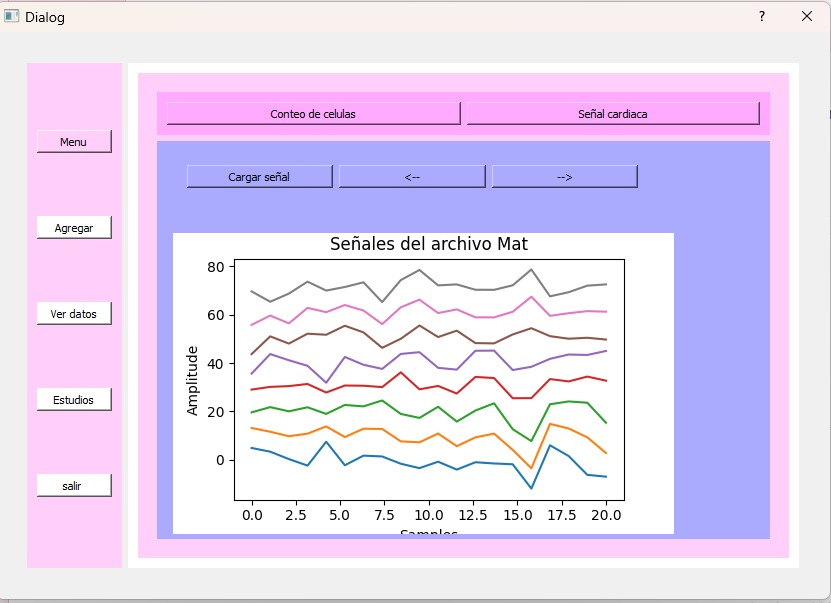


Fig.12 Prueba de interacción del usuario con el botón Señal Cardiaca incluido en Estudios

Referencias

[1] "How to use SQLite3 with Python," IONOS Digital Guide, Jul. 26, 2023. [Online]. Available: https://www.ionos.com/digitalguide/websites/web-development/sqlite3-python/. [Acceso: Jun. 4, 2024].

[2] OpenCV, "OpenCV Documentation," OpenCV, 2024. [Online]. Available: https://docs.opencv.org/4.x/. [Accessed: Jun. 4, 2024].

[3] Martin Fitzpatrick, "PyQt Basic Widgets," PythonGUIs, [Online]. Available: https://www.pythonguis.com/tutorials/pyqt-basic-widgets/. [Accessed: Jun. 4, 2024].

Para citar el artículo "Estadística en Python: Media, Mediana, Varianza y Percentiles - Parte III" del blog de Adrián Istán en formato IEEE, puedes utilizar la siguiente referencia:

[4] A. Istán, "Estadística en Python: Media, Mediana, Varianza y Percentiles - Parte III," Adrián Istán Blog, [Online]. Available: https://blog.adrianistan.eu/estadistica-python-media-mediana-varianza-percentiles-parte-iii. [Accessed: Jun. 4, 2024].