UNIVERSIDAD
DE ANTIOQUIA
Facultad de Ingeniería

PROYECTO FINAL INFORMÁTICA II 2025-1

Este proyecto tiene como objetivo principal el desarrollo de un **sistema interactivo para el análisis y visualización de datos biomédicos, siguiendo la arquitectura Modelo-Vista-Controlador (MVC) e integrado con una base de datos** . Este enfoque no solo reforzará sus habilidades de programación, sino que también les brindará una experiencia práctica en el desarrollo de aplicaciones robustas y bien estructuradas, algo fundamental en la bioingeniería.

Planteamiento del proyecto

Desarrollar un aplicativo de escritorio que permita a los usuarios cargar, procesar, analizar y visualizar diferentes tipos de datos biomédicos (imágenes médicas DICOM/NIfTI, tambié formatos jpg, png, señales de archivos .mat, datos tabulares, desde y hacia archivos csv, excel, y bases de datos SQL ó NoSQL), ofreciendo una interfaz gráfica intuitiva para la interacción.

El aplicativo debe tener un nombre, el cual uds se inventaran (sean creativos),

Características Requeridas

1. **Login:**
   - Ingresar un usuario que esté previamente ingresado en una base de datos (SQL o MONGO)
   - Debe tener dos tipos de usuarios , expertos en imágenes y experto en señales, en ambos casos se debe guardar información en bases de datos.
   - Según el usuario elegido se desplegará una ventana con un menú diferente alusivo a su área.

2. **Carga de Datos Diversos:**
   - **Imágenes Médicas:** Permitir la carga y visualización de archivos DICOM y NIfTI. Se espera que puedan extraer metadatos (PyDICOM) y manipular las imágenes (NumPy, OpenCV).
   - **Señales Biomédicas:** Carga de archivos .mat (por ejemplo, señales de EEG, ECG). Deberán manejar la estructura de estos archivos y extraer las señales (SciPy.io si es necesario, o manipulación directa de matrices con NumPy).
   - **Datos Tabulares/Bases de Datos:** Conexión a una base de datos (puede ser SQL o NoSQL para cargar datos de pacientes, resultados de análisis, etc. (Pandas para manipulación).

3. **Procesamiento y Análisis de Datos:**
   - Implementación de funciones para el procesamiento básico de imágenes (filtrado, binarización, umbralización, transformación morfológica, detección de bordes con OpenCV).
   - Capacidad para realizar análisis estadísticos básicos sobre las señales o datos tabulares (NumPy, Pandas).
   - Algoritmos simples de procesamiento de señales (ej. filtrado, detección de picos).

4. **Visualización Interactiva:**
   - Uso de Matplotlib para graficar imágenes (imshow), señales (plot) y datos tabulares (barras, histogramas, pies, etc).
   - La interfaz PyQt deberá permitir la interacción con las visualizaciones

5. **Arquitectura MVC:**
   El proyecto debe estar claramente estructurado en un Modelo (lógica de negocio, manejo de datos, procesamiento), una Vista (interfaz gráfica de usuario con PyQt) y un Controlador (manejo de eventos de la UI, comunicación entre Modelo y Vista). Es crucial que estas capas estén bien desacopladas.

Requerimientos técnicos:

Para imágenes DICOM:

1. Al igual que en el último parcial, se debe extraer la información del paciente, pero esta vez se debe guardar en la bases de datos elegida, por lo que se debe crear una tabla (sql) o colección (nosql)según el caso, con la estructura e información, y una columna o llave adicional que guarde la ruta donde queda almacenado la carpeta DICOM y otra con el archivo NIFTI.
   NOTA: No importa que la información esté anonimizada.
2. Para la visualización deben reconstruir la imagen 3D y contar con una interfaz que visualice las imágenes en los 3 ejes; Sagital, coronal y axial, de forma conjunta, independiente del archivo que se cargue (En el interior de la interfaz) , **NO** se aceptan ventanas emergente creadas con plt.show(), estas imágenes se deben visualizar todos los cortes usando la herramienta **slider de Pyqt**., por tanto deben haber 3 sliders correspondientes a cada plano.
3. Y una última función donde convierten a NIFTI, las carpetas de archivos DICOM.

Para las imágenes JPG y PNG:

1. La interfaz debe permitir hacer procesos de cambio de espacios de color, ecualización de la imagen, binarización cierre y apertura y una opción de conteo de células (para imágenes de este tipo), el usuario debe poder manipular el tamaño de kernel.
2. **INVESTIGAR** sobre algún otro método de OPENCV o librería que no se haya mostrado en clase e implementarlo en esta sección de imágenes en este formato. **NOTA:** Si implementan una librería diferente pueden adaptarla al aplicativo en general como uds lo consideren.

Para archivos MAT:

1. Debe mostrar las llaves asociadas en un QComboBox, para que el usuario elija la llave correspondiente al array, en caso equivocarse que saque un QmessageBox con un mensaje que diga no es un arreglo, vuelva a intentarlo. En caso de elegir la llave correcta , debe visualizar las señales en la ventana de forma conjunta, cómo se ha mostrado en clase, adicional, la interfaz debe poder permitir graficar un canal o segmento de canales en un intervalo dado por el usuario. En el grafico se debe mostrar el título del gráfico, las leyendas y los nombres de los ejes.
2. Debe haber un botón para calcular promedio a lo largo del eje 1 , y mostrar el resultado. En un gráfico tipo stem en la interfaz

Para archivos CSV:

1. Tras cargar un archivo CSV, la interfaz debe permitir, elegir dos columnas para hacer un gráfico de dispersión (scatter)
2. Debe tener un área(Qtable) donde se pueda ver los datos cargados.

Para la interfaz gráfica:

1. SOLO SE ACEPTARÁ EN **PYQT**
2. Las interfaces, además del nombre dado por uds, deben tener un diseño personalizado (colores, fuentes letras, iconos, imágenes, etc…)

Para la base de datos:

1. Como ya notaron, debe haber una tabla para usuarios del aplicativo, otra para los archivos DICOM y NIFTI
2. Crear otra tabla o colección para los demas tipos de archivos trabajados, esta tabla debe tener un código identificador ,tipo de archivo trabajado (csv, mat, jpg, png), nombre del archivo cargado, fecha en la se trabajó , y ruta de su ubicación.

ENTREGABLE

1. Diagrama de clases (con LUCIDCHART)
2. Diseño de las interfaces (Son pantallazos de las interfaces que hicieron)
3. Código funcionando (Un comprimido con todos los archivos, bases de datos, etc)
4. Link de GITHUB
5. Manual de usuario
6. Sustentación

OBSERVACIONES

a. La nota se asignará de acuerdo con la calidad de los trabajos realizados verificando las condiciones y comparando entre los diferentes trabajos de manera que se favorezca una sana competencia.
b. Los trabajos que adicionen funcionalidades extras al aplicativo (y las sustenten) a las ya solicitadas, sumaran puntos adicionales para mejorar notas.
c. El proyecto será en grupos de 3 o 4 personas
d. Se bebe utilizar la arquitectura MVC, sin esto no se califica.
e. Es claro que NO deben pasárselo entre uds, cualquier intento de fraude será un cero (0.0).
f. Sabemos que el usaremos IA , pero háganlo responsable y éticamente, chicos, es solo una herramienta que tiene como objetivo ayudar a potenciar sus conocimientos.
g. El trabajo debe ser cargado en el TEAMS, solo una vez, es decir, uno de los dos integrantes hará la carga a más tardar el día martes 15 DE JULIO DE 2025 A LAS 11:59PM. IMPORTANTE: EL TRABAJO NO SE RECIBIRÁ DESPUÉS DE ESTA HORA, ES DECIR A LAS 12:00AM ESTE 20% SERÁ CERO (0.0).
h. Si al ejecutar el script hay algún error, sin importar cual sea, el trabajo se calificará sobre 3.9.