



**UNIVERSIDAD
DE ANTIOQUIA**

Facultad de Ingeniería

Informática 2: Unidad 3 – Taller Evaluativo

Introducción a la informática médica.

Monitor: Juan Esteban Pineda Lopera
jesteban.pineda1@udea.edu.co

1 Objetivo.

Desarrollar una aplicación en Python para automatizar la lectura, extracción y almacenamiento de metadatos de archivos DICOM, aplicando los estándares de la informática médica y herramientas de software libre.

2 Introducción y contexto.

En la informática médica, los sistemas de comunicación y archivado de imágenes (PACS) son fundamentales. Estos sistemas se comunican utilizando el estándar DICOM para garantizar la interoperabilidad. En este taller, se simulará una parte de este flujo: Se darán un conjunto de estudios médicos en formato DICOM, y la tarea será procesarlos para hacer su información accesible y consultable, como lo haría un software real tipo ‘dcm4chee’.

Se utilizará la librería ‘pydicom’ para interactuar con los archivos y se almacenarán los resultados en una base de datos, demostrando una aplicación práctica de bioingeniería en el manejo de datos clínicos.

3 Requerimientos e instrucciones.

A continuación se describen los requisitos evaluativos del trabajo. Este trabajo debe entregarse como un repositorio en *GitHub*. Puede realizarse individualmente o en parejas. Debe dejarse claro en el archivo `README.md` el nombre de los integrantes del equipo.

3.1 Preparación del entorno

- Crear un entorno virtual de *Python* para el proyecto.

- Instalar las librerías necesarias.
- Descargar un conjunto de archivos *DICOM* de prueba. Puede usar los que provee la librería `pydicom` en su **repositorio** (en el directorio `data_store/data`), o cualquier otra base de datos que encuentre libremente.

4 Desarrollo de la aplicación.

Su aplicación deberá ser capaz de ejecutar los siguientes pasos de forma organizada. Se recomienda el uso de Programación Orientada a Objetos (POO) para encapsular la lógica, por ejemplo, creando una clase ProcesadorDICOM.

4.1 Carga de archivos DICOM.

- Implementar una función o método que reciba la ruta de un directorio.
- Este método debe escanear el directorio, identificar todos los archivos DICOM y cargarlos utilizando `pydicom.dcmread()`. Debe asegurarse de manejar posibles errores si un archivo no es un DICOM válido.

4.2 Extracción de metadatos.

Para cada archivo DICOM cargado, extraer la siguiente información. Investigar el nombre del *tag* correspondiente en el estandar DICOM:

- Identificador del paciente.
- Nombre del paciente.
- Identificador único del estudio.
- Descripción del estudio.
- Fecha del estudio.
- Modalidad de la imagen (CT, MR...).
- Número de filas de la imagen.
- Número de columnas de la imagen.

Nota: Es muy posible que alguno(s) de los tags no estén disponibles debido a procesos de anonimizado; debe manejar estos casos donde algún tag no esté presente en algún archivo.

4.3 Estructuración de los datos.

Almacenar los metadatos extraídos de todos los archivos en un **Dataframe de Pandas**. Cada fila debe representar un archivo *DICOM* y cada columna uno de los *tags* extraídos.

4.4 Análisis de imagen.

Para cada imagen *DICOM* (accesible en el atributo `.pixel_array`), realizar un cálculo simple usando la librería `numpy` del valor de intensidad promedio de los píxeles de cada imagen. Añadir este valor como una nueva columna llamada **IntensidadPromedio** en el **Dataframe** creado anteriormente.

4.5 Informe y discusión.

En el repositorio, además del código, debé ir adjunto el `README.md` que incluya de manera organizada:

1. Breve descripción del proyecto.
2. Explica brevemente por qué DICOM y HL7 son cruciales para la interoperabilidad en salud y en qué se diferencian conceptualmente.
3. Pregunte teórica: ¿Que relevancia clínica o de pre-procesamiento podría tener el análisis de la distribución de intensidades en una imagen médica.?
4. Mencionar dificultades encontradas y la importancia de las herramientas de *Python* para el análisis de datos médicos.

5 Rúbrica de evaluación.

Table 1: Tabla de Criterios de Evaluación

Criterio de Evaluación	Descripción	Porcentaje
1. Funcionalidad del Código	El script se ejecuta sin errores y cumple con todos los pasos solicitados (carga, extracción y análisis).	40%
2. Calidad del Código y POO	El código está bien estructurado, es legible y utiliza correctamente funciones y/o clases para organizar la lógica.	25%
3. Correcta Extracción de Metadatos	El programa extrae correctamente todos los tags DICOM solicitados y los organiza en un DataFrame de Pandas.	15%
4. Informe y Discusión (<code>README.md</code>)	El informe responde a todas las preguntas teóricas, demostrando comprensión de los conceptos de estándares y software libre.	20%
Total		100%