



**UNIVERSIDAD  
DE ANTIOQUIA**

**Facultad de Ingeniería**

## **Informática 1: Unidad 3 y 4 – Taller Evaluativo**

### Gestión de información en computación y Manejo de bases de datos.

Monitor: Juan Esteban Pineda Lopera  
[jesteban.pineda1@udea.edu.co](mailto:jesteban.pineda1@udea.edu.co)

---

### **Objetivo.**

Conceptos como algoritmos de ordenamiento, expresiones regulares y manejo de archivos (.txt, .csv, .json) permiten estructurar adecuadamente grandes volúmenes de datos generados por sensores, instrumentos clínicos y plataformas de monitoreo. Además, la capacidad de usar herramientas del sistema para manipular rutas, directorios y registros es esencial para garantizar procesos reproducibles, trazables y compatibles con flujos de trabajo de análisis biomédico.

De igual forma, la integración de Python con bases de datos NoSQL como MongoDB resulta especialmente relevante para escenarios donde se requiere almacenar y consultar información heterogénea, longitudinal o de alta variabilidad, como señales fisiológicas, historiales clínicos estructurados y metadatos experimentales. Estas tecnologías, junto con el uso adecuado de control de versiones mediante GitHub, constituyen competencias fundamentales para el desarrollo de soluciones computacionales modernas en bioingeniería, donde la confiabilidad, organización y disponibilidad de los datos son pilares para la investigación, el diagnóstico asistido por computador y la toma de decisiones basada en evidencia.

### **Ítems de evaluación**

#### **1. Procesamiento y ordenamiento de datos (15%)**

Generar al menos 50 registros biomédicos aleatorios, con los siguientes campos:

- `id` (Identificación), de la forma "ID-001".
- `fr` (Edad), de la forma "18 Años".

- fc (Frecuencia cardiaca), de la forma "090ppm".
- spo2 (Saturación periférica de oxígeno), de la forma "95%".

Además, ordenar los datos por frecuencia cardiaca.

## **2. Gestión de archivos: TXT, CSV y JSON (15%)**

- Guardar los datos ordenados en formatos .txt y .csv.
- Exportar e importar los datos en formato .json.
- Implementar manejo de excepciones y verificación de rutas.

## **3. Herramientas del sistema (IO, os, pathlib) (10%)**

- Crear estructura de carpetas (/data/txt, /data/csv, /data/json).
- Mover y organizar automáticamente los archivos generados.
- Generar un archivo log.txt con registro de acciones.

## **4. Validación con expresiones regulares (10%)**

- En el ingreso de datos, verificar que los campos de cada registro (id, fc, fr, spo2) cumplan el patrón definido en el apartado 1.

## **5. Gestión de base de datos NoSQL (MongoDB) (25%)**

- Conectar Python con MongoDB Atlas.
- Crear una colección llamada signos\_vitales.
- Insertar los registros validados.
- Realizar consultas:
  - Promedio de frecuencia cardiaca.
  - Documentos con SpO<sub>2</sub> menor a 94.
- Exportar consultas a resultados\_mongo.json.

## **6. Integración final del sistema (15%)**

- Crear un archivo main.py con un menú que permita:
  - Ingresar datos.
  - Exportar/Importar archivos.
  - Validar datos.
  - Conectarse a MongoDB y ejecutar consultas (a través del ID).

## **7. Uso de GitHub (10%)**

- Crear un repositorio privado.
- Incluir código y documentación en un README.
- Registrar al menos 5 commits con mensajes adecuados.

## **Rúbrica**

<b>Ítem</b>	<b>Porcentaje</b>
Procesamiento y ordenamiento de datos	15%
Gestión de archivos (TXT, CSV, JSON)	15%
Herramientas del sistema	10%
Expresiones regulares	10%
MongoDB	25%
Integración del sistema	15%
GitHub	10%