

### *Proyecto final Informática 1 – 2025\_2, Bioingeniería*

### **SISTEMA DE BITÁCORA PERSONAL DE SALUD Y BIENESTAR CON ANÁLISIS INTELIGENTE**

#### **Objetivo General**

Desarrollar una aplicación que permita a los usuarios registrar, visualizar y analizar su información personal de salud y estilo de vida. El sistema debe almacenar datos estructurados (hábitos, síntomas, mediciones) en una base de datos relacional, MySQL, y datos no estructurados (notas personales, fotos de alimentos, registros de sueño, etc.) en bases de datos no relacional, MongoDB. El objetivo es fomentar el autocuidado mediante el análisis de tendencias, alertas personalizadas y visualizaciones. Ya que cada diseño es diferente créenle un nombre (**sean creativos!!**).

#### **Módulos y Funcionalidades del sistema**

##### **1. Login y Roles (SQL)**

**Roles:** Administrador y Usuario

- ⌚ Solo el administrador podrá añadir, visualizar, modificar o eliminar usuarios y permisos (CRUD), estos datos, en base de datos relacional SQL.
- ⌚ Cada usuario accede solo a su información
- ⌚ Cada uno de los roles tiene acceso limitado según permisos (crear usuario, ver registro, cargar información, etc.). Es decir:

**Los Administradores** pueden:

- ⌚ Crear/Ver/modificar/eliminar administradores
- ⌚ Crear/Ver/modificar/eliminar Usuarios
- ⌚ Crear/Ver/modificar/eliminar registros de salud
- ⌚ Crear/Ver/modificar/eliminar archivos personales

**Los Usuarios** por su parte solo pueden acceder a su información:

- ⌚ Crear/Ver/modificar sus registros de salud
- ⌚ Crear/Ver/modificar/borrar sus archivos personales

En síntesis:

**Solo el administrador** tiene acceso total.

**El usuario** crea entradas en SQL (Registro Diaria de salud) y MongoDB (Notas personales y archivos adjuntos).

- ⌚ Las contraseñas deben almacenarse en CSV



### 2. Registro Diario de Salud (SQL)

Ingresos de datos diarios:

- ⌚ Estado de ánimo (escala 1–10)
- ⌚ Horas de sueño
- ⌚ Actividad física (tipo, duración)
- ⌚ Alimentación (descripción, foto opcional)
- ⌚ Síntomas (dolor, fatiga, etc.)
- ⌚ **Mediciones:** presión arterial, glucosa, bpm, peso, etc.

### 3. Notas Personales y Archivos Adjuntos (MongoDB)

- ⌚ Registro de notas libres sobre el día, sensaciones, eventos importantes.
- ⌚ Adjuntar archivos como fotos de comidas, capturas de apps de salud, etc. (Estas se reflejarán como rutas dentro de la colección)

**El Almacenamiento con metadatos (collection):** fecha, tipo, etiquetas.

### 4. Visualización y Análisis

- ⌚ Valores de tendencia (peso, glucosa, estado de ánimo, sueño, etc)
- ⌚ Detección de patrones (p.ej: Menos sueño → peor estado de ánimo)
- ⌚ Alertas personalizadas (p.ej: “3 días con presión alta”), estos análisis se deben hacer cada cierto tiempo de forma automática.

### 5. Búsqueda y Filtros

- ⌚ Buscar por fecha, palabra clave, tipo de entrada.
- ⌚ Filtros por síntomas, hábitos, estado de ánimo.

### 6. Exportación de Reportes

- ⌚ Generar un resumen semanal o mensual en JSON o TXT.
- ⌚ Incluye estadísticas, notas destacadas.

## Requerimientos Técnicos

### Bases de Datos

#### MySQL:

- ❖ Administrador (ID, nombre )
- ❖ Usuarios (ID, nombre, edad, correo, contraseña)
- ❖ Registros diarios (fecha, ID usuario, sueño, actividad, síntomas, mediciones)

#### MongoDB:

- ❖ Notas personales (id\_usuario, fecha, texto, etiquetas), a continuación un ejemplo:



```
{  
    "_id": "nota_001",  
    "id_usuario": "usr_123",  
    "fecha": "2025-11-16",  
    "texto": "Hoy me sentí con poca energía. Dormí mal y tuve dolor de cabeza  
leve.",  
    "etiquetas": ["dolor", "sueño", "ánimo bajo"],  
    "estado_animo": 4,  
    "ubicacion": "Rionegro, Santander",  
    "clima": {  
        "temperatura": 18,  
        "condicion": "lluvioso"  
    },  
    "actividad_destacada": "Caminata ligera de 20 minutos"  
}
```

- ❖ Archivos adjuntos (ruta, tipo, fecha, ID usuario, descripción), ), a continuación un ejemplo:

```
{  
    "_id": "archivo_045",  
    "id_usuario": "usr_123",  
    "fecha": "2025-11-16",  
    "ruta_archivo": "/uploads/usr_123/2025/11/almuerzo_saludable.jpg",  
    "tipo": "imagen",  
    "descripcion": "Almuerzo: ensalada de quinua con aguacate y pollo a la  
plancha",  
    "categoria": "alimentacion",  
    "metadatos": {  
        "calorias_estimadas": 520,  
        "horario_comida": "13:15",  
        "fuente": "foto tomada por el usuario"  
    }  
}
```

- ❖ De forma general se puede crear mas colecciones enfocadas en el registro inteligente de los datos para monitoreo, registro y trazabilidad. **Esto es opcional**

En general según lo que consulten tanto las tablas de SQL como las colecciones de Mongo pueden variar en estructura. **NOTA: Puntos extra si explican mas detalles**

### Funciones específicas:

- Las funciones CRUD para cada entidad deben estar implementadas para MySQL y MongoDB.

**NOTA:** Ambas bases de datos MySQL y MONGO deberá estar configurada con los siguientes parámetros:

- ⌚ Nombre de la base de datos: FP\_Info12025\_2
- ⌚ Usuario: informatica1
- ⌚ Contraseña: info2025\_2
- ⌚ Tablas: estarán nombradas y configuradas según lo considerado por cada equipo.



- b) Implementación de módulos de funciones para manejo de validación de datos (alfabéticos, numéricos, etc.) en un módulo aparte.
- c) Submenús específicos para acceder y visualizar información de ambas bases de datos.

### Manejo de excepciones:

- 🌀 Validaciones de los campos de entrada y manejo de errores utilizando try/except.
- 🌀 Implementar mensajes de error personalizados y validaciones en cada campo.

### Documentación:

El código debe estar debidamente documentado con explicaciones de funciones y lógica, accesible mediante la función help().

### Formato de entrega:

- 🌀 Entregar el código fuente y las bases de datos pre-cargadas en SQL y MongoDB. Es decir que las tablas ya deben contar con información previamente ingresada. Al igual que el archivo CSV y demás archivos creados, como evidencia de ejecución previa.
- 🌀 **SUSTENTACIÓN DEL PROYECTO**, presentación en vivo, explicando cada funcionalidad.

## **REQUERIMIENTOS DE ENTREGA:**

1. El trabajo se realizará en grupos de 3 INTEGRANTES
2. Deben subir el código a **GITHUB** y adicionar link en la entrega
3. Se deben entregar todos los archivos (.py) diseñados en Spyder o Visual Studio o en Jupyter Notebook (.ipynb) y el archivo de la base de datos en MySQL y MONGO, incluirá las tablas y la colecciones de información previamente cargada para probar. Deben asegurarse de enviar correctamente los archivos ya que la revisión se realizará única y exclusivamente con los archivos cargados en la tarea de TEAMS que se abrirá para tal efecto.
4. En la cabecera del script se deben colocar los nombres de los autores como un comentario (integrantes del grupo) y el código debe estar debidamente documentado. Descripción de las variables, descripción de los procesos y resultados a mostrar.
5. El trabajo debe ser cargado en TEAMS, solo una vez, es decir, uno de los dos integrantes hará la carga .
6. Cualquier intento de fraude o copia de código entre los diferentes equipos, por mínimo que sea, una función, un ciclo, lo que sea.; implicará una calificación de 0.0 para todos los involucrados sin posibilidad de recuperación.
7. Por otro lado sé que van a usar IA , pero chicos háganlo a conciencia como herramienta para aprender, tengan claridad sobre lo que están haciendo, entiendan que la idea no!! es que se los haga.
8. Si al ejecutar el script hay algún error, sin importar cual sea, el trabajo se calificará sobre 3.93.



# UNIVERSIDAD DE ANTIOQUIA

Facultad de Ingeniería

## Ejemplo de funcionamiento:

### 1. Inicio de sesión:

- ⌚ El sistema solicita al usuario el nombre de usuario y password
- ⌚ La contraseña ingresada se compara con el password almacenado en la base de datos
- ⌚ Si el usuario es autenticado correctamente, el sistema verifica el rol asignado con los permisos correspondientes.

### 2. Interfaz de menú según el rol:

#### Administrador:

1. Gestión de administradores
2. Gestión de Usuarios
  - a) Gestión de registros de salud
  - b) Gestión archivos personales
3. Cerrar sesión

#### Usuario:

1. Usuario se registra e inicia sesión.
2. Registra su día: durmió 6 horas, caminó 30 minutos, comió saludable, se sintió con energía.
3. Adjunta una foto de su almuerzo.
4. El sistema muestra que cuando duerme menos de 7 horas, su estado de ánimo baja.
5. Genera un reporte de su semana para reflexionar.

**NOTA:** Los menús propuestos no son obligatorios , en ese orden , pueden ser más opciones según el criterio , y lo investigado por uds, por lo que estan sujetos a cambios

