```
+---------------------+          +-------------------------+
|      tb_user        |          |       tb_csvData        |
+---------------------+          +-------------------------+
| usuarioID (PK)      |          | confidence              |
| nombreResponsable   |          | model                   |
| institucion         |          | processedImage          |
| username            |          | conductor               |
| password            |          | peaton                  |
| rol                 |          | bicicleta               |
| estado              |          | carro                   |
| cedula              |          | furgoneta               |
+---------------------+          | camion                  |
                                 | triciclo                |
                                 | bus                     |
                                 | moto                    |
                                 +-------------------------+

+---------------------+
|     tb_auditoria    |
+---------------------+
| lng_id_accion (PK)  |
| usuarioID           |
| dt_fecha            |
| str_ip_host         |
| str_nombre_tabla    |
| str_accion          |
| txt_descripcion     |
| txt_valor           |
| dt_fecha_creacion   |
| dt_fecha_actualizacion |
+---------------------+

Archivo CSV: detections_log.csv
---------------------------------
| Fecha & Hora | Conductor | Peatón | Bicicleta | Carro | Furgoneta | Camión | Triciclo | Bus | Moto |

Archivo CSV: confidence_averages.csv
--------------------------------------
| Clase | Conductor | Peatón | Bicicleta | Carro | Furgoneta | Camión | Triciclo | Bus | Moto |
```

Justificación del Diseño

1. **Modularidad y Separación de Responsabilidades:**
   - Mantener tablas independientes (`tb_user`, `tb_csvData`, `tb_auditoria`) asegura una clara separación de funciones y facilita el mantenimiento.
2. **Simplicidad y Rendimiento:**
   - Evitar relaciones complejas mejora la eficiencia de las consultas y permite que cada módulo escale de manera independiente.
3. **Flexibilidad en el Almacenamiento de Datos:**
   - Los datos procesados y los promedios de confianza se manejan de manera separada, optimizando el procesamiento especializado.
4. **Auditoría Independiente:**
   - Registrar las acciones de usuarios en `tb_auditoria` sin relaciones directas permite un seguimiento claro y no intrusivo de actividades.

Justificación de la No Relación

- **Seguridad y Control de Acceso:**
  - La gestión de usuarios es más segura y sencilla sin relaciones directas con otras tablas.

- **Eficiencia en el Procesamiento de Datos:**
  - Mantener los datos procesados independientes permite un procesamiento más eficiente y especializado.
- **Modularidad y Escalabilidad:**
  - Facilita el mantenimiento y la expansión de cada módulo sin afectar a los demás, permitiendo una escalabilidad eficiente.

Uso Sin Relaciones

- **Operación de Procesamiento de Imágenes:**
  - Un usuario sube una imagen, el backend de Python Flask la procesa y guarda los resultados en `tb_csvData` y archivos CSV, mientras se registra la acción en `tb_auditoria`.
- **Consulta de Resultados:**
  - Un administrador verifica las acciones en `tb_auditoria` sin necesidad de relacionarse con los datos procesados en `tb_csvData`.

Este diseño asegura un sistema robusto, fácil de gestionar y adaptable a futuras expansiones.

1. **Integridad de los Datos:** La base de datos relacional garantiza la integridad y consistencia de los datos mediante la definición de llaves primarias y la validación de datos en las columnas.
2. **Escalabilidad:** La estructura modular permite añadir nuevas tablas y campos según se necesiten, sin afectar significativamente la arquitectura existente.
3. **Seguridad:** El uso de roles y estados en la tabla de usuarios permite un control granular sobre el acceso y las acciones permitidas.
4. **Flexibilidad:** La configuración de las tablas y sus relaciones permite la integración con otros sistemas y servicios, facilitando la adaptación a futuros requerimientos.
5. **Optimización de Consultas:** La estructura de las tablas está diseñada para facilitar la realización de consultas eficientes, mejorando el rendimiento del sistema en general.

Asociación del Diseño de la Base de Datos con los Requisitos del Proyecto

1. **Gestión de Usuarios:**
   - **Requisito:** La aplicación debe gestionar diferentes tipos de usuarios con distintos roles (ADMIN, PARTICIPANTE, PUBLICO), manejar la autenticación y autorización, y almacenar información relevante de los usuarios.
   - **Tabla:** `tb_user`
     - **Columnas Relevantes:**
       - `usuarioID`: Identificador único para cada usuario.
       - `nombreResponsable`, `institucion`: Información adicional opcional.
       - `username`, `password`: Credenciales para autenticación.
       - `rol`: Control de acceso basado en roles.

        - `estado`: Indica si el usuario está activo o no.
        - `cedula`: Identificación adicional del usuario.
2. **Almacenamiento de Datos Procesados:**
    - **Requisito:** El sistema debe almacenar datos de las imágenes procesadas, incluyendo el nivel de confianza de los modelos y las detecciones realizadas.
    - **Tabla:** `tb_csvData`
        - **Columnas Relevantes:**
            - `confidence`, `model`, `processedImage`: Información del procesamiento de imágenes.
            - `conductor`, `peaton`, `bicicleta`, `carro`, `furgoneta`, `camion`, `triciclo`, `bus`, `moto`: Datos de detecciones por clase de objeto.
3. **Auditoría de Acciones:**
    - **Requisito:** Registrar las acciones realizadas por los usuarios en el sistema para fines de auditoría.
    - **Tabla:** `tb_auditoria`
        - **Columnas Relevantes:**
            - `lng_id_accion`: Identificador único para cada acción.
            - `usuarioID`: Relación con el usuario que realizó la acción.
            - `dt_fecha`, `str_ip_host`, `str_nombre_tabla`, `str_accion`: Detalles de la acción realizada.
            - `txt_descripcion`, `txt_valor`: Descripción y valores de la acción.
            - `dt_fecha_creacion`, `dt_fecha_actualizacion`: Timestamps para auditoría.
4. **Gestión de Detecciones y Promedios de Confianza:**
    - **Requisito:** El sistema debe procesar imágenes, guardar los resultados y calcular promedios de confianza para las detecciones.
    - **Archivos CSV:** `detections_log.csv` y `confidence_averages.csv`
        - **Columnas Relevantes:**
            - **`detections_log.csv`**: Almacena las detecciones realizadas en cada imagen procesada.
                - `Fecha & Hora`, `Conductor`, `Peatón`, `Bicicleta`, `Carro`, `Furgoneta`, `Camión`, `Triciclo`, `Bus`, `Moto`
            - **`confidence_averages.csv`**: Almacena los promedios de confianza por cada clase de objeto.
                - `Clase`, `Conductor`, `Peatón`, `Bicicleta`, `Carro`, `Furgoneta`, `Camión`, `Triciclo`, `Bus`, `Moto`

Justificación de la Asociación

- **Estructura Modular:** La división entre la gestión de usuarios, almacenamiento de datos procesados y auditoría permite una clara separación de responsabilidades y facilita el mantenimiento del sistema.
- **Seguridad y Control de Acceso:** La tabla `tb_user` asegura que solo los usuarios autorizados puedan acceder y realizar acciones en el sistema, controlando los permisos mediante roles y estados.

- **Rastreo de Actividades:** La tabla `tb_auditoria` permite un rastreo detallado de las actividades realizadas por los usuarios, proporcionando transparencia y capacidad de auditoría.
- **Almacenamiento Eficiente:** Las tablas `tb_csvData` y los archivos CSV permiten un almacenamiento estructurado y eficiente de los datos de procesamiento de imágenes y los resultados de las detecciones, asegurando que el sistema pueda manejar grandes volúmenes de datos de manera efectiva.
- **Facilidad de Integración:** La estructura de la base de datos permite una fácil integración con otros sistemas y servicios, así como la posibilidad de expandir y adaptar el sistema según las necesidades futuras.

Requisitos Específicos del Documento

- **Requisitos del Proyecto:**
  - Gestión de usuarios.
  - Almacenamiento y procesamiento de datos de imágenes.
  - Auditoría de acciones.
  - Cálculo y almacenamiento de promedios de confianza.
- **Estructura de la Base de Datos:**
  - Tablas en Node.js (Sequelize ORM) para la gestión de usuarios y auditoría de acciones.
  - Archivos CSV en Python (Flask) para el almacenamiento de datos procesados y promedios de confianza.
- **Objetivos del Proyecto:**
  - Proveer una plataforma eficiente y segura para la gestión de un modelo predictivo para el análisis de imágenes de células cancerígenas utilizando IA generativa.
  - Facilitar la auditoría y control de accesos mediante una adecuada gestión de usuarios y registro de acciones.
  - Asegurar la integridad y consistencia de los datos almacenados y procesados.