

# INFORME TÉCNICO – SISTEMA SIS-EXP

**Nombre:** Hernán Domínguez (8-970-599)

**Profesor:** Maryon Torres

**Materia:** Base de Datos II

## 1. Diseño de la base de datos

La base de datos SIS-EXP fue diseñada para gestionar procesos legales manejados por usuarios (abogados), aseguradoras, personas asociadas y expedientes.

### Tablas principales:

- **usuarios:** almacena a los abogados del sistema. Incluye id UUID, nombre completo, usuario único, contraseña (hash bcrypt) y timestamps.
- **personas:** representa individuos asociados a aseguradoras.
- **aseguradoras:** entidades aseguradoras con un vínculo a una persona.
- **expedientes:** tabla central que registra los casos legales, relacionando usuarios, aseguradoras y abogados.

## 2. Diagrama Entidad-Relación

- Un usuario puede tener muchos expedientes.
- Una aseguradora está asociada a una sola persona.
- Una aseguradora puede estar presente en múltiples expedientes.
- Un expediente pertenece a un único usuario, una única aseguradora y un único abogado.

Representación conceptual: usuarios (1) → (N) expedientes ← (1) aseguradoras → (1) personas.

Para generar una mejor experiencia en la visualización del diagrama, hemos generado este a través de la herramienta Mermaid Live, al cual puede ingresar haciendo [click aquí](#).

### **3. Normalización a 1FN, 2FN y 3FN**

- **1FN:** todos los atributos son atómicos; no existen listas ni valores compuestos.
- **2FN:** todas las tablas dependen completamente de su clave primaria (UUID), sin dependencias parciales.
- **3FN:** se eliminaron dependencias transitivas. Por ejemplo, la información personal de una aseguradora se almacena en la tabla personas, evitando duplicaciones.

### **4. Integración con Python**

A través del archivo *conexion.py* se implementan:

- Conexión segura mediante variables de entorno con el uso de la librería dotenv.
- Funciones utilitarias `getConnexion` y `getCursor`, para conectar la base de datos y obtener el cursor para ejecutar las consultas.
- Funciones CRUD como `crearExpediente`, `encontrarUsuario`, `getExpedientes`, `getExpedientesConteo` y `getExpedientesTotales`.

El archivo *datos.py* crea tablas, inserta usuarios, aseguradoras, personas y expedientes, además de generar vistas SQL. En otras palabras, es quien inicializa la base de datos desde 0.

### **5. Vistas SQL**

- **vista\_expedientes:** combina expedientes, aseguradoras, personas y usuarios para consultar información consolidada.
- **vista\_conteo\_expedientes:** devuelve el conteo de expedientes por usuario y estado.
- **vista\_expedientes\_totales:** vista avanzada que permite filtrado por aseguradora, tipo de proceso y abogado.

## 6. API Rest con Flask

### Endpoints principales:

- POST /login: autenticación con bcrypt.
- POST /expedientes: crea un expediente validando campos obligatorios.
- GET /expedientes: lista expedientes de un usuario.
- GET /expedientes/fecha/: filtra por rango de fechas.
- GET /expedientes/conteo: devuelve estadísticas.
- GET /expedientes/totales: filtrado avanzado para reportes.

## 7. Lecciones Aprendidas

1. Las vistas simplifican drásticamente el backend y evitan múltiples joins repetitivos, lo cual puede generar errores a la hora de realizar actualizaciones en el código.
2. La normalización adecuada previene inconsistencias de datos, evita redundancias y se convierte en una excelente práctica para el correcto uso de una base de datos.
3. Las variables de entorno, con el uso de la librería dotenv, permite un manejo seguro y flexible de credenciales entre entornos.