

Sistema Integrado de Gestión del Archivo de Documentos Técnicos

1. Introducción

El Sistema Integrado de Gestión del Archivo de Documentos Técnicos es una solución diseñada para optimizar la organización, almacenamiento, control de versiones y acceso seguro a documentos técnicos dentro de una organización. El sistema permite centralizar la información, clasificarla por categorías, mantener un historial de versiones, gestionar permisos de acceso y registrar todas las acciones realizadas sobre los documentos. Su finalidad es mejorar la trazabilidad de los archivos técnicos y reducir el tiempo de búsqueda, asegurando que la información esté disponible para el personal autorizado de manera rápida y confiable.

1.2 Objetivo

Objetivo general

Diseñar e implementar una base de datos relacional que soporte un sistema integral para la gestión de archivos de documentos técnicos, garantizando un acceso eficiente, seguro y trazable.

Objetivos específicos

- **Clasificación y organización:** Permitir la categorización de documentos técnicos según su tipo, área o proyecto.
- **Control de versiones:** Mantener un historial completo de versiones y cambios realizados en los documentos.
- **Seguridad y permisos:** Asignar roles y niveles de acceso para proteger la información sensible.
- **Trazabilidad:** Registrar en un historial (log) las acciones de creación, actualización, eliminación y descarga.
- **Soporte a la gestión logística y analítica:** Proveer datos para reportes de uso, ubicación física/virtual y actividad de usuarios.

2. Situación problemática

Actualmente, la gestión de documentos técnicos en muchas organizaciones se realiza de forma desordenada, con archivos dispersos en carpetas locales, discos externos o correos electrónicos. Esta situación provoca:

- **Pérdida de información** por falta de respaldos y control de versiones.
- **Dificultad de acceso** debido a una clasificación deficiente o inexistente.
- **Riesgos de seguridad** por ausencia de control de permisos y registros de actividad.
- **Demoras en procesos** de auditoría, producción o mantenimiento al no encontrar rápidamente la documentación necesaria.

La implementación de esta base de datos soluciona estas brechas al centralizar, estructurar y proteger la información, permitiendo búsquedas rápidas, control de cambios y trazabilidad completa.

3. Modelo de negocio

La solución está diseñada para una empresa del sector industrial que maneja proyectos de ingeniería y mantenimiento.

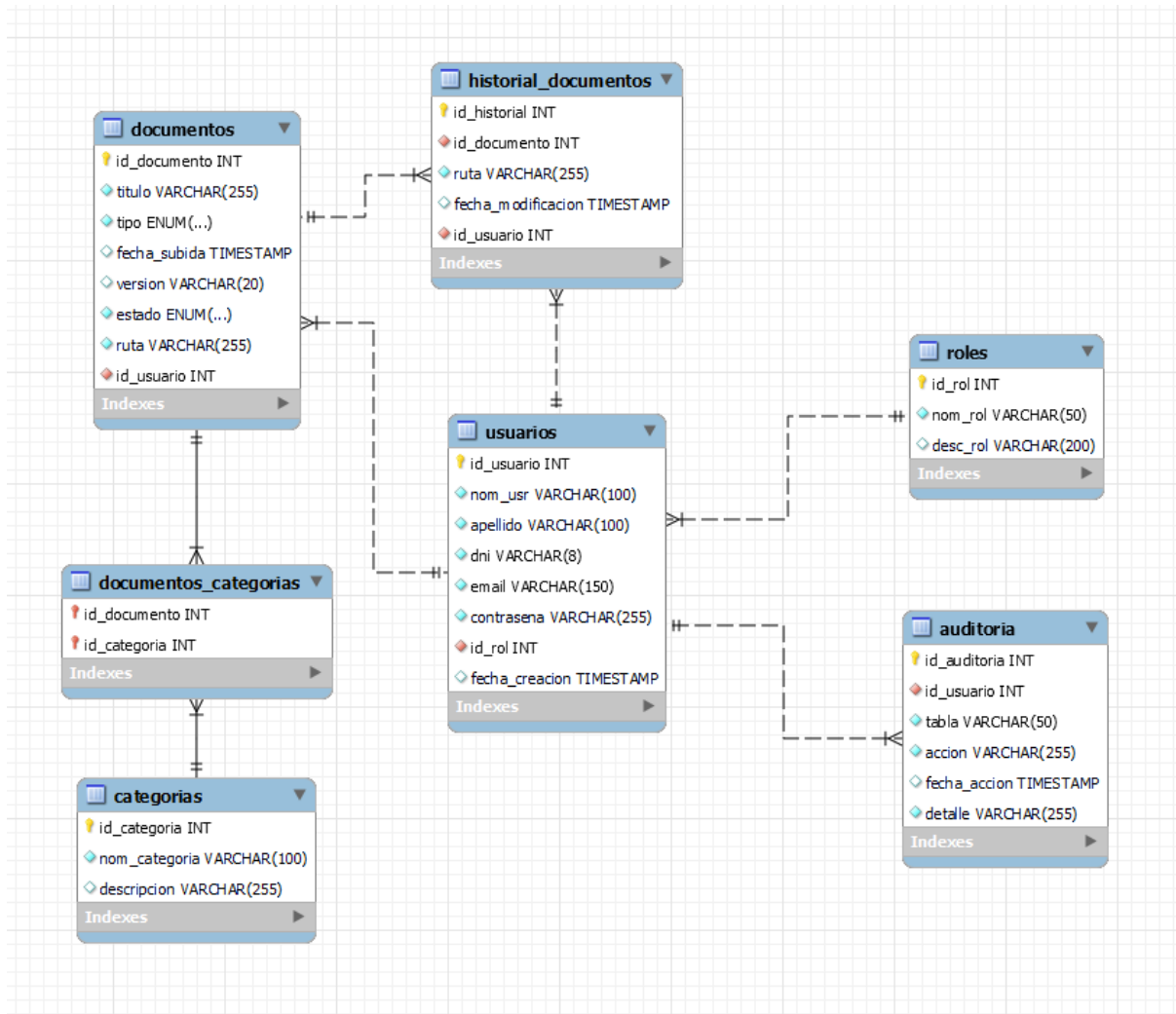
Roles principales:

- **Administrador:** Configura categorías, usuarios, roles y ubicaciones de almacenamiento.
- **Editor:** Carga documentos, gestiona versiones y actualiza metadatos.
- **Lector:** Consulta y descarga documentos autorizados.

Procesos principales:

- Creación de categorías y ubicación de almacenamiento.
- Registro y carga de un documento técnico.
- Generación y control de versiones del documento.
- Etiquetado para facilitar búsquedas temáticas.
- Registro automático en el historial de auditoría.

4. Diagrama Entidad-Relación (E-R)



Relaciones principales:

- Un usuario pertenece a un rol.
- Un documento pertenece a una categoría y tiene múltiples versiones.
- Una versión está almacenada en una ubicación física/virtual.
- Las acciones realizadas se registran en el historial de auditoría.

5. Listado de tablas

- **Tabla: Usuarios**

Abrev.	Nombre completo	Tipo de dato	Tipo de clave	Restricciones	Descripción
id_usr	ID de usuario	INT	PK	NOT NULL, AUTO_INCREMENT	Identificador único
nom_usr	Nombre del usuario	VARCHAR(100)		NOT NULL	Nombre
apellido	Apellido del usuario	VARCHAR(100)		NOT NULL	Apellido
dni	DNI del usuario		UNIQUE	NOT NULL	DNI
email	Correo electrónico	VARCHAR(150)	UNIQUE	NOT NULL	Validado con formato email
rol	Rol del usuario	ENUM('Admin','Editor','Consulta')		NOT NULL	Controla permisos
password	Contraseña	VARCHAR(255)		NOT NULL	Encriptada

- **Tabla: Roles**

Abrev.	Nombre completo	Tipo de dato	Tipo de clave	Restricciones
id_rol	ID de rol	INT	PK	NOT NULL, AUTO_INCREMENT
nom_rol	Nombre del rol	VARCHAR(50)	UNIQUE	NOT NULL
desc_rol	Descripción del rol	VARCHAR(200)		

- **Tabla: Documentos**

Abrev.	Nombre completo	Tipo de dato	Tipo de clave	Restricciones	Descripción
id_doc	ID de documento	INT	PK	NOT NULL, AUTO_INCREMENT	
titulo	Título del documento	VARCHAR(200)		NOT NULL	
tipo	Tipo de documento	ENUM('Plano','Manual','Informe','Contrato')		NOT NULL	
fecha_subida	Fecha de subida	DATETIME		DEFAULT CURRENT_TIMESTAMP	
version	Versión del documento	VARCHAR(20)		DEFAULT '1.0'	
estado	Estado del documento	ENUM('Activo','Archivado','Obsoluto')		NOT NULL	
ruta	Ruta del archivo	VARCHAR(255)		NOT NULL	Ubicación en servidor o nube
id_usr	Usuario que subió el documento	INT	FK	NOT NULL	Referencia: Usuarios.id_usr

- **Tabla: Categorías**

Abrev.	Nombre completo	Tipo de dato	Tipo de clave	Restricciones	Descripción
id_cat	ID de categoría	INT	PK	NOT NULL, AUTO_INCREMENT	
nom_cat	Nombre de la categoría	VARCHAR(100)	UNIQUE	NOT NULL	
desc_cat	Descripción	VARCHAR(255)			

- **Tabla: Documentos Categorías**

Abrev.	Nombre completo	Tipo de dato	Tipo de clave	Restricciones	Descripción
id_doc	ID de documento	INT	PK, FK	NOT NULL	Referencia: Documentos.id_doc
id_cat	ID de categoría	INT	PK, FK	NOT NULL	Referencia: Categorías.id_cat

- Tabla: Historial Documentos

Abrev.	Nombre completo	Tipo de dato	Tipo de clave	Restricciones	Descripción
id_hist	ID de historial	INT	PK	NOT NULL, AUTO_INCREMENT	
id_doc	ID de documento	INT	FK	NOT NULL	
id_usr	ID de usuario	INT	FK	NOT NULL	
accion	Acción realizada	ENUM('Creación','Edición','Eliminación','Descarga')		NOT NULL	
fecha	Fecha de la acción	DATETIME		DEFAULT CURRENT_TIMESTAMP	

- Tabla: Auditoría

Abrev.	Nombre completo	Tipo de dato	Tipo de clave	Restricciones	Descripción
id_aud	ID de auditoría	INT	PK	NOT NULL, AUTO_INCREMENT	
tabla	Nombre de la tabla	VARCHAR(50)		NOT NULL	
accion	Acción	ENUM('INSERT','UPDATE','DELETE')		NOT NULL	
fecha	Fecha de registro	DATETIME		DEFAULT CURRENT_TIMESTAMP	
usuario	Usuario que ejecutó	VARCHAR(100)			

6. Scrip de BD

LINK: https://github.com/lxDiexl/Proyecto_Sql

-- Creación de la base de datos

CREATE DATABASE IF NOT EXISTS SistemaArchivos;

USE SistemaArchivos;

-- Tabla: Roles

CREATE TABLE Roles (

id_rol INT AUTO_INCREMENT PRIMARY KEY,

nom_rol VARCHAR(50) NOT NULL,

desc_rol VARCHAR(200)

);

-- Tabla: Usuarios

CREATE TABLE Usuarios (

id_usuario INT AUTO_INCREMENT PRIMARY KEY NOT NULL,

nom_usr VARCHAR(100) NOT NULL,

apellido VARCHAR(100) NOT NULL,

dni VARCHAR(8) UNIQUE NOT NULL,

email VARCHAR(150) UNIQUE NOT NULL,

contrasena VARCHAR(255) NOT NULL,

id_rol INT NOT NULL,

fecha_creacion TIMESTAMP DEFAULT CURRENT_TIMESTAMP,

FOREIGN KEY (id_rol) REFERENCES Roles(id_rol)

);

-- Tabla: Documentos

CREATE TABLE Documentos (

id_documento INT AUTO_INCREMENT PRIMARY KEY,

titulo VARCHAR(255) NOT NULL,

tipo ENUM('PLANO', 'MANUAL', 'INFORME', 'CONTRATO') NOT NULL,

fecha_subida TIMESTAMP DEFAULT CURRENT_TIMESTAMP,

```
version VARCHAR(20) DEFAULT '1.0',
estado ENUM('Activo', 'Archivado', 'Obsoleto') NOT NULL,
ruta VARCHAR(255) NOT NULL,
id_usuario INT NOT NULL,
FOREIGN KEY (id_usuario) REFERENCES Usuarios(id_usuario)
);
```

-- Tabla: Categorías

```
CREATE TABLE Categorías (
    id_categoria INT AUTO_INCREMENT PRIMARY KEY,
    nom_categoria VARCHAR(100) UNIQUE NOT NULL,
    descripcion VARCHAR(255)
);
```

-- Tabla intermedia: Documentos_Categorías

```
CREATE TABLE Documentos_Categorías (
    id_documento INT NOT NULL,
    id_categoria INT NOT NULL,
    PRIMARY KEY (id_documento, id_categoria),
    FOREIGN KEY (id_documento) REFERENCES Documentos(id_documento),
    FOREIGN KEY (id_categoria) REFERENCES Categorías(id_categoria)
);
```

-- Tabla: Historial_Documentos

```
CREATE TABLE Historial_Documentos (
    id_historial INT AUTO_INCREMENT PRIMARY KEY,
    id_documento INT NOT NULL,
    ruta VARCHAR(255) NOT NULL,
    fecha_modificacion TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    id_usuario INT NOT NULL,
    FOREIGN KEY (id_documento) REFERENCES Documentos(id_documento),
    FOREIGN KEY (id_usuario) REFERENCES Usuarios(id_usuario)
);
```


);

-- Tabla: Auditoría

```
CREATE TABLE Auditoria (  
    id_auditoria INT AUTO_INCREMENT PRIMARY KEY,  
    id_usuario INT NOT NULL,  
    tabla VARCHAR(50) NOT NULL,  
    accion VARCHAR(255) NOT NULL,  
    fecha_accion TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
    detalle VARCHAR(255) NOT NULL,  
    FOREIGN KEY (id_usuario) REFERENCES Usuarios(id_usuario)  
);
```

ENTREGA 2

Listado de objetos (con descripción, objetivo y tablas)

Vistas

1. vw_documentos_detalle

Descripción: Muestra información consolidada de cada documento, incluyendo título, autor, correo del autor, fecha de subida, estado, versión actual, número total de versiones y las categorías a las que está asignado. Integra datos de varias tablas relacionadas.

Objetivo: Facilitar la consulta integral de los documentos con todos sus metadatos principales en una sola vista.

Tablas que manipula:

- Documentos
- Usuarios
- Documentos_Categorias
- Categorias
- Historial_Documentos

2. vw_historial_documentos

Descripción: Muestra la línea de tiempo de las modificaciones realizadas a cada documento, indicando ruta del archivo, fecha de modificación y el usuario responsable.

Objetivo: Brindar un seguimiento de las versiones de cada documento y el responsable de cada actualización.

Tablas que manipula:

- Historial_Documentos
- Documentos
- Usuarios

3. vw_documentos_por_categoria

Descripción: Agrupa los documentos por categorías y devuelve la cantidad de documentos asociados a cada una de ellas.

Objetivo: Ofrecer una métrica rápida que permita visualizar la distribución de documentos según categorías.

Tablas que manipula:

- Categorias
- Documentos_Categorias

4. vw_documentos_sin_categoria

Descripción: Muestra todos los documentos que no están relacionados con ninguna categoría.

Objetivo: Identificar documentos que aún no han sido clasificados para facilitar su organización.

Tablas que manipula:

- Documentos
- Documentos_Categorias

5. vw_auditoria_reciente

Descripción: Muestra los registros más recientes de la tabla de auditoría, incluyendo la fecha, el usuario, la tabla afectada, el tipo de acción y un detalle.

Objetivo: Monitorear en tiempo real las acciones realizadas sobre el sistema, con fines de seguridad y control.

Tablas que manipula:

- Auditoria
- Usuarios

Funciones (MySQL)

1. fn_ultima_version(id_documento INT) RETURNS INT

Descripción: Cuenta la cantidad de registros en el historial de un documento específico, lo que representa el número de versiones existentes.

Objetivo: Obtener de forma rápida la última versión registrada de un documento.

Tablas que manipula:

- Historial_Documentos

2. fn_tiene_categoria(id_documento INT, id_categoria INT) RETURNS TINYINT

Descripción: Verifica si un documento ya está asociado a una categoría específica. Retorna 1 si existe la relación y 0 en caso contrario.

Objetivo: Evitar duplicidades al asignar categorías a documentos.

Tablas que manipula:

- Documentos_Categorias

3. fn_normalizar_titulo(t VARCHAR)

Descripción: Procesa un texto eliminando espacios sobrantes, convirtiéndolo a mayúsculas y estandarizando el formato.

Objetivo: Mejorar las búsquedas de documentos asegurando uniformidad en los títulos.

Tablas que manipula:

- No manipula tablas (función de utilidad de cadena).

Stored Procedures

1. sp_crear_documento(p_titulo, p_ruta, p_id_usuario, OUT p_id_documento)

Descripción: Inserta un nuevo documento en la tabla Documentos, asignando título, ruta, usuario autor y estableciendo la versión inicial en 1. Su inserción es registrada automáticamente en la auditoría por el trigger correspondiente.

Objetivo: Registrar documentos nuevos en el sistema de forma controlada.

Tablas que manipula:

- Documentos
- Auditoria (por trigger)

2. sp_asignar_categoria(p_id_documento, p_id_categoria)

Descripción: Asocia un documento a una categoría determinada, validando primero que la relación no exista para evitar duplicidad. La acción es registrada en la auditoría.

Objetivo: Mantener la relación entre documentos y categorías de manera controlada.

Tablas que manipula:

- Documentos_Categorias
- Auditoria (por trigger)

3. sp_agregar_version(p_id_documento, p_ruta, p_id_usuario, OUT p_version)

Descripción: Crea una nueva entrada en el historial de documentos, representando una nueva versión. Actualiza también la versión en la tabla Documentos y registra la acción en la auditoría.

Objetivo: Controlar el versionado de los documentos garantizando trazabilidad de los cambios.

Tablas que manipula:

- Historial_Documentos
- Documentos
- Auditoria (por trigger)

4. sp_buscar_documentos(p_texto)

Descripción:

Permite buscar documentos aplicando normalización a los títulos y tipos para mejorar los resultados. Retorna coincidencias con información del autor y la cantidad de versiones.

Objetivo: Ofrecer un buscador eficiente de documentos.

Tablas que manipula:

- Documentos
- Usuarios
- Historial_Documentos

- **Triggers**

1. trg_documentos_ai (AFTER INSERT ON Documentos)

Descripción: Se activa automáticamente al insertar un documento y registra la acción en la tabla de auditoría.

Objetivo: Dejar constancia de los documentos nuevos creados en el sistema.

Tablas que manipula:

- Documentos
- Auditoria

2. trg_documentos_au (AFTER UPDATE ON Documentos)

Descripción: Se ejecuta cuando un documento es actualizado, registrando el cambio en la auditoría.

Objetivo: Garantizar trazabilidad de las modificaciones de los documentos.

Tablas que manipula:

- Documentos
- Auditoria

3. trg_historial_ai (AFTER INSERT ON Historial_Documentos)

Descripción: Cada vez que se registra una nueva versión de un documento en el historial, este trigger guarda la acción en la auditoría.

Objetivo: Tener registro detallado del versionado de los documentos.

Tablas que manipula:

- Historial_Documentos
- Auditoria

4. trg_doc_cat_ai (AFTER INSERT ON Documentos_Categorias)

Descripción: Se activa cuando se asigna una categoría a un documento y registra el evento en la auditoría.

Objetivo: Dejar constancia de la clasificación de documentos en categorías.

Tablas que manipula:

- Documentos_Categorias
- Auditoria