



# **MANUAL TÉCNICO**



**CREADO POR:**

**SHARIT DELGADO PINZON - DURLY YURANI SANCHEZ CARRILLO - ANYI SOLAYI GONZALEZ TAPIAS**

# MANUAL TÉCNICO

CREADO POR:

SHARIT DELGADO PINZON  
DURLY YURANI SANCHEZ CARRILLO  
ANYI SOLAYI GONZALEZ TAPIAS

TECNOLOGÍA EN ANÁLISIS Y DESARROLLO DE SOFTWARE – ADSO

FICHA: 2893733

INSTRUCTOR

CARLOS EDUARDO PEREZ RUEDA

SERVICIO NACIONAL DE APRENDIZAJE – SENA

CENTRO DE SERVICIOS EMPRESARIALES Y TURÍSTICO – C.S.E.T.

# BUCARAMANGA – SANTANDER

2025

## Tabla de contenido

Introducción .....	4
Pre-requisitos de instalación del sistema en el servidor .....	5
Estándares .....	5
Controles de auditoría implementados en el sistema. ....	6
Configuración de GestorDocs para uso en producción o local:.....	8
Parámetros a modificar para producción:.....	8
Configuración de Correo Electrónico: .....	8
Modelo de datos (ER) .....	9
Diccionario de datos.....	10
Tabla Solicitudes Registro (solicitudes_registro) .....	10
Tabla Usuarios (usuarios) .....	10
Tabla Carpetas (carpetas) .....	11
Tabla Archivos (archivos) .....	12
Tabla Compartidos (compartidos) .....	12
Tabla Solicitudes Compartidos (solicitudes_compartidos) .....	13
Tabla Notificaciones (notificaciones).....	13
Tabla Tokens Recuperación (tokens_recuperacion) .....	14
Instalación .....	15
<i>Instalación de dependencias</i> .....	15
16.0 Instalar Apache24:.....	15
16.1 Instalar PHP y las extensiones: .....	15
16.2 Instalar Composer: .....	15
16.2 Instalar PostgreSQL y configurarlo: .....	16
Diagrama de despliegue .....	21

## Introducción

El presente manual técnico tiene como objetivo proporcionar una guía detallada para los desarrolladores, ingenieros de sistemas y otros profesionales involucrados en el manejo del software **GestorDocs**. En él, se abordan aspectos clave relacionados con la arquitectura, instalación, configuración y funcionamiento del sistema.

## Pre-requisitos de instalación del sistema en el servidor

**Sistema operativo:** El software es compatible con cualquier sistema operativo, ya sea Linux o Windows.

**Motor de base de datos:** Se utiliza PostgreSQL como motor de base de datos.

**Hardware:** El software fue desarrollado y probado en un servidor HP ProLiant DL350 con las siguientes especificaciones:

- Memoria RAM: 8 GB
- Procesador: Xeon de 2.4

**Navegadores Compatibles:** El software funciona en cualquier navegador web. Durante las pruebas, se verificó su funcionamiento en Google Chrome y Microsoft Edge.

## Estándares

En esta parte, detallamos las normas y prácticas que guían nuestro trabajo. Esto incluye la identificación que asignamos a los archivos y datos utilizados en el desarrollo del software, así como las convenciones que seguimos para garantizar coherencia y calidad.

Para identificar las funciones, siempre las comenzamos con la palabra “fun”.

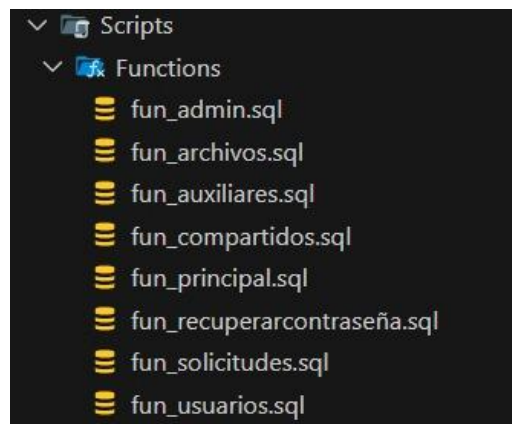
- Los scripts de las funciones se encuentran en la carpeta 'Scripts/Functions/', donde cada archivo contiene las funciones SQL relacionadas con un controlador específico. La nomenclatura sigue el patrón:

`fun_[nombre_controlador].sql`

Donde [nombre\_controlador] corresponde al controlador o modelo al que hace referencia

### Ejemplo:

`fun_admin.sql` - Contiene todas las consultas SQL relacionadas con el controlador y modelo del administrador



Para el manejo de la mayoría de los estilos presentes en la página web, utilizamos un archivo bootstrap.css

Las tablas del sistema siguen una nomenclatura autodescriptiva sin prefijos, donde cada nombre identifica claramente la entidad de datos que almacena, facilitando la comprensión del modelo relacional.

<code>usuarios</code>	- Gestión de usuarios del sistema
<code>solicitudes_registro</code>	- Solicitudes pendientes de aprobación
<code>carpetas</code>	- Estructura jerárquica de directorios
<code>archivos</code>	- Gestión de archivos del sistema
<code>compartidos</code>	- Control de archivos compartidos
<code>solicitudes_compartidos</code>	- Solicitudes para compartir archivos
<code>notificaciones</code>	- Sistema de notificaciones
<code>tokens_recuperacion</code>	- Tokens para recuperación de contraseñas

### Controles de auditoría implementados en el sistema.

Para generar seguimiento ante las creaciones y actualizaciones en la base de datos se utilizó un trigger. El cual se encarga de mostrar que usuario inserto un dato y el día que lo hizo. También, si sucede una actualización me muestra que usuario realizó la actualización y el día que lo hizo.

Utilizamos la siguiente estructura para el funcionamiento del trigger y lo implementamos en las tablas para poder tener el respectivo seguimiento y control de la creación y actualización de los datos.

```
-- Función de auditoria fun_act_tab()
CREATE OR REPLACE FUNCTION fun_act_tab() RETURNS TRIGGER AS
$$
BEGIN
    -- Para operaciones INSERT
    IF TG_OP = 'INSERT' THEN
        NEW.usr_insert = CURRENT_USER;
        NEW.fec_insert = CURRENT_TIMESTAMP;
        RETURN NEW;
    END IF;

    -- Para operaciones UPDATE
    IF TG_OP = 'UPDATE' THEN
        NEW.usr_update = CURRENT_USER;
        NEW.fec_update = CURRENT_TIMESTAMP;
        RETURN NEW;
    END IF;

    RETURN NEW;
END;
$$
LANGUAGE plpgsql;
```

```

-- TRIGGERS DE AUDITORÍA PARA TODAS LAS TABLAS
-- =====

-- Trigger para USUARIOS
CREATE TRIGGER tri_act_usuarios BEFORE INSERT OR UPDATE ON usuarios
FOR EACH ROW EXECUTE PROCEDURE fun_act_tab();

-- Trigger para SOLICITUDES_REGISTRO
CREATE TRIGGER tri_act_solicitudes_registro BEFORE INSERT OR UPDATE ON solicitudes_registro
FOR EACH ROW EXECUTE PROCEDURE fun_act_tab();

-- Trigger para CARPETAS
CREATE TRIGGER tri_act_carpetas BEFORE INSERT OR UPDATE ON carpetas
FOR EACH ROW EXECUTE PROCEDURE fun_act_tab();

-- Trigger para ARCHIVOS
CREATE TRIGGER tri_act_archivos BEFORE INSERT OR UPDATE ON archivos
FOR EACH ROW EXECUTE PROCEDURE fun_act_tab();

-- Trigger para COMPARTIDOS
CREATE TRIGGER tri_act_compartidos BEFORE INSERT OR UPDATE ON compartidos
FOR EACH ROW EXECUTE PROCEDURE fun_act_tab();

-- Trigger para SOLICITUDES_COMPARTIDOS
CREATE TRIGGER tri_act_solicitudes_compartidos BEFORE INSERT OR UPDATE ON solicitudes_compartidos
FOR EACH ROW EXECUTE PROCEDURE fun_act_tab();

-- Trigger para NOTIFICACIONES
CREATE TRIGGER tri_act_notificaciones BEFORE INSERT OR UPDATE ON notificaciones
FOR EACH ROW EXECUTE PROCEDURE fun_act_tab();

-- Trigger para TOKENS_RECUPERACION
CREATE TRIGGER tri_act_tokens_recuperacion BEFORE INSERT OR UPDATE ON tokens_recuperacion
FOR EACH ROW EXECUTE PROCEDURE fun_act_tab();

```

## Configuración de GestorDocs para uso en producción o local:

El archivo Config.php es el corazón de la configuración del sistema GestorDocs. Contiene todas las constantes y parámetros necesarios para el funcionamiento del sistema, incluyendo conexión a base de datos, URLs, APIs externas y configuración de correo electrónico.

```
const BASE_URL = "http://localhost/gestordocs/";
```

### Parámetros a modificar para producción:

**BASE\_URL:** http://localhost/gestordocs/

**En local se usa :** http://localhost/gestordocs/

**En Producción se usa :** https://tudominio.com/gestordocs/

### Configuración de la base de datos en Config:

```
const USER = "Gestordocs_user";
const PASS = "PgSena2024";
const DB = "gestion_archivo";
```

- **HOST:** IP del servidor de base de datos o nombre del host
- **PORT:** Puerto del servidor (usualmente 5432)
- **USER:** Usuario específico para producción (no usar 'postgres')
- **PASS:** Contraseña segura y compleja
- **DB:** Nombre de la base de datos en producción

## Configuración de Correo Electrónico:

```
*****
Configuracion del envio de correo
*****/

// Configuración SMTP
define(constant_name: 'SMTP_HOST', value: 'smtp.gmail.com');
define(constant_name: 'SMTP_PORT', value: 587);
define(constant_name: 'SMTP_SECURE', value: 'tls');
define(constant_name: 'SMTP_AUTH', value: true);
define(constant_name: 'SMTP_USERNAME', value: 'solayitapias1@gmail.com');
define(constant_name: 'SMTP_PASSWORD', value: 'uwadfayhtzzakivm');

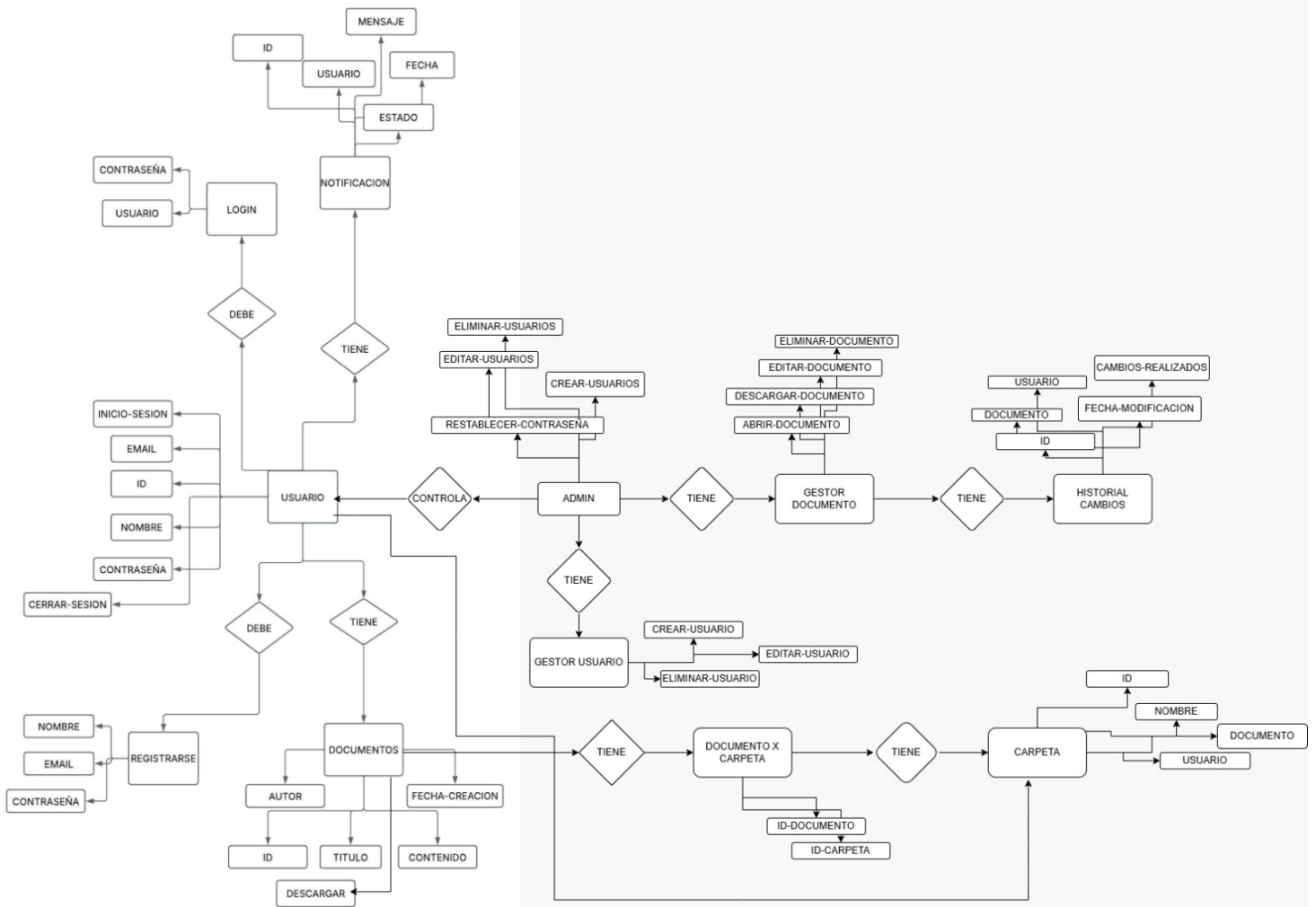
// Información del remitente
define(constant_name: 'MAIL_FROM_EMAIL', value: 'solayitapias1@gmail.com');
define(constant_name: 'MAIL_FROM_NAME', value: 'Sistema de Gestión de Archivos');

// Configuración general
define(constant_name: 'MAIL_CHARSET', value: 'UTF-8');
define(constant_name: 'MAIL_IS_HTML', value: true);
?>
```

Es la configuración de SMTP depende de que servicio de correo utiliza en este caso GestorDocs utiliza una cuenta de Google.



## Modelo de datos (ER)



## Diccionario de datos

**Tabla Solicitudes Registro (solicitudes\_registro)**

Campo	Tipo de Dato	Longitud	Llave	Nulo	Descripción
id	SERIAL	-	PK	No	Identificador único de solicitud
nombre	VARCHAR	100	-	No	Nombre del solicitante
apellido	VARCHAR	100	-	No	Apellido del solicitante
correo	VARCHAR	100	-	No	Correo electrónico
telefono	VARCHAR	15	-	No	Número telefónico
direccion	VARCHAR	255	-	No	Dirección física
clave	VARCHAR	200	-	No	Contraseña encriptada
fecha_solicitud	TIMESTAMP WITH TIME ZONE	-	-	No	Fecha de la solicitud
estado	INTEGER	-	-	No	0=Pendiente, 1=Aprobada, 2=Rechazada
id_usuario_admin	INTEGER	-	FK	Sí	Admin que procesó la solicitud
fecha_procesado	TIMESTAMP WITH TIME ZONE	-	-	Sí	Fecha de procesamiento
usr_insert	VARCHAR	100	-	Sí	Usuario que insertó el registro
fec_insert	TIMESTAMP WITH TIME ZONE	-	-	Sí	Fecha de inserción
usr_update	VARCHAR	100	-	Sí	Usuario que actualizó el registro
fec_update	TIMESTAMP WITH TIME ZONE	-	-	Sí	Fecha de última actualización

**Tabla Usuarios (usuarios)**

Campo	Tipo de Dato	Longitud	Llave	Nulo	Descripción
id	SERIAL	-	PK	No	Identificador único del usuario
nombre	VARCHAR	100	-	No	Nombre del usuario
apellido	VARCHAR	100	-	No	Apellido del usuario
correo	VARCHAR	100	UK	No	Correo electrónico único
telefono	VARCHAR	15	-	No	Número telefónico
direccion	VARCHAR	255	-	No	Dirección física
clave	VARCHAR	200	-	No	Contraseña encriptada

fecha	TIMESTAMP WITH TIME ZONE	-	-	No	Fecha de registro
estado	INTEGER	-	-	No	Estado activo (1) /inactivo (0)
rol	INTEGER	-	-	No	Rol: 0=Usuario, 1=Admin, 2=SuperAdmin
avatar	VARCHAR	255	-	Sí	Ruta de imagen de avatar
fecha_ultimo_cambio_clave	TIMESTAMP WITH TIME ZONE	-	-	Sí	Última actualización de contraseña
usr_insert	VARCHAR	100	-	Sí	Usuario que insertó el registro
fec_insert	TIMESTAMP WITH TIME ZONE	-	-	Sí	Fecha de inserción
usr_update	VARCHAR	100	-	Sí	Usuario que actualizó el registro
fec_update	TIMESTAMP WITH TIME ZONE	-	-	Sí	Fecha de última actualización

**Tabla Carpetas (carpetas)**

Campo	Tipo de Dato	Longitud	Llave	Nulo	Descripción
id	SERIAL	-	PK	No	Identificador único de carpeta
nombre	VARCHAR	255	-	No	Nombre de la carpeta
fecha_create	TIMESTAMP WITH TIME ZONE	-	-	No	Fecha de creación
estado	INTEGER	-	-	No	Estado activo (1) /eliminado (0)
elimina	TIMESTAMP WITH TIME ZONE	-	-	Sí	Fecha de eliminación lógica
id_usuario	INTEGER	-	FK	No	Propietario de la carpeta
id_carpeta_padre	INTEGER	-	FK	Sí	Carpeta padre (estructura jerárquica)
usr_insert	VARCHAR	100	-	Sí	Usuario que insertó el registro
fec_insert	TIMESTAMP WITH TIME ZONE	-	-	Sí	Fecha de inserción
usr_update	VARCHAR	100	-	Sí	Usuario que actualizó el registro
fec_update	TIMESTAMP WITH TIME ZONE	-	-	Sí	Fecha de última actualización

**Tabla Archivos (archivos)**

<b>Campo</b>	<b>Tipo de Dato</b>	<b>Longitud</b>	<b>Llave</b>	<b>Nulo</b>	<b>Descripción</b>
id	SERIAL	-	PK	No	Identificador único del archivo
nombre	VARCHAR	255	-	No	Nombre del archivo
tipo	VARCHAR	100	-	No	Tipo/extensión del archivo
fecha_create	TIMESTAMP WITH TIME ZONE	-	-	No	Fecha de creación
estado	INTEGER	-	-	No	Estado activo (1) /eliminado (0)
elimina	TIMESTAMP WITH TIME ZONE	-	-	Sí	Fecha de eliminación lógica
id_carpeta	INTEGER	-	FK	Sí	Carpeta contenedora
id_usuario	INTEGER	-	FK	No	Propietario del archivo
tamano	BIGINT	-	-	Sí	Tamaño en bytes
usr_insert	VARCHAR	100	-	Sí	Usuario que insertó el registro
fec_insert	TIMESTAMP WITH TIME ZONE	-	-	Sí	Fecha de inserción
usr_update	VARCHAR	100	-	Sí	Usuario que actualizó el registro
fec_update	TIMESTAMP WITH TIME ZONE	-	-	Sí	Fecha de última actualización

**Tabla Compartidos (compartidos)**

<b>Campo</b>	<b>Tipo de Dato</b>	<b>Longitud</b>	<b>Llave</b>	<b>Nulo</b>	<b>Descripción</b>
id	SERIAL	-	PK	No	Identificador único del compartido
id_archivo_original	INTEGER	-	FK	No	Archivo original compartido
id_usuario_propietario	INTEGER	-	FK	No	Usuario propietario del archivo
id_usuario_receptor	INTEGER	-	FK	No	Usuario receptor del archivo
fecha_aceptado	TIMESTAMP WITH TIME ZONE	-	-	No	Fecha de aceptación
nombre_personalizado	VARCHAR	255	-	Sí	Nombre personalizado del archivo
estado	INTEGER	-	-	No	Estado activo (1) /eliminado (0)

fecha_eliminado	TIMESTAMP WITH TIME ZONE	-	-	Sí	Fecha de eliminación
usr_insert	VARCHAR	100	-	Sí	Usuario que insertó el registro
fec_insert	TIMESTAMP WITH TIME ZONE	-	-	Sí	Fecha de inserción
usr_update	VARCHAR	100	-	Sí	Usuario que actualizó el registro
fec_update	TIMESTAMP WITH TIME ZONE	-	-	Sí	Fecha de última actualización

**Tabla Solicitudes Compartidos (solicitudes\_compartidos)**

Campo	Tipo de Dato	Longitud	Llave	Nulo	Descripción
id	SERIAL	-	PK	No	Identificador único de solicitud
fecha_add	TIMESTAMP WITH TIME ZONE	-	-	No	Fecha de creación de solicitud
correo	VARCHAR	100	-	No	Correo del usuario a compartir
estado	INTEGER	-	-	No	Estado activo (1) /eliminado (0)
elimina	TIMESTAMP WITH TIME ZONE	-	-	Sí	Fecha de eliminación
id_archivo	INTEGER	-	FK	No	Archivo a compartir
id_usuario	INTEGER	-	FK	No	Usuario que envía la solicitud
aceptado	SMALLINT	-	-	Sí	0=Pendiente, 1=Aceptado, 2=Rechazado
usr_insert	VARCHAR	100	-	Sí	Usuario que insertó el registro
fec_insert	TIMESTAMP WITH TIME ZONE	-	-	Sí	Fecha de inserción
usr_update	VARCHAR	100	-	Sí	Usuario que actualizó el registro
fec_update	TIMESTAMP WITH TIME ZONE	-	-	Sí	Fecha de última actualización

**Tabla Notificaciones (notificaciones)**

Campo	Tipo de Dato	Longitud	Llave	Nulo	Descripción
id	SERIAL	-	PK	No	Identificador único de notificación

id_usuario	INTEGER	-	FK	Sí	Usuario receptor de la notificación
id_carpeta	INTEGER	-	FK	Sí	Carpeta relacionada
id_solicitud	INTEGER	-	FK	Sí	Solicitud relacionada
nombre	VARCHAR	255	-	Sí	Nombre descriptivo
evento	VARCHAR	50	-	Sí	Tipo de evento
fecha	TIMESTAMP WITH TIME ZONE	-	-	No	Fecha de la notificación
leida	SMALLINT	-	-	Sí	0=No leída, 1=Leída
usr_insert	VARCHAR	100	-	Sí	Usuario que insertó el registro
fec_insert	TIMESTAMP WITH TIME ZONE	-	-	Sí	Fecha de inserción
usr_update	VARCHAR	100	-	Sí	Usuario que actualizó el registro
fec_update	TIMESTAMP WITH TIME ZONE	-	-	Sí	Fecha de última actualización

**Tabla Tokens Recuperación (tokens\_recuperacion)**

<b>Campo</b>	<b>Tipo de Dato</b>	<b>Longitud</b>	<b>Llave</b>	<b>Nulo</b>	<b>Descripción</b>
id	SERIAL	-	PK	No	Identificador único del token
id_usuario	INTEGER	-	FK	No	Usuario solicitante
token	VARCHAR	100	UK	No	Token único de recuperación
fecha_creacion	TIMESTAMP WITH TIME ZONE	-	-	No	Fecha de creación del token
fecha_expiracion	TIMESTAMP WITH TIME ZONE	-	-	No	Fecha de expiración
usado	SMALLINT	-	-	Sí	0=No usado, 1=Usado
ip_solicitud	INET	-	-	Sí	IP desde donde se solicitó
user_agent	TEXT	-	-	Sí	Navegador/aplicación usada
usr_insert	VARCHAR	100	-	Sí	Usuario que insertó el registro
fec_insert	TIMESTAMP WITH TIME ZONE	-	-	Sí	Fecha de inserción
usr_update	VARCHAR	100	-	Sí	Usuario que actualizó el registro
fec_update	TIMESTAMP WITH TIME ZONE	-	-	Sí	Fecha de última actualización

## Instalación

### *Instalación de dependencias*

#### 16.0 Instalar Apache24:

Para la configuración inicial del entorno se necesita un servidor sea apache o de su preferencia y se necesita tener habilitado los siguientes módulos:

```
• LoadModule rewrite_module modules/mod_rewrite.so
```

Y tener habilitado el **AllowOverride** de la siguiente manera:

```
• AllowOverride All
• Require all granted
```

#### 16.1 Instalar PHP y las extensiones:

Se recomienda instalar PHP como mínimo la versión 7.1 para la compatibilidad, así mismo se necesita tener habilitadas las siguientes extensiones:

```
• extension=mbstring
• extension=curl
• extension=odbc
• extension=openssl
• extension=pdo_odbc
• extension=pdo_pgsql
• extension=pgsql
```

Y aparte necesitaremos las siguientes:

```
• extension=soap
• extension=sockets
• extension=zip
```

#### 16.2 Instalar Composer:

Composer es necesario para gestionar las dependencias de PHP. Instálelo globalmente desde la página oficial del Composer.

[Composer download](#)

Esto descargará e instalará PHPMailer, Firebase JWT y cualquier otra dependencia definida en el archivo composer.json del proyecto.

## 16.2 Instalar PostgreSQL y configurarlo:

Se recomienda instalar como mínimo el **PostgreSQL** en versión 16, crear un usuario y una base de datos con el nombre `gestion_archivo` o el de tu preferencia y cambiar esas credenciales en el archivo `Config.php` ubicado en el `Config/Config.php`.

El script de la base de datos de instalación es el siguiente:

```
Scripts > scriptdb.sql
1  -- =====
2  -- SISTEMA DE GESTIÓN DE ARCHIVOS Y USUARIOS
3  -- Versión: 2.0
4  -- Base de Datos: PostgreSQL
5  -- Descripción: Script completo para crear la estructura de BD
6  -- =====
7
8  -- Configurar el esquema y verificar conexión
9  DO $$
10 BEGIN
11     RAISE NOTICE 'Iniciando creación de base de datos del sistema de archivos...';
12     RAISE NOTICE 'PostgreSQL versión: %', version();
13 END $$;
14
15 -- =====
16 -- ELIMINACIÓN DE OBJETOS EXISTENTES (si existen)
17 -- =====
18
19 -- Eliminar triggers de auditoria
20 DROP TRIGGER IF EXISTS tri_act_usuarios ON usuarios;
21 DROP TRIGGER IF EXISTS tri_act_solicitudes_registro ON solicitudes_registro;
22 DROP TRIGGER IF EXISTS tri_act_carpetas ON carpetas;
23 DROP TRIGGER IF EXISTS tri_act_archivos ON archivos;
24 DROP TRIGGER IF EXISTS tri_act_compartidos ON compartidos;
25 DROP TRIGGER IF EXISTS tri_act_solicitudes_compartidos ON solicitudes_compartidos;
26 DROP TRIGGER IF EXISTS tri_act_notificaciones ON notificaciones;
27 DROP TRIGGER IF EXISTS tri_act_tokens_recuperacion ON tokens_recuperacion;
28
29 -- Eliminar triggers
30 DROP TRIGGER IF EXISTS trigger_nueva_solicitud ON solicitudes_registro;
31 DROP TRIGGER IF EXISTS update_usuarios_fecha ON usuarios;
32 DROP TRIGGER IF EXISTS update_carpetas_modtime ON carpetas;
33
34 -- Eliminar funciones
35 DROP FUNCTION IF EXISTS fun_act_tab();
36 DROP FUNCTION IF EXISTS notificar_solicitud_registro();
37 DROP FUNCTION IF EXISTS update_usuario_fecha();
38
39
40 -- Eliminar tablas en orden correcto (dependencias)
41 DROP TABLE IF EXISTS notificaciones CASCADE;
42 DROP TABLE IF EXISTS solicitudes_compartidos CASCADE;
43 DROP TABLE IF EXISTS compartidos CASCADE;
44 DROP TABLE IF EXISTS archivos CASCADE;
45 DROP TABLE IF EXISTS carpetas CASCADE;
46 DROP TABLE IF EXISTS solicitudes_registro CASCADE;
47 DROP TABLE IF EXISTS tokens_recuperacion CASCADE;
48 DROP TABLE IF EXISTS usuarios CASCADE;
49
50 -- =====
51 -- CREACIÓN DE TABLAS CON CAMPOS DE AUDITORÍA INTEGRADOS
52 -- =====
53
54 -- TABLA: usuarios
55 -- Gestiona los usuarios del sistema con roles y autenticación
56 CREATE TABLE usuarios (
57     id SERIAL PRIMARY KEY,
58     nombre VARCHAR(100) NOT NULL,
59     apellido VARCHAR(100) NOT NULL,
60     correo VARCHAR(100) NOT NULL UNIQUE,
61     telefono VARCHAR(15) NOT NULL,
62     direccion VARCHAR(255) NOT NULL,
63     clave VARCHAR(200) NOT NULL,
64     fecha TIMESTAMP WITH TIME ZONE NOT NULL DEFAULT CURRENT_TIMESTAMP,
65     estado INTEGER NOT NULL DEFAULT 1,
66     rol INTEGER NOT NULL,
67     avatar VARCHAR(255) DEFAULT 'Assets/images/avatar.jpg',
68     fecha_ultimo_cambio_clave TIMESTAMP WITH TIME ZONE,
69     -- Campos de auditoria
70     usr_insert VARCHAR(100) DEFAULT CURRENT_USER,
71     fec_insert TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP,
72     usr_update VARCHAR(100),
73     fec_update TIMESTAMP WITH TIME ZONE,
```



```

Scripts > scriptdb.sql
56 CREATE TABLE usuarios (
75     -- Constraints adicionales
76     CONSTRAINT chk_usuarios_estado CHECK (estado IN (0, 1)),
77     CONSTRAINT chk_usuarios_rol CHECK (rol IN (0, 1, 2)) -- 0: Usuario, 1: Admin, 2: Super Admin
78 );
79
80 -- TABLA: solicitudes_registro
81 -- Gestiona las solicitudes de registro pendientes de aprobación
82 CREATE TABLE solicitudes_registro (
83     id SERIAL PRIMARY KEY,
84     nombre VARCHAR(100) NOT NULL,
85     apellido VARCHAR(100) NOT NULL,
86     correo VARCHAR(100) NOT NULL,
87     telefono VARCHAR(15) NOT NULL,
88     direccion VARCHAR(255) NOT NULL,
89     clave VARCHAR(200) NOT NULL,
90     fecha_solicitud TIMESTAMP WITH TIME ZONE NOT NULL DEFAULT CURRENT_TIMESTAMP,
91     estado INTEGER NOT NULL DEFAULT 0,
92     id_usuario_admin INTEGER,
93     fecha_procesado TIMESTAMP WITH TIME ZONE,
94     -- Campos de auditoria
95     usr_insert VARCHAR(100) DEFAULT CURRENT_USER,
96     fec_insert TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP,
97     usr_update VARCHAR(100),
98     fec_update TIMESTAMP WITH TIME ZONE,
99
100     -- Constraints
101     CONSTRAINT chk_solicitudes_estado CHECK (estado IN (0, 1, 2)), -- 0: Pendiente, 1: Aprobada, 2: Rechazada
102     CONSTRAINT fk_solicitudes_admin
103         FOREIGN KEY (id_usuario_admin)
104         REFERENCES usuarios(id)
105         ON DELETE SET NULL ON UPDATE CASCADE
106 );
107
108 -- TABLA: carpetas
109 -- Estructura jerárquica de carpetas del sistema

```

```

Scripts > scriptdb.sql
110 CREATE TABLE carpetas (
112     nombre VARCHAR(255) NOT NULL,
113     fecha_create TIMESTAMP WITH TIME ZONE NOT NULL DEFAULT CURRENT_TIMESTAMP,
114     estado INTEGER NOT NULL DEFAULT 1,
115     elimina TIMESTAMP WITH TIME ZONE,
116     id_usuario INTEGER NOT NULL,
117     id_carpetas_padre INTEGER,
118     -- Campos de auditoria
119     usr_insert VARCHAR(100) DEFAULT CURRENT_USER,
120     fec_insert TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP,
121     usr_update VARCHAR(100),
122     fec_update TIMESTAMP WITH TIME ZONE,
123
124     -- Constraints
125     CONSTRAINT chk_carpetas_estado CHECK (estado IN (0, 1)),
126     CONSTRAINT fk_carpetas_usuario
127         FOREIGN KEY (id_usuario)
128         REFERENCES usuarios(id)
129         ON DELETE CASCADE ON UPDATE CASCADE,
130     CONSTRAINT fk_carpetas_padre
131         FOREIGN KEY (id_carpetas_padre)
132         REFERENCES carpetas(id)
133         ON DELETE CASCADE ON UPDATE CASCADE
134 );
135
136 -- TABLA: archivos
137 -- Gestiona los archivos del sistema
138 CREATE TABLE archivos (
139     id SERIAL PRIMARY KEY,
140     nombre VARCHAR(255) NOT NULL,
141     tipo VARCHAR(100) NOT NULL,
142     fecha_create TIMESTAMP WITH TIME ZONE NOT NULL DEFAULT CURRENT_TIMESTAMP,
143     estado INTEGER NOT NULL DEFAULT 1,
144     elimina TIMESTAMP WITH TIME ZONE,
145     id_carpetas INTEGER,
146     id_usuario INTEGER NOT NULL,
147     tamaño BIGINT

```

```

Scripts > scriptdb.sql
169 CREATE TABLE compartidos (
170     id SERIAL PRIMARY KEY,
171     id_archivo_original INTEGER NOT NULL,
172     id_usuario_propietario INTEGER NOT NULL,
173     id_usuario_receptor INTEGER NOT NULL,
174     fecha_aceptado TIMESTAMP WITH TIME ZONE NOT NULL DEFAULT CURRENT_TIMESTAMP,
175     nombre_personalizado VARCHAR(255),
176     estado INTEGER NOT NULL DEFAULT 1,
177     fecha_eliminado TIMESTAMP WITH TIME ZONE,
178     -- Campos de auditoria
179     usr_insert VARCHAR(100) DEFAULT CURRENT_USER,
180     fec_insert TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP,
181     usr_update VARCHAR(100),
182     fec_update TIMESTAMP WITH TIME ZONE,
183
184     -- Constraints
185     CONSTRAINT chk_compartidos_estado CHECK (estado IN (0, 1)),
186     CONSTRAINT fk_archivo_compartido_original
187         FOREIGN KEY (id_archivo_original)
188         REFERENCES archivos(id)
189         ON DELETE CASCADE ON UPDATE CASCADE,
190     CONSTRAINT fk_usuario_propietario_compartido
191         FOREIGN KEY (id_usuario_propietario)
192         REFERENCES usuarios(id)
193         ON DELETE CASCADE ON UPDATE CASCADE,
194     CONSTRAINT fk_usuario_receptor_compartido
195         FOREIGN KEY (id_usuario_receptor)
196         REFERENCES usuarios(id)
197         ON DELETE CASCADE ON UPDATE CASCADE,
198     -- Evitar que un usuario se comparta archivos a sí mismo
199     CONSTRAINT chk_diferentes_usuarios CHECK (id_usuario_propietario != id_usuario_receptor)
200 );
201
202 -- TABLA: solicitudes_compartidos (anteriormente detalle_archivos)
203 -- Gestiona las solicitudes de compartir archivos
204 CREATE TABLE solicitudes_compartidos (
205     id SERIAL PRIMARY KEY,

```

```

Scripts > scriptdb.sql
203 -- Gestiona las solicitudes de compartir archivos
204 CREATE TABLE solicitudes_compartidos (
205     id SERIAL PRIMARY KEY,
206     fecha_add TIMESTAMP WITH TIME ZONE NOT NULL DEFAULT CURRENT_TIMESTAMP,
207     correo VARCHAR(100) NOT NULL,
208     estado INTEGER NOT NULL DEFAULT 1,
209     elimina TIMESTAMP WITH TIME ZONE,
210     id_archivo INTEGER NOT NULL,
211     id_usuario INTEGER NOT NULL,
212     aceptado SMALLINT DEFAULT 0,
213     -- Campos de auditoria
214     usr_insert VARCHAR(100) DEFAULT CURRENT_USER,
215     fec_insert TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP,
216     usr_update VARCHAR(100),
217     fec_update TIMESTAMP WITH TIME ZONE,
218
219     -- Constraints
220     CONSTRAINT chk_solicitudes_estado CHECK (estado IN (0, 1)),
221     CONSTRAINT chk_solicitudes_aceptado CHECK (aceptado IN (0, 1, 2)), -- 0: Pendiente, 1: Aceptado, 2: Rechazado
222     CONSTRAINT fk_solicitudes_compartidos_archivo
223         FOREIGN KEY (id_archivo)
224         REFERENCES archivos(id)
225         ON DELETE CASCADE ON UPDATE CASCADE,
226     CONSTRAINT fk_solicitudes_compartidos_usuario
227         FOREIGN KEY (id_usuario)
228         REFERENCES usuarios(id)
229         ON DELETE CASCADE ON UPDATE CASCADE
230 );
231
232 -- TABLA: notificaciones
233 -- Sistema de notificaciones del aplicativo
234 CREATE TABLE notificaciones (
235     id SERIAL PRIMARY KEY,
236     id_usuario INTEGER,
237     id_carpeta INTEGER,
238     id_solicitud INTEGER,
239     fecha VARCHAR(255)

```

```

Scripts > scriptdb.sql
264
265 -- TABLA: tokens_recuperacion
266 CREATE TABLE tokens_recuperacion (
267     id SERIAL PRIMARY KEY,
268     id_usuario INTEGER NOT NULL,
269     token VARCHAR(100) NOT NULL UNIQUE,
270     fecha_creacion TIMESTAMP WITH TIME ZONE NOT NULL DEFAULT CURRENT_TIMESTAMP,
271     fecha_expiration TIMESTAMP WITH TIME ZONE NOT NULL,
272     usado SMALLINT DEFAULT 0 CHECK (usado IN (0, 1)), -- 0: No usado, 1: Usado
273     ip_solicitud INET,
274     user_agent TEXT,
275     -- Campos de auditoria
276     usr_insert VARCHAR(100) DEFAULT CURRENT_USER,
277     fec_insert TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP,
278     usr_update VARCHAR(100),
279     fec_update TIMESTAMP WITH TIME ZONE,
280
281     -- Restricción de clave foránea
282     CONSTRAINT fk_token_recuperacion_usuario
283         FOREIGN KEY (id_usuario)
284         REFERENCES usuarios(id)
285         ON DELETE CASCADE ON UPDATE CASCADE
286 );
287
288 -- =====
289 -- CREACIÓN DE ÍNDICES PARA OPTIMIZACIÓN
290 -- =====
291
292 -- Índices para usuarios
293 CREATE INDEX idx_usuarios_correo ON usuarios(correo);
294 CREATE INDEX idx_usuarios_estado ON usuarios(estado);
295 CREATE INDEX idx_usuarios_rol ON usuarios(rol);
296 CREATE INDEX idx_usuarios_fecha_cambio_clave ON usuarios(fecha_ultimo_cambio_clave);
297
298 -- Índices para solicitudes_registro
299 CREATE INDEX idx_solicitudes_correo ON solicitudes_registro(correo);

```

```

300 -- =====
301
302 -- Índices para usuarios
303 CREATE INDEX idx_usuarios_correo ON usuarios(correo);
304 CREATE INDEX idx_usuarios_estado ON usuarios(estado);
305 CREATE INDEX idx_usuarios_rol ON usuarios(rol);
306 CREATE INDEX idx_usuarios_fecha_cambio_clave ON usuarios(fecha_ultimo_cambio_clave);
307
308 -- Índices para solicitudes_registro
309 CREATE INDEX idx_solicitudes_correo ON solicitudes_registro(correo);
310 CREATE INDEX idx_solicitudes_estado ON solicitudes_registro(estado);
311 CREATE INDEX idx_solicitudes_fecha ON solicitudes_registro(fecha_solicitud);
312
313 -- Índices para carpetas
314 CREATE INDEX idx_carpetas_usuario ON carpetas(id_usuario);
315 CREATE INDEX idx_carpetas_padre ON carpetas(id_carpetas_padre);
316 CREATE INDEX idx_carpetas_estado ON carpetas(estado);
317 CREATE INDEX idx_carpetas_fecha_create ON carpetas(fecha_create);
318
319 -- Índices para archivos
320 CREATE INDEX idx_archivos_carpetas ON archivos(id_carpetas);
321 CREATE INDEX idx_archivos_usuario ON archivos(id_usuario);
322 CREATE INDEX idx_archivos_estado ON archivos(estado);
323 CREATE INDEX idx_archivos_tipo ON archivos(tipo);
324 CREATE INDEX idx_archivos_fecha_create ON archivos(fecha_create);
325
326 -- Índices para compartidos
327 CREATE INDEX idx_archivo_original ON compartidos(id_archivo_original);
328 CREATE INDEX idx_usuario_propietario ON compartidos(id_usuario_propietario);
329 CREATE INDEX idx_usuario_receptor ON compartidos(id_usuario_receptor);
330 CREATE INDEX idx_compartidos_estado ON compartidos(estado);
331
332 -- Índices para solicitudes_compartidos
333 CREATE INDEX idx_solicitudes_compartidos_archivo ON solicitudes_compartidos(id_archivo);
334 CREATE INDEX idx_solicitudes_compartidos_usuario ON solicitudes_compartidos(id_usuario);
335 CREATE INDEX idx_solicitudes_compartidos_aceptado ON solicitudes_compartidos(aceptado);

```

Junto con el script de instalación ejecutar las siguientes funciones:

```
-- =====
-- FUNCIONES DEL SISTEMA
-- =====

-- Función para actualizar timestamp automáticamente en carpetas
CREATE OR REPLACE FUNCTION update_modified_column()
RETURNS TRIGGER AS $$
BEGIN
    NEW.fecha_create = CURRENT_TIMESTAMP;
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;

-- Función para actualizar fecha en usuarios
CREATE OR REPLACE FUNCTION update_usuario_fecha()
RETURNS TRIGGER AS $$
BEGIN
    NEW.fecha = CURRENT_TIMESTAMP;
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;

-- Función para notificar a administradores sobre nuevas solicitudes
CREATE OR REPLACE FUNCTION notificar_solicitud_registro()
RETURNS TRIGGER AS $$
BEGIN
    -- Notificar a todos los administradores (rol = 1) sobre la nueva solicitud
    INSERT INTO notificaciones (id_usuario, id_solicitud, nombre, evento, fecha, leida)
    SELECT id, NEW.id, NEW.nombre || ' ' || NEW.apellido, 'NUEVA_SOLICITUD', CURRENT_TIMESTAMP, 0
    FROM usuarios WHERE rol = 1;

    RETURN NEW;
END;
$$ LANGUAGE plpgsql;
```

Y los siguientes triggers:

```
416 CREATE TRIGGER trigger_nueva_solicitud
422
423 -- =====
424 -- TRIGGERS DE AUDITORIA PARA TODAS LAS TABLAS
425 -- =====
426
427 -- Trigger para USUARIOS
428 CREATE TRIGGER tri_act_usuarios BEFORE INSERT OR UPDATE ON usuarios
429 FOR EACH ROW EXECUTE PROCEDURE fun_act_tab();
430
431 -- Trigger para SOLICITUDES_REGISTRO
432 CREATE TRIGGER tri_act_solicitudes_registro BEFORE INSERT OR UPDATE ON solicitudes_registro
433 FOR EACH ROW EXECUTE PROCEDURE fun_act_tab();
434
435 -- Trigger para CARPETAS
436 CREATE TRIGGER tri_act_carpetas BEFORE INSERT OR UPDATE ON carpetas
437 FOR EACH ROW EXECUTE PROCEDURE fun_act_tab();
438
439 -- Trigger para ARCHIVOS
440 CREATE TRIGGER tri_act_archivos BEFORE INSERT OR UPDATE ON archivos
441 FOR EACH ROW EXECUTE PROCEDURE fun_act_tab();
442
443 -- Trigger para COMPARTIDOS
444 CREATE TRIGGER tri_act_compartidos BEFORE INSERT OR UPDATE ON compartidos
445 FOR EACH ROW EXECUTE PROCEDURE fun_act_tab();
446
447 -- Trigger para SOLICITUDES_COMPARTIDOS
448 CREATE TRIGGER tri_act_solicitudes_compartidos BEFORE INSERT OR UPDATE ON solicitudes_compartidos
449 FOR EACH ROW EXECUTE PROCEDURE fun_act_tab();
450
451 -- Trigger para NOTIFICACIONES
452 CREATE TRIGGER tri_act_notificaciones BEFORE INSERT OR UPDATE ON notificaciones
453 FOR EACH ROW EXECUTE PROCEDURE fun_act_tab();
454
455 -- Trigger para TOKENS_RECUPERACION
456 CREATE TRIGGER tri_act_tokens_recuperacion BEFORE INSERT OR UPDATE ON tokens_recuperacion
457 FOR EACH ROW EXECUTE PROCEDURE fun_act_tab();
```

## Diagrama de despliegue

