

# **Diseño de un software de correspondencia para la alcaldía municipal de Gachetá**

Yeferson Camilo Zaque Bautista

Asesor

Edgar Alonso Bojacá

Universidad Nacional Abierta y a Distancia UNAD

Escuela de Ciencias Básicas, Tecnología e Ingeniería ECBTI

Ingeniería de Sistemas

2023

**Nota de aceptación:**

---

---

---

---

---

---

Firma del Jurado 1

---

Firma del Jurado 2

### **Dedicatoria**

A Dios todo poderoso por bendecirme con mucha salud y sabiduría para llevar a cabo este sueño de formación profesional. A mi madre por acompañarme desde el inicio con sus palabras y su sabiduría que me motivaba guiándome por el buen camino, a mi padre por su ejemplo de perseverancia para salir adelante y no rendirme, a mi familia por sus palabras de motivación y a los docentes de la Universidad Nacional Abierta y a Distancia UNAD que hicieron parte de mi proceso formativo por su guía, acompañamiento y paciencia.

## **Agradecimientos**

Agradezco a la Alcaldía Municipal de Gachetá por permitirme realizar el proceso de investigación y recolección de información necesaria que permitió el desarrollo del proyecto, a la Señora Elisa Cárdenas por enseñarme paso a paso sobre el proceso de correspondencia en la entidad y demás funcionarios de la entidad que accedieron de manera amable a la entrevistar donde me explicaron como gestionaban la correspondencia, los problemas generados y las sugerencias que tenían hacia el proceso.

Agradezco a la Universidad Nacional Abierta y a Distancia UNAD, por brindar las herramientas y recursos necesarios para formarme como profesional en el campo de la Ingeniería de Sistemas con habilidades útiles para mi vida laboral.

Gracias a todos los profesores, tutores, compañeros y amigos que estuvieron presentes colaborándome a lo largo de este camino de formación, con sus consejos que me ayudaron a mejorar de manera constante, por sus palabras de aliento para seguir esforzándome hasta finalizar el desarrollo del software de correspondencia para la Alcaldía Municipal de Gachetá.

## Resumen

En la Alcaldía Municipal de Gachetá, la oficina de recepción y correspondencia desempeña un papel fundamental en la gestión de solicitudes, quejas y reclamos, sin embargo, se han identificado deficiencias en el proceso, como la falta de organización en la recepción de documentos, la ausencia de medidas de respaldo de datos y la necesidad de que los ciudadanos se trasladen hasta la entidad para realizar sus trámites y consultas; por esta razón, para abordar los problemas identificados se propone la implementación de un software de correspondencia, el cual sea una solución automatizada que agiliza la recepción de documentos, la generación de radicaciones y el seguimiento de solicitudes, lo que mejoraría significativamente la eficiencia y accesibilidad de los servicios de correspondencia; para su desarrollo se siguen rigurosamente las mejores prácticas del ciclo de vida del software, incluyendo un análisis detallado de los requerimientos, el diseño cuidadoso de interfaces de usuario y el modelado UML, que sirve como base estructural del software, y se identifican e implementan los servicios esenciales para garantizar el correcto funcionamiento del sistema; además, se recomienda a la entidad establecer un plan de copias de seguridad constante, que permita preservar la integridad de la información en caso de posibles incidentes informáticos, asegurando la disponibilidad y confidencialidad de los datos, y para garantizar una transición exitosa hacia esta nueva solución tecnológica, se propone un plan de capacitación que se centre en proporcionar al personal una comprensión sólida del funcionamiento del software, fomentar el uso responsable de la tecnología y crear conciencia sobre la importancia de mantener la integridad de la información, tanto para la entidad como para los ciudadanos.

**Palabras claves:** Algoritmos, correspondencia, bases de datos, tratamiento de datos, solicitudes, archivo, comunicación, flujo de información, virtualización de procesos.

## Abstract

In the Municipal Mayor's Office of Gachetá, the reception and correspondence office plays a fundamental role in the management of requests, complaints and claims, however, deficiencies have been identified in the process, such as the lack of organization in the reception of documents, the absence of data backup measures and the need for citizens to travel to the entity to carry out their procedures and consultations; For this reason, to address the problems identified, the implementation of a correspondence software is proposed, which is an automated solution that streamlines the receipt of documents, the generation of filings and the follow-up of requests, which would significantly improve the efficiency and accessibility of correspondence services; for its development, the best practices of the software life cycle are rigorously followed, including a detailed analysis of the requirements, the careful design of user interfaces and UML modeling, which serves as the structural basis of the software, and the essential services are identified and implemented to ensure the proper functioning of the system; In addition, it is recommended that the entity establish a constant backup plan to preserve the integrity of the information in case of possible computer incidents, ensuring the availability and confidentiality of the data, and to guarantee a successful transition to this new technological solution, a training plan is proposed that focuses on providing the personnel with a solid understanding of the operation of the software, encouraging the responsible use of the technology and creating awareness of the importance of maintaining the integrity of the information, both for the entity and for the citizens.

**Keywords:** Algorithms, archiving, communication, correspondence, databases, data processing, information flow, process virtualization, requests.

## Tabla de Contenido

Introducción .....	34
Planteamiento del Problema .....	35
Definición del problema.....	35
Planteamiento del problema.....	36
Justificación .....	37
Objetivos.....	39
Objetivo General .....	39
Objetivos Específicos.....	39
Marco Referencial.....	40
Marco de antecedentes .....	40
Marco contextual.....	41
Marco teórico .....	43
Gestión de correspondencia.....	43
Tipos de solicitud y tiempos para dar respuesta .....	45
Software para gestión de correspondencia .....	45
Análisis y diseño de un software de correspondencia .....	47
Tipos de software.....	49
Despliegue de software web .....	50
Marco conceptual .....	51
Software tipo web.....	51
Tecnologías para Desarrollo Web .....	52
REST API .....	53

Base de Datos .....	54
Base de Datos MongoDB .....	54
Entorno de Ejecución.....	54
Docker y los Contenedores.....	55
Marco legal.....	55
Metodología .....	57
Resultados .....	59
Recolección de información: .....	59
Definición de población .....	59
Definición de muestra .....	59
Preguntas orientadoras .....	60
Historias de usuario .....	60
Definición de requerimientos.....	64
Requerimientos Funcionales .....	64
Requerimientos no Funcionales .....	67
Diagramas de análisis .....	69
Diagrama de casos de uso .....	69
Diagrama de Actividades .....	72
Diagrama de clases.....	84
Diagrama de despliegue .....	86
Diagrama modelo de base de datos .....	88
Diseño y prototipado.....	89
Definición del nombre de la aplicación.....	89



Diseño del logotipo .....	89
Diseño del Favicon (web) .....	90
Elección de colores.....	90
Diseño de interfaces de usuario.....	91
Prototipado de la Aplicación .....	115
Modelado de bases de datos.....	117
Definición de modelos .....	117
Relación de entidades.....	124
Documentación de Servicio API.....	125
Tipo de Autenticación .....	125
Manejo de Información .....	125
Definición de Rutas .....	126
Rutas para gestión de archivos .....	126
Rutas para gestionar las asignaciones de dependencias .....	127
Rutas de Gestión para Cargos.....	129
Rutas de Gestión para Peticionarios .....	130
Rutas de Gestión de Dependencias.....	131
Rutas de gestión de entidades .....	132
Rutas de Gestión de Usuarios .....	133
Rutas para gestionar las invitaciones.....	135
Rutas de Gestión de permisos.....	136
Rutas de Gestión de Solicitudes .....	137
Rutas de Gestión Tipos de Solicitud.....	138

Rutas de Autorización.....	139
Rutas para gestionar el resumen de datos .....	142
Rutas para la gestión de los selectores de datos .....	143
Rutas para gestionar las asignaciones de radicados.....	145
Ruta para gestión de referencias .....	146
Ruta para gestionar los modales .....	147
Ruta para gestión del timeline .....	150
Rutas para la gestión de copias de seguridad .....	151
Desarrollo del Servicio API .....	153
Configuración del proyecto.....	153
Configuración de TypeScript.....	154
Configuración de la aplicación .....	155
Integración de servicio de logger.....	156
Configuración de variables de entorno .....	158
Instalación y configuración del ORM .....	160
Configuración del servicio principal de emails .....	161
Estructuración de carpetas .....	162
Configuración de modelos de datos .....	164
Definición de interfaces.....	164
Definición de esquema de modelo.....	165
Documentación del esquema .....	165
Definición y exportación del modelo .....	166
Gestión de permisos de aplicación .....	166

Definición de permisos .....	166
Inicialización de permisos .....	171
Validación de permisos .....	172
Gestión de sesiones y accesos .....	172
Generación del token .....	173
Verificación del token de sesión .....	174
Middleware para verificar la sesión.....	175
Gestión de archivos .....	175
Acceso a los archivos .....	176
Inicialización del servicio por primera vez .....	177
Gestión de fechas para radicados .....	179
Formatos de fechas .....	179
Añadir días laborales a fecha.....	180
Calcular diferencia en días laborales entre dos fechas .....	180
Registro de radicados .....	181
Generación de copias de seguridad y restauración de datos .....	183
Copia de seguridad de base de datos .....	183
Copia de seguridad de archivos.....	184
Restauración de copia de base de datos.....	184
Restauración de archivos cargados.....	185
Automatización de tareas de mantenimiento y seguimiento.....	186
Tarea de eliminación de archivos temporales.....	186
Tarea de eliminación de sesiones invalidas .....	187

Tarea de notificación de tiempo restante para responder a radicados .....	187
Desarrollo del Servicio Frontend .....	189
Creación del proyecto.....	189
Creación de rutas de aplicación.....	189
Implementación del estado global (Redux).....	192
Instalación de herramientas .....	192
Creación del store .....	192
Creación de slice.....	193
Integración del estado global.....	194
Instalación de herramienta de desarrollo .....	195
Implementación de Docker .....	196
Configuración del servicio frontend.....	196
Configuración del servicio backend.....	197
Configuración general y unión de los servicios .....	199
Puesta en marcha de los contenedores .....	201
Pruebas de la aplicación Filesoft.....	202
Instalación de la herramienta de pruebas .....	202
Configuración de comandos de inicio.....	203
Inicio de la herramienta.....	203
Manual de uso del servicio API .....	206
Cabecera de solicitud requerida .....	206
Archivos .....	206
Listar archivos .....	207

Obtener archivo .....	209
Subir archivo.....	210
Actualizar archivo.....	213
Eliminar archivo .....	215
Asignación dependencias .....	217
Listar asignación dependencias .....	217
Manual de usuario para el uso de Filesoft .....	219
Conoce Filesoft APP .....	219
Consulta y seguimiento de radicado.....	221
Iniciar sesión en la aplicación .....	224
Restablecer contraseña .....	225
Acceder al panel de administración .....	228
Gestionar el perfil de usuario .....	229
Ver información del perfil.....	229
Actualizar información de perfil.....	230
Gestión de sesiones.....	232
Cerrar sesión .....	232
Gestiona los radicados.....	233
Visualizar listado de radicados .....	233
Visualizar información de radicado .....	235
Generar radicado.....	235
Actualizar cargo del petionario.....	243
Actualizar entidad del petionario .....	245

Actualizar destinatario .....	246
Actualizar asignación radicados .....	247
Asignar responsable.....	248
Eliminar asignación .....	249
Actualizar documento principal.....	250
Cargar archivos adjuntos .....	252
Actualizar número CDs .....	253
Visualizar línea de tiempo .....	254
Actualizar información de radicado .....	255
Eliminar radicado .....	256
Gestiona los archivos .....	257
Previsualizar archivos.....	257
Descargar archivo .....	259
Eliminar archivo .....	259
Gestiona las asignaciones de radicados.....	260
Ver listado de asignaciones.....	260
Ver asignación.....	261
Generar incidencia.....	262
Actualizar incidencia .....	264
Eliminar incidencia.....	265
Establecer respuesta.....	266
Gestiona las invitaciones.....	267
Invitar funcionario .....	268

Ver invitación.....	270
Actualizar invitación.....	270
Eliminar invitación .....	271
Gestiona los funcionarios .....	272
Registrar funcionario .....	272
Ver información de funcionario.....	275
Actualizar funcionario .....	276
Eliminar funcionario.....	277
Gestiona los ciudadanos .....	277
Registrar ciudadano .....	277
Ver información de ciudadano.....	279
Actualizar ciudadano.....	279
Eliminar ciudadano.....	280
Gestiona las entidades .....	281
Registrar entidad.....	281
Actualizar entidad.....	282
Eliminar entidad .....	284
Gestiona las entidades .....	284
Registrar entidad.....	284
Ver detalles de entidad.....	285
Eliminar entidad .....	287
Gestiona las dependencias.....	287
Registrar dependencia.....	288

Ver información de dependencia .....	288
Actualizar dependencia.....	289
Eliminar dependencia .....	289
Gestiona la asignación de dependencias .....	290
Generar asignación .....	290
Ver asignación.....	292
Actualizar asignación .....	292
Gestiona las copias de seguridad.....	293
Copia de seguridad a base de datos .....	293
Copia de seguridad a archivos .....	294
Restablecer base de datos .....	295
Restablecer archivos .....	297
Conclusiones.....	299
Recomendaciones .....	300
Bibliografía .....	301
Apéndices.....	304



### Lista de Tablas

Tabla 1 Tipos de derechos de petición y plazos establecidos .....	45
Tabla 2 Historias de usuario identificadas .....	61
Tabla 3 Definición de requerimientos funcionales .....	64
Tabla 4 Definición de requerimientos no funcionales .....	67
Tabla 5 Detalles de casos de uso.....	69
Tabla 6 Detalle de clases.....	85
Tabla 7 Definición de relaciones entre modelos .....	124
Tabla 8 Grupo de rutas para gestión de archivos .....	126
Tabla 9 Definición de rutas para gestionar las asignaciones de dependencias .....	128
Tabla 10 Grupo de rutas para gestionar los cargos .....	129
Tabla 11 Grupo de rutas para gestión de peticionarios .....	130
Tabla 12 Grupo de rutas para gestionar las dependencias .....	131
Tabla 13 Grupo de rutas para gestionar las entidades.....	132
Tabla 14 Grupo de rutas para gestión de usuarios .....	134
Tabla 15 Grupo de rutas para gestión de invitaciones .....	135
Tabla 16 Grupo de rutas para gestión de permisos .....	136
Tabla 17 Grupo de rutas para solicitudes.....	137
Tabla 18 Grupo de rutas para gestión de tipos de solicitud .....	138
Tabla 19 Grupo de rutas de autorización .....	139
Tabla 20 Grupo de rutas para obtener resumen de datos .....	142
Tabla 21 Grupo de rutas para la gestión de selectores de información.....	144
Tabla 22 Grupo de rutas para gestionar la asignación de radicados .....	145

Tabla 23	Grupo de rutas para gestionar las referencias .....	146
Tabla 24	Ruta para obtener el timeline de un radicado.....	151
Tabla 25	Grupo de rutas para gestionar las copias de seguridad .....	151
Tabla 26	Variables de entorno de la aplicación.....	158
Tabla 27	Permisos de la aplicación.....	167
Tabla 28	Parámetros de URL para listar los archivos .....	207
Tabla 29	Listado de respuestas al solicitar el listado de archivos.....	208
Tabla 30	Parámetros de URL para obtener un archivo .....	209
Tabla 31	Respuestas del servicio al solicitar un archivo.....	210
Tabla 32	Parámetro de URL opcional al subir un archivo .....	211
Tabla 33	Cuerpo de la solicitud para cargar un archivo.....	211
Tabla 34	Respuestas del servicio al subir un archivo .....	212
Tabla 35	Parámetro de URL para actualizar un archivo .....	213
Tabla 36	Parámetros necesarios al actualizar un archivo.....	213
Tabla 37	Respuestas del servicio al actualizar un archivo .....	214
Tabla 38	Parámetros para eliminar un archivo .....	215
Tabla 39	Respuestas del servicio al intentar eliminar un archivo .....	216
Tabla 40	Respuestas del servicio al obtener el listado de asignaciones de dependencias .....	218

## Lista de figuras

Figura 1	Localización y símbolos del municipio de Gachetá .....	41
Figura 2	Edificio Alcaldía Municipal de Gachetá.....	42
Figura 3	Localización Geográfica Edificio Alcaldía de Gachetá.....	42
Figura 4	Proceso de gestión de correspondencia .....	43
Figura 5	Arquitectura cliente - servidor .....	49
Figura 6	Arquitectura de microservicios.....	50
Figura 7	Diagrama de casos de uso entre usuario y sistema.....	69
Figura 8	Diagrama de actividades del caso de uso invitar usuario .....	72
Figura 9	Diagrama de actividades del caso de uso registrar usuario .....	73
Figura 10	Diagrama de actividades del caso de uso iniciar sesión .....	74
Figura 11	Diagrama de actividades del caso de uso consultar.....	74
Figura 12	Diagrama de actividades del caso de uso insertar .....	75
Figura 13	Diagrama de actividades del caso de uso editar .....	76
Figura 14	Diagrama de actividades del caso de uso eliminar.....	77
Figura 15	Diagrama de actividades del caso de uso respaldar.....	78
Figura 16	Diagrama de actividades del caso de uso restaurar .....	79
Figura 17	Diagrama de actividades de caso de uso descargar .....	80
Figura 18	Diagrama de actividades del caso de uso generar reporte.....	81
Figura 19	Diagrama de actividades del caso de uso filtrar .....	82
Figura 20	Diagrama de actividades del caso de uso notificar.....	83
Figura 21	Diagrama UML de clases .....	84
Figura 22	Diagrama de servicios y despliegue .....	87

Figura 23	Modelo de base de datos MongoDB .....	88
Figura 24	Logotipo de la aplicación .....	89
Figura 25	Icono identificador de web .....	90
Figura 26	Colores seleccionados .....	91
Figura 27	Página de inicio de filesoft .....	92
Figura 28	Página de consulta de radicados .....	93
Figura 29	Página de políticas de privacidad de filesoft .....	94
Figura 30	Página de terminos y condiciones de uso de filesoft.....	95
Figura 31	Página acerca de filesoft.....	96
Figura 32	Página de inicio de sesión a la aplicación .....	97
Figura 33	Página de olvidé contraseña de acceso .....	97
Figura 34	Página de validación de funcionarios invitados .....	98
Figura 35	Página de validación de correo electrónico de funcionario.....	98
Figura 36	Página de establecer nueva contraseña de acceso .....	99
Figura 37	Página de registro de nuevo funcionario .....	99
Figura 38	Página del listado de solicitudes o radicados .....	100
Figura 39	Página de generación de nueva solicitud o radicado .....	101
Figura 40	Modal de asignación de nuevo número de radicado .....	102
Figura 41	Modal de edición de nuevo tipo de solicitud.....	102
Figura 42	Página de ver solicitud o radicado.....	103
Figura 43	Modal de edición del cargo del ciudadano .....	104
Figura 44	Modal de asignación de usuarios responsables .....	104
Figura 45	Modal de carga para documento principal .....	105

Figura 46	Modal de carga para adjuntos.....	105
Figura 47	Modal de previsualización de documentos.....	106
Figura 48	Modal de edición para cantidad de CDs.....	106
Figura 49	Modal de creación de incidencia de radicado.....	107
Figura 50	Página para invitar un nuevo funcionario a que se registre.....	107
Figura 51	Página de listado de funcionarios.....	108
Figura 52	Modal de listado de funcionarios invitados.....	109
Figura 53	Página de visualización de funcionario.....	109
Figura 54	Modal de gestión de dependencia del funcionario.....	110
Figura 55	Modal de gestión de permisos del funcionario.....	110
Figura 56	Página de listado de ciudadanos.....	111
Figura 57	Página para insertar un nuevo ciudadano.....	112
Figura 58	Página de visualización de ciudadano.....	112
Figura 59	Modal para insertar un nuevo cargo.....	113
Figura 60	Página de listado de cargos.....	113
Figura 61	Modal de edición de cargos.....	114
Figura 62	Página de listado de entidades.....	114
Figura 63	Modal de edición de entidad.....	115
Figura 64	Prototipado de la aplicación en Figma.....	116
Figura 65	Atributos del modelo de dependencia.....	117
Figura 66	Atributos del modelo de permiso.....	118
Figura 67	Atributos del modelo de invitación.....	118
Figura 68	Atributos del modelo de asignación de dependencia.....	119

Figura 69 Atributos del modelo de archivo.....	119
Figura 70 Atributos del modelo de usuario.....	120
Figura 71 Atributos del modelo de cargos .....	120
Figura 72 Atributos del modelo de peticionario .....	121
Figura 73 Atributos del modelo de entidad.....	121
Figura 74 Atributos del modelo de tipo de solicitud.....	122
Figura 75 Atributos del modelo de solicitud.....	122
Figura 76 Atributos del modelo de asignación de solicitud.....	123
Figura 77 Atributos del modelo de incidencia de solicitud.....	123
Figura 78 Atributos del modelo de línea de tiempo.....	124
Figura 79 Ejemplo formato JSON .....	126
Figura 80 Comando de inicialización del proyecto API .....	153
Figura 81 Representación del archivo package.json del proyecto filessoft.....	154
Figura 82 Ejecución del comando de instalación de la dependencia typescript .....	154
Figura 83 Contenido del archivo tsconfig.json .....	155
Figura 84 Comando de instalación de las librerías base de la aplicación.....	155
Figura 85 Código base de la aplicación .....	156
Figura 86 Comando de instalación de herramientas para logger .....	157
Figura 87 Configuración de la función logger.....	157
Figura 88 Comando de instalación de Mongoose.....	160
Figura 89 Código de configuración de Mongoose ORM.....	161
Figura 90 Instalación del paquete de nodemailer .....	161
Figura 91 Código de configuración del servicio de email con nodemailer .....	162

Figura 92 Estructura básica de carpetas del proyecto .....	162
Figura 93 Ejemplo de definición de interfaz y tipado de modelo .....	164
Figura 94 Ejemplo de definición de esquema para modelo de datos.....	165
Figura 95 Ejemplo de documentación de esquema.....	165
Figura 96 Ejemplo de definición y exportación de modelo .....	166
Figura 97 Código de validación de permisos al iniciar el proceso de la aplicación .....	171
Figura 98 Middleware de validación de permisos .....	172
Figura 99 Respuesta de seguridad al no tener una sesión activa .....	173
Figura 100 Utilidad de generación de token de acceso.....	174
Figura 101 Utilidad para verificar los tokens de sesión.....	174
Figura 102 Código base del middleware que verifica la sesión de un usuario al realizar la solicitud.....	175
Figura 103 Ruta y base para cargar archivos al servidor .....	176
Figura 104 Estructura de la información del archivo almacenado .....	177
Figura 105 Registro de eventos al inicializar el servicio API.....	178
Figura 106 Correo electrónico enviado al usuario administrador al iniciar el servicio .....	178
Figura 107 Utilidad para formateo de fechas.....	179
Figura 108 Base de la utilidad para añadir días laborales a una fecha .....	180
Figura 109 Respuesta de la utilidad para obtener la diferencia en días laborales entre dos fechas .....	181
Figura 110 Pasos para la generación del radicado en el servidor .....	181
Figura 111 Correo electrónico que notifica la edición de un radicado .....	183
Figura 112 Contenido de la copia de seguridad descargada .....	184

Figura 113	Contenido de la copia de seguridad de los archivos .....	184
Figura 114	Respuesta de restauración de base de datos.....	185
Figura 115	Respuesta de restauración de archivos exitosa .....	185
Figura 116	Registro de tareas al iniciar la aplicación .....	186
Figura 117	Ejecución de la tarea de eliminación de archivos temporales .....	186
Figura 118	Ejecución de la tarea de eliminación de sesiones inválidas.....	187
Figura 119	Ejecución de la tarea para notificar el tiempo restante para responder un radicado	188
Figura 120	Creación del proyecto de Next JS .....	189
Figura 121	Representación de la estructura de directorios para generar las rutas físicas de la aplicación .....	190
Figura 122	Instalación de la dependencia de desarrollo de SASS.....	191
Figura 123	Estructura del archivo de configuración del proyecto .....	191
Figura 124	Instalación del gestor de estado global.....	192
Figura 125	Configuración del store de la aplicación .....	193
Figura 126	Configuración del store para manejar los datos de la ui.....	194
Figura 127	Integración del store con ayuda del provider de la librería de React Redux .....	194
Figura 128	Revisión de la herramienta Redux DevTools .....	195
Figura 129	Contenido del archivo dockerignore del servicio frontend .....	196
Figura 130	Archivo de configuración de la imagen del servicio frontend.....	197
Figura 131	Archivo dockerignore del servicio backend .....	198
Figura 132	Archivo de configuración del servicio backend .....	198
Figura 133	Configuración de la base de datos en el archivo Docker Compose .....	199
Figura 134	Configuración del servicio backend en el archivo Docker Compose.....	200



Figura 135 Configuración del servicio frontend en el archivo Docker Compose .....	200
Figura 136 Ejecución de la aplicación con ayuda de Docker .....	201
Figura 137 Resumen de los servicios desde el cliente Docker Desktop .....	201
Figura 138 Creación de nueva rama para probar la aplicación.....	202
Figura 139 Instalación de la herramienta de pruebas Cypress.....	203
Figura 140 Comandos de cy:open y cy:run agregados para pruebas.....	203
Figura 141 Elección del tipo de pruebas que se realizarán .....	204
Figura 142 Selección de navegador para hacer las pruebas.....	204
Figura 143 Listado de pruebas creadas .....	205
Figura 144 Ejecución de pruebas de validación de secciones .....	205
Figura 145 Ejemplo de cabecera Authorization.....	206
Figura 146 Acción de clic sobre el enlace para consultar un radicado .....	219
Figura 147 Acción de clic sobre el enlace para conocer las políticas de privacidad .....	220
Figura 148 Acción de clic sobre el enlace para conocer los términos de uso .....	220
Figura 149 Acción de clic en el enlace para conocer acerca de la aplicación .....	221
Figura 150 Campo de texto para digitar el número de radicado a consultar .....	222
Figura 151 Digitación del año de radicación .....	222
Figura 152 Clic sobre el botón de buscar radicado.....	223
Figura 153 Línea de tiempo del radicado consultado .....	223
Figura 154 Clic sobre el botón de inicio de sesión .....	224
Figura 155 Digitación del correo electrónico para iniciar sesión en la aplicación .....	224
Figura 156 Solicitud de acceso al panel de administración .....	225
Figura 157 Clic en el enlace para ir a el formulario para restablecer la contraseña .....	225

Figura 158	Formulario para comenzar el proceso para restablecer la contraseña .....	226
Figura 159	Clic en el botón para comenzar el proceso de restablecimiento.....	226
Figura 160	Recepción de correo de restablecimiento de contraseña .....	227
Figura 161	Contenido del correo de restablecimiento de contraseña .....	227
Figura 162	Formulario de restablecimiento de contraseña .....	228
Figura 163	Panel de administración de Filesoft.....	229
Figura 164	Componente de gestión de perfil.....	230
Figura 165	Acción para visualizar la información del perfil .....	230
Figura 166	Acción para actualizar la información del perfil .....	231
Figura 167	Clic en el botón para guardar los cambios en el perfil .....	231
Figura 168	Acción para acceder a la gestión de sesiones .....	232
Figura 169	Acción de cerrar sesión.....	233
Figura 170	Navegar al listado de radicados.....	233
Figura 171	Filtrado por fechas y acciones de exportación .....	234
Figura 172	Buscador de registros por palabras clave .....	234
Figura 173	Expandir información de un registro de la tabla .....	235
Figura 174	Acción de ver información de un radicado.....	235
Figura 175	Botón para genera un nuevo radicado .....	236
Figura 176	Cambiar número de radicado automático.....	236
Figura 177	Cambiar fecha de radicación automática.....	237
Figura 178	Campo para registrar el asunto del radicado .....	237
Figura 179	Campo para registrar las observaciones del radicado.....	237
Figura 180	Seleccionar fecha del documento principal.....	238

Figura 181	Campo para establecer el destinatario del radicado .....	238
Figura 182	Campo para establecer los folios del documento principal .....	239
Figura 183	Establecer CDs recibidos.....	239
Figura 184	Establecer días hábiles para dar respuesta.....	239
Figura 185	Selección del peticionario que realiza la solicitud .....	240
Figura 186	Selección del cargo del peticionario.....	240
Figura 187	Selección de la entidad asociada al peticionario .....	241
Figura 188	Carga de documento principal y adjuntos .....	241
Figura 189	Selección del tipo de oficio .....	242
Figura 190	Selección del medio de recepción .....	242
Figura 191	Selección de dependencias a donde se asignará el radicado .....	242
Figura 192	Generación del radicado .....	243
Figura 193	Acción rápida para actualizar el cargo del peticionario.....	243
Figura 194	Formulario para registrar un cargo .....	244
Figura 195	Botón para actualizar el cargo del peticionario .....	244
Figura 196	Acción rápida para actualizar la entidad del peticionario.....	245
Figura 197	Actualizar la entidad del peticionario .....	245
Figura 198	Acción rápida para actualizar el destinatario.....	246
Figura 199	Guardar los cambios del destinatario .....	246
Figura 200	Acción rápida para actualizar las asignaciones .....	247
Figura 201	Selección de funcionarios y dependencias a asignar .....	247
Figura 202	Generar las asignaciones al radicado.....	248
Figura 203	Acción rápida para establecer responsable de radicado .....	248

Figura 204	Confirmar la asignación de responsables .....	249
Figura 205	Acción de eliminar una asignación de radicado .....	249
Figura 206	Confirmar eliminación de asignación.....	250
Figura 207	Acción de actualizar documento principal .....	250
Figura 208	Componente de carga de archivos.....	251
Figura 209	Previsualización de archivo cargado .....	251
Figura 210	Guardar los cambios en el documento principal .....	252
Figura 211	Acción de cargar adjuntos.....	252
Figura 212	Confirmar carga de archivos adjuntos.....	253
Figura 213	Acción de actualizar CDs .....	253
Figura 214	Confirmar actualización de CDs .....	254
Figura 215	Acción de visualizar línea de tiempo.....	254
Figura 216	Línea de tiempo de radicado en panel de administración.....	255
Figura 217	Acción de actualizar información de radicado .....	255
Figura 218	Guardar cambios en la información del radicado.....	256
Figura 219	Acción de eliminar radicado.....	256
Figura 220	Confirmación de eliminación de radicado.....	257
Figura 221	Acción de previsualización de archivos .....	258
Figura 222	Clic en el botón para imprimir el documento .....	258
Figura 223	Formas de cerrar el modal .....	258
Figura 224	Acción de descargar el archivo.....	259
Figura 225	Acción de eliminar archivos .....	260
Figura 226	Confirmar eliminación de archivo.....	260

Figura 227	Ver listado de asignaciones de radicados.....	261
Figura 228	Acción para visualizar los detalles de una asignación de radicados.....	261
Figura 229	Acción de generar incidencia.....	262
Figura 230	Campo para digitar el consecutivo de la incidencia .....	262
Figura 231	Campo para registrar el asunto de la incidencia .....	263
Figura 232	Editor de Mark Down para ingresar el contenido de la incidencia .....	263
Figura 233	Guardar la información de la incidencia .....	264
Figura 234	Acción de editar una incidencia.....	264
Figura 235	Guardar los cambios de la incidencia.....	265
Figura 236	Acción para eliminar una incidencia .....	265
Figura 237	Confirmación de eliminación de incidencia .....	265
Figura 238	Marcar como respuesta del radicado a la incidencia actual .....	266
Figura 239	Recepción de correo de respuesta.....	266
Figura 240	Contenido de la respuesta del radicado .....	267
Figura 241	Acción de invitar funcionario .....	268
Figura 242	Listado de dependencias para asignar al funcionario .....	268
Figura 243	Establecer como administrador el funcionario invitado .....	269
Figura 244	Guardar la información de la invitación y notificar .....	269
Figura 245	Acción de ver detalles de invitación.....	270
Figura 246	Acción de actualizar invitación .....	270
Figura 247	Guardar los cambios en la información de la invitación .....	271
Figura 248	Acción de eliminar una invitación.....	271
Figura 249	Confirmación de eliminación de invitación .....	272

Figura 250	Contenido del correo de registro de funcionario .....	272
Figura 251	Verificación de invitación exitosa .....	273
Figura 252	Guardar información del funcionario .....	273
Figura 253	Confirmación de registro de funcionario exitoso .....	274
Figura 254	Contenido del correo de verificación de cuenta .....	274
Figura 255	Confirmación de verificación exitosa.....	275
Figura 256	Acción de ver los detalles de un funcionario.....	275
Figura 257	Acción de editar la información de un funcionario .....	276
Figura 258	Confirmación de actualización de los datos del funcionario.....	276
Figura 259	Acción de eliminar un funcionario .....	277
Figura 260	Acción de registrar un ciudadano .....	277
Figura 261	Formulario de registro de ciudadano.....	278
Figura 262	Guardar los datos del ciudadano.....	279
Figura 263	Acción de ver los detalles de un ciudadano.....	279
Figura 264	Acción de editar un ciudadano .....	280
Figura 265	Guardar la edición del ciudadano .....	280
Figura 266	Acción de eliminar un ciudadano .....	281
Figura 267	Acción de registrar entidad.....	281
Figura 268	Confirmar el registro de la entidad.....	282
Figura 269	Acción de ver los detalles de una entidad.....	282
Figura 270	Acción de editar una entidad .....	283
Figura 271	Cambiar estado de entidad.....	283
Figura 272	Confirmar la edición en los cambios de la entidad.....	283

Figura 273 Acción de eliminar una entidad .....	284
Figura 274 Acción de insertar una entidad .....	285
Figura 275 Registro de entidad en la aplicación .....	285
Figura 276 Acción de ver los detalles de la entidad.....	286
Figura 277 Acción de actualizar la información de una entidad.....	286
Figura 278 Cambiar estado de una entidad.....	286
Figura 279 Guardar los cambios de edición de información de la entidad.....	287
Figura 280 Acción de eliminar una entidad .....	287
Figura 281 Acción de registrar una dependencia.....	288
Figura 282 Acción de ver detalles de una dependencia .....	288
Figura 283 Acción de editar dependencia .....	289
Figura 284 Guardar cambios de la dependencia .....	289
Figura 285 Acción de eliminar una dependencia.....	290
Figura 286 Acción de generar asignación de dependencia .....	290
Figura 287 Selección del funcionario al que se le asignará la dependencia .....	291
Figura 288 Selección de la dependencia a asignar.....	291
Figura 289 Guardar la asignación de dependencia .....	291
Figura 290 Acción de visualizar una asignación de dependencia.....	292
Figura 291 Acción de actualizar asignación .....	292
Figura 292 Guardar edición en la asignación de dependencia.....	293
Figura 293 Ir al módulo de generación de copias de seguridad.....	293
Figura 294 Acción de generación de copia de seguridad de base de datos.....	294
Figura 295 Confirmación de generación de copia de seguridad de base de datos.....	294

Figura 296 Acción de generar copia de seguridad de archivos.....	295
Figura 297 Iniciar proceso de copia de seguridad de archivos .....	295
Figura 298 Módulo de restauración de copias de seguridad.....	296
Figura 299 Acción de restaurar base de datos.....	296
Figura 300 Carga del archivo de copia de base de datos .....	297
Figura 301 Acción de restaurar los archivos de la aplicación.....	297
Figura 302 Iniciar restauración de archivos.....	298
Figura 303 Cerrar ventana de restauración de archivos.....	298



**Lista de Apéndices**

Apéndice A <i>Decretos, normas y leyes</i> .....	304
--	-----

## Introducción

La oficina de correspondencia en la Alcaldía Municipal de Gachetá cumple un rol fundamental dentro de la entidad, esto se debe a que su principal función es recibir y gestionar solicitudes, quejas y reclamos de ciudadanos o entidades, esta asegura que todas las comunicaciones se atiendan de manera adecuada y eficiente, lo que contribuye a una administración transparente y receptiva a las necesidades de la comunidad.

Sin embargo, este proceso ha presentado dificultades que afectan su eficiencia y efectividad en la recepción de documentos, la falta de copias de seguridad y la organización inadecuada de archivos, además requiere que los ciudadanos se acerquen físicamente para realizar y consultar sus solicitudes, lo cual plantea grandes obstáculos en un municipio donde gran parte de la población pertenece a la zona rural.

Por esta razón en el presente documento se plantea una solución tecnológica moderna que afronte el problema encontrado, siendo este un software de correspondencia que permite la recepción de documentos y generación de radicaciones, los cuales pueden ser monitoreados por los funcionarios de control o incluso la ciudadanía puede conocer el estado de sus solicitudes sin tener que acercarse a la entidad, además permite controlar el flujo del proceso, gestionando los funcionarios asignados a cada radicado y permitiendo generar incidencias sobre este hasta establecer una respuesta a la solicitud.

Para su desarrollo se siguen las diferentes etapas del ciclo de vida del software, analizando los requerimientos funcionales y no funcionales, generando diagramas que funcionan como base para conocer las interacciones del software, el diseño de interfaces de usuario, la selección de nombre, colores y demás elementos que identifiquen el software, la identificación de los servicios involucrados y la manera en que se comunican para procesar la información.

## Planteamiento del Problema

### Definición del problema

El área de correspondencia de la alcaldía municipal del Gachetá se encarga de recibir todas las solicitudes, quejas y reclamos que la comunidad pueda tener, estas son enviadas a la dependencia de destino donde se hará revisión de la solicitud y se dará respuesta al ciudadano; este proceso de radicación se realiza de la siguiente manera:

En la oficina se reciben los documentos necesarios para generar la solicitud, el encargado del área registrará los datos de la solicitud asignándole un número de radicado el cual se imprimirá o se escribirá de manera manual sobre el documento, hecho esto se le entrega una copia al peticionario y se hace el escaneo de los documentos para almacenar una copia digital en el equipo local para finalizar el proceso enviando toda la información a la dependencia encargada de dar solución al radicado.

La dependencia recibe el documento, realiza la impresión y en caso de ser necesario hace una reasignación a otras dependencias hasta dar solución al radicado dentro de los tiempos estipulados, una vez completada la solicitud se le envía la respuesta a través de correo electrónico o se le notifica por llamada telefónica para que se acerque a la entidad, finalizando con la notificación al área de correspondencia para que modifique el estado del radicado en el formato de correspondencia.

Así es como se realiza todo el proceso de correspondencia en el cual se pueden encontrar diferentes problemas que pueden optimizarse con una implementación tecnológica así reduciendo su impacto dentro de la entidad, los problemas encontrados son:

La información se almacena de manera local sin ninguna copia de seguridad que permita una recuperación de la información en caso de fallas en el equipo de correspondencia, además al

no tener una relación directa del radicado con los documentos adjuntos se hace complicado realizar una búsqueda eficiente en el sistema de archivos local.

No es necesario realizar la impresión o escritura manual de la información de asignación del radicado debido a que las nuevas tecnologías permiten escanear y colocar digitalmente todos estos datos para entregar directamente una copia o notificar de manera digital al correo electrónico con la información.

Se generan copias e impresiones innecesarias al estar reasignando el radicado a través de diferentes dependencias, lo que causa una mala organización de los archivos que pueden llevar a la confusión y pérdida de datos al no estar todos informados de los cambios que se van realizando.

Además de manera general el problema es que los peticionarios obligatoriamente deben acercarse a la oficina de correspondencia para realizar sus radicados y a su vez deben acercarse para consultar el estado del mismo cuando han pasado varios días y no han obtenido respuesta, un problema grave teniendo en cuenta que la mayor parte de la población vive a kilómetros de la zona urbana del municipio (ubicación de la alcaldía municipal) dejando la posibilidad de virtualizar este proceso de manera general tanto para los peticionarios como para las dependencias involucradas.

### **Planteamiento del problema**

¿Cómo optimizar el proceso de correspondencia en la alcaldía municipal de Gachetá a través del diseño de un software para la gestión de solicitudes realizadas a las diferentes dependencias de la entidad?

## Justificación

La implementación de herramientas tecnológicas en organizaciones y entidades públicas han traído muy buenos resultados al optimizar la mayoría de sus procesos, en el caso de la Alcaldía Municipal de Gachetá al hacer la implementación de un software de correspondencia se obtendría lo siguiente:

Fácil acceso a la información de los radicados, persistencia de información con ayuda de copias de seguridad, mejor organización, reducción en el número de impresiones, implementación de formatos digitales y facilidad de seguimiento al estado de los procesos.

Se busca que este proceso de radicados de solicitudes, quejas, reclamos o felicitaciones se sistematice completamente permitiendo a sus usuarios realizar la solicitud desde sus hogares por medio de internet, de esta manera se permitirá hacer seguimiento de los procesos por el mismo medio; esta implementación tendría un impacto positivo en la población del municipio evitando que las personas deban desplazarse largas distancias para generar estos radicados, así disminuyendo sus gastos en transporte y documentación como habitualmente lo hacen y solo se deban acercar a la entidad en casos específicos.

La sistematización del área de correspondencia tendrá un impacto medio ambiental al disminuir la cantidad de papel utilizado en cada proceso, debido a que ya no se requeriría realizar una impresión en cada dependencia donde deba ser revisada la solicitud, además al poder automatizar la generación de reportes bajo criterios específicos se disminuirá también el uso y desperdicio de papel a tal punto de poder reemplazar los informes físicos por los informes generados por el software en diferentes formatos accesibles por diversos dispositivos.

Desde el campo disciplinar como estudiante se busca aplicar diversos conocimientos adquiridos durante el desarrollo del programa de ingeniería de sistemas, en este se seguirán todos

los lineamientos establecidos en el ciclo de vida del software con el apoyo de diversas técnicas de investigación y recolección de datos que permitirán analizar desde la planeación del proyecto hasta la ejecución del mismo buscando los mejores resultados, así demostrando las capacidades generadas para el análisis y resolución de problemas en el campos social con uso de herramientas y metodologías modernas.

## Objetivos

### Objetivo General

Diseñar un software de correspondencia que permita optimizar la gestión de requerimientos y solicitudes realizadas a las dependencias de la alcaldía municipal de Gachetá.

### Objetivos Específicos

Analizar las variables que inciden en el problema encontrado a través de la diagramación UML para entender la interacción y comportamiento del software a diseñar además del modelado de base de datos que especifica la estructura de la información a almacenar.

Identificar los servicios involucrados en la aplicación web, determinado cómo se realiza la comunicación entre ellos y cuáles son las respuestas generadas ante eventos específicos.

Diseñar la aplicación web integrando los servicios de comunicación REST con el servicio de almacenamiento de base de datos que se comunicará directamente con el usuario a través de la interfaz.

## Marco Referencial

### Marco de antecedentes

De acuerdo con (Gomez, 2015) en su tesis denominada “Análisis de un prototipo de software para la recepción, entrega y envío de correspondencia en la alcaldía de Taraira – Vaupés”, de la Corporación Universitaria Minuto de Dios (UNIMINUTO), analiza detalladamente el impacto que genera la implementación de tecnologías en el área de correspondencia de la alcaldía municipal de Tairara en el departamento de Vaupés, proponiendo teóricamente un software capaz de realizar la gestión de los procesos de correspondencia en la entidad, así aumentado la eficiencia y organización para dar respuesta a las solicitudes de los ciudadanos.

De acuerdo con (Romero & Valderrama, 2018) en su tesis de grado “Sistema de información para el control de correspondencia y almacén en la alcaldía de Pasca”, de la Universidad de Cundinamarca, el cual menciona que las tareas de correspondencia y almacén de la alcaldía municipal de pasca se realizan de forma manual y esto dificulta la tarea de los responsables de estas áreas, generando diversos problemas como duplicidad de información, lentitud en los procesos, falta de documentación, entre otros problemas que con ayuda de la implementación de un software que registre, gestione y automatice gran parte de estos procesos en la entidad mejore la eficiencia y reduzca la mayoría de los problemas mencionados en la alcaldía municipal de Pasca del departamento de Cundinamarca.

De acuerdo con (Quiceno, 2019) en su tesis de grado denominada “Software de gestión documental y ventanilla única para administraciones municipales de sexta categoría”, de la Universidad Autónoma de Manizales, menciona que las entidades públicas, en especial las alcaldías elaboran, tramitan, contestan y archivan documentos en el desarrollo de sus actividades cotidianas, para lo cual requieren el almacenamiento de documentos producidos y recibidos,



tramitados y archivados de tal forma, que se puedan encontrar de manera sencilla, para esto realiza la propuesta de un software de gestión documental para ventanilla única que registre las solicitudes y permita un acceso a esta información dentro de la entidad facilitando el proceso hasta dar respuesta, además de disminuir los problemas de organización y sincronización de la información en todas las áreas de las entidades donde se pueda realizar su implementación.

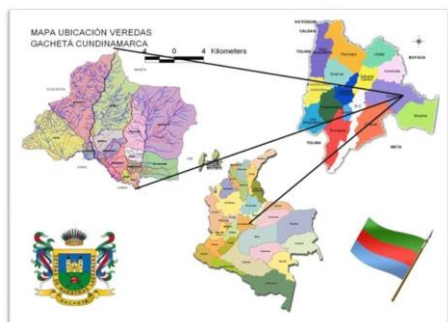
De acuerdo con (Ascencio & Pinilla, 2017) en su tesis de grado denominada “Sistema de información para la ventanilla única de la alcaldía municipal de Ubaté”, de la Universidad de Cundinamarca, propone un software para gestión de archivos buscando aumentar su accesibilidad dentro de la entidad y automatizando el proceso de correspondencia el cual se enfoca en la recepción, radicación y distribución de comunicados a todos los ciudadanos del municipio de Ubaté en el departamento de Cundinamarca así facilitando las tareas a los funcionarios de la entidad.

### **Marco contextual**

El edificio de la alcaldía se encuentra localizado en el casco urbano del municipio de Gachetá, en la carrera 4ª frente al parque principal y la iglesia.

### **Figura 1**

*Localización y símbolos del municipio de Gachetá*



*Fuente.* Localización municipio de Gachetá (Guzmán, 1993)

Como referencia, el municipio de Gacheta se encuentra a 92 kilómetros de la ciudad de Bogotá, Vía Carretera La Calera o Carretera Norte a Briseño, Sopó, Páramo, peaje El Amoladero, Sueva y por último Gachetá, en este nos desplazamos por la Carrera 8, se toma la Calle 6ª hasta la carretera 4 hasta llegar al parque principal donde encontraremos el edificio de la alcaldía municipal a mano derecha.

## Figura 2

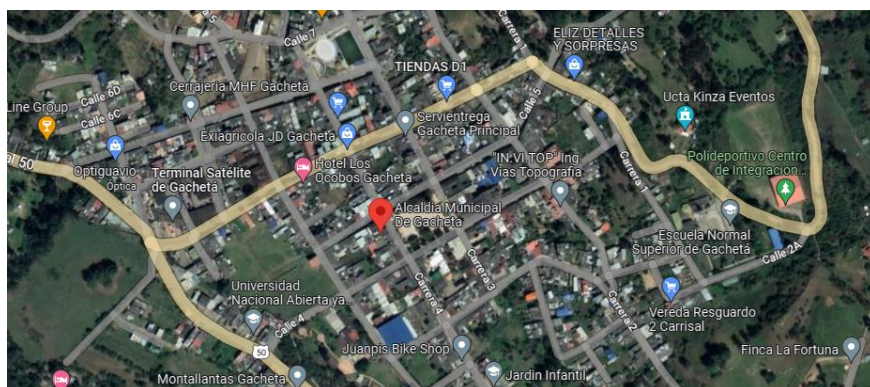
### *Edificio Alcaldía Municipal de Gachetá*



Fuente. Edificio Alcaldía de Gachetá (Vallejo, 2019)

## Figura 3

### *Localización Geográfica Edificio Alcaldía de Gachetá*



Fuente. Imagen tomada de Google Maps

## Marco teórico

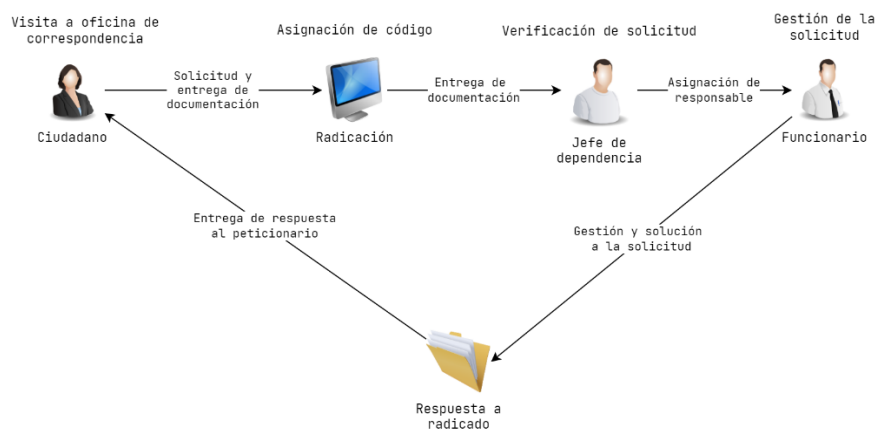
En las entidades gubernamentales es fundamental contar con una oficina de correspondencia ya sea física o virtual que permita a los ciudadanos realizar peticiones, quejas, reclamos o felicitaciones, además de poder radicar documentación de diferentes tipos la cual está sujeta a ciertas normativas que se deben cumplir a la hora de generar una respuesta al ciudadano u otras entidades o empresas.

### *Gestión de correspondencia*

La gestión de la correspondencia se realiza a través de las etapas de recepción en ventanilla o correo electrónico, los cuales son radicados con la asignación de un código de identificación que se entrega al ciudadano para que pueda realizar el seguimiento de la solicitud, a su vez esta información junto a los documentos adjuntos son entregados al jefe de la dependencia destino para que lo asigne al funcionario correspondiente, este se encargará de generar la respuesta al solicitante y entregársela ya sea de manera digital o en físico.

## Figura 4

### *Proceso de gestión de correspondencia*



*Fuente.* Elaboración propia en el software Draw.io

En la *Solicitud y entrega de documentación*, el ciudadano, entidad o empresa se acerca hasta la oficina de correspondencia de la entidad o envía un correo electrónico con el documento principal y los documentos adjuntos que sean necesarios para su radicación.

En la *radicación y asignación de código*, el encargado del área registra los datos de la solicitud junto a la documentación pertinente a la cual le asigna un código con el que se identificará la solicitud durante en todo el proceso de gestión.

En la *entrega a la dependencia correspondiente y validación de la solicitud*, el encargado revisará detalladamente la solicitud para verificar si a la dependencia le corresponde dar respuesta o se debe hacer una reasignación a otra dependencia; si la solicitud es válida para dar respuesta se asignará al funcionario correspondiente dentro de la dependencia.

En la *gestión de la solicitud*, el funcionario realizará todo el proceso hasta dar respuesta al ciudadano, dentro de estos pasos puede solicitar ayuda a otras dependencias o funcionarios dependiendo la necesidad, además, puede generar incidencias para que sean evaluadas por todos los funcionarios a los que se les haya asignado la solicitud.

Por último, en la respuesta y entrega al ciudadano, el funcionario se hará responsable de enviar a través de medios digitales o informar de alguna manera al ciudadano que su solicitud ha sido resuelta de manera satisfactoria, esto con el fin de no incurrir en desacato a las normas vigentes de tiempos establecidos para dar respuestas a las solicitudes de los ciudadanos; alguna falla puede generar problemas judiciales a la entidad por su ineficiente gestión. (*Acuerdo 60 de 2001 Archivo General de la Nación - Gestor Normativo, 2015*)

### ***Tipos de solicitud y tiempos para dar respuesta***

La siguiente información es establecida por el gobierno nacional de Colombia para que sean acatadas por todas las entidades gubernamentales en la gestión de solicitudes, teniendo en cuenta que todos son tratados como derechos de petición asumiendo que su tiempo de respuesta varía de acuerdo con el tipo de derecho de petición.

#### **Tabla 1**

##### *Tipos de derechos de petición y plazos establecidos*

Tipo de derecho de petición	Plazo establecido para dar respuesta
Derecho de petición general	15 días hábiles
Derecho de petición para solicitud de información	30 días hábiles
Derecho de petición general con copia	10 días hábiles
Derecho de petición especial	Mínimo 24 horas

*Nota.* Tiempos establecidos por el gobierno nacional para dar respuesta a solicitudes.

El no cumplimiento de esta normativa puede provocar acciones legales por parte del ciudadano hacia la entidad, por esta razón es fundamental mantener una gestión correcta de las solicitudes llevando ese orden cronológico y de importancia. (*Normalización de tiempos de respuesta a derechos de petición en Colombia | Insights | Holland & Knight, s. f.*)

### ***Software para gestión de correspondencia***

El software para gestión de correspondencia es una herramienta que permite a las entidades gestionar de manera eficiente y efectiva toda la información que recibe y emite en su día a día. Su objetivo principal es automatizar el proceso de gestión de correspondencia, lo que

ayuda a las entidades a ahorrar tiempo y recursos en su gestión manual, esto se logra a través de varias funcionalidades que ofrece y que sustituyen procesos de la entidad que en el pasado se manejaban de manera manual, las principales etapas son:

1. Recepción y registro de correspondencia: Permite registrar la correspondencia que llega a la entidad, incluyendo los datos de contacto del remitente, la fecha de recepción, el tipo de documento, entre otros datos relevantes que ayudan en la gestión de la solicitud.

2. Seguimiento y trazabilidad de la correspondencia: Lleva un registro detallado de los estados por los que pasa cada solicitud, desde que es recibida hasta que es radicada o contestada por un funcionario.

3. Asignación de responsabilidades: Le permite asignar tareas y responsabilidades específicas a los administradores de entidades para monitorear y administrar las solicitudes.

4. Generación de informes: Permite generar informes detallados de las solicitudes generadas, incluyendo estadísticas, gráficos y datos relevantes para la toma de decisiones dentro de la entidad, además de poder analizar fácilmente el porcentaje de cumplimiento de los ciudadanos, entidades o empresas solicitantes.

5. Alertas de tiempo para seguimiento de plazos: Tiene la capacidad de establecer términos y plazos para el manejo de cada solicitud y asignar alertas horarias para recordar a los administradores de la entidad dar seguimiento a la solicitud y brindar una pronta respuesta.

Con estas funcionalidades obtienen varias ventajas como mayor eficiencia y productividad, mejora en la calidad del servicio, mayor seguridad y confidencialidad, y facilidad de acceso y consulta. Estas ventajas permiten a la entidad gestionar grandes cantidades de información de manera efectiva, mejorar sus procesos y tomar decisiones. (Solutions, s. f.)

### ***Análisis y diseño de un software de correspondencia***

El análisis y diseño de un software de este tipo implica una amplia investigación de la necesidad y un análisis de los procesos que realizan diferentes entidades para dar respuesta a las solicitudes de los ciudadanos, por esta razón se divide en diferentes etapas donde es evaluado por profesionales en áreas como desarrollo de software, diseñadores gráficos, analistas de sistemas o de requerimientos, entre otros.

En la *primera fase* se observa y se analiza detalladamente la necesidad, si es posible se realizan entrevistas con personas asociadas a entidades que manejen este proceso de correspondencia, para así conocer como lo realizan, ver las falencias que se pueden optimizar y definir las problemáticas encontradas las cuales son la base de la oportunidad de desarrollo del software.

En la *segunda etapa* se toman los datos de la fase anterior y se saca una a una las funcionalidades que debe contener el software, esto se acostumbra a hacer en dos fases, una con la definición de las historias de usuario y la segunda con la definición final de los requerimientos funcionales y no funcionales del software, esta etapa es realizada por un analista de sistemas o analista de requerimientos.

En la *tercera etapa* se busca facilitar la comprensión del problema, por esta razón usando de base los modelos UML se grafican las acciones fundamentales del software usando diagramas como el de casos de uso, el de despliegue, el de actividades, entre otros que puedan representar la información de manera visual; esto por lo general lo realiza un arquitecto de software pues es de las etapas más importantes ya que es la base de todo desarrollo.

En la *cuarta etapa* se realiza el modelado de la base de datos, siendo esta la que almacenará toda la información ya sea en tablas o en documentos que se relacionaran entre sí

para crear restricciones y atributos especiales que faciliten la consulta de datos, esta incluye los modelos encontrados, sus atributos (con tipo de dato y restricciones) y las relaciones entre estos.

En la *quinta etapa* se diseñan las interfaces de usuario, estas serán las encargadas de mostrar la información a los usuarios del software, por lo cual, es fundamental que cumplan varios criterios de accesibilidad que aumenten el porcentaje de posibles usuarios, para esta tarea existen profesionales en diseño UI/UX que se encargan de elegir colores, diseños, arquitecturas, etc.

La *sexta etapa* ya implica la programación del software, en esta se deben implementar una a una las funcionalidades definidas en la etapa de definición de requerimientos, por lo general es la que lleva más tiempo ya que dependiendo de la metodología de desarrollo se puede implementar la etapa de pruebas tras el desarrollo de alguna funcionalidad con el fin de encontrar fallas a corto plazo y no hasta finalizar todo el software; los encargados de esta etapa son desarrolladores de software.

La *séptima etapa* se enfoca en las pruebas del software, busca verificar que cada funcionalidad esté acorde a lo solicitado y responda de manera correcta ante la interacción y entrega de datos, para esto existen profesionales en aplicación de pruebas unitarias, e2e, etc.

Las anteriores etapas pueden variar de acuerdo con la complejidad del software a desarrollar, pero sigue usando como base el ciclo de vida establecido, solo que con leves modificaciones que ayuden a generar un software de calidad que cumpla con diferentes estándares como de accesibilidad, funcionalidad o rendimiento los cuales son fundamentales.

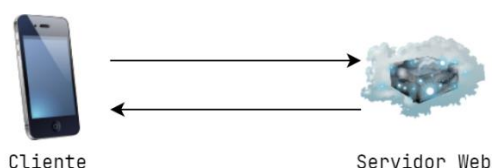


### ***Tipos de software***

Lo anterior que describe las fases o etapas para el desarrollo de software puede ser aplicado sin importar el tipo de plataforma de ejecución, pueden ser aplicaciones web, de escritorio, de servidor, para dispositivos móviles, entre otras, pero debido a su fácil accesibilidad, despliegue y desarrollo la mejor opción es una aplicación web.

### **Figura 5**

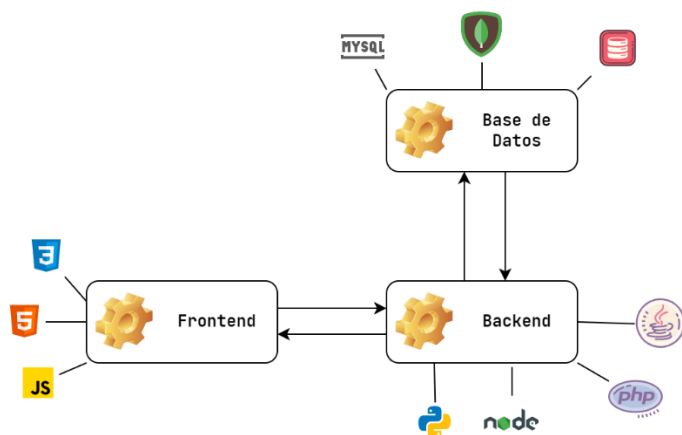
#### *Arquitectura cliente - servidor*



*Fuente.* Elaboración propia en el software Draw.io

Este tipo de software por lo general utilizan la arquitectura cliente – servidor el cual el cliente es la parte que solicita los recursos y servicios, mientras que el servidor es la parte que los proporciona. El cliente y el servidor se comunican entre sí a través de una red, generalmente mediante un protocolo estándar como HTTP o TCP/IP.

La lógica de negocio se encuentra en el lado del servidor, esta puede estar estructurada siguiendo arquitecturas de software establecidas, como la arquitectura monolítica, arquitectura basada en microservicios, arquitectura basada en contenedores, arquitectura basada en eventos, entre otros, para la cual se selecciona una arquitectura básica de microservicios para separar el frontend, el backend y la base de datos; se selecciona ya que permite la separación de responsabilidades, lo que facilita el mantenimiento y la escalabilidad.

**Figura 6***Arquitectura de microservicios*

*Fuente.* Elaboración propia en el software Draw.io

***Despliegue de software web***

El despliegue consiste en llevar el software web desde el ambiente de desarrollo hasta un ambiente de producción, donde pueda ser visitado por los usuarios desde cualquier parte. Este proceso puede incluir varias tareas, como la configuración del servidor, la implementación del código de la aplicación, la configuración de la base de datos, la configuración de las redes, entre otras que permiten hacer una gestión óptima de los recursos y la seguridad del software.

Una buena estrategia para el despliegue es garantizar que la aplicación se ejecute correctamente y sin interrupciones en el ambiente de producción. Un despliegue mal planificado o implementado puede provocar errores, inestabilidad, y un tiempo de inactividad significativo que puede afectar negativamente la experiencia de los usuarios y los resultados de la entidad. Por lo tanto, es importante prestar atención al proceso de despliegue y asegurarse de que se realice de manera eficiente y segura.

## **Marco conceptual**

El tipo de software para diseñar y responder a la necesidad de la entidad es de tipo web el cual se elige debido a que su acceso se dará a través de un navegador web diferentes tipos de dispositivos, permitiendo una mayor seguridad de la información al almacenarse en un servidor externo a la entidad que puede otorgar el acceso a un número específico y limitado de usuarios.

### ***Software tipo web***

Los softwares tipo web permiten a los usuarios acceder a ella a través de un servidor web utilizando un navegador con conexión a internet. El navegador es responsable de solicitar los recursos de la aplicación, interpretarlos y presentarlos al usuario en una interfaz gráfica. Estas aplicaciones son muy populares debido a su independencia del sistema operativo y al uso eficiente de los recursos del dispositivo, lo que las hace convenientes para su uso en equipos con limitaciones de recursos. (GCFGlobal, 2019)

Estas al ser accedidas desde un navegador web siguen la arquitectura cliente servidor donde:

**Cliente.** Es el navegador, siendo el responsable de recibir y procesar la información de una aplicación web. Este realiza solicitudes al servidor para obtener recursos y enviar información utilizando los encabezados o el cuerpo de la petición. Para mostrar correctamente la información al usuario, el navegador utiliza tecnologías como HTML, CSS y JavaScript para renderizar la interfaz gráfica de la aplicación.

**Servidor.** Es responsable de procesar las solicitudes del cliente y manejar la lógica de negocio en una aplicación web. Cuando un cliente solicita acceso a una página específica, el servidor verifica si el cliente tiene permiso y si la solicitud se realizó correctamente. Si todo está en orden, el servidor envía la información necesaria para cargar la página en el cliente. El

servidor generalmente está conectado a una base de datos y a un sistema de archivos para almacenar y manipular la información según sea necesario. (Schiaffarino, 2019)

### ***Tecnologías para Desarrollo Web***

Para seleccionar una tecnología para desarrollar aplicaciones web se tienen muchas opciones, por ejemplo para el lado del cliente se pueden aplicar las tecnologías bases como HTML, CSS y JavaScript pero ya existen frameworks como React, Angular, Vue JS, Svelte JS, entre otros que agilizan el desarrollo y conexión con el servidor, para trabajar del lado del servidor aún hay más opciones debido a que existen herramientas y frameworks diseñados en muchos lenguajes de programación, por ejemplo para JavaScript se tiene Node JS, para C# se tiene .NET, para Java se maneja Spring Bot, para Python está Django y eso mencionando los más populares en general. (KeepCoding, 2022a)

**Tecnología Elegida del Lado del Cliente.** Para conseguir una aplicación que tenga buen rendimiento y pueda escalar fácilmente, se decidió utilizar Next JS, que es un marco de desarrollo basado en Node JS. Permite trabajar con componentes usando React para el renderizado en el cliente, pero la mayor ventaja es que renderiza la información en el servidor, lo que significa que el cliente puede ver la información procesada inmediatamente sin tener que procesarla él mismo, lo que mejora el rendimiento en el cliente y una mayor experiencia.

Esto lo logra utilizando una técnica llamada "*Server Side Rendering*" que procesa y renderiza la información antes de enviarla al cliente. Es decir, la información se envía ya preparada y renderizada, por lo que el navegador del cliente no tiene que pasar por el proceso de carga de scripts para mostrarla. Esta técnica se desarrolló porque se encontraron problemas con los indexadores de páginas web, como Google, que no podían procesar correctamente el contenido. (Aplyca, 2022)

**Tecnología Elegida del Lado del Servidor.** El marco principal de desarrollo para el servidor es Node JS, un entorno de ejecución de JavaScript que permite ejecutar tareas de forma asíncrona, lo que mejora el rendimiento de las aplicaciones construidas con este marco. Node JS ofrece funcionalidades integradas como la creación de servidores web y la gestión de archivos, lo que lo hace efectivo para el tipo de aplicación a diseñar. Este ha sido uno de los pilares en la revolución de las aplicaciones web y de escritorio modernas, incluyendo notificaciones push, sockets, eventos sincrónicos y muchas más características. (Henry, 2021)

### ***REST API***

Es un conjunto de protocolos y definiciones que responden a las solicitudes de los clientes. A diferencia de las aplicaciones web tradicionales que envían archivos HTML, CSS, JavaScript e imágenes, una REST API responde con información en formato JSON o XML (aunque el formato XML se ha vuelto menos común en estas aplicaciones en los últimos años). El cliente interpreta esta información y decide qué mostrar al usuario y qué información no es útil. (RedHat, 2020)

Estas aplicaciones responden a cuatro tipos de solicitudes diferentes (GET, POST, PUT y DELETE), las cuales corresponden a las cuatro operaciones básicas de cualquier base de datos (CRUD). Esto permite que la aplicación pueda identificar de manera sencilla qué acción desea el cliente realizar tan solo a través de la solicitud que este envía, la cual puede contener información predefinida y/o información adicional proporcionada por el cliente. Por esta razón, se utiliza este tipo de aplicación en el lado del servidor, y en particular, se puede utilizar Express JS, un marco de trabajo de Node que facilita y agiliza el desarrollo de estas aplicaciones.

### ***Base de Datos***

Después de definir las tecnologías a utilizar en un proyecto, es necesario elegir el tipo y gestor de base de datos que se utilizará. En la actualidad, existen dos tipos principales de bases de datos: las relacionales y las no relacionales. Las bases de datos relacionales funcionan mediante una estructura de tablas que se pueden referenciar entre sí mediante llaves especiales que identifican cada columna. Para gestionarlas se utiliza SQL, un lenguaje de consulta que permite crear bases de datos, leer, insertar, actualizar y eliminar registros. (Marín, 2022)

Por otro lado, las bases de datos no relacionales funcionan mediante la estructura de documentos y aunque no existe un estándar común, la mayoría almacena la información en formato JSON. Un ejemplo de base de datos no relacional muy popular es Mongo DB.

### ***Base de Datos MongoDB***

Es una base de datos no relacional de código abierto que permite el almacenamiento y consulta de datos de manera flexible y escalable, tiene capacidad de manejar grandes cantidades de datos de manera eficiente, su facilidad de uso y su flexibilidad en cuanto a la estructura de los datos que puede almacenar. Este utiliza un modelo de documentos que permite el almacenamiento de datos en un formato similar a JSON, lo que lo hace especialmente útil para aplicaciones web y móviles.

### ***Entorno de Ejecución***

Cada uno de los servicios se debe ejecutar de manera individual y comunicarse entre sí, sin importar si se encuentran en la misma red o en localizaciones diferentes, siguiendo la arquitectura de microservicios se cumple esta condición en el despliegue, para el cual se puede utilizar una herramienta de contenedores como lo es Docker para ejecutar independientemente

cada servicio (frontend, backend y base de datos) y comunicarlo a través de la exposición de puertos lógicos en el equipo donde se ejecuta.

### ***Docker y los Contenedores***

Esta es una plataforma de contenedores que permite a los desarrolladores empacar, distribuir y ejecutar aplicaciones en cualquier entorno. Un contenedor de Docker es una unidad de software liviana y portátil que encapsula una aplicación y sus dependencias, lo que facilita la migración de aplicaciones entre diferentes entornos y sistemas operativos. Este permite la creación de imágenes de contenedores que incluyen todo lo necesario para que una aplicación se ejecute, desde el código fuente hasta las bibliotecas y herramientas necesarias. (Amazon Web Services, 2020)

Permite la gestión de contenedores y la preparación para despliegues a gran escala, lo que lo hace ideal para implementaciones en la nube y en entornos de producción.

### **Marco legal**

Este se constituye por diferentes normas, leyes y decretos que determinan la gestión de tecnologías por parte de los ciudadanos al igual que en las entidades gubernamentales o empresas, las cuales dan a conocer los derechos, deberes y sanciones tras el incumplimiento, en el apéndice A, se describen de manera individual de manera detallada.

- Ley 1341 de 2009 Ley de Tecnologías de la Información y las Comunicaciones, fomenta la competencia y protección de los derechos de los usuarios, a través de la regulación en el desarrollo de nuevas tecnologías.

- Ley 29 de 1990 Fomento de la investigación científica y el desarrollo tecnológico, la cual crea condiciones favorables para la generación de conocimiento, la capacidad innovadora y la investigación científica.
- Ley 1581 de 2012 Protección de datos personales, la cual establece los principios y obligaciones para la protección de datos personales en Colombia, garantizando el derecho a la privacidad y protección de la información personal.
- Ley 962 de 2005 Racionalización de trámites y procedimientos administrativos, establece la reserva legal de permisos y requisitos, trámites de ley, información y publicidad, y el fortalecimiento tecnológico de la Administración Pública.
- Decreto 1078 de 2015 Estructura y reglamentación del sector de tecnología, expide el Decreto Único Reglamentario del Sector de Tecnologías de la Información y las Comunicaciones en Colombia, estableciendo su estructura y régimen reglamentario.
- Ley 1755 de 2015 Derecho fundamental de petición, establece las normas y plazos para que las personas puedan presentar peticiones de manera respetuosa y recibir una respuesta pronta y completa por parte de las autoridades y entidades gubernamentales.



## Metodología

Para la ejecución del proyecto de desarrollo de software se realizará una **investigación de campo** la cual se apoya en entrevistas y observaciones para la recolección de información, con esto se busca analizar detalladamente el problema encontrado en la entidad, lo que permite definir las funciones que se deben implementar en el software para una solución u optimización del problema; además se complementará con una **investigación documental** que permitirá definir elementos legales como el tiempo establecido para cada tipo de solicitud de acuerdo a su prioridad, también ayudará en la definición de la estructura de la información almacenada y los lineamientos establecidos para la generación de reportes individuales o generales.

Esta investigación se realizará utilizando como población a los funcionarios de la alcaldía de Gachetá que son un total de 35 personas, de esta se toma una muestra de 15 personas a las que se les realizará las entrevistas enfocadas a conocer los posibles problemas que tengan actualmente y cómo es el proceso que realizan en cada solicitud asignada.

Después de aplicar la investigación se define que para el desarrollo del software se seguirá todo el ciclo de vida que involucra un software en general, este consta de cinco fases en total que se deben ejecutar de manera ordenada hasta cumplir el objetivo general, esas son:

**Fase 1. Planeación.** Se debe definir la viabilidad, los recursos necesarios, los costos de desarrollo y los posibles riesgos que se deben enfrentar.

**Fase 2. Análisis.** El objetivo es analizar con detalle lo que realmente se necesita y debe realizar el software, esto comúnmente se conoce como levantamiento de requerimientos que involucra detalles de rendimiento, seguridad, arquitectura a utilizar y estructura de la información.

**Fase 3. Diseño.** En esta fase se estudian las diversas opciones que se podrían

implementar a la solución y la estructura que tendrá el mismo, inicialmente la solución no es la más óptima por lo que con el paso del tiempo se irá optimizando en base al objetivo general hasta obtener una propuesta sólida.

**Fase 4. Implementación.** Se requiere la elección de las herramientas a utilizar, en este se incluyen los lenguajes de programación, editores de texto, la lógica y arquitectura de la aplicación, en esta etapa ya se codifica lo planeado y se documenta de acuerdo con los resultados de las fases anteriores.

**Fase 5. Pruebas.** Esta va de la mano con la etapa de implementación ya que se realizan por secciones, esto permite hacer pruebas unitarias, aunque la codificación no haya finalizado por completo, aquí se buscan pequeñas fallas o funciones que se puedan optimizar para lograr una mejor experiencia o rendimiento en el software.

## **Resultados**

### **Recolección de información:**

Para definir unas bases sólidas sobre la necesidad y el software de correspondencia a construir para la Alcaldía Municipal de Gachetá, se requiere información confiable que permita conocer la situación actual de la entidad en el área de correspondencia y cuáles son los procesos a detalle que se llevan a cabo para la gestión de la correspondencia en la entidad.

El método de recolección de información que se utilizó es la entrevista, esto porque permite interactuar directamente con los funcionarios de la entidad y conocer los procesos que llevan a cabo, las dificultades que han tenido y así definir las oportunidades de mejora desde un punto de vista de implementación tecnológica.

### **Definición de población**

La alcaldía municipal de Gachetá cuenta con un total de 53 funcionarios en su plantilla, entre los que se incluyen tanto funcionarios de planta como contratistas. Los funcionarios de planta son aquellos que han sido nombrados mediante un proceso de selección regido por la ley y que ocupan un cargo de manera permanente. Por otro lado, los contratistas son aquellos que prestan servicios a la entidad de manera temporal, mediante un contrato que establece las condiciones de su vinculación.

### **Definición de muestra**

Se seleccionan los encargados de cada una de las dependencias principales de la entidad, para un total de 12 funcionarios que se tomarán como muestra, las dependencias involucradas

son: correspondencia, hacienda, salud, secretaría desarrollo social, secretaría de gobierno, secretaría ejecutiva, contratación, control interno, planeación, almacén, desarrollo económico y sisbén.

### **Preguntas orientadoras**

Se establecen cuales son las preguntas que se tomarán como base para entrevistar a los funcionarios seleccionados, estas ayudarán a determinar las necesidades de la entidad en el área de correspondencia, lo cual será la base para definir las historias de usuario, las preguntas orientadoras son: ¿Cómo gestiona los procesos asignados por la dependencia de correspondencia?, ¿Ha tenido algún problema al gestionar el proceso? de ser así ¿Cuál ha sido el problema?, ¿Cuáles son los tiempos que maneja para cada tipo de solicitud? Si existiera un software para gestionar los procesos de correspondencia, ¿Qué funcionalidades debería tener?, ¿Qué personas tienen acceso a la información manejada en la solicitud?, ¿Cómo y cada cuanto realizan reportes de información? Y por último ¿Cómo almacenan los documentos anexos a cada solicitud?

### **Historias de usuario**

Las historias de usuario son una técnica utilizada en el desarrollo de software para describir las funcionalidades de la aplicación en un lenguaje sencillo que todos puedan entender y expliquen el porqué de la funcionalidad que se implementará, estos son diferentes a los requerimientos de la aplicación pues no abordan un lenguaje técnico, las historias de usuario encontradas después de realizar las entrevistas son:

**Tabla 2***Historias de usuario identificadas*

<b>No.</b>	<b>Historia de usuario</b>
1	Como usuario de correspondencia quiero poder registrar las nuevas solicitudes para gestionarlas.
2	Como usuario de correspondencia quiero poder ver el estado de todas las solicitudes para hacerles seguimiento.
3	Como usuario de correspondencia quiero poder registrar donde quedan los documentos físicos de una solicitud para poder ubicarlos con mayor facilidad.
4	Como usuario de correspondencia quiero poder generar un reporte general o por dependencia para presentar ante la dependencia de control.
5	Como usuario de correspondencia quiero poder subir un archivo Excel con la información antigua para almacenarla.
6	Como usuario de correspondencia quiero gestionar los usuarios asignados para limitar el acceso a la información.
7	Como usuario de dependencia quiero poder ver todas las solicitudes asignadas a mi usuario para poder gestionarlas.
8	Como usuario de dependencia quiero poder recibir alertas de tiempo para poder estar atento a responder las solicitudes en el tiempo establecido.
9	Como usuario de dependencia quiero poder buscar una solicitud específica para ver su estado.
10	Como usuario de dependencia quiero recibir una alerta en el software y por correo cada que se asigne una solicitud para estar al tanto de las asignaciones.

- 11 Como usuario de dependencia quiero poder reasignar alguna solicitud a otra dependencia para informar o solicitar ayuda.
  - 12 Como usuario de dependencia quiero poder crear incidencias sobre la solicitud para adjuntara archivos o especificar mensajes.
  - 13 Como usuario de dependencia quiero poder ver los correos asociados al correo para estar al tanto de posibles solicitudes.
  - 14 Como usuario de dependencia quiero poder generar informes de acuerdo con todas las solicitudes asignadas para conocer el cumplimiento de respuesta a las mismas.
  - 15 Como usuario de dependencia quiero que solo las dependencias asignadas tengan acceso a los detalles y adjuntos de una solicitud para una mayor seguridad y privacidad de los procesos.
  - 16 Como usuario de seguimiento quiero poder todas las solicitudes generadas, además de todos los detalles de la solicitud para tener un control de las solicitudes en proceso y finalizadas.
  - 17 Como usuario de seguimiento quiero recibir notificaciones de las solicitudes que estén prontas a vencer para hablar con la dependencia encargada.
  - 18 Como usuario de seguimiento quiero poder generar informes de todas las solicitudes en general para presentar ante entes de control.
  - 19 Como usuario de gobierno quiero poder ver toda la información de las solicitudes, esto incluye documentos adjuntos, detalles, incidencias y dependencias asignadas para conocer lo trabajado en la entidad.
  - 20 Como usuario administrativo quiero poder crear los usuarios y asignarles los permisos que tendrán el sistema para definir exactamente quien podrá tener acceso al sistema.
-

- 21 Como usuario administrativo quiero poder generar y restaurar copias de seguridad de los archivos y los registros para mantener a salvo la información en caso de generarse algún problema.
- 22 Como usuario administrativo quiero tener acceso al listado de ingresos que se han generado en el sistema para mantener un seguimiento de los usuarios.
- 23 Como usuario administrativo quiero tener acceso a los archivos de registro para ver el comportamiento del sistema.

---

*Nota.* Historias de usuarios identificadas tras entrevistar los funcionarios.

### Definición de requerimientos

Los requerimientos definirán el funcionamiento y características que deberá cumplir el software de correspondencia, dentro de esta están sus limitaciones en rendimiento, las funcionalidades de usuario, elementos de seguridad, entre otros que determinan a mayor detalle el funcionamiento del software.

### Requerimientos Funcionales

Después de la lectura y análisis de las historias de usuario se encontraron los siguientes requerimientos funcionales:

#### Tabla 3

##### *Definición de requerimientos funcionales*

No.	Requerimiento funcional
1	El sistema permitirá el acceso a los usuarios que estén registrados y tengan credenciales de acceso.
2	La sesión se mantendrá activa usando el método de autenticación de Bearer Token.
3	El sistema permitirá el registro de nuevos usuarios a través de un usuario administrador el cual los agregará como usuarios válidos para registro.
4	El sistema enviará un correo a los usuarios que se hayan agregado para el registro con el fin de que finalicen el proceso y generen sus credenciales.
5	El sistema listará todas las solicitudes registradas hasta la fecha.
6	El sistema será capaz de generar paginaciones de la información para un mayor orden de visualización.
7	El sistema será capaz de filtrar la información por diferentes parámetros, como fechas, identificadores, tipos de solicitud, por dependencias, entre otras.



- 8 El sistema permitirá registrar nuevas solicitudes a través de un formulario donde se solicite la información esencial de la solicitud.
  - 9 El sistema generará automáticamente el identificador de la solicitud, pero debe dar la posibilidad de editarlo.
  - 10 El sistema debe ser capaz de editar un archivo PDF y ponerle en la primera hoja un sello que identifique los datos de la solicitud.
  - 11 El sistema debe permitir asignar una prioridad preestablecida o asignar un tiempo límite de respuesta.
  - 12 El sistema debe permitir adjuntar archivos PDF a las solicitudes.
  - 13 El sistema debe permitir registrar el lugar donde se almacenarán los documentos físicos.
  - 14 El sistema debe dar la posibilidad de registrar si los documentos se entregaron en físico o solo se envió por medio digital.
  - 15 El sistema debe permitir asignar la solicitud a una o varias dependencias.
  - 16 El sistema informará por notificación y correo que se ha realizado una asignación a la dependencia.
  - 17 El sistema debe mostrar las solicitudes asignadas a un usuario de acuerdo con la dependencia a la que pertenece.
  - 18 El sistema debe darle prioridad a las solicitudes que estén próximas a llegar a su tiempo límite de respuesta.
  - 19 El sistema debe mostrar los detalles de alguna solicitud requerida.
  - 20 El sistema debe permitir reasignar la solicitud a otra dependencia en caso de ser requerida.
-

- 21 El sistema debe permitir agregar incidencias con mensajes y archivos adjuntos si se requieren pasos intermedios.
  - 22 El sistema debe permitir cerrar una solicitud agregando un mensaje de cierre y un archivo adjunto.
  - 23 El sistema debe generar una línea de tiempo desde que se registra la solicitud hasta que se finaliza.
  - 24 El sistema debe permitir el acceso a la información únicamente a las dependencias asignadas.
  - 25 El sistema debe ser capaz de mostrar los archivos adjuntos a las solicitudes.
  - 26 El sistema debe permitir descargar la información en los dispositivos que se soliciten y estén autorizados.
  - 27 El sistema debe permitir generar copias de seguridad de la base de datos.
  - 28 El sistema debe permitir restaurar una copia de seguridad en caso de corromperse la información.
  - 29 El sistema debe permitir descargar un archivo comprimido con todos los archivos que se han almacenado de las solicitudes.
  - 30 El sistema debe generar archivos de registro que almacenen las acciones del sistema en general, el acceso a un usuario, la solicitud de un archivo, etc.
  - 31 El sistema debe permitir elegir el acceso que tendrá cada tipo de usuario.
  - 32 El sistema debe tener una interfaz de usuario que cumpla con los estándares de accesibilidad.
  - 33 El sistema debe ser capaz de realizar automáticamente un mantenimiento para eliminar archivos temporales y caché.
-

- 34 El sistema debe mostrar estadísticas generales de todas las solicitudes registradas.

---

*Nota.* Se definen las funciones que deberá realizar el sistema

### **Requerimientos no Funcionales**

De acuerdo con las historias de usuario se definen los siguientes requisitos no funcionales que determinaran algunas de las limitaciones que debe afrontar el sistema de correspondencia:

#### **Tabla 4**

*Definición de requerimientos no funcionales*

No.	Requerimiento no funcional
1	El sistema debe ser capaz de soportar más de 100 conexiones simultaneas.
2	El sistema debe responder al usuario en menos de 5 segundos.
3	El sistema debe soportar la carga y descarga de información de manera simultánea con al menos 20 usuarios.
4	Todos los cambios de la información se deben ver reflejados en todos los usuarios en menos de 3 segundos.
5	La solicitud para crear un nuevo usuario solo se puede generar por un usuario que tenga acceso a la gestión de usuarios.
6	La adición y eliminación de permisos de un usuario solo puede ser realizada por un usuario con los suficientes permisos para su gestión.
7	El sistema se debe realizar aplicando patrones de diseño modernos que ayuden a incrementar la seguridad en general.
8	La copia de seguridad de los datos se realiza cada día a las 6 de la tarde cuando ya no haya usuarios conectados.

---

- 9 Las conexiones externas con otras aplicaciones se deben realizar de manera segura aplicando mecanismos de encriptación.
- 10 Si se detectan irregularidades con las solicitudes automáticamente el sistema bloqueará a los usuarios involucrados o en definitiva dejará de recibir y enviar información para proteger la información.
- 11 El sistema solo aceptará solicitudes de dispositivos con una IP conocida.
- 12 El tiempo para aprender a utilizar las funciones del sistema debe ser menor a 4 horas.
- 13 El servicio REST debe contar con una documentación que explique que realiza cada URL, los parámetros que recibe, la autenticación que se aplica y la respuesta que se obtendrá.
- 14 El servicio WEB debe contar con una interfaz de usuario que cumpla con las normativas de accesibilidad.
- 15 El servicio WEB debe contar con tutoriales y explicaciones de lo que puede realizar cada pantalla, cada botón, etc.
- 16 El sistema debe ser capaz de adaptarse a diferentes tamaños de monitores.
- 17 La probabilidad de que falle el sistema debe ser menor al 5% para mantener elevado el tiempo de operatividad.
- 18 El sistema debe consumir menos de 400mb de memoria RAM.
- 19 Se comprueba el estado del sistema con ayuda de herramientas de automatización de pruebas como Cypress y Postman.
- 20 El sistema no revelará información a usuarios no autorizados.
- 21 Los archivos deben ser almacenados bajo los lineamientos establecidos para el archivo digital de información.

---

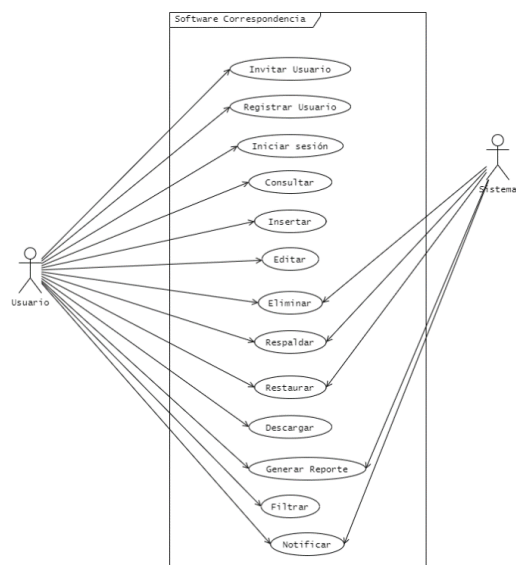
*Nota.* Definición de los requerimientos no funcionales que determinan limitaciones para afrontar.

## Diagramas de análisis

### Diagrama de casos de uso

**Figura 7**

*Diagrama de casos de uso entre usuario y sistema*



*Fuente.* Elaboración propia.

Se describen de manera general cuáles serán las acciones que podrá realizar un usuario y el mismo sistema, esto con el fin de ir modularizando el sistema entre diferentes secciones que harán más fácil su entendimiento y desarrollo, estos casos de uso son:

**Tabla 5**

*Detalles de casos de uso*

No.	Caso de uso	Detalles
1	Invitar usuario	Un usuario registrado con los permisos necesarios tendrá la capacidad de registrar un correo nuevo para invitarlo a realizar un registro en el sistema, en este registro del correo se define a que dependencia

corresponde y cuáles serán los permisos que tendrá disponibles. (Ver Figura 8)

- 2 Registrar usuario Un usuario que haya sido invitado podrá acceder al enlace de registro para ingresar sus datos y credenciales de acceso que podrá utilizar para ingresar al sistema. (Ver Figura 9)
  - 3 Iniciar sesión Los usuarios que estén ya registrados podrán acceder usando las credenciales que se registraron con anterioridad, si se pasan las validaciones tendrá acceso al dashboard de administración. (Ver Figura 10)
  - 4 Consultar Los usuarios con acceso podrán consultar toda la información a la que se les haya autorizado, dentro de estos pueden estar registros, documentos, imágenes y demás elementos que se almacenen en el sistema. (Ver Figura 11)
  - 5 Insertar Los usuarios con acceso podrán insertar datos al sistema dependiendo de los permisos y el módulo donde lo desee, los usuarios pueden agregar registros, imágenes, documentos, incidencias, entre otros elementos que se alojen en el sistema. (Ver Figura 12)
  - 6 Editar Los usuarios con acceso podrán editar los registros que estén a su cargo, la información nuevamente pasará bajo validación antes de que se registre. (Ver Figura 13)
  - 7 Eliminar Los usuarios con acceso podrán realizar el borrado lógico de su información, es decir que esta solo dejará de ser visible, pero se
-

- mantendrá hasta que el usuario decida borrar el registro definitivamente. (Ver Figura 14)
- 8 Respaldar Un usuario con los permisos podrá realizar copias de seguridad de la base de datos en general, descargarlas y almacenarlas como desee con el fin de mantener a salvo la información. (Ver Figura 15)
- 9 Restaurar Un usuario con permisos podrá restaurar una copia de seguridad antigua en caso de que la información se corrompa o sufra algún daño, dando la posibilidad de mantener el sistema funcionando. (Ver Figura 16)
- 10 Descargar Los usuarios con permisos podrán descargar la información que esté a su cargo, ya sea un solo registro, un archivo, varios archivos en otro comprimido, etc. (Ver Figura 17)
- 11 Generar reporte Los usuarios ya registrados y que hayan trabajado en el sistema tendrán la posibilidad de generar reportes en base a diferentes parámetros. (Ver Figura 18)
- 12 Filtrar Todos los usuarios registrados tienen la posibilidad de filtrar la información en base a parámetros como fechas, códigos, nombres y demás parámetros que se consideren necesarios. (Ver Figura 19)
- 13 Notificar Cuando los usuarios realicen modificaciones o asignaciones se enviarán alertas o correos a los usuarios que afecten estos cambios con el fin de que estén al tanto. (Ver Figura 20)

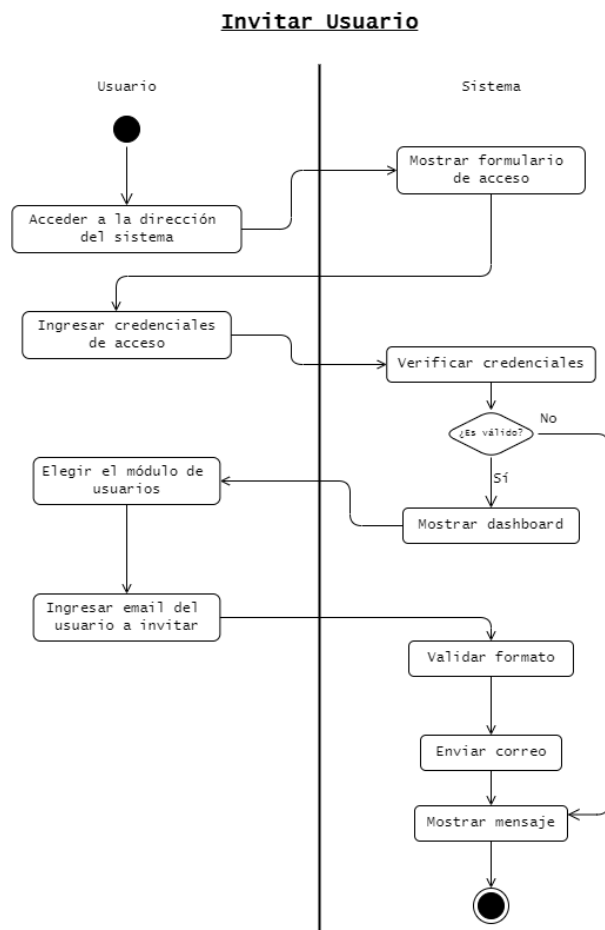
---

*Nota.* Se definen los casos de uso generales para el software.

## Diagrama de Actividades

**Figura 8**

*Diagrama de actividades del caso de uso invitar usuario*



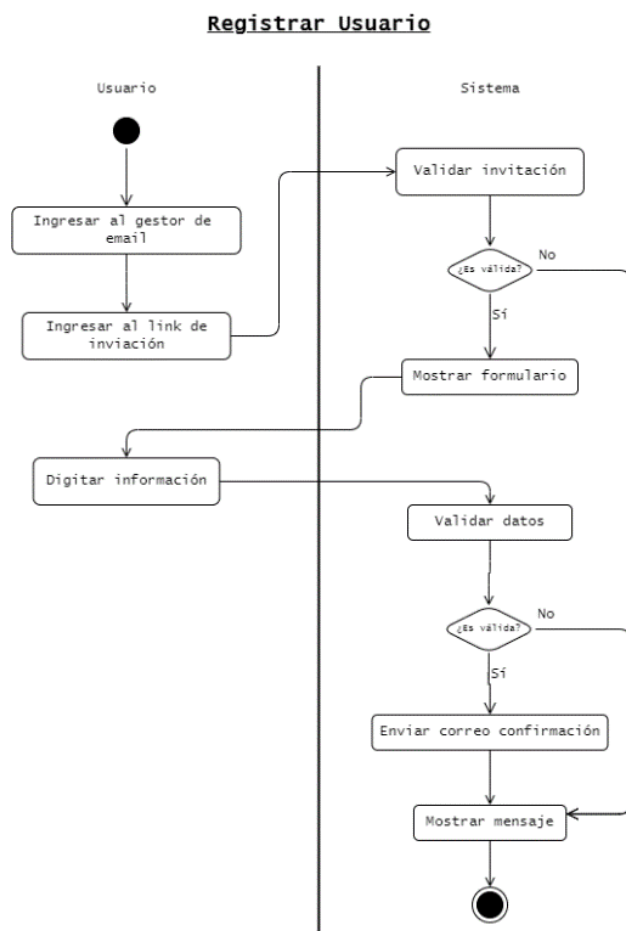
*Fuente.* Elaboración propia con Draw.io

Se establecen las acciones necesarias para llevar a cabo la invitación de un usuario para que se registre dentro del sistema de Filesoft, estas son acciones tomadas de manera general por lo que su implementación además del correo electrónico se deben establecer datos base del funcionario como su nombre, la dependencia en la que estará y los permisos que tendrá dentro del sistema.



**Figura 9**

*Diagrama de actividades del caso de uso registrar usuario*

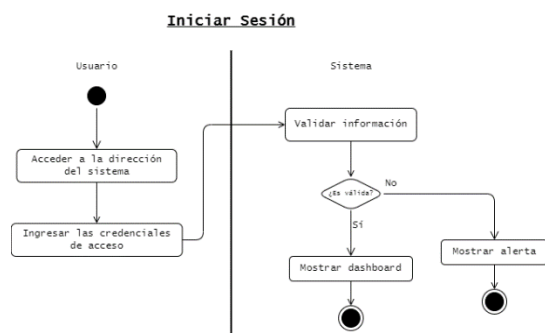


*Fuente.* Elaboración propia en Draw.io

Como se puede observar en las actividades este únicamente se logrará a través de invitación para preservar la seguridad del sistema y que el administrador pueda controlar a detalle los diferentes usuarios que estén en la aplicación, en esta actividad el funcionario ya ingresa más información y se valida que su invitación sea correcta para realizar el registro de manera definitiva.

**Figura 10**

*Diagrama de actividades del caso de uso iniciar sesión*

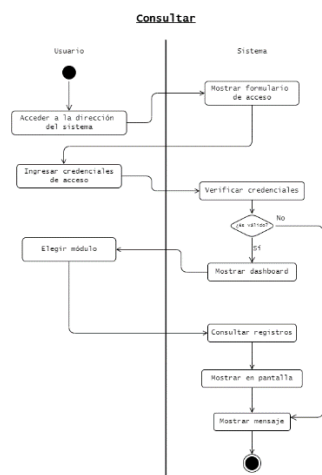


*Fuente.* Elaboración propia en Draw.io

Esta la actividad esencial para acceder al sistema, en esta tras el registro del funcionario, se podrá acceder al dashboard con el correo electrónico y la contraseña asignada durante el registro, esto se valida y si es correcto se da el acceso o si no se informa del error sucedido.

**Figura 11**

*Diagrama de actividades del caso de uso consultar*

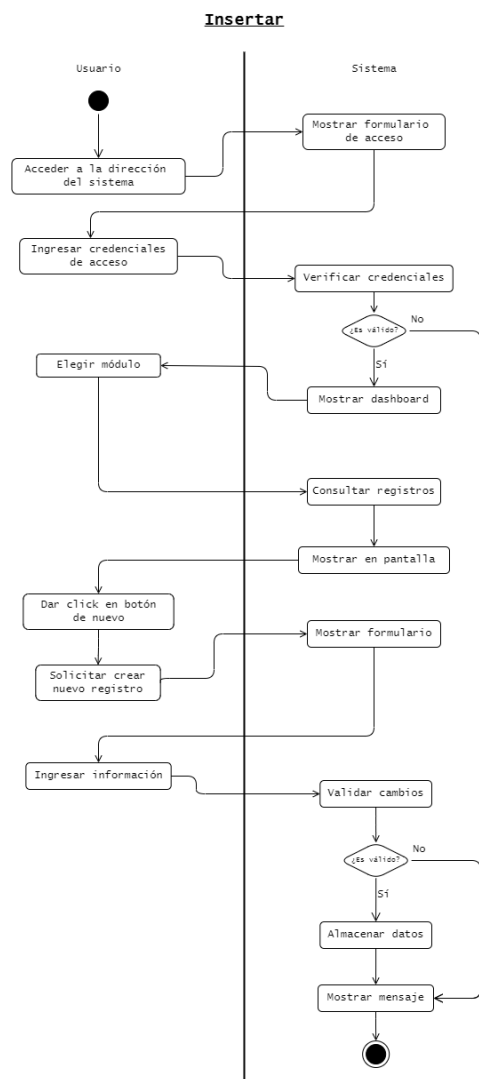


*Fuente.* Elaboración propia en Draw.io

Esta es una de las actividades que más se lleva a cabo dentro de la aplicación debido a que se está en constante consulta y lectura de los datos almacenados para cada uno de los módulos como los radicados, ciudadanos, entidades, etc.

Figura 12

Diagrama de actividades del caso de uso insertar

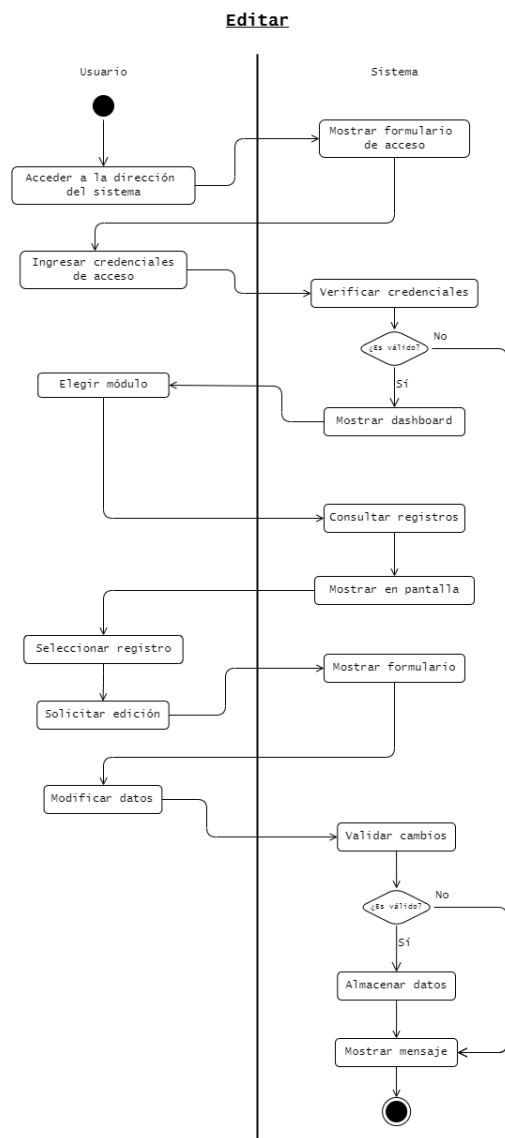


Fuente. Elaboración propia en Draw.io

La anterior figura muestra el paso a paso para insertar un registro dentro de la aplicación y este no solo aplica a los radicados, ya que para los otros módulos se realiza de la misma manera ingresando, realizando validaciones y por último registrando en la base de datos para su posterior lectura por los funcionarios.

Figura 13

Diagrama de actividades del caso de uso editar

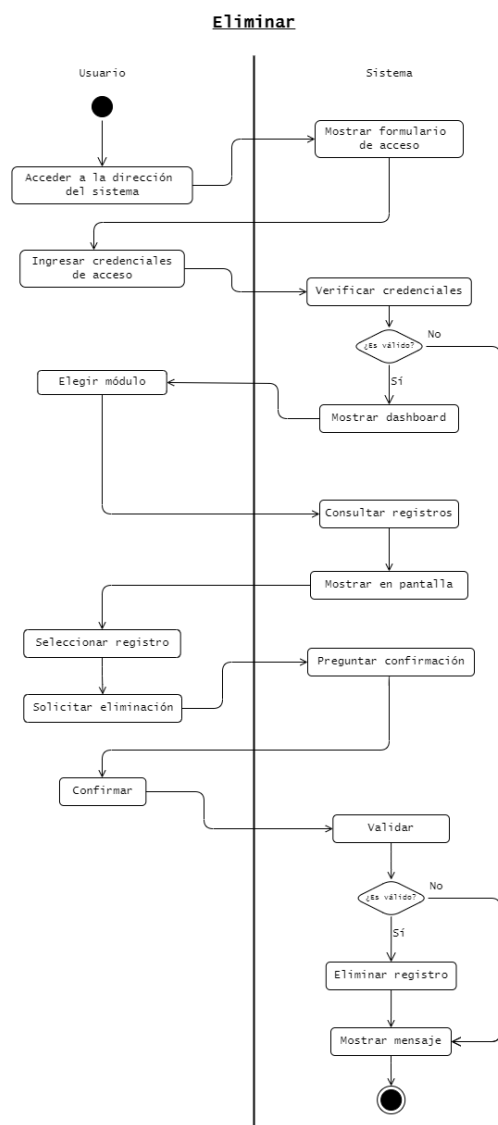


Fuente. Elaboración propia en Draw.io

Esta actividad es una de las que contiene más pasos debido a que además de realizar las validaciones de acceso, se deben realizar consultas para obtener el registro que se desea editar, para posteriormente ser manipulado por el usuario que edita los datos que crea necesarios y vuelve a enviar estos datos para su actualización en base de datos.

Figura 14

Diagrama de actividades del caso de uso eliminar

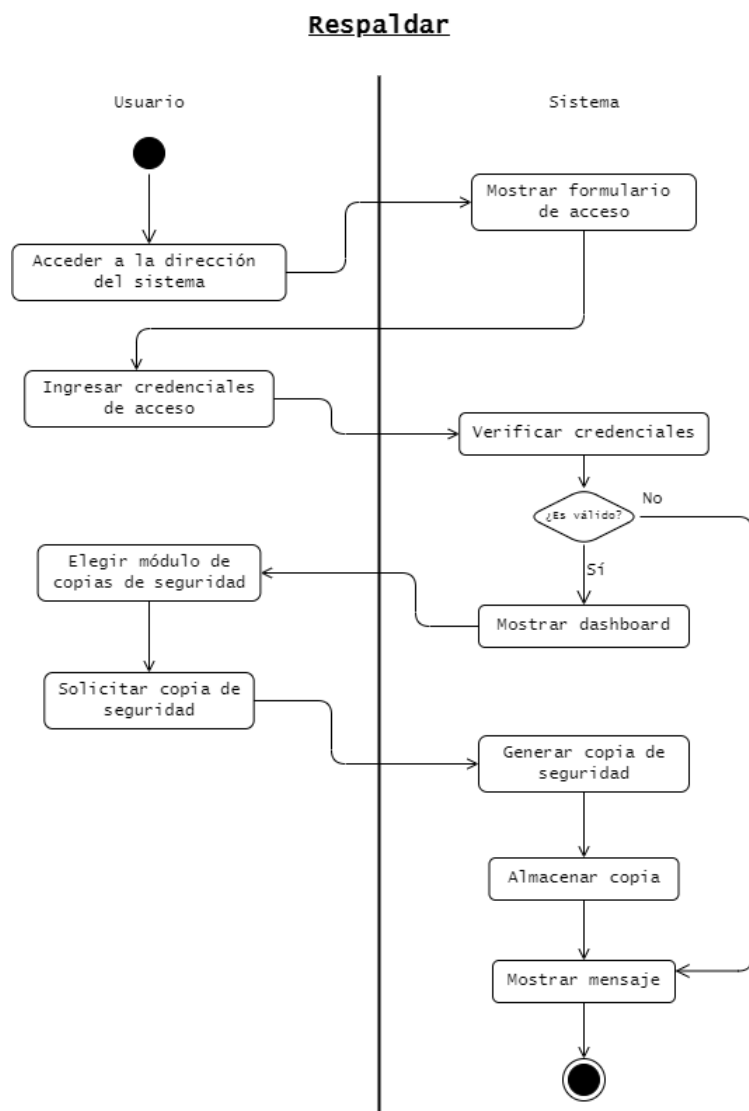


Fuente. Elaboración propia en Draw.io

Esta actividad es aplicada a todos los módulos y solo requiere el identificador del registro que se desea eliminar, lo que amplía los pasos son las validaciones y confirmaciones que debe realizar el usuario ya que es un proceso que pone en riesgo la información almacenada y no se debe permitir eliminar un registro de manera accidental.

Figura 15

Diagrama de actividades del caso de uso respaldar

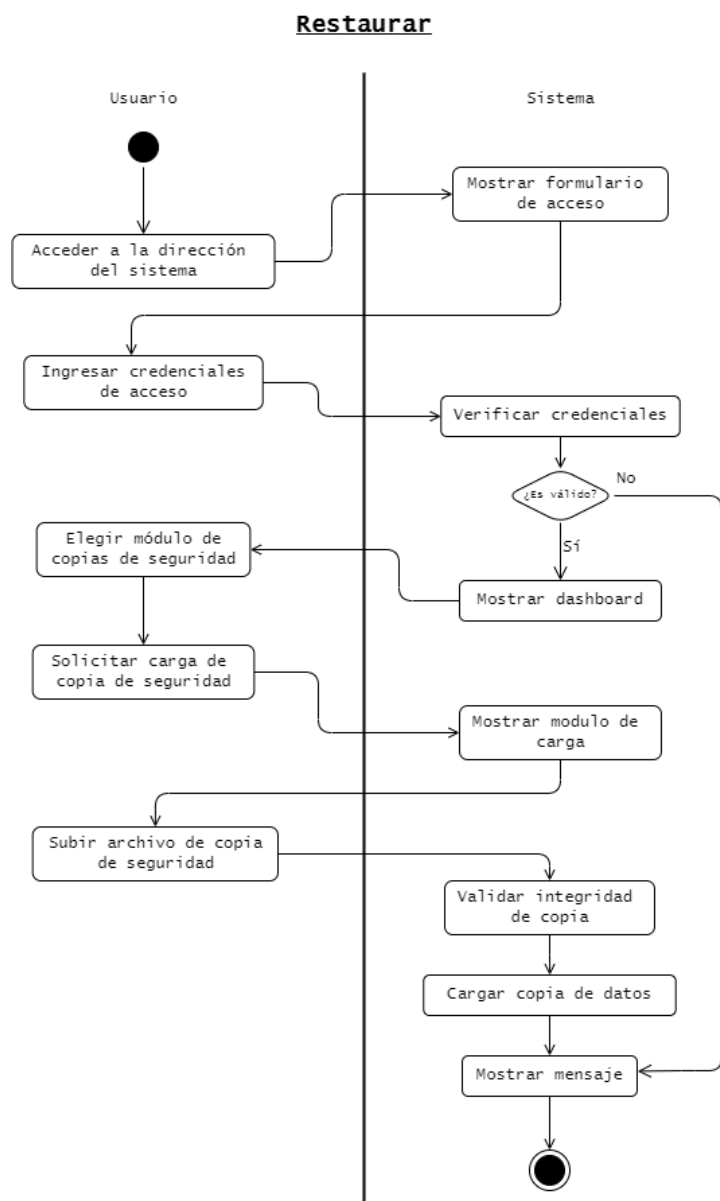


Fuente. Elaboración propia en Draw.io

Esta actividad es muy importante porque permite tener un respaldo de la información de la base de datos y de los archivos asociados a usuarios, radicados y sus respectivas incidencias, así entregando un archivo que el administrados o usuario que realizó la solicitud almacenará en un dispositivo de confianza.

Figura 16

Diagrama de actividades del caso de uso restaurar

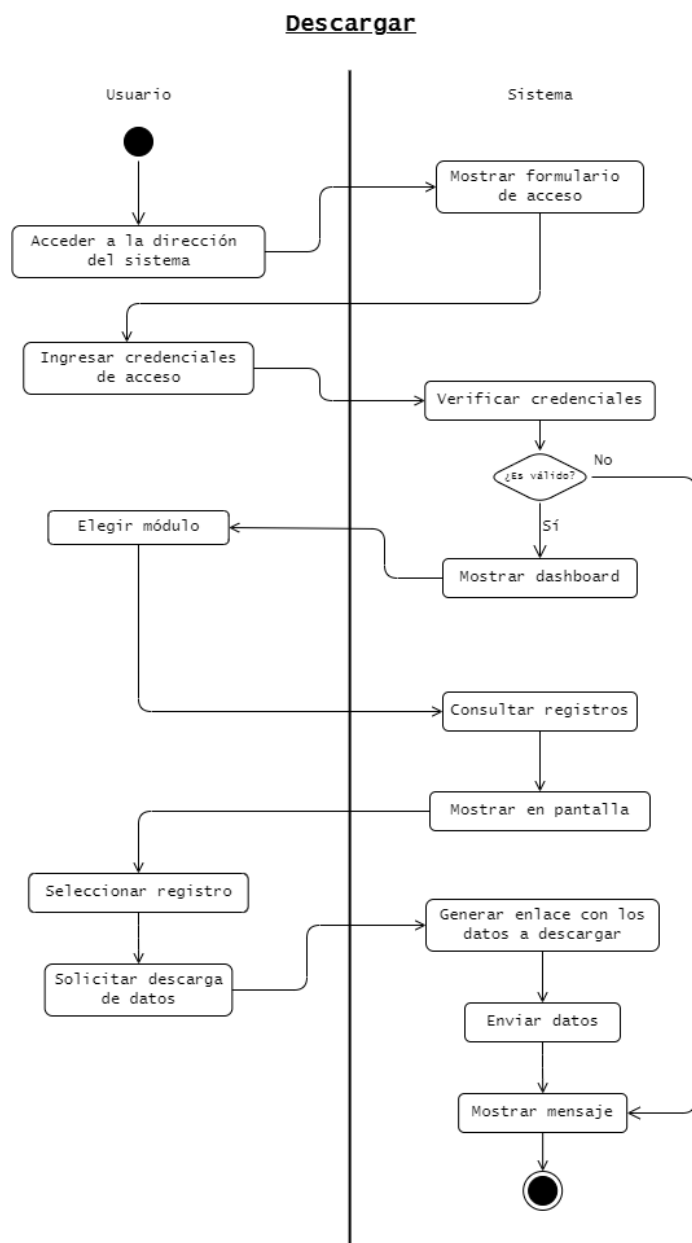


Fuente. Elaboración propia en Draw.io

Esta actividad está relacionada con la anterior, ya que se debe cargar el archivo de copia de seguridad generada y automáticamente se cargarán los registros y archivos dentro de la aplicación, así permitiendo volver a un anterior estado de la aplicación en caso de que la información haya sido corrompida de alguna manera.

Figura 17

Diagrama de actividades de caso de uso descargar



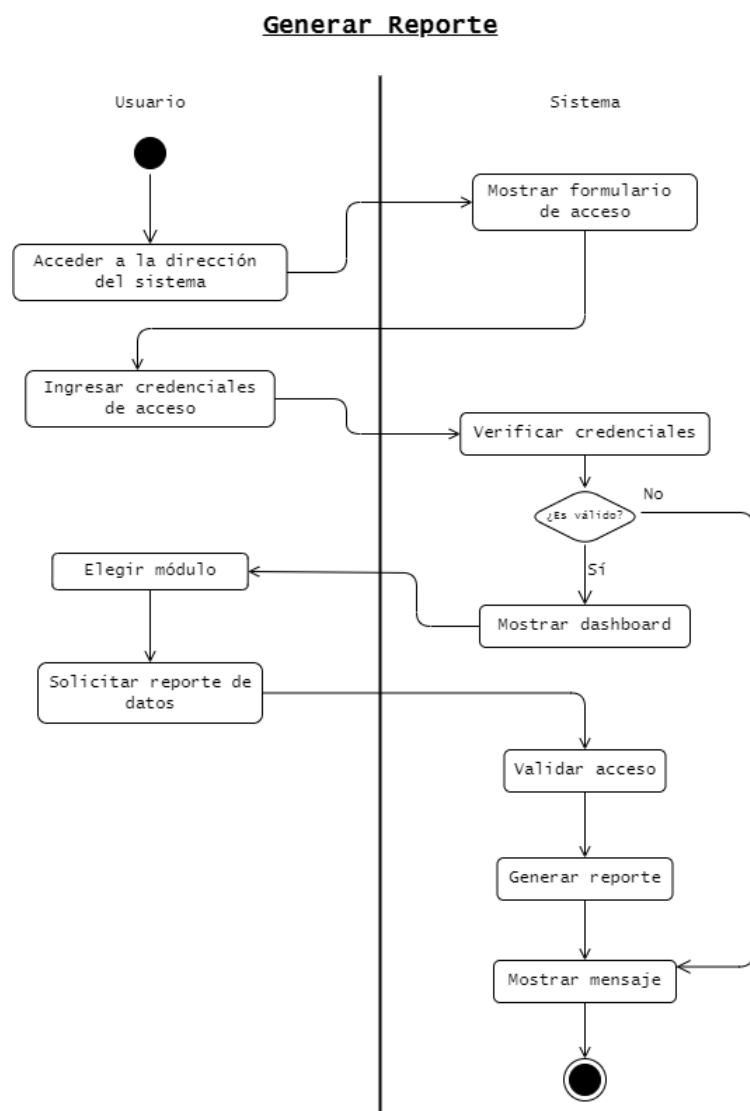
Fuente. Elaboración propia en Draw.io

Esta actividad define los pasos que se llevarán a cabo para descargar algún archivo asociado a los registros de la aplicación, estos pueden ser archivos PDF, documentos de Office u otro archivo que se haya guardado anteriormente.



Figura 18

Diagrama de actividades del caso de uso generar reporte

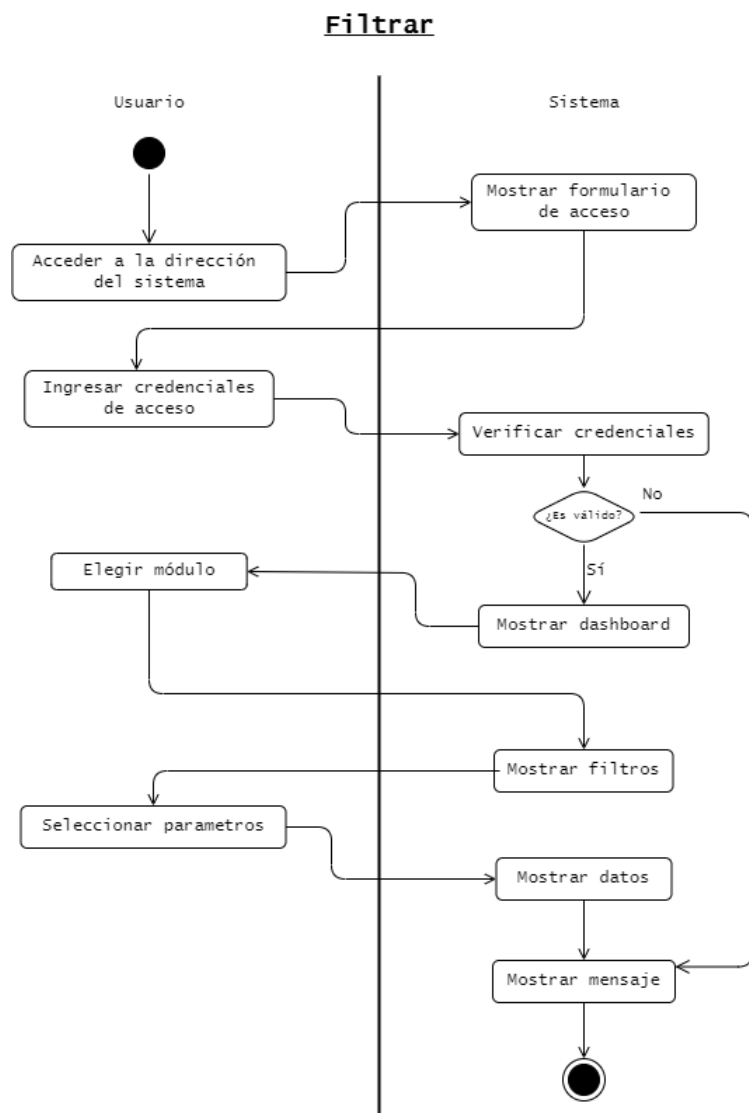


Fuente. Elaboración propia en Draw.io

Esta actividad está relacionada con la de lectura debido a que se deben obtener los datos que se mostrarán en el informe, además que cuando se realiza la solicitud se pueden definir parámetros para limitar la información que se colocará dentro del informe.

Figura 19

Diagrama de actividades del caso de uso filtrar

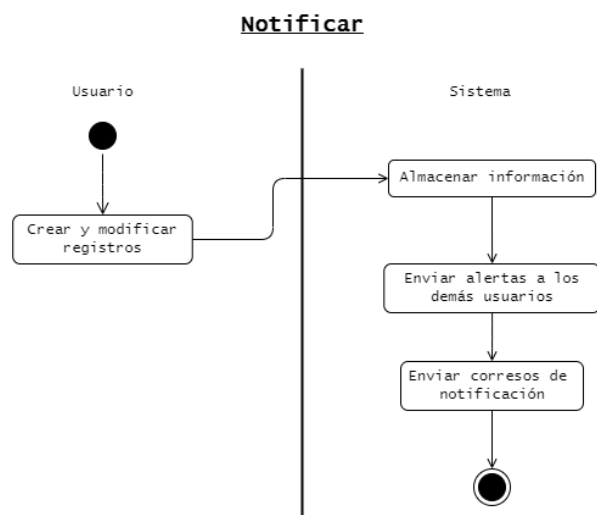


Fuente. Elaboración propia en Draw.io

Una actividad relacionada con la lectura, ya que obtiene datos almacenados, pero realizando limitaciones con algunos parámetros establecidos como fechas, códigos, nombres y demás elementos que ayuden a identificar un registro, además es algo esencial para encontrar de manera ágil algún registro.

**Figura 20**

*Diagrama de actividades del caso de uso notificar*



*Fuente.* Elaboración propia en Draw.io

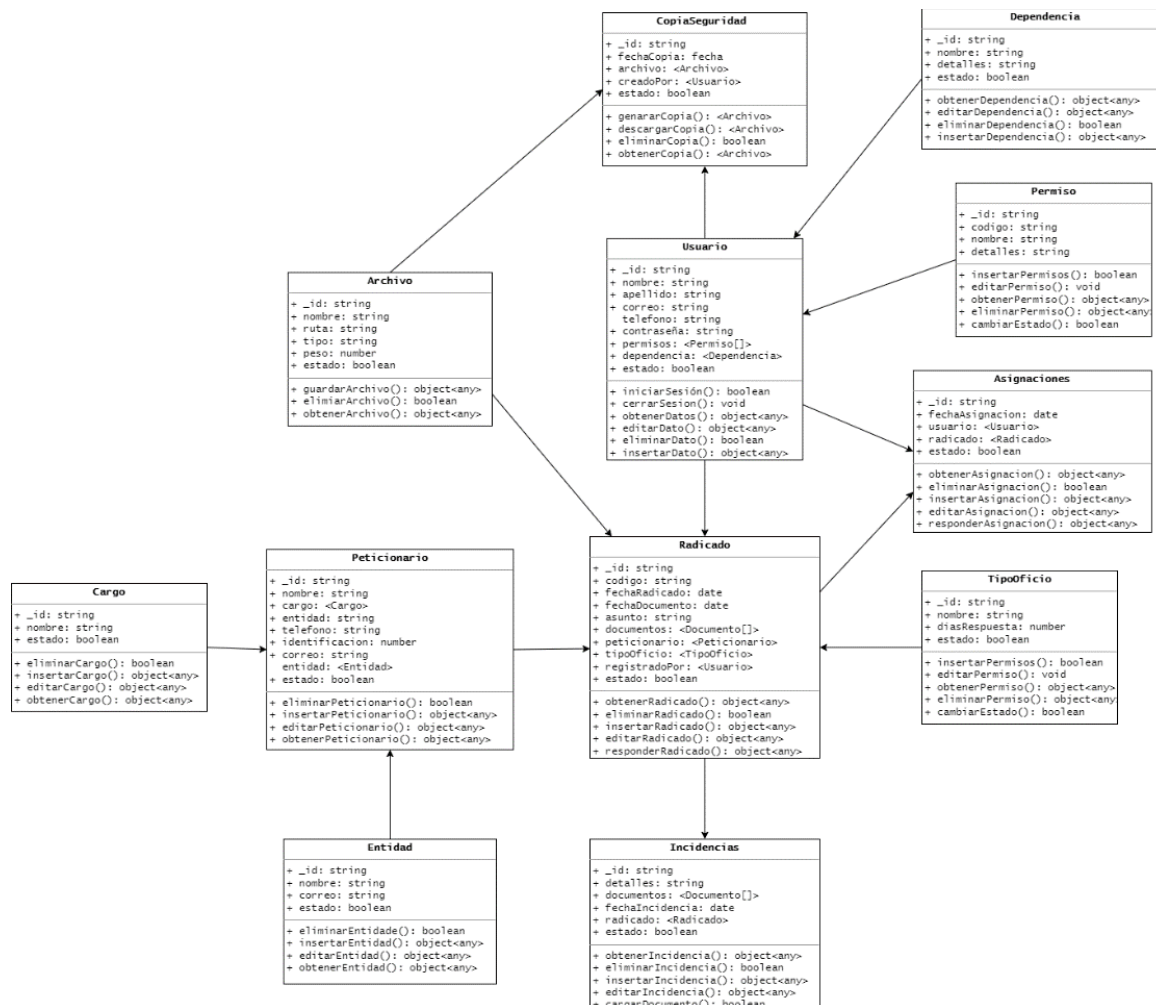
Esta actividad, aunque es sencilla, es importante para que todos los usuarios asociados a un registro se mantengan informados de los cambios que se realicen, y puedan notificar a tiempo alguna inconsistencia.

## Diagrama de clases

El objetivo del diagrama de clases es mostrar de manera sencilla los atributos y métodos con los que contarán los módulos del sistema.

**Figura 21**

*Diagrama UML de clases*



*Fuente.* Elaboración propia.

Esta define las principales clases que funcionan como abstracción para el modelo general de bases de datos, además de determinar las acciones que permitirá cada una de ellas, en la siguiente tabla se explica la función de cada clase:

**Tabla 6***Detalle de clases*

No.	Clase	Detalles
1	Usuario	Representa los usuarios que estarán registrados en el sistema y los cuales tendrán acceso a las diferentes secciones de acuerdo con los permisos que se le sean asignados.
2	Permiso	Representa los permisos que serán accesibles dentro del sistema de manera general.
3	Dependencia	Se refiere a las diferentes áreas de la entidad a donde se podrán asignar los radicados que se registren.
4	Radicado	Registra las solicitudes de los ciudadanos las cuales serán asignadas a las dependencias correspondientes para su solución y respuesta.
5	Asignación	Almacena todas las asignaciones y reasignaciones que se realicen de un radicado con el fin de dar el acceso únicamente a los usuarios designados.
6	Archivo	Representa lógicamente la información de un archivo digital almacenado y perteneciente a un radicado, esta información ayudará a ubicarlo y mostrarlo ante un usuario para su lectura o descarga.
7	Peticionario	Representa la información de la persona que realizó la solicitud, con el fin de poder saber a quién pertenece una solicitud y saber su información de contacto.

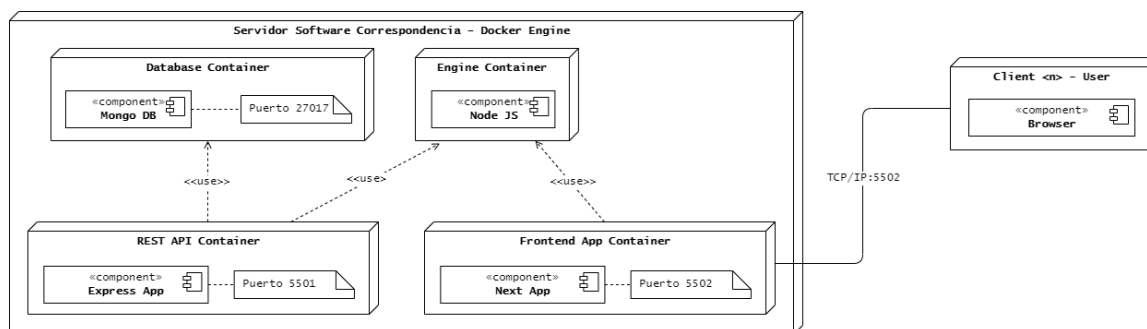
8	Cargo	Se refiere a los diferentes cargos que pueden tener los peticionarios, se representa como clase independiente debido a que tiene sus propios atributos independientes.
9	Entidad	Representa la información de una entidad, por lo general a la que pertenece un peticionario.
10	TipoOficio	Representa el tipo de oficio que se le asigna a un radicado, con el fin de determinar los plazos en los que se debe responder.
11	Incidencia	Representa los datos de las diferentes incidencias que se puedan generar de un radicado.
12	CopiaSeguridad	Representa la información de las copias de seguridad que se hayan generado, estas indican donde se pueden encontrar a nivel de archivo digital para su descarga.

---

*Nota.* Se describen cada una de las clases abstractas con sus atributos y métodos.

### **Diagrama de despliegue**

El objetivo del siguiente diagrama es mostrar cuáles serán las aplicaciones que se utilizarán dentro del entorno de ejecución, además de especificar los puertos lógicos que utilizará dentro del servidor:

**Figura 22***Diagrama de servicios y despliegue*

*Fuente.* Elaboración propia.

**Database Container:** Contenedor encargado de gestionar la base de datos para la cual se seleccionó Mongo DB el cual ofrece una imagen para su ejecución que se realiza en el puerto 27017.

**Engine Container:** Contenedor que ejecutará el motor de procesamiento, en nuestro caso es Node JS el cual permite mantener arriba los servicios necesarios para el software.

**REST API Container:** Contenedor encargado de gestionar y mantener el servicio REST que se encarga de todo el modelo lógico del software de manera general, esta utiliza una aplicación de Express JS que responde bajo los verbos HTTP para dar una respuesta.

**Frontend App Container:** Contenedor que ejecuta la aplicación encargada de mostrar la interfaz de usuario, esta se desarrolla bajo el framework de Next JS que permite su ejecución en el servidor de manera rápida y segura.

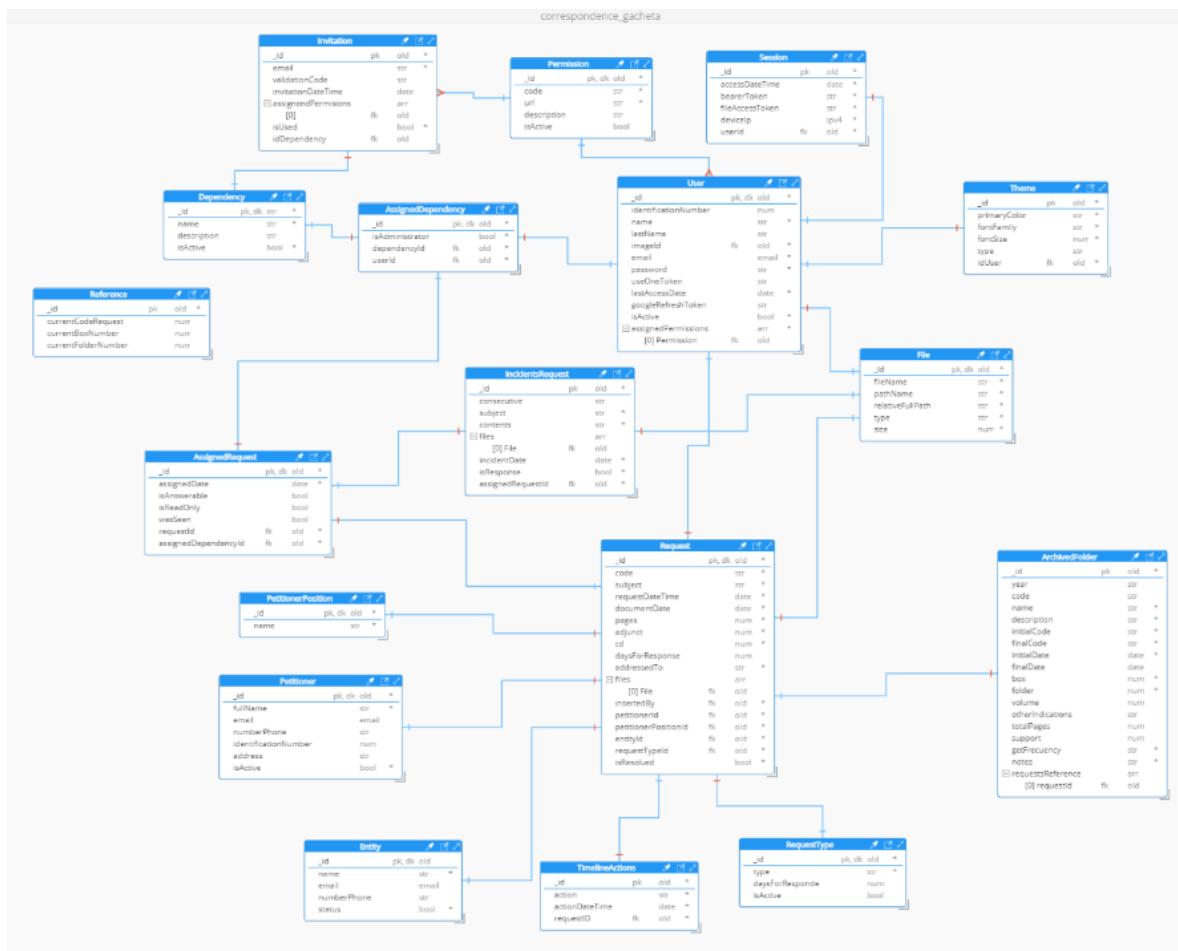
**Client n:** Hace referencia a los diferentes clientes que podrán usar la interfaz de usuario, estos se conectarán desde un navegador a través del puerto 5502 que expone la aplicación de Next JS.

## Diagrama modelo de base de datos

El modelo de base de datos se realizó con la aplicación de Hackolade que permite realizar modelos para bases de datos tipo no relacional, en especial esquemas de Mongo DB.

**Figura 23**

*Modelo de base de datos MongoDB*



*Fuente.* Generado con el software Hackolade.

La documentación general de la base de datos representa cada uno de los modelos, su tipo de dato y las restricciones entre colecciones, es generada por el software Hackolade, después del modelado realizado.



## Diseño y prototipado

Al tener todo el análisis de requerimientos y qué información estará involucrada en el software de correspondencia es posible comenzar con la planeación del proyecto, definiendo el nombre del software, colores, iconos, imágenes, fuentes y demás elementos que serán parte del resultado final.

### Definición del nombre de la aplicación

Después de ver palabras relacionadas con el objetivo de la aplicación se usan dos en especial: Archivo y Software. Estas se traducen a inglés y se combinan para generar el nombre de **FILESOFT**, siendo este el nombre a utilizar para referirnos al aplicativo de correspondencia dentro de los diseños.

### Diseño del logotipo

El logotipo de la aplicación es realizado usando la tipografía de Montserrat, incluyendo dos iconos que además de reemplazar las letras, tienen un significado enfocado al objetivo de la aplicación, como lo es el lápiz y el documento, y por último se realizan algunos retoques quedando el logotipo como se muestra en la siguiente figura.

#### Figura 24

*Logotipo de la aplicación*



*Fuente.* Diseño propio en la aplicación de Figma.

### **Diseño del Favicon (web)**

Este se diseña usando de base lo mencionado para el logotipo, únicamente se realiza un cambio en el ordenamiento de las letras para generar un icono más compacto que funcione como identificador en el diseño web, en especial cuando se muestre en el navegador al generar una pestaña de la aplicación, dando como resultado el icono que se muestra en la siguiente figura.

#### **Figura 25**

*Icono identificador de web*



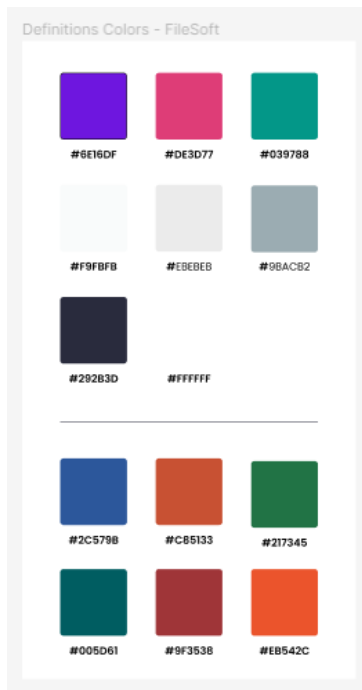
*Fuente.* Diseño propio en la aplicación de Figma.

### **Elección de colores**

Se hace un análisis del entorno de aplicaciones que utilizan en la entidad para elegir una gama de colores que sean familiares, sencillos y agradables a la vista de los usuarios, que aplicados sobre la interfaz de usuario combinen y cumplan con los estándares de accesibilidad, permitiendo una legibilidad alta en la información que se muestre en la aplicación.

## Figura 26

### *Colores seleccionados*



*Fuente.* Diseño propio en la aplicación de Figma.

### **Diseño de interfaces de usuario**

Se representan detalladamente como serán las interfaces de usuario, la información que contendrán, los colores seleccionados y la distribución de los elementos en sí, el objetivo es mostrar una interfaz amigable, intuitiva y fácil de utilizar por parte de los usuarios que hagan uso de la aplicación.

Figura 27

Página de inicio de filesoft

**FILESOFT** Inicio Consultar radicados Documentación Iniciar sesión

## Organiza tu correspondencia en un solo lugar

¡Hola y bienvenido a nuestro software de correspondencia! Te ayudaremos a administrar y organizar toda tu correspondencia de manera eficiente, para que nunca te sientas abrumado o desorganizado.

**Gestión de documentos**  
Puedes permitir que los usuarios suban, almacenen y organicen sus documentos de correspondencia en el software.

**Notificación y recordatorio**  
Envía recordatorios y notificaciones a los usuarios para que no se olviden ningún correo o mensaje importante.

**Gestión en equipo**  
Permite gestionar y coordinar en tu redacción de manera sencilla y eficiente.

**Recepción de mensajes**  
Envío y recepción de mensajes mediante función de correo electrónico integrado.

### Servicios

## Gestión de correspondencia eficiente

En nuestra sección de servicios, ofrecemos una amplia variedad de herramientas para ayudarte a gestionar tu correspondencia de manera eficiente. Desde la subida y almacenamiento de documentos, hasta el envío y recepción de mensajes y la colaboración en equipo, nuestros servicios te ayudarán a mantener tu correspondencia organizada y accesible en todo momento.

[Ver documentación](#)

### Proceso

## Flujo de trabajo

**Recepción**  
Se recibe y se registra la correspondencia que entra al sistema.

**Asignación**  
El radicado es asignado al área correspondiente de la redacción como destino.

**Gestión**  
El área encargada recibe todo lo correspondiente para dar solución al radicado.

**Respuesta**  
Se informa que el radicado generado ya tiene una respuesta.

### Consulta

## Seguimiento en tiempo real de tus radicados

Con nuestra función de seguimiento y consulta, puedes mantener al tanto del progreso de tus radicados en cualquier momento. Nuestro timeline te muestra las actualizaciones más recientes de cada radicado, para que sepas exactamente en qué etapa se encuentra cada uno. ¡Nunca más tendrás que preocuparte por perder el seguimiento de tus radicados!

[Consultar radicados](#)

**FILESOFT** Inicio Consultar radicados Documentación Política de privacidad Términos de uso Acerca de la aplicación

Todos los derechos reservados. Copyright 2022 Filesoft S.A.S. Ninguna parte de este documento puede ser reproducida, distribuida o transmitida en forma alguna o por cualquier medio electrónico o mecánico sin el consentimiento escrito del propietario de los derechos de autor.

Fuente. Elaboración propia en Figma.

La anterior imagen representa un resumen del aplicativo, es la primera pantalla que verá el usuario por esta razón es atractiva y tiene una distribución de elementos fáciles de recordar.

Figura 28

## Página de consulta de radicados

**FILESOFT** Inicio [Consultar radicado](#) Documentación [Iniciar sesión](#)

## Consultar radicado

Puedes ver fácilmente la fecha y el estado actual de cada uno de tus radicados. Además, nuestro timeline sencillo te muestra las actualizaciones más recientes de cada radicado, para que siempre estés al tanto de su progreso.

Q Número de radicado [Buscar](#)

**1 3 3 0** [Ver radicado en sistema](#)

**Fecha radicación:** 23 de enero de 2023, a las 12:23:00 horas  
**Estado:** [Con respuesta](#)

### Línea de tiempo

- Recepción en oficina de correspondencia**
  - Se generó el número de radicado 1330 en el sistema. (23 de enero de 2023, a las 12:23:00 horas)
  - Se cargaron los documentos adjuntos al radicado. (23 de enero de 2023, a las 12:30:00 horas)
- Asignación de radicado a dependencia destino**
  - Se envió el radicado a la dependencia de Gobierno. (23 de enero de 2023, a las 12:35:00 horas)
- Gestión del radicado**
  - Se ha visualizado el radicado por la dependencia de Gobierno. (25 de enero de 2023, a las 08:15:00 horas)
  - Se ha reasignado a un funcionario de la dependencia de Gobierno. (26 de enero de 2023, a las 09:32:03 horas)
  - Se ha creado una incidencia del radicado en el sistema. (29 de enero de 2023, a las 15:15:00 horas)
- Respuesta del radicado**
  - Se cargaron los documentos de respuesta al sistema. (30 de enero de 2023, a las 18:23:00 horas)
  - Se envió la respuesta a través de correo electrónico. (30 de enero de 2023, a las 18:30:00 horas)

[Visualizar respuesta](#)

**FILESOFT** Inicio [Consultar radicado](#) Documentación [Política de privacidad](#) [Términos de uso](#) [Acerca de la aplicación](#)

Todos los derechos reservados. Copyright 2023 Yeferson Zaque. Ninguna parte de este software puede ser reproducida, distribuida, o transmitida en forma alguna sin el permiso previo por escrito del propietario de los derechos de autor.

*Fuente.* Elaboración propia en Figma

La anterior imagen representa la pantalla de consultar radicado el cual permite al usuario hacer seguimiento a su solicitud por medio del número de radicado, mostrando información básica de su solicitud.

Figura 29

## Página de políticas de privacidad de filesoft

**FILESOFT** [Inicio](#) [Consultar notificado](#) [Documentación](#) [Iniciar sesión](#)

## Políticas de privacidad

A continuación se detallan las prácticas de recopilación, uso y protección de la información personal de los usuarios de nuestra aplicación de correspondencia. Lea atentamente para entender cómo manejamos su información personal y cómo puede controlar su uso.

### Recopilación de datos de funcionarios

En nuestra aplicación de correspondencia recopilamos información personal de funcionarios que utilizan nuestra plataforma con el propósito de garantizar el correcto funcionamiento de la misma, incluyendo los siguientes tipos de información: nombres, correos electrónicos, cargos, direcciones, teléfonos y otros datos necesarios para el uso y funcionamiento del servicio; esta información es esencial para garantizar la correcta gestión de correspondencia y el uso seguro de la aplicación.

Los datos recopilados de funcionarios son tratados de forma confidencial y solo se comparten con terceros cuando es estrictamente necesario y en los casos que lo exigen las regulaciones aplicables. Ofrecemos a los funcionarios la posibilidad de acceder, corregir o eliminar sus datos en cualquier momento.

La aplicación también mantiene registros de acceso y auditoría para garantizar la seguridad de la información y detectar cualquier posible vulneración. Aseguramos la encriptación de los métodos de seguridad física y lógica para proteger la información recopilada.

### Recopilación de datos de ciudadanos

Nuestra aplicación de correspondencia se preocupa por la privacidad de los ciudadanos, por eso recopilamos solo la información necesaria para ofrecerles un servicio de calidad. Los datos recopilados incluyen nombres, direcciones, números de teléfonos y direcciones, solo esto con el objetivo de poder gestionar correctamente la correspondencia y brindar una mejor experiencia al usuario.

La recopilación de datos se realiza con el previo consentimiento del ciudadano y se cumplen todas las regulaciones de privacidad de su país o ciudad. Los ciudadanos tienen siempre acceso a su información y pueden solicitar corrección o eliminación de la misma en cualquier momento.

Además, garantizamos la seguridad de la información recopilada mediante medidas físicas y digitales para evitar cualquier tipo de vulneración y no compartimos la información con terceros a no ser que el ciudadano lo solicite.

En resumen, brindamos un servicio de calidad y al mismo tiempo protegemos la privacidad de los ciudadanos.

### Seguridad de la aplicación

La seguridad de la información personal de nuestros usuarios es una prioridad para nosotros. Es por eso que implementamos medidas de seguridad físicas, digitales y de procesos para proteger la información recopilada en nuestra aplicación de correspondencia.

Nuestras medidas de seguridad incluyen la encriptación de los datos sensibles, autenticación de usuarios, y auditorías regulares de procesos para detectar y prevenir vulneraciones. Además, implementamos actualizaciones de tecnologías y sistemas para proteger los datos de nuestros usuarios.

En caso de una vulneración de seguridad, se tomarán medidas inmediatas para investigar el incidente, notificar a las autoridades y brindar una respuesta adecuada. Informaremos a los afectados de cualquier vulneración de seguridad si esto es requerido por las regulaciones de su país o ciudad.

Nuestros usuarios pueden ayudar a proteger sus datos personales mediante el uso de contraseñas seguras y manteniendo su información actualizada. También pueden ponerse en contacto con nosotros si tienen preguntas o inquietudes sobre la seguridad de su información personal.

En resumen, ponemos todo nuestro esfuerzo para garantizar la seguridad de la información personal de nuestros usuarios, pero también es importante que los usuarios tomen medidas para proteger su información personal.

### Gestión de correo

En nuestra aplicación de correspondencia, nos esforzamos por ofrecer un servicio eficiente de gestión de correo electrónico a nuestros funcionarios. Para ello, utilizamos las direcciones de correo electrónico proporcionadas por cada funcionario, para poder recibir y enviar correspondencia en nombre de este.

La información de los correos electrónicos es tratada con confidencialidad y solo será compartida con terceros en caso de ser necesario y con autorización previa del funcionario. También garantizamos la seguridad de los correos y su contenido mediante medidas de seguridad digital.

Los funcionarios tienen siempre acceso a su historial de envíos y puede optar por recibir o no notificaciones de correo, así como eliminar o corregir sus correos en caso de ser necesario.

Es importante mencionar que esta política se rige por las regulaciones y leyes de privacidad y seguridad de su país o ciudad. Además, garantizamos transparencia en el uso de la información y brindamos opciones para controlar la información personal.

### Acceso con los servicios de Google

Nuestra aplicación de correspondencia ofrece la opción de iniciar sesión mediante los servicios de Google, lo que permite a los usuarios acceder rápidamente a nuestra plataforma sin necesidad de recordar otro nombre de usuario o contraseña. Al hacerlo, se permite a nuestra aplicación acceder a la información básica de su perfil de Google.

Esta información se utilizará para mejorar la experiencia del usuario en nuestra plataforma, y para brindar un servicio más personalizado, todo incluido el uso de esta información para facilitar la gestión de correspondencia.

Sin embargo, es importante mencionar que al utilizar los servicios de Google, los usuarios están sujetos a sus políticas y términos de servicios de Google, así como siempre tienen la opción de desvincular su cuenta de Google de nuestra aplicación en cualquier momento y tienen siempre acceso a su información.

[Inicio](#) [Consultar notificado](#) [Documentación](#) [Política de privacidad](#) [Términos de uso](#) [Ayuda de la aplicación](#)

Todos los derechos reservados. Copyright 2024 Filesoft. Ninguna parte de este software puede ser reproducida, distribuida o transmitida en forma alguna sin el permiso previo por escrito del propietario de los derechos de autor.

Fuente. Elaboración propia en Figma

La anterior imagen representa la página de políticas de privacidad la cual informa al usuario sobre como se trata la información dentro de la aplicación y como lo hacen cada uno de los servicios implementados.

Figura 30

*Página de terminos y condiciones de uso de filesoft*

**FILESOFT** Inicio Consultar radicado Documentación [Iniciar sesión](#)

## Términos de uso

Al utilizar nuestra aplicación, aceptas cumplir con estos términos y cualquier otra regulación aplicable. Te invitamos a leer detenidamente estos términos antes de utilizar nuestra plataforma.

1. La aplicación de correspondencia solo puede ser utilizada por personas autorizadas. Es responsabilidad del usuario asegurar que tiene permiso para utilizar la aplicación.
2. Al utilizar nuestra aplicación, el usuario garantiza que la información suministrada es veraz, precisa y actual. El usuario será el único responsable de cualquier consecuencia resultante de proporcionar información falsa.
3. El usuario se compromete a no utilizar la aplicación de manera ilegal, fraudulenta o dañina, y a no reproducir, duplicar, copiar, vender, revender o explotar con fines comerciales cualquier aspecto de la aplicación.
4. La aplicación se proporciona "tal cual" y nosotros no garantizamos la disponibilidad, seguridad o rendimiento de la misma. El usuario asume todos los riesgos asociados con el uso de la aplicación.
5. Nos reservamos el derecho a modificar, suspender o discontinuar la aplicación o cualquier parte de ella sin previo aviso.
6. El usuario se compromete a no utilizar la aplicación para cualquier propósito inapropiado o ilegal, como el acoso, la difamación, el hostigamiento, la discriminación o la invasión de la privacidad de otras personas.
7. El usuario se compromete a no utilizar la aplicación para enviar correo no deseado, spam o cualquier otra forma de comunicación no solicitada.
8. El usuario se compromete a no utilizar la aplicación para la transmisión de virus informáticos, malware u otros programas maliciosos.
9. El usuario se compromete a cumplir con todas las leyes y regulaciones aplicables al uso de la aplicación, incluyendo leyes y regulaciones de privacidad y seguridad de la información.
10. Es importante mencionar que estos términos de uso son solo un ejemplo y deben adaptarse y actualizarse según las regulaciones y leyes aplicables a tu país o ciudad y la forma en la que tu aplicación esta desarrollada y su uso.

**Advertencia:** El incumplimiento de las políticas de uso de nuestra aplicación de correspondencia puede resultar en la terminación inmediata del acceso a la misma. Si se detecta cualquier uso ilegal, fraudulento o dañino de la aplicación, tomaremos medidas legales apropiadas. El usuario será responsable de cualquier daño causado a nosotros o a terceros como resultado del incumplimiento de estas políticas. Por favor, asegúrate de cumplir con todas las reglas y regulaciones establecidas en nuestras políticas de uso para garantizar un uso seguro y legítimo de la aplicación.

**FILESOFT** Inicio Consultar radicado Documentación Política de privacidad Términos de uso Acerca de la aplicación

Todos los derechos reservados. Copyright 2023 Yeferson Zaque. Ninguna parte de este software puede ser reproducida, distribuida, o transmitida en forma alguna sin el permiso previo por escrito del propietario de los derechos de autor.

*Fuente.* Elaboración propia en Figma.

La anterior imagen presenta al usuario la pantalla de términos de uso expresando los diferentes lineamientos establecidos para el uso de la aplicación, dando a conocer que el incumplimiento de estos puede tener sanciones a nivel del sistema e incluso con el sistema penal.

## Figura 31

### *Página acerca de filesoft*



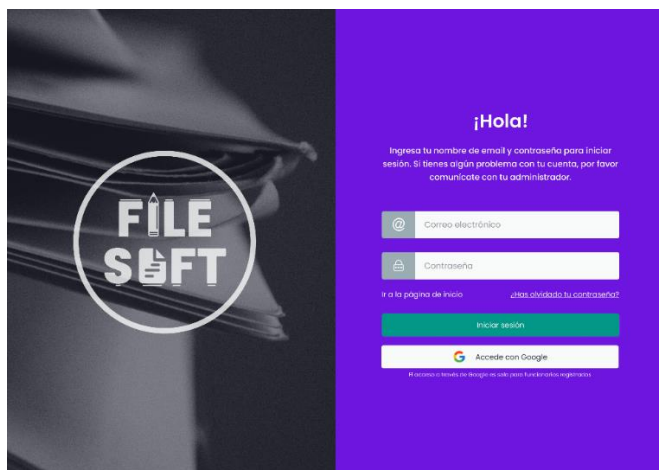
*Fuente.* Elaboración propia en Figma.

La anterior imagen presenta al usuario la información de los desarrolladores de la aplicación, los objetivos con que se construyó y de donde nace la idea del proyecto.



## Figura 32

### *Página de inicio de sesión a la aplicación*

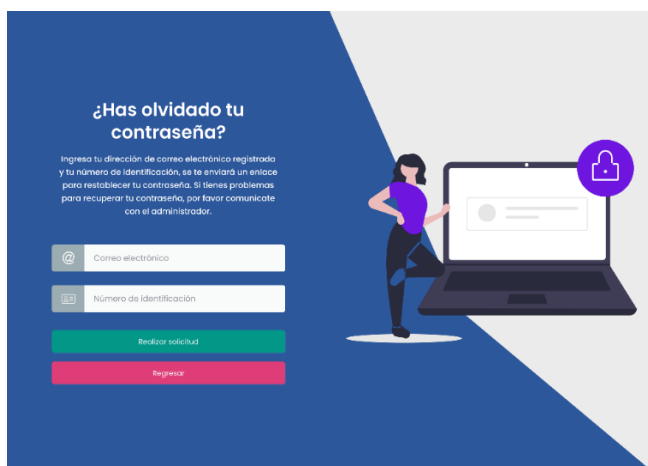


*Fuente.* Elaboración propia en Figma

La anterior imagen presenta al usuario el formulario para acceder al dashboard de administración con sus credenciales de acceso.

## Figura 33

### *Página de olvidé contraseña de acceso*

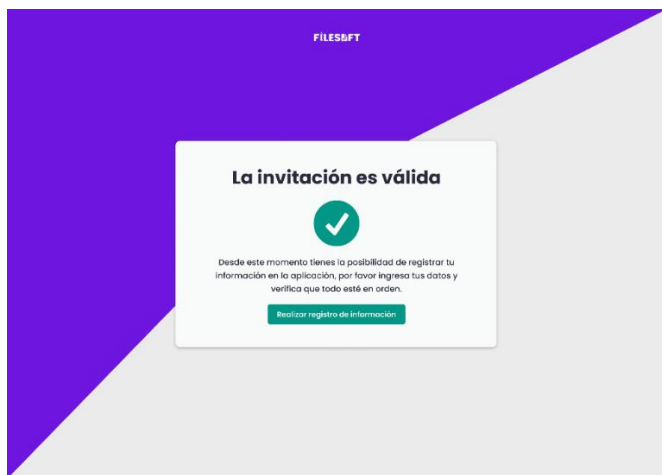


*Fuente.* Elaboración propia en Figma

La anterior imagen presenta al usuario la posibilidad de solicitar un cambio de contraseña en caso de que la haya olvidado y no tenga acceso al sistema.

## Figura 34

### *Página de validación de funcionarios invitados*



*Fuente.* Elaboración propia en Figma

La anterior imagen presenta al usuario una ventana con la información de la validación de invitación de usuario, informándole si es válida, errónea o no existe.

## Figura 35

### *Página de validación de correo electrónico de funcionario*

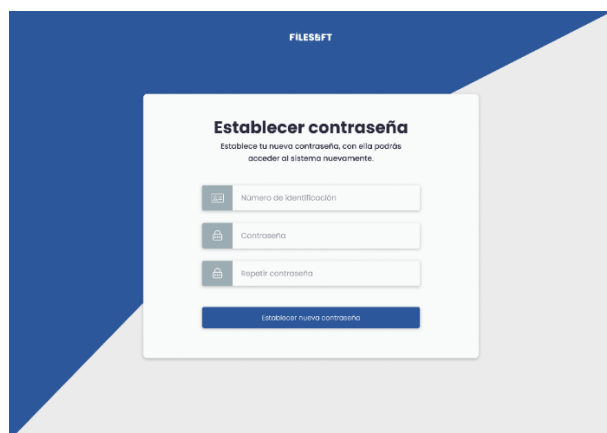


*Fuente.* Elaboración propia en Figma

La anterior imagen presenta al usuario la información de la validación de su correo registrado, valida que sea correcto o en caso contrario le informará el error.

### Figura 36

*Página de establecer nueva contraseña de acceso*

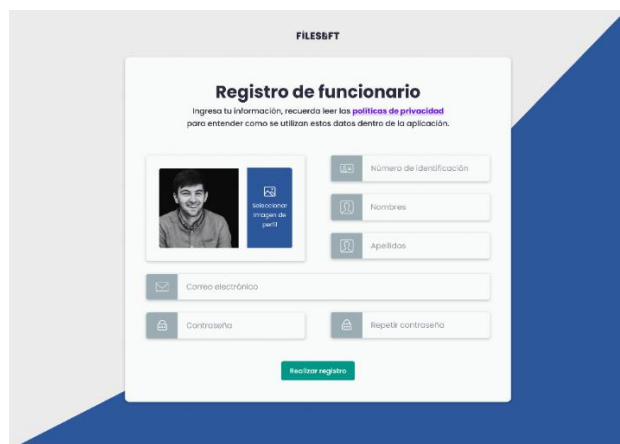
El formulario, titulado "Establecer contraseña", se encuentra sobre un fondo azul y gris. Incluye un campo para el "Número de identificación", campos para "Contraseña" y "Repetir contraseña", y un botón "Establecer nueva contraseña".

*Fuente.* Elaboración propia en Figma

La anterior imagen presenta al usuario el formulario para que establezca una nueva contraseña en el caso que la haya olvidado, por seguridad se le solicita su número de identificación.

### Figura 37

*Página de registro de nuevo funcionario*

El formulario, titulado "Registro de funcionario", incluye un campo para "Número de identificación", campos para "Nombres" y "Apellidos", un campo para "Correo electrónico", campos para "Contraseña" y "Repetir contraseña", y un botón "Realizar registro".

*Fuente.* Elaboración propia en Figma

La anterior imagen presenta el formulario de registro del funcionario que permitirá agregar la información e imagen que se almacenará en el sistema.

Figura 38

Página del listado de solicitudes o radicados

**Radicados**  
Dashboard > Radicados

Se te ha asignado el radicado **2023**, aún no lo has visto y tienes hasta el 30/01/2023 para responder

Se te ha asignado el radicado **2033**, aún no lo has visto y tienes hasta el 30/01/2023 para responder

El radicado **2029** está próximo a vencer, te quedan 3 días para dar una respuesta.

El radicado **2044** está próximo a vencer, te quedan 12 horas para dar una respuesta.

Total radicados: **120**

Pendientes: **12**

Completados: **118**

Escribe alguna palabra clave del radicado

Imprimir Excel PDF CSV 2023 Filtros + Generar radicado

Código: Ejm, 2023 Asunto: Ejm, derecho de petición Peticionario: Sin selección Estado: Sin selección

Tipo de oficio: Sin selección Dependencia: Sin selección Cargo: Sin selección

Entidad: Sin selección Fecha inicial: Sin selección Fecha final: 23/01/2023 Tiempo restante: Ejm, 2 días

Limpiar filtros Aplicar filtro

Código	Asunto	Peticionario	Fecha radicado	Tipo oficio	Estado
0004	Solicitud acceso fondo educativo	Claudia Milena Mahecha	23/01/2023 Restante: 3 días	PQRSD	Pendiente
0003	Tramite de legalización	Harrison Smith Romero	23/01/2023 Restante: 3 días	Derecho de petición	Pendiente
0002	Citacion comité de seguimiento electoral	Laura Estefania Martin Sarmiento	13/01/2023	PQRSD	Completado
0001	Respuesta oficio 24 visita para tala de arb...	Julian Vargas Pedraza	03/01/2023	PQRSD	Completado

Radicado No. 0004 **Pendiente**

Asunto: Solicitud acceso fondo educativo  
Tipo de oficio: PQRSD  
Fecha radicación: 23 de enero de 2023, 08:35:00 AM  
Fecha del documento: 18 de enero de 2023  
Tiempo restante: 3 días

Peticionario: Claudia Milena Mahecha (email@email.com)  
Cargo del peticionario: Ciudadano General  
Entidad: Ente territorial (enteterritorial@email.com)

Dependencia destino: Secretaría de desarrollo económico, Secretaría de desarrollo social.  
Dirigido a: Carmen Yanet Arévalo

Folios: 10  
Adjuntos: 0  
CDs: 0

Se han cargado 3 archivos

Página 1 de 10 | Ir a página: 1 | Listar 50 registros

Yeferson Zaque

Fuente. Elaboración propia en Figma

La anterior imagen muestra la lista de todos los radicados registrados en el sistema, además de las diferentes funciones que se permitirán al usuario para su respectiva gestión.

## Figura 39

### Página de generación de nueva solicitud o radicado

FILES&FT

- Inicio
- Radicados**
- Funcionarios
- Ciudadanos
- Entidades
- Cargos
- Tipos Solicitud

### Generar radicado

Dashboard > Radicados > Generar radicado

**Información del radicado**

Número radicado: 0004

Asunto del radicado:  
Ejm. Solicitud de acceso a...

Tipo de solicitud: Selecciona una opción | X | V

Fecha radicado: 06/02/2023 | X | C | La hora se agrega automáticamente desde el equipo

Tiempo de respuesta: Días hábiles para respuesta | días hábiles

1 día 2 días 10 días 15 días 30 días

Establece primero el tiempo de respuesta

**Información del peticionario**

Número de identificación: Ejm. 329867453 | X | Validar

Correo electrónico: (Recomendado) Ejm. email@email.com | X

Nombre del peticionario: Ejm. Luis Carlos Ortiz Pérez | X

Teléfono: Ejm. 3126673457 | X | Dirección de residencia: Ejm. Vereda Bombita | X

Entidad: Selecciona una opción | X | V | Cargo: Selecciona una opción | X | V

**Información de destino**


Dirigida a: Escribe o selecciona el nombre de la persona | X | V | El autocompletado le puede ayudar

Dependencias destino: Selecciona una o varias dependencias | X | V | Los jefes de las dependencias asignadas serán notificados del nuevo radicado.

**Carga de documentos**


Número de CDs: 0 CDs

Documento principal: Formato PDF No seleccionado

 Selecciona o arrastra el documento aquí

No has seleccionado ningún archivo

Adjuntos: No seleccionado

 Selecciona o arrastra los documentos aquí

No has seleccionado ningún archivo

Generar radicado

Fuente. Elaboración propia en Figma

La anterior imagen presenta al funcionario el formulario para registrar un nuevo radicado en el sistema.

**Figura 40***Modal de asignación de nuevo número de radicado*
*Fuente.* Elaboración propia en Figma

La anterior imagen presenta el formulario que permite editar el número de radicado de una solicitud existente.

**Figura 41***Modal de edición de nuevo tipo de solicitud*
*Fuente.* Elaboración propia en Figma

La anterior imagen presenta al funcionario un formulario para que edite el tiempo de respuesta de una solicitud existente.

Figura 42

Página de ver solicitud o radicado

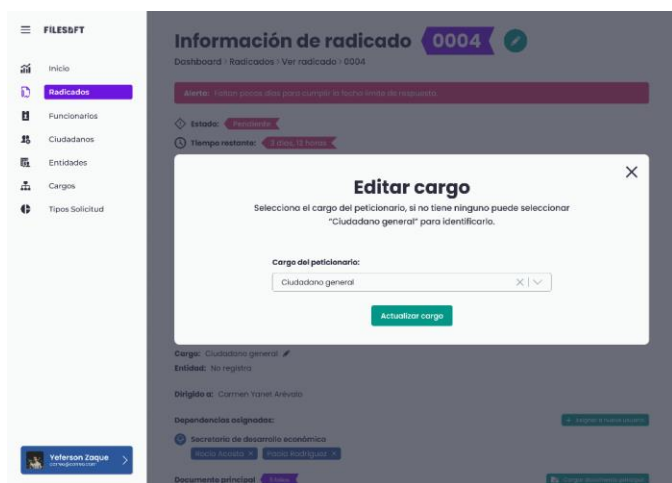
The screenshot displays a web interface for managing requests. On the left is a sidebar menu with the following items: Inicio, Radicados (highlighted), Funcionarios, Ciudadanos, Entidades, Cargos, and Tipos Solicitud. The main content area is titled 'Información de radicado' with the ID '0004'. Below the title is a breadcrumb trail: Dashboard > Radicados > Ver radicado > 0004. A pink alert banner states: 'Alerta: Faltan pocos días para cumplir la fecha límite de respuesta.' The status is 'Pendiente' with a remaining time of '3 días, 12 horas'. The request is assigned to 'Rocio Acosta' and 'Paola Rodríguez'. The subject is 'Solicitud acceso fondo educativo' and the type is 'PQRSD'. The request date is '23 de enero de 2023, 08:25:00 AM', with '15 días hábiles' remaining for response. The deadline is '12 de febrero de 2023, 05:00 PM'. The petitioner is 'Claudia Milena Mahecha'. Contact information includes ID '1010101010', email 'email@email.com', and no phone or address registered. The petitioner is a 'Ciudadano general' from 'Carmen Yanet Arévalo'. The request is directed to 'Secretaría de desarrollo económico', with 'Rocio Acosta' and 'Paola Rodríguez' as dependents. There is one main document (PDF) and four attachments (DOCX, PNG, XSLX, DWO). No CDs are registered. One incident is recorded. A note at the bottom states: 'Redacción de carta de solicitud al fondo de educación', generated by Rocio Acosta on Jan 30, 2023, for a request for access to higher education funding. A user profile for 'Yeferson Zaque' is visible in the bottom left corner.

Fuente. Elaboración propia en Figma.

La anterior imagen presenta al funcionario una visión amplia de la información de un radicado existente, incluyendo sus documentos adjuntos e información de las asignaciones.

**Figura 43**

*Modal de edición del cargo del ciudadano*

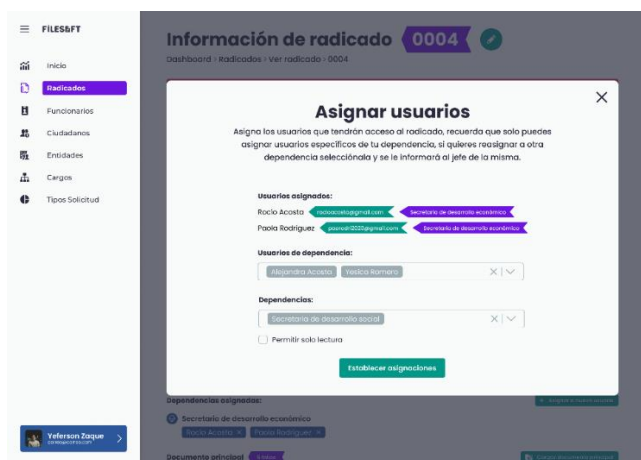


*Fuente.* Elaboración propia en Figma

La anterior imagen presenta al funcionario un formulario para que edite el cargo de un peticionario.

**Figura 44**

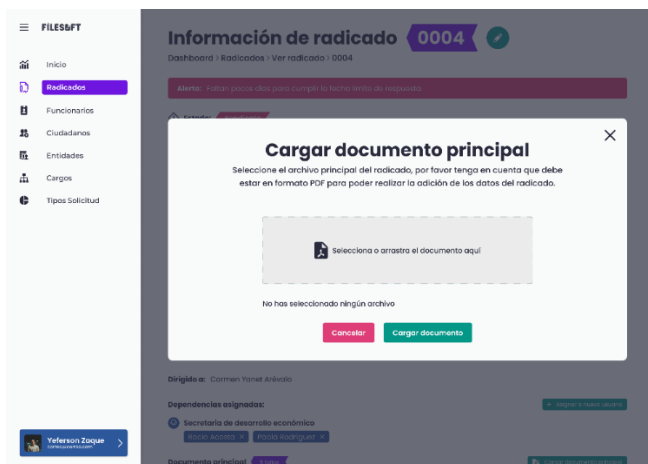
*Modal de asignación de usuarios responsables*



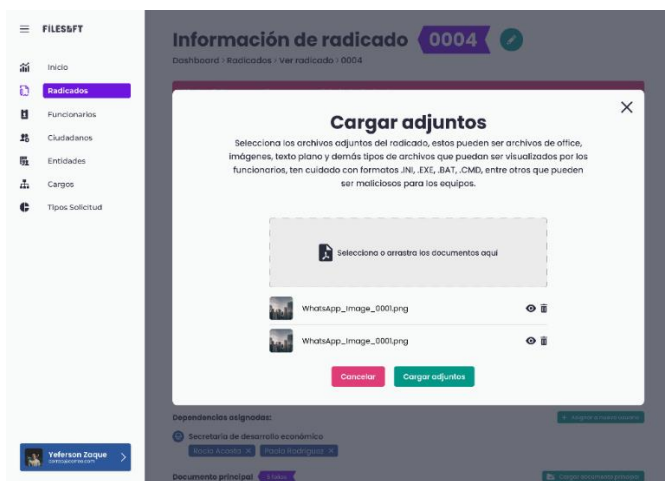
*Fuente.* Elaboración propia en Figma

La anterior imagen permite al funcionario asignar o eliminar las asignaciones de usuarios a un radicado.



**Figura 45***Modal de carga para documento principal**Fuente.* Elaboración propia en Figma

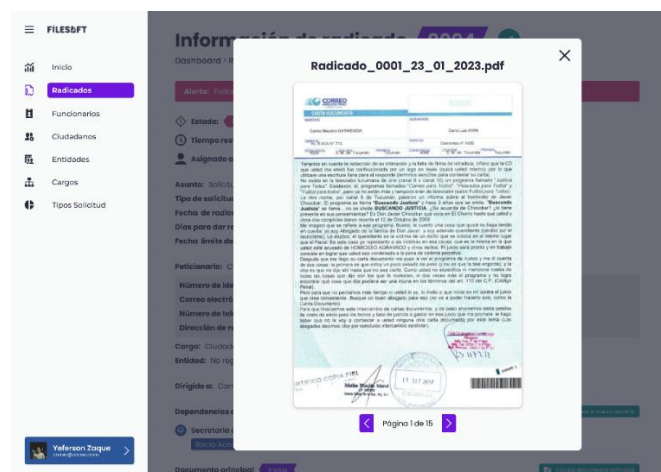
La anterior imagen permite al funcionario cargar el documento principal de cada solicitud teniendo en cuenta que es obligatorio.

**Figura 46***Modal de carga para adjuntos**Fuente.* Elaboración propia en Figma

La anterior imagen presenta al usuario el formulario para que cargue los archivos adjuntos, teniendo en cuenta que no son obligatorios su carga.

Figura 47

Modal de previsualización de documentos

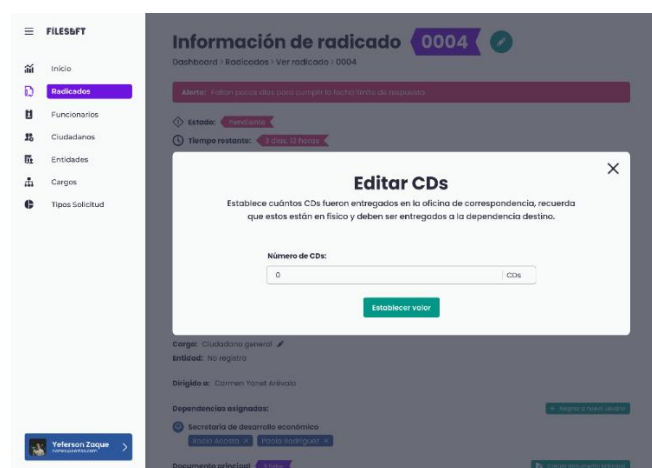


Fuente. Elaboración propia en Figma

La anterior imagen presenta al funcionario una previsualización de los documentos cargados en el sistema y que sean compatibles con previsualizadores del navegador.

Figura 48

Modal de edición para cantidad de CDs

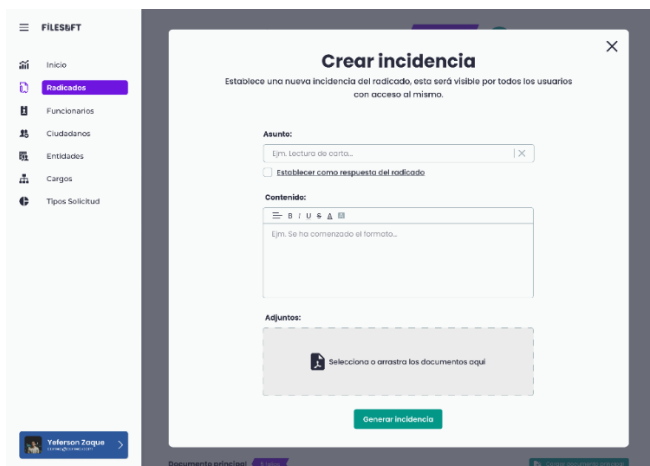


Fuente. Elaboración propia en Figma

La anterior imagen presenta un formulario para editar la cantidad de CDs que están adjuntos a la solicitud.

**Figura 49**

*Modal de creación de incidencia de radicado*

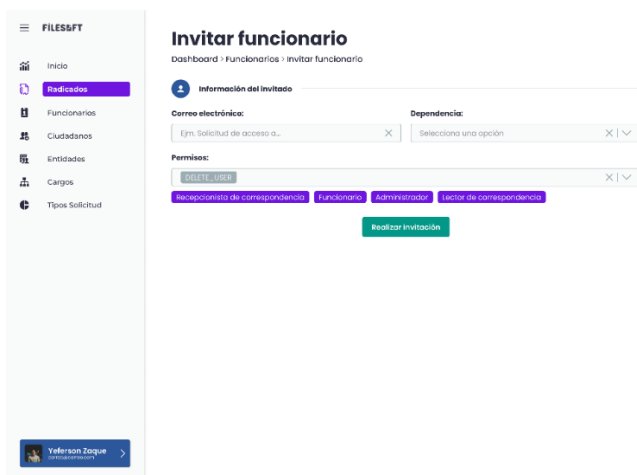


*Fuente.* Elaboración propia en Figma

La anterior imagen presenta el formulario para la generación de una nueva incidencia de una solicitud.

**Figura 50**

*Página para invitar un nuevo funcionario a que se registre*



*Fuente.* Elaboración propia en Figma

La anterior imagen permite al administrador invitar nuevos funcionarios para que accedan al sistema de filesoft.

Figura 51

Página de listado de funcionarios

**Funcionarios**  
Dashboard > Funcionarios

Total funcionarios: 120  
Inactivos: 0  
Activos: 5

Digitación de información del funcionario

Imprimir Excel PDF CSV Filtrar Ver invitaciones Invitar funcionario

Número de identificación: Ejm. 20234567  
Nombres: Ejm. Alejandra  
Apellidos: Ejm. Acosta  
Correo electrónico: Ejm. ale2022@rmail.com  
Estado: Sin selección  
Permisos: Selecciona una o varias opciones

Limpiar filtros Aplicar filtro

Identificación	Nombres	Apellidos	Email	Estado
1008654327	Luis	Romero	luisromero@gmail.com	Activo
Identificación: 1008654327 Nombres: Luis Apellidos: Romero Correo electrónico: luisromero@gmail.com Estado: Activo Permisos: READ_USER CREATE_USER UPDATE_USER DELETE_USER				
3044607	Maria	Rojas	mariarj@gmail.com	Activo
10875634	Juan	Velandia	juanv@gmail.com	Inactivo
153678923	Rocio	Martín	rociom2@gmail.com	Inactivo

Página 1 de 10 | Ir a página: 1 | Listar 50 registros

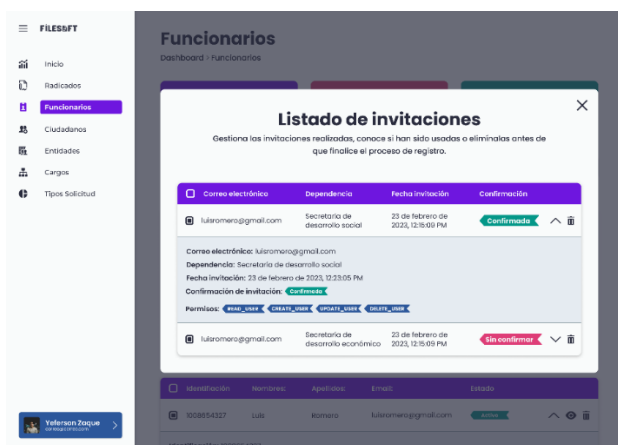
Yeferson Zaque correo@correo.com

Fuente. Elaboración propia en Figma

La anterior imagen presenta al funcionario una vista general de los de todos los funcionarios registrados en la aplicación.

## Figura 52

### Modal de listado de funcionarios invitados

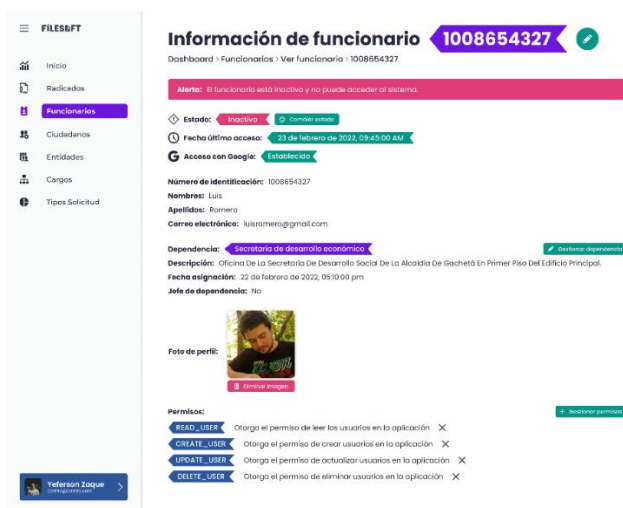


Fuente. Elaboración propia en Figma

La anterior imagen presenta un listado de todas las invitaciones generadas y su estado actual.

## Figura 53

### Página de visualización de funcionario

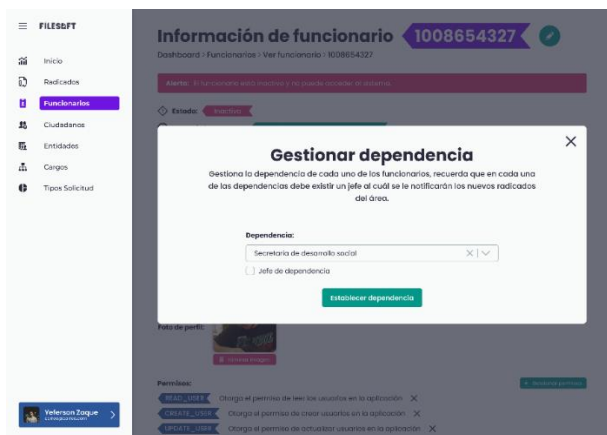


Fuente. Elaboración propia en Figma

La anterior imagen presenta la información de un funcionario, teniendo en cuenta su imagen y los permisos asignados a él.

**Figura 54**

*Modal de gestión de dependencia del funcionario*

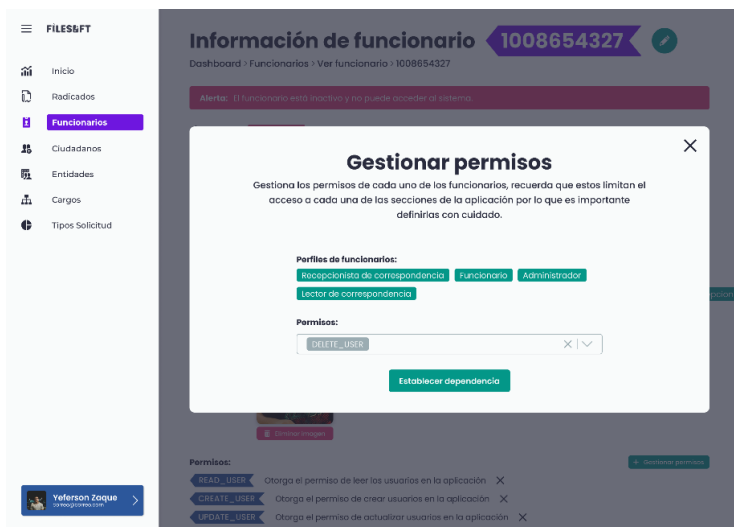


*Fuente.* Elaboración propia en Figma

La anterior imagen presenta la posibilidad de editar la dependencia de un funcionario, además de presentar si su asignación será como administrador de esta.

**Figura 55**

*Modal de gestión de permisos del funcionario*



*Fuente.* Elaboración propia en Figma

La anterior imagen da la posibilidad de de gestionar los permisos que tiene asignado un funcionario, agregando o eliminando según sea necesario.

Figura 56

Página de listado de ciudadanos

The screenshot displays a web application interface for managing citizens. On the left is a sidebar with navigation options: Inicio, Radicados (3), Funcionarios, Ciudadanos (selected), Entidades, Cargos, and Tipos Solicitud. The main content area is titled 'Ciudadanos' and includes a dashboard with three summary cards: 'Total ciudadanos' (12), 'Inactivos' (0), and 'Activos' (5). Below these are search and filter controls, including a search bar and buttons for 'Imprimir', 'Excel', 'PDF', 'CSV', 'Filtrar', and 'Insertar ciudadano'. A form section allows for data entry with fields for 'Número de identificación', 'Nombre completo', 'Teléfono', 'Correo electrónico', and 'Dirección'. Below the form are 'Limpiar filtros' and 'Aplicar filtro' buttons. The main part of the interface is a table listing citizens with columns for 'Identificación', 'Nombre completo', 'Email', 'Teléfono', and 'Estado'. The first row shows Dayana Lopez with ID 1008654327 and status 'Activo'. A detailed view of this citizen is shown below the table, including their identification number, full name, phone number, email, address, creation date, and status. At the bottom, there are navigation arrows, page information ('Página 1 de 10'), and a 'Listar 50 registros' dropdown.

Fuente. Elaboración propia en Figma

La anterior imagen presenta una vista detallada de los ciudadanos registrados en la aplicación, esto se realiza automáticamente en cada solicitud para dar un auto completado.

## Figura 57

### Página para insertar un nuevo ciudadano

Fuente. Elaboración propia en Figma

La anterior imagen presenta la posibilidad de registrar un nuevo ciudadano en la aplicación, esto es opcional pues al registra la solicitud se genera el registro del ciudadano en caso de no existir.

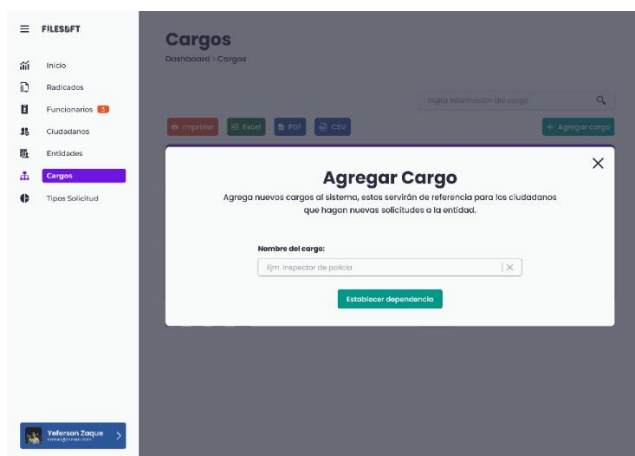
## Figura 58

### Página de visualización de ciudadano

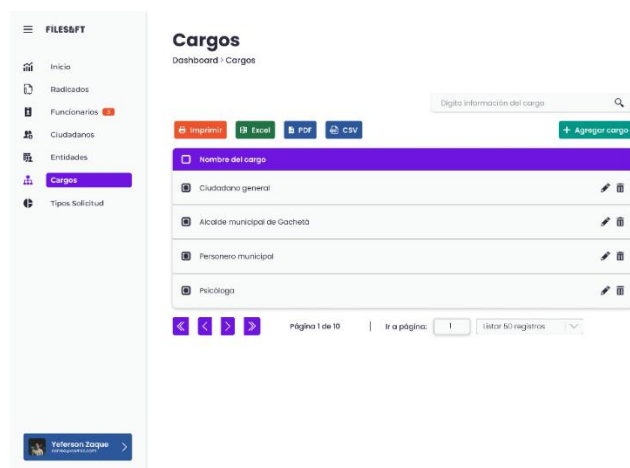
Fuente. Elaboración propia en Figma

La anterior imagen presenta la información en general del ciudadano.

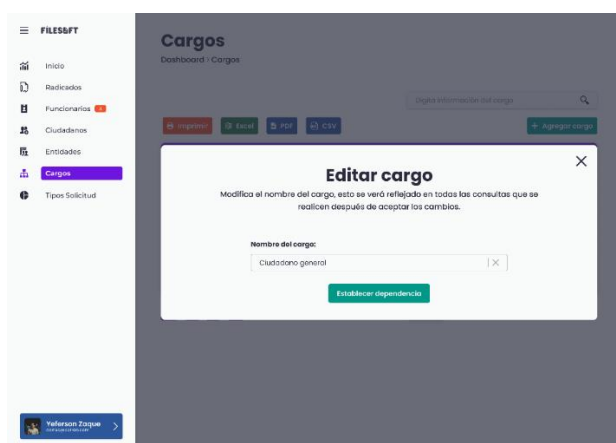


**Figura 59***Modal para insertar un nuevo cargo**Fuente.* Elaboración propia en Figma

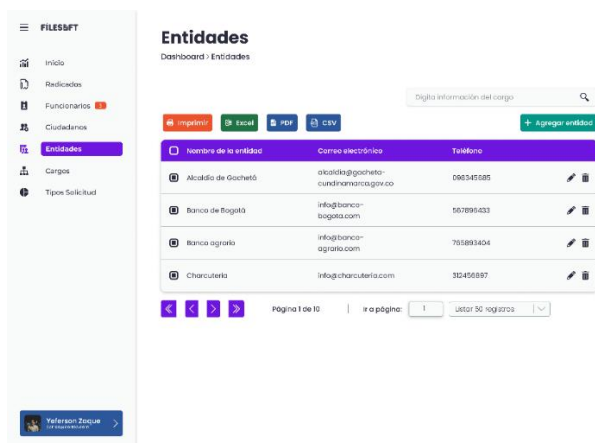
La anterior imagen presenta la posibilidad de agregar un nuevo cargo de los ciudadanos, esta es independiente para facilitar su registro.

**Figura 60***Página de listado de cargos**Fuente,* Elaboración propia en Figma

La anterior imagen presenta el listado de cargos registrados en la aplicación, los cuales son la base del autocompletado.

**Figura 61***Modal de edición de cargos**Fuente.* Elaboración propia en Figma

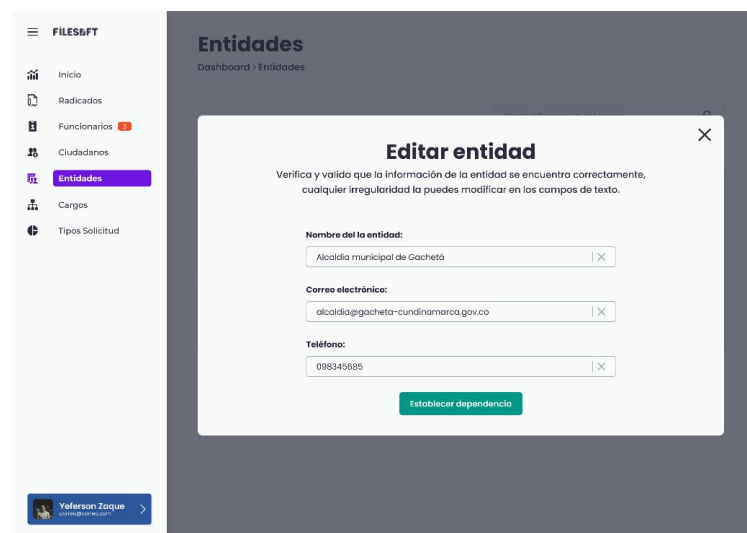
La anterior imagen presenta el formulario para editar la información de un cargo específico.

**Figura 62***Página de listado de entidades**Fuente.* Elaboración propia en Figma

La anterior imagen presenta el listado de entidades registradas en la aplicación, estas también se registran automáticamente al registrar una solicitud o manualmente en su correspondiente pestaña.

## Figura 63

### Modal de edición de entidad



*Fuente.* Elaboración propia en Figma

La anterior imagen presenta el formulario para editar la información de una entidad específica.

## Prototipado de la Aplicación

El prototipado permite definir cuáles son las interacciones que se pueden realizar dentro de la aplicación, solo que se pueden definir desde el diseño y ver su funcionamiento con ayuda de animaciones y transiciones que dan idea del software en etapa de desarrollo, esto es esencial para ver que animaciones pueden ir en conjunto con la interfaz de usuario.

## Figura 64

### *Prototipado de la aplicación en Figma*



*Fuente.* Ventana de prototipado de la aplicación de Figma

Este software permite agregar interacciones por parte del diseñador UI que serán presentadas a los desarrolladores, lo cual es útil para ver su viabilidad y complejidad, también permite la navegación entre páginas lo cual da una idea más clara del objetivo a cumplir.

## Modelado de bases de datos

Es una de las etapas más importantes en el desarrollo de una aplicación, esto se debe a que la base de datos será la encargada de almacenar y relacionar toda la información de los diferentes modelos que se encuentran en la etapa de análisis, estos en el servicio REST son modelados en código con ayuda del ORM para facilitar la lectura, registro, actualización y eliminación de los registros.

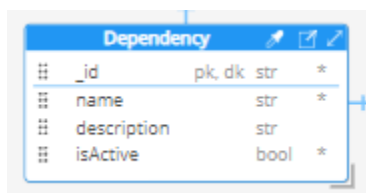
El diseño se realizó con ayuda del software Hackolade que ofrece múltiples herramientas que facilitan el diseño y relación de los modelos, además de tener un enfoque específico en MongoDB que es el gestor de bases de datos seleccionado.

### Definición de modelos

El modelo “Dependency” se encarga de almacenar la información de las dependencias de la entidad, de esta se almacena un identificador único, el nombre de la dependencia, su descripción y su estado para saber si está activo dentro del sistema y se puede seleccionar.

### Figura 65

*Atributos del modelo de dependencia*



Dependency			
::	_id	pk, dk, str	*
::	name	str	*
::	description	str	
::	isActive	bool	*

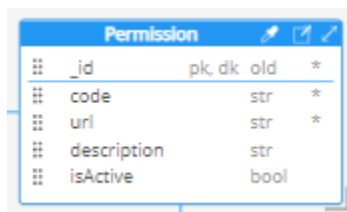
*Fuente.* Elaboración propia en el software Hackolade.

El modelo “Permission” se encarga de almacenar la información de los permisos que se pueden asignar dentro de la aplicación de correspondencia, de estos se almacena el código del

permiso, la ruta donde se gestionará, una breve descripción y el estado de este, permitiendo así establecer manualmente los alcances de los usuarios.

### Figura 66

#### *Atributos del modelo de permiso*



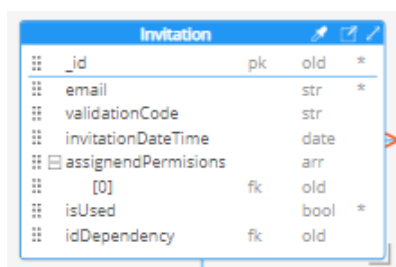
Permission			
::	_id	pk, dk, old	*
::	code	str	*
::	url	str	*
::	description	str	
::	isActive	bool	

*Fuente.* Elaboración propia en el software Hackolade.

El modelo “Invitation” se encargará de almacenar la información de las invitaciones que se realicen a nuevos funcionarios para que se registren en el software, de este se almacena el correo electrónico del invitado, un código de verificación, la fecha, permisos, dependencia del invitado y si la invitación fue utilizada.

### Figura 67

#### *Atributos del modelo de invitación*



Invitation			
::	_id	pk, old	*
::	email	str	*
::	validationCode	str	
::	invitationDateTime	date	
::	assignendPermissions	arr	
::	[0]	fk, old	
::	isUsed	bool	*
::	idDependency	fk, old	

*Fuente.* Elaboración propia en el software Hackolade

El modelo “AssignedDependency” se encarga de almacenar la información de las asignaciones de los usuarios a su respectiva dependencia, de esta establece si es administrador de esta, la dependencia relacionada y el usuario que se asignó a dicha dependencia.

**Figura 68**

*Atributos del modelo de asignación de dependencia*

AssignedDependency			
_id	pk, dk	old	*
isAdministrator		bool	*
dependencyId	fk	old	*
userId	fk	old	*

*Fuente.* Elaboración propia en el software Hackolade.

El modelo “File” es el encargado de almacenar toda la información de los archivos digitales que procese y almacene la aplicación, de estos almacenará el nombre del archivo, la ruta completa, la ruta relativa, el tipo de archivo, la extensión y el tamaño en bytes.

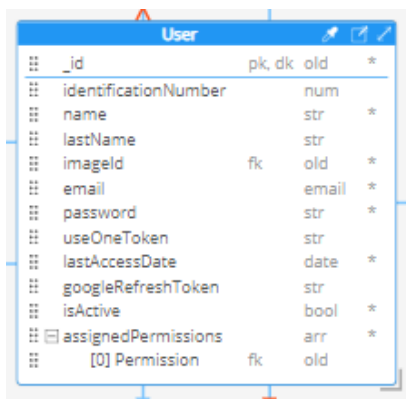
**Figura 69**

*Atributos del modelo de archivo*

File			
_id	pk, dk	old	*
fileName		str	*
pathName		str	*
relativeFullPath		str	*
type		str	*
extension		str	*
size		num	*

*Fuente.* Elaboración propia en el software de Hackolade.

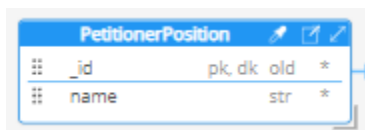
El modelo “User” almacena la información de los funcionarios de la entidad, dentro de esta información se incluye el número de identificación, los nombres, los apellidos, la imagen de perfil, el correo electrónico, la contraseña, un token de verificación, la fecha del último acceso, los tokens de seguridad de Google, los permisos y su estado dentro del sistema.

**Figura 70***Atributos del modelo de usuario*


User			
_id	pk, dk	old	*
identificationNumber		num	*
name		str	*
lastName		str	*
imageId	fk	old	*
email		email	*
password		str	*
useOneToken		str	*
lastAccessDate		date	*
googleRefreshToken		str	*
isActive		bool	*
assignedPermissions		arr	*
[0] Permission	fk	old	

*Fuente.* Elaboración propia en el software Hackolade.

El modelo “PetitionerPosition” almacena el cargo de los peticionarios de la entidad, establecerán una relación al registrarse el radicado para no generarla de manera directa con el modelo, así facilitando las posibles modificaciones que se realicen.

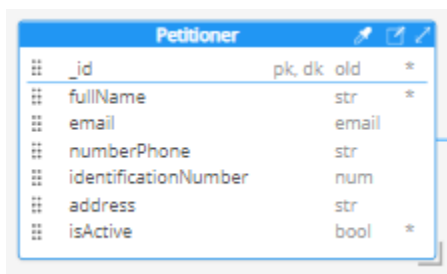
**Figura 71***Atributos del modelo de cargos*


PetitionerPosition			
_id	pk, dk	old	*
name		str	*

*Fuente.* Elaboración propia en el software de Hackolade.

El modelo “Petitioner” será el encargado de almacenar y gestionar la información de los peticionarios o ciudadanos que realicen las solicitudes a la entidad, este almacenará información esencial como el nombre, el correo electrónico, el número de teléfono, el número de identificación, la dirección y su estado dentro del sistema.

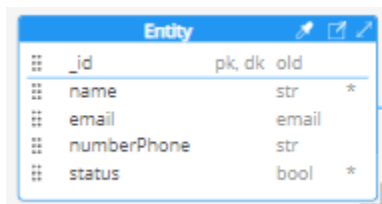


**Figura 72***Atributos del modelo de peticionario*


Attribute	Type	Other
_id	pk, dk	old *
fullName	str	*
email	email	
numberPhone	str	
identificationNumber	num	
address	str	
isActive	bool	*

*Fuente.* Elaboración propia en el software Hackolade.

El modelo “Entity” almacenará toda la información de las entidades que se relacionen con los peticionarios, algo a tener en cuenta es que al igual que con el modelo de cargo este no tendrá una relación directa con el peticionario si no solo se registrará dentro del registro de la solicitud.

**Figura 73***Atributos del modelo de entidad*


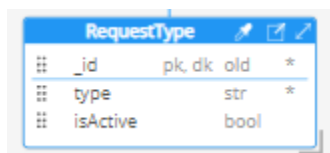
Attribute	Type	Other
_id	pk, dk	old
name	str	*
email	email	
numberPhone	str	
status	bool	*

*Fuente.* Elaboración propia en el software de Hackolade.

El modelo de “RequestType” ayudará a referenciar los tipos de solicitud que pueden realizar los peticionarios, este se relacionará directamente con la solicitud y almacenará un identificador único al igual que el tipo y el estado de este.

**Figura 74**

*Atributos del modelo de tipo de solicitud*



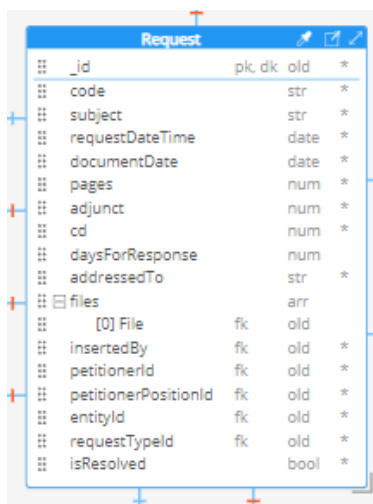
RequestType			
pk, dk, old	_id		*
	type	str	*
	isActive	bool	

*Fuente.* Elaboración propia en el software de Hackolade.

El modelo “Request” será el que almacenará todos los datos de las solicitudes que se realicen a la entidad, por esta razón es de los más extensos ya que registra el código de la solicitud, el encabezado, la fecha, la fecha del documento, el número de folios, adjuntos, CDs, los días para dar respuesta, a quién va dirigido, la información del solicitante y una referencia de si se le dio respuesta.

**Figura 75**

*Atributos del modelo de solicitud*



Request			
pk, dk, old	_id		*
	code	str	*
	subject	str	*
	requestDateTime	date	*
	documentDate	date	*
	pages	num	*
	adjunct	num	*
	cd	num	*
	daysForResponse	num	*
	addressedTo	str	*
	files	arr	*
	[0] File	fk	old
	insertedBy	fk	old
	petitionerId	fk	old
	petitionerPositionId	fk	old
	entityId	fk	old
	requestTypeId	fk	old
	isResolved	bool	*

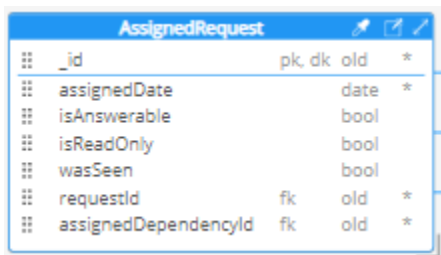
*Fuente.* Elaboración propia en el software Hackolade.

El modelo “AssignedRequest” almacenará la relación entre los funcionarios y una solicitud, así permitiendo facilitar el acceso solo a personas autorizadas pues almacenará una referencia de la solicitud, de un usuario que tendrá acceso, la fecha de asignación, si solo tiene

permisos de lectura, si ya fue visto por el usuario y si es el encargado de dar respuesta a dicha solicitud.

**Figura 76**

*Atributos del modelo de asignación de solicitud*



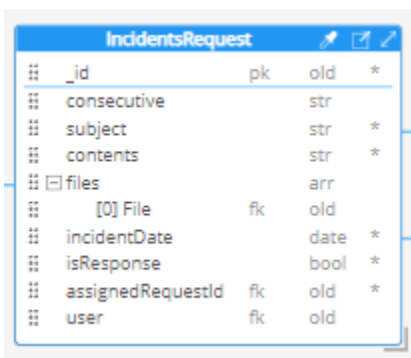
Column	Attributes
_id	pk, dk, old, *
assignedDate	date, *
isAnswerable	bool
isReadOnly	bool
wasSeen	bool
requestId	fk, old, *
assignedDependencyId	fk, old, *

*Fuente.* Elaboración propia en el software de Hackolade.

El modelo de “IncidentRequest” será el que almacenará alguna incidencia que se realice sobre una solicitud, esta almacenará el código del consecutivo, el contenido de la incidencia, los archivos asociados, la fecha en que se realizó y si es la respuesta a la solicitud como tal.

**Figura 77**

*Atributos del modelo de incidencia de solicitud*



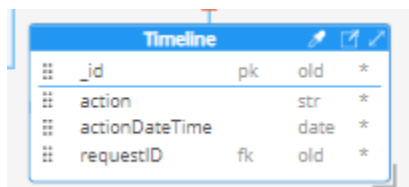
Column	Attributes
_id	pk, old, *
consecutive	str
subject	str, *
contents	str, *
files	arr
[0] File	fk, old, *
incidentDate	date, *
isResponse	bool, *
assignedRequestId	fk, old, *
user	fk, old

*Fuente.* Elaboración propia en el software de Hackolade.

El modelo de “Timeline” será el encargado de almacenar la línea de tiempo de las acciones que se realicen sobre una solicitud, esta tiene el fin de facilitar al peticionario ver el estado de solicitud y su avance dentro de la entidad.

## Figura 78

*Atributos del modelo de línea de tiempo*



Timeline			
_id	pk	old	*
action		str	*
actionDateTime		date	*
requestID	fk	old	*

*Fuente.* Elaboración propia en el software de Hackolade.

## Relación de entidades

Usando el modelo de relación de MongoDB se definen las siguientes relaciones que con ayuda del ORM ayudarán a gestionar de manera sencilla la información de la relación y procesar los datos en un tiempo mínimo.

**Tabla 7**

*Definición de relaciones entre modelos*

Modelo	Modelos relacionados
Invitation	Permission, Dependency
AssignedDependency	Dependency, User
User	File, Permission
Session	User
AssignedRequest	Request, AssignedDependency
Request	File, User, Petitioner, PetitionerPosition, Entity, RequestType
Timeline	Request
IncidentRequest	File, AssignedRequest, User

*Nota.* Se definen las relaciones existentes entre los modelos de la base de datos.

## **Documentación de Servicio API**

Contiene un conjunto de peticiones HTTP que se utilizan para interactuar con los diferentes recursos de la aplicación. Incluye rutas para crear, leer, actualizar y eliminar radicados, así como para gestionar información de funcionarios y otros recursos relacionados con la correspondencia. Además, también se pueden incluir rutas para la autenticación y autorización de usuarios. Estas rutas se definen mediante verbos HTTP (GET, POST, PUT, DELETE) y se utilizan para interactuar con los recursos de la aplicación, permitiendo a los usuarios enviar y recibir información de manera segura y organizada. Estas rutas son una pieza fundamental en el diseño de la arquitectura de la aplicación y son necesarias para poder desarrollar y probar correctamente el software.

### **Tipo de Autenticación**

Se usará el tipo de autenticación de Bearer Token, el cual nos dice la página Keepcoding que “es un tipo de header de una petición HTTP que carga con información relevante al tipo de usuario con el objetivo de darle autoridad para ejecutar la acción requerida en la petición. En este orden de ideas, la cabecera authorization bearer suele contener el JWT o JSON Web Token que se genera cuando el usuario inicia sesión”. (KeepCoding, 2022)

### **Manejo de Información**

La información será recibida en formato JSON, el cual es un objeto que tiene claves y valores para identificar cada dato, es muy popular hasta el día de hoy debido a su simplicidad y al peso que este tiene en la transferencia de cliente a servidor o viceversa.

## Figura 79

### Ejemplo formato JSON

Body raw (json)

```

json
{
  "description": "Otorga acceso al listado de radicados o solicitudes",
  "isActive": false
}

```

*Fuente.* Software de Postman.

## Definición de Rutas

Estas rutas son las que llama la aplicación del cliente para obtener información, insertarla, actualizarla, eliminarla, etc. Siguiendo la estructura de API REST se usará la dirección de desarrollo <http://localhost:5454> la cual será base en la definición de cada uno de los grupos de URL.

### *Rutas para gestión de archivos*

Rutas que permiten gestionar los archivos que se carguen dentro de la aplicación, teniendo la posibilidad de visualizarlos, listar todos los archivos cargados, cargar nuevos archivos, actualizar o eliminar alguno de estos.

## Tabla 8

### *Grupo de rutas para gestión de archivos*

Título	URL	HTTP	Descripción
Listar Archivos	/archivos	GET	Lista todos los archivos registrados en la aplicación o solo los asociados al

			usuario específico, esto depende de los permisos que se registren.
Obtener Archivo	/archivos/{id}	GET	Obtiene un archivo específico con el identificador y el token de acceso del usuario que lo solicita.
Subir Archivo	/archivos	POST	Permite subir un archivo al servidor y retorna la información de la subida del archivo.
Actualizar Archivo	/archivos/{id}	PUT	Actualiza la información de un archivo específico, realiza la carga del nuevo archivo y actualiza la información.
Eliminar Archivo	/archivos/{id}	DELETE	Permite eliminar un archivo de la aplicación, se debe tener en cuenta que este archivo no esté asociado a ningún elemento, en caso de estarlo no se permitirá su eliminación.

---

*Nota.* Se definen las rutas que se encargarán de gestionar los archivos en la aplicación.

### ***Rutas para gestionar las asignaciones de dependencias***

Contiene las rutas que permiten gestionar las asignaciones de dependencias de la aplicación, para esto se requiere que los usuarios tengan los permisos necesarios o en caso necesario se denegarán las solicitudes.

**Tabla 9***Definición de rutas para gestionar las asignaciones de dependencias*

Título	URL	HTTP	Descripción
Listar Asignación Dependencias	/asignacion- dependencias	GET	Obtiene el listado de todas las asignaciones de dependencias que se han realizado, mostrando los funcionarios y las dependencias involucradas en cada una.
Obtener Asignación Dependencias	/asignacion- dependencias/{id}	GET	Permite obtener la información de una asignación de acuerdo al identificar de la misma, esta mostrará la información del usuario y la dependencia.
Crear Asignación Dependencia	/asignacion- dependencias	POST	Permite crear una nueva asignación de una dependencia a un usuario específico, se requiere el id del usuario, el id de la dependencia y especificar si este será o no administrador.
Actualizar Asignación Dependencia	/asignacion- dependencias/{id}	PUT	Permite actualizar la asignación de dependencia y definir el usuario es administrador de esta.



Eliminar	Permite eliminar de manera lógica
Asignación	una asignación de dependencia,
Dependencia	únicamente utilizando el identificador del registro.

---

*Nota. Se definen las rutas necesarias para gestionar las asignaciones de dependencias*

### ***Rutas de Gestión para Cargos***

Gestiona las profesiones que pueden tener los peticionarios, este solo almacena lo básico para tener una referencia de la información dentro de las solicitudes o radicados.

#### **Tabla 10**

*Grupo de rutas para gestionar los cargos*

Titulo	URL	HTTP	Descripción
Listar Cargos	/api/cargo-peticionario	GET	Lista todas las posiciones o profesiones de los peticionarios de la entidad y que servirán de apoyo en la aplicación.
Obtener Cargo	/api/cargo-peticionario/{id}	GET	Obtiene la posición o profesión de un peticionario de acuerdo con su identificador.
Crear Cargo	/api/cargo-peticionario	POST	Almacena una nueva posición o cargo de un peticionario en la aplicación.

---

Actualizar Cargo	/api/cargo-peticionario/{id}	PUT	Actualiza el nombre de un cargo o posición de un peticionario de la entidad.
Eliminar Cargo	/api/cargo-peticionario/{id}	DELETE	Elimina un cargo de peticionarios de acuerdo con su identificador.
Inicializar Cargos	/api/inicializar/cargo-peticionarios	POST	Inicializa una lista de cargos básicos para los peticionarios de la entidad.

---

*Nota.* Listado de rutas para gestionar los cargos de los peticionarios.

### ***Rutas de Gestión para Peticionarios***

Grupo de rutas que permiten gestionar los peticionarios o ciudadanos que tendrán la posibilidad de realizar solicitudes, las cuales se almacenarán en la aplicación.

#### **Tabla 11**

##### *Grupo de rutas para gestión de peticionarios*

Titulo	URL	HTTP	Descripción
Listar Peticionarios	/api/peticionario	GET	Obtiene un listado de todos los peticionarios registrados, este además permite realizar búsquedas y filtros sobre la información antes de obtenerla.
Obtener Peticionario	/api/peticionario/{id}	GET	Obtiene la información de un peticionario teniendo en cuenta su identificador.

---

Crear Petionario	/api/petionario	POST	Inserta un nuevo petionario que la entidad podrá gestionar y manejar para crear solicitudes.
Actualizar Petionario	/api/petionario/{id}	PUT	Actualiza la información de un petionario de acuerdo con su identificador.
Actualizar Estado Petionario	/api/petionario/{id}	PATCH	Actualiza el estado de un petionario, determinando si este puede o no generar solicitudes en el sistema.
Eliminar Petionario	/api/petionario/{id}	DELETE	Elimina permanentemente toda la información de un petionario.

---

*Nota.* Listado de rutas para trabajar la información de los petionarios.

### ***Rutas de Gestión de Dependencias***

Gestiona las dependencias de la entidad, crear, eliminar, modificar, o actualizar alguna de ellas, siempre que el usuario que lo realice tenga los permisos para realizarlo.

#### **Tabla 12**

*Grupo de rutas para gestionar las dependencias*

Titulo	URL	HTTP	Descripción
Listar Dependencias	/api/dependencia	GET	Obtiene un listado de todas las dependencias registradas para la entidad.

---

Obtener Dependencia	/api/dependencia/{id}	GET	Obtiene una dependencia especifica de acuerdo con el identificador de esta.
Crear Dependencia	/api/dependencia	POST	Inserta una nueva dependencia de la entidad y la almacena en la base de datos.
Actualizar Dependencia	/api/dependencia/{id}	PUT	Actualiza la información de una dependencia especifica.
Actualizar Estado Dependencia	/api/dependencia/{id}	PATCH	Actualiza y cambia el estado de una dependencia para identificar si esta está activa o no.
Eliminar Dependencia	/api/dependencia/{id}	DELETE	Elimina una dependencia de acuerdo con su identificador único.

---

*Nota.* Listado de rutas para gestionar las dependencias de la aplicación.

### ***Rutas de gestión de entidades***

Grupo de rutas que permite gestionar la información de las entidades que se asociarán a las solicitudes y a los ciudadanos, se pueden crear, leer, actualizar y eliminar los registros.

#### **Tabla 13**

##### *Grupo de rutas para gestionar las entidades*

Título	URL	HTTP	Descripción
Listar Entidades	/entidades	GET	Obtiene un listado con la información de las entidades registradas en la

---

			aplicación, realizando filtros y búsquedas.
Obtener Entidad	/entidades/{id}	GET	Obtiene la información de una entidad únicamente con su identificador.
Crear Entidad	/entidades	POST	Permite insertar una nueva entidad a la aplicación, validando que esta no exista con anterioridad.
Actualizar Entidad	/entidades/{id}	PUT	Permite actualizar la información de una entidad según el identificador dado en la solicitud.
Actualizar Estado Entidad	/entidades/{id}	PATCH	Permite actualizar y definir si una entidad está activa o no dentro de la aplicación.
Eliminar Entidad	/entidades/{id}	DELETE	Permite eliminar de manera lógica un registro de dependencia con el identificador de esta.

---

*Nota.* Se definen las rutas que permiten gestionar la información de las entidades

### ***Rutas de Gestión de Usuarios***

Gestionará las acciones que puede realizar un administrador para gestionar los usuarios de la aplicación, entre ellos editar su información, cambiar su estado, eliminarlo y obtener el listado de todos los usuarios registrados en la aplicación.

**Tabla 14***Grupo de rutas para gestión de usuarios*

Titulo	URL	HTTP	Descripción
Listar Usuarios	/api/usuario	GET	Obtiene una lista de cada uno de los usuarios registrados en la aplicación, estos solo son posible de acceder por parte de un administrador.
Obtener Usuario	/api/usuario/{id}	GET	Obtiene la información de un usuario de acuerdo con su identificador, este obtiene la información de manera detallada.
Crear Usuario	/api/usuario	POST	Informa al usuario que no es posible crear usuario directamente si no debe ir a la sección de invitaciones.
Actualizar Usuario	/api/usuario/{id}	PUT	Actualiza general de un usuario registrado en la aplicación.
Actualizar Estado Usuario	/api/usuario/{id}	PATCH	Cambia el estado de un usuario, denegando o permitiéndole el acceso al sistema
Eliminar Usuario	/api/usuario/{id}	DELETE	Elimina permanentemente un usuario del sistema.

*Nota.* Listado de rutas para gestionar los usuarios o funcionarios de la aplicación.

### ***Rutas para gestionar las invitaciones***

Este grupo de rutas permite varias acciones como generar una invitación que será enviada al correo del usuario que se especifique, obtener información de una o todas las solicitudes registradas, además permite editarlas o eliminarlas.

**Tabla 15**

#### *Grupo de rutas para gestión de invitaciones*

Titulo	URL	HTTP	Descripción
Invitar usuario	/invitaciones	POST	Genera una invitación para que el funcionario que se indicó reciba un correo electrónico con la información para que haga su registro en la aplicación.
Listar Invitaciones	/invitaciones	GET	Lista y página las invitaciones que se han realizado en la aplicación.
Obtener Invitación	/invitaciones/{id}	GET	Obtiene toda la información de una invitación de acuerdo a su identificador.
Actualizar Invitación	/invitaciones/{id}	PUT	Permite actualizar la información de una invitación antes de que sea utilizada, si ésta ya fue usada no se pueden realizar cambios.
Eliminar Invitación	/invitaciones/{id}	DELETE	Elimina de manera permanente una invitación del sistema, por lo tanto el

usuario invitado ya no tendrá la posibilidades de registrarse en caso de que no lo haya hecho.

---

*Fuente.* Se definen las rutas que ayudarán a gestionar la información de las invitaciones.

### ***Rutas de Gestión de permisos***

Gestionará los permisos de la aplicación, no se permite agregar nuevos permisos debido a que por defecto se cargan todos los permisos de la aplicación y solo será posible actualizar su estado global.

**Tabla 16**

*Grupo de rutas para gestión de permisos*

Titulo	URL	HTTP	Descripción
Listar Permisos	/api/permisos	GET	Obtiene la lista de todos los permisos que contiene la aplicación.
Obtener Permiso	/api/permisos/{id}	GET	Obtiene la información de un permiso de acuerdo con su identificador, detallando la información de este.
Actualizar Permiso	/api/permisos/{id}	PUT	Actualiza la descripción o el estado de un permiso dentro de la aplicación.
Insertar Permiso	/permisos	POST	Gestiona la creación de un permiso, por seguridad esto no es posible, lo cual se le indica al usuario que quiera realizar la acción.

---



Eliminar Permiso	/api/permisos/{id}	DELETE	Gestiona la eliminación de un permiso, por seguridad esto no es posible, lo cual se le indica al usuario que quiera realizar la acción.
------------------	--------------------	--------	---

---

*Nota.* Se listan las rutas para manejar los permisos de la aplicación

### ***Rutas de Gestión de Solicitudes***

Gestionará todas rutas para la gestión de solicitudes o radicados por parte de los ciudadanos, esta será administrada por parte de los funcionarios que tengan los permisos necesarios para realizar estas acciones.

#### **Tabla 17**

##### *Grupo de rutas para solicitudes*

Titulo	URL	HTTP	Descripción
Listar Solicitudes	/api/solicitudes	GET	Lista todas las solicitudes realizadas a la entidad, en esta se establecen filtros y validaciones de seguridad de acuerdo con los permisos que posea cada usuario.
Obtener Solicitud	/api/solicitudes/{id}	GET	Obtiene toda la información de una solicitud de acuerdo con su identificador.

---

Crear Solicitud	/api/solicitudes	POST	Crea una nueva solicitud por parte de los ciudadanos del municipio, este almacena todo lo necesario para su resolución.
Actualizar Solicitud	/api/solicitudes/{id}	PUT	Actualiza la información básica de una solicitud, identificándola por su identificar o id.
Eliminar Solicitud	/api/solicitudes/{id}	DELETE	Elimina permanentemente una solicitud registrada en la aplicación.

---

*Nota.* Se listan las rutas para trabajar las solicitudes o radicados.

### ***Rutas de Gestión Tipos de Solicitud***

Gestiona los tipos de solicitud que permite la entidad, por defecto se definen algunos encontrados en el estudio previo de la entidad.

#### **Tabla 18**

*Grupo de rutas para gestión de tipos de solicitud*

Titulo	URL	HTTP	Descripción
Listar Tipos Solicitud	/api/tipo-solicitud	GET	Obtiene el listado de todos los tipos de solicitud que la entidad gestiona.
Obtener Tipo Solicitud	/api/tipo-solicitud/{id}	GET	Obtiene la información de un tipo de solicitud de acuerdo a su identificador único.
Crear Tipo Solicitud	/api/tipo-solicitud	POST	Genera un nuevo tipo de solicitud para la entidad, es importante determinar

---

			cuántos días hábiles tiene el nuevo tipo de solicitud.
Actualizar Tipo Solicitud	/api/tipo-solicitud/{id}	PUT	Actualiza la información de un tipo de solicitud de acuerdo con su tipo de identificador único.
Actualizar Estado Tipo Solicitud	/api/tipo-solicitud/{id}	PATCH	Actualiza el estado de un tipo de solicitud y define si está o no está activo.
Eliminar Tipo Solicitud	/api/tipo-solicitud/{id}	DELETE	Elimina un tipo de solicitud teniendo en cuenta su identificador.

---

*Nota.* Listado de rutas para gestionar los tipos de solicitud.

### ***Rutas de Autorización***

Agrupar las rutas relacionadas con la autenticación de usuarios. Dentro de esta carpeta se encuentran todas las rutas necesarias para permitir que los usuarios inicien sesión en la aplicación, se registren, cambien sus contraseñas, entre otras funciones relacionadas con la autenticación. Estas rutas son esenciales para garantizar la seguridad de la aplicación y controlar el acceso a los recursos y funciones restringidos.

#### **Tabla 19**

##### *Grupo de rutas de autorización*

Titulo	URL	HTTP	Descripción
Validar Token Acción	/api/auth/validar-accion/{token}	GET	Realiza la validación de que el token de un solo uso sea válido.

---

Validar Invitación	/api/auth/validar- invitacion/{token}	GET	Cuando un administrador genera una invitación, esta valida que el identificador de la invitación sea válido y esté vigente.
Validar Email Registro	/api/auth/validar- registro/{token}	GET	Valida que el registro del usuario invitado se generó de manera correcta y activa la cuenta para poder iniciar sesión.
Validar Sesión funcionario	/api/auth/validar-session	GET	Valida que la sesión del funcionario sea válida y permita continuar con el uso de la aplicación.
Iniciar Sesión	/api/auth/iniciar-sesion	POST	El usuario accede con sus credenciales para poder generar un token de acceso que le permitirá acceder al dashboard de administración.
Restablecer Contraseña	/api/auth/restablecer- password/{token}	POST	Cuando el usuario olvide su contraseña y haya solicitado la actualización de esta se le permitirá establecer una nueva.
Olvidé Contraseña	/api/auth/olvide- password	POST	Si el usuario olvidó su contraseña puede solicitar su restablecimiento.

---

Registrar Usuario Invitado	/api/auth/registrar/{code}	POST	Permite registrar toda la información del nuevo usuario o funcionario, este es el paso inicial para que se otorgue acceso al Dashboard de administración.
Actualizar Información Usuario Autenticado	/api/auth/actualizar-informacion	PUT	El usuario autenticado puede actualizar su información, cabe destacar que esta no incluye información sensible como identificación y correo electrónico. Esto se debe hacer como solicitud a un administrador.
Actualizar Información Usuario Autenticado	/auth/actualizar-informacion	PATCH	El usuario autenticado puede actualizar su información, cabe destacar que esta no incluye información sensible como identificación y correo electrónico. Esto se debe hacer como solicitud a un administrador.
Obtener sesiones	/auth/sesiones	GET	Obtiene un resumen de las sesiones que el usuario ha generado en la aplicación, incluyendo las sesiones

---

que se encuentran activas y las sesiones que no lo están.

Cerrar sesión	/auth/cerrar-sesion/{id}	DELETE	Cierra una sesión activa que se encuentre asociada al usuario que realizó la solicitud, hay que tener en cuenta que esta permite limitar que no se pueda cerrar la sesión desde donde se realizó la solicitud de cierre de sesión.
---------------	--------------------------	--------	--

---

*Nota.* Se definen las rutas encargadas de la seguridad y autorización

### ***Rutas para gestionar el resumen de datos***

Este grupo de rutas permiten acceder a un resumen de la información más importantes de la aplicación, para disminuir la carga de datos enviada desde las otras rutas.

#### **Tabla 20**

*Grupo de rutas para obtener resumen de datos*

Titulo	URL	HTTP	Descripción
Usuarios	/resumen/usuarios	GET	Obtiene un resumen de los usuarios registrados en la aplicación, limitando los registros mostrados.
Asignación Dependencias	/resumen/asignacion-dependencias	GET	Obtiene un resumen de la información de las asignaciones de dependencias que se han realizado.

---

Cargos	/resumen/cargos	GET	Obtiene un resumen de la información de los cargos de los peticionarios o ciudadanos.
Peticionarios	/resumen/peticionarios	GET	Obtiene el resumen de la información de los peticionarios registrados.
Dependencias	/resumen/dependencias	GET	Obtiene el resumen de la información de las dependencias registradas.
Entidades	/resumen/entidades	GET	Obtiene el resumen de la información de las entidades.
Invitaciones	/resumen/invitaciones	GET	Obtiene el resumen de la información de las invitaciones registradas.
Permisos	/resumen/permisos	GET	Obtiene el resumen de los permisos registrados en la aplicación.
Tipos Solicitud	/resumen/tipos- solicitud	GET	Obtiene el resumen de los tipos de solicitud registrados en la aplicación.

---

*Nota.* Se definen las rutas en las que se podrá obtener un resumen de los datos almacenados.

### ***Rutas para la gestión de los selectores de datos***

Este grupo de rutas permite gestionar la información que se va a mostrar en los selectores de datos de la interfaz de usuarios, limitando la carga de información enviada.

**Tabla 21***Grupo de rutas para la gestión de selectores de información*

Título	URL	HTTP	Descripción
Dependencias	/select-data/dependencias	GET	Obtiene los datos necesarios para el selector de dependencias.
Permisos	/select-data/permisos	GET	Devuelve la información necesaria para gestionar los selectores de permisos de usuarios.
Usuarios	/select-data/usuarios	GET	Ruta que permite obtener los datos que se mostrarán en el control de selección de usuarios.
Ciudadanos	/select-data/ciudadanos	GET	Obtiene la información necesaria para el control select de la UI mostrando los ciudadanos en la aplicación.
Cargos	/select-data/cargos	GET	Obtiene la información necesaria para el selector de cargos dentro de la UI.
Entidades	/select-data/entidades	GET	Obtiene la información necesaria para mostrar en los select de tipos de solicitudes que se mostrarán en la UI.
Tipos Solicitud	/select-data/tipos-solicitud	GET	Obtiene la información necesaria para los selectores de información relacionada a los tipos de solicitud a elegir en la UI.



Usuarios	/select-	GET	Obtiene la información de los
Dependencias	data/dependencias-		usuarios de la dependencia del
Asignar	usuarios-asignar		funcionario que realizó la solicitud, además de mostrar las dependencias que aún faltan por asignar al radicado definido.

---

*Nota.* Se definen las rutas que obtendrán la información para los selectores de datos.

### ***Rutas para gestionar las asignaciones de radicados***

Gestiona la información de las asignaciones de solicitudes a los funcionarios, permitiendo listar las solicitudes asignadas al usuario autenticado y obtener información de la solicitud asignada de manera específica.

#### **Tabla 22**

##### *Grupo de rutas para gestionar la asignación de radicados*

Titulo	URL	HTTP	Descripción
Listar	/asignacion-radicados	GET	Obtiene el listado de solicitudes
Asignación			asignadas al usuario que está
Radicados			autenticado y ha solicitado la información.
Obtener	/asignacion-	GET	Obtiene la información de una
Asignación	radicados/{id}		solicitud o radicado asignado a un funcionario, este valida si la asignación pertenece o no al usuario

---

que la solicitó, si no es así deniega el acceso a esta.

---

Nota. Se definen las rutas que permiten gestionar la asignación de radicados.

### ***Ruta para gestión de referencias***

Contiene el grupo de rutas que permite obtener valores de referencia para que la aplicación funcione correctamente.

#### **Tabla 23**

##### *Grupo de rutas para gestionar las referencias*

Titulo	URL	HTTP	Descripción
Número Siguiete Radicado	/referencias/siguiente- solicitud	GET	Obtiene el número de la siguiente solicitud que se registrará.
Obtener Fecha Limite	/referencias/fecha- limite	GET	Obtiene la información de la fecha límite desde la fecha actual hasta los días que se especifiquen en la solicitud, hay que tener en cuenta que este cálculo se realiza en días hábiles.

---

Nota. Se definen las rutas para obtener algunas referencias esenciales para la aplicación.

### ***Ruta para gestionar los modales***

Grupo de rutas de utilidades que permiten gestionar las acciones que realizan los diferentes modales de la aplicación, entre ellos los que gestionan acciones fundamentales de las asignaciones de radicados.

Titulo	URL	HTTP	Descripción
Actualizar Días Respuesta	/modal/solicitudes /actualizar-dias- respuesta	PATCH	Ruta que permite actualizar de manera específica los días para dar respuesta a una solicitud, lo cual actualiza de manera automática la fecha límite.
Actualizar Petionario	/modal/solicitudes /actualizar- petionario	PATCH	Ruta que actualiza en el radicado el petionario que la radicó.
Actualizar Cargo	/modal/solicitudes /actualizar-cargo	PATCH	Actualiza el cargo del petionario que realizó la solicitud, esto lo hace de manera parcial en función de los modales de UI.
Actualizar Entidad	/modal/solicitudes /actualizar-entidad	PATCH	Permite actualizar la entidad del petionario que realizó la solicitud, esto desde el modal implementado desde la UI.

Actualizar Destinatario	/modal/solicitudes /actualizar- destinatario	PATCH	Actualiza la información del destinatario de la solicitud a través del modal de la UI.
Actualizar Asignaciones Radicado	/modal/solicitudes /actualizar- asignaciones	PATCH	Permite actualizar las asignaciones de usuarios a un radicado, solo permite asignar usuarios de su misma dependencia, en caso de que requiera de otra dependencia debe enviar la dependencia a la que desea asignar y se les notificará a los usuarios administradores de esta.
Actualizar Documento Principal	/modal/solicitudes /actualizar- documento-principal	PATCH	Permite actualizar el documento principal asociado a una solicitud, este elimina el anterior y almacena el que se carga en la solicitud.
Cargar Documento Principal	/modal/solicitudes /cargar-documento- principal	PATCH	Permite cargar el documento principal de una solicitud, valida que no exista anteriormente ninguno cargado, de ser así almacena el archivo principal.
Cargar Adjuntos	/modal/solicitudes /cargar-documentos- adjuntos	PATCH	Permite cargar archivos adjuntos a una solicitud, estos se añaden a los demás archivos que tenga asociada la solicitud.

Actualizar CD	/modal/solicitudes /actualizar-cd	PATCH	Permite actualizar el número de CDs que se adjuntaron a una solicitud.
Eliminar Archivo Solicitud	/modal/solicitudes /eliminar-archivo- solicitud	PATCH	Permite eliminar un archivo de la solicitud, no importa si es documento principal o documento adjunto, este se elimina de la base de datos, del sistema de archivos del servidor y también de los registros asociados a la solicitud.
Eliminar Asignación	/modal/solicitudes /eliminar-asignacion	PATCH	Permite eliminar la asignación de un funcionario a un radicado específico.
Actualizar Encargado Radicado	/modal/solicitudes /actualizar- responsable-radicado	PATCH	Permite actualizar y definir quién será el funcionario encargado de dar respuesta a una solicitud entre los diferentes usuarios asignados a esta.
Generar Incidencia	/modal/solicitudes /generar-incidencia	PATCH	Permite generar una incidencia para un radicado específico de acuerdo con el usuario autenticado, validando si este está asociado o no al radicado que intenta agregarle esta incidencia.
Actualizar Incidencia	/modal/solicitudes /actualizar-incidencia	PATCH	Ruta que permite actualizar los datos de una incidencia, además permite

			cargar archivos adicionales a esta misma incidencia.
Eliminar Incidencia	/modal/solicitudes /eliminar-incidencia	PATCH	Permite eliminar una incidencia específica de acuerdo con su identificador, además elimina los archivos asociados a este en el sistema de archivos y en la base de datos.
Eliminar Archivo Incidencia	/modal/solicitudes /eliminar-archivo-incidencia	PATCH	Ruta que permite eliminar un archivo asociado a una incidencia generada por un funcionario, éste valida que solo quien creó la incidencia puede hacer esto y si pasa las validaciones el archivo se elimina del sistema de archivos y de la base de datos.

---

*Nota.* Se definen las rutas que permiten gestionar los datos desde los modales

### ***Ruta para gestión del timeline***

Contiene la ruta que permite obtener la información detallada de cada paso por el que ha pasado un radicado específico, cada una de sus acciones, con su respectiva fecha, los cuales se registran hasta que se da respuesta al mismo.

**Tabla 24***Ruta para obtener el timeline de un radicado*

Título	URL	HTTP	Descripción
Actualizar	/timeline	GET	Obtiene la información necesaria para
Días	/?code={code}		generar la línea de tiempo de un
Respuesta	&year={año}		radicado.

*Nota.* Se define la ruta que obtiene la línea de tiempo de las acciones

***Rutas para la gestión de copias de seguridad***

Grupo de rutas que permiten crear una copia de seguridad de los archivos y la base de datos de la aplicación, además de permitir cargar estas copias dentro del sistema.

**Tabla 25***Grupo de rutas para gestionar las copias de seguridad*

Título	URL	HTTP	Descripción
Copia	/backups/base-de-	GET	Ruta que permite generar una copia
Seguridad	datos		de seguridad de la base de datos de la
Base de Datos			aplicación para que la descargue el
			usuario que tenga permisos para esta
			acción.
Copia	/backups/archivos-	GET	Ruta que permite obtener la copia de
Seguridad	cargados		seguridad de los archivos de la
Archivos			aplicación que se hayan cargado como
			imágenes de perfil y de los radicados.

Restaurar Base de Datos	/backups/restaurar-base-de-datos	PUT	Permite restaurar una copia de seguridad de la base de datos, cargando el archivo comprimido con la información que se desea restaurar, hay que tener en cuenta que fallos en la información base de la aplicación llevará a una falla que evitará este tipo de acciones.
Restaurar Archivos	/backups/restaurar-archivos-cargados	PUT	Realiza la restauración de los archivos cargados a una aplicación teniendo en cuenta los archivos que se hayan subido a esta, el archivo debe conservar su información de descarga y formato para que sea válido.

---

Nota. Se definen las rutas para poder generar y restaurar las copias de seguridad.



## Desarrollo del Servicio API

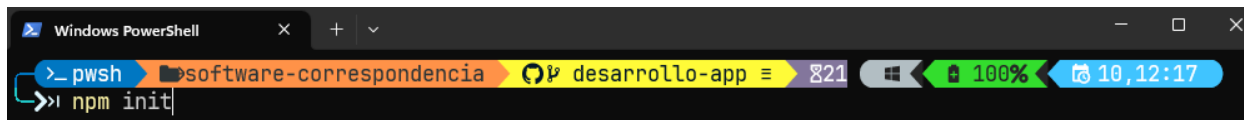
Con la definición del modelo de base de datos y cómo se procesará la información del lado del servidor en el servicio API, el siguiente paso es realizar la configuración del proyecto, instalación de dependencias y codificación del código base del servicio, este se realiza con TypeScript el cual es un lenguaje de programación desarrollado por Microsoft que una vez finalizado el proyecto se compila a JavaScript, su uso se debe a que nativamente el lenguaje de programación de JavaScript no es fuertemente tipado, es decir que el tipo de dato que almacenan las variables puede ser mutado sin ningún problema, causando problemas que con ayuda de TypeScript se pueden prevenir y solucionar en desarrollo y no hasta producción.

### Configuración del proyecto

Node JS tiene su herramienta y gestor de paquetes denominada NPM, con el cual además de poder instalar o desinstalar paquetes se pueden crear sus proyectos de manera rápida y sencilla, con el comando “npm init” se configura el proyecto determinando la versión, el nombre de la aplicación, tipo de licencia, descripción del proyecto.

### Figura 80

*Comando de inicialización del proyecto API*



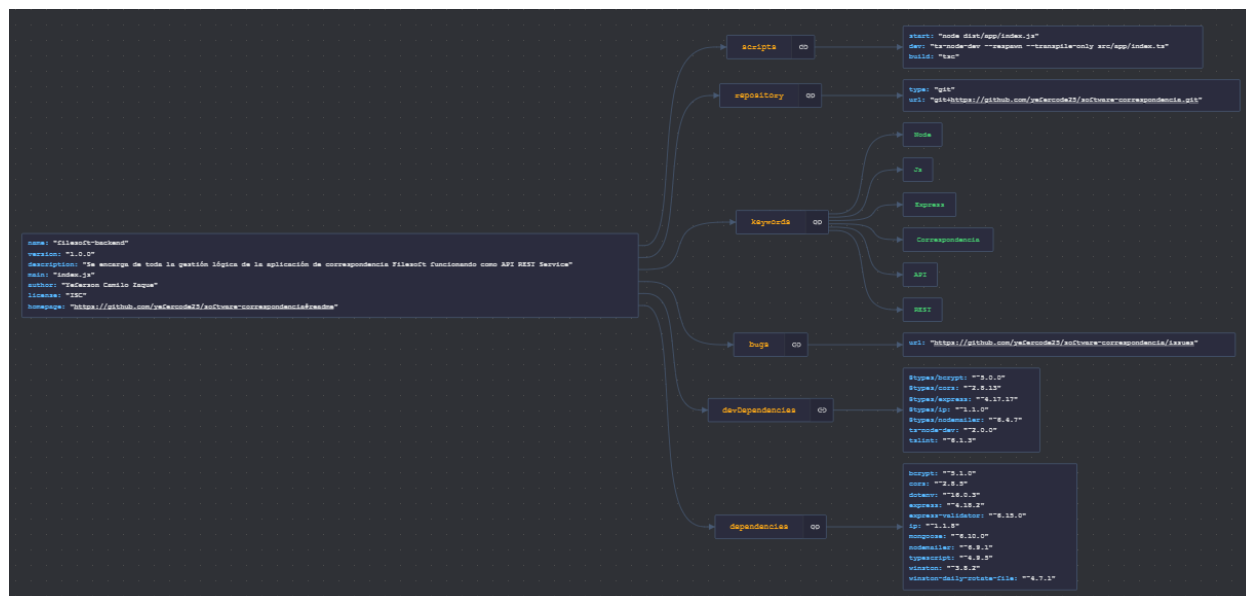
*Fuente.* Windows Power Shell

El resultado es un archivo denominado “package.json” que contiene toda la información del proyecto, incluyendo las librerías de desarrollo o de producción que se instalen, los comandos de ejecución y demás información necesaria para la ejecución de este, su mayor

ventaja es que por ejemplo si pasamos nuestra aplicación a producción, no es necesario subir todas las librerías instaladas debido a que con el comando “npm install” reinstala todo lo necesario.

## Figura 81

*Representación del archivo package.json del proyecto filesoft*



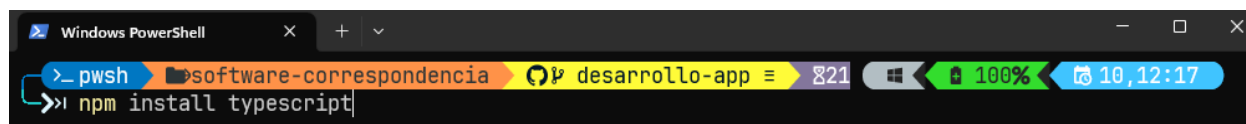
*Fuente.* Elaboración propia en JSON Crack Extension.

## Configuración de TypeScript

Su configuración requiere de su instalación a través del comando “npm install typescript” el cual descarga todo lo necesario para su entorno de ejecución en el proyecto.

## Figura 82

*Ejecución del comando de instalación de la dependencia typescript*

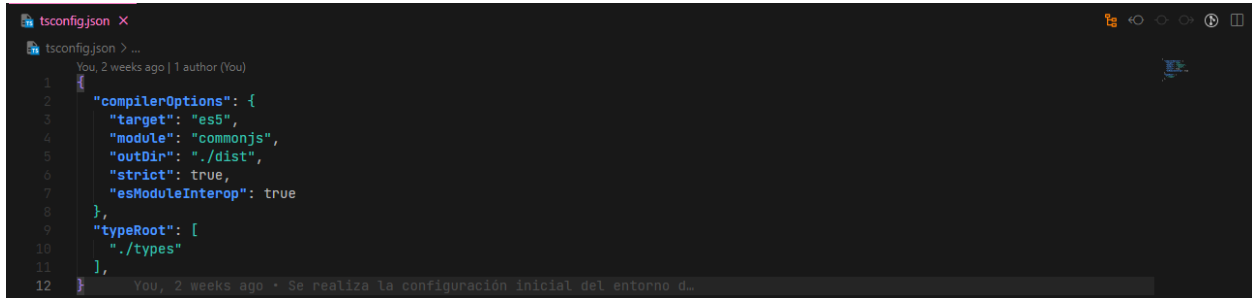


*Fuente.* Windows Power Shell

El segundo paso es configurarlo de tal manera que interprete los archivos del proyecto y los transpile a el lenguaje de programación JavaScript, para esto se crea un nuevo archivo denominado “tsconfig.json” que contiene las opciones del compilador, tipados y excepciones.

### Figura 83

*Contenido del archivo tsconfig.json*



```
tsconfig.json
You, 2 weeks ago | 1 author (You)
1 {
2   "compilerOptions": {
3     "target": "es5",
4     "module": "commonjs",
5     "outDir": "./dist",
6     "strict": true,
7     "esModuleInterop": true
8   },
9   "typeRoot": [
10    "./types"
11  ],
12 }
```

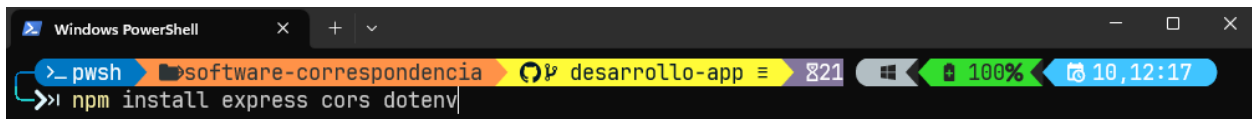
*Fuente.* Visual Studio Code

### Configuración de la aplicación

Es la base de toda la aplicación debido a que es la que cargará todas las rutas, middlewares, controladores, servicios y demás elementos que permitan el funcionamiento de la aplicación, para esto se requiere express que es el marco de trabajo principal, dotenv que ayuda a configurar gestionar las variables de entorno y cors que permite limitar el acceso de la API.

### Figura 84

*Comando de instalación de las librerías base de la aplicación*



```
Windows PowerShell
>_ pswsh
>> npm install express cors dotenv
```

*Fuente.* Windows Power Shell

Teniendo instaladas todas estas librerías se procede a generar la base de la aplicación que es el punto de partida y que ejecutará Node JS para poner en marcha el servicio, este archivo contendrá middlewares, rutas, validaciones de inicio de la aplicación y las rutas a las que responderá la API, para lograr lo que se muestra a continuación.

## Figura 85

### *Código base de la aplicación*

```

47  /**
48  * Middleware CORS: Configuración de Cross Origin
49  * Middleware Body Parse: Establece el objeto 'body' de la petición
50  * Middleware IP Client: Establece la IP del cliente en el objeto 'body'
51  */
52  app.use(cors());
53  app.use(express.json());
54  app.use(ipClient.setIPClient);
55
56  /**
57  * - Rutas de autenticación (/api/auth)
58  */
59  app.use('/api/auth', authRouter);
60
61  /**
62  * Establecimiento del servidor en estado de escucha, punto inicial
63  * de la aplicación.
64  */
65  app.listen(port as number, () => {
66    const msg = `El servidor está corriendo en el puerto ${port} (http://localhost:${port})`;
67    logger.info(msg);
68  });

```

*Fuente.* Visual Studio Code

### *Integración de servicio de logger*

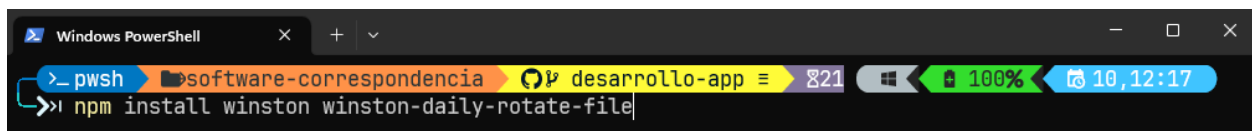
El servicio es el encargado de registrar todos los eventos que se realicen en la aplicación y que como desarrollador se decide almacenar, como la información de las solicitudes, los errores que se presenten y en que módulos de esta se generaron; los archivos de log son creados cada día y al cumplir los 30 días se comprimen los más antiguos, en cada log registrado se almacena la fecha, el tipo de log y los detalles de este, por ejemplo “2023-02-27T01:15:42.172Z info: El servidor está corriendo en el puerto 5454 (<http://localhost:5454>)”

Esto se logra con el paquete de “winston” y “winston-daily-rotate-file” que facilitan su gestión y solo es necesario crear una función que sea posible importarla dentro de cualquier

módulo de la aplicación, que para hacer reutilización de este código se coloca dentro de una función de utilidad que se podrá importar en cualquier archivo de la aplicación para registrar los eventos más importantes de la aplicación, en este caso para optimizar se opta por registrar únicamente los errores que se hayan presentado en la aplicación, accesos denegados, solicitudes sin permisos y demás eventos que presenten un posible problema para la aplicación.

## Figura 86

*Comando de instalación de herramientas para logger*

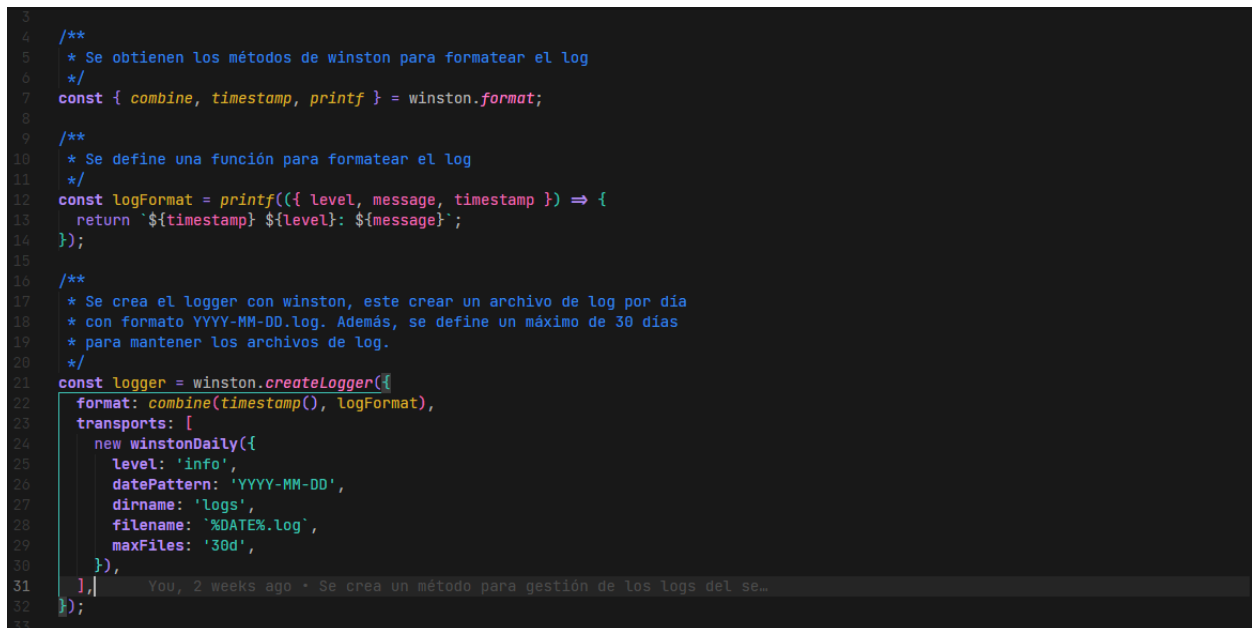


```
Windows PowerShell
>_ pwsh
>> npm install winston winston-daily-rotate-file
```

*Fuente.* Windows Power Shell

## Figura 87

*Configuración de la función logger*



```

3
4  /**
5   * Se obtienen los métodos de winston para formatear el log
6   */
7   const { combine, timestamp, printf } = winston.format;
8
9   /**
10  * Se define una función para formatear el log
11  */
12  const logFormat = printf(({ level, message, timestamp }) => {
13    return `${timestamp} ${level}: ${message}`;
14  });
15
16  /**
17  * Se crea el logger con winston, este crear un archivo de log por día
18  * con formato YYYY-MM-DD.log. Además, se define un máximo de 30 días
19  * para mantener los archivos de log.
20  */
21  const logger = winston.createLogger({
22    format: combine(timestamp(), logFormat),
23    transports: [
24      new winstonDaily({
25        level: 'info',
26        datePattern: 'YYYY-MM-DD',
27        dirname: 'logs',
28        filename: '%DATE%.log',
29        maxFiles: '30d',
30      }),
31    ],
32  });
33

```

*Fuente.* Visual Studio Code

### ***Configuración de variables de entorno***

Estas son variables que se configuran en el sistema operativo o en el servidor donde se ejecuta la aplicación, son utilizadas para almacenar información sensible o información de configuración que se utiliza en la aplicación.

Se pueden acceder utilizando el objeto `process.env` de Node JS y contiene todas las variables de entorno definidas en el sistema operativo o servidor, para acceder a una variable de entorno específica, se puede utilizar la sintaxis `process.env.NOMBRE_VARIABLE`.

**Tabla 26**

#### *Variables de entorno de la aplicación*

Variable	Descripción
APP_PORT	Es el puerto en el que se ejecutara la aplicación.
APP_EMAIL_USER	Es el correo electrónico que se registrará por primera vez en la aplicación y desde el cual se harán todas las gestiones como envío de notificaciones.
APP_EMAIL_PASSWORD	Es la contraseña del correo electrónico principal de la aplicación.
APP_EMAIL_GOOGLE_PASSWORD	Es la contraseña de la aplicación en Google, debido a que se usará el servicio SMTP del correo de Gmail pero es la contraseña que se usará para el correo.

APP_EMAIL_HOST	Es el host del correo electrónico principal de la aplicación.
APP_EMAIL_PORT	Es el puerto del correo electrónico principal de la aplicación.
APP_EMAIL_ENCRYPTION	Es el tipo de encriptación del correo electrónico principal de la aplicación.
APP_DB_MANAGER	Es el gestor de base de datos de la aplicación, por defecto se usa Mongo DB.
APP_DB_URL	Es la url de la base de datos de la aplicación, esta por lo general incluye el usuario la contraseña, el servidor, el puerto y el nombre de la base de datos a la que se desea acceder.
APP_ADMIN_EMAIL	Correo electrónico del administrador del sistema y que recibirá el correo de registro una vez se haya inicializado la aplicación por primera vez.
APP_ADMIN_NAME	Nombre del usuario administrador que funcionará como referencia para las notificaciones.
APP_FRONTEND_URL	Establece la ruta en la que se encuentra alojado el servicio frontend de la aplicación.
APP_JWT_SECRET	Es una cadena de texto que actuará como contraseña para encriptar y desencriptar los

---

tokens de acceso, es muy importante para la seguridad que se evite exponerla pues los tokens podrían ser firmados por tercer y hacerlos pasar cómo válidos.

NODE\_ENV

Define en que estado está la aplicación, si en desarrollo o en producción.

---

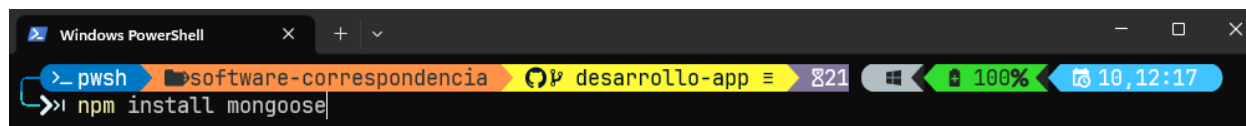
*Nota.* Se detallan las variables de entorno que se asociaran a la aplicación.

### ***Instalación y configuración del ORM***

El ORM a utilizar es Mongoose, este permite modelar y manipular los datos de MongoDB de una manera más intuitiva y sencilla, al proporcionar una interfaz basada en objetos para interactuar con la base de datos. Con Mongoose, se pueden definir esquemas para los documentos almacenados en MongoDB, lo que ayuda a asegurar que los datos sean coherentes y estructurados de manera consistente. Además, Mongoose también proporciona características útiles como validación de datos, middlewares y soporte para consultas complejas, lo que hace que trabajar con MongoDB sea más eficiente y fácil.

### **Figura 88**

*Comando de instalación de Mongoose*



```
Windows PowerShell
>_ pwsh C:\software-correspondencia\desarrollo-app 821
>> npm install mongoose
```

*Fuente.* Windows Power Shell

La configuración es sencilla, solo es ejecutar su función de conexión antes de que se ejecute la inicialización de la aplicación, esto ya nos da acceso a la creación de modelos,



generación de consultas, creación de documentos y registros entre otras cosas que nos proporciona este tipo de herramientas.

## Figura 89

### *Código de configuración de Mongoose ORM*

```

1  import mongoose from 'mongoose';
2  import { logger } from '../utils';
3
4  /**
5   * Realiza la conexión con la base de datos en MongoDB
6   * y muestra si la conexión fue exitosa o no.
7   */
8  const conectarMongoDB = async () => {
9    try {
10     await mongoose.connect(process.env.APP_DB_URL as string);
11     logger.info(`Base de datos conectada en ${mongoose.connection.host}:${mongoose.connection.port}`);
12   } catch (error) {
13     logger.error(error);
14     process.exit(1);
15   }
16 }

```

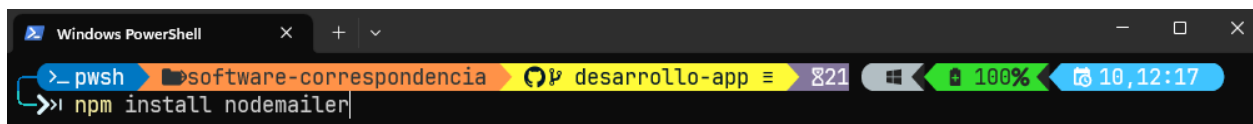
*Fuente.* Visual Studio Code

### *Configuración del servicio principal de emails*

La aplicación al tener la capacidad de enviar correos electrónicos en determinados eventos debe configurarse con ayuda del paquete de “nodemailer” el cual conecta con un servicio de correo electrónico a través de sus credenciales, en el caso del proyecto se utiliza una cuenta de Gmail para su configuración.

## Figura 90

### *Instalación del paquete de nodemailer*



```

Windows PowerShell
>_ pwsh
>> npm install nodemailer

```

*Fuente.* Visual Studio Code

## Figura 91

*Código de configuración del servicio de email con nodemailer*

```

5  /**
6  * Retorna el transportador de correo electrónico para el administrador
7  * que gestionará los correos electrónicos de la aplicación.
8  *
9  * @returns {Transporter} Transportador de correos electrónicos
10 */
11 const getTransporterStaticEmail = () => {
12   return nodemailer.createTransport({
13     host: process.env.APP_EMAIL_HOST as string,
14     port: process.env.APP_EMAIL_PORT as unknown as number,
15     secure: false,
16     auth: {
17       user: process.env.APP_EMAIL_USER as string,
18       pass: process.env.APP_EMAIL_GOOGLE_PASSWORD as string
19     }
20   });
21 };

```

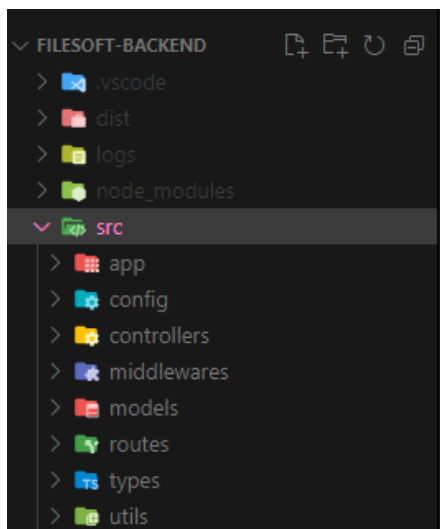
*Fuente.* Visual Studio Code

## *Estructuración de carpetas*

El proyecto seguirá una arquitectura donde se separen en módulos todas sus funcionalidades, esto siguiendo patrones de arquitectura modernos que facilitaran su gestión, siendo así un proyecto sencillo de escalar y de implementar nuevas funcionalidades.

## Figura 92

*Estructura básica de carpetas del proyecto*



*Fuente.* Visual Studio Code File Explorer

**Directorio Dist.** Fue el directorio que se le dijo a TypeScript que creara para almacenar la transpilación del código, es decir es el código fuente final de la aplicación y el que se debe enviar a producción.

**Directorio Logs.** Almacena todos los archivos de registros de eventos que la aplicación va generando a lo largo de su vida útil, es fundamental para ver cómo se comporta, que solicitudes se realizaron y en especial conocer cuáles fueron los errores que se han generado.

**Directorio App.** Almacena los archivos base de la aplicación para inicializar la aplicación.

**Directorio Config.** Almacena todos los archivos de configuración para servicios externos y esenciales, por ejemplo, la base de datos, el servicio de correo electrónico, entre otros que ayudarán a comportarse de manera correcta la aplicación.

**Directorio Controllers.** Contiene todo el código de lógica de negocio para cada una de las rutas establecidas para la API, estos son los que reciben la solicitud, la procesan y regresan una respuesta al cliente.

**Directorio Middlewares.** Contiene funciones fundamentales para el funcionamiento de la aplicación, entre ellas las funciones de autenticación y seguridad, validación de solicitudes, registros de eventos y otras más que se ejecutarán antes que el controlador principal de cada ruta.

**Directorio Models.** Contiene los modelos de la base de datos, en este es donde se exportan cada uno de ellos para ser usados en otra parte de la aplicación para crear registros, consultarlos, actualizarlos, eliminarlos, etc.

**Directorio Routes.** Almacena los archivos que definen cada una de las rutas válidas dentro de la aplicación, cualquier ruta sin definir no dará alguna respuesta al cliente si no presentará errores, por esta razón en este directorio se definen una a una las rutas válidas.

**Directorio Types.** En si no es fundamental para la aplicación, pero si para el desarrollo de esta debido a que almacena interfaces de código que representan la estructura de los objetos, modelos o variables dentro de la aplicación.

**Directorio Utils.** Presenta diferentes funciones para ser usadas dentro de la aplicación, ya sea para trabajar con textos, números, fechas, etc. Es una colección de utilidades que facilitarán el desarrollo y puesta en marcha de servicios.

## Configuración de modelos de datos

Cada uno de los modelos de datos establecidos para la base de datos se representa de manera que sea posible gestionarlos a través del lenguaje de programación TypeScript, esto a través de la definición de interfaces, implementación del ORM e implementación de funciones especiales para cada uno de ellos.

### *Definición de interfaces*

TypeScript al permitir tipar estrictamente cada uno de los objetos y variables recomienda crear una interfaz o tipo para cada modelo, esto con el fin de facilitar el desarrollo y lectura del código base de la aplicación.

## Figura 93

### *Ejemplo de definición de interfaz y tipado de modelo*

```

8  /**
9   * Interfaz de archivos que registra y asocia los archivos almacenados
10  * con los diferentes modelos de la aplicación.
11  */
12  You, 2 weeks ago | 1 author (You)
13  export interface IFileModel extends mongoose.Document {
14    fileName: string;
15    path: string;
16    relativePath: string;
17    type: FileType;
18    extension: string;
19    size: number;
20  }
21  You, 2 weeks ago • Se ha creado el modelo de archivos

```

*Fuente.* Visual Studio Code

### *Definición de esquema de modelo*

El esquema del modelo será el encargado de especificarle al ORM como se comportará el modelo, cuál será el nombre de cada campo, el tipo de datos que almacenará y las restricciones que debe tener en cuenta al trabajar con los registros, la figura 94 representa el esquema del modelo de “File” al cual con la sintaxis <Interfaz> se le asigna el tipado que debe cumplir.

### **Figura 94**

*Ejemplo de definición de esquema para modelo de datos*

```

17 const FileSchema = new Schema<IFileModel>({
18   fileName: {
19     type: String,
20     required: true,
21   },
22 > path: {--
25   },
26 > relativePath: {--
29   },
30 > type: {--
34   },
35 > extension: {--
38   },
39 > size: {--
42   },
43 }, { timestamps: true });

```

*Fuente.* Visual Studio Code

### *Documentación del esquema*

Siguiendo las buenas prácticas de programación se especifica una descripción del esquema, las variables que tendrá, el tipo de dato y lo que es cada una de las variables dentro del modelo que se definirá.

### **Figura 95**

*Ejemplo de documentación de esquema*

```

6  /**
7   * Esquema de archivos que registra y asocia los archivos almacenados
8   * con los diferentes modelos de la aplicación.
9   *
10  * @param {string} fileName Nombre del archivo
11  * @param {string} path Ruta absoluta del archivo
12  * @param {string} relativePath Ruta relativa del archivo
13  * @param {string} type Tipo de archivo dentro de la petición
14  * @param {string} extension Extensión del archivo
15  * @param {number} size Tamaño del archivo en bytes
16  */

```

*Fuente.* Visual Studio Code

### ***Definición y exportación del modelo***

El último paso es definir el modelo y exportarlo para que pueda ser accedido desde cualquier parte de la aplicación, gracias a que JavaScript permite modularizar y exportar un elemento sin ser definido dentro de una variable.

### **Figura 96**

#### *Ejemplo de definición y exportación de modelo*

```
45  /**
46  * Exporta el modelo de archivos para ser utilizado en la aplicación
47  * y que sea referenciado por otros modelos.
48  */
49  export default mongoose.model<IFileModel>('File', FileSchema);
```

*Fuente.* Visual Studio Code

### **Gestión de permisos de aplicación**

La parte principal y fundamental de la aplicación es la seguridad y se busca brindar la mayor seguridad posible limitando el acceso a los datos en base a los permisos que cada usuario tenga, en base a lo anterior se opta por no crear roles de acceso si no modularizar los permisos, es decir que, si en la aplicación existen los radicados, existirán cuatro permisos base que son la lectura, creación, actualización y eliminación de cada uno de ellos, siguiendo este método se definen todos los permisos necesarios para la aplicación y que se asocian con su respectivo modelo de datos.

#### ***Definición de permisos***

A continuación, se presentan todos los permisos que requiere la aplicación para su funcionamiento y que se cargarán cuando se inicie la aplicación por primera vez, esto se explica más a detalle en la sección de inicialización de permisos.

**Tabla 27***Permisos de la aplicación*

Permiso	Descripción
CREATE_USER	Permiso para crear un nuevo usuario.
READ_USER	Permiso para leer información de un usuario.
UPDATE_USER	Permiso para actualizar información de un usuario.
DELETE_USER	Permiso para eliminar un usuario específico.
CREATE_INVITATION	Permiso para crear una invitación y enviarla a un nuevo funcionario.
READ_INVITATION	Permiso para leer la información de una invitación enviada a un funcionario.
UPDATE_INVITATION	Permiso para actualizar la información de una invitación enviada a un funcionario.
DELETE_INVITATION	Permiso para cancelar una invitación enviada a un funcionario.
READ_PERMISSION	Permiso para leer la información de un permiso.
UPDATE_PERMISSION	Permiso para actualizar la información de un permiso.
READ_DEPENDENCY	Permiso para leer la información de una dependencia
CREATE_DEPENDENCY	Permiso para crear una dependencia en la aplicación.

UPDATE_DEPENDENCY	Permiso para actualizar la información de una dependencia.
DELETE_DEPENDENCY	Permiso para eliminar una dependencia de la aplicación.
READ_REQUEST_TYPE	Permiso para leer la información de un tipo de solicitud.
CREATE_REQUEST_TYPE	Permiso para crear un tipo de solicitud en la aplicación.
UPDATE_REQUEST_TYPE	Permiso para actualizar la información de un tipo de solicitud.
DELETE_REQUEST_TYPE	Permiso para eliminar un tipo de solicitud de la aplicación.
READ_PETITIONER	Permiso para leer la información de un peticionario.
CREATE_PETITIONER	Permiso para crear un peticionario en la aplicación.
UPDATE_PETITIONER	Permiso para actualizar la información de un peticionario.
DELETE_PETITIONER	Permiso para eliminar un peticionario de la aplicación.
READ_PETITIONER_POSITION	Permiso para leer la información de un cargo de peticionario.
CREATE_PETITIONER_POSITION	Permiso para crear un cargo de peticionario en la aplicación.

---



UPDATE_PETITIONER_POSITION	Permiso para actualizar la información de un cargo de peticionario.
DELETE_PETITIONER_POSITION	Permiso para eliminar un cargo de peticionario de la aplicación.
READ_ASSIGNED_DEPENDENCY	Permiso para leer la información de una dependencia asignada a un funcionario.
CREATE_ASSIGNED_DEPENDENCY	Permiso para crear una dependencia asignada a un funcionario.
UPDATE_ASSIGNED_DEPENDENCY	Permiso para actualizar la información de una dependencia asignada a un funcionario.
DELETE_ASSIGNED_DEPENDENCY	Permiso para eliminar una dependencia asignada a un funcionario.
READ_ENTITY	Permiso para leer la información de una entidad.
CREATE_ENTITY	Permiso para crear una entidad en la aplicación.
UPDATE_ENTITY	Permiso para actualizar la información de una entidad.
DELETE_ENTITY	Permiso para eliminar una entidad de la aplicación.
READ_FILE	Permiso para leer la información de un archivo subido a la aplicación.
CREATE_FILE	Permiso para crear subir un archivo a la aplicación.
UPDATE_FILE	Permiso para actualizar la información de un archivo subido a la aplicación.

---

DELETE_FILE	Permiso para eliminar un archivo subido a la aplicación.
VIEW_ALL_FILES	Permiso para acceder a todos los archivos subidos a la aplicación, sin importar su asignación.
READ_REQUEST	Permiso para leer la información de las solicitudes o radicados registrados en la aplicación.
CREATE_REQUEST	Permiso para crear solicitudes o radicados dentro de la aplicación.
UPDATE_REQUEST	Permiso para actualizar la información de una solicitud o radicado dentro de la aplicación.
DELETE_REQUEST	Permiso para eliminar una solicitud o radicado dentro de la aplicación.
NOT_ASSIGN_REQUEST	Este permiso evita que al usuario que lo posea no puedan asignarle solicitudes o radicados.
READ_ASSIGNED_REQUEST	Permiso para leer la información de una solicitud o radicado asignado a un funcionario.
CREATE_ASSIGNED_REQUEST	Permiso para crear una asignación de solicitud o radicado a un funcionario.
UPDATE_ASSIGNED_REQUEST	Permiso para actualizar la información de una asignación de solicitud o radicado a un funcionario.
DELETE_ASSIGNED_REQUEST	Permiso para eliminar una asignación de solicitud o radicado a un funcionario.

---

CREATE_BACKUP_DB	Permiso para crear una copia de seguridad de la base de datos.
CREATE_BACKUP_FILES	Permiso para crear una copia de seguridad de los archivos subidos a la aplicación.
UPLOAD_BACKUP_DB	Permiso para subir una copia de seguridad de la base de datos.
UPLOAD_BACKUP_FILES	Permiso para subir una copia de seguridad de los archivos subidos a la aplicación.

---

Nota. Se definen todos los permisos que se establecerán dentro de la aplicación.

### *Inicialización de permisos*

De manera predeterminada todos estos permisos se registran en la base de datos una vez se inicie la aplicación por primera vez, además después del primer inicio validará y si hay algún cambio se registrará en la base de datos de forma automática, este proceso lo realiza la aplicación como se muestra a continuación.

### **Figura 97**

#### *Código de validación de permisos al iniciar el proceso de la aplicación*

```
/**
 * Valida los permisos de los funcionarios y los registra en la base de datos si no existen.
 *
 * La función utiliza la librería 'texto' para formatear el nombre de cada permiso y luego los
 * compara con los permisos registrados en la base de datos. Si no hay permisos registrados, la
 * función inserta todos los permisos en la base de datos. Si hay permisos registrados, la función
 * compara los permisos nuevos con los permisos registrados y solo inserta los permisos nuevos en
 * la base de datos.
 *
 * @returns {Promise<void>} Una promesa que se resuelve cuando se han registrado los permisos de los funcionarios en la base de datos.
 */
const validarPermisos = async (): Promise<void> => {...
```

Fuente. Visual Studio Code

## Validación de permisos

Al ya estar inicializados los permisos se implementa la manera en que la aplicación valide si el usuario que realizó la solicitud de la información o realizar una acción concreta tenga los permisos necesarios, para realizarlo se implementa un Middleware el cual se implementa en cada ruta que se necesite proteger, este requiere una lista con los permisos que se deben validar que tenga el usuario, un valor de falso o verdadero que indique si se debe validar si el usuario necesita todos los permisos de la lista o al menos uno de ellos y por último un mensaje de error que se le envía al usuario en caso que no tenga los permisos establecidos.

### Figura 98

#### Middleware de validación de permisos

```
/**
 * Middleware que valida si el usuario tiene los permisos necesarios para realizar una acción.
 *
 * @param permisosRequeridosNecesarios Lista de permisos requeridos para realizar la acción.
 * @param validarTodosLosPermisos Indica si se deben validar todos los permisos requeridos o al menos uno.
 * @param mensajeDeError Mensaje de error que se devuelve si el usuario no tiene los permisos necesarios.
 * @returns Middleware que verifica si el usuario tiene los permisos necesarios.
 */
const validarPermisos = (
  permisosRequeridosNecesarios: IPermisosList[],
  validarTodosLosPermisos: boolean = false,
  mensajeDeError: string = RESPONSE_MESSAGE.message
) => {
```

Fuente. Visual Studio Code

## Gestión de sesiones y accesos

Al ser un software que manejará información sensible se implementa un modelo de seguridad con ayuda de los JSON Web Token (JWT) que se firman desde este servicio y se envían al servicio frontend, así cada vez que el cliente haga una petición, se deben incluir en las cabeceras de solicitud este token, si no las rutas que están protegidas denegarán el acceso como se muestra a continuación.

## Figura 99

*Respuesta de seguridad al no tener una sesión activa*

```
"code": 500,  
"message": "Se ha producido un error al validar la sesión del funcionario, es posible que el token de sesión no  
sea válido o que haya expirado, por favor vuelva a iniciar sesión",  
"errors": [],  
"result": null,  
"isSuccessful": false
```

*Fuente.* Respuesta del cliente Postman

Con esto se logra que el servicio sea capaz de identificar si el usuario tiene una sesión activa, la cual no almacena el servidor si no que directamente la almacena el cliente y lo único que necesita es este token que se envía en cada solicitud.

### ***Generación del token***

Este token es generado con ayuda de la librería de “jwt” la cual ayuda a generar de manera sencilla estos tokens que tendrán una vigencia de 24 horas, necesitando solo la información que se almacenará (payload) y la clave privada que se cargó en las variables de entorno y con la cual firmará el token y que se usará posteriormente para validar si la firma es válida, esto se logra implementado una utilidad que retorna una promesa debido a que la librería original no tiene soporte para asincronismo dando paso a posibles errores en su generación, esto debido a que no es capaz de esperar a que el token sea generado y pueda retornar información errónea.

## Figura 100

### Utilidad de generación de token de acceso

```
/**
 * Genera un token JWT para el usuario con el UID proporcionado.
 *
 * @param uid El UID del usuario para el que se generará el token.
 * @returns Una promesa que se resuelve con el token JWT generado.
 * @throws {Error} Si hay un error al generar el token.
 */
const generarJWT = (uid: string): Promise<string> => {
  return new Promise((resolve, reject) => {
    const payload = { uid };
    jwt.sign(payload, process.env.APP_JWT_SECRET || '', {
      expiresIn: '24h',
    }, (err, token) => {
      if (err) {
        logger.error('[generarJWT] Error al generar el token de acceso.');
```

Fuente. Visual Studio Code

### Verificación del token de sesión

Este se crea en conjunto con la anterior utilidad y requiere del token de sesión que se haya enviado desde el cliente, una vez se tenga el token se válida verificando que no haya expirado y que la firma del token coincida con la firma que se colocó en su generación, firma que se obtiene de las variables de entorno; si este proceso se lleva de manera correcta continua la aplicación y retorna el identificador del usuario que se usa dentro de la aplicación para otras actividades.

## Figura 101

### Utilidad para verificar los tokens de sesión

```
const validarJWT = (token: string): Promise<string> => {
  return new Promise((resolve, reject) => {
    try {
      jwt.verify(token, process.env.APP_JWT_SECRET || '', (err, decoded) => {
        if (err) {...
```

Fuente. Visual Studio Code

### ***Middleware para verificar la sesión***

Al tener las utilidades generadas se implementa un middleware que se agrega en cada una de las rutas que se necesita una sesión activa para acceder a la información, este se encarga de ver si la solicitud tiene una cabecera de autorización y si cumple el formato de bearer token, luego valida el token y obtiene el identificador del usuario que posteriormente se busca en la base de datos, si este existe y está todo correcto se busca la sesión del usuario y se agregan los permisos del usuario, el identificador el usuario y el de la sesión para que sean usadas cuando sean requeridas por algún controlador.

### **Figura 102**

*Código base del middleware que verifica la sesión de un usuario al realizar la solicitud*

```
/**
 * Middleware para validar el token de sesión de un usuario con ayuda de jsonwebtoken.
 *
 * @param {Request} req Petición HTTP.
 * @param {Response} res Respuesta HTTP.
 * @param {NextFunction} next Función para continuar con la ejecución del middleware.
 * @returns {Promise<Response | void>} Respuesta HTTP o void.
 */
const validarSesion = async (req: Request, res: Response, next: NextFunction) => {
  const { authorization } = req.headers;
```

*Fuente.* Visual Studio Code

### **Gestión de archivos**

La aplicación tiene la capacidad de recibir archivos y almacenarlos en su sistema de archivos, esto se logra con la librería “multer” la cual verifica si existe un archivo en la solicitud, lo mueve a una carpeta temporal y luego lo expone al controlador dentro de la solicitud; esto se puede aplicar de manera dinámica debido a que se limitan la cantidad de archivos que se pueden enviar, los formatos de archivos y el tamaño máximo que pueden pesar, en este caso se limita a 50Mb de carga por archivo y si una ruta requiere más peso se establece un parámetro para remover el límite definido.

## Figura 103

### *Ruta y base para cargar archivos al servidor*

```

/**
 * URL base para la carpeta de subida de archivos.
 */
const baseUrlUploads = `${__dirname}/../..../uploads`;

/**
 * Middleware para manejar la subida de un archivo utilizando `multer`.
 *
 * @param {string} paramFile Nombre del campo del formulario que contiene el archivo a subir.
 * @param {string[]} allowedFormats Formatos permitidos para subir el archivo.
 * @returns Middleware para subir un archivo utilizando `multer`.
 * @example
 * uploadSingleFileStorage('avatar');
 */
const uploadSingleFileStorage = (paramFile: string, allowedFormats: string[] = [], isLimit = true) => {...
};

```

*Fuente.* Visual Studio Code

Lo anterior se expone como un middleware que se usa cada vez que se requiera cargar un archivo, normalizando el nombre del archivo cargado quitando tildes, eliminando espacios y asignando un código único de cinco dígitos para evitar que existan dos archivos dentro de la carpeta de archivos temporales.

### *Acceso a los archivos*

Por motivos de seguridad los archivos no se pueden exponer libremente a través del servicio, para solucionarlo cada vez que se carga un archivo y se mueve a su respectiva carpeta, se almacena una referencia en la base de datos, con el nombre del archivo, la ruta en el servidor, la ruta relativa de acceso, el tipo de archivo cargado, la extensión y el tamaño de este, para posteriormente asociar el identificador de este a un registro y sea más fácil obtener la información.



## Figura 104

### *Estructura de la información del archivo almacenado*

```

_id: ObjectId('64796ccb1ce2238834b007a2')
fileName: "1tuaryn7vo-yeferson-zaque.png"
path: "D:\UNAD - Ingeniería de sistemas\software-correspondencia\filesoft-bac..."
relativePath: "uploads\pictures\1tuaryn7vo-yeferson-zaque.png"
type: "PROFILE"
extension: "image/png"
size: 328377
createdAt: 2023-06-02T04:15:07.501+00:00
updatedAt: 2023-06-02T04:15:07.501+00:00
__v: 0

```

*Fuente.* Software de Mongo DB Compass

Para exponer al usuario un archivo, este debe tener el identificador del archivo y también un token de acceso para archivos, este token se envía al cliente al iniciar la sesión y consta de una longitud de veinte caracteres aleatorios que se validan, verifican la sesión del usuario y envía el archivo al cliente para su visualización o descarga.

Este token es enviado a través de los parámetros de la URL como “fileAccessToken” y si este no está presente o no es válida la sesión, se regresa un mensaje de error indicando que no se tienen los permisos para visualizar el archivo, un ejemplo de URL es

<http://api/archivos/64932f110087f083deb9a38e?fileAccessToken=6kbVhTkw254g9r1hQbd8tOVcMCjZhh&download=true>, además se presenta el parámetro “download” que permite descargar el archivo directamente.

### **Inicialización del servicio por primera vez**

Al iniciarse el servicio por primera vez realiza la conexión a la base de datos para hacer validaciones si el usuario administrador está registrado al igual que los datos base de la aplicación para su funcionamiento como los permisos, los tipos de solicitud, las entidades y los cargos preestablecidas, registrando todo en el archivo de logs.

## Figura 105

### Registro de eventos al inicializar el servicio API

```

info: Base de datos conectada en 127.0.0.1:27017 {"timestamp":"2023-09-03T23:52:56.280Z"}
info: Se han registrado las dependencias en la base de datos. {"timestamp":"2023-09-03T23:52:56.527Z"}
info: Se han registrado los cargos de peticionarios en la base de datos. {"timestamp":"2023-09-03T23:52:56.592Z"}
info: Se han registrado los tipos de solicitud en la base de datos. {"timestamp":"2023-09-03T23:52:56.597Z"}
info: Se han registrado los permisos en la base de datos. {"timestamp":"2023-09-03T23:52:56.606Z"}
info: El servidor está corriendo en el puerto 5454 (http://localhost:5454) {"timestamp":"2023-09-03T23:52:56.626Z"}
info: El servidor está conectado a internet (Verificación de conexión con el servidor de correo electrónico) {"timestamp":"2023-09-03T23:52:56.647Z"}
info: El servidor está conectado a internet (Verificación de la cuenta de correo electrónico del administrador) {"timestamp":"2023-09-03T23:52:56.717Z"}
info: El servidor está listo para enviar correos electrónicos (applicationfilessoft@gmail.com) {"timestamp":"2023-09-03T23:52:57.884Z"}
info: Se ha enviado un correo electrónico de registro a zaquecamilo@gmail.com para que pueda registrarse en la aplicación {"timestamp":"2023-09-03T23:52:59.423Z"}
info: Se ha enviado un correo electrónico de registro a zaquecamilo@gmail.com para que pueda registrarse en la aplicación {"timestamp":"2023-09-03T23:52:59.425Z"}

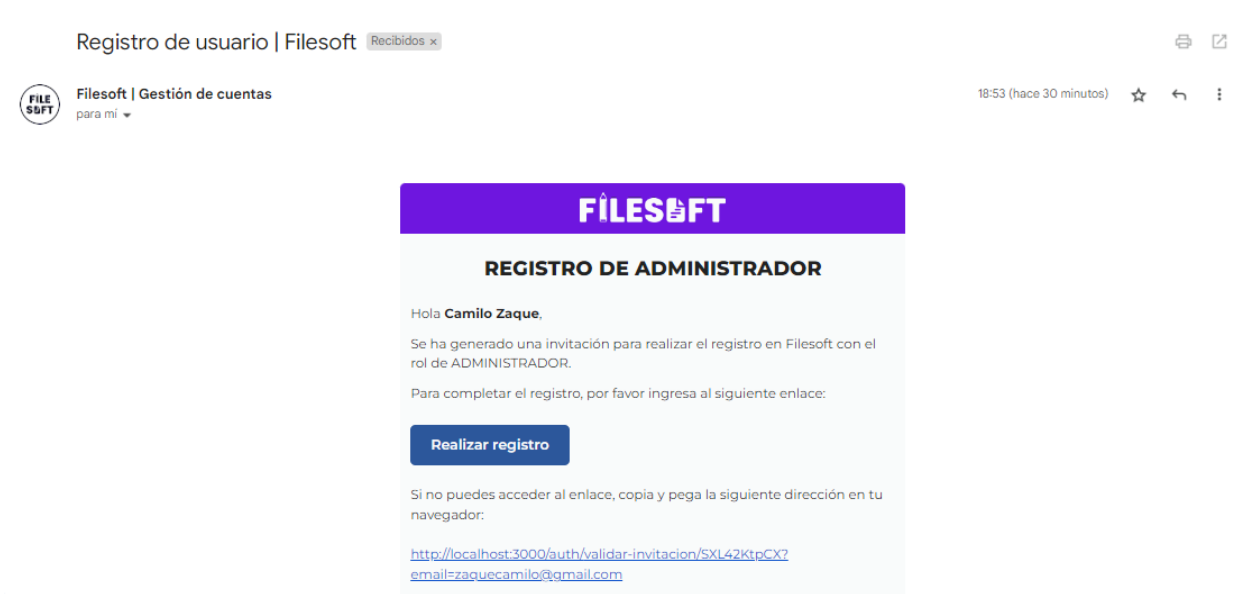
```

Fuente. Windows Power Shell

Además de esto si el usuario administrador no está registrado, se genera una invitación de registro automáticamente que llegará al correo con la información y el enlace que se debe seguir para completar el registro como administrador, el correo se presenta a continuación.

## Figura 106

### Correo electrónico enviado al usuario administrador al iniciar el servicio



Fuente. Correo recibido en bandeja de entrada de Gmail.

El registro del usuario se explica en el servicio frontend debido a que este servicio solo regresa información en formato JSON lo cual no es muy descriptivo en los mensajes y la interfaz en que se muestran.

## Gestión de fechas para radicados

Al tener que gestionar las fechas límite que los radicados tienen para dar respuesta, se implementa la librería de `day.js` que permite realizar algunas operaciones entre fechas de manera más sencilla, como formatear fechas, añadir días, establecer horas, etc. Todas estas esenciales para una correcta gestión y notificación a los usuarios, usando como base las fechas en formato ISO 8601 que al ser un estándar facilitan su manejo y reducen la posibilidad de errores.

### *Formatos de fechas*

Las fechas en formato ISO 8601 no son tan fáciles de entender para un usuario común ya que se representan como “2021-08-01T00:00:00.000Z” por lo que se genera una utilidad que tome este formato y lo regrese en uno más entendible como “01 de agosto de 2021, 00:00:00 a.m.” así permitiendo una lectura rápida.

## Figura 107

### *Utilidad para formateo de fechas*

```
/**
 * Convierte la fecha a un string con el formato indicado
 *
 * @param {boolean} incluirHora Indica si se debe incluir la hora en el formato
 * @returns {string} Fecha formateada en formato "dd de MMMM de yyyy, HH:mm:ss a"
 * @example 2021-08-01T00:00:00.000Z → "01 de agosto de 2021, 00:00:00 a.m."
 */
mostrarFormato: (incluirHora: boolean = true): string => {
  const dayjsDate = dayjs(fecha!);
  return dayjsDate.format('DD [de] MMMM [de] YYYY${incluirHora ? ', hh:mm:ss A' : ''});
},
```

Fuente. Visual Studio Code

### ***Añadir días laborales a fecha***

Las fechas que manejan las utilidades se manejan en base a días hábiles y laborales, por esta razón se toma una fecha base y los días laborales que se desean añadir, para lograrlo se genera un ciclo que vaya añadiendo día a día el día pero validando que no sea sábado o domingo, además se debe verificar que no sea un día festivo, para tener estas referencias se usa la librería de “date-holidays” que tiene validación si una fecha específica es día festivo, con estas herramientas se realiza el ciclo hasta finalizar y retorna la nueva fecha en formato ISO 8601 para ser procesada posteriormente.

### **Figura 108**

*Base de la utilidad para añadir días laborales a una fecha*

```
/**
 * Retorna en formato ISO la fecha que se envíe como parametro aumentando los días que
 * se especifiquen, estos días deben solo contar los días hábiles, es decir, de lunes a viernes
 *
 * @param {number} daysToAdd Cantidad de días hábiles a aumentar
 * @returns {string} Fecha en formato ISO
 * @example "2021-08-01T00:00:00.000Z", 5 → "2021-08-08T00:00:00.000Z"
 */
addBusinessDays: (daysToAdd: number): string => {
```

*Fuente.* Visual Studio Code.

### ***Calcular diferencia en días laborales entre dos fechas***

Esta es una utilidad generada para recibir dos fechas de las cuales calculará cuantos días laborales hay entre las dos fechas, además de definir si la fecha ya pasó incluyendo las horas, minutos y segundos que faltan para la hora de la fecha final, para lograrlo se usa la misma lógica que al añadir días laborales, es esta ocasión se genera un contador de días como referencia y se va sumando cuando el día sea diferente de sábado, domingo o día festivo.

## Figura 109

*Respuesta de la utilidad para obtener la diferencia en días laborales entre dos fechas*

```

134     "remainingTime": {
135         "days": 30,
136         "hours": 5,
137         "minutes": 52,
138         "seconds": 0,
139         "hasPassed": true
140     },

```

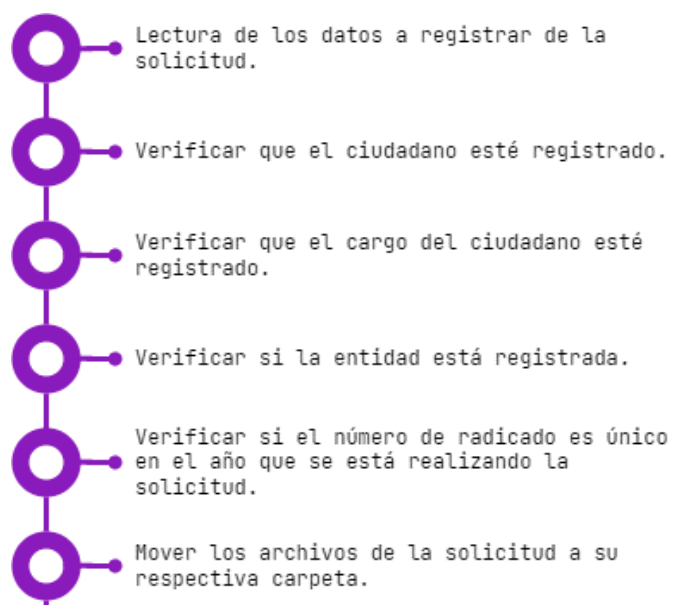
*Fuente.* Software de Postman

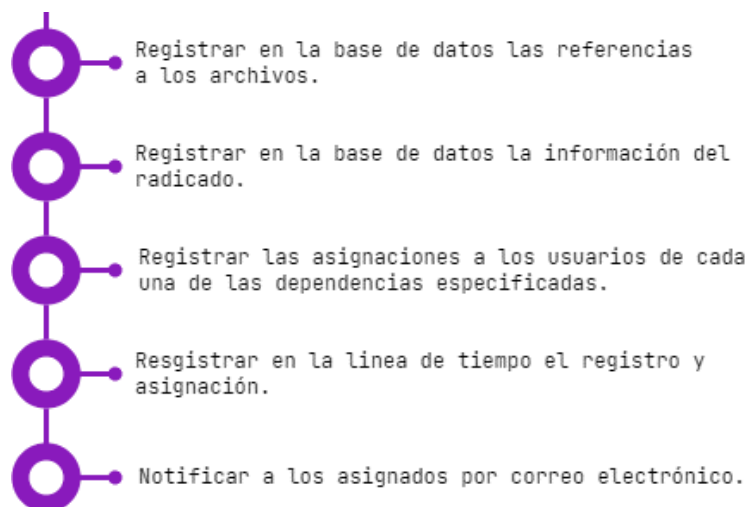
## Registro de radicados

En este servicio se maneja toda la lógica de negocio para registrar cada uno de los radicados, aunque no se maneje una manera visual desde el servicio son varias las acciones que se llevan a cabo para el registro de la información, entre ellas varias validaciones, generación de registros para radicados, archivos, líneas de tiempo, asignación y notificación por correo electrónico.

## Figura 110

*Pasos para la generación del radicado en el servidor*





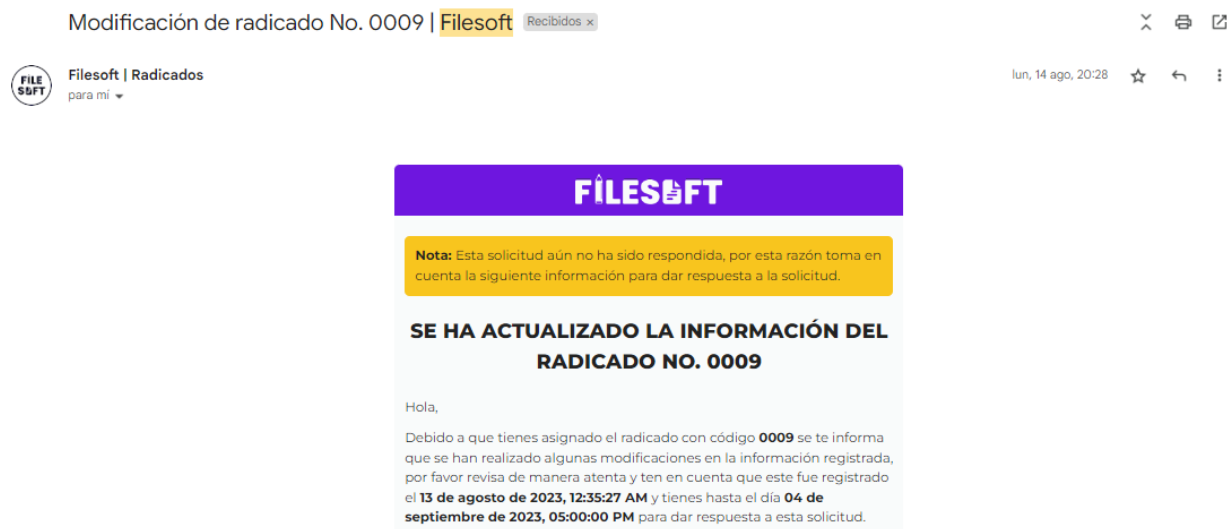
*Fuente.* Elaboración propia en Draw.io

Como se muestra en la anterior imagen son varios los pasos que debe realizar el servidor para registrar de manera correcta un radicado, esto se hace debido a que la gestión de radicados la función principal de la aplicación y un fallo en esta gestión podría traer errores de alto valor a la entidad, así que una vez realizado el registro se generan dos correos electrónicos, uno para el funcionario al que se le realizó la asignación y otro al ciudadano que realizó la solicitud.

Después para su lectura, actualización o eliminación se permiten diferentes peticiones de tipo HTTP desde el cliente, el cual con los suficientes permisos puede cargar archivos adjuntos, actualizar el documento principal, cambiar los días para dar respuesta, establecer un responsable, eliminar o agregar un funcionario, generar incidencias a un radicado específico y actualizar datos otros datos importantes del radicado, que tras su modificación se notifica por correo electrónico a todos los funcionarios asignados que se realizó la edición de este radicado y verifiquen la nueva información.

## Figura 111

*Correo electrónico que notifica la edición de un radicado*



*Fuente.* Correo recibido en bandeja de entrada de Gmail.

## Generación de copias de seguridad y restauración de datos

La aplicación para preservar la integridad de los datos y los archivos que se almacenan permite generar copias de seguridad de la base de datos y de los archivos que se han cargado en la aplicación, para posteriormente si se presenta un fallo sea posible restaurar estos datos a un punto anterior y así la información se mantiene de manera segura.

### *Copia de seguridad de base de datos*

Esta copia se genera con ayuda de una herramienta que genera un archivo en formato JSON por cada una de las colecciones que estén registradas en la base de datos, esto lo almacena en el servidor de manera independiente por lo que se usa el paquete de 7zip para comprimir la carpeta y enviarla al usuario que realizó la solicitud de copia de seguridad.

## Figura 112

### Contenido de la copia de seguridad descargada

Nombre	Tamaño	Comprimido	Tipo	Modificado	CRC32
..			Carpeta de archivos		
assigneddependencies.json	2.595	15.484	Archivo de origen ...	29/08/2023 3:4...	58F6E5B8
assignedrequests.json	1.012	?	Archivo de origen ...	29/08/2023 3:4...	D9684D43
dependencies.json	5.233	?	Archivo de origen ...	29/08/2023 3:4...	8FFF578A
entities.json	1.930	?	Archivo de origen ...	29/08/2023 10:...	A55EF426
files.json	23.645	?	Archivo de origen ...	29/08/2023 3:4...	20683792
incidentrequests.json	1.727	?	Archivo de origen ...	29/08/2023 3:4...	78EDEA9A
invitations.json	1.365	?	Archivo de origen ...	29/08/2023 3:4...	795E538C
permissions.json	13.769	?	Archivo de origen ...	29/08/2023 3:4...	C1349340
petitionerpositions.json	7.184	?	Archivo de origen ...	29/08/2023 3:4...	05EB0C48
petitioners.json	2.606	?	Archivo de origen ...	29/08/2023 3:4...	4350B5E6
references.json	2	?	Archivo de origen ...	29/08/2023 3:4...	0D4C8B29
requests.json	4.013	?	Archivo de origen ...	29/08/2023 3:4...	452F3CF8
requesttypes.json	2.999	?	Archivo de origen ...	29/08/2023 3:4...	DA287A92
sessions.json	25.033	?	Archivo de origen ...	29/08/2023 3:4...	7EF802CF
timelines.json	38.038	?	Archivo de origen ...	29/08/2023 3:4...	C76416E6
users.json	4.234	?	Archivo de origen ...	29/08/2023 3:4...	EB33987E

Fuente. Software de WinRAR.

### Copia de seguridad de archivos

Esta copia de seguridad se realiza integrando los archivos de fotos de perfil y archivos de radicados e incidencias, y los archivos temporales son eliminados antes de empezar a comprimir las otras carpetas, una vez finalizado el proceso, este archivo es enviado al usuario que realizó la solicitud y se almacena la copia temporalmente.

## Figura 113

### Contenido de la copia de seguridad de los archivos

Nombre	Tamaño	Comprimido	Tipo	Modificado	CRC32
..			Carpeta de archivos		
pictures	464.813	?	Carpeta de archivos	17/07/2023 3:4...	
requests	12.125.490	?	Carpeta de archivos	21/08/2023 10:...	

Fuente. Software de WinRAR.

### Restauración de copia de base de datos

Para realizar la restauración de los datos se requiere el archivo de base de datos y que este no haya sido alterado de ninguna manera, ni renombrado ya que se informará que no se permitirá



su carga en la aplicación, si todo sale correcto se descomprime el archivo en el servidor, se recorre uno a uno los archivos de los modelos y usando los modelos del ORM se va restaurando uno a uno los datos de la aplicación con su respectiva información.

### Figura 114

*Respuesta de restauración de base de datos*

```
1  [2]
2  "code": 200,
3  "message": "Se ha restaurado la copia de seguridad de la base de datos correctamente",
4  "errors": [],
5  "result": null,
6  "isSuccessful": true
7  [3]
```

*Fuente.* Software de Postman.

### *Restauración de archivos cargados*

Para realizar la restauración de los archivos de la aplicación se realizan las mismas validaciones que al restaurar la base de datos, una vez realizada la validación se procede a borrar toda la información que esté presente y tras su eliminación y se mueven las carpetas de la copia de seguridad hasta la carpeta de archivos cargados, finalizando de esta manera la restauración e informando al usuario.

### Figura 115

*Respuesta de restauración de archivos exitosa*

```
1  [2]
2  "code": 200,
3  "message": "Se restauró la copia de seguridad de los archivos correctamente",
4  "errors": [],
5  "result": null,
6  "isSuccessful": true
7  [3]
```

*Fuente.* Software de Postman.

## Automatización de tareas de mantenimiento y seguimiento

Para mantener el servicio en correcto funcionamiento, se programan diversas tareas que se ejecutan algunos días con el fin de liberar espacio en el servidor, eliminar incongruencias en los datos y mantener a salvo la información de la aplicación, esto se logra con ayuda de “node-cron” el cual es un gestor de tareas programadas al que se le define una hora específica y los días en los que debe ejecutarse.

### Figura 116

*Registro de tareas al iniciar la aplicación*

```
info: [TaskRegister] Se ha registrado la tarea para eliminar los archivos temporales subidos al servidor {"timestamp":"2023-09-08T00:44:35.773Z"}
info: [TaskRegister] Se ha registrado la tarea para eliminar las sesiones inválidas {"timestamp":"2023-09-08T00:44:35.781Z"}
info: [TaskRegister] Se ha registrado la tarea para verificar el tiempo restante de las solicitudes {"timestamp":"2023-09-08T00:44:35.781Z"}
```

*Fuente.* Windows Power Shell

### *Tarea de eliminación de archivos temporales*

Se encarga de eliminar todos los archivos temporales que se hayan quedado cargados en el servidor, estos archivos pueden quedarse tras posibles errores en la carga y creación de registros, por esta razón se programa que todos los sábados a medianoche se eliminen de manera definitiva, esto es esencial para liberar espacio en el servidor y para evitar que existan archivos que no estén asociados a ningún registro.

### Figura 117

*Ejecución de la tarea de eliminación de archivos temporales*

```
info: [StartClearTempFilesTask] Se ha iniciado la tarea para eliminar los archivos temporales subidos al servidor {"timestamp":"2023-09-07T17:15:00.830Z"}
info: [NotFoundTempFiles] no se ha encontrado archivos temporales subidos al servidor {"timestamp":"2023-09-07T17:15:00.834Z"}
```

*Fuente.* Windows Power Shell

### ***Tarea de eliminación de sesiones invalidas***

Se encarga de revisar en la base de datos los registros de sesiones que aún tengan un token, en base a esto se valida si la sesión está vigente, si no lo está, se elimina el token de sesión y de acceso a archivos de cada registro que no sea una sesión valida, esto se hace con el fin de evitar irregularidades y liberar la base de datos de información que ya no es necesaria; esta tarea se ejecuta todos los días a medianoche.

### **Figura 118**

#### *Ejecución de la tarea de eliminación de sesiones inválidas*

```

info: [StartClearInvalidSessionsTask] Se ha iniciado la tarea para eliminar las sesiones inválidas {"timestamp":"2023-09-07T17:43:00.171Z"}
error: [validarJWT] Error al validar el token de acceso. {"timestamp":"2023-09-07T17:43:00.199Z"}
info: [ClearInvalidSession] Se ha eliminado la sesión inválida con id: 64a1bc3195b476fe7c26a3e6 {"timestamp":"2023-09-07T17:43:00.243Z"}
error: [validarJWT] Error al validar el token de acceso. {"timestamp":"2023-09-07T17:43:00.245Z"}
info: [ClearInvalidSession] Se ha eliminado la sesión inválida con id: 64ea9cbd2b5bb1983ac87910 {"timestamp":"2023-09-07T17:43:00.249Z"}
error: [validarJWT] Error al validar el token de acceso. {"timestamp":"2023-09-07T17:43:00.251Z"}
info: [ClearInvalidSession] Se ha eliminado la sesión inválida con id: 64f80e159535601771303c09 {"timestamp":"2023-09-07T17:43:00.254Z"}
info: [EndClearInvalidSessionsTask] Se ha finalizado la tarea para eliminar las sesiones inválidas {"timestamp":"2023-09-07T17:43:00.256Z"}

```

*Fuente.* Windows Power Shell

### ***Tarea de notificación de tiempo restante para responder a radicados***

Se encarga de validar los radicados que aún no han tenido respuesta y enviar un correo electrónico a cada uno de los funcionarios asignados cuando para cumplirse la fecha límite falten siete, tres, dos y un día hábil para responder o la fecha límite ya haya sido superada; esta tarea se ejecuta de lunes a viernes a las 7:00 AM para que antes de iniciar la jornada laboral los funcionarios estén informados.

**Figura 119***Ejecución de la tarea para notificar el tiempo restante para responder un radicado*

```
info: [StartVerifyRemainingTimeTask] Se ha iniciado la tarea para verificar el tiempo restante de las solicitudes {"timestamp":"2023-09-08T00:23:00.721Z"}
info: [NotFoundAssignments] No se han encontrado asignaciones para la solicitud con id: 64c9e19b41048f0f330dce61 {"timestamp":"2023-09-08T00:23:01.451Z"}
info: [NotFoundAssignments] No se han encontrado asignaciones para la solicitud con id: 64d865fc1335b2b063072d45 {"timestamp":"2023-09-08T00:23:01.455Z"}
```

*Fuente.* Windows Power Shell

La automatización de estas tareas de mantenimiento y seguimiento desempeña un papel crucial en garantizar la eficiencia y la integridad del servicio, liberar espacio en el servidor, eliminar archivos temporales, sesiones inválidas, y notificar a los funcionarios sobre los radicados pendientes.

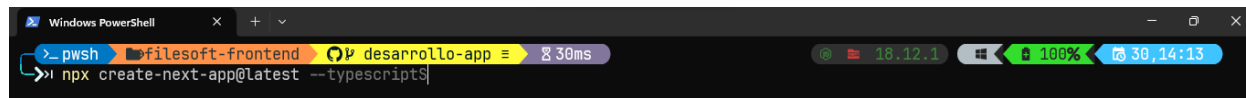
## Desarrollo del Servicio Frontend

### Creación del proyecto

El proyecto se desarrolla con el framework de Next JS el cual utiliza como base la librería de React para manejar la interfaz de usuario, eso implica que toda la aplicación actuará como una SPA o Single Page Application la cual se tratan de diferentes módulos que se van cargando a medida que se solicitan los recursos al servidor y evitan que las páginas tengan que recargarse al solo actualizar los componentes que se requieran, para crear este proyecto usa el comando:

### Figura 120

*Creación del proyecto de Next JS*



*Fuente.* Windows Power Shell

Este genera la estructura de carpetas base que incluye un directorio de páginas con su manejo de direccionamiento estático y dinámico que facilitan el desarrollo del servicio que se mostrará a los usuarios, además separa y permite agregar diferentes módulos de funciones para la gestión de las páginas como autenticación, consulta de datos al servicio API REST, creación de estados globales, integración de librerías externas, entre otras características que dan como resultado una aplicación optimizada y con buen rendimiento.

### Creación de rutas de aplicación

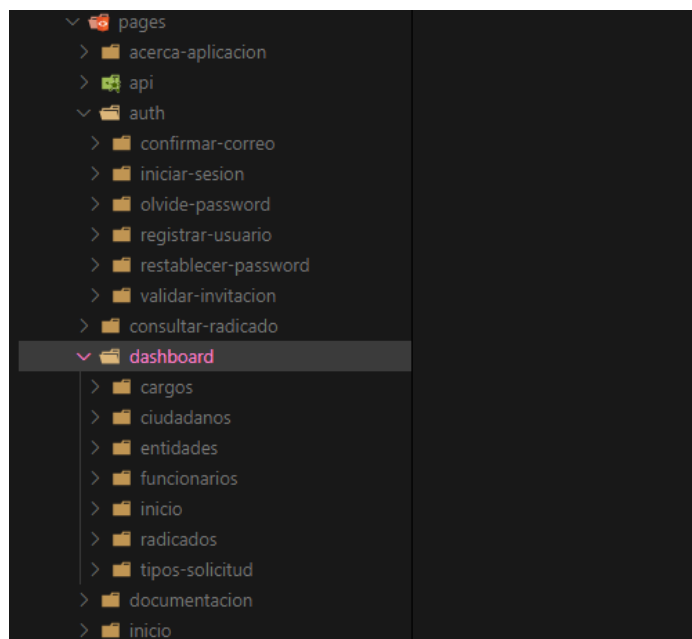
El Framework de Next JS maneja un direccionamiento dinámico y sencillo, el cual crea una ruta de aplicación por cada archivo que se agregue en el directorio de /pages, es decir que si

se quiere crear una ruta a la que se acceda con <http://localhost:3000/inicio> solo se debe crear un archivo con el nombre inicio.tsx o la otra posibilidad es crear un directorio y dentro de él agregar un archivo index dejando la ruta inicio/index.tsx, de esta manera se trabaja la creación e integración de nuevas rutas a las que puede acceder el usuario.

A continuación, se presenta la estructura base de las carpetas donde se integran las páginas públicas, las páginas de autenticación y la base para la creación del dashboard de administración.

### Figura 121

*Representación de la estructura de directorios para generar las rutas físicas de la aplicación*



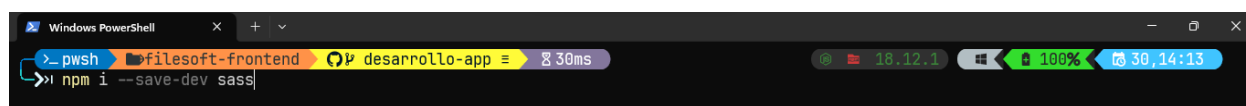
*Fuente.* Visual Studio Code

La continuación del desarrollo implica tomar el diseño generado en el software de Figma y con el lenguaje de etiquetas de HTML y siguiendo las buenas prácticas de React se crean diferentes componentes que se reutilizan a lo largo de la aplicación como por ejemplo botones, títulos, textos, enlaces y demás tipos de componentes que conforman una aplicación.

Por otra parte, la aplicación de los estilos se realiza con CSS3 con el intérprete de SASS el cual se instala dentro del proyecto para poder trabajar con la extensión “scss”, esto se realiza porque facilita la aplicación de los estilos y la especificidad que esta contiene, ya que dentro de un módulo se pueden agregar estilos más etiquetas sin tener que salir del bloque ni repetir los nombres de las clases, además que facilita la aplicación convenciones para nombrar cada una de las clases.

**Figura 122**

*Instalación de la dependencia de desarrollo de SASS*

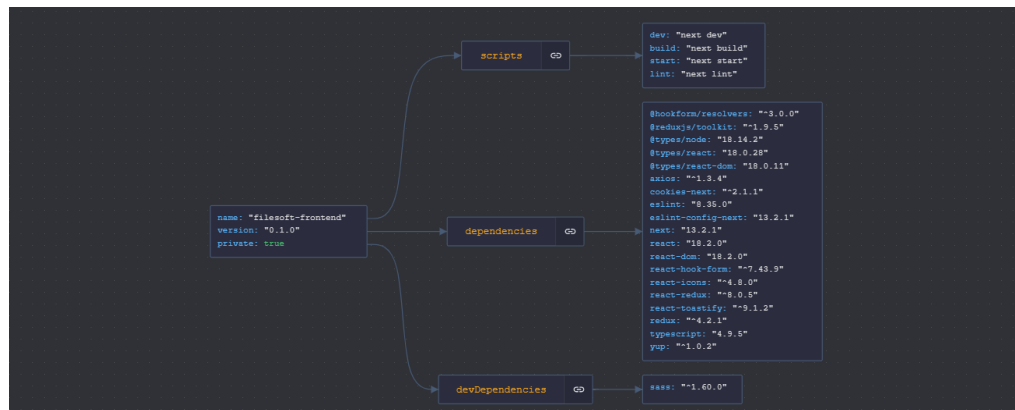


*Fuente.* Windows Power Shell

Además de la anterior existen dependencias propias del Framework y utilidades para gestionar la aplicación, estas se detallan dentro del archivo package.json el cual contiene la información del proyecto, las dependencias de producción y desarrollo, el tipo de licenciamiento y los scripts de ejecución que comenzarán la aplicación para producción o desarrollo.

**Figura 123**

*Estructura del archivo de configuración del proyecto*



*Fuente.* Extensión JSON Crack.

## Implementación del estado global (Redux)

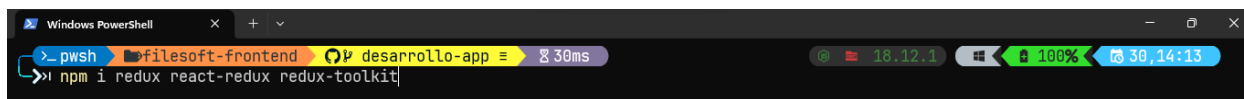
El estado global de la aplicación es aquel que almacenará datos de la aplicación como información de la sesión, del usuario autenticado, el listado de los radicados, el listado de usuarios, entre otros datos que son importantes para mantener en ejecución el servicio del cliente y pueda visualizar de manera correcta la información que envía el servicio API REST, para esto se utiliza la librería de Redux con sus herramientas de Redux Toolkit y React Redux que son necesarias para la aplicación de Next JS.

### *Instalación de herramientas*

La instalación se realiza por medio de la terminal ejecutando comandos recomendados en la documentación, siendo necesarios Redux, Redux Toolkit y React Redux para hacer la integración:

### **Figura 124**

#### *Instalación del gestor de estado global*



*Fuente.* Windows Power Shell

### *Creación del store*

El store de la aplicación es aquel que contiene toda la información de la aplicación, por esta razón es también conocido como la única fuente de la verdad debido a que, si requiero información para algún componente, se debe solicitar al store directamente, este se crea como una variable normal que se exporta por defecto para su implementación y al usar TypeScript se



exporta el tipado para del dispatch (Encargado de ejecutar las acciones que modifiquen el store) y para la aplicación en general.

## Figura 125

### *Configuración del store de la aplicación*



```

TS index.ts x
filesoft-frontend > src > redux > TS index.ts > default
You, 4 days ago | 1 author (You)
1 import { configureStore } from '@reduxjs/toolkit';
2 import authSlice from './auth-slice';
3 import uiSlice from './ui-slice/index';
4
5 const store = configureStore({
6   reducer: {
7     auth: authSlice,
8     ui: uiSlice,
9   },
10 });
11
12 export type AppDispatch = typeof store.dispatch;
13 export type RootState = ReturnType<typeof store.getState>;
14 export default store;
  
```

*Fuente.* Visual Studio Code

### *Creación de slice*

Los slice hacen referencia a un módulo de la aplicación, es decir que, si se almacenarán los usuarios y los radicados, existirá un slice para los usuarios y otro para los radicados, además estos contienen las funciones que se encargarán de cambiar el estado de estos módulos, por ejemplo, para leer, actualizar, eliminar, entre otras acciones que pueden provocar cambios en la información almacenada; a continuación, se presenta la estructura de un slice donde se le asigna un nombre, un estado inicial y las acciones que tendrá.

## Figura 126

*Configuración del store para manejar los datos de la ui*

```

1 import { PayloadAction, createSlice } from "@reduxjs/toolkit";
2 import { IUIMState } from "@types/redux/ui";
3 import { IMenuDispatch } from "@types/redux/ui";
4
5 const uiInitialState: IUIMState = {
6   loading: true,
7 };
8
9 const uiSlice = createSlice({
10  name: "ui",
11  initialState: uiInitialState,
12  reducers: {
13    setLoading(state, action: PayloadAction<boolean>) {
14      state.loading = action.payload;
15    },
16  },
17 });
18
19 export const { setLoading } = uiSlice.actions;
20
21 export default uiSlice.reducer;

```

*Fuente.* Visual Studio Code

## *Integración del estado global*

Con todo configurado, se realiza la integración en la aplicación usando React Redux el cual proporciona un componente llamado Provider que como su nombre lo indica es el encargado de proveer los datos globalmente a la aplicación, algo a tener en cuenta es que la información estará disponible dentro de todos los componentes que envuelva, por esa razón es fundamental ponerlo en lo más alto de la aplicación, en este caso en el archivo `_app.tsx` que es la base de la aplicación.

## Figura 127

*Integración del store con ayuda del provider de la librería de React Redux*

```

8 export default function App({ Component, pageProps }: AppProps) {
9   return (
10    <Provider store={store}>
11      <Component {...pageProps} />
12      <ToastContainer
13        closeOnClick
14      />
15    </Provider>
16  );
17 }
18
19 }

```

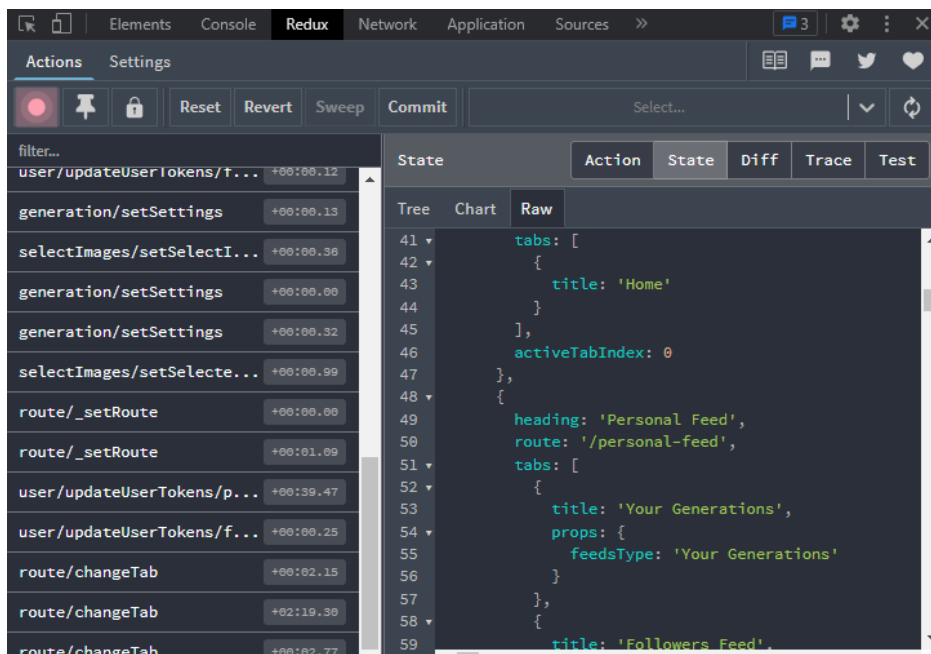
*Fuente.* Visual Studio Code

## Instalación de herramienta de desarrollo

Para facilitar el desarrollo y conocer el comportamiento del estado de la aplicación, se instala la herramienta de Redux DevTools que es una extensión de Google Chrome que permite ver las acciones ejecutadas, el estado del Store y manipularlo a nuestro gusto para realizar pruebas, a continuación, se presenta una imagen con la interfaz de la herramienta:

**Figura 128**

### Revisión de la herramienta Redux DevTools



Fuente. Google Chrome

## Implementación de Docker

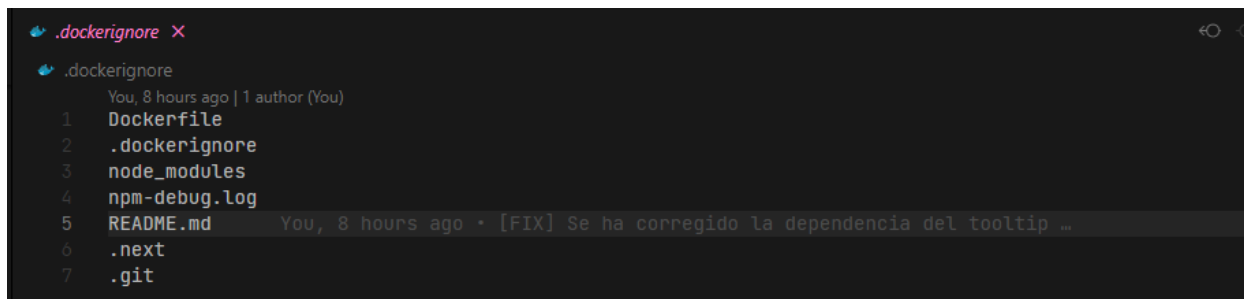
Para asegurar la eficiencia y la portabilidad de la aplicación, se utilizar Docker para separar y encapsular cada uno de los componentes clave de la aplicación, incluyendo el frontend, el backend y la base de datos, esta herramienta se convirtió en un elemento esencial ya que permite definir y orquestar múltiples contenedores de manera coordinada, para esto se crea un archivo de configuración de Docker Compose que describe la estructura de nuestra aplicación y las relaciones entre los contenedores, esto facilita la gestión y el despliegue ya que inicia y detiene todos los servicios relacionados con un solo comando.

### Configuración del servicio frontend

Para comenzar la configuración del servicio se crea un archivo dockerignore en el cual se especifican los directorios y archivos que se deben excluir cuando se genere su respectiva imagen, por ejemplo, no se incluyen los módulos de node js, la carpeta de git, la carpeta generada para producción de nextjs, entre otros elementos comunes en este tipo de servicios.

### Figura 129

*Contenido del archivo dockerignore del servicio frontend*



```
.dockerignore
1 Dockerfile
2 .dockerignore
3 node_modules
4 npm-debug.log
5 README.md
6 .next
7 .git
```

*Fuente.* Visual Studio Code

El siguiente paso es crear un archivo Dockerfile que contiene toda la configuración del servicio, estos contenedores trabajan sobre Linux por lo que lo primero que se hace es establecer

un directorio de trabajo, luego mover el archivo de configuración de node, instalar las dependencias, mover el archivo fuente, construir la aplicación de producción, exponer el puerto 3000 y ejecutar el servicio.

### Figura 130

*Archivo de configuración de la imagen del servicio frontend*

```

Dockerfile
You, 8 hours ago | 1 author (You)
1 FROM node:18-alpine AS base
2
3 # Install dependencies only when needed
4 FROM base AS deps
5 # Check https://github.com/nodejs/docker-node/tree/b4117f9333da4138b03a546ec926ef50a31506c3#nodealpine to understand why libc6-compat
6 RUN apk add --no-cache libc6-compat
7 WORKDIR /app You, 8 hours ago · [FIX] Se ha corregido la dependencia del tooltip ...
8
9 # Install dependencies based on the preferred package manager
10 COPY package.json package-lock.json* ./
11 RUN npm ci
12
13
14 # Rebuild the source code only when needed
15 FROM base AS builder
16 WORKDIR /app
17 COPY --from=deps /app/node_modules ./node_modules
18 COPY . .
19
20 # Next.js collects completely anonymous telemetry data about general usage.
21 # Learn more here: https://nextjs.org/telemetry
22 # Uncomment the following line in case you want to disable telemetry during the build.
23 # ENV NEXT_TELEMETRY_DISABLED 1
24
25 RUN npm run build
26
27 # Production image, copy all the files and run next
28 FROM base AS runner
29 WORKDIR /app

```

*Fuente.* Visual Studio Code

### Configuración del servicio backend

Al igual que con el anterior servicio se crea un archivo que al momento de construir la imagen ignore los módulos de node, también los archivos de copia de seguridad, las copias de seguridad y los archivos de registro de eventos, así optimizando el peso de la aplicación.

## Figura 131

### *Archivo dockerignore del servicio backend*

```

1  node_modules/
2  npm-debug.log
3  yarn-error.log
4
5
6  # Archivos y directorios específicos de TypeScript
7  *.tsbuildinfo
8
9  # Dependiendo de tu entorno de desarrollo, puedes ignorar otros archivos
10 # que no sean necesarios para la imagen Docker. Por ejemplo, si estás usando
11 # Visual Studio Code, puedes ignorar los siguientes archivos y directorios:
12 .vscode/
13 .vscode/settings.json
14 .vscode/tasks.json
15 .vscode/launch.json
16 .vscode/extensions.json
17
18 # Cualquier otro archivo o directorio que no sea necesario para la imagen Docker
19 /dist
20 /logs
21 /src/backups
22 /src/uploads

```

*Fuente.* Visual Studio Code

El siguiente paso es generar el archivo de configuración del servicio, para esto de igual manera se crea un espacio de la aplicación dentro del contenedor, luego se establece que se usará node como entorno de ejecución, luego se copian todos los archivos necesarios a la aplicación, se instalan las dependencias, se crea el paquete de construcción, se expone el puerto 5454 y se inicia el servicio.

## Figura 132

### *Archivo de configuración del servicio backend*

```

1  # Utiliza una imagen base de Node.js LTS
2  FROM node:18-alpine as build
3
4  # Establece el directorio de trabajo en /app
5  WORKDIR /app
6
7  # Copia el archivo de definición de paquetes y el archivo de bloqueo de npm
8  COPY package.json package-lock.json ./
9
10 # Instala las dependencias de producción
11 RUN npm install
12
13 # Copia el código fuente de la aplicación
14 COPY . .
15
16 # Ejecuta el comando de construcción (ajusta según tu proyecto)
17 RUN npm run build
18
19 # Segunda etapa para la imagen final
20 FROM node:18-alpine
21
22 # Establece el directorio de trabajo en /app
23 WORKDIR /app

```

*Fuente.* Visual Studio Code

## Configuración general y unión de los servicios

Para configurar la aplicación y unir los servicios se crea un archivo Docker Compose el cual permite establecer los servicios que se utilizarán y como se comunicarán, en este se establece la base de datos la cual usa una imagen de Mongo DB, se expone el puerto por defecto que es el 27017 y se le asigna un volumen físico el cual guardara una copia de la información en caso de que se requieran copias de seguridad adicionales.

### Figura 133

*Configuración de la base de datos en el archivo Docker Compose*

```
1 version: '3.1'
2 services:
3   mongo:
4     container_name: mongo-db-service
5     image: mongo
6     restart: always
7     ports:
8       - 27017:27017
9     volumes:
10      - ./filesoft-backend/data:/data/db
```

*Fuente.* Visual Studio Code

Luego se configura el servicio backend al cual se le dice dónde está el archivo de configuración, cuáles serán los puertos físicos y virtuales que usará, el enlace a la base de datos, las variables de entorno que se establecerán al ser requeridas y los volúmenes de datos para los registros de eventos, las copias de seguridad y las cargas de archivos.

**Figura 134**

*Configuración del servicio backend en el archivo Docker Compose*

```

12 filesoft-backend:
13   container_name: filesoft-backend
14   build: ./filesoft-backend
15   restart: always
16   expose:
17     - "5454"
18   ports:
19     - "5454:5454"
20   links:
21     - mongo
22   environment:
23     - APP_PORT=5454
24     - APP_EMAIL_USER=[REDACTED]
25     - APP_EMAIL_PASSWORD=[REDACTED]
26     - APP_EMAIL_GOOGLE_PASSWORD=[REDACTED]
27     - APP_EMAIL_HOST=smtp.gmail.com
28     - APP_EMAIL_PORT=587
29     - APP_EMAIL_ENCRYPTION=tls
30     - APP_DB_MANAGER=mongodb
31     - APP_DB_URL=mongodb://mongo:27017/filesoft
32     - APP_ADMIN_EMAIL=[REDACTED]
33     - APP_ADMIN_NAME=[REDACTED]
34     - APP_FRONTEND_URI=http://localhost:3000
35     - APP_JWT_SECRET=[REDACTED]
36     - NODE_ENV=production
37   volumes:
38     - ./filesoft-backend/logs:/logs
39     - ./filesoft-backend/dist/uploads:/dist/uploads
40     - ./filesoft-backend/dist/backups:/dist/backups

```

*Fuente.* Visual Studio Code

Y se finaliza realizando la integración del servicio frontend, definiendo los puertos que se van a exponer, la ruta donde se encuentra el archivo de configuración y se define que ante cualquier error se reinicie el servicio y se vuelva a exponer.

**Figura 135**

*Configuración del servicio frontend en el archivo Docker Compose*

```

filesoft-frontend:
  container_name: filesoft-frontend
  build: ./filesoft-frontend
  restart: always
  expose:
    - "3000"
  ports:
    - "3000:3000"

```

*Fuente.* Visual Studio Code

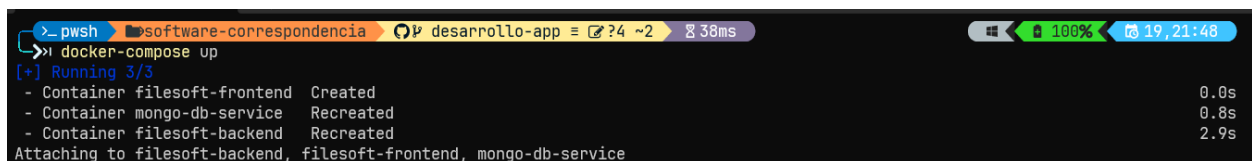


## Puesta en marcha de los contenedores

Esta es la parte final de la implementación de contenedores, para esto solo debemos entrar a la terminal y desde el directorio donde se encuentra el archivo Docker-compose.yml y se ejecuta el comando “docker-compose up”, esto comenzará a descargar lo necesario para ejecutar todos los servicios como uno.

### Figura 136

*Ejecución de la aplicación con ayuda de Docker*



```

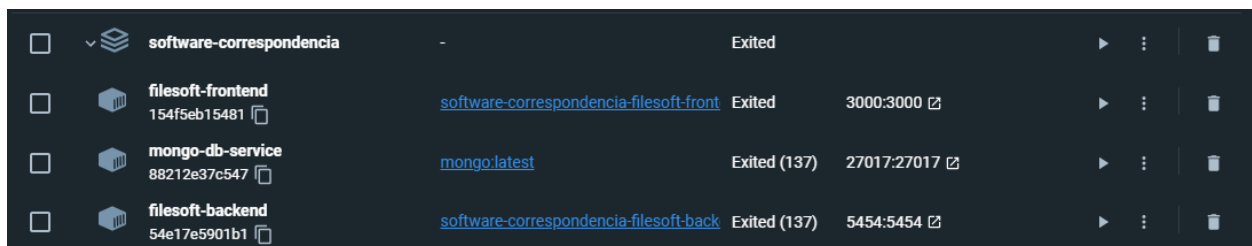
> pwsh
> docker-compose up
[+] Running 3/3
 - Container filesoft-frontend Created 0.0s
 - Container mongo-db-service Recreated 0.8s
 - Container filesoft-backend Recreated 2.9s
Attaching to filesoft-backend, filesoft-frontend, mongo-db-service
  
```

*Fuente.* Windows Power Shell

Esto se puede verificar desde el cliente de Docker Desktop el cual muestra el grupo de servicios y cada una de las imágenes que se están utilizando para su ejecución.

### Figura 137

*Resumen de los servicios desde el cliente Docker Desktop*



Container Name	Image	Status	Ports
software-correspondencia	-	Exited	
filesoft-frontend 154f5eb15481	software-correspondencia-filesoft-front	Exited	3000:3000
mongo-db-service 88212e37c547	mongo:latest	Exited (137)	27017:27017
filesoft-backend 54e17e5901b1	software-correspondencia-filesoft-back	Exited (137)	5454:5454

*Fuente.* Docker Desktop

## Pruebas de la aplicación Filessoft

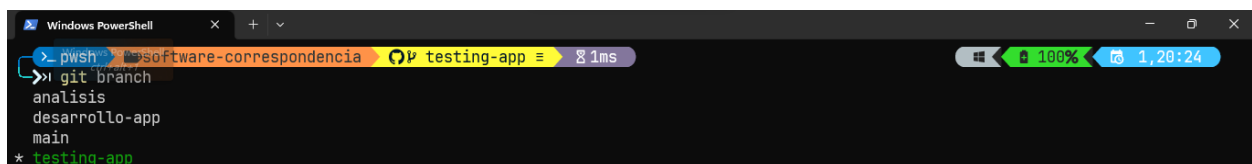
En el desarrollo de software una parte fundamental son las pruebas, debido a que estas permiten ver el comportamiento de la aplicación web en diferentes escenarios, permitiendo que en desarrollo se encuentren problemas a temprana edad de la aplicación para poder solucionarlos o desarrollar nuevas funciones que mitiguen el impacto generado; para probar la aplicación se utiliza Cypress la cual es una herramienta de pruebas e2e o por componentes, al querer probar el sistema en general se usa el enfoque e2e el cual es capaz de simular el comportamiento de un usuario sobre la aplicación, como el llenado de formularios, interacción con botones, validación de componentes, entre otras funcionalidades.

### Instalación de la herramienta de pruebas

Esta herramienta al igual que el resto de los paquetes instalados se realiza por medio de la terminal con los comandos dados en la documentación oficial encontrada en [www.cypress.io/](http://www.cypress.io/), pero algo para tener en cuenta antes de hacerlo es que esto es solo desarrollo por lo que es recomendable hacerlo en paralelo al proyecto creado y no exactamente sobre el mismo debido a que esto no será parte de la aplicación final, para esto como se trabaja con el gestor de versiones de Git, se crea una nueva rama para realizar las pruebas y así únicamente unir las cada vez que se requiera comprobar el funcionamiento particular de la aplicación.

### Figura 138

*Creación de nueva rama para probar la aplicación*

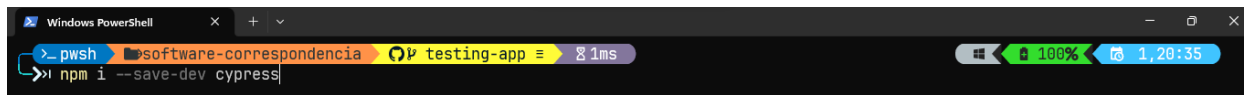


```
Windows PowerShell
> .\pwsh -c cd software-correspondencia & cd testing-app & git branch
analisis
desarrollo-app
main
* testing-app
```

*Fuente.* Windows Power Shell

## Figura 139

*Instalación de la herramienta de pruebas Cypress*



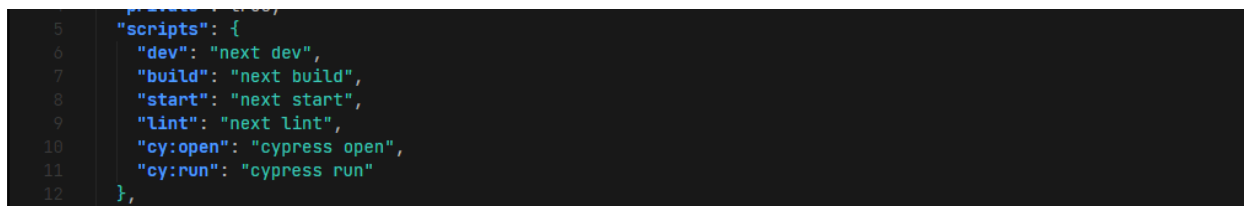
*Fuente.* Windows Power Shell

## Configuración de comandos de inicio

Para poder ejecutar las pruebas se deben configurar dos comandos que serán los encargados de iniciar el entorno de Cypress, estos son el comando de inicio con navegador que mostrará gráficamente cada una de las pruebas y el comando de inicio por terminal que ejecutará todas las pruebas sin un entorno gráfico, aunque da la posibilidad de tener capturas de pantalla y videos del proceso.

## Figura 140

*Comandos de cy:open y cy:run agregados para pruebas*



*Fuente.* Visual Studio Code

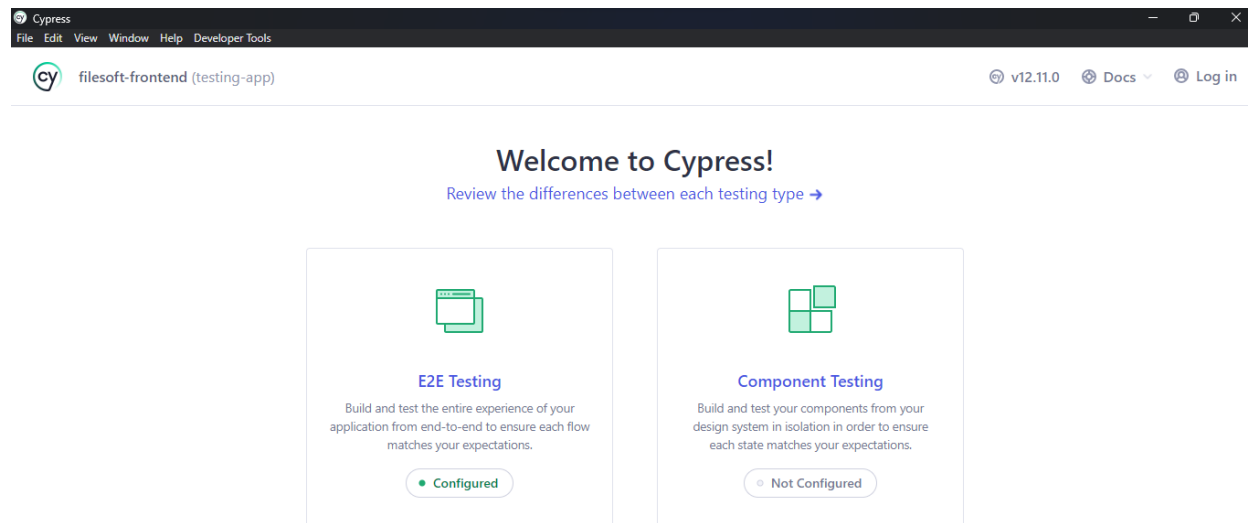
## Inicio de la herramienta

La herramienta una vez ejecutada crea sus respectivos directorios donde se organizan las pruebas, ya sea por componentes o e2e, estas son leídas por la herramienta y va ejecutando una por una.

Para comenzar se elige el entorno a configurar, para la aplicación se elige E2E Testing que configurará todo el entorno como se muestra a continuación:

### Figura 141

*Elección del tipo de pruebas que se realizarán*

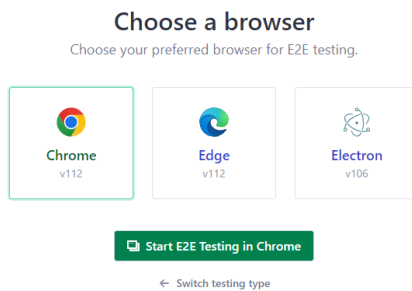


*Fuente.* Entorno gráfico de Cypress

El siguiente paso es seleccionar el navegador donde se ejecutarán las pruebas, se puede seleccionar entre Google Chrome, Microsoft Edge y Electron los cuales serán controlados por la herramienta, para la aplicación se probará con Google Chrome:

### Figura 142

*Selección de navegador para hacer las pruebas*

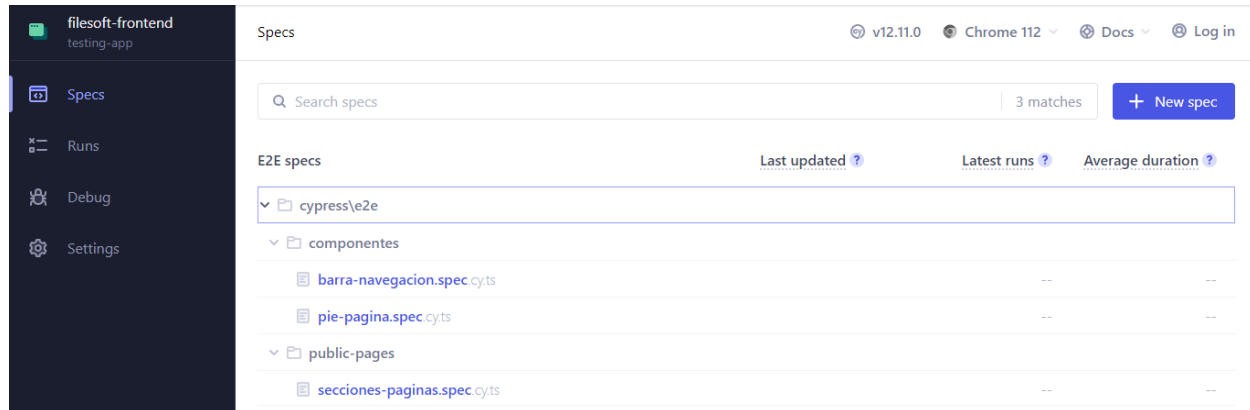


*Fuente.* Entorno gráfico de Cypress

En la siguiente ventana se presenta el listado de pruebas creadas, las cuales se pueden seleccionar una a una y se encargará de ejecutarse y controlar el navegador como si de un usuario se tratase:

**Figura 143**

*Listado de pruebas creadas*

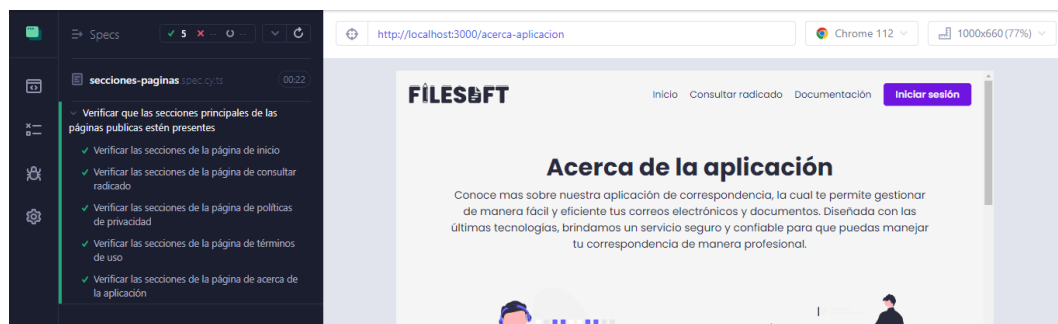


*Fuente.* Entorno gráfico de Cypress.

Al seleccionar una prueba esta detallará el paso a paso para su ejecución y mostrará los títulos establecidos dentro del diseño de estas, por ejemplo, la siguiente imagen presenta las pruebas que se le hicieron a todas las secciones públicas buscando que existan las principales a mostrar al usuario, esta tomó un total de 22 segundos y pasó todas las pruebas establecidas:

**Figura 144**

*Ejecución de pruebas de validación de secciones*



*Fuente.* Entorno gráfico de Cypress.

## Manual de uso del servicio API

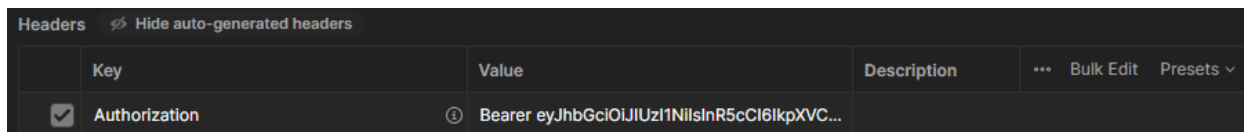
A continuación, se presentan el manual para desarrolladores que define las rutas que soporta la API, el tipo de solicitud HTTP, los datos que se deben enviar en la solicitud y las posibles respuestas que se pueden obtener, para esta documentación se usa la ruta <https://filessoft.com> como referencia de un posible dominio para la aplicación, sin embargo, su despliegue puede conllevar un dominio diferente, por lo que solo se usará de referencia.

### Cabecera de solicitud requerida

Para las rutas protegidas se requiere que la cabecera de solicitud “Authorization” esté presente y contenga el token de sesión para que el servidor pueda identificar al usuario que realice la solicitud.

### Figura 145

*Ejemplo de cabecera Authorization*



Key	Value	Description	...	Bulk Edit	Presets
<input checked="" type="checkbox"/> Authorization	Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVC...				

*Fuente.* Software de Postman.

### Archivos

Estas rutas permiten gestionar los archivos que se carguen dentro de la aplicación, entre estas acciones se permiten cargar más documentos, actualizar un documento cargado, obtener el archivo físico del servidor y eliminarlo de manera permanente de la base de datos y del sistema de archivos.

### **Listar archivos**

Lista todos los archivos registrados en la aplicación o solo los asociados al usuario específico, esto depende de los permisos que se registren a cada uno de los usuarios, la petición debe ser tipo GET hacia la ruta <https://filessoft.com/api/archivos>, la cual puede aceptar los siguientes parámetros de URL (query params) que en las rutas de listar archivo siempre son las mismas, por esta razón se referenciará a esta tabla al explicar esas solicitudes.

#### **Tabla 28**

##### *Parámetros de URL para listar los archivos*

Parámetro	Descripción	Ejemplo
page	Establece la página que se mostrará al obtener los datos paginados del servidor.	1 (por defecto)
limit	Establece el número de registros que se mostrarán por página.	25 (por defecto)
sort	Establece el valor o cambio de la colección que realizará el ordenamiento de los datos.	createdAt
q	Establece un valor de búsqueda, actúa como filtro a los campos de la colección.	“Yeferson”
startDate	Establece la fecha de inicio desde donde se quiere que se muestren los	2021-08-01T00:00:00.000Z

	registros, usando de referencia la fecha de creación.	
endDate	Establece la fecha de finalización desde donde se quiere que se muestren los registros, usando de referencia la fecha de creación.	2021-09-12T00:00:00.000Z

---

*Nota.* Se detallan los parámetros para mostrar el listado de archivos registrados.

De esta manera existen algunas validaciones que permiten que las solicitudes no vayan a generar un error dentro del servidor o en caso de que la solicitud sea correcta retorne la información de los archivos registrados.

### **Tabla 29**

*Listado de respuestas al solicitar el listado de archivos*

Respuesta	Mensaje de la API	Causas
Solicitud inválida	La información enviada no es válida.	Los parámetros de URL no son correctos o no corresponden a los que soporta el servidor.
Listado de archivos	Listado de archivos subidos al servidor y asociados a su respectiva petición.	La solicitud se realizó de manera correcta.

---

*Nota.* Se detallan las posibles respuestas del servidor al solicitar el listado de archivos.



**Obtener archivo**

Obtiene un archivo específico con el identificador y el token de acceso del usuario que lo solicita, este se puede obtener haciendo una petición HTTP tipo GET a la dirección <https://filessoft.com/api/archivos/64932f110087f083deb9a38e> pasando los siguientes parámetros de URL.

**Tabla 30***Parámetros de URL para obtener un archivo*

Parámetro	Descripción	Ejemplo
fileAccessToken	Establece el token de acceso a los archivos el cual es requerido para poder validar los permisos y si puede acceder al archivo.	6kbVhTkw254g9r1hQbd8tOVcMCjZhh
download	Establece si el archivo se debe descargar automáticamente o solo se hará la visualización.	False (por defecto)

*Nota.* Se detallan los parámetros que se requieren para solicitar un archivo.

De esta manera existen algunas validaciones que permiten que las solicitudes no vayan a generar un error dentro del servidor o en caso de que la solicitud sea correcta retorne el archivo requerido.

**Tabla 31***Respuestas del servicio al solicitar un archivo*

Respuesta	Mensaje de la API	Causas
Solicitud inválida	La información enviada no es válida.	Los parámetros de URL no son correctos.
Error token acceso	Se ha presentado un error al validar el token de acceso al archivo solicitado.	El token enviado no es válido.
Sin permisos	El usuario no cuenta con los permisos necesarios para obtener el archivo solicitado.	No se cuenta con los permisos necesarios para leer el archivo.
Usuario no existe	El usuario asociado al token no está registrado en la aplicación.	Inconsistencia en el token, por lo que se debe iniciar sesión nuevamente.

---

Nota. Se detallan las posibles respuestas del servidor al solicitar un archivo específico.

***Subir archivo***

Permite subir un archivo al servidor y retorna la información de la subida del archivo, para esto se debe hacer una petición tipo POST a la dirección <https://filessoft.com/api/archivos>, la cual tiene el siguiente parámetro de URL.

**Tabla 32***Parámetro de URL opcional al subir un archivo*

Parámetro	Descripción	Ejemplo
skipDangerousFiles	Establece si se deben omitir archivos potencialmente peligrosos y que se carguen al servidor.	false (por defecto)

*Nota.* Se detallan los parámetros que se requieren para subir un archivo.

Además de esto requiere que junto a la solicitud se envíe en el cuerpo de la solicitud lo siguiente, lo cual es necesario para llevar a cabo la carga del archivo.

**Tabla 33***Cuerpo de la solicitud para cargar un archivo*

Parámetro	Descripción
fileUpload	Archivo que se cargará al servidor y se almacenará en el sistema de archivos únicamente.

*Nota.* Se establece el cuerpo de la solicitud para subir un archivo.

De esta manera existen algunas validaciones que permiten que las solicitudes no vayan a generar un error dentro del servidor o en caso de que la solicitud sea correcta guarde el archivo en el servidor.

**Tabla 34***Respuestas del servicio al subir un archivo*

Respuesta	Mensaje de la API	Causas
Sin permisos	El usuario no cuenta con los permisos necesarios para obtener el archivo solicitado.	No se cuenta con los permisos necesarios para leer el archivo.
Sin archivo	Se debe enviar un archivo para cargarlo en el servidor.	No se cargó un archivo en el cuerpo de la solicitud.
Archivo potencialmente peligroso	El archivo que intentas cargar tiene una extensión que puede ser peligrosa (exe). Si estás seguro de que el archivo es seguro, envía el parámetro <code>skipDangerousFiles=true</code> para cargarlo bajo tu propio riesgo.	El archivo tiene una extensión que puede ser peligrosa si no se tiene cuidado en su ejecución.
Archivo subido	Se ha subido el archivo correctamente.	El archivo se ha almacenado en el servidor.

*Nota.* Se detallan las posibles respuestas del servidor al subir un archivo.

### **Actualizar archivo**

Actualiza la información de un archivo específico, realiza la carga del nuevo archivo y actualiza la información, para esto se debe hacer una solicitud HTTP de tipo PUT a la dirección <https://filesoft.com/api/archivos/:id>, la cual tiene los siguientes parámetros de URL.

**Tabla 35**

*Parámetro de URL para actualizar un archivo*

Parámetro	Descripción	Ejemplo
skipDangerousFiles	Establece si se deben omitir archivos potencialmente peligrosos y que se carguen al servidor.	false (por defecto)
:id	Identificador del archivo que se desea actualizar.	64a717f579ae7af964563fe9

*Nota.* Se detallan los parámetros que se requieren para actualizar un archivo.

Además de esto requiere que junto a la solicitud se envíe en el cuerpo de la solicitud lo siguiente, lo cual es necesario para llevar a cabo la actualización del archivo.

**Tabla 36**

*Parámetros necesarios al actualizar un archivo*

Parámetro	Descripción
fileUpload	Archivo que se cargará al servidor y se almacenará en el sistema de archivos únicamente.

*Nota.* Se establece el cuerpo de la solicitud para actualizar un archivo.

De esta manera existen algunas validaciones que permiten que las solicitudes no vayan a generar un error dentro del servidor o en caso de que la solicitud sea correcta actualice el archivo del servidor.

**Tabla 37**

*Respuestas del servicio al actualizar un archivo*

Respuesta	Mensaje de la API	Causas
Solicitud inválida	La información enviada no es válida.	El identificador del archivo no tiene un formato de Mogo DB.
Sin permisos	El usuario no cuenta con los permisos necesarios para obtener el archivo solicitado.	No se cuenta con los permisos necesarios para leer el archivo.
Sin archivo	Se debe enviar un archivo para cargarlo en el servidor.	No se cargó un archivo en el cuerpo de la solicitud.
Archivo no existe	El archivo que se intenta actualizar no existe.	El identificador del archivo no pertenece a ningún archivo almacenado.
Archivo potencialmente peligroso	El archivo que intentas cargar tiene una extensión que puede ser peligrosa (exe). Si estás seguro de que el archivo es seguro, envía	El archivo tiene una extensión que puede ser peligrosa si no se tiene cuidado en su ejecución.

el parámetro  
 skipDangerousFiles=true  
 para cargarlo bajo tu propio  
 riesgo.

Archivo actualizado	El archivo se ha actualizado correctamente.	El archivo se ha actualizado en el servidor.
---------------------	---	--

---

*Nota.* Se detallan las posibles respuestas del servidor al actualizar un archivo.

### ***Eliminar archivo***

Permite eliminar un archivo de la aplicación, se debe tener en cuenta que este archivo no esté asociado a ningún elemento, en caso de estarlo no se permitirá su eliminación, para esto se debe realizar una petición tipo DELETE a la dirección <https://filessoft.com/api/archivos/:id>, esta requiere el siguiente parámetro:

### **Tabla 38**

#### *Parámetros para eliminar un archivo*

Parámetro	Descripción	Ejemplo
forceDelete	Establece si se debe forzar la eliminación del archivo en caso de que tenga algunos permisos que lo impidan.	false (por defecto)
:id	Identificador del archivo que se desea eliminar.	64a717f579ae7af964563fe9

---

*Nota.* Se detallan los parámetros que se requieren para eliminar un archivo.

De esta manera existen algunas validaciones que permiten que las solicitudes no vayan a generar un error dentro del servidor o en caso de que la solicitud sea correcta elimine el archivo del servidor y la base de datos.

**Tabla 39**

*Respuestas del servicio al intentar eliminar un archivo*

Respuesta	Mensaje de la API	Causas
Solicitud inválida	La información enviada no es válida.	El identificador del archivo no tiene un formato de Mogo DB.
Sin permisos	El usuario no cuenta con los permisos necesarios para obtener el archivo solicitado.	No se cuenta con los permisos necesarios para leer el archivo.
Archivo asignado a proceso	El archivo que intenta eliminar está asignado a procesos activos, por esta razón no puede ser eliminado, si desea eliminarlo debe hacerlo forzando la eliminación puede enviar el parámetro <code>forceDelete=true</code> .	El archivo está asignado a algún elemento dentro de la aplicación.



Archivo en uso	Se ha presentado un error al eliminar el archivo solicitado, puede que el archivo esté siendo usado por otro usuario.	El archivo está abierto por lo que no se podrá eliminar hasta que se cierre el programa que lo tiene en uso.
Archivo eliminado	Se ha eliminado el archivo solicitado de manera exitosa	El archivo se ha eliminado del servidor y la base de datos.

---

*Nota.* Se detallan las posibles respuestas del servidor al actualizar un archivo.

### **Asignación dependencias**

Contiene las rutas que permiten gestionar las asignaciones de dependencias de la aplicación, así permitiendo que se generen asignaciones, que se actualicen y se eliminen de la aplicación.

#### ***Listar asignación dependencias***

Obtiene el listado de todas las asignaciones de dependencias que se han realizado, mostrando los funcionarios y las dependencias involucradas en cada una, para esto se debe hacer una solicitud tipo GET a <https://filessoft.com/api/asignacion-dependencias>, usando los parámetros de URL definidos en la solicitud de listar archivos.

**Tabla 40***Respuestas del servicio al obtener el listado de asignaciones de dependencias*

Respuesta	Mensaje de la API	Causas
Solicitud inválida	La información enviada no es válida.	El identificador del archivo no tiene un formato de Mogo DB.
Sin permisos	No tienes permisos para listar asignaciones de dependencias.	No se cuenta con los permisos necesarios para leer las asignaciones.
Listado de asignaciones	Listado de asignaciones de dependencias registradas.	Se ha obtenido el listado de manera correcta.

*Nota.* Se detallan las posibles respuestas del servidor obtener el listado de asignaciones.

## Manual de usuario para el uso de Filesoft

En esta guía, encontrarás toda la información necesaria para utilizar de manera efectiva el software de correspondencia y sacar el máximo provecho de todas sus funcionalidades, esta es una herramienta que permite gestionar y organizar eficientemente toda la correspondencia, controlando tus documentos de manera centralizada, agilizando los procesos de comunicación y mejorando la productividad de tu organización.

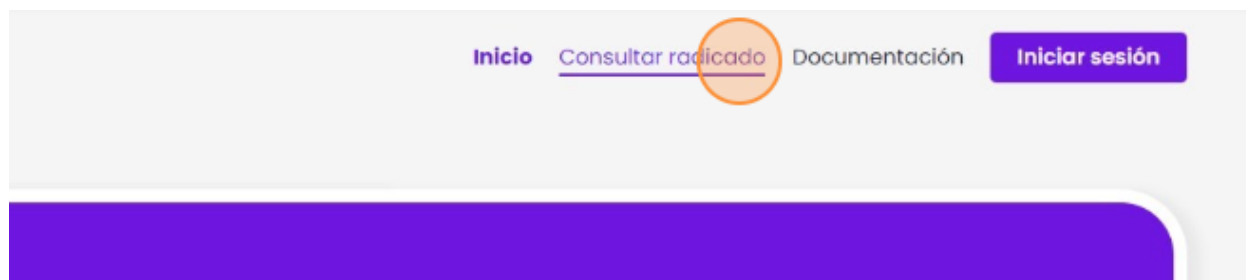
### Conoce Filesoft APP

Accede a la dirección <https://filesoft.com/inicio> desde tu navegador preferido, se recomienda usar Google Chrome, aquí podrás conocer un resumen detallado de la aplicación, sus servicios y una visión general del objetivo de la aplicación.

Si deseas realizar el seguimiento de una solicitud o radicado puedes dar clic en el enlace "Consultar radicado" que te llevará a la dirección <https://filesoft.com/consultar-radicado>, su uso se explica más a detalle en la sección de "Consulta y seguimiento de radicado" de esta documentación.

### Figura 146

*Acción de clic sobre el enlace para consultar un radicado*



*Fuente.* Capturado con el software Scribehow.

Si deseas conocer las políticas de privacidad de la aplicación puedes dar clic en "Políticas de privacidad" el cual te direccionará a <https://filesoft.com/politicas-privacidad> donde conocerás más a detalle cómo se realiza la recopilación de datos de los funcionarios, los ciudadanos, cómo se maneja la seguridad de la aplicación, la gestión del correo electrónico y las políticas de acceso con Google.

### Figura 147

*Acción de clic sobre el enlace para conocer las políticas de privacidad*



*Fuente.* Capturado con el software Scribehow.

Para conocer los términos y condiciones de la aplicación puedes dar clic en "Términos de uso" que te llevará a <https://filesoft.com/terminos-uso> donde se detalla cuáles son los elementos que se deben tener en cuenta al usar la aplicación y se advierte que su mal uso puede llevar a varias consecuencias.

### Figura 148

*Acción de clic sobre el enlace para conocer los términos de uso*

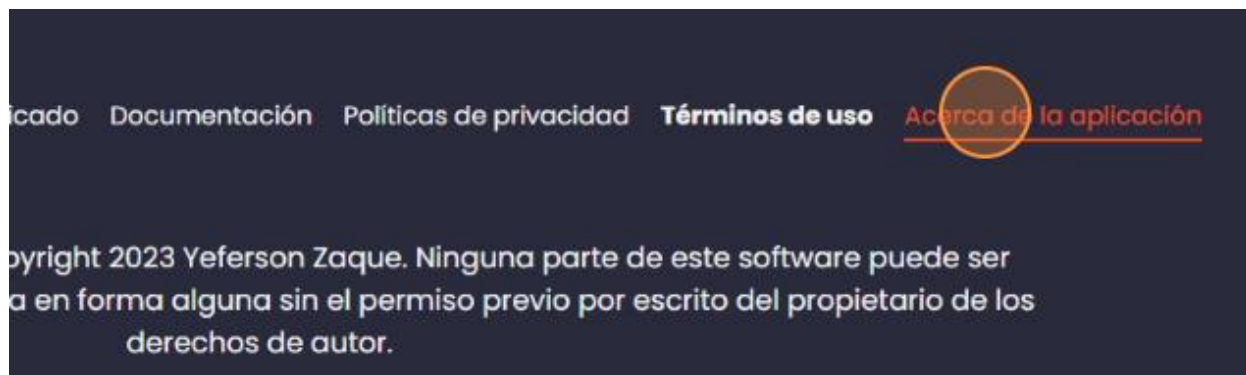


*Fuente.* Capturado con el software Scribehow.

Por último, para conocer más sobre la aplicación, cómo surgió, que necesidad está solucionando y las personas involucradas en el proyecto puedes dar clic en "Acerca de la aplicación" que te llevará a <https://filessoft.com/acerca-aplicacion> y se te mostrará la información más a detalle.

### **Figura 149**

*Acción de clic en el enlace para conocer acerca de la aplicación*



*Fuente.* Capturado con el software Scribehow.

### **Consulta y seguimiento de radicado**

Accede a la dirección <https://filessoft.com/consultar-radicado> y da clic en el campo con nombre "Número de radicado" y escribe en el campo de texto el número de radicado que deseas buscar, por ejemplo "0001".

**Figura 150**

*Campo de texto para digitar el número de radicado a consultar*

*Fuente.* Capturado con el software Scribehow.

Da clic en el campo con nombre "Año de radicación", y escribe el año en que realizaste la solicitud, por ejemplo "2023".

**Figura 151**

*Digitación del año de radicación*

*Fuente.* Capturado con el software Scribehow.

Da clic en el botón "Buscar" para empezar el proceso de búsqueda y si existe mostrará la información de la línea de tiempo por la que ha pasado el radicado.

### Figura 152

*Clic sobre el botón de buscar radicado*

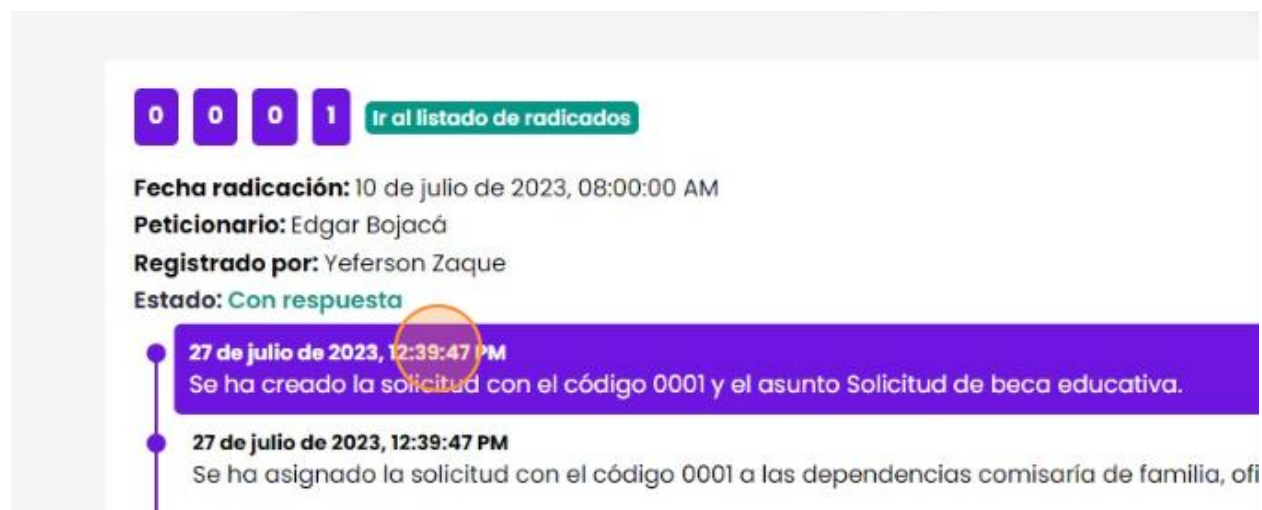


*Fuente.* Capturado con el software Scribehow.

Por último, visualiza la línea de tiempo del radicado que consultaste, en el cual se expone la información de manera simple.

### Figura 153

*Línea de tiempo del radicado consultado*



*Fuente.* Capturado con el software Scribehow.

## Iniciar sesión en la aplicación

Accede desde el navegador a la dirección <https://filessoft.com/inicio> o a cualquiera de las páginas públicas de la aplicación y da clic en la parte superior derecha en el botón "Iniciar sesión" o accede directamente a la ruta <https://filessoft.com/auth/iniciar-sesion> donde se mostrará el formulario de inicio de sesión.

### Figura 154

*Clic sobre el botón de inicio de sesión*

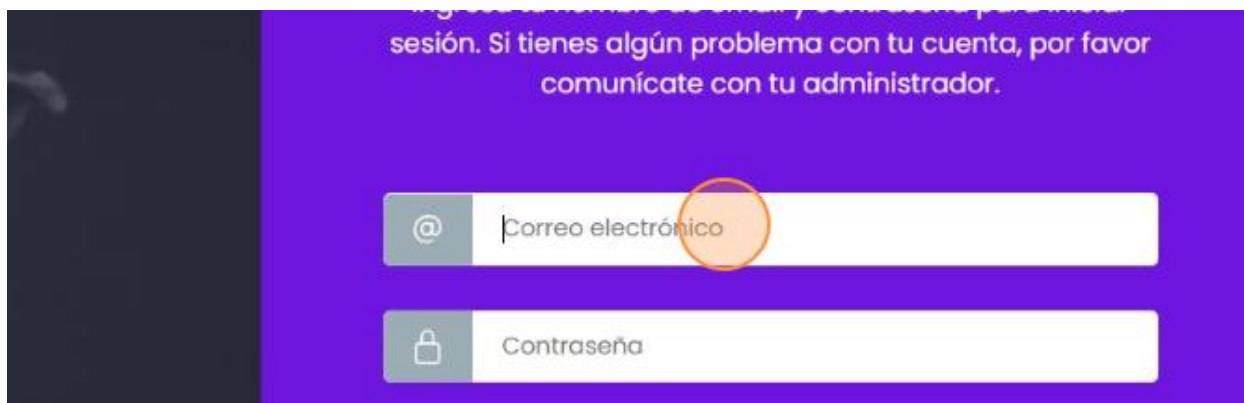


*Fuente.* Capturado con el software Scribehow.

Da clic en el campo de "Correo electrónico" y escribe tú correo electrónico con el que te registraste en la aplicación, por ejemplo "correoejemplo@gmail.com".

### Figura 155

*Digitación del correo electrónico para iniciar sesión en la aplicación*



*Fuente.* Capturado con el software Scribehow.

Da clic en el campo de "Contraseña" y digita la contraseña que estableciste en tú registro y finaliza dando clic en "Iniciar sesión" para hacer la solicitud y acceder al panel de administración.



**Figura 156**

*Solicitud de acceso al panel de administración*



*Fuente.* Capturado con el software Scribehow.

**Restablecer contraseña**

Accede a la aplicación en <https://filessoft.com/inicio> y da clic en el botón "Iniciar sesión" que te dirigirá a <https://filessoft.com/auth/iniciar-sesion>, en el formulario de inicio de sesión da clic en el enlace "¿Has olvidado tu contraseña?" que te dirigirá a <https://filessoft.com/auth/olvide-password>.

**Figura 157**

*Clic en el enlace para ir a el formulario para restablecer la contraseña*



*Fuente.* Capturado con el software Scribehow.

En el formulario de recuperación de contraseña da clic en el campo de "Correo electrónico", y escribe tú correo electrónico con el que estás registrado en la aplicación, por ejemplo [correodeprueba@gmail.com](mailto:correodeprueba@gmail.com), luego da clic en el campo de "Número de identificación" y escribe el documento con el que te registraste en la aplicación.

### Figura 158

*Formulario para comenzar el proceso para restablecer la contraseña*



Ingresa tu dirección de correo electrónico registrada y tu número de identificación, se te enviará un enlace para restablecer tu contraseña. Si tienes problemas para recuperar tu contraseña, por favor comunícate con el administrador.

@

10101010

*Fuente.* Capturado con el software Scribehow.

Da clic en el botón "Realizar solicitud" para comenzar el proceso, si todo está bien se te indicará que se ha enviado un correo electrónico con la información de restablecimiento.

### Figura 159

*Clic en el botón para comenzar el proceso de restablecimiento*



10101010

**Realizar solicitud**

*Fuente.* Capturado con el software Scribehow.

Accede a tu gestor de correos electrónicos, como por ejemplo Gmail, verifica que el correo de restablecimiento de contraseña ha llegado y da clic en él para acceder a su contenido.

### Figura 160

*Recepción de correo de restablecimiento de contraseña*



*Fuente.* Capturado con el software Scribehow.

Abre el correo recibido y da clic en el botón de "Restablecer contraseña" para realizar este proceso.

### Figura 161

*Contenido del correo de restablecimiento de contraseña*



*Fuente.* Capturado con el software Scribehow.

Esto te dirigirá al formulario de restablecimiento, para comenzar da clic en el campo de "Número de identificación", digita tú número de identificación con el cual has realizado el proceso desde el inicio, luego da clic en el campo "Contraseña" e ingresa tú nueva contraseña con la que deseas acceder a la aplicación, finaliza el diligenciado del formulario dando clic en el campo "Confirmar contraseña" y digita la misma contraseña que escribiste en el paso anterior.

### Figura 162

*Formulario de restablecimiento de contraseña*

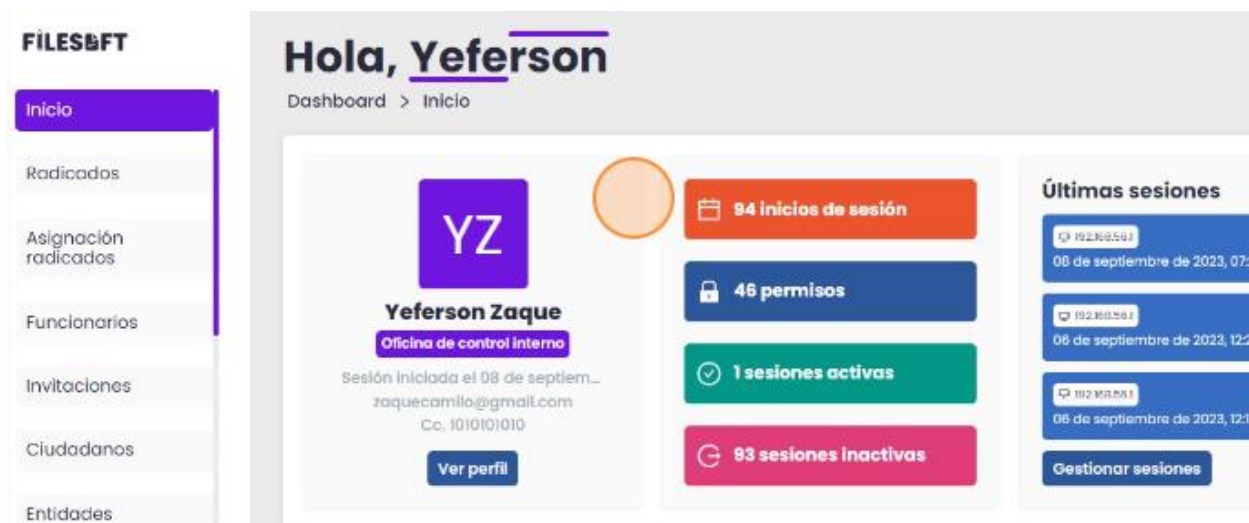
The image shows a password reset form. On the left, there is a solid purple vertical bar. To its right, there are three input fields. The first field has a person icon and contains the text '1010101010'. The second field has a lock icon and contains several dots. The third field has a lock icon and contains the text 'Confirmar contraseña', with an orange circle highlighting the text. Below these fields is a blue button with the text 'Establecer nueva contraseña'.

*Fuente.* Capturado con el software Scribehow.

Por último, da clic en el botón "Establecer nueva contraseña" para realizar la actualización de contraseña y se te indicará que el proceso se realizó de manera exitosa, así que da clic en el botón "Ir a iniciar sesión" para acceder a <https://filessoft.com/auth/iniciar-sesion>.

### Acceder al panel de administración

Inicia sesión en la aplicación con tus credenciales correctas siguiendo la guía "Iniciar sesión en la aplicación", con esto accederás a la dirección <https://filessoft.com/dashboard/inicio> donde podrás ver un resumen de tu perfil y las diferentes secciones.

**Figura 163***Panel de administración de Filesoft*

*Fuente.* Capturado con el software Scribehow.

## Gestionar el perfil de usuario

Esta guía proporciona instrucciones paso a paso sobre cómo ver la información de tu perfil, actualizar tus datos, gestionar las sesiones activas y cerrar la sesión que estás utilizando.

### *Ver información del perfil*

Accede al panel de administración en la dirección <https://filesoft.com/dashboard/inicio>, da clic en el componente donde se ubica tú foto de perfil, tú nombre y tú correo electrónico, lo que desplegará un panel en la parte inferior derecha con más información.

## Figura 164

*Componente de gestión de perfil*

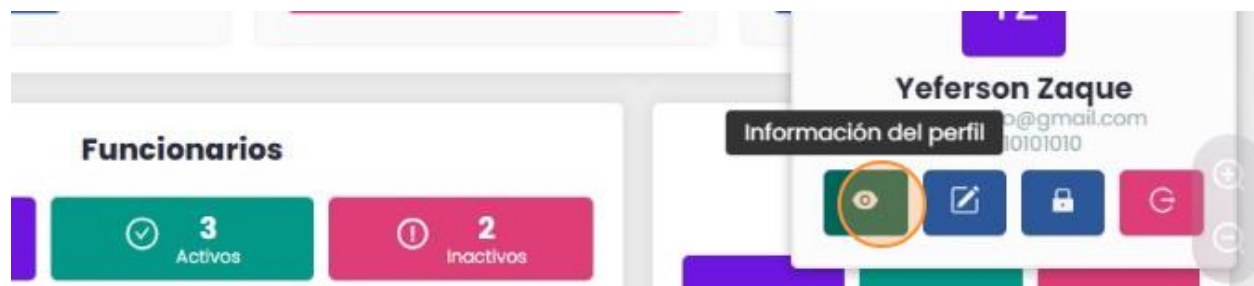


*Fuente.* Capturado con el software Scribehow.

Da clic en el botón de "Información de perfil" ubicado en el panel que se desplegó y visualiza la información de tú perfil, como la información que registraste y los permisos que tienes asignados a tú perfil.

## Figura 165

*Acción para visualizar la información del perfil*



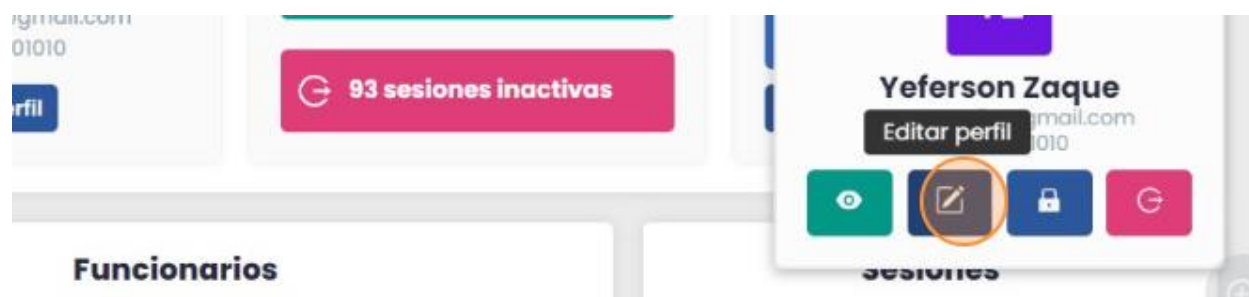
*Fuente.* Capturado con el software Scribehow.

### *Actualizar información de perfil*

Ingresa al panel de administración en cualquier ruta, por ejemplo, a <https://filessoft.com/dashboard/inicio>, da clic en el componente donde se ubica tú foto de perfil, tú nombre y tú correo electrónico, lo que desplegará un panel en la parte inferior derecha con más información, y en el panel desplegado da clic en el botón de "Editar perfil" ubicado en el panel que se desplegó.

**Figura 166**

*Acción para actualizar la información del perfil*



*Fuente.* Capturado con el software Scribehow.

Puedes actualizar tus nombres y apellidos directamente, para otros datos debes realizar la solicitud a un administrador, además puedes cargar una nueva foto de perfil, la cual reemplazará tu foto de perfil actual, cuando estés seguro para guardar los cambios da clic en el botón "Actualizar funcionario".

**Figura 167**

*Clic en el botón para guardar los cambios en el perfil*



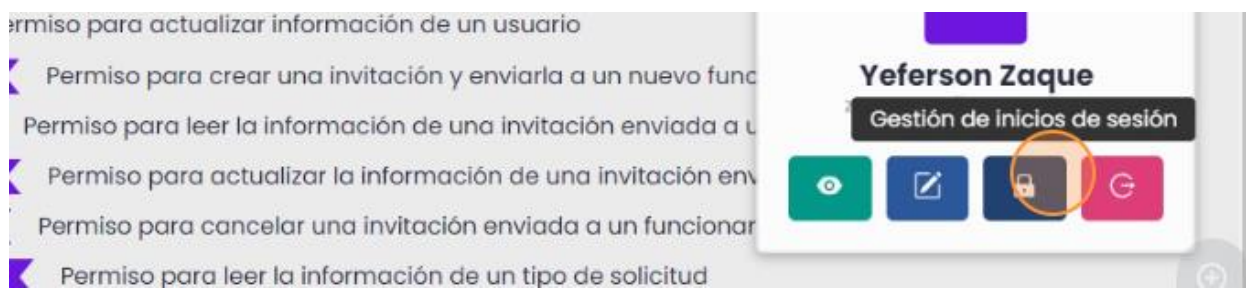
*Fuente.* Capturado con el software Scribehow.

### ***Gestión de sesiones***

Ingresa al panel de administración en cualquier ruta, por ejemplo, a <https://filessoft.com/dashboard/auth/perfil>, da clic en el componente donde se ubica tú foto de perfil, tú nombre y tú correo electrónico, lo que desplegará un panel en la parte inferior derecha con más información y da clic en el botón de "Gestión de inicios de sesión" ubicado en el panel que se desplegó.

### **Figura 168**

*Acción para acceder a la gestión de sesiones*

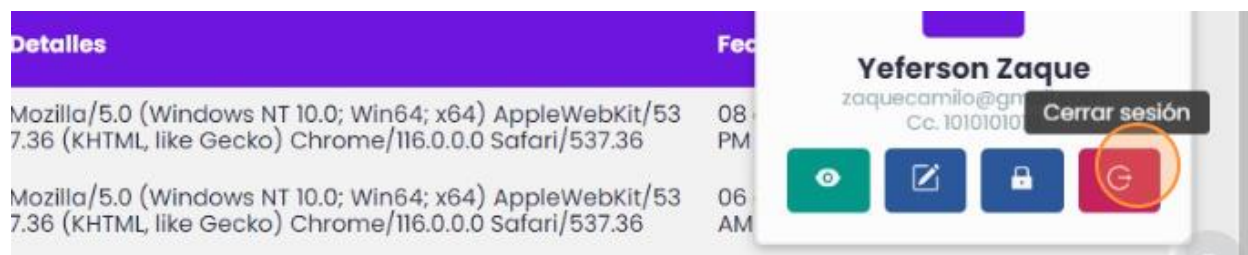


*Fuente.* Capturado con el software Scribehow.

### ***Cerrar sesión***

Ingresa al panel de administración en cualquier ruta, por ejemplo, a <https://filessoft.com/dashboard/auth/perfil>, da clic en el componente donde se ubica tú foto de perfil, tú nombre y tú correo electrónico, lo que desplegará un panel en la parte inferior derecha con más información y da clic en el botón de "Cerrar sesión" ubicado en el panel que se desplegó lo cual te dirigirá a la página de inicio de sesión.



**Figura 169***Acción de cerrar sesión*

*Fuente.* Capturado con el software Scribehow.

## Gestiona los radicados

Esta guía proporciona instrucciones paso a paso sobre cómo visualizar la lista de radicados registrados de forma sencilla. También se explica cómo realizar diversas acciones, como imprimir, exportar datos a distintos formatos de archivo, filtrar información en función de fechas y palabras clave, y ampliar los detalles de un documento concreto, al igual que la manera de actualizarlo o eliminarlo.

### *Visualizar listado de radicados*

En el panel de administración si tienes los permisos necesarios puedes dar clic en en botón de "Radicados".

**Figura 170***Navegar al listado de radicados*

*Fuente.* Capturado con el software Scribehow.

Visualiza el listado de radicados registrados de manera sencilla dentro de la tabla, y si lo requieres puedes imprimir o exportar los datos a un archivo Excel, PDF y CSV o puedes dar clic en "Mostrar registros desde ..." y "Mostrar registros hasta..." para realizar un filtro de la información entre ese rango de fechas.

**Figura 171**

*Filtrado por fechas y acciones de exportación*



*Fuente.* Capturado con el software Scribehow.

También puedes buscar por palabras clave la información que deseas filtrar y encontrar rápidamente.

**Figura 172**

*Buscador de registros por palabras clave*



*Fuente.* Capturado con el software Scribehow.

Puedes dar clic en el icono para expandir la información del radicado y ver más datos sin tener que visualizarlo directamente.

### Figura 173

*Expandir información de un registro de la tabla*



Id	Peticionario	Fecha radicación	Fecha límite	Tipo oficina	Estado
1	Zacker Ciudadano	06 de septiembre de 2023, 11:17:52 PM	27 de septiembre de 2023, 05:00:00 PM	Invitaciones	Completado

Expandir la información de este radicado

*Fuente.* Capturado con el software Scribehow.

### *Visualizar información de radicado*

Navega al listado de radicados en la dirección <https://filessoft.com/dashboard/radicados> y da clic en el icono "Ver detalles" para ver la información más detallada de un radicado.

### Figura 174

*Acción de ver información de un radicado*



Id	Peticionario	Fecha radicación	Fecha límite	Tipo oficina	Estado
1	Zacker Ciudadano	06 de septiembre de 2023, 11:17:52 PM	27 de septiembre de 2023, 05:00:00 PM	Invitaciones	Completado

Ver detalles

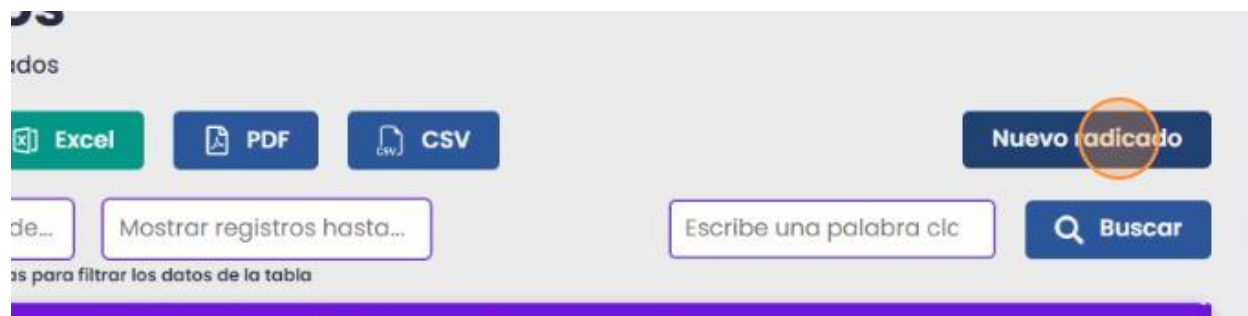
*Fuente.* Capturado con el software Scribehow.

### *Generar radicado*

Navega al listado de radicados en la dirección <https://filessoft.com/dashboard/radicados> y da clic en el botón "Nuevo radicado".

**Figura 175**

*Botón para genera un nuevo radicado*

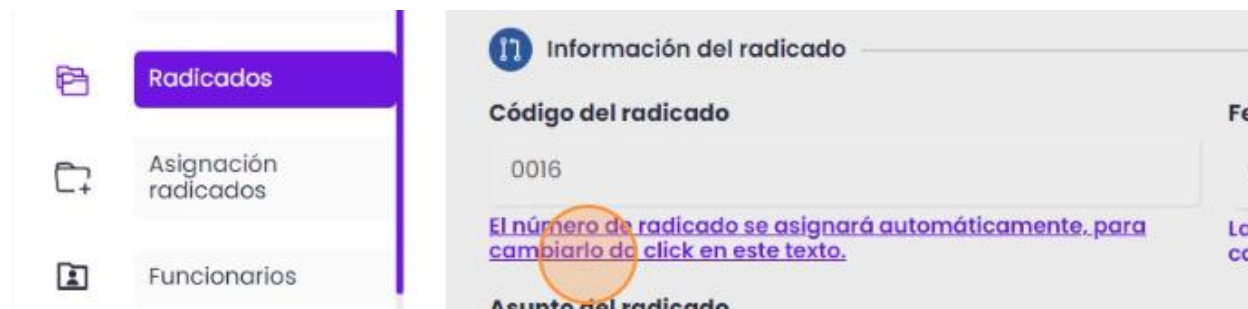


*Fuente.* Capturado con el software Scribehow.

El número de radicado se asigna de manera automática, pero si deseas añadirlo de manera manual, da clic sobre "El número de radicado se asignará de manera automática, para cambiarlo da clic en este texto." el cual hará editable el campo de "Número de radicado".

**Figura 176**

*Cambiar número de radicado automático*

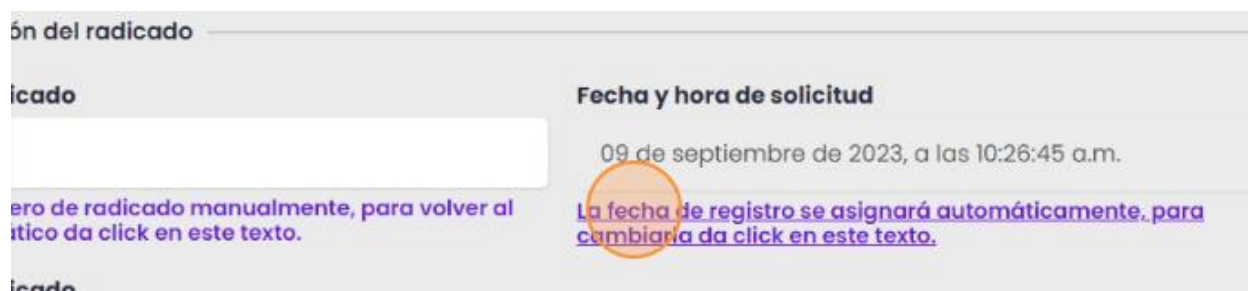


*Fuente.* Capturado con el software Scribehow.

La fecha del radicado se asigna de manera automática, si deseas cambiarla da clic sobre el texto "La fecha de registro se asignará automáticamente, para cambiarla da clic en este texto." para seleccionar manualmente.

## Figura 177

### *Cambiar fecha de radicación automática*



Asunto del radicado

Asunto del radicado

Fecha y hora de solicitud

09 de septiembre de 2023, a las 10:26:45 a.m.

Para registrar manualmente, para volver al registro da click en este texto.

La fecha de registro se asignará automáticamente, para cambiarla da click en este texto.

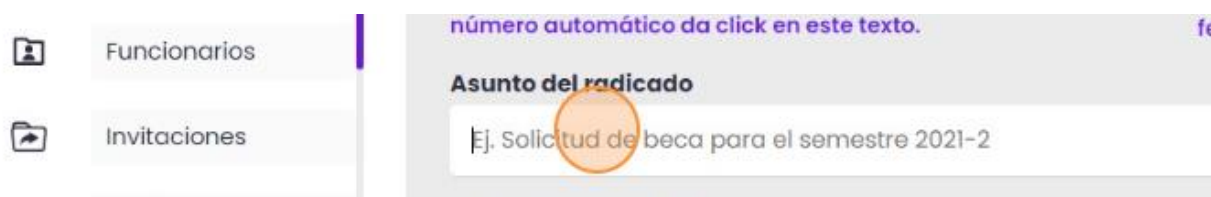
Asunto del radicado

*Fuente.* Capturado con el software Scribehow.

Da clic sobre el campo de "Asunto del radicado" y escribe el asunto del radicado, por ejemplo "Asunto de prueba para el manual".

## Figura 178

### *Campo para registrar el asunto del radicado*



Funcionarios

Invitaciones

número automático da click en este texto.

Asunto del radicado

Ej. Solicitud de beca para el semestre 2021-2

*Fuente.* Capturado con el software Scribehow.

Da clic sobre el campo de "Observaciones" y digita las observaciones del radicado, por ejemplo "Sin observaciones", este campo es opcional.

## Figura 179

### *Campo para registrar las observaciones del radicado*



Observaciones Opcional

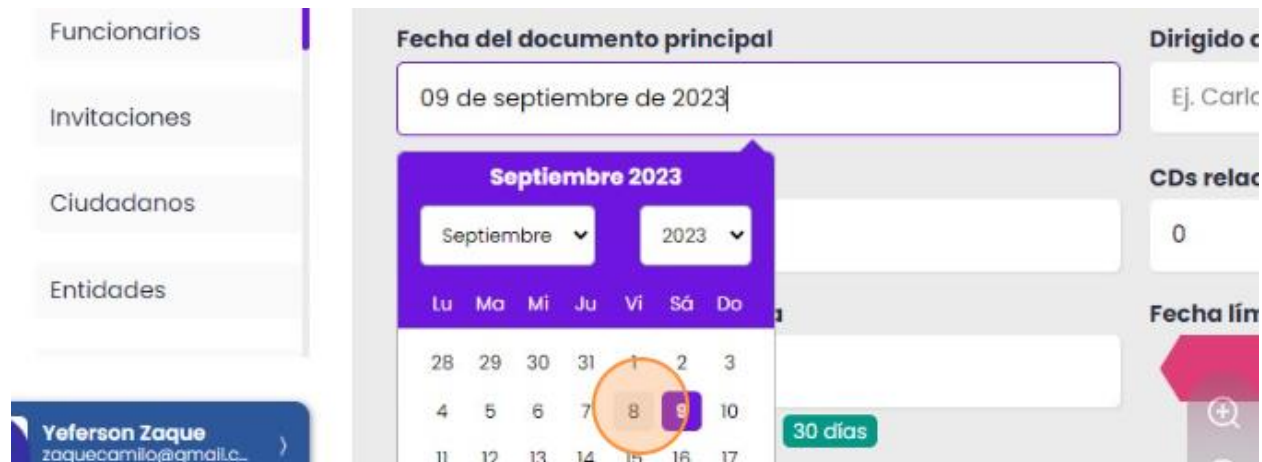
Ej. Se entrega en la oficina de bienestar universitario

*Fuente.* Capturado con el software Scribehow.

Da clic sobre el campo "Seleccione la fecha del documento principal" y selecciona dentro del calendario la fecha asociada al documento principal que se va a radicar.

### Figura 180

*Seleccionar fecha del documento principal*



*Fuente.* Capturado con el software Scribehow.

Da clic sobre el campo "Dirigido a" y digita el nombre de la persona a quién le debe llegar el radicado, por ejemplo "Persona de prueba".

### Figura 181

*Campo para establecer el destinatario del radicado*

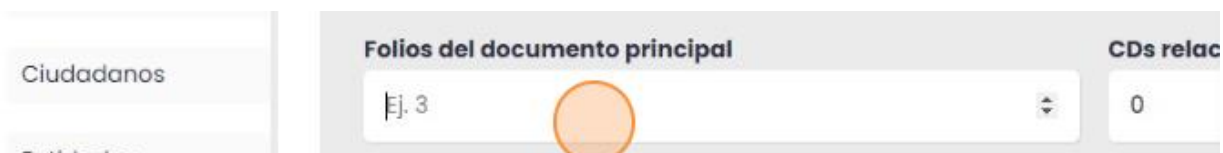


*Fuente.* Capturado con el software Scribehow.

Da clic sobre el campo "Folios del documento principal" y digita el número de folios (hoja) del documento principal, por ejemplo "12".

**Figura 182**

*Campo para establecer los folios del documento principal*



The image shows a portion of a web interface. On the left, there is a sidebar with a button labeled 'Ciudadanos'. The main area contains a form with two input fields. The first field is titled 'Folios del documento principal' and contains the text 'Ej. 3'. The second field is titled 'CDs relacion' and contains the number '0'. An orange circle is drawn around the 'Folios del documento principal' field.

*Fuente.* Capturado con el software Scribehow.

Clic sobre el campo "CDs relacionados" y escribe el número de CDs que se recibieron, por defecto "0".

**Figura 183**

*Establecer CDs recibidos*



The image shows a close-up of a web form. There are two input fields. The first field is labeled 'ento principal' and is empty. The second field is labeled 'CDs relacionados' and contains the number '0'. An orange circle is drawn around the 'CDs relacionados' field.

*Fuente.* Capturado con el software Scribehow.

Da clic sobre el campo "Días hábiles para dar respuesta" y digita o selecciona los días hábiles para dar respuesta a la solicitud, al hacerlo automáticamente se calculará la fecha límite y se mostrará al lado derecho del campo.

**Figura 184**

*Establecer días hábiles para dar respuesta*



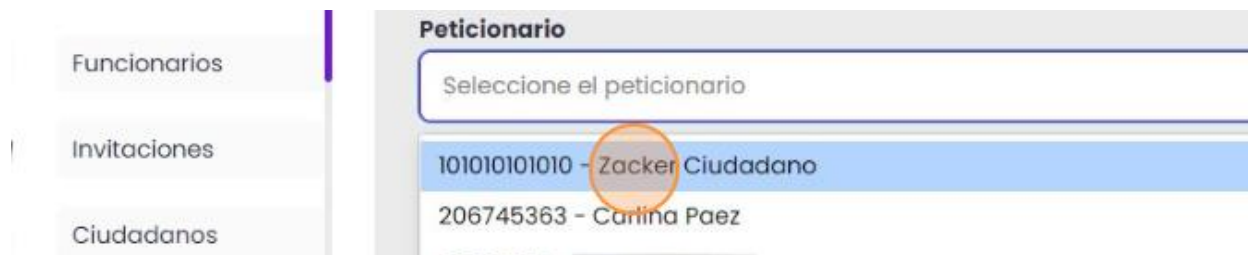
The image shows a web form with a sidebar on the left containing a button labeled 'Entidades' and a name 'Yeferson Zaque'. The main area has a form with a field titled 'Días hábiles para dar respuesta' containing the number '15'. Below this field are five buttons labeled '1 día', '3 días', '10 días', '15 días', and '30 días'. An orange circle is drawn around the 'Días hábiles para dar respuesta' field.

*Fuente.* Capturado con el software Scribehow.

Da clic sobre el campo de "Peticionario" y del listado que se despliegue selecciona el ciudadano, puedes escribir una parte del nombre o del número de identificación y se filtrará.

### Figura 185

*Selección del peticionario que realiza la solicitud*

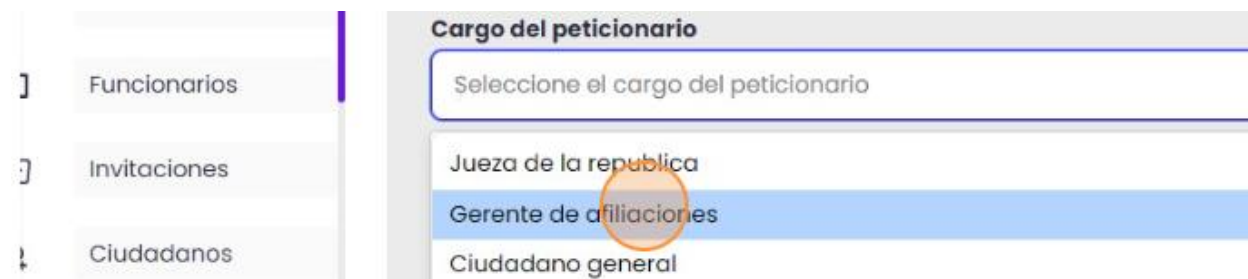


*Fuente.* Capturado con el software Scribehow.

Da clic sobre el campo de "Cargo del peticionario" y del listado que se despliega selecciona uno de los cargos.

### Figura 186

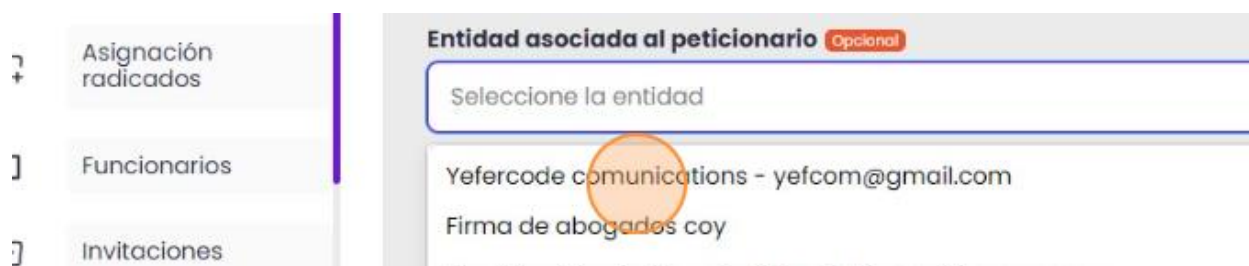
*Selección del cargo del peticionario*



*Fuente.* Capturado con el software Scribehow.

Da clic sobre el campo "Entidad asociada al peticionario" y del listado que se despliega selecciona la entidad, recuerda que este campo es opcional.



**Figura 187***Selección de la entidad asociada al peticionario**Fuente.* Capturado con el software Scribehow.

Da clic sobre el campo "Selecciona o arrastra el archivo a cargar aquí" y selecciona el archivo principal, lo mismo hazlo con los archivos adjuntos, pero ten en cuenta que pueden ser opcionales, solo el documento principal es requerido.

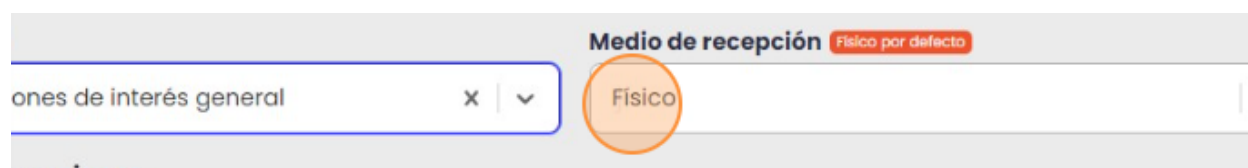
**Figura 188***Carga de documento principal y adjuntos**Fuente.* Capturado con el software Scribehow.

Da clic sobre el campo "Tipo de oficio" y de la lista desplegable selecciona el tipo que se asociará a la solicitud.

**Figura 189***Selección del tipo de oficio*

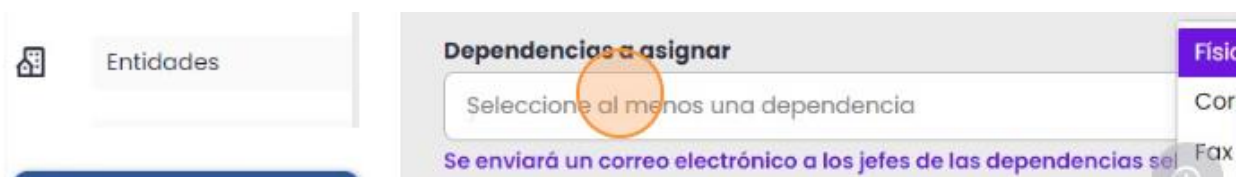
*Fuente.* Capturado con el software Scribehow.

Da clic en el campo de "Medio de recepción" y de la lista desplegable selecciona el modo en que se recibió el radicado, por defecto se tiene que se recibió en "Físico".

**Figura 190***Selección del medio de recepción*

*Fuente.* Capturado con el software Scribehow.

Da clic en el campo "Dependencias a asignar" y selecciona una o varias dependencias a las que se le notificará de la asignación.

**Figura 191***Selección de dependencias a donde se asignará el radicado*

*Fuente.* Capturado con el software Scribehow.

Cuando estés seguro de que deseas registrar la información, da clic en el botón "Generar radicado".

### Figura 192

#### *Generación del radicado*



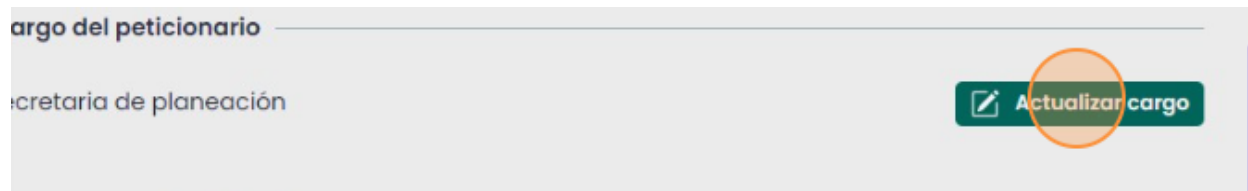
*Fuente.* Capturado con el software Scribehow.

#### *Actualizar cargo del peticionario*

Accede a los detalles de un radicado, por ejemplo, a la dirección <https://filesoft.com/dashboard/radicados/detalles/64f94f3ff40440a3ddb11c22> y da clic en el botón "Actualizar cargo".

### Figura 193

#### *Acción rápida para actualizar el cargo del peticionario*



*Fuente.* Capturado con el software Scribehow.

Si el cargo no está registrado puedes hacerlo directamente dando clic en "Registrar cargo" y en el modal desplegado ingresa la información del cargo y finaliza dando clic en el botón "Registrar cargo" para guardar la información.

### Figura 194

*Formulario para registrar un cargo*

*Fuente.* Capturado con el software Scribehow.

Selecciona del listado de cargos el que deseas establecer como cargo del peticionario y para guardar los cambios da clic en el botón "Actualizar cargo del peticionario".

### Figura 195

*Botón para actualizar el cargo del peticionario*

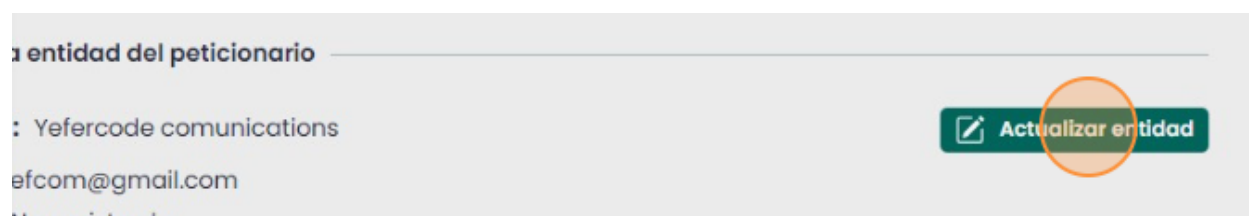
*Fuente.* Capturado con el software Scribehow.

### ***Actualizar entidad del peticionario***

Accede a los detalles de un radicado, por ejemplo, a la dirección <https://filesoft.com/dashboard/radicados/detalles/64f94f3ff40440a3ddb11c22> y da clic en el botón "Actualizar entidad".

### **Figura 196**

*Acción rápida para actualizar la entidad del peticionario*



*Fuente.* Capturado con el software Scribehow.

Si la entidad que deseas asociar no está registrada puedes dar clic en el botón "Registrar entidad", al hacer el registro, selecciona la entidad a asociar y da clic en el botón "Actualizar entidad del peticionario" para guardar los cambios.

### **Figura 197**

*Actualizar la entidad del peticionario*



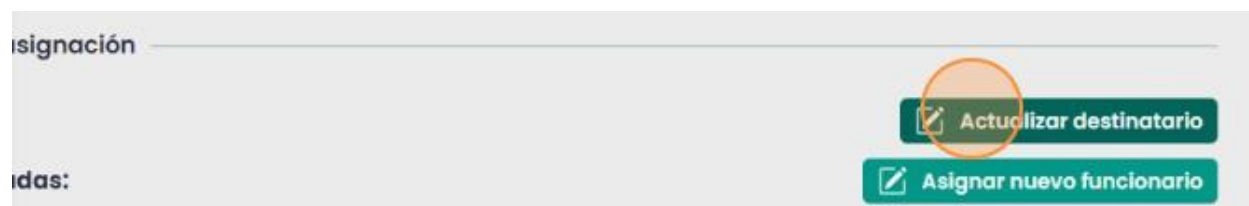
*Fuente.* Capturado con el software Scribehow.

### **Actualizar destinatario**

Accede a los detalles de un radicado, por ejemplo, a la dirección <https://filessoft.com/dashboard/radicados/detalles/64f94f3ff40440a3ddb11c22> y da clic en el botón de "Actualizar destinatario".

### **Figura 198**

*Acción rápida para actualizar el destinatario*



*Fuente.* Capturado con el software Scribehow.

Actualiza la información que creas necesaria y da clic en el botón "Actualizar destinatario del radicado" para guardar los cambios y notificar a los funcionarios asociados.

### **Figura 199**

*Guardar los cambios del destinatario*



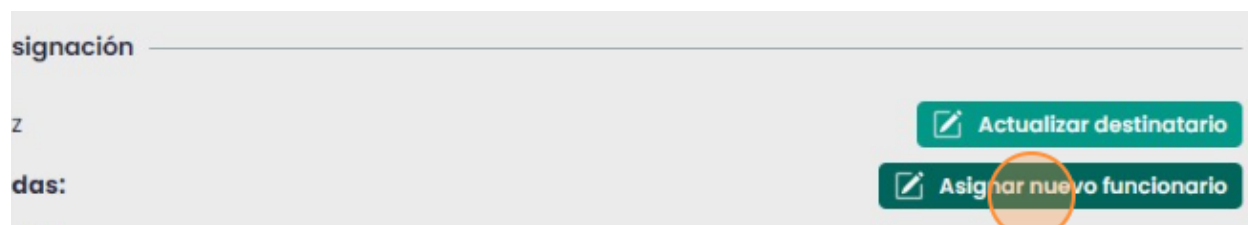
*Fuente.* Capturado con el software Scribehow.

### *Actualizar asignación radicados*

Accede a los detalles de un radicado, por ejemplo, a la dirección <https://filesoft.com/dashboard/radicados/detalles/64f94f3ff40440a3ddb11c22> y da clic en el botón "Asignar nuevo funcionario".

### **Figura 200**

*Acción rápida para actualizar las asignaciones*



*Fuente.* Capturado con el software Scribehow.

Da clic sobre la lista que dice "Seleccione los funcionarios" para desplegar los funcionarios de tu dependencia a los que puedes reasignar o da clic sobre la lista "Seleccione las dependencias" para desplegar todo el listado de dependencias, puedes seleccionar una o varias para notificar a sus administradores.

### **Figura 201**

*Selección de funcionarios y dependencias a asignar*



*Fuente.* Capturado con el software Scribehow.

Da clic en el botón "Generar asignaciones" para realizar la solicitud, guardar los cambios y notificar a los funcionarios implicados.

### Figura 202

*Generar las asignaciones al radicado*



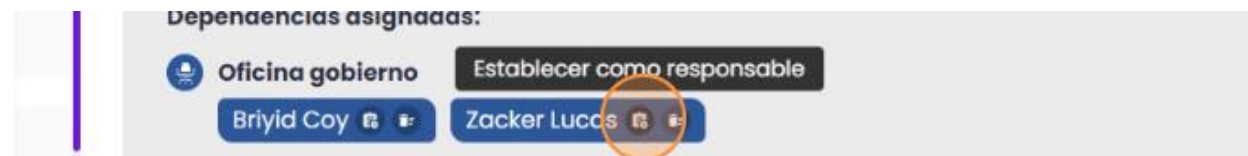
*Fuente.* Capturado con el software Scribehow.

### *Asignar responsable*

Accede a los detalles de un radicado, por ejemplo, a la dirección <https://filessoft.com/dashboard/radicados/detalles/64f94f3ff40440a3ddb11c22> y desplázate hasta la sección "Información de asignación", busca el funcionario que deseas asignar y da clic sobre el icono "Establecer como responsable".

### Figura 203

*Acción rápida para establecer responsable de radicado*



*Fuente.* Capturado con el software Scribehow.

Se te desplegará un modal, si estás seguro de realizar la acción da clic sobre el botón "Asignar".



**Figura 204**

*Confirmar la asignación de responsables*



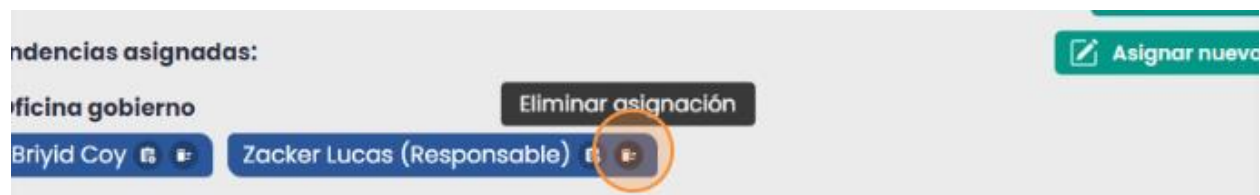
*Fuente.* Capturado con el software Scribehow.

**Eliminar asignación**

Accede a los detalles de un radicado, por ejemplo, a la dirección <https://filesoft.com/dashboard/radicados/detalles/64f94f3ff40440a3ddb11c22> y desplázate hasta la sección "Información de la asignación", ubica el funcionario y da clic en el icono "Eliminar asignación".

**Figura 205**

*Acción de eliminar una asignación de radicado*



*Fuente.* Capturado con el software Scribehow.

Si estás seguro de lo que vas a hacer y entiendes las advertencias dadas, da clic en el botón "Eliminar asignación".

**Figura 206***Confirmar eliminación de asignación*

*Fuente.* Capturado con el software Scribehow.

***Actualizar documento principal***

Accede a los detalles de un radicado, por ejemplo, a la dirección <https://filesoft.com/dashboard/radicados/detalles/64f94f3ff40440a3ddb11c22> y da clic en el botón "Actualizar documento principal".

**Figura 207***Acción de actualizar documento principal*

*Fuente.* Capturado con el software Scribehow.

Da clic sobre el campo "Selecciona o arrastra el archivo a cargar aquí", esto desplegará el gestor de archivos del sistema para que selecciones el documento principal.

## Figura 208

*Componente de carga de archivos*



*Fuente.* Capturado con el software Scribehow.

Si lo deseas y quieres verificar puedes previsualizar el documento seleccionado dando clic en este icono.

## Figura 209

*Previsualización de archivo cargado*



*Fuente.* Capturado con el software Scribehow.

Da clic en el campo "Folios del documento principal" y digita la cantidad de folios (hojas) que tiene el documento seleccionado, cuando estés seguro da clic en el botón "Actualizar documento principal" para guardar los cambios.

**Figura 210**

*Guardar los cambios en el documento principal*



*Fuente.* Capturado con el software Scribehow.

**Cargar archivos adjuntos**

Accede a los detalles de un radicado, por ejemplo, a la dirección <https://filesoft.com/dashboard/radicados/detalles/64f94f3ff40440a3ddb11c22> y da clic sobre el botón "Cargar adjuntos".

**Figura 211**

*Acción de cargar adjuntos*



*Fuente.* Capturado con el software Scribehow.

Da clic sobre el campo "Selecciona o arrastra el archivo a cargar aquí", esto desplegará el gestor de archivos del sistema para seleccionar los archivos adjuntos a cargar, los documentos seleccionados aparecerán y podrán ser visualizados antes de cargarlos, si ya estás seguro da clic en el botón "Cargar adjuntos" para guardar los cambios y asociar los archivos.

**Figura 212**

*Confirmar carga de archivos adjuntos*



*Fuente.* Capturado con el software Scribehow.

**Actualizar número CDs**

Accede a los detalles de un radicado, por ejemplo, a la dirección <https://filesoft.com/dashboard/radicados/detalles/64f94f3ff40440a3ddb11c22> y da clic sobre el botón "Actualizar CDs".

**Figura 213**

*Acción de actualizar CDs*



*Fuente.* Capturado con el software Scribehow.

Da clic sobre el campo "CDs relacionados" y digita el número de CDs que se recibieron y se asociarán al radicado, si estás seguro del cambio da clic en el botón "Actualizar CDs asociados" para guardar la información.

**Figura 214***Confirmar actualización de CDs*

*Fuente.* Capturado con el software Scribehow.

### ***Visualizar línea de tiempo***

Accede a los detalles de un radicado, por ejemplo, a la dirección <https://filesoft.com/dashboard/radicados/detalles/64f94f3ff40440a3ddb11c22> y en la parte inferior da clic sobre el icono "Ver timeline del radicado" el cual te llevará a la página donde se expone esta línea de tiempo.

**Figura 215***Acción de visualizar línea de tiempo*

*Fuente.* Capturado con el software Scribehow.

Visualiza la línea de tiempo del radicado donde se registra cada una de las acciones que se van realizando sobre el radicado.

## Figura 216

*Línea de tiempo de radicado en panel de administración*

**Línea de tiempo de radicado** 0015

ard > Timeline > 64f94f3ff40440a3ddb11c22

0 1 5 Ir al listado de radicados

**Radicación:** 06 de septiembre de 2023, 11:17:52 PM  
**Origenario:** Zacker Ciudadano  
**Radicado por:** Yeferson Zaque  
**Estado:** Con respuesta

**6 de septiembre de 2023, 11:19:11 PM**  
 Se ha creado la solicitud con el código 0015 y el asunto Última prueba de email.

**6 de septiembre de 2023, 11:19:11 PM**  
 Se ha asianado la solicitud con el código 0015 a las dependencias comisaría de familia.

*Fuente.* Capturado con el software Scribehow.

## *Actualizar información de radicado*

Accede a los detalles de un radicado, por ejemplo, a la dirección <https://filesoft.com/dashboard/radicados/detalles/64f94f3ff40440a3ddb11c22> y da clic en el botón "Editar radicado #####" para direccionarte al formulario de edición.

## Figura 217

*Acción de actualizar información de radicado*

ma prueba de email

o: Invitaciones

**Radicación:** 06 de septiembre de 2023

**Radicado por:** Yeferson Zaque

15 días hábiles

Editar radicado 0015

Editar

*Fuente.* Capturado con el software Scribehow.

En el formulario de edición podrás cambiar elementos como el número de radicado, la fecha de edición, el asunto, las observaciones, la fecha del documento principal, la persona a quién va dirigido, y demás información; cuando estés seguro de realizar la actualización de los datos da clic en el botón "Actualizar radicado" el cual guardará la información.

### Figura 218

*Guardar cambios en la información del radicado*



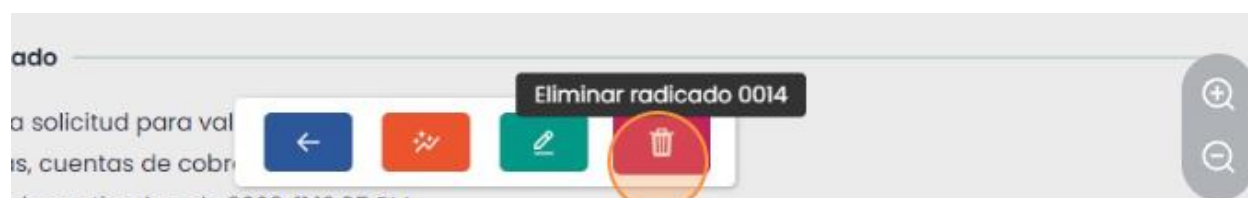
*Fuente.* Capturado con el software Scribehow.

### Eliminar radicado

Accede a los detalles de un radicado, por ejemplo, a la dirección <https://filesoft.com/dashboard/radicados/detalles/64f94f3ff40440a3ddb11c22>, da clic en el icono "Ver detalles" y en la parte inferior de los detalles del radicado da clic en el icono "Eliminar radicado #####".

### Figura 219

*Acción de eliminar radicado*



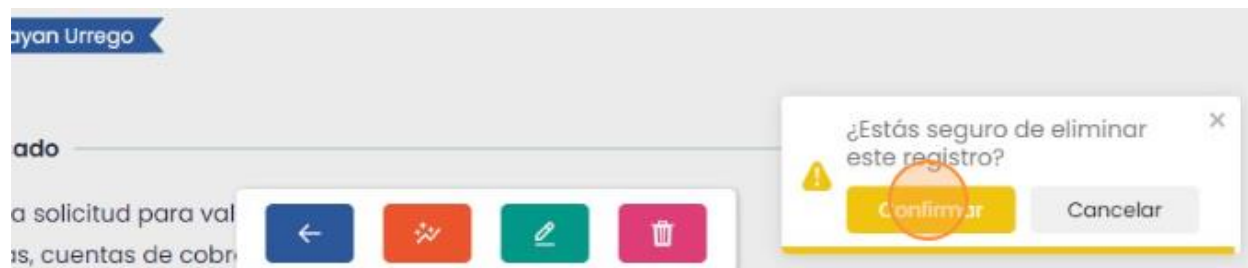
*Fuente.* Capturado con el software Scribehow.



Si estás seguro de eliminar el registro, da clic en el botón "Confirmar" de la alerta emergente.

## Figura 220

### *Confirmación de eliminación de radicado*



*Fuente.* Capturado con el software Scribehow.

## Gestiona los archivos

Esta guía proporciona instrucciones paso a paso sobre cómo previsualizar archivos e imprimir documentos cargados y asociados a los registros, descargar los archivos y eliminarlos.

### *Previsualizar archivos*

Navega a cualquier página donde se tengan archivos cargados y se despliegue el listado de archivos, como por ejemplo la dirección <https://filsoft.com/dashboard/radicados/detalles/64f94f3ff40440a3ddb11c22> y ubica el componente donde se listan los archivos, este por lo general tiene el nombre del archivo, la extensión y tres iconos de acción, una vez hayas ubicado el componente donde se listan los archivos, da clic sobre el icono "Visualizar archivo".

## Figura 221

*Acción de previsualización de archivos*



*Fuente.* Capturado con el software Scribehow.

Si el archivo es extensión PDF tienes la posibilidad de imprimirlo dando clic en el botón "Imprimir documento" el cual desplegará la ventana de impresión del navegador.

## Figura 222

*Clic en el botón para imprimir el documento*



*Fuente.* Capturado con el software Scribehow.

Para cerrar el modal puedes dar clic en la zona gris del modal o directamente dar clic sobre el icono "X".

## Figura 223

*Formas de cerrar el modal*



*Fuente.* Capturado con el software Scribehow.

### ***Descargar archivo***

Navega a cualquier página donde se tengan archivos cargados y se despliegue el listado de archivos, como por ejemplo la dirección <https://filsoft.com/dashboard/radicados/detalles/64f94f3ff40440a3ddb11c22>, ubica el componente donde se listan los archivos y da clic sobre el icono "Descargar archivo" para que se descargue automáticamente o se despliegue la ventana de confirmación del navegador.

### **Figura 224**

*Acción de descargar el archivo*



*Fuente.* Capturado con el software Scribehow.

Puedes abrir el documento en otra ventana del navegador, el archivo descargado tendrá un nombre similar a "2023-09-09-0015-L8bhe-main-file.pdf".

### ***Eliminar archivo***

Navega a cualquier página donde se tengan archivos cargados y se despliegue el listado de archivos, como por ejemplo la dirección <https://filsoft.com/dashboard/radicados/detalles/64f94f3ff40440a3ddb11c22>, ubica el componente donde se listan los archivos y da clic sobre el botón "Eliminar archivo" para desplegar el modal de confirmación.

## Figura 225

*Acción de eliminar archivos*



*Fuente.* Capturado con el software Scribehow.

Si has leído la advertencia y estás seguro de eliminar el archivo de forma definitiva da clic sobre el botón "Eliminar".

## Figura 226

*Confirmar eliminación de archivo*



*Fuente.* Capturado con el software Scribehow.

## Gestiona las asignaciones de radicados

Esta guía proporciona instrucciones paso a paso sobre cómo ver la lista de asignaciones de radicados, generar incidencias y establecer las respuestas a los radicados.

### *Ver listado de asignaciones*

Ve en tu navegador al panel de administración, como por ejemplo a la dirección a dirección <https://filessoft.com/dashboard/inicio> y da clic en el botón "Asignación radicados".

**Figura 227**

*Ver listado de asignaciones de radicados*



*Fuente.* Capturado con el software Scribehow.

### *Ver asignación*

Ve en tú navegador al panel de administración, como por ejemplo a la dirección a dirección <https://filessoft.com/dashboard/inicio> y da clic en el icono "Ver detalles" para acceder a la información de la asignación.

**Figura 228**

*Acción para visualizar los detalles de una asignación de radicados*

to	Peticionario	Fecha radicación	Fecha límite	Tipo oficio	Estado
rando la a de ación	Zacker Ciudadano	06 de septiem- bre de 2023, 11:00:15 PM	27 de sep- tiembre de 2023, 05:00:00 PM	Comunica- ciones de interés general	Pendiente <span>Ver detalles</span>

*Fuente.* Capturado con el software Scribehow.

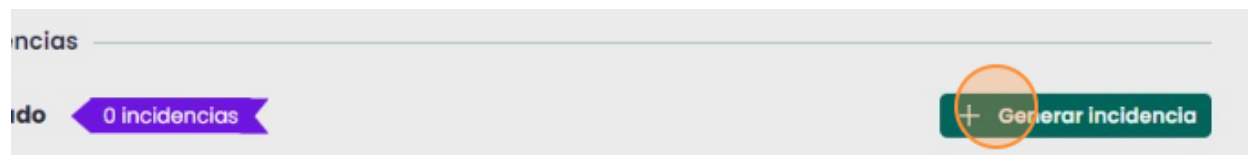
Visualiza la información del radicado que se te asignó, tendrás la posibilidad de editar los días para dar respuesta, cambiar el peticionario, actualizar el cargo y la entidad del peticionario, actualizar el destinatario, reasignar a nuevos funcionarios o dependencias, actualizar el documento principal, cargar documentos adjuntos y editar la cantidad de CDs asociados, estas guías las puedes encontrar en la sección de "Radicados".

### **Generar incidencia**

Navega hasta la página de detalles de la asignación donde se muestra la información detallada del radicado, por ejemplo <https://filessoft.com/dashboard/asignacion-radicados/detalles/64f94c40559e8864e5a251cc> y da clic en el botón de "Generar incidencia".

### **Figura 229**

*Acción de generar incidencia*



*Fuente.* Capturado con el software Scribehow.

Da clic en el campo "Consecutivo de la incidencia" y escribe el número del consecutivo, por ejemplo "0002".

### **Figura 230**

*Campo para digitar el consecutivo de la incidencia*



*Fuente.* Capturado con el software Scribehow.

Da clic en el campo "Asunto de la incidencia" y describe el asunto de la incidencia, por ejemplo "Incidencia de prueba".

**Figura 231**

*Campo para registrar el asunto de la incidencia*



*Fuente.* Capturado con el software Scribehow.

Da clic en el campo de "Contenido de la incidencia", el cual tiene un editor de Mark Down para que puedas dar estilos a lo que digites y escribe el contenido de la incidencia, por ejemplo "Conoce como se genera una incidencia en la aplicación.", ten en cuenta que es muy importante que expliques a detalle las razones para su generación.

**Figura 232**

*Editor de Mark Down para ingresar el contenido de la incidencia*



*Fuente.* Capturado con el software Scribehow.

Da clic en el clic en el campo de "Documentos adjuntos" para seleccionar los archivos que se asociarán a la incidencia, se recomienda que sea un archivo PDF, en caso de que sea otro tipo de archivo solo se aceptan formatos de imagen, PDF y cualquier tipo de documento del paquete de Office; cuando estés seguro de lo que registraste, da clic en el botón "Generar incidencia" para almacenar la información de la incidencia.

**Figura 233***Guardar la información de la incidencia**Fuente.* Capturado con el software Scribehow.**Actualizar incidencia**

Navega hasta la página de detalles de la asignación donde se muestra la información detallada del radicado, por ejemplo <https://filessoft.com/dashboard/asignacion-radicados/detalles/64f94c40559e8864e5a251cc> y da clic en el botón "Editar incidencia", ten en cuenta que este solo aparecerá si tú fuiste el que generaste la incidencia.

**Figura 234***Acción de editar una incidencia**Fuente.* Capturado con el software Scribehow.

Actualiza la información que creas necesario, ya sea el consecutivo, el asunto, el contenido o cargar más archivos adjuntos y cuando estés seguro de la información ingresada da clic en el botón "Actualizar incidencia" para almacenar los cambios.



**Figura 235**

*Guardar los cambios de la incidencia*



*Fuente.* Capturado con el software Scribehow.

**Eliminar incidencia**

Navega hasta la página de detalles de la asignación donde se muestra la información detallada del radicado, por ejemplo <https://filessoft.com/dashboard/asignacion-radicados/detalles/64f94c40559e8864e5a251cc> y da clic en el botón "Eliminar incidencia".

**Figura 236**

*Acción para eliminar una incidencia*



*Fuente.* Capturado con el software Scribehow.

Lee la advertencia y si estás seguro de continuar da clic en el botón de "Eliminar".

**Figura 237**

*Confirmación de eliminación de incidencia*



*Fuente.* Capturado con el software Scribehow.

### ***Establecer respuesta***

Navega hasta la página de detalles de la asignación donde se muestra la información detallada del radicado, por ejemplo <https://filessoft.com/dashboard/asignacion-radicados/detalles/64f94c40559e8864e5a251cc>, sigue los pasos de la guía "Asignación radicado - Generar incidencia" para comenzar el proceso y cuando estés llenando la información de la incidencia, para establecer que será la respuesta del radicado da clic en "¿Es la respuesta del radicado? (No)" y verifica que ha cambiado a "(Si)".

### **Figura 238**

*Marcar como respuesta del radicado a la incidencia actual*

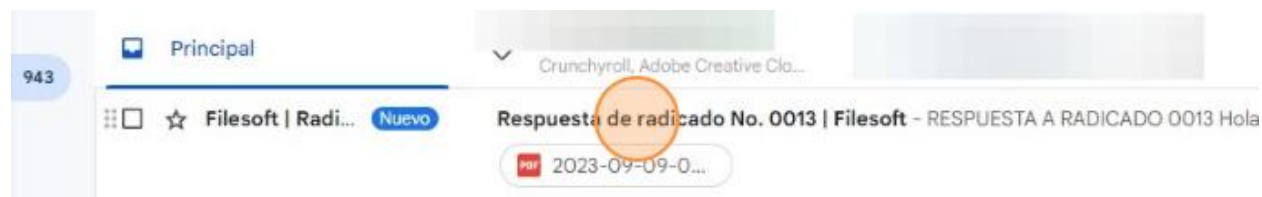


*Fuente.* Capturado con el software Scribehow.

Una vez hayas generado la incidencia y se te informe que se generó de manera correcta, al peticionario le llegará un correo electrónico a su bandeja de entrada, con el asunto "Respuesta de radicado No. ##### | Filessoft" como por ejemplo "Respuesta de radicado No. 0013 | Filessoft".

### **Figura 239**

*Recepción de correo de respuesta*



*Fuente.* Capturado con el software Scribehow.

El contenido del correo informa que se ha dado respuesta a su solicitud, además de incluir los archivos adjuntos que se cargaron en la aplicación y le da la posibilidad de acceder y visualizar la línea de tiempo.

### Figura 240

*Contenido de la respuesta del radicado*



*Fuente.* Capturado con el software Scribehow.

### Gestiona las invitaciones

Esta guía proporciona instrucciones paso a paso sobre cómo generar una invitación para un funcionario, esto asignando los permisos que tendrá al seguir el proceso de registro, además de poder ver el listado, actualizar y eliminar las invitaciones.

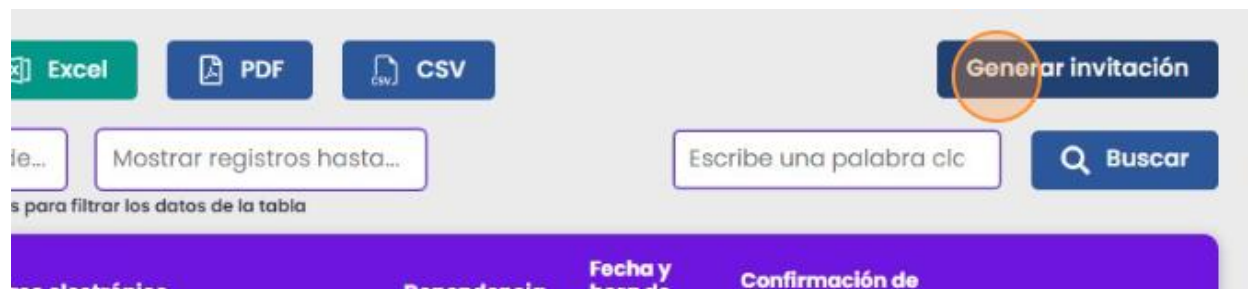
### ***Invitar funcionario***

Navega al listado de invitaciones en la dirección

<https://filesoft.com/dashboard/invitaciones> y da clic en el botón "Generar invitación".

### **Figura 241**

*Acción de invitar funcionario*

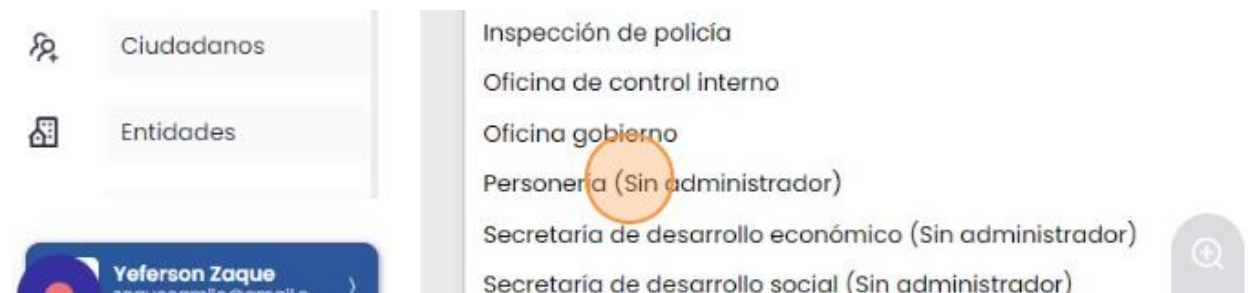


*Fuente.* Capturado con el software Scribehow.

Da clic en el campo "Nombre del funcionario" y Digita el nombre del funcionario que deseas invitar, por ejemplo "Usuario Prueba", luego da clic sobre el campo "Correo electrónico" y digita el correo del funcionario que se registrará y a donde se enviará la invitación, luego da clic sobre el campo "Dependencias a asignar" para desplegar el listado de dependencias registradas, se indica si no tienen administrador o si están inactivas, ten en cuenta que si no están activas no podrás generar la invitación.

### **Figura 242**

*Listado de dependencias para asignar al funcionario*



*Fuente.* Capturado con el software Scribehow.

Si el funcionario va a ser administrador de la dependencia, da clic sobre el campo "¿Es administrador? (No)" y verifica que ha cambiado a (Si).

### Figura 243

*Establecer como administrador el funcionario invitado*

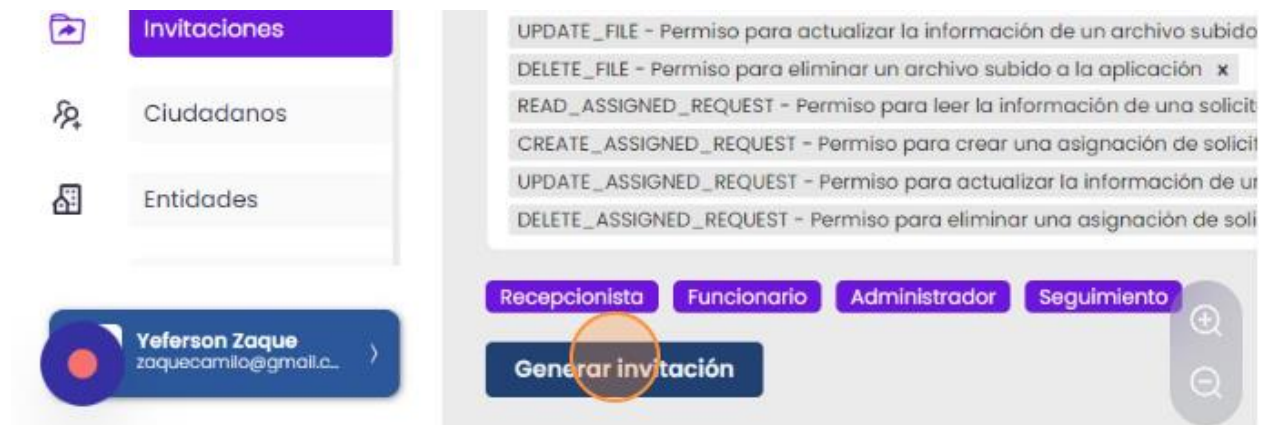


*Fuente.* Capturado con el software Scribehow.

Da clic sobre el campo "Permisos a asignar" para desplegar el listado de permisos, o selecciona uno de los perfiles preestablecidos que asignará los permisos automáticamente y si estás seguro de que deseas generar la invitación y los datos ingresados son correctos, da clic en el botón "Generar invitación".

### Figura 244

*Guardar la información de la invitación y notificar*



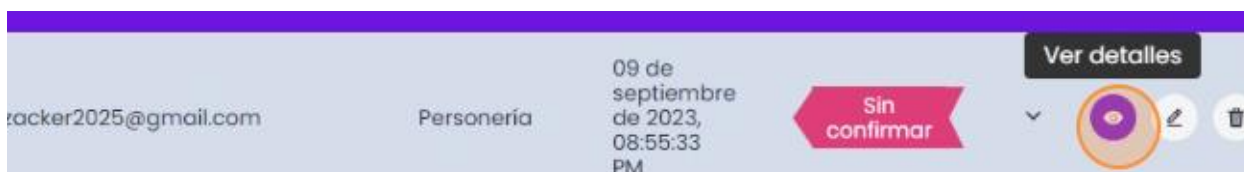
*Fuente.* Capturado con el software Scribehow.

### ***Ver invitación***

Navega al listado de invitaciones en la ruta <https://filesoft.com/dashboard/invitaciones>, da clic en el icono "Ver detalles" para acceder a los detalles de la invitación y visualiza la información de la invitación, como la información básica registrada del funcionario y los permisos asignados.

### **Figura 245**

#### *Acción de ver detalles de invitación*



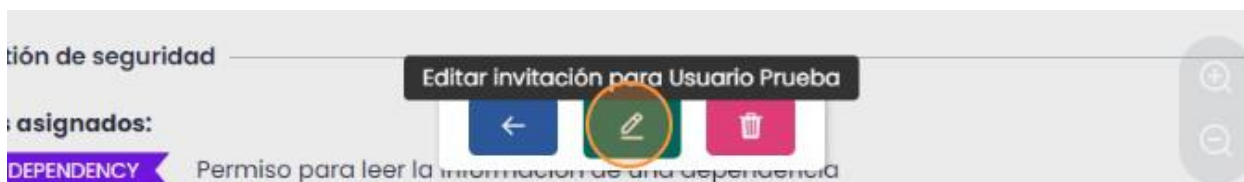
*Fuente.* Capturado con el software Scribehow.

### ***Actualizar invitación***

Navega al listado de invitaciones o a los detalles de una invitación en la ruta <https://filesoft.com/dashboard/invitaciones/detalles/64fd22150cf39e83d7a7acb8> y en la parte inferior da clic en el icono "Editar invitación para ####" donde se indica el nombre del funcionario invitado.

### **Figura 246**

#### *Acción de actualizar invitación*



*Fuente.* Capturado con el software Scribehow.

En cada uno de los campos que sea necesario actualiza la información de la invitación, como el nombre del funcionario, el correo, la dependencia, si será administrador y los permisos asignados y si estás seguro de los datos ingresados, da clic en el botón "Actualizar invitación" para guardar los cambios.

### Figura 247

*Guardar los cambios en la información de la invitación*



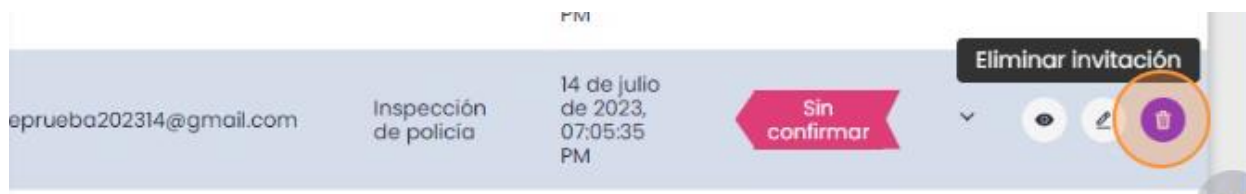
*Fuente.* Capturado con el software Scribehow.

### ***Eliminar invitación***

Navega al listado de invitaciones en la ruta <https://filessoft.com/dashboard/invitaciones> y da clic en el icono "Eliminar invitación".

### Figura 248

*Acción de eliminar una invitación*



*Fuente.* Capturado con el software Scribehow.

Si estás seguro y conoces los riesgos de realizar la eliminación, en la alerta da clic en el botón "Confirmar" para eliminar la invitación.

## Figura 249

### *Confirmación de eliminación de invitación*



*Fuente.* Capturado con el software Scribehow.

## Gestiona los funcionarios

Esta guía proporciona instrucciones paso a paso sobre cómo registrar a un empleado, desde la navegación a la página de registro, la introducción de la información necesaria y la verificación del correo, además de permitir su edición, visualización y eliminación.}

### **Registrar funcionario**

Accede al gestor de correo que utilices, por ejemplo [[Gmail]], verifica que te haya llegado un correo con asunto "Registro de funcionario", abre el correo y da clic sobre el botón "Realizar registro".

## Figura 250

### *Contenido del correo de registro de funcionario*



*Fuente.* Capturado con el software Scribehow.



Esto te llevará a la página donde se verificará que la invitación es válida, si todo sale correcto da clic en el botón "Realizar registro de información".

### Figura 251

*Verificación de invitación exitosa*



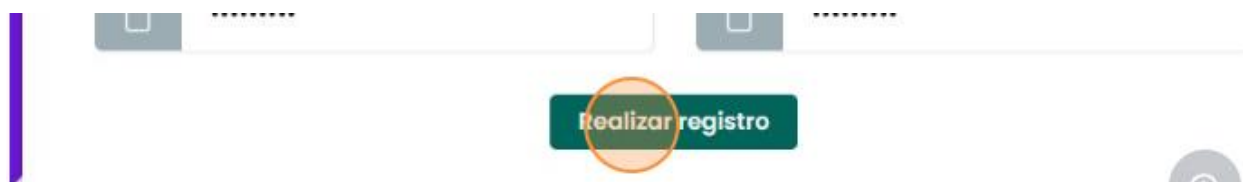
*Fuente.* Capturado con el software Scribehow.

Esto te llevará al formulario de registro, en él da clic sobre el campo "Número de identificación" y escribe tu número de identificación, por ejemplo "20202020", da clic sobre el campo "Nombres" y escribe tus nombres, posteriormente da clic en el campo "Apellidos" y digita tus apellidos.

Da clic sobre el campo "Contraseña" y digita la contraseña con la que accederás al sistema y finaliza dando clic sobre el campo "Confirmar contraseña" y digita la misma contraseña que ingresaste en el paso anterior, cuando estés seguro de los datos ingresados da clic en el botón "Realizar registro" para comenzar la solicitud de registro.

### Figura 252

*Guardar información del funcionario*

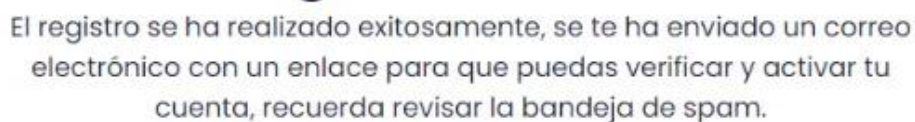


*Fuente.* Capturado con el software Scribehow.

Si todo sale de manera correcta, se te informa que se ha enviado un correo de confirmación y tienes la posibilidad de ir a la página de inicio dando clic en el botón "Ir a inicio".

### Figura 253

*Confirmación de registro de funcionario exitoso*



El registro se ha realizado exitosamente, se te ha enviado un correo electrónico con un enlace para que puedas verificar y activar tu cuenta, recuerda revisar la bandeja de spam.



Ir a inicio

*Fuente.* Capturado con el software Scribehow.

Ve a tú gestor de correos, te llegará un nuevo correo con asunto "Verificación de correo", entra en él y da clic en el botón "Verificar cuenta".

### Figura 254


*Contenido del correo de verificación de cuenta*



Hola **Usuario**,

Gracias por registrarte en FileSoft. Para poder acceder a tu cuenta, necesitamos que verifiques tu correo electrónico.

Para ello, haz clic en el siguiente botón:



Verificar cuenta

O si no puedes acceder al enlace, copia y pega la siguiente dirección en tu navegador:

*Fuente.* Capturado con el software Scribehow.

Si todo salió bien se te informará que se confirmó el correo y puedes ir a la página de inicio de sesión dando clic en el botón "Iniciar sesión".

**Figura 255**

*Confirmación de verificación exitosa*



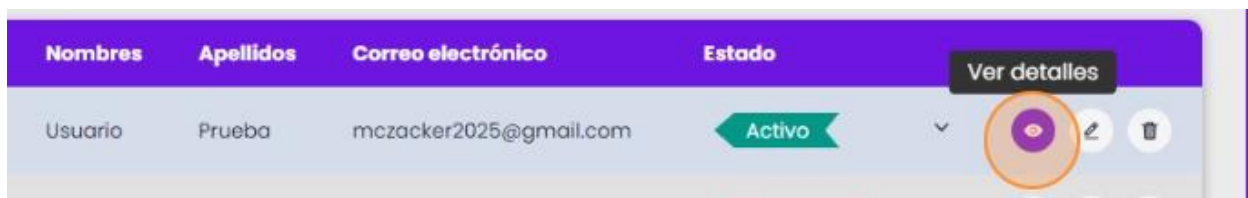
*Fuente.* Capturado con el software Scribehow.

***Ver información de funcionario***

Navega al listado de radicados en la ruta <https://filessoft.com/dashboard/funcionarios>, da clic sobre el icono "Ver detalles" del registro que quieres visualizar y observa la información completa del funcionario, sus datos, la dependencia donde está asignado y los permisos que tiene asociados.

**Figura 256**

*Acción de ver los detalles de un funcionario*



*Fuente.* Capturado con el software Scribehow.

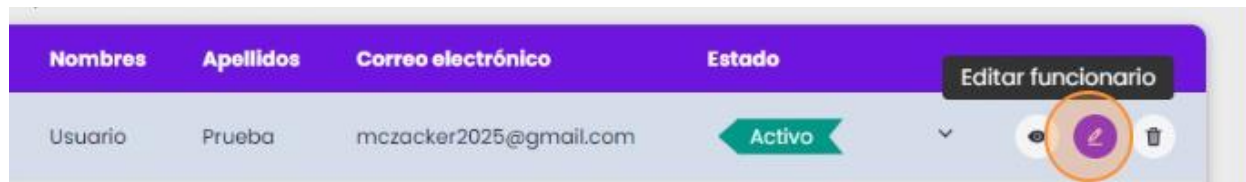
### **Actualizar funcionario**

Navega al listado de funcionarios en la dirección

<https://filesoft.com/dashboard/funcionarios> y da clic en el icono "Editar funcionario".

### **Figura 257**

*Acción de editar la información de un funcionario*



*Fuente.* Capturado con el software Scribehow.

En el formulario actualiza la información que sea necesaria, como su número de identificación, sus nombres, su correo, los permisos que tendrá asignados y cuando estés seguro de la información a actualizar, da clic en el botón "Actualizar funcionario".

### **Figura 258**

*Confirmación de actualización de los datos del funcionario*



*Fuente.* Capturado con el software Scribehow.

### ***Eliminar funcionario***

Navega al listado de funcionarios en la ruta <https://filesoft.com/dashboard/funcionarios>, elije el funcionario que deseas eliminar, da clic en el icono "Eliminar funcionario" y si estás seguro de eliminar el funcionario da clic en el botón "Confirmar" de la ventana emergente.

### **Figura 259**

*Acción de eliminar un funcionario*



*Fuente.* Capturado con el software Scribehow.

### **Gestiona los ciudadanos**

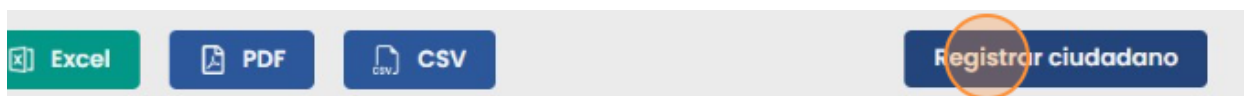
Esta guía proporciona instrucciones paso a paso sobre cómo registrar un ciudadano en la aplicación, para esto se requiere su nombre completo, correo electrónico y número de teléfono, además la visualización, edición y eliminación de los ciudadanos.

### ***Registrar ciudadano***

Navega al listado de ciudadanos en la dirección <https://filesoft.com/dashboard/ciudadanos> y da clic en el botón "Registrar ciudadano".

### **Figura 260**

*Acción de registrar un ciudadano*



*Fuente.* Capturado con el software Scribehow.

Da clic en el campo de texto "Nombre completo del ciudadano" y Escribe el nombre del ciudadano, por ejemplo "Ciudadano Prueba", luego da clic en el campo de "Número de identificación" y registra la identificación del ciudadano, por ejemplo "10101010101", luego da clic en el campo de texto "Correo electrónico Opcional" y digita el correo electrónico del ciudadano, por ejemplo [ciudadanoprueba@gmail.com](mailto:ciudadanoprueba@gmail.com), recuerda que el campo es opcional pero recomendado ya que se requiere para enviar los correos de notificación, por último finaliza la digitación dando clic en el campo de texto "Número de teléfono Opcional" y digitando el número de teléfono del ciudadano, por ejemplo "3115467898", recuerda que este campo es opcional.

### Figura 261

*Formulario de registro de ciudadano*

ciudadano

Nombre completo del ciudadano

Número de identificación

10101010101

Correo electrónico Opcional

ciudadanoprueba@gmail.com

Número de teléfono Opcional

Ej. 1234567890

17

*Fuente.* Capturado con el software Scribehow.

Si estás seguro de la información a registrar del ciudadano, da clic en el botón "Registrar ciudadano" para confirmar el registro.

## Figura 262

*Guardar los datos del ciudadano*



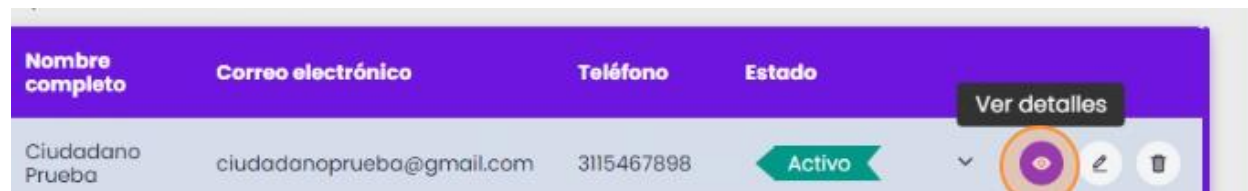
*Fuente.* Capturado con el software Scribehow.

## *Ver información de ciudadano*

Navega al listado de ciudadanos en la ruta <https://filessoft.com/dashboard/ciudadanos>, identifica el ciudadano a visualizar, da clic sobre el icono "Ver detalles" y visualiza la información del ciudadano de manera detallada.

## Figura 263

*Acción de ver los detalles de un ciudadano*



*Fuente.* Capturado con el software Scribehow.

## Actualizar ciudadano

Navega al listado de ciudadanos en la ruta <https://filessoft.com/dashboard/ciudadanos> y da clic en el icono "Editar funcionario" en el momento que lo hayas identificado.

## Figura 264

*Acción de editar un ciudadano*



*Fuente.* Capturado con el software Scribehow.

Ingresa los datos a actualizar del ciudadano, como su nombre, identificación, correo, teléfono, dirección y estado, cuando estés seguro de la información a actualizar da clic en el botón "Actualizar ciudadano".

## Figura 265

*Guardar la edición del ciudadano*



*Fuente.* Capturado con el software Scribehow.

## *Eliminar ciudadano*

Navega al listado de ciudadanos en la ruta <https://filessoft.com/dashboard/ciudadanos>, identifica el ciudadano que deseas eliminar y da clic sobre el icono "Eliminar ciudadano", si estás seguro de eliminar el registro da clic en el botón "Confirmar" de la ventana emergente.



**Figura 266***Acción de eliminar un ciudadano*

Nombre completo	Correo electrónico	Teléfono	Estado	
Ciudadano Prueba	ciudadanoprueba@gmail.com	3115467898	Activo	<div style="border: 1px solid black; padding: 2px;">Eliminar ciudadano</div>    
Zacker Ciudadano	mczacker2001@gmail.com	No registrado	Activo	   

*Fuente.* Capturado con el software Scribehow.

## Gestiona las entidades

Esta guía proporciona instrucciones paso a paso sobre cómo registrar una entidad en la aplicación, además de visualizar su información, realizar actualizaciones y eliminar los registros.

### *Registrar entidad*

Navega al listado de entidades en la ruta <https://filessoft.com/dashboard/entidades> y da clic en el botón "Registrar entidad".

**Figura 267***Acción de registrar entidad*

*Fuente.* Capturado con el software Scribehow.

Da clic sobre el campo de texto "Nombre de la entidad" y escribe el nombre de la entidad, por ejemplo "Alcaldía de Gachalá", ten en cuenta que los campos de "Correo

electrónico" y "Número de teléfono" son opcionales, para finalizar da clic sobre el botón "Registrar entidad" para guardar el registro.

### Figura 268

*Confirmar el registro de la entidad*



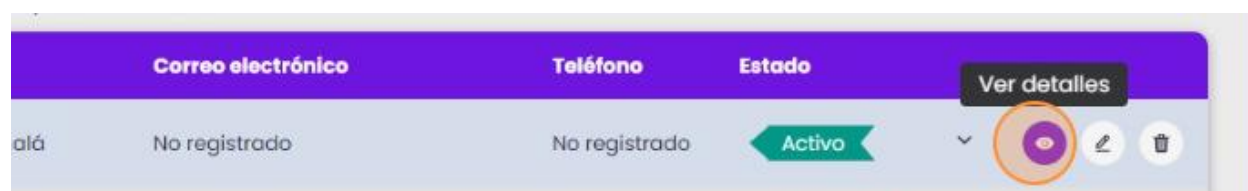
*Fuente.* Capturado con el software Scribehow.

Ver detalles de entidad

Navega al listado de entidades en la dirección <https://filessoft.com/dashboard/entidades> y da clic en el icono "Ver detalles" una vez hayas identificado la entidad.

### Figura 269

*Acción de ver los detalles de una entidad*



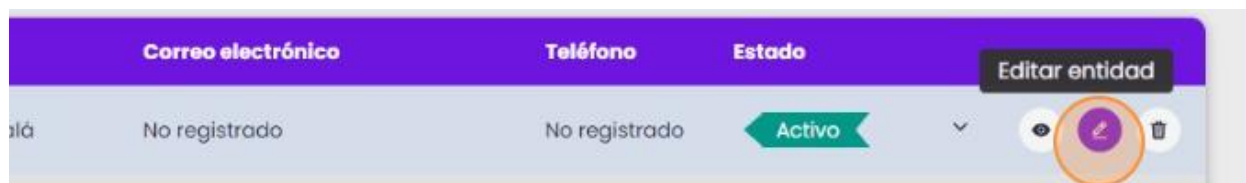
*Fuente.* Capturado con el software Scribehow.

**Actualizar entidad**

Navega al listado de entidades en la dirección <https://filessoft.com/dashboard/entidades>, identifica el registro que deseas actualizar y da clic sobre el icono "Editar entidad".

**Figura 270**

*Acción de editar una entidad*

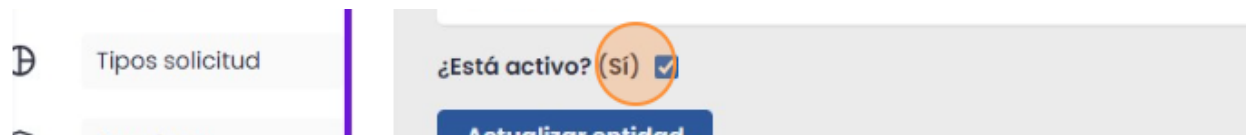


*Fuente.* Capturado con el software Scribehow.

Da clic sobre los campos que deseas actualizar y digita la nueva información, puedes dar clic sobre el campo "¿Está activo? (Sí)" para cambiar el estado de la entidad.

**Figura 271**

*Cambiar estado de entidad*



*Fuente.* Capturado con el software Scribehow.

Si estás seguro de los cambios realizados da clic sobre el botón "Actualizar entidad" para almacenar los cambios.

**Figura 272**

*Confirmar la edición en los cambios de la entidad*



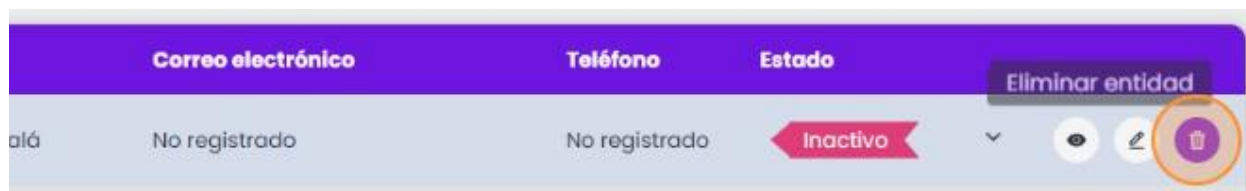
*Fuente.* Capturado con el software Scribehow.

### ***Eliminar entidad***

Navega al listado de entidades en la dirección <https://filessoft.com/dashboard/entidades>, identifica la entidad a eliminar y da clic sobre el icono "Eliminar entidad", da clic sobre el botón "Confirmar" si estás seguro de eliminar el registro.

### **Figura 273**

#### *Acción de eliminar una entidad*



*Fuente.* Capturado con el software Scribehow.

### **Gestiona las entidades**

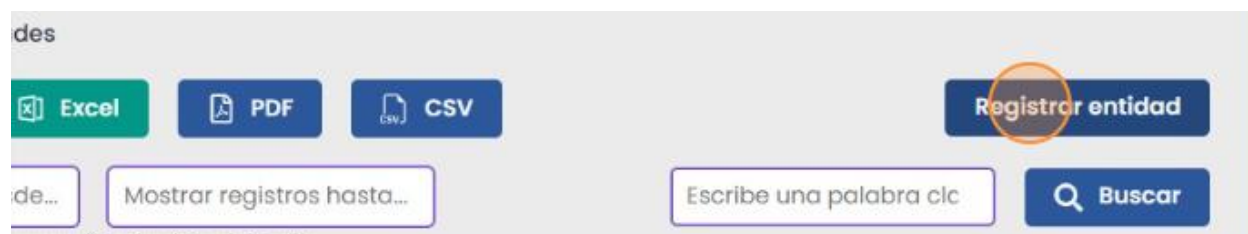
Esta guía proporciona instrucciones paso a paso sobre cómo registrar una entidad en la aplicación, además de visualizar su información, actualizarla o eliminarla de acuerdo con lo que sea requerido.

### ***Registrar entidad***

Navega al listado de entidades en la ruta <https://filessoft.com/dashboard/entidades> y da clic en el botón "Registrar entidad".

## Figura 274

### *Acción de insertar una entidad*



*Fuente.* Capturado con el software Scribehow.

Da clic sobre el campo de texto "Nombre de la entidad" y Escribe el nombre de la entidad, por ejemplo "Alcaldía de Gachalá", ten en cuenta que los campos de "Correo electrónico" y "Número de teléfono" son opcionales, cuando estés seguro da clic sobre el botón "Registrar entidad" para guardar el registro.

## Figura 275

### *Registro de entidad en la aplicación*



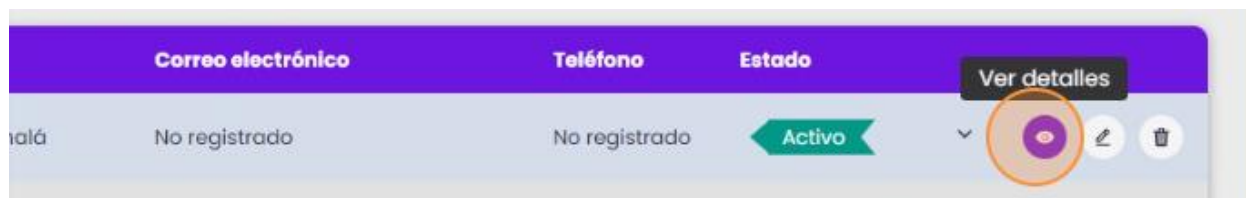
*Fuente.* Capturado con el software Scribehow.

### *Ver detalles de entidad*

Navega al listado de entidades en la ruta <https://filessoft.com/dashboard/entidades> y da clic en el icono "Ver detalles" una vez hayas identificado la entidad, en esta página a la que te dirige podrás ver más detalles de la entidad, como su nombre, su correo electrónico, su número de teléfono y su estado.

**Figura 276**

*Acción de ver los detalles de la entidad*



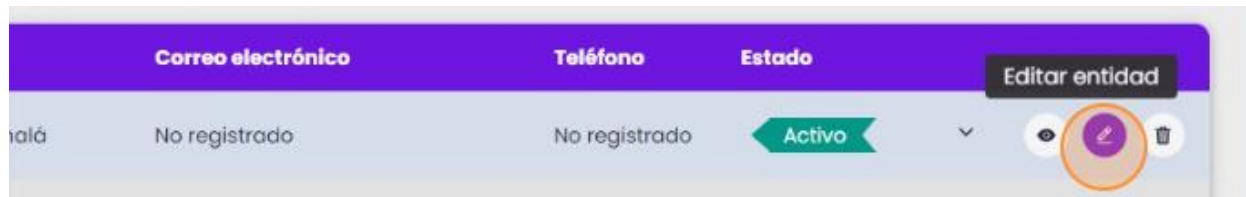
*Fuente.* Capturado con el software Scribehow.

Actualizar entidad

Navega al listado de entidades en la ruta <https://filesoft.com/dashboard/entidades>, identifica el registro que deseas actualizar y da clic sobre el icono "Editar entidad".

**Figura 277**

*Acción de actualizar la información de una entidad*

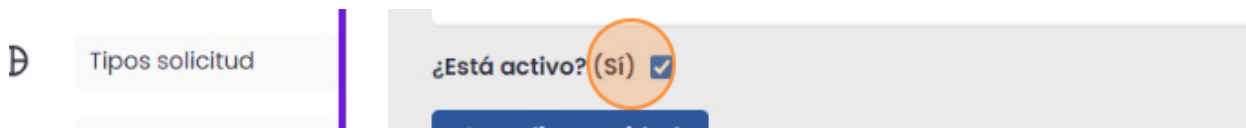


*Fuente.* Capturado con el software Scribehow.

Da clic sobre los campos que deseas actualizar y digita la nueva información, si lo deseas puedes dar clic sobre el campo "¿Está activo? (Sí)" para cambiar el estado de la entidad.

**Figura 278**

*Cambiar estado de una entidad*



*Fuente.* Capturado con el software Scribehow.

Si estás seguro de los cambios realizados da clic sobre el botón "Actualizar entidad" para almacenar los cambios.

### Figura 279

*Guardar los cambios de edición de información de la entidad*



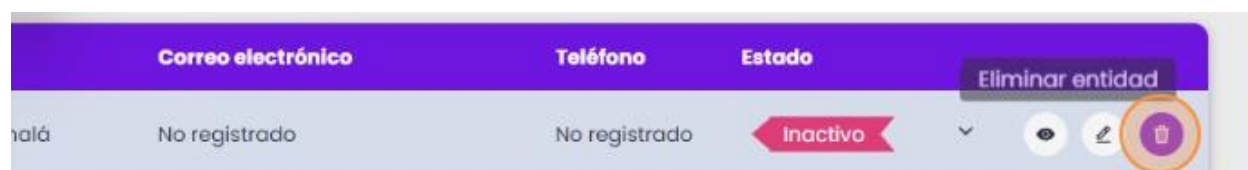
*Fuente.* Capturado con el software Scribehow.

### *Eliminar entidad*

Navega al listado de entidades en la dirección <https://filessoft.com/dashboard/entidades>, identifica la entidad a eliminar, da clic sobre el icono "Eliminar entidad" y da clic sobre el botón "Confirmar" si estás seguro de eliminar el registro.

### Figura 280

*Acción de eliminar una entidad*



*Fuente.* Capturado con el software Scribehow.

### **Gestiona las dependencias**

Esta guía proporciona instrucciones paso a paso sobre cómo registrar una dependencia en el sistema, visualizarlas, editar su información y eliminarlas.

### ***Registrar dependencia***

Navega al listado de dependencias en la ruta <https://filessoft.com/dashboard/dependencias> y da clic en el botón "Registrar dependencia".

### **Figura 281**

*Acción de registrar una dependencia*



*Fuente.* Capturado con el software Scribehow.

### ***Ver información de dependencia***

Navega al listado de dependencias en la ruta <https://filessoft.com/dashboard/dependencias>, identifica la dependencia, da clic en el icono "Ver detalles" y visualiza la información de la dependencia seleccionada.

### **Figura 282**

*Acción de ver detalles de una dependencia*

Descripción	Estado	Fecha registro	Fecha última actualización
Sin descripción	Inactivo	09 de septiembre de 2023, 11:40:54 PM	09 de septiembre de 2023, 11:40:54 PM

Ver detalles

*Fuente.* Capturado con el software Scribehow.



### Actualizar dependencia

Navega al listado de dependencias en la ruta <https://filessoft.com/dashboard/dependencias>, identifica el registro y da clic en el icono "Editar dependencia".

### Figura 283

#### Acción de editar dependencia

Descripción	Estado	Fecha registro	Fecha última actualización	
Sin descripción	Inactivo	09 de septiembre de 2023, 11:40:54 PM	09 de septiembre de 2023, 11:40:54 PM	Editar dependencia 

Fuente. Capturado con el software Scribehow.

Cambia la información de la dependencia de acuerdo con lo que se requiera, por ejemplo, un cambio en el estado de la dependencia y si estás seguro de los cambios realizados da clic en el botón "Actualizar dependencia" para guardar los cambios.

### Figura 284

#### Guardar cambios de la dependencia

Copia seguridad

Yeferson Zaque  
zoquecamlio@gmail.c...

¿Está activa? (Sí)

Actualizar dependencia

Fuente. Capturado con el software Scribehow.

### Eliminar dependencia

Navega al listado de dependencias en la ruta <https://filessoft.com/dashboard/dependencias>, identifica el registro y da clic sobre el icono "Eliminar dependencia", en la ventana emergente da clic en el botón "Confirmar" si estás seguro de la eliminación.

**Figura 285***Acción de eliminar una dependencia*

Asignación	Estado	Fecha registro	Fecha última actualización	
Asignación	Activo	09 de septiembre de 2023, 11:40:54 PM	09 de septiembre de 2023, 11:48:43 PM	Eliminar dependencia [Ocultar] [Editar] [Eliminar]

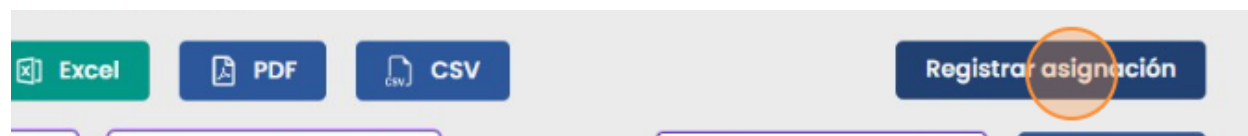
*Fuente.* Capturado con el software Scribehow.

## Gestiona la asignación de dependencias

Esta guía proporciona instrucciones paso a paso sobre cómo generar una asignación de dependencia para un funcionario, ver las que se han generado, editar una asignación o eliminarla.

### *Generar asignación*

Navega al listado de asignaciones en la ruta <https://filessoft.com/dashboard/asignacion-dependencias> y da clic en el botón "Registrar asignación".

**Figura 286***Acción de generar asignación de dependencia*

*Fuente.* Capturado con el software Scribehow.

Da clic en el campo "Usuario a asignar" y selecciona a un funcionario registrado al cual se le realizará la asignación.

### Figura 287

*Selección del funcionario al que se le asignará la dependencia*

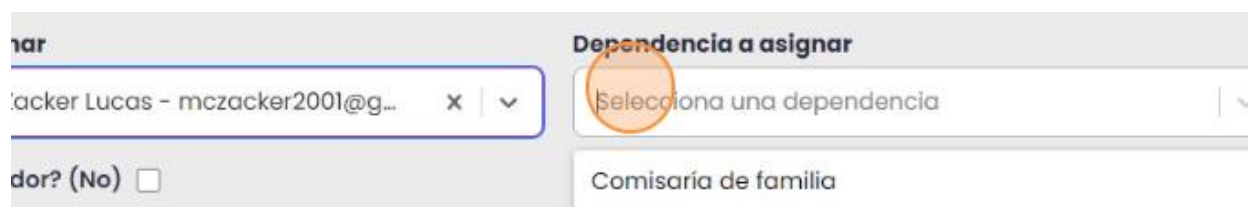


*Fuente.* Capturado con el software Scribehow.

Da clic en el campo "Dependencia a asignar" y selecciona una dependencia a donde se desea asignar, puedes establecer si será el administrador de esta.

### Figura 288

*Selección de la dependencia a asignar*



*Fuente.* Capturado con el software Scribehow.

Cuando estés seguro da clic en el botón "Generar asignación" para guardar la asignación, puede que el funcionario ya tenga asignación, lo cual se informará a través de una alerta.

### Figura 289

*Guardar la asignación de dependencia*



*Fuente.* Capturado con el software Scribehow.

### *Ver asignación*

Navega al listado de asignaciones en la ruta <https://filesoft.com/dashboard/asignacion-dependencias>, da clic en el icono "Ver detalles" cuando estés seguro del registro a visualizar.

### **Figura 290**

*Acción de visualizar una asignación de dependencia*



*Fuente.* Capturado con el software Scribehow.

### *Actualizar asignación*

Navega al listado de asignaciones en la ruta <https://filesoft.com/dashboard/asignacion-dependencias> y cuando identifiques el registro da clic sobre el icono "Editar asignación dependencia".

### **Figura 291**

*Acción de actualizar asignación*



*Fuente.* Capturado con el software Scribehow.

Puedes dar clic sobre el campo "Dependencia a asignar" y seleccionar la dependencia que desees, al igual que puedes establecer si será administrador en la dependencia, al estar seguro de los cambios puedes dar clic sobre el botón "Actualizar asignación".

**Figura 292**

*Guardar edición en la asignación de dependencia*

*Fuente.* Capturado con el software Scribehow.

## Gestiona las copias de seguridad

Esta guía proporciona instrucciones paso a paso sobre cómo crear una copia de seguridad de una base de datos desde el panel de administración de la aplicación, además de respaldar los archivos y restaurar los datos cuando sea necesario.

### *Copia de seguridad a base de datos*

Navega a la sección de copias de seguridad en la ruta

<https://filesoft.com/dashboard/copia-seguridad> y da clic sobre el botón "Generar copia de seguridad".

**Figura 293**

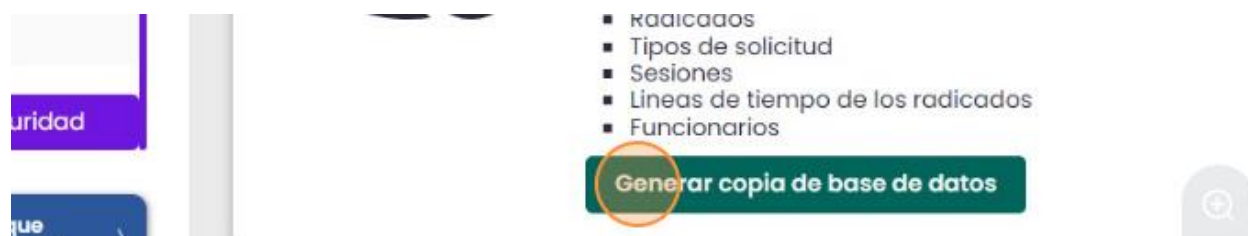
*Ir al módulo de generación de copias de seguridad*

*Fuente.* Capturado con el software Scribehow.

Lee la información de lo que incluirá la copia de seguridad y da clic sobre el botón "Generar copia de base de datos".

### Figura 294

*Acción de generación de copia de seguridad de base de datos*



*Fuente.* Capturado con el software Scribehow.

Si estás seguro, da clic en el botón "Generar copia de seguridad de la base de datos" del modal desplegado y almacena la copia de seguridad en un lugar seguro sin realizar modificaciones en ella y da clic en el botón "Cerrar ventana" para finalizar.

### Figura 295

*Confirmación de generación de copia de seguridad de base de datos*



*Fuente.* Capturado con el software Scribehow.

### *Copia de seguridad a archivos*

Navega a la sección de copias de seguridad en la ruta

<https://filesoft.com/dashboard/copia-seguridad> y da clic sobre el botón "Generar copia de

seguridad", Lee con atención los detalles de la copia de seguridad de archivos y lo que almacenará, luego da clic sobre el botón "Generar copia de archivos".

### Figura 296

*Acción de generar copia de seguridad de archivos*



*Fuente.* Capturado con el software Scribehow.

Si estás seguro, da clic en el botón "Generar copia de seguridad de los archivos" para comenzar el proceso y Almacena el archivo de copia de seguridad en un lugar seguro y da clic en el botón "Cerrar ventana" para finalizar.

### Figura 297

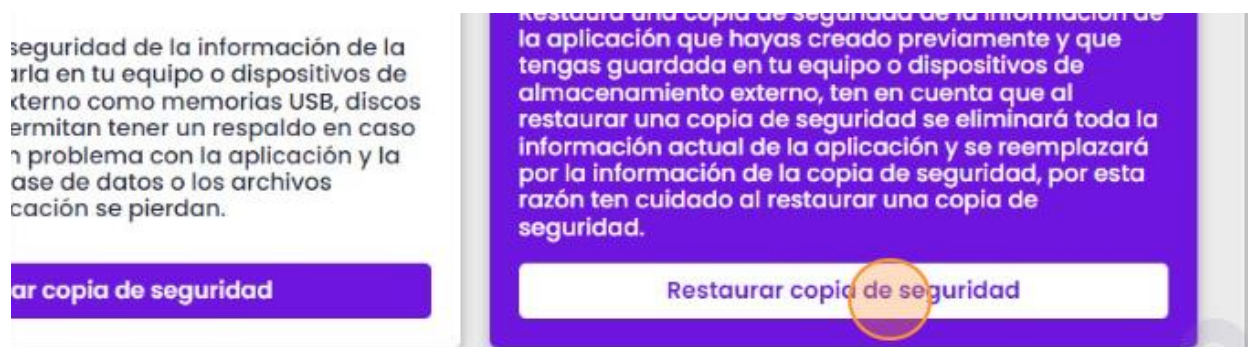
*Iniciar proceso de copia de seguridad de archivos*



*Fuente.* Capturado con el software Scribehow.

### **Restablecer base de datos**

Navega a la sección de copias de seguridad en la ruta <https://filesoft.com/dashboard/copia-seguridad> y da clic en el botón "Restaurar copia de seguridad".

**Figura 298***Módulo de restauración de copias de seguridad*

*Fuente.* Capturado con el software Scribehow.

Lee la información de lo que sucederá al realizar la restauración y da clic en el botón "Restaurar copia de base de datos".

**Figura 299***Acción de restaurar base de datos*

*Fuente.* Capturado con el software Scribehow.

Da clic en el campo "Selecciona o arrastra el archivo a cargar" y selecciona el archivo de copia de base de datos, da clic sobre el botón "Restaurar copia de seguridad de la base de datos" para comenzar el proceso.



**Figura 300**

*Carga del archivo de copia de base de datos*



*Fuente.* Capturado con el software Scribehow.

**Restablecer archivos**

Navega a la sección de copias de seguridad en la ruta <https://filesoft.com/dashboard/copia-seguridad>, da clic en el botón "Restaurar copia de seguridad" y lee lo que sucederá al restablecer la copia de los archivos, si estás seguro da clic en el botón "Restaurar copia de archivos".

**Figura 301**

*Acción de restaurar los archivos de la aplicación*



*Fuente.* Capturado con el software Scribehow.

Da clic sobre el campo "Archivo de copia de seguridad" y selecciona el archivo de copia de seguridad de archivos, di estás seguro da clic en el botón "Restaurar copia de seguridad de los archivos" para comenzar el proceso.

### Figura 302

*Iniciar restauración de archivos*



*Fuente.* Capturado con el software Scribehow.

Este proceso puede llevar varios segundos e incluso minutos, pero cuando finalice informará que todo salió correcto y puedes dar clic en el botón "Cerrar ventana".

### Figura 303

*Cerrar ventana de restauración de archivos*



*Fuente.* Capturado con el software Scribehow.

## Conclusiones

Se realiza la implementación de la diagramación UML y el modelado de base de datos, logrando una comprensión profunda y completa de las variables que influyen en el problema que se abordan en este proyecto, proporcionando una visión integral de la interacción y el comportamiento que se requiere en el software de correspondencia, por otro lado, al modelar la base de datos, se define la estructura de la información que se almacena buscando ser eficientes y organizados.

Durante el proceso de investigación, se logran identificar los servicios clave que forman parte de la aplicación web, lo que proporciona una visión completa de su arquitectura y se reconocen los diversos componentes involucrados en la transferencia de información, comprendiendo la manera en que se establece la comunicación; estableciendo una base sólida y esencial para el diseño y desarrollo exitoso de la aplicación de correspondencia.

El diseño de la aplicación integra los servicios REST con el servicio de almacenamiento de base de datos, creando una arquitectura que garantiza una comunicación eficiente y segura entre el usuario y el sistema; el servicio REST permite una interacción fluida mientras que la base de datos proporciona un almacenamiento robusto y escalable para la información, este se diseña cuidadosamente para asegurar que el software funcione de manera óptima y cumpla con los estándares de calidad ante los usuarios.

## Recomendaciones

Después de analizar detenidamente el proceso de correspondencia en la Alcaldía Municipal de Gachetá y comprender la organización existente en la recepción, organización, asignación, seguimiento y respuesta a los radicados, se recomienda la implementación de un software de correspondencia, debido a que es fundamental para abordar los problemas de organización y comunicación, permitiendo centralizar y automatizar la gestión de solicitudes, quejas y reclamos de los ciudadanos o entidades, mejorando la eficiencia en la recepción de documentos, la asignación precisa a las dependencias correspondientes, el seguimiento en tiempo real del estado de los radicados y la entrega de respuestas ágiles a los ciudadanos.

Dada la importancia crítica de preservar la integridad de la información de los radicados y los problemas identificados en el almacenamiento y la falta de copias de seguridad, se aconseja implementar un plan de copias de seguridad que incluya la programación regular de copias de seguridad automáticas de la base de datos y los archivos adjuntos relacionados con los radicados, además, se recomienda almacenar estas copias en ubicaciones seguras, dentro o fuera de las instalaciones para garantizar la recuperación de datos en caso de fallos técnicos.

Para garantizar una transición exitosa, se recomienda realizar una capacitación exhaustiva al personal involucrado en el proceso, incluyendo los funcionarios de la oficina de correspondencia, los encargados de las dependencias y cualquier otro funcionario que participe en la gestión de solicitudes; esta capacitación debe enfocarse en el uso efectivo del nuevo software, la comprensión de los procedimientos actualizados y la importancia de la gestión de copias de seguridad, para esto se deben establecer procedimientos de monitoreo y control para garantizar la adopción adecuada de las nuevas prácticas y tecnologías.

## Bibliografía

*Acuerdo 60 de 2001 Archivo General de la Nación - Gestor Normativo.* (2015, 1 diciembre).

Función Pública.

<https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=10551>

Amazon Web Services. (2020). *Contenedores de Docker | ¿Qué es Docker? | AWS.* Amazon Web Services, Inc. <https://aws.amazon.com/es/docker/>

Aplyca. (2022, 14 septiembre). *NextJS: ¿el futuro de la web? ¿por qué y cuándo usarlo? -*

*Aplyca.* Aplyca Tecnología SAS. <https://www.aplyca.com/blog/nextjs-el-futuro-web-que-es-nextjs>

Ascencio, A. P. D., & Pinilla, P. C. L. (2017). *Sistema de información para la ventanilla única de la alcaldía municipal de Ubaté* [Tesis de grado]. Universidad de Cundinamarca.

*Decreto 1078 de 2015 Sector de Tecnologías de la Información y las Comunicaciones - Gestor Normativo.* (2023, 19 enero). Función Pública.

<https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=77888>

GCFGGlobal. (2019). *Informática Básica: ¿Qué son las aplicaciones web?* GCFGGlobal.org.

<https://edu.gcfglobal.org/es/informatica-basica/que-son-las-aplicaciones-web/1/>

Gomez, A. G. C. (2015). *Análisis de un prototipo de software para la recepción, entrega y envío de correspondencia en la alcaldía de Taraira - Vaupés* [Tesis de ingeniería]. Corporación Universitaria Minuto de Dios - UNIMINUTO.

Henry, R. (2021, 1 noviembre). *¿Qué es Node.js y para qué se utiliza? | Henry Blog.* Henry.

[https://blog.soyhenry.com/que-es-node-js-y-para-que-se-utiliza/amp/?utm\\_source=google](https://blog.soyhenry.com/que-es-node-js-y-para-que-se-utiliza/amp/?utm_source=google)

KeepCoding, R. (2022a, agosto 25). *7 tecnologías para desarrollo web.* KeepCoding Tech School. <https://keepcoding.io/blog/7-tecnologias-para-desarrollo-web/>

- KeepCoding, R. (2022b, noviembre 28). *Cabecera authorization bearer*. KeepCoding Tech School. <https://keepcoding.io/blog/cabecera-authorization-bearer/>
- Ley 29 de 1990 - Gestor Normativo*. (2015, 1 diciembre). Función Pública. <https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=254>
- Ley 962 de 2005 - Gestor Normativo*. (2022, 19 noviembre). Función Pública. <https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=17004>
- Ley 1341 de 2009 - Gestor Normativo*. (2022, 25 mayo). Función Pública. <https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=36913>
- Ley 1581 de 2012 - Gestor Normativo*. (2022, 2 marzo). Función Pública. <https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=49981>
- Ley 1755 de 2015 - Gestor Normativo*. (2022, 15 marzo). Función Pública. <https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=65334>
- Marín, R. (2022, 26 septiembre). *Los gestores de bases de datos más usados en la actualidad*. Canal Informática y TICS. <https://www.inesem.es/revistadigital/informatica-y-tics/los-gestores-de-bases-de-datos-mas-usados/>
- Normalización de tiempos de respuesta a derechos de petición en Colombia | Insights | Holland & Knight*. (s. f.). <https://www.hklaw.com/en/insights/publications/2022/05/normalizacion-de-tiempos-de-respuesta-a-derechos-de-peticion>
- Quiceno, J. C. Q. (2019). *Software de gestión documental y ventanilla única para administraciones municipales de sexta categoría* [Tesis de grado]. Universidad Autónoma de Manizales.

RedHat. (2020). *¿Qué es una API de REST?* <https://www.redhat.com/es/topics/api/what-is-a-rest-api>

Romero, R. R. A. R., & Valderrama, V. P. M. (2018). *Sistema de información para el control de correspondencia y almacén en la alcaldía de Pasca* [Tesis de grado]. Universidad de Cundinamarca.

Schiaffarino, A. (2019, 7 agosto). *Modelo cliente servidor*. Infranetworking.  
<https://blog.infranetworking.com/modelo-cliente-servidor/>

Solutions, D. S. (s. f.). *Docmanager - Software para gestión de Correspondencia*.  
[https://docmanager.com.co/ES/Productos/Software\\_Control\\_Correspondencia](https://docmanager.com.co/ES/Productos/Software_Control_Correspondencia)

Vallejo, G. V. (2019, febrero). *Planta Física Alcaldía de Gachetá*. Viajar en Verano.  
<https://www.viajarenverano.com/gacheta-cundinamarca/>

## Apéndices

### Apéndice A

#### *Decretos, normas y leyes*

- Ley 1341 de 2009 Ley de Tecnologías de la Información y las Comunicaciones

Por la cual se establece el marco regulatorio para el desarrollo de las Tecnologías de la Información y las Comunicaciones en el país.

Esta Ley se enfoca en fomentar la competencia en el mercado de las TIC, garantizar la protección de los derechos de los usuarios de los servicios de tecnología, promover la inversión en infraestructura en áreas rurales y de bajos ingresos para reducir la brecha digital, y fomentar la innovación y el desarrollo tecnológico en el sector de las TIC.

- Ley 29 de 1990 Fomento de la investigación científica y el desarrollo tecnológico

Por la cual se dictan disposiciones para el fomento de la investigación científica y el desarrollo tecnológico y se otorgan facultades extraordinarias.

Esta Ley fomenta y se orienta el adelanto científico y tecnológico en Colombia, e incorpora la ciencia y la tecnología a los planes y programas de desarrollo económico - social del país. El Estado tiene la obligación de crear condiciones favorables para la generación de conocimiento científico y tecnología nacionales, estimular la capacidad innovadora del sector productivo, organizar un sistema nacional de información científica y tecnológica, y dar incentivos a la creatividad.

Además, ampara la investigación científica, lo cual es necesario para el desarrollo tecnológico en nuestro país.

- Ley 1581 de 2012 Protección de datos personales

Por la cual se dictan las disposiciones generales para la protección de datos personales.



Establece los principios que deben guiar el tratamiento de la información personal, los derechos de los titulares de la información, las obligaciones de los responsables del tratamiento de los datos, las medidas de seguridad que se deben aplicar y las sanciones por incumplimiento.

Esta Ley garantiza el derecho a la privacidad de las personas y la protección de sus datos personales, siendo regulada por diferentes entidades dependiendo del contexto donde sean utilizados los datos de la persona.

- Ley 962 de 2005 Racionalización de trámites y procedimientos administrativos

Por la cual se dictan disposiciones sobre racionalización de trámites y procedimientos administrativos de los organismos y entidades del Estado y de los particulares que ejercen funciones públicas o prestan servicios públicos.

La ley establece la reserva legal de permisos, licencias y requisitos, el procedimiento para establecer los trámites autorizados por la ley, la información y publicidad, y el fortalecimiento tecnológico de la Administración Pública. Además, establece que las Asambleas Departamentales y los Concejos Distritales o Municipales solo pueden adoptar medidas para la implementación o aplicación de los trámites creados o autorizados por la ley.

- Decreto 1078 de 2015 Estructura y reglamentación del sector de tecnología

Por medio del cual se expide el Decreto Único Reglamentario del Sector de Tecnologías de la Información y las Comunicaciones.

Establece la estructura del sector de tecnologías de la información y las comunicaciones representando cuales son cada una de las entidades pertenecientes al sector de tecnología y su nivel de importancia dentro del mismo, además establece el régimen reglamentario del sector de tecnologías de la información y las comunicaciones determinando las normativas vigentes que se deben acoger por las entidades, empresas o personas para la gestión de tecnologías.

- Ley 1755 de 2015 Derecho fundamental de petición

Por medio de la cual se regula el Derecho Fundamental de Petición y se sustituye un título del Código de Procedimiento Administrativo y de lo Contencioso Administrativo.

La Ley establece el derecho que tiene toda persona para presentar peticiones de manera respetuosa a las autoridades o entidades gubernamentales por motivos de interés general o particular, determinando que se debe dar una pronta resolución de manera completa a las solicitudes del solicitante; estableciendo así los términos y normas que se deben tener en cuenta por parte del solicitante al igual que las que debe tener en cuenta la entidad o entidad gubernamental.