



**UNIVERSIDAD PRIVADA DE TACNA**

**FACULTAD DE INGENIERIA**

**Escuela Profesional de Ingeniería de Sistemas**

**Proyecto *“Implementación de un Aplicativo Web  
para la Gestión de Acervos Digitales en el  
Gobierno Regional de Tacna”***

*Curso: Construcción de Software II*

*Docente: Mag. Ricardo Eduardo Valcárcel Alvarado*

Integrantes:

***Arenas Paz Soldan, Miguel Jesus***

***(2017059282)***

***Christian Alexander, Céspedes Medina***

***(2010036257)***

**Tacna – Perú**

**2025**

***Sistema Implementación de un Aplicativo web para la  
Gestión de Acervos Digitales en el Gobierno Regional  
de Tacna***  
**Documento de Arquitectura de Software**

**Versión {1.4}**

CONTROL DE VERSIONES					
Versión	Hecha por	Revisada por	Aprobada por	Fecha	Motivo
1.0	MAPS	CACM	CACM	22/09/2024	Versión original
1.2	MAPS	CACM	CACM	17/09/2024	Avance del documento
1.3	MAPS	CACM	CACM	24/10/2024	Corrección de observaciones
1.4	MAPS	CACM	CACM	28/03/2025	Corrección de diagramas de secuencia de Gestionar Persona, Cargos, Usuarios y Áreas

## INDICE GENERAL

1.	Introducción .....	6
1.1	Propósito .....	6
1.2	Alcance .....	6
1.3	Definición, siglas y abreviaturas .....	6
1.4	Referencias .....	6
1.5	Visión General .....	6
2.	Representación Arquitectónica .....	7
2.1	Escenarios .....	7
2.2	Vista lógica .....	9
2.3	Vista del Proceso .....	10
2.4	Vista del Desarrollo .....	11
2.5	Vista Física .....	12
3.	Objetivos y limitaciones arquitectónicas .....	13
3.1	Disponibilidad .....	13
3.2	Seguridad .....	13
3.3	Adaptabilidad .....	13
3.4	Rendimiento .....	13
4.	Análisis de Requerimientos .....	14
4.1	Requerimientos Funcionales .....	14
4.2	Requerimientos No Funcionales .....	16
5.	Vistas de Caso de Uso .....	17
6.	Vista Lógica .....	19
6.1	Diagramas de Secuencia .....	19
6.2	Diagrama de Clases .....	26
6.3	Diagrama Contextual .....	27
7.	Vista de Procesos .....	28
7.1	Diagrama de Proceso Actual .....	28
7.2	Diagrama de Proceso Propuesto .....	28
8.	Vista de Despliegue .....	29
8.1	Diagrama de Contenedor .....	29
9.	Vista de Implementación .....	30
9.1	Diagrama de Paquetes .....	30
9.2.	Diagrama de Componentes .....	31
10.	Vista de Datos .....	32
10.1.	Diagrama Entidad Relación .....	32



11.	Calidad .....	33
11.1.	Escenario de Seguridad.....	33
11.2.	Escenario de Usabilidad.....	33
11.3.	Escenario de Adaptabilidad .....	34
11.4.	Escenario de Disponibilidad .....	35
11.5.	Otro Escenario.....	36

## 1. Introducción

### 1.1 Propósito

Definir los detalles de la arquitectura que será utilizada para desarrollar el aplicativo. Para realizar esto, será necesario especificar cómo son la vista lógica, física, de proceso y de desarrollo. Mencionar las características de la arquitectura elegida. Relacionar los requerimientos funcionales y no funcionales con la implementación de la arquitectura del sistema.

### 1.2 Alcance

- Este documento abarca el desarrollo e implementación del Aplicativo Web para la Gestión de Acervos Digitales para el Gobierno Regional de Tacna, que se centra específicamente en los procesos sobre el área de reserva.

### 1.3 Definición, siglas y abreviaturas

- **SAD:** Documento de Arquitectura de Software.
- **MVC:** Modelo Vista Controlador.

### 1.4 Referencias

RUP03: IBM Rational Unified Process. <http://www.rational.com/rup,2003>.

### 1.5 Visión General

El proyecto tiene como objetivo desarrollar un aplicativo web integral que optimice la gestión de acervos digitales en el Gobierno Regional de Tacna. Este sistema permitirá la administración eficiente de documentos y archivos digitales, garantizando su seguridad, accesibilidad y trazabilidad. Con funcionalidades como la gestión de usuarios, roles, permisos, firma electrónica, y la automatización de eventos, el aplicativo busca mejorar la eficiencia operativa y la transparencia en la gestión pública. Se espera que este sistema facilite la colaboración entre los distintos departamentos del gobierno, reduciendo el uso de papel y promoviendo un enfoque más sostenible en la gestión documental.

## 2. Representación Arquitectónica

### 2.1 Escenarios

- **Escenario de Funcionalidad**

**Necesidad:** Se busca que los usuarios sean capaces de entrar en el aplicativo web en cualquier momento que ellos quieran o deseen.

**Solución:** Mantener el aplicativo web activa durante el tiempo las 24 horas del día, siendoposible su uso en ese intervalo de tiempo, en caso de no poder acceder también se puedellamar a soporte para acceder.

**Justificación:** Al ser una página web, debe estar disponible en todo momento.

- **Escenario de Usabilidad**

**Necesidad:** Se precisa que los usuarios puedan entender rápida e intuitivamente el uso delaplicativo web.

**Solución:** Implementación de una interfaz sencilla y atractiva al ojo para que los usuariospuedan realizar su navegación desde el primer minuto.

**Justificación:** Al hacer que una interfaz sea fácil de manipular, provoca que los usuariosse sientan relajados de usarla.

- **Escenario de Confiabilidad**

**Necesidad:** Se requiere que los usuarios tengan la confianza de realizar o usar cualquier acción o función dentro de la página y que posean una página web segura de usar o navegar para generar confianza en la misma.

**Solución:** Implementación de creación de cuentas de usuarios para acceder al aplicativo web que se encargará de proteger toda la información o datos de los clientes.

**Justificación:** El uso de las cuentas de usuarios proporciona cierto nivel de seguridad a cualquier sistema o página que necesite resguardarse de cualquier ataque o circunstancia, además de ser cómodo para los clientes de usar.

- **Escenario de Rendimiento**

**Necesidad:** Se requiere que el sistema corra con normalidad de forma rápida y eficiente, cumpliendo con la carga de los procesos que requieran los clientes sin demora.

**Solución:** Correcta y buena implementación de los requerimientos establecidos sin ningún fallo, generando tranquilidad en las operaciones de los usuarios.

**Justificación:** Al implementarse de buena manera las funciones del aplicativo web, se evitan posibles errores que retrasen y ralenticen los procesos realizados dentro del sistema.

- **Escenario de Mantenibilidad**

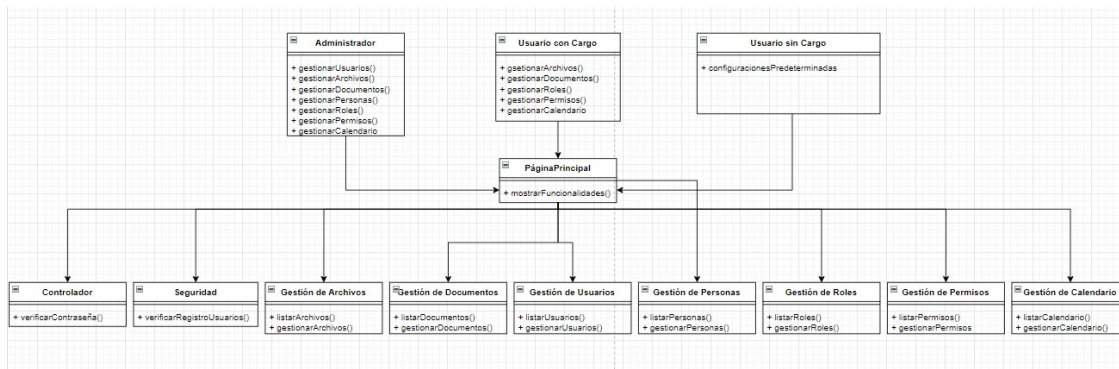
Necesidad: Se requieren que el aplicativo web sea capaz de mantenerse dentro de lo posible de manera sencilla en caso de cualquier problema y se adapte a las adversidades. Solución: Revisar constantemente el funcionamiento de la página web en busca de posibles fallos que comprometan el sistema para los usuarios.

Justificación: Tratándose el proyecto sobre la creación de un nuevo sitio web y las mejoras e integraciones de funciones, se requiere dar resultados a la altura de lo establecido.



## 2.2 Vista lógica

### Diagrama de Vista Lógica

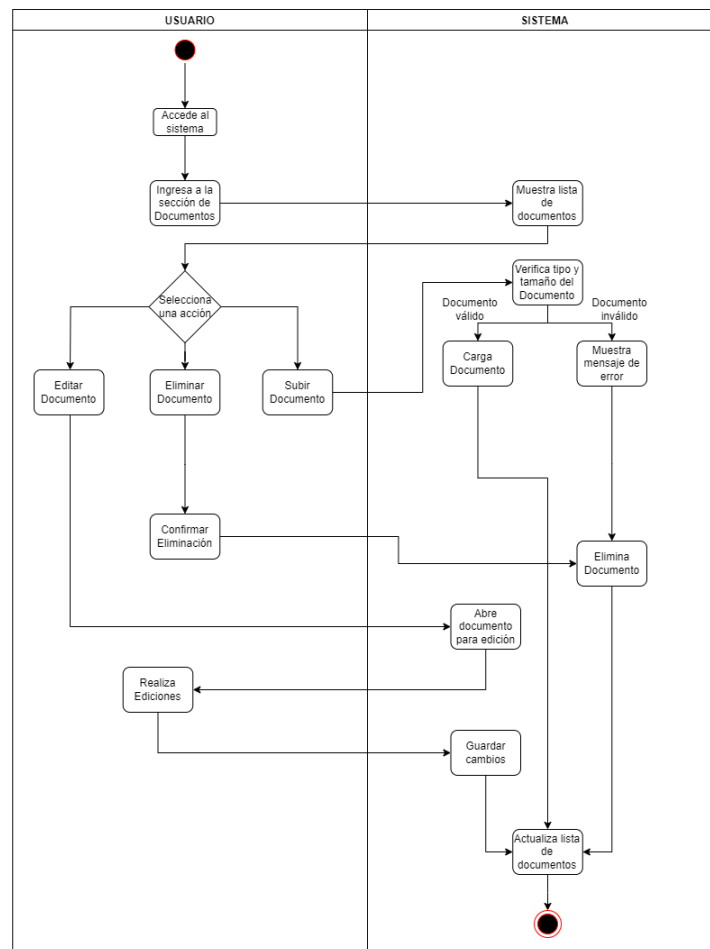


**Fuente:** Elaboración Propia

**Descripción:** La Vista Lógica es un diagrama que muestra la estructura organizacional de un sistema de software, dividido en tres módulos principales (Autenticación, Módulo con Cargo y Usuario sin cargo) que controlan los niveles de acceso y se conectan con diferentes funcionalidades del sistema, permitiendo visualizar claramente la arquitectura y navegación del software.

## 2.3 Vista del Proceso

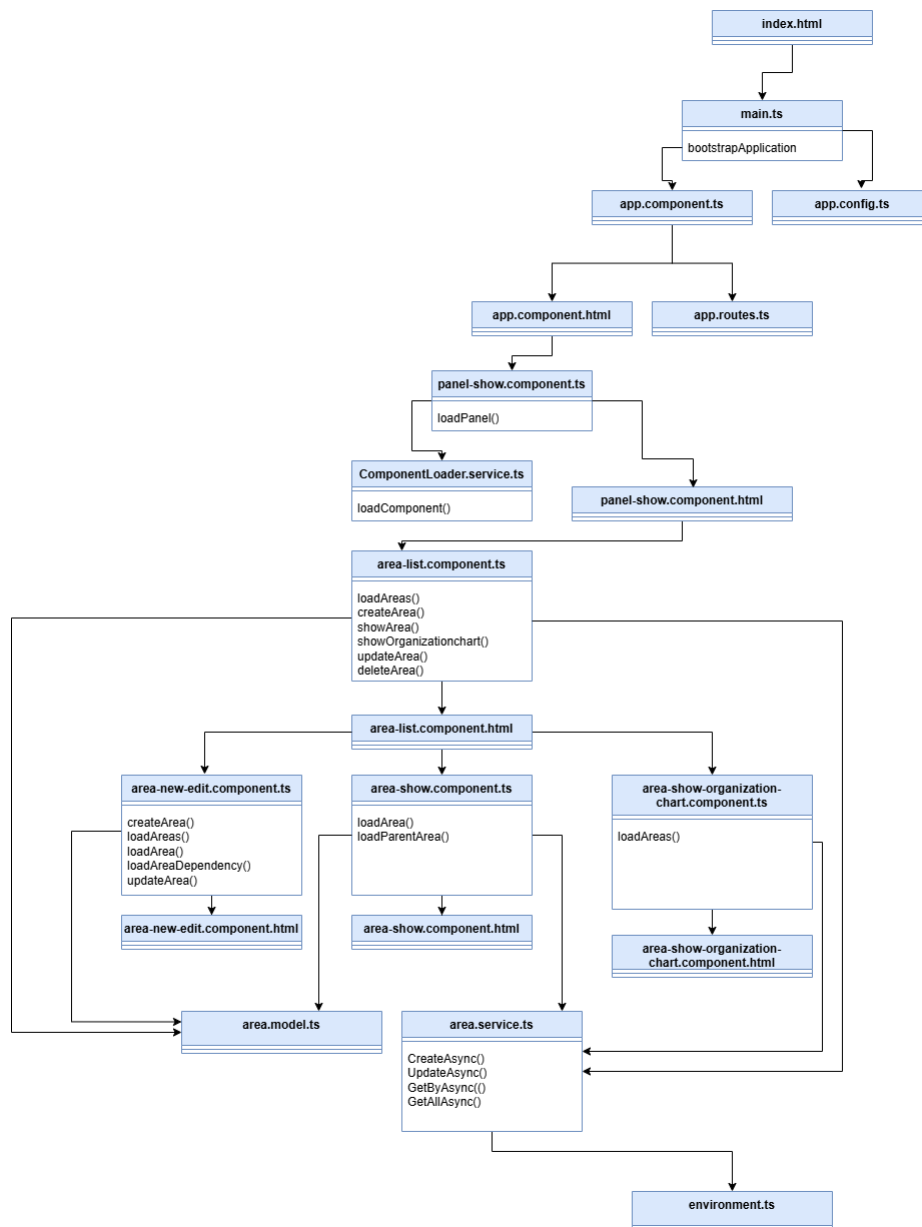
### Diagrama de Vista del Proceso



**Fuente:** Elaboración Propia

**Descripción:** La Vista del Proceso es un diagrama que ilustra el flujo de trabajo dividido en dos secciones: Usuario y Sistema. Esta muestra la secuencia de acciones para el manejo de documentos, desde su inicio hasta su finalización, incluyendo diferentes operaciones como editar, consultar y eliminar documentos, representando la interacción entre el usuario y el sistema de manera secuencial y ordenada.

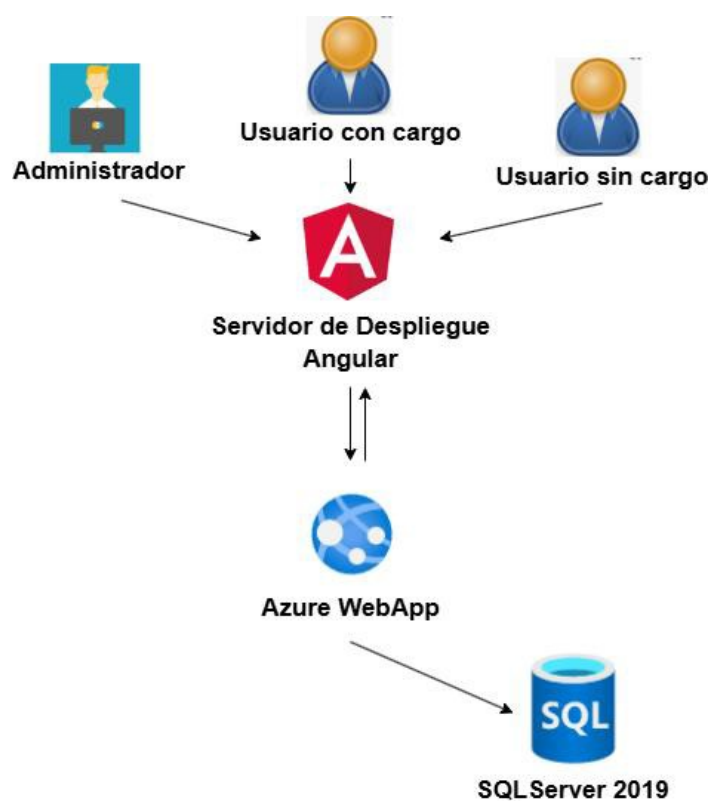
## 2.4 Vista del Desarrollo



**Fuente:** Elaboración Propia.

**Descripción:** El Diagrama de Vista de Desarrollo muestra la estructura jerárquica de archivos del proyecto Angular, desde del index.html hasta los componentes específicos, ilustrando cómo los diferentes archivos (.ts, .html) se relacionan y dependen entre sí para formar la aplicación web.

## 2.5 Vista Física



**Fuente:** Elaboración Propia.

**Descripción:** El Diagrama de Vista Física muestra la arquitectura de despliegue del sistema, donde los diferentes usuarios interactúan con un Servidor Angular, el cual mantiene una comunicación continua con Azure WebApp, que a su vez se conecta a SQL Server 2019 para el manejo de datos.

### **3. Objetivos y limitaciones arquitectónicas**

#### **3.1 Disponibilidad**

- Objetivo: Garantizar que el sistema esté disponible para los usuarios en todo momento, minimizando los tiempos de inactividad no planificados y manteniendo un alto nivel de acceso al sistema.
- Limitación: Puede haber momentos de inactividad planificados para mantenimientos y actualizaciones del sistema, durante los cuales funcionalidades podrían no estar disponibles.

#### **3.2 Seguridad**

- Objetivo: Proteger los datos y la integridad del sistema contra accesos no autorizados, ataques cibernéticos y pérdidas de información.
- Limitación: Aunque se implementaran medidas de seguridad, como encriptación de datos y autenticación de usuarios, no se puede garantizar una protección absoluta contra todas las posibles amenazas de seguridad.

#### **3.3 Adaptabilidad**

- Objetivo: Permitir que el sistema se adapte de manera flexible a cambios en los requisitos del negocio y en el entorno tecnológico, asegurando su funcionamiento continuo y efectivo.
- Limitación: La capacidad de adaptabilidad del sistema puede estar limitada por restricciones de recursos, tiempo y presupuesto, lo que podría afectar la rapidez con la que se pueden implementar cambios y mejoras.

#### **3.4 Rendimiento**

- Objetivo: Garantizar que el sistema pueda mejorar cargas de trabajo elevadas de manera eficiente y sin experimentar retrasos significativos, proporcionando una experiencia de usuario fluida y satisfactoria.
- Limitación: Aunque se realizarán optimizaciones de rendimiento, como la optimización del código y el uso de técnicas de caching puede haber limitaciones en la capacidad de procesamiento y almacenamiento que podrían afectar al rendimiento del sistema en momentos de alta demanda.

## 4. Análisis de Requerimientos

### 4.1 Requerimientos Funcionales

Número	Requerimiento Funcional	Casos de uso	Prioridad
RF - 01	El sistema deberá de proporcionar un usuario y contraseñaválidas para poder tener acceso al sistema	Gestión de Usuarios	1
RF - 02	El sistema permitirá registrar a los usuarios.	Gestión de Usuarios	1
RF - 03	El sistema permitirá gestionar los roles, permisos y el espaciode almacenamiento por usuario	Gestión de Permisos	1
RF - 04	El sistema permitirá configurar los archivos permitidos y sutamaño máximo para todos los usuarios	Gestión de Archivos	1
RF - 05	El sistema permitirá almacenar los archivos con encriptaciónen el servidor de archivos	Gestión de Documentos	1
RF - 06	El sistema permitirá compartir archivos con otros usuarios del sistema. (lectura o escritura)	Gestión de Archivos	1
RF - 07	El sistema permitirá eliminar archivos.	Gestión de Archivos	3
RF - 08	El sistema permitirá firmar el documento digitalmente para suuso posterior	Gestión de Documentos	1
RF - 09	El sistema permitirá enviar documentos a los usuariosinternos. (SGTD)	Gestión de Documentos	1
RF - 10	El sistema permitirá dar seguimiento al documento y ver elestado de todo el ciclo de vida del documento.	Gestión de Documentos	2
RF - 11	El sistema permitirá visualizar el documento	Gestión de Calendario	3
RF - 12	El sistema permitirá a los usuarios contar con un calendariode eventos	Gestión de Calendario	2
RF - 13	El sistema añadirá automáticamente eventos a los calendarios de los usuarios que acepten invitaciones, verificando la disponibilidad de horarios y, en caso de conflictos, asignandoun representante para asistir en su lugar.	Gestión de Calendario	1
RF- 14	El sistema deberá notificar al usuario cuando un documento es recibido o recibe respuesta de uno enviado.	Gestión de Documentos	2

RF -15	El sistema deberá notificar al usuario cuando un evento es creado a partir de la aceptación de un documento de invitación.	Gestión de Calendario	2
RF- 16	El sistema deberá notificar al usuario que un archivo le ha sido compartido.	Gestión de Archivos	2

**Fuente:** Elaboración Propia.

**Descripción:** La Tabla de Requerimientos Funcionales describe las características y capacidades específicos que debe tener el sistema, organizadas por identificador (RF), incluyendo funciones de gestión de usuarios, documentos, archivos y calendario, cada una con su respectiva prioridad de implementación.

## 4.2 Requerimientos No Funcionales

Número	Requerimiento No Funcional	Descripción	Prioridad
RF - 01	Eficiencia	Todo proceso o funcionalidad de la aplicación deberá de tener un tiempo de respuesta de 10 segundos máximo.	2
RF - 02	Disponibilidad	La aplicación deberá funcionar al 90% en caso de error o no esté disponible, notificará con un mensaje de error con el servicio.	2
RF - 03	Usabilidad	El tiempo de aprendizaje del aplicativo por un usuario deberá ser menor a 2 horas	2
RF - 04	Escalabilidad	La aplicación mantendrá su buen funcionamiento y rendimiento tras los cambios o crecimiento que pueda aplicarse a lo largo de su ciclo de vida.	2
RF - 05	Seguridad	El tiempo de aprendizaje del aplicativo por un usuario deberá ser menor a 2 horas	2

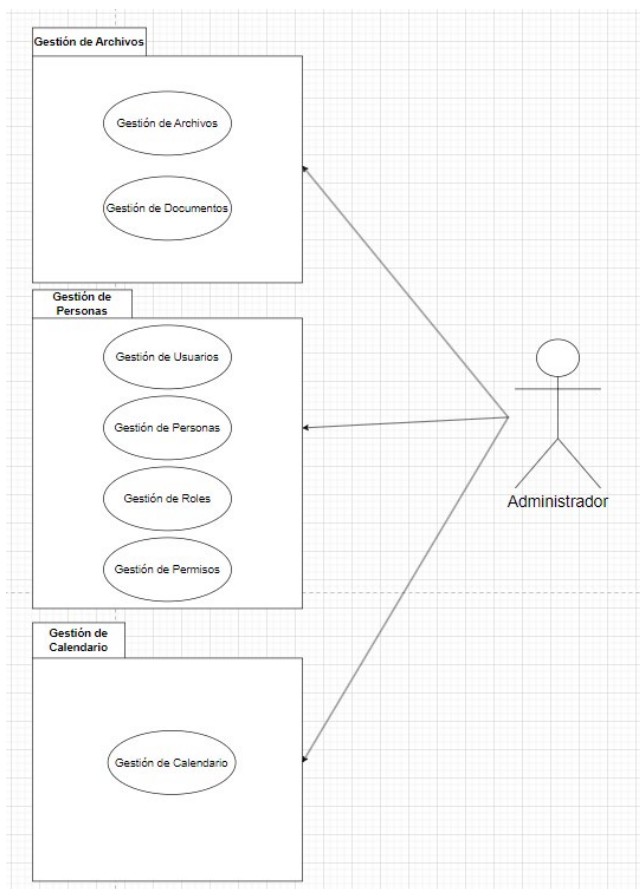
**Fuente:** Elaboración Propia.

**Descripción:** La Tabla de Requerimientos No Funcionales especifica las características cualitativas del sistema, incluyendo eficiencia (tiempo de respuesta), disponibilidad, usabilidad, escalabilidad y seguridad, cada una con prioridad 2, que definen el comportamiento y rendimiento general del sistema en lugar de sus funcionalidades específicas.



## 5. Vistas de Caso de Uso

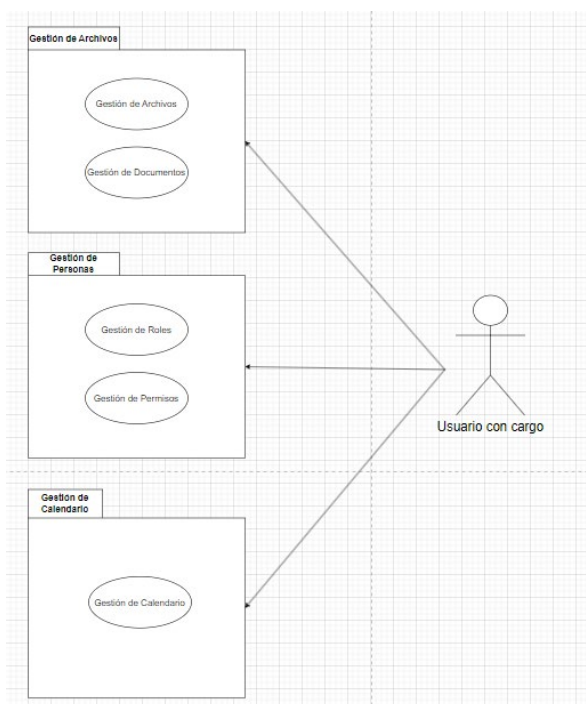
### - Administrador



**Fuente:** Elaboración Propia.

**Descripción:** El diagrama de Caso de Uso del Administrador muestra las diferentes funcionalidades a las que tiene acceso este rol, organizadas en tres módulos principales: Gestión de Archivos (que incluye gestión de archivos y documentos), Gestión de Permisos (que abarca gestión de usuarios, permisos, roles y permisos) y Gestión de Calendario, representando todas las interacciones que puede realizar el administrador en el sistema

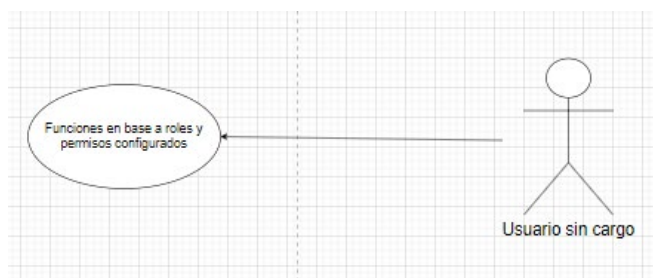
## - Usuario con cargo



**Fuente:** Elaboración Propia.

**Descripción:** El diagrama de Casos de Uso del Usuario con Cargo ilustra las funcionalidades disponibles para este tipo de usuario, distribuidas en tres módulos: Gestión de Archivos (incluye gestión de archivos y documentos), Gestión de Permisos (con gestión de roles y permisos) y Gestión de Calendario, mostrando un nivel de acceso más limitado que el del Administrador.

## - Usuario sin cargo



**Fuente:** Elaboración Propia.

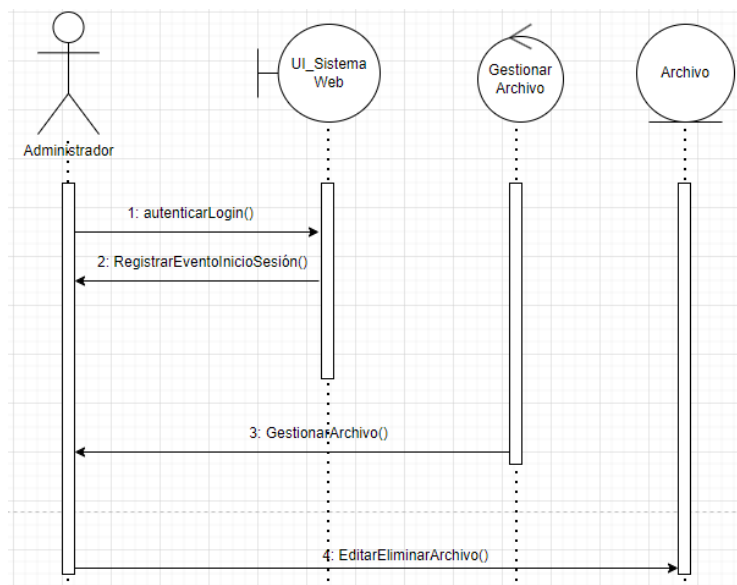
**Descripción:** El diagrama de Casos de Uso del Usuario sin Cargo ilustra las funcionalidades disponibles para este tipo de usuario, distribuidas en un módulo que serían las Funciones en base a roles y permisos configurados.

## 6. Vista Lógica

### 6.1 Diagramas de Secuencia

#### ➤ Diagrama de Secuencia de Gestionar Archivo.

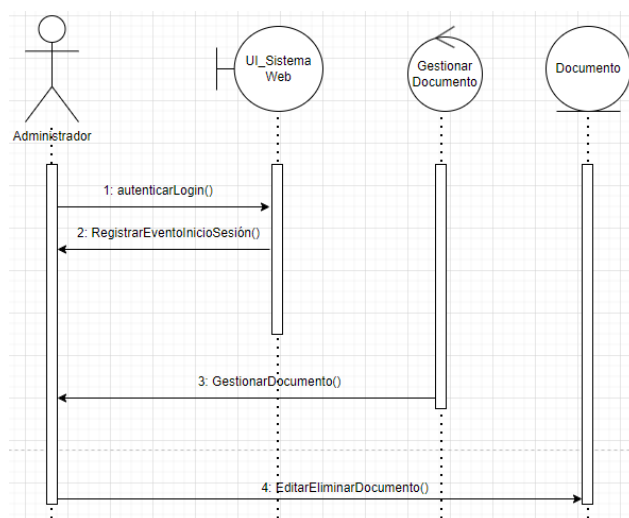
Fuente:Elaboración Propia



Descripción: Este diagrama muestra cómo los diferentes componentes del sistema interactúan entre sí para llevar a cabo las tareas relacionadas con la gestión de archivo, desde la autenticación del usuario hasta la manipulación de los archivos en sí.

#### ➤ Diagrama de Secuencia de Gestionar Documento.

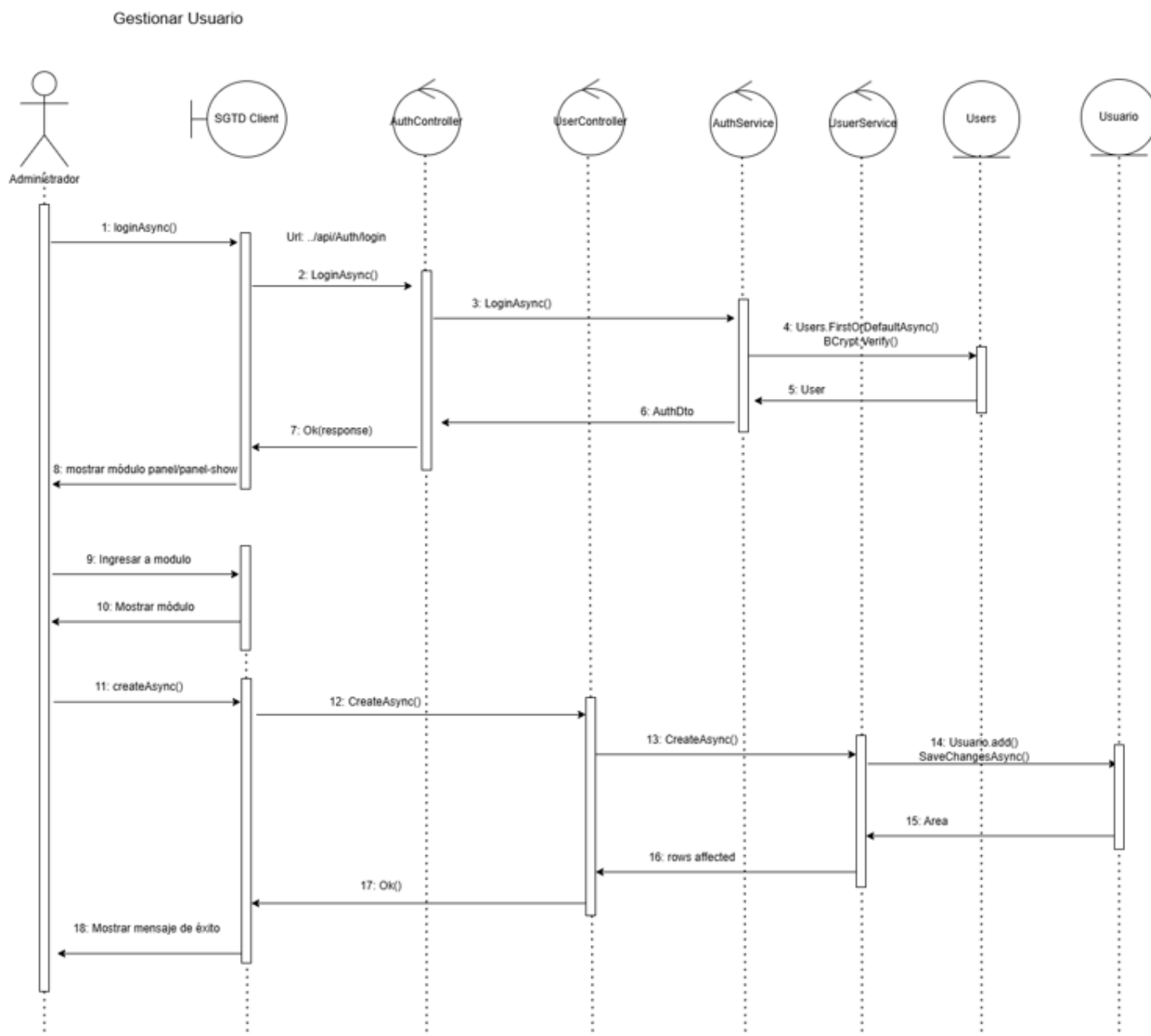
Fuente:Elaboración Propia



Descripción: Este diagrama ilustra el flujo de control y secuencia de mensajes entre los diferentes componentes del sistema de gestión de documentos.

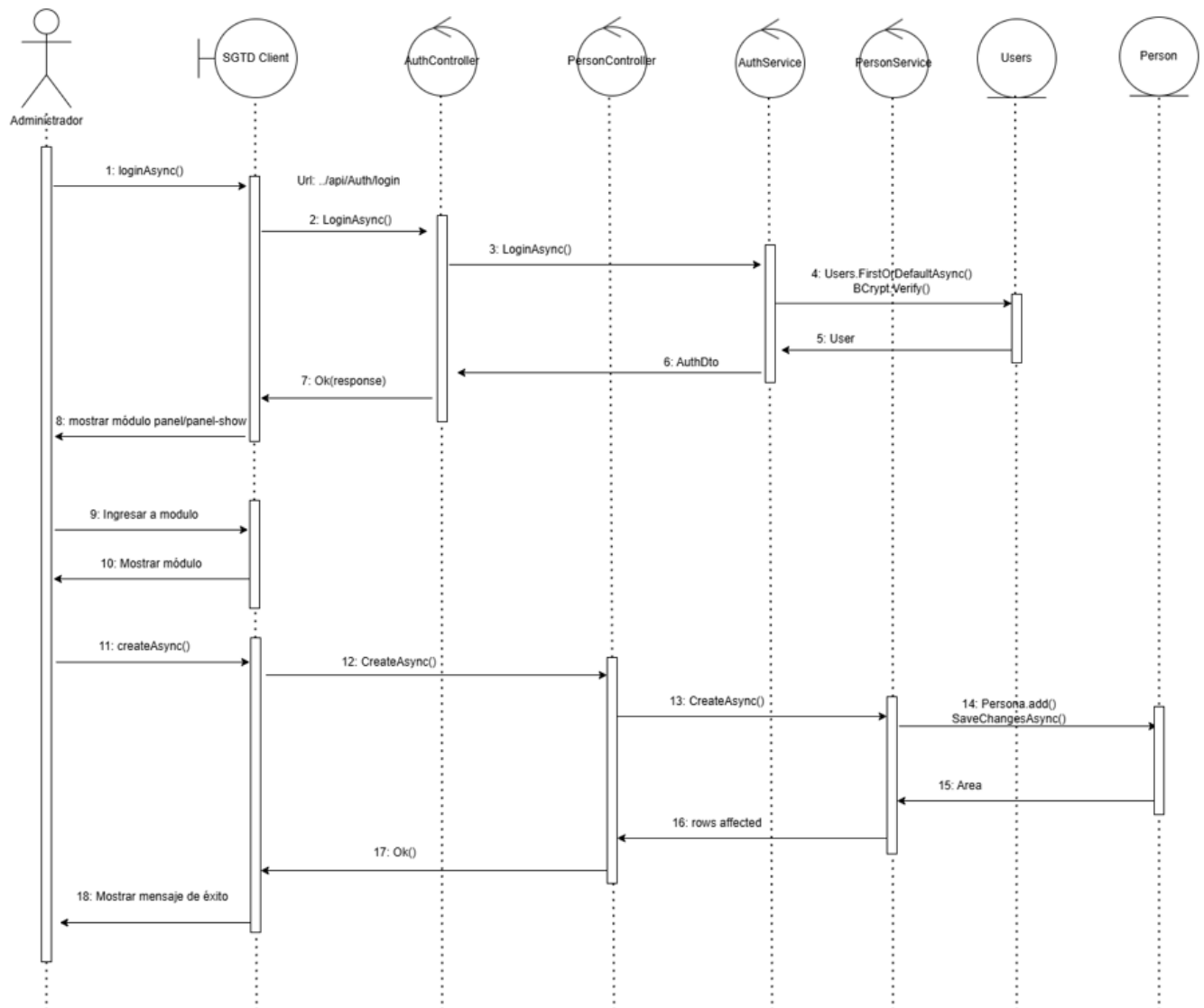
➤ Diagrama de Secuencia de Gestionar Usuario.

Fuente:Elaboración Propia



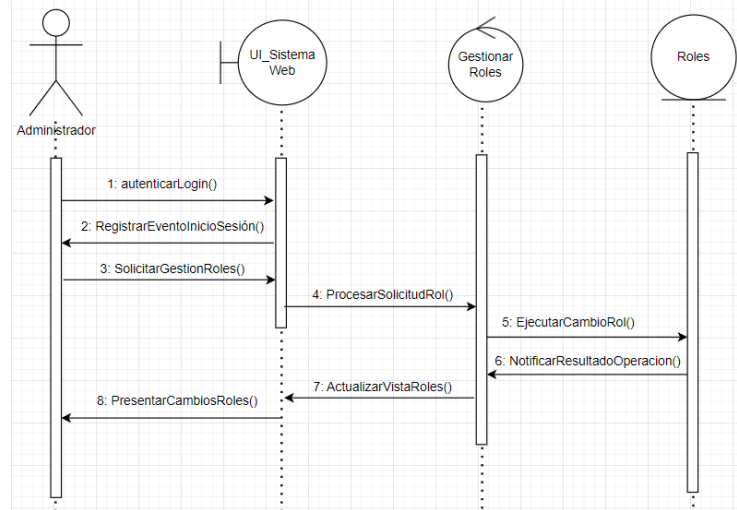
Descripción: Este diagrama detalla los pasos dentro del proceso del caso de uso Gestionar Usuario donde se destacan las acciones del mismo.

➤ Diagrama de Secuencia de Gestionar Persona.  
Fuente:Elaboración propia



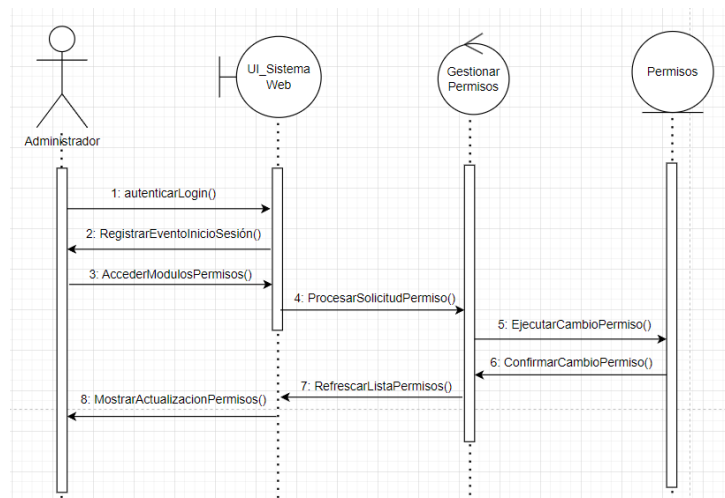
Descripción: Este diagrama muestra cómo el sistema maneja la autenticación de usuario hasta las operaciones específicas sobre gestión de personas.

➤ Diagrama de Secuencia de Gestionar Rol. Fuente: Elaboración Propia



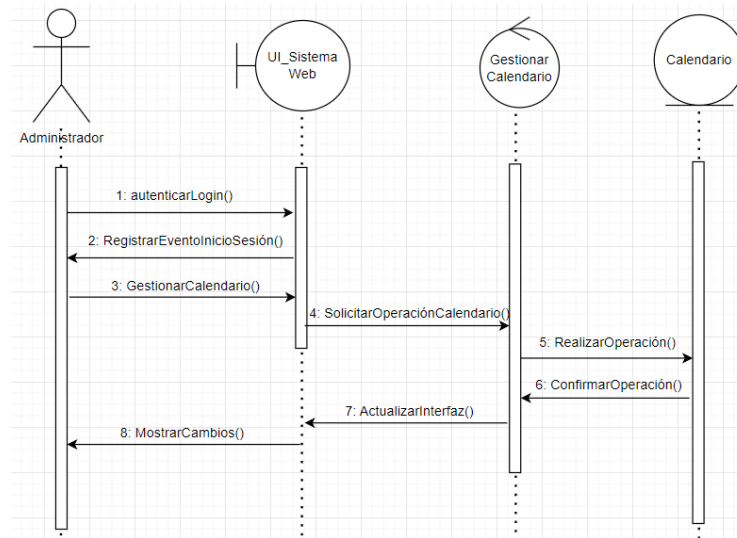
Descripción: Este diagrama muestra cómo el sistema maneja la autenticación de usuario hasta las operaciones específicas sobre gestión de roles.

➤ Diagrama de Secuencia de Gestionar Permisos. Fuente:Elaboración Propia



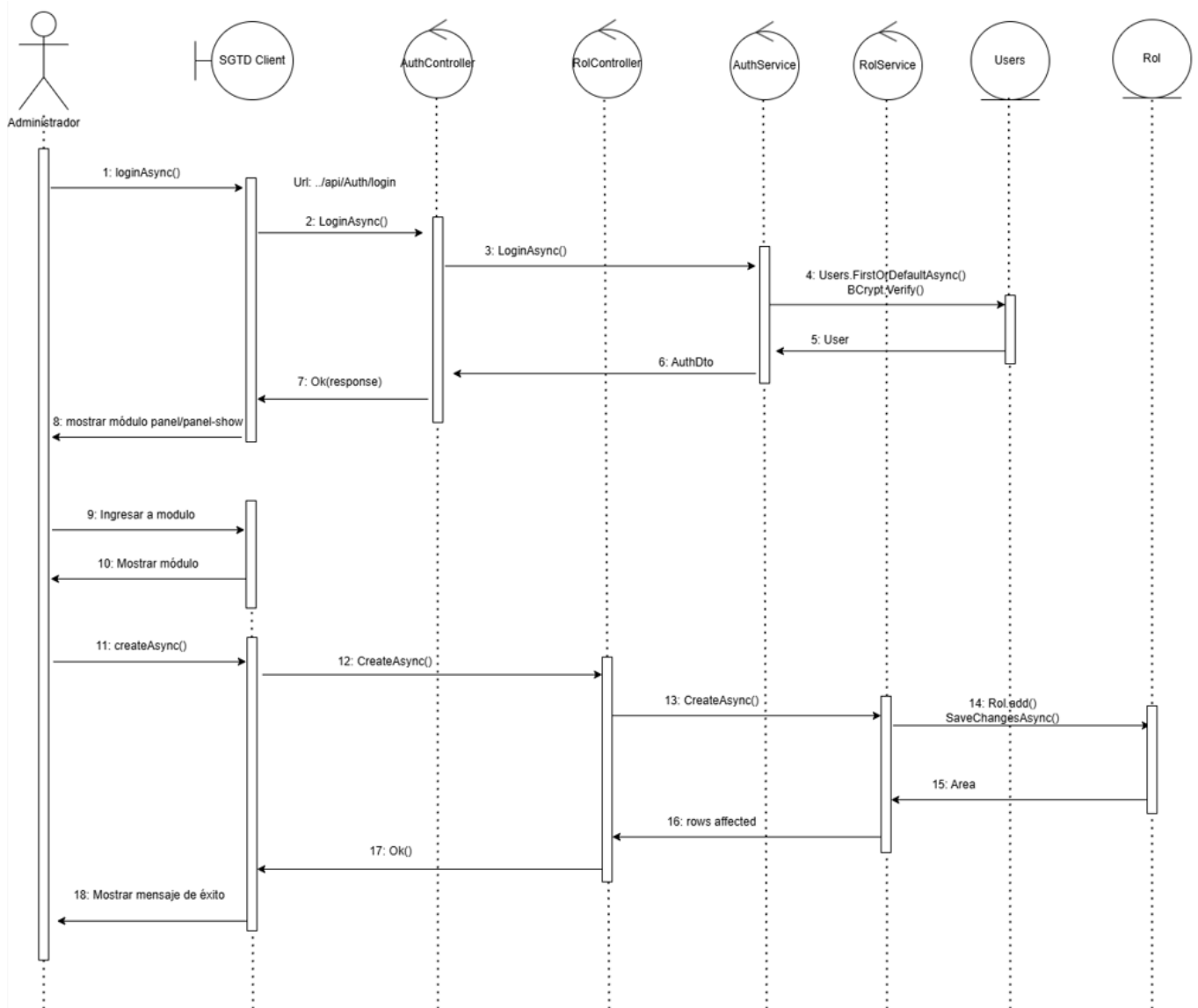
Descripción: Este diagrama muestra cómo el sistema maneja la autenticación de usuario hasta las operaciones específicas sobre gestión de permisos.

- Diagrama de Secuencia de Gestionar Calendario.  
Fuente:Elaboración propia.



Descripción: Este diagrama muestra cómo el sistema maneja la autenticación de usuario hasta las operaciones específicas sobre gestión de calendario.

➤ Diagrama de Secuencia de Gestionar Cargo.  
Fuente:Elaboración propia.

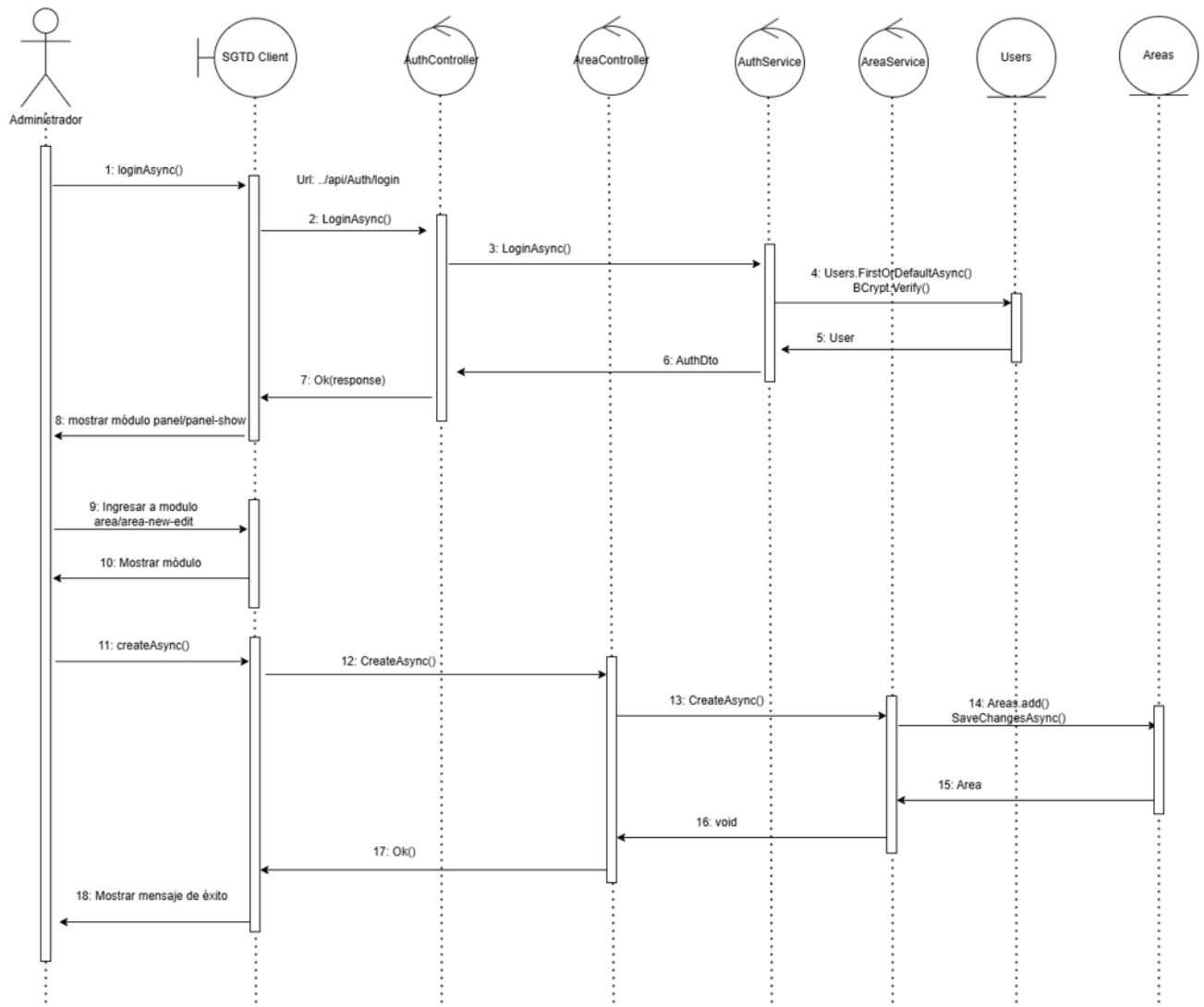


Descripción: Este diagrama muestra cómo el sistema maneja la autenticación de usuario hasta las operaciones específicas sobre gestión de cargo.



➤ Diagrama de Secuencia de Gestionar Area.

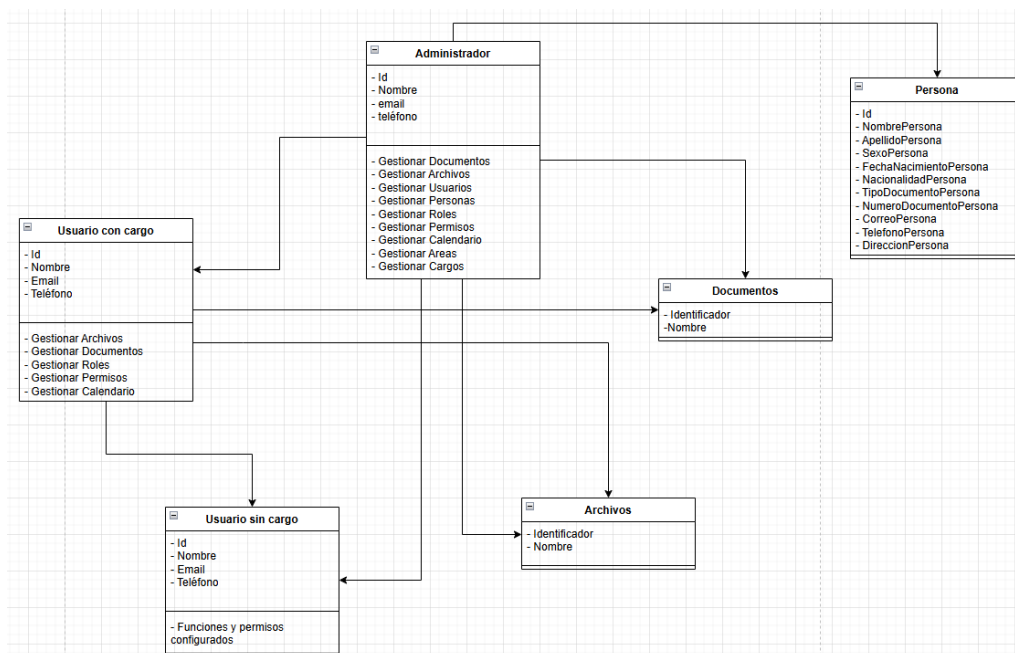
Fuente:Elaboración propia.



Descripción: Este diagrama muestra cómo el sistema maneja la autenticación de usuario hasta las operaciones específicas sobre gestión de áreas.

## 6.2 Diagrama de Clases

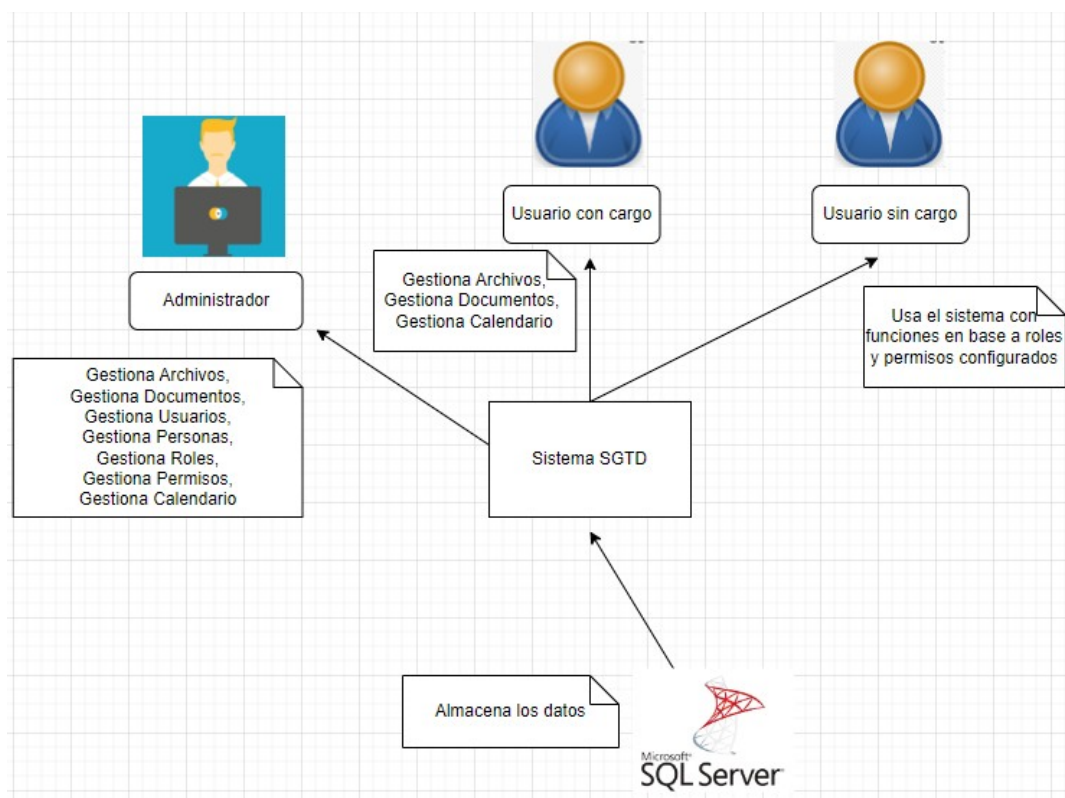
### Diagrama de Clases



**Fuente:** Elaboración Propia

**Descripción:** El Diagrama de Clases muestra la estructura de un sistema de gestión con sus componentes principales, cada clase contiene sus atributos y métodos específicos.

### 6.3 Diagrama Contextual



**Fuente:** Elaboración Propia

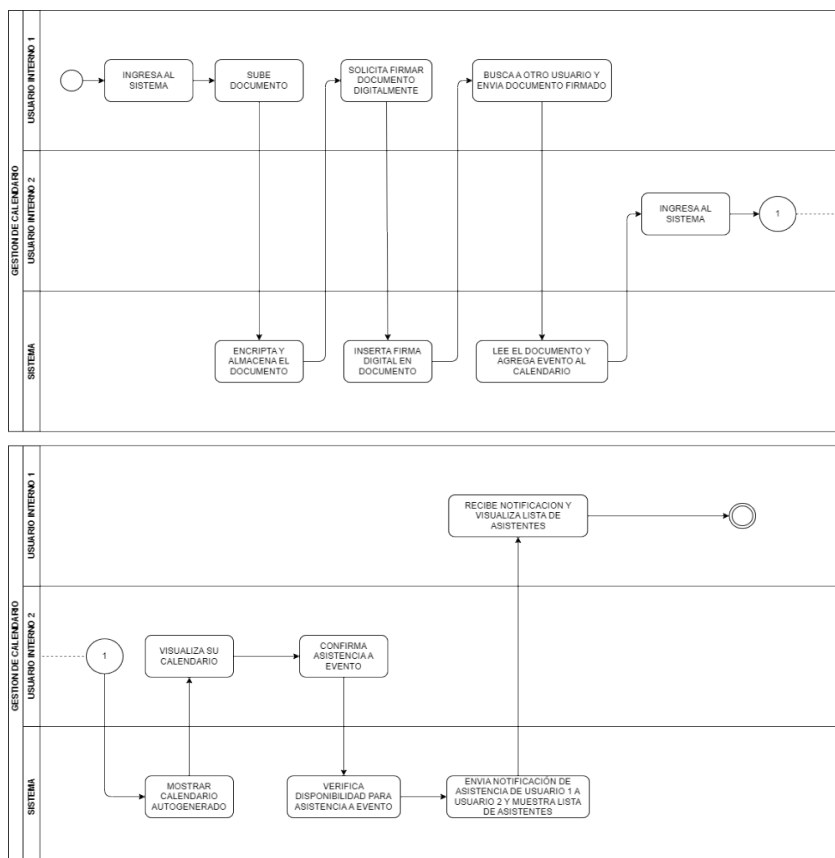
**Descripción:** El Diagrama Contextual muestra la interacción entre el Sistema SGTD, sus usuarios (Administrador y usuarios con/sin cargo) y SQL Server, definiendo cómo cada rol gestionará y accederá a las diferentes funcionalidades del sistema.

## 7. Vista de Procesos

### 7.1 Diagrama de Proceso Actual

- ❖ Diagrama de trámite documentario interno
  - No cuenta con diagrama de proceso actual
- ❖ Diagrama de almacenamiento de archivos
  - No cuenta con diagrama de proceso actual
- ❖ Diagrama de gestión de calendario
  - No cuenta con un diagrama de gestión de calendario estandarizado ya que actualmente cada colaborador gestiona sus eventos de la manera más conveniente posible.

### 7.2 Diagrama de Proceso Propuesto

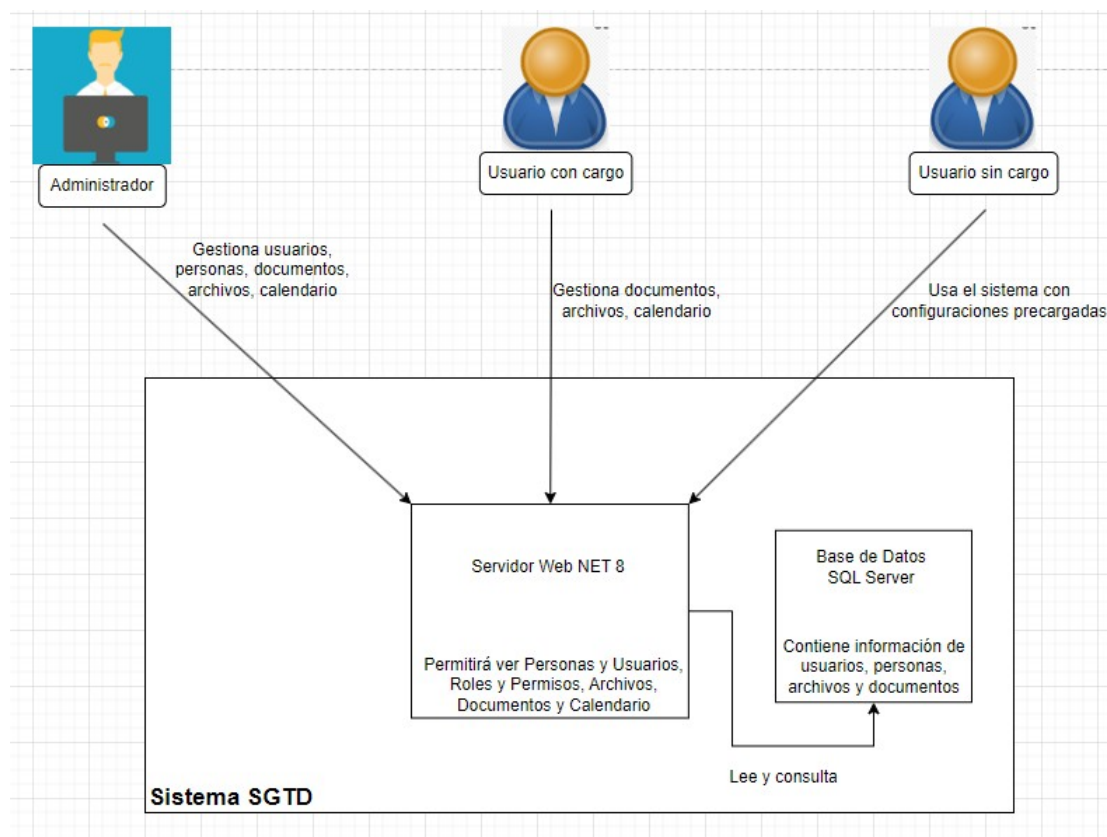


**Fuente:** Elaboración Propia

**Descripción:** Este diagrama muestra la interacción entre 2 usuarios internos y el sistema en la que el usuario ingresa al sistema, sube un documento el sistema encripta y almacena el documento luego el usuario solicita firmar dicho documento, el sistema inserta la firma digital, este usuario busca a otro usuario y envía el documento, el sistema agrega evento al calendario.

## 8. Vista de Despliegue

### 8.1 Diagrama de Contenedor

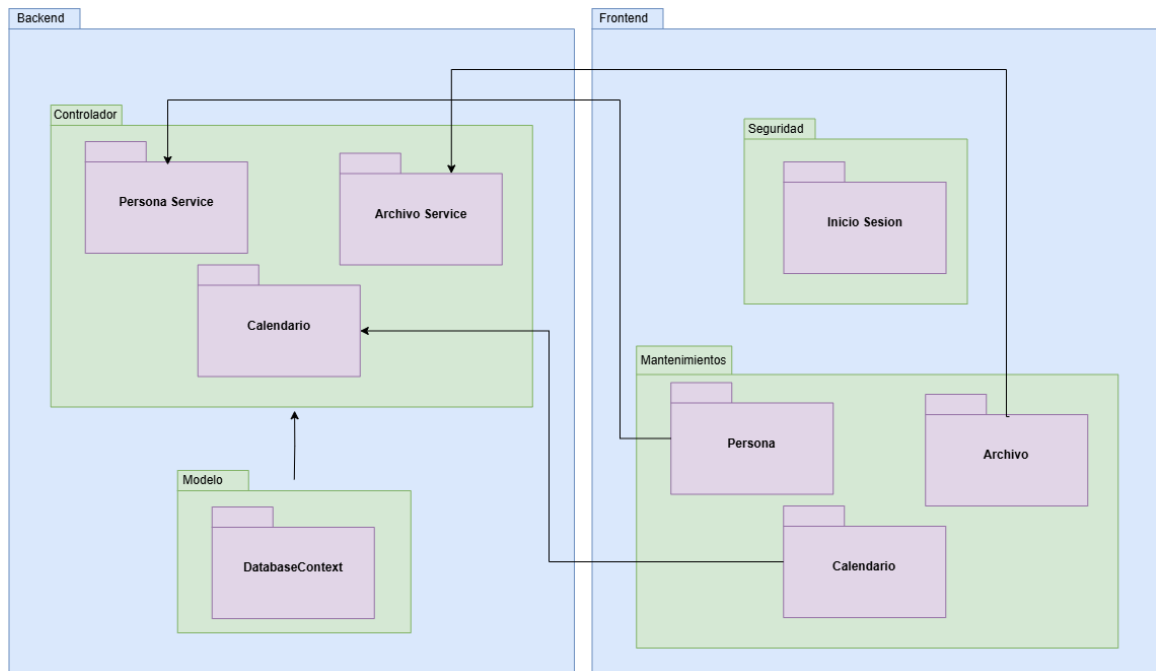


**Fuente:** Elaboración Propia

**Descripción:** El Diagrama de Contenedor muestra la arquitectura del Sistema SGTD, donde diferentes tipos de usuarios interactúan con un Servidor Web .NET que gestiona roles y permisos, conectándose a una base de datos SQL Server que almacena toda la información del sistema

## 9. Vista de Implementación

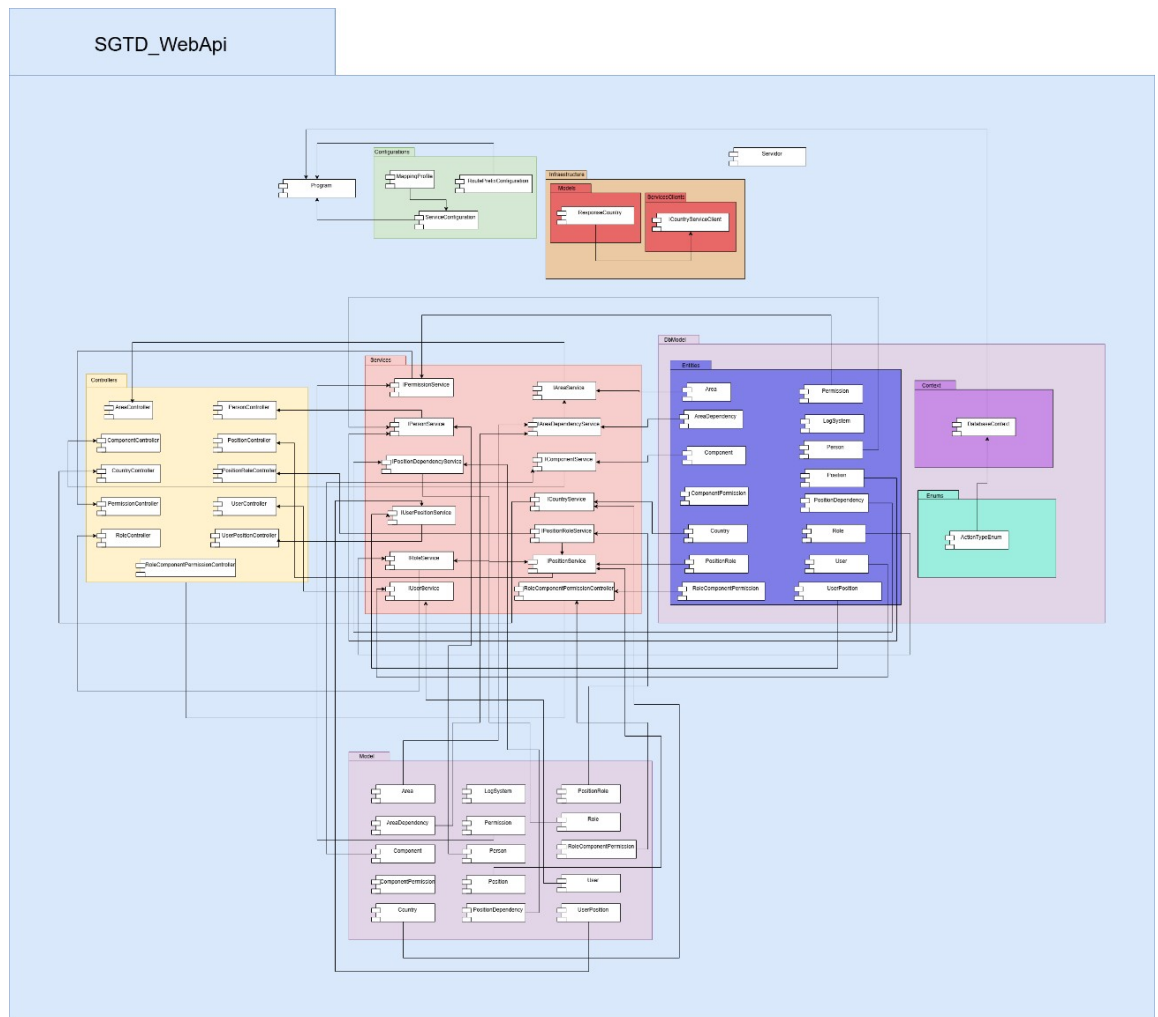
### 9.1 Diagrama de Paquetes



**Fuente:** Elaboración Propia

**Descripción:** El Diagrama de Paquetes ilustra la organización modular del sistema, dividido en dos secciones principales: Backend (con servicios d Personal, Archivo y Calendario) y Frontend (con Vistam Seguridad y Mantenimiento), mostrando cómo se estructuran los diferentes componentes del software.

## 9.2. Diagrama de Componentes



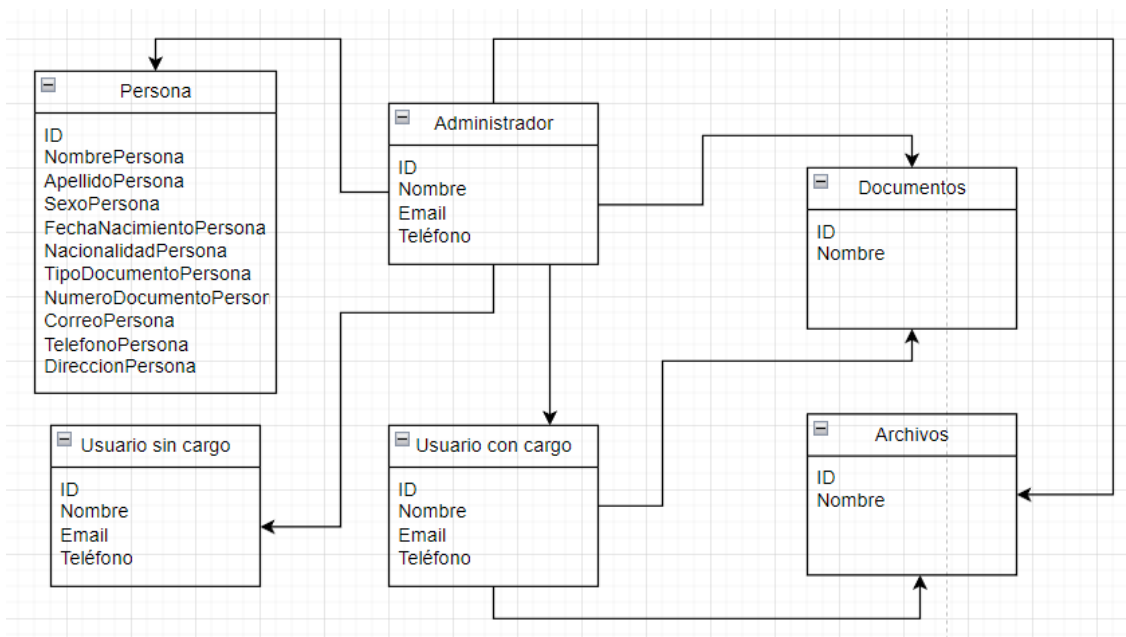
**Fuente:** Elaboración Propia

**Descripción:** Se muestra una vista rápida del Diagrama de componentes donde se muestra la estructura de componentes de a WebApi SGTD, visualizando cómo los diferentes módulos y servicios están interconectados, organizados en capas por colores que representan distintas funcionalidades del sistema, con sus respectivas dependencias y relaciones entre los componentes.

**Se Anexa el Diagrama para su mejor Visualización**

## 10. Vista de Datos

### 10.1. Diagrama Entidad Relación



**Fuente:** Elaboración Propia

**Descripción:** El Diagrama Entidad Relación muestra la estructura de la base de datos del sistema, donde se conectan las entidades principales (Persona, Administrador, Documentos, Archivos y Usuarios con(sin cargo)) y sus atributos, definiendo cómo se relacionan los datos entre sí.



## **11. Calidad**

### **11.1.Escenario de Seguridad**

- Necesidad: Es fundamental garantizar la protección de los datos sensibles de los usuarios y la integridad del sistema contra posibles ataques o vulnerabilidades.
- Solución: Implementar un sistema de encriptación para proteger la comunicación entre el navegador del usuario y el servidor web. Además, se establecerán medidas básicas de autenticación de usuarios, como contraseñas seguras y verificación de correo electrónico, para bloquear posibles intentos de acceso no autorizado.
- Justificación: Dado que la página web maneja información confidencial de los usuarios, como datos personales y transacciones financieras, es crucial mantener un entorno seguro y confiable para proteger la privacidad y la confianza de los clientes.

### **11.2.Escenario de Usabilidad**

- Necesidad: Se precisa que los usuarios puedan entender rápida e intuitivamente el uso de la página web.
- Solución: Implementación de una interfaz sencilla y atractiva al ojo para que los usuarios puedan realizar su navegación desde el primer minuto.
- Justificación: Al hacer que una interfaz sea fácil de manipular, provoca que los usuarios se sientan relajados de usarla y no alejarlos de la empresa.



### 11.3. Escenario de Adaptabilidad

- Necesidad: Es esencial que el sistema web sea capaz de adaptarse a cambios en los requisitos del negocio y en el entorno tecnológico, garantizando su funcionamiento continuo y efectivo.
- Solución: Adoptar el patrón arquitectónico Modelo-Vista- Controlador (MVC), que divide la aplicación en tres componentes principales: el Modelo, que gestiona los datos y la lógica de negocio; la Vista, que se encarga de la presentación de la interfaz de usuario; y el Controlador, que coordina las interacciones entre el Modelo y la Vista. Al utilizar MVC, se logra una arquitectura flexible y modular que facilita la incorporación de nuevas funcionalidades y a adaptación a cambios en los requisitos del negocio o en el entorno tecnológico.
- Justificación: Dado que el entorno tecnológico y las demandas del negocio pueden cambiar con el tiempo, es crucial que el sistema web pueda adaptarse de manera ágil y eficiente. El uso de MVC proporciona una estructura clara y organizada que facilita la evolución que facilita la evolución y la mantenibilidad del sistema, permitiendo realizar modificaciones en cada componente de forma independiente y sin afectar al resto de la aplicación. Esto garantiza que el sistema pueda mantener su relevancia y utilidad a lo largo del tiempo, brindando una experiencia continua y satisfactoria a los usuarios

#### 11.4. Escenario de Disponibilidad

- Necesidad: Se reconoce que la disponibilidad continua del sistema web no será un punto fuerte del proyecto. Se utilizará una única instancia para cada servicio necesario, lo que puede implicar tiempos de inactividad planificados durante mantenimientos y actualizaciones.
- Solución: Aunque la disponibilidad continua del sistema no será un punto fuerte del proyecto, se implementarán medidas básicas para minimizar los tiempos de inactividad no planificados. Se utilizará una única instancia para cada servicio necesario, pero se establecerán procedimientos para realizar mantenimientos y actualizaciones de manera programada, minimizando así el impacto en la disponibilidad del sistema. Además, se establecerá un proceso de monitoreo básico para detectar y abordar rápidamente posibles problemas que puedan surgir.
- Justificación: Aunque la disponibilidad continua del sistema no será prioridad, se tomarán medidas para minimizar los tiempos de inactividad no planificados. Sin embargo, es importante tener en cuenta que habrá momentos en los que ciertos servicios pueden estar temporalmente no disponibles debido a mantenimientos programados o actualizaciones.

## 11.5.Otro Escenario

### ➤ Escenario de Escalabilidad

- Necesidad: Es fundamental que el sistema web pueda crecer de manera flexible y eficiente para adaptarse al aumento en la demanda de usuarios y servicios, sin experimentar degradación en el rendimiento.
- Solución: Se implementará una solución básica de escalabilidad utilizando servicios en la nube que permite agregar recursos según sea necesario para manejar picos de demanda. Esto puede incluir el uso de instancias de servidor escalables y servicios de base de datos gestionados. Además, se establecerán umbrales de capacidad y alertas para identificar y abordar proactivamente cualquier necesidad de escalado.
- Justificación: A medida que el sistema web gane popularidad y aumente su base de usuarios, es crucial que pueda escalar de manera eficiente para mantener un rendimiento óptimo. La implementación de una arquitectura escalable garantiza que el sistema pueda crecer sin problemas y seguir proporcionando una experiencia de usuario satisfactoria, incluso en momentos de alta demanda. Esto es esencial para la competitividad y el éxito a largo plazo del proyecto.