



INSTITUTO POLITÉCNICO NACIONAL
Escuela Superior de Cómputo
ESCOM

Trabajo Terminal

**“Sistema de Almacenamiento Seguro de Archivos
basado en Secreto Compartido”**

2016-B009

Presentan

Armenta García Guadalupe Javier
Cárdenas Castillo Víctor Hugo
Moreno Zárate Víctor Gibrán

Directores

Dra. Sandra Díaz Santiago
M. en C. Axel Ernesto Moreno Cervantes



Mayo 2017

**INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO
SUBDIRECCIÓN ACADÉMICA**

No TT: 2016-B009
Mayo 2017

Documento Técnico

Sistema de Almacenamiento Seguro de Archivos basado en Secreto Compartido

Presentan:

Armenta García Guadalupe Javier
Cárdenas Castillo Víctor Hugo
Moreno Zárate Víctor Gibrán

Directores:

Dra. Sandra Díaz Santiago
M. en C. Axel Ernesto Moreno Cervantes

Resumen: El presente manuscrito contiene la documentación del Trabajo Terminal I con título *Sistema de Almacenamiento Seguro de Archivos basado en Secreto Compartido*. En este Trabajo Terminal se desarrollará un sistema de almacenamiento de archivos distribuido capaz de proveer confidencialidad, integridad, autenticación y disponibilidad a la información almacenada en dichos sistemas, a través del uso de un esquema de partición de secretos.

Palabras clave: Criptografía simétrica, Sistema Distribuido, Secreto Compartido, Repositorio de archivos.

Índice

1. Introducción	1
2. Preliminares	5
2.1. Sistemas distribuidos	5
2.1.1. Tipos de sistemas distribuidos.	5
2.1.2. Propiedades de los sistemas distribuidos.	7
2.2. Criptografía	9
2.2.1. Ataques Criptográficos	10
2.3. Criptografía asimétrica	11
2.4. Criptografía simétrica	11
2.4.1. Cifrado por bloques	12
2.4.2. Modos de operación	14
2.5. Funciones Hash	16
2.5.1. Propiedades Básicas	16
2.5.2. Resistencia a Ataques Criptoanalíticos	16
2.5.3. Principales Funciones Hash	17
3. Esquemas de secreto compartido	18
3.1. Esquema de umbral de Shamir	18
3.1.1. Proceso de inicialización y compartición de las partes	18
3.1.2. Proceso de recuperación	19
3.1.3. Ejemplo	19
3.2. Robust Computational Secret Sharing	21
4. Análisis	24
4.1. Herramientas tecnológicas	24
4.1.1. Sistemas Operativos	24
4.1.2. Lenguajes de Programación	24
4.1.3. Bibliotecas a utilizar	26
4.1.4. Servidores	27
4.1.5. Sistemas Gestores de Bases de Datos	27
4.1.6. Frameworks para sistemas distribuidos	28
4.2. Estudio de factibilidad	29
4.2.1. Factibilidad Técnica	29
4.2.2. Factibilidad Operativa	31
4.2.3. Factibilidad Económica	33
4.3. Metodología	34
4.4. Requerimientos	36
4.4.1. Requerimientos Funcionales	36
4.4.2. Requerimientos No Funcionales	36
4.4.3. Requerimientos de Seguridad	37
4.5. Reglas del negocio	38
4.5.1. Archivos	38
4.5.2. Nombre de Usuario	39

4.5.3.	Contraseña del Usuario	39
4.5.4.	Inicio de sesión	39
4.5.5.	Carpetas	40
4.6.	Umbrales de calidad y límite de errores	40
5.	Diseño	44
5.1.	Casos de Uso	44
5.2.	Documentación de los Casos de Uso	45
5.2.1.	CU1 Registrar Cuenta de Usuario	45
5.2.2.	CU2 Iniciar sesión	47
5.2.3.	CU3 Subir un archivo	48
5.2.4.	CU4 Descargar un archivo	50
5.2.5.	CU5 Enlistar Archivos y Carpetas	51
5.2.6.	CU6 Eliminar Archivo	51
5.2.7.	CU7 Eliminar Carpeta	52
5.2.8.	CU8 Mover Archivo	53
5.2.9.	CU9 Mover Carpeta	54
5.2.10.	CU10 Crear Carpeta	55
5.2.11.	CU11 Renombrar Archivo	57
5.2.12.	CU12 Renombrar Carpeta	58
5.2.13.	CU13 Copiar Archivo	59
5.2.14.	CU14 Modificar cuenta de usuario	60
5.2.15.	CU15 Eliminar cuenta	61
5.2.16.	CU16 Modificar contraseña	62
5.2.17.	CU17 Modificar Nombre de Usuario	64
5.2.18.	CU18 Recuperar cuenta	66
5.3.	Arquitectura del sistema	67
5.4.	Diseño de la base de datos	68
5.4.1.	Diagrama Entidad-Relación	68
5.4.2.	Modelo Relacional	68
5.5.	Diagrama de clases	69
5.6.	Diagramas de secuencia	70
5.6.1.	DS1 Registrar Cuenta de Usuario	70
5.6.2.	DS2 Iniciar sesión	71
5.6.3.	DS3 Subir un archivo	72
5.6.4.	DS4 Descargar un archivo	73
5.6.5.	DS6 Eliminar Archivo	74
5.6.6.	DS7 Eliminar Carpeta	75
5.6.7.	DS8 Mover Archivo	76
5.6.8.	DS9 Mover Carpeta	77
5.6.9.	DS10 Crear Carpeta	78
5.6.10.	DS11 Renombrar Archivo	79
5.6.11.	DS12 Renombrar Carpeta	80
5.6.12.	DS13 Copiar Archivo	81
5.6.13.	DS15 Eliminar cuenta	82
5.6.14.	DS16 Modificar contraseña	83
5.6.15.	DS17 Modificar Nombre de Usuario	84

5.6.16. DS18 Recuperar cuenta	84
5.7. Interfaz de usuario	85
5.7.1. PI1 Inicio de Sesión	85
5.7.2. PI2 Registro de Cuenta de Usuario	86
5.7.3. PI3 Recuperar contraseña	87
5.7.4. PP1 Pantalla Principal	88
5.7.5. PPC1 Renombrar Carpeta	88
5.7.6. PPC2 Renombrar Archivo	89
5.7.7. PPC3 Nueva Carpeta	89
5.7.8. PPC4 Copiar Archivo	89
5.7.9. PG1 Visualizar detalles de Usuario	90
5.7.10. PG2 Modificar Usuario	91
5.7.11. PMSG1 Mensaje de confirmación	91
5.7.12. PMSG2 Mensaje de información	91
5.8. Requisitos de diseño	92
5.8.1. Protocolos de seguridad	92
5.8.2. Cifrado y descifrado utilizando algoritmos de cifrado por bloques	92
5.8.3. Modos de operación	92
5.8.4. Generadores de número pseudoaleatorios	92
5.8.5. Funciones Hash	93
5.8.6. Otros requisitos	93
5.9. Superficie de ataques	93
5.10. Modelo de riesgos	94
5.10.1. Modelo de Amenazas STRIDE	94

Índice de figuras

2.1. Ejemplo de un sistema de cómputo en clúster [9].	6
2.2. Alto grado de heterogeneidad en cómputo en malla [10].	6
2.3. Esquema de criptografía asimétrica [12].	11
2.4. Modo de operación a) ECB, b) CBC, c) CFB y d)OFB [14].	15
3.1. Pseudocódigo del algoritmo Robust Computational Secret Sharing [5].	22
4.1. Diagrama de la metodología SDL [39].	35
5.1. Diagrama de Casos de Uso del Sistema.	44
5.2. Diagrama de la Arquitectura del sistema.	67
5.3. Diagrama Entidad-Relación del sistema.	68
5.4. Modelo Relacional de la base de datos del sistema.	68
5.5. Diagrama de clases del sistema.	69
5.6. Diagrama de secuencia para Registrar Cuenta de Usuario.	70
5.7. Diagrama de secuencia para Iniciar sesión.	71
5.8. Diagrama de secuencia para Subir un archivo.	72
5.9. Diagrama de secuencia para Descargar un archivo.	73
5.10. Diagrama de secuencia para Eliminar un Archivo.	74
5.11. Diagrama de secuencia para Eliminar una Carpeta.	75
5.12. Diagrama de secuencia para Mover un Archivo.	76
5.13. Diagrama de secuencia para Mover una Carpeta.	77
5.14. Diagrama de secuencia para Crear una Carpeta.	78
5.15. Diagrama de secuencia para Renombrar un Archivo.	79
5.16. Diagrama de secuencia para Renombrar una Carpeta.	80
5.17. Diagrama de secuencia para Copiar un Archivo.	81
5.18. Diagrama de secuencia para Eliminar cuenta.	82
5.19. Diagrama de secuencia para Modificar contraseña.	83
5.20. Diagrama de secuencia para Modificar Nombre de Usuario.	84
5.21. Diagrama de secuencia para Recuperar cuenta.	84
5.22. Pantalla de Inicio de sesión del sistema.	85
5.23. Pantalla de registro de una Cuenta de Usuario.	86
5.24. Pantalla de recuperación de la contraseña de Usuario.	87
5.25. Pantalla Principal del sistema.	88
5.26. Pantalla del Panel para Renombrar Carpeta.	88
5.27. Pantalla del Panel para Renombrar Archivo.	89
5.28. Pantalla del Panel para crear una nueva Carpeta.	89
5.29. Pantalla del Panel para ingresar un nombre al copiar un Archivo.	89

Índice de tablas

1.1.	Resumen comparativo de proveedores comerciales de almacenamiento.	2
2.1.	Funciones Hash más usadas [17].	17
3.1.	Tabla que muestra los x_i escogidos y los valores y_i obtenidos	19
3.2.	Puntos a repartir entre los 5 participantes	20
3.3.	Puntos a usar para la recuperación del secreto	20
4.1.	Popularidad de algunos lenguajes de programación [28].	26
4.2.	Herramientas de software a utilizar.	30
4.3.	Especificaciones de los equipos Samsung np3v4a.	30
4.4.	Especificaciones de los equipos Lenovo y510p	31
4.5.	Especificaciones de los equipos HP envi-4	31
4.6.	Relación de días laborales periodo de desarrollo	32
4.7.	Roles a desempeñar para el desarrollo del sistema	32
4.8.	Roles a desempeñar para el desarrollo del sistema.	33
4.9.	Servicios	33
4.10.	Costo del software	33
4.11.	Costo del Hardware	34
4.12.	Personal	34
4.13.	Umbrales de calidad y límite de errores (Servidor).	42
4.14.	Umbrales de calidad y límite de errores (Cliente).	43
5.1.	Detección y clasificación de las Amenazas por Gravedad y Probabilidad.	96

Capítulo 1

Introducción

Actualmente, los sistemas de almacenamiento de archivos constituyen un servicio cada vez más utilizado tanto por usuarios comunes como por grandes compañías, quienes los usan para compartir archivos o para almacenar copias de seguridad. El objetivo de estos sistemas es proveer soluciones efectivas para almacenar documentos personales (como son documentos de texto, fotos, música, archivos arbitrarios, etc.), permitiendo acceder a estos desde cualquier lugar[1]. Sin embargo, la información almacenada en estos sistemas puede ser susceptible a diversos ataques, por ejemplo, puede ser vista, modificada e inclusive destruida por entidades no autorizadas. Por lo tanto, un aspecto de suma importancia es garantizar la seguridad de la información almacenada en dichos sistemas.

Una forma de proteger la información almacenada en estos sistemas es emplear algún mecanismo de cifrado, los cuales generalmente utilizan una o varias llaves. Las llaves de cifrado son elementos primordiales para garantizar la seguridad de la información, por lo que debe hacerse un manejo adecuado de las mismas [2]. La gestión de las llaves incluye la generación, almacenamiento y distribución de las llaves dentro del sistema. De no administrar correctamente el almacenamiento de las llaves dentro del sistema, un atacante podría vulnerar el servidor donde éstas se almacenan, robarlas y acceder a la información que haya sido cifrada utilizando esas llaves.

Otro de los problemas a los que se enfrentan los sistemas de almacenamiento es la disponibilidad de la información, la cual se puede ver afectada por la corrupción de datos, ya sea por fallos en la transmisión de estos, o fallas internas en los servidores donde se almacena la información. Entre algunos ejemplos de estos incidentes[2], se encuentra la falta de disponibilidad del servicio de almacenamiento de archivos de Amazon S3, debido a la corrupción de datos que tuvo como origen un servidor balanceador de carga defectuoso, o un fallo en el esquema de autenticación del sistema de almacenamiento de archivos Dropbox, debido a una actualización mal implementada.

Actualmente, los principales proveedores comerciales de almacenamiento de archivos, tales como Dropbox, Google Drive y MEGA, utilizan el cifrador por bloques *Advanced Encryption Standard* (AES, por sus siglas en inglés), con claves de 128, 192 y 256 bits, para cifrar y descifrar los archivos subidos a sus plataformas. La transferencia de la información del usuario a los servidores se hace a través de una conexión *SSL*, pudiendo incluir algunos esquemas de seguridad extra como es la verificación en dos pasos, mediante el envío de códigos en mensaje

de texto, llamadas telefónicas o códigos de seguridad temporales únicos, siendo un ejemplo el servicio proporcionado por Dropbox[3] y por Google Drive[4].

Sistema	Servicios que ofrece	Características en seguridad
Dropbox	<ul style="list-style-type: none"> ■ 2 GB de almacenamiento gratuito, 1 TB versión de pago. ■ Máximo tamaño de archivo: 10 GB. ■ Máximo ancho de banda: 20 GB al día. ■ Soporta múltiples formatos de archivos. ■ Clientes para Sistemas Operativos Windows, OSX, Linux, y plataformas móviles. 	<ul style="list-style-type: none"> ■ Los archivos se cifran con AES de 256 bits. ■ Aplica el protocolo SSL y TLS entre las aplicaciones y servidores. ■ Túneles seguros protegido por AES de 128 bits o superior. ■ La verificación de dos pasos está disponible al iniciar sesión.
Google Drive	<ul style="list-style-type: none"> ■ 15 GB de espacio gratuito. ■ 100 GB - 30 TB de almacenamiento versión de pago. ■ Clientes para Sistemas Operativos: Windows, Android, OSX. 	<ul style="list-style-type: none"> ■ Uso de protocolo SSL/TLS. ■ Uso TLS de 256 bits para transmisión de mensajes con otros servidores de correo. ■ Para la etapa de intercambio de claves utiliza el algoritmo RSA con llaves de 2048 bits. ■ Las llaves privadas no se mantienen en almacenamiento persistente.
MEGA	<ul style="list-style-type: none"> ■ Cifrado del lado del cliente (AES). ■ Clientes para Sistemas Operativos: Windows, Linux, OSX, Android, Windows Phone, iOS, BlackBerry. ■ Plan gratuito: 50 GB de almacenamiento (10 GB ancho de banda). ■ Plan de Pago: Desde 500 GB de almacenamiento hasta 4TB. (Desde 1 TB ancho de banda hasta 8 TB). 	<ul style="list-style-type: none"> ■ Todas las operaciones criptográficas simétricas se basan en AES-128, en modo de operación CBC. ■ Cada archivo y cada nodo de carpeta utiliza su propia clave generada de forma aleatoria. ■ Cada cuenta de usuario tiene un par de claves de 2048 bits para recibir de forma segura datos, como claves compartidas o claves de archivos/carpetas.

Tabla 1.1: Resumen comparativo de proveedores comerciales de almacenamiento.

Algunos otros sistemas implementan un esquema de secreto compartido, entre ellos se encuentran el servicio de *IBM Cloud Object Storage* para almacenamiento en la nube, y *SPx™ Technology* ofrecido por Security First[5]. IBM Cloud Object Storage es un sistema de almacenamiento distribuido por IBM con un enfoque empresarial, cuyo objetivo es proveer flexibilidad para almacenar cualquier tipo de contenido. Este se distribuye como servicio en IBM Cloud a través de los servicios de Standard Cloud Object Storage, Vault Cloud Storage o bien en servidores dedicados en centros de datos de IBM Cloud[6].

Este servicio se basa en una tecnología llamada SecureSlice que combina cifrado, codificación, borrado y dispersión geográfica de datos. Esta tecnología divide los datos en diferentes segmentos, los expande y finalmente los codifica con piezas de datos redundantes. Estos fragmentos de datos se almacenan a través de diferentes ubicaciones geográficas o a través de

diferentes dispositivos[7]. El costo de este servicio no es dado a conocer por IBM de manera pública, sin embargo, es aproximadamente un 25 % más barato que el servicio de almacenamiento de Amazon Web Services S3[7].

SPxTM Technology, es un servicio desarrollado por la compañía Security First, para el almacenamiento de información en la nube. Este servicio utiliza el cifrador por bloques AES con claves de 256 bits, funciones de división de información de forma criptográfica (similares a la de Secreto compartido) y otras funciones patentadas por la compañía desarrolladora[8].

Las etapas que se realizan durante el procesamiento y almacenamiento de la información en este servicio son[8]:

- Cifrado de la información.
- Generación aleatoria de segmentos de información.
- Separación de la información a nivel de bits.
- Generación de redundancia para cada segmento de información.
- Autenticación.
- Dispersión de la información.

Solución propuesta

El presente Trabajo Terminal, busca ofrecer una alternativa para garantizar la seguridad de la información en un sistema de almacenamiento de archivos. El principal objetivo es desarrollar un sistema de almacenamiento de archivos distribuido capaz de proveer confidencialidad, confiabilidad, integridad, autenticación y disponibilidad a los datos almacenados en los diferentes nodos del sistema, a través del uso del esquema de compartición de secretos, conexiones seguras utilizando TLS, y un algoritmo de cifrado por bloques simétrico.

Los objetivos específicos de este trabajo son:

- Garantizar la disponibilidad de la información, así como la integridad de la misma, dentro de los umbrales del esquema del secreto compartido.
- Otorgar un servicio que brinde seguridad de la información, dentro de los umbrales del esquema del secreto compartido.
- Comprobar la funcionalidad del esquema de compartición de secretos.

Para hacerlo se implementará el protocolo criptográfico *Robust Computational Secret Sharing (RCSS)*, propuesto por Hugo Krawczyk en 1993 [5].

Dicho protocolo, está conformado a su vez, por distintas primitivas criptográficas tales como cifradores por bloque, funciones hash, y un esquema de compartición de secretos, los cuales se describirán en los siguientes capítulos.

Al utilizarlo el protocolo RCSS, será posible brindar las siguientes características:

- Confidencialidad: Evitar que la información esté disponible o sea revelada a individuos, entidades o procesos no autorizados.
- Integridad: Mantener y garantizar la exactitud de los datos a través de todo el proceso de tal forma que no se pueden modificar de forma no autorizada o no detectada.

- Autenticación: Garantizar que cada entidad en una comunicación es quien dice ser.

Adicionalmente, el sistema será capaz de proveer tolerancia a fallos, dado que, en caso de que algún servidor de almacenamiento sufriera de algún fallo o la información se corrompiera durante su transmisión, este esquema permite reconstruir y descifrar el archivo original a partir de las partes restantes.

Al término de la realización de este Trabajo Terminal, se espera tener en completo funcionamiento el Sistema de Almacenamiento Seguro de Archivos, proporcionando los servicios enunciados en el objetivo (confidencialidad, integridad, autenticación y disponibilidad de los datos, dentro de los umbrales del esquema del secreto compartido) y haciendo uso de las tecnologías seleccionadas para la realización de éste. Comprobando así la funcionalidad práctica de este esquema.

Además de ello, se espera proporcionar la documentación del sistema, como es el manual de usuario, el reporte técnico y las pruebas realizadas al sistema, las cuales estarán apegadas a los lineamientos de la metodología SDL (*Microsoft Security Development Lifecycle*).

Entre las pruebas a llevarse a cabo sobre sistema se encuentran:

- Pruebas de Análisis dinámico del sistema: Consiste en pruebas en tiempo de ejecución para buscar problemas críticos de seguridad, como son fugas de memoria y escalado de privilegios.
- Pruebas de exploración de vulnerabilidades mediante datos aleatorios: Consiste en introducir datos aleatorios o con formatos erróneos en el sistema.
- Pruebas de detección y corrección de errores cuando algún fragmento de información proporcionado por el servidor de almacenamiento es erróneo o inconsistente.
- Pruebas sobre los algoritmos de cifrados, para verificar su correcta funcionalidad, con vectores aprobados por el *National Institute of Standards and Technology* (NIST).

Organización del documento

El resto del presente documento está organizado de la siguiente manera: En el capítulo 2 se describen conceptos tanto de sistemas distribuidos como de criptografía, los cuales son necesarios para comprender el trabajo desarrollado en este trabajo terminal. El capítulo 3, describe en detalle el protocolo RCSS, así como sus distintos componentes. Finalmente, los capítulos 4 y 5, contienen el análisis y diseño del sistema que se desarrollará.

Capítulo 2

Preliminares

En este capítulo se hace mención de conceptos relacionados con Sistemas Distribuidos y con Criptografía, los cuales son necesarios para la comprensión de este Trabajo Terminal. En ambos casos, se mencionan nociones básicas y conceptos específicos aplicados al sistema a desarrollarse.

2.1. Sistemas distribuidos

Existen diferentes definiciones sobre qué es un sistema distribuido. Tanembaum [9] lo define como una colección de computadoras independientes que dan al usuario la impresión de constituir un único sistema coherente.

Coloudis [10] define un sistema distribuido como aquel en el que los componentes localizados en computadores, conectados en red, comunican y coordinan sus acciones únicamente mediante el uso de mensajes.

Esta definición comprende diversos aspectos importantes. El sistema distribuido consta de componentes autónomos además de que los usuarios creen que realmente interactúan con un sistema único, esto significa que dichos componentes deben colaborar entre sí. Dicha forma de colaboración radica en el fondo del desarrollo de los sistemas distribuidos.

2.1.1. Tipos de sistemas distribuidos.

A continuación, se señalan las diferencias entre los distintos tipos de sistemas distribuidos de cómputo, sistemas distribuidos de información, y sistemas distribuidos embebidos.

Sistemas distribuidos de cómputo

Una clase importante de sistemas distribuidos es la utilizada para realizar tareas de cómputo de alto rendimiento. Se puede hacer una distinción entre dos subgrupos.

- **Cómputo en *clúster*:** Es homogéneo, es decir, el hardware subyacente consta de una colección de estaciones de trabajo similares, o computadoras personales, conectadas cercanamente por medio de una red de área local de alta velocidad. Además, cada nodo

ejecuta el mismo sistema operativo. Se utiliza para la programación en paralelo donde un solo programa (de cálculo intenso) corre paralelamente en múltiples máquinas.

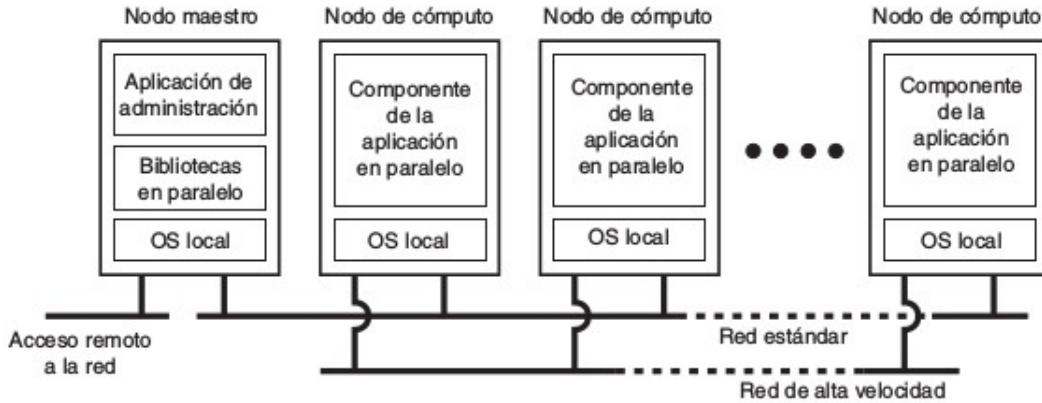


Figura 2.1: Ejemplo de un sistema de cómputo en clúster [9].

- **Cómputo en malla (grid):** Tiene un alto grado de heterogeneidad respecto al hardware, sistemas operativos, redes, dominios administrativos, políticas de seguridad, etcétera. Este subgrupo consta de sistemas distribuidos construidos generalmente como un conjunto de sistemas de cómputo, en donde cada sistema podría caer dentro de un dominio administrativo diferente. Una cuestión clave en un sistema de cómputo en grid es reunir los recursos de diferentes organizaciones para permitir la colaboración de un grupo de personas o instituciones. Tal colaboración se realiza en la forma de una organización virtual. La gente que pertenece a la misma organización virtual tiene derechos de acceso a los recursos que proporciona la organización.

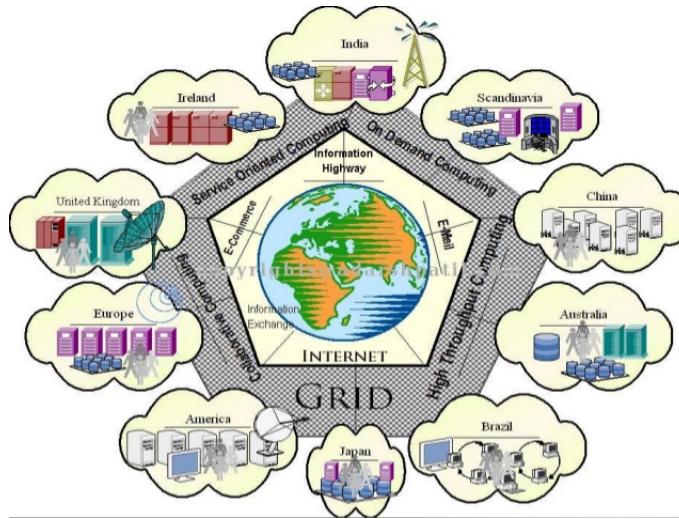


Figura 2.2: Alto grado de heterogeneidad en cómputo en malla [10].

Sistemas distribuidos de información

Son sistemas organizacionales y corporativos, los cuales implican la conjunción de aplicaciones que inter-operan gracias a una red. Inicialmente los sistemas constaban de un equipo

que ejecutaba un servidor (con frecuencia servidor de base de datos) y de programas remotos llamados clientes. Dichos clientes son capaces de enviar peticiones y recibir respuesta de servidor. La integración a nivel más bajo y sencillo se realizaba registrando en los programas cliente cierto número de peticiones dirigidas a distintos servidores y dentro de una petición más grande ejecutarla como una transacción distribuida. Lo anterior a generado una industria de sistemas que se concentra en la integración de sistemas empresariales.

Sistemas distribuidos masivos

Sistemas distribuidos en los cuales la inestabilidad es el comportamiento predeterminado. Son con frecuencia dispositivos que se caracterizan por ser pequeños, de baterías, portátiles, y tienen una conexión inalámbrica. Tiene carencia general de control administrativo humano.

Los dispositivos son configurados para descubrir automáticamente su ambiente y “adapitarse” de la mejor manera posible cumpliendo con los tres siguientes requerimientos para aplicaciones móviles:

- Incluir cambios contextuales.
- Fomentar composiciones a la medida.
- Reconocer el intercambio como algo común.

2.1.2. Propiedades de los sistemas distribuidos.

1. **Heterogeneidad:** Internet permite que los usuarios accedan a servicios y ejecuten aplicaciones sobre un conjunto heterogéneo de redes y computadoras aplicable a los siguientes elementos:

- Redes.
- Hardware de computadoras
- Sistemas operativos
- Lenguajes de programación
- Implementaciones de diferentes desarrolladores

2. **Extensibilidad:** Es la característica que determina si el sistema puede ser extendido y reimplementado en diversos aspectos. Determinada por el grado en el cual se pueden añadir nuevos servicios de compartición de recursos y ponerlos a disposición para el uso por una variedad de programas cliente. A dichos sistemas se les denomina *Sistemas distribuidos abiertos* los cuales pueden ser extensibles en nivel de hardware y software.

3. **Seguridad:** La seguridad de los recursos de información tiene tres componentes:

- Confidencialidad: Protección contra el descubrimiento por individuos no autorizados.
- Integridad: Protección contra la alteración o corrupción
- Disponibilidad: Protección contra interferencia con los procedimientos de acceso a recursos.

4. **Escalabilidad:** Es la propiedad deseable de un sistema, una red o un proceso, que indica su habilidad para reaccionar y adaptarse sin perder calidad, o bien manejar el crecimiento continuo de trabajo de manera fluida, o bien para estar preparado para hacerse más grande sin perder calidad en los servicios ofrecidos. Se dice que un sistema es *escalable* si conserva su efectividad cuando ocurre un incremento significativo en el número de recursos y el número de usuarios.

El diseño de los sistemas distribuidos escalables presenta los siguientes retos:

- *Control del coste de los recursos físicos.*
- *Control de las pérdidas de prestaciones.*
- *Prevención de desbordamiento de recursos de software.*
- *Evitación de cuellos de botella de prestaciones.*

5. **Tratamiento de fallos:** Los sistemas computacionales a veces fallan. Cuando aparecen fallos en el hardware o el software, los programas pueden producir resultados incorrectos o pudieran parar antes de haber completado el cálculo pedido. Estas son algunas técnicas para tratar fallos:

- *Deteccción de fallos:* Algunos fallos son detectables (por ejemplo, el uso de funciones hash para detectar datos corruptos en mensajes), en otros casos no es posible detectarlos. El reto está en revolver fallos que no pueden detectarse, pero si pueden esperarse.
- *Enmascaramiento de fallos:* Algunos fallos que han sido detectados pueden ocurrir o atenuarse, por ejemplo: Los mensajes pueden retransmitirse cuando falla la recepción.
- *Tolerancia de fallos:* Existen fallos en lo que es posible que no sea práctico detectarlos y ocultarlos. Los clientes pueden diseñarse para tolerar ciertos fallos, lo que implica que los usuarios tendrán que tolerarlos generalmente.
- *Recuperación frente a fallos:* La recuperación implica el diseño de software en el que, tras una caída del servidor, el estado de los datos pueda reponerse o retractarse (roll-back) a una situación anterior.
- *Redundancia:* Puede lograrse que los servicios toleren fallos mediante el empleo redundante de componentes.

6. **Concurrencia:** Para que un objeto sea seguro en un entorno concurrente, sus operaciones deben sincronizarse de forma que sus datos permanezcan consistentes. Esto puede lograrse mediante diversas técnicas.

7. **Transparencia:** Se define como transparencia como la ocultación al usuario y al programador de aplicaciones de la separación de los componentes en un sistema distribuido, de forma que se perciba el sistema como un todo más que como una colección de componentes independientes.

El Modelo de Referencia para el Procesamiento Distribuido Abierto (RM-ODP: *Reference Model for Open Distributed Processing*) de la Organización Internacional de

Estándares [ISO/IEC 10026-1:1992] identifica ocho formas de transparencia:

- a) *Transparencia de acceso*: Permite acceder a los recursos local y remotos empleando operaciones idénticas.
- b) *Transparencia de ubicación*: Permite acceder a los recursos sin conocer su localización.
- c) *Transparencia de concurrencia*: Permite que varios procesos operen concurrentemente sobre recursos compartidos sin interferencia mutua.
- d) *Transparencia de replicación*: Permite utilizar múltiples ejemplares de cada recurso para aumentar la fiabilidad y las prestaciones sin que los usuarios y los programadores de aplicaciones necesiten de su conocimiento.
- e) *Transparencia frente a fallos*: Permite ocultar fallos, dejando que los usuarios y programadores de aplicación completen sus tareas a pesar de fallos del hardware o de los componentes software.
- f) *Transparencia de movilidad*: Permite la reubicación de recursos y clientes en un sistema sin afectar la operación de los usuarios y los programas.
- g) *Transparencia de prestaciones*: Permite reconfigurar el sistema para mejorar las prestaciones según varía su carga.
- h) *Transparencia al escalado*: Permite al sistema y a las aplicaciones expandirse en tamaño sin cambiar la estructura del sistema o los algoritmos de aplicación.

2.2. Criptografía

La Criptología (del griego krypto: 'oculto' y logos: 'palabra') es, tradicionalmente, la disciplina científica que se dedica al estudio de la escritura secreta, es decir, estudia los mensajes que, procesados de cierta manera, se convierten en difíciles o imposibles de leer por entidades no autorizadas [11].

Disciplinas que comprende:

- **Criptografía**: Se ocupa del estudio de los algoritmos, protocolos y sistemas que se utilizan para proteger la información y dotar de seguridad a las comunicaciones y a las entidades que se comunican.
- **Criptoanálisis**: Se ocupa de obtener el significado de mensajes construidos mediante Criptografía sin tener autorización para ello. Podríamos decir que el criptoanálisis tiene un objetivo opuesto al de la criptografía. Su objetivo es buscar el punto débil de las técnicas criptográficas para explotarla y así reducir o eliminar la seguridad que teóricamente aportaba esa técnica criptográfica.

La criptografía, del griego “criptos” (oculto o secreto) y “grafe” (escritura), es el estudio de técnicas para realizar comunicaciones seguras en presencia de terceras entidades, conocidas como adversarios.

De manera más general, la criptografía consiste en la creación y análisis de algoritmos, protocolos y sistemas que se utilizan para proteger la información privada y dotar de seguridad a las comunicaciones y a las entidades implicadas.

La criptografía no es el único medio de proporcionar seguridad a la información, sino más bien un conjunto de técnicas para lograrlo.

Los servicios que la criptografía proporciona son los siguientes:

- Privacidad o Confidencialidad: Evitar que la información esté disponible o sea revelada a individuos, entidades o procesos no autorizados.
- Integridad de datos: Mantener y garantizar la exactitud e integridad de los datos a través de todo el proceso de tal forma que los datos no se pueden modificar de forma no autorizada o no detectada. La manipulación de datos incluye acciones como inserción, eliminación y sustitución de información.
- Autenticación: Garantizar que cada entidad en una comunicación es quien dice ser. Para establecer una comunicación entre dos entidades, ambas deben de poder identificarse entre sí.
- No repudio: Evitar que alguna entidad o individuo niegue que realizó una acción, como puede ser el envío de un mensaje. Deben de existir mecanismos para resolver controversias en caso de que alguna de las entidades participantes niegue determinadas acciones.

2.2.1. Ataques Criptográficos

Ataques a esquemas de cifrado

El objetivo de estos tipos de ataque es recuperar texto claro a partir del texto cifrado, o inclusive deducir la llave del cifrado. Este tipo de ataques también se consideran ataques de tipo pasivo.

- Ataque de sólo texto cifrado conocido: En este tipo de ataque, el adversario intenta deducir la llave de cifrado o el texto en claro solamente a partir del texto cifrado.
- Ataque de texto en claro conocido: Es aquel donde el adversario tiene acceso a alguna porción del texto en claro y su correspondiente texto cifrado.
- Ataque de texto en claro selecto: En este caso, el adversario elige textos en claro al azar y obtiene sus correspondientes textos cifrados, posteriormente, el adversario utiliza cualquier información deducida a partir de ellos para recuperar el texto plano correspondiente a un texto cifrado no analizado con anterioridad.
- Ataque de texto en claro selecto adaptativo: A diferencia del anterior, el adversario elige el texto plano dependiendo del texto cifrado recibido de anteriores peticiones.
- Ataque de texto cifrado selecto: Es aquel donde el atacante elige un texto cifrado al azar y posteriormente obtiene el texto claro.
- Ataque de texto cifrado selecto adaptativo: De igual forma que el anterior, el adversario puede elegir los textos cifrados con base en textos en claro obtenidos previamente.

2.3. Criptografía asimétrica

Se dice que un método de cifrado es asimétrico (o de llave pública) cuando existe un par de claves (e, d) utilizadas en los procesos de cifrado/descifrado, en este caso, la llave e es de acceso público, mientras que la otra llave d es mantenida en secreto. Para que el esquema se considere seguro, debe ser computacionalmente inviable calcular d a partir de e .

Este esquema la llave pública define una transformación de cifrado E_e , mientras que la llave privada define la transformación de descifrado asociada D_d . Una entidad B que desee enviar un mensaje m a la entidad A debe obtener una copia de la llave pública de A , usar la transformación de cifrado para obtener el mensaje cifrado $c = E_e(m)$ y enviarlo a A . De tal forma que la entidad A para descifrar c aplica la transformación de descifrado de la forma $m = D_d(c)$ para obtener el mensaje original.

Los objetivos de la criptografía asimétrica son proveer confidencialidad, integridad de los datos, autenticación y no repudio. Para proporcionar las características de autenticación y no repudio, la criptografía asimétrica hace uso de las firmas digitales.

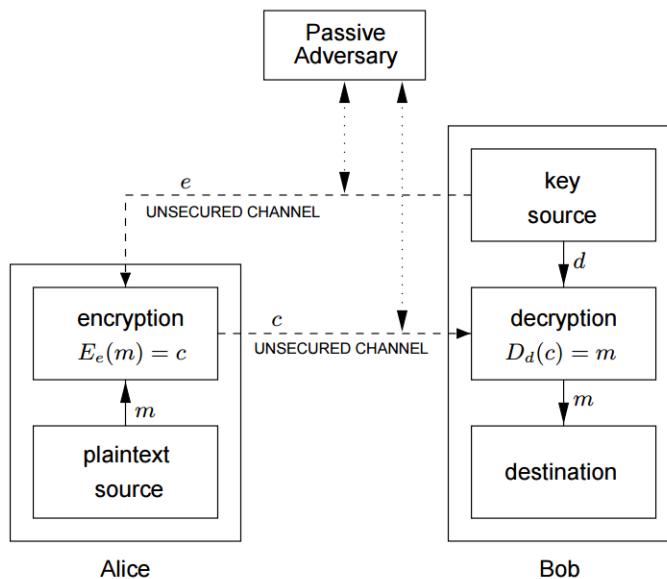


Figura 2.3: Esquema de criptografía asimétrica [12].

2.4. Criptografía simétrica

El cifrado de clave secreta también se le denomina como cifrado simétrico, porque se utiliza la misma clave para cifrar y descifrar el mensaje.

Los sistemas de cifrado de clave secreta utilizan una sola clave que comparten el remitente y el destinatario. Ambos deben poseer la misma clave; el remitente cifra el mensaje mediante la clave y el destinatario descifra el mensaje con la misma clave. Para poder establecer una comunicación privada, tanto el remitente como el destinatario deben mantener la clave en

secreto [13].

Este tipo de cifrado tiene características que lo hacen inadecuado para su uso general:

- El cifrado de clave secreta requiere una clave para cada par de personas que necesitan comunicarse de forma privada. El número necesario de claves aumenta considerablemente a medida que se incrementa el número de participantes.
- Las claves se deben de compartir por pares de comunicadores, por lo que las claves se deberán distribuir a los participantes. La necesidad de transmitir claves secretas las hace vulnerables al robo.
- Los participantes sólo pueden comunicarse mediante un acuerdo previo. No puede enviar un mensaje cifrado utilizable a alguien de forma espontánea. Tanto una como la otra persona deben establecer acuerdos para comunicarse compartiendo claves.

2.4.1. Cifrado por bloques

Definición 2.4.1. *Un cifrado por bloques puede ser visto como una función $E : K \times M \rightarrow C$, donde $K = \{0, 1\}^k$ y $M = C = \{0, 1\}^n$. Por lo tanto, K , M y C son conjuntos finitos no vacíos de cadenas de bits, los cuales son llamados el espacio de llaves, el espacio de mensaje y el espacio del cifrado, respectivamente. Los parámetros n y k son llamados la longitud del bloque y la longitud de la llave, respectivamente[14].*

Un cifrado por bloques tiene como entrada un mensaje de n -bits (también llamada *texto en claro*) y una llave de k -bits, produciendo un texto cifrado de longitud n -bits.

Para un texto en claro y un texto cifrado de n -bits bajo una llave fija K , la función de cifrado es una biyección, definiendo así una permutación de cadenas de n -bits. Cada llave define una biyección diferente.

Al mapeo inverso de la función anterior se le denomina *función de descifrado* o $D_K(C)$. Donde $C = E_K(M)$ es el texto cifrado resultante de aplicar la función de cifrado bajo K al texto en claro P .

DES

El *Data Encryption Standard* (DES) fue desarrollado en 1975 por IBM como resultado de una convocatoria realizada por el National Institute of Standards and Technology (NIST). Se encuentra definido en el estándar estadounidense FIPS 46-2 [12].

DES está basado en una red Feistel, este cifrado por bloques consiste en un numero de rondas, donde cada ronda contiene corrimientos de bits, sustituciones no lineales (S-Box) y operaciones lógicas (XOR).

Como los demás esquemas de cifrado, DES recibe dos entradas: El texto en claro y la llave secreta. Los bloques de cifrado son de 64 bits mientras que la llave es de 56 bits. Éste consta de 16 rondas en las cuales se efectúan diversas operaciones sobre el texto en claro y la llave con la cual se cifra dicho texto.

Descripción del esquema de cifrado

Operaciones sobre los datos

- $IP(X) = L_0 R_0$
- $L_i = R_{i-1}$
- $R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$
- $y = IP^{-1}(R_{16} L_{16})$

Donde f se describe de la siguiente manera:

- Se hace una expansión de R_{i-1} mediante el uso de la E-BOX la cual usa los 32 bits de R_{i-1} para hacer crecer su tamaño hasta 48 bits.
- Una vez hecho esto se hace \oplus entre el resultado de la E-BOX con la sub-llave K_i
- Posteriormente en bloques de 6 bits se hace uso de las S-BOX donde en cada una dan como resultado 4 bits lo cual hace volver a la longitud original de 32 bits de R_{i-1}
- Por último se pasa este resultado a través de la caja de permutación P-BOX.

Generación de llaves

- $PC - 1$
- $C_i = LS_i(C_{i-1})$
- $D_i = LS_i(D_{i-1})$
- $K_i = PC - 2(C_i D_i)$

AES

El cifrado por bloques AES es un algoritmo de cifrado simétrico desarrollado por los estudiantes Vincent Rijmen y Joan Daemen de la Katholieke Universiteit Leuven en Bélgica, bajo el nombre “Rijndael” fue presentado en 1997 al concurso organizado por el Instituto Nacional de Normas y Tecnologías (NIST) para elegir el mejor algoritmo de cifrado; éste algoritmo ganó el concurso transformándose en un estándar en el año 2002, con algunos cambios fue posteriormente renombrado a AES (*Advanced Encryption Standard*) y se convirtió en uno de los algoritmos más utilizados en la actualidad.

En el año 2003, el gobierno de los Estados Unidos anunció que el algoritmo era lo suficientemente seguro y que podía ser usado para protección nacional de información. Hasta el momento no se conocen ataques eficientes, los únicos conocidos son los denominados ataques de canal auxiliar.

El tamaño del bloque es de 128 bits (16 bytes) los cuales se ven como una matriz de 4x4 llamada estado.

La longitud de la clave puede ser de 128, 192 y 256, lo cual permite 3 implementaciones AES-128, AES-192 y AES-256.

Descripción del esquema de cifrado

El proceso de cifrado del algoritmo consiste en aplicar a cada estado un conjunto de operaciones agrupadas en lo que se denominan rondas, el algoritmo realiza 11 rondas, donde en cada ronda se aplica una sub-clave diferente.

Las operaciones realizadas en cada ronda son las siguientes:

- SubBytes: Cada byte del estado se reemplaza por otro valor de acuerdo con la tabla de sustitución de bytes S-Box del cálculo de las sub-claves.
- ShiftRows: En cada fila del estado, a excepción de la primera, se rotan circularmente hacia la izquierda los bytes, en la segunda fila se rotan una posición, en la tercera dos posiciones y en la cuarta tres posiciones.
- MixColumns: A cada columna del estado se le aplica una transformación lineal, esto es multiplicarlo por una matriz predeterminada en el campo GF.
- AddRoundKey: Se aplica la misma operación que en la ronda inicial pero utilizando otra subclave.

Estas operaciones son realizadas en las 9 rondas intermedias, mientras que en la ronda inicial se realiza la operación *AddRoundKey*, y para la última ronda a diferencia de las otras 9 rondas no se lleva a cabo la operación de *MixColumns*.

2.4.2. Modos de operación

Un cifrado a bloques cifra un texto claro en bloques de tamaño fijo de n -bits. Para mensajes que exceden la longitud de n -bits la solución más sencilla es dividir el mensaje en bloques de n -bits y cifrar cada uno de forma separada. Este modo de operación (denominado ECB) presenta diferentes desventajas, ya que no oculta patrones que pueden presentarse en la información, dado que dos bloques de datos idénticos implican bloques de texto en claro idénticos.

Esta desventaja motivó el desarrollo de otros modos de operación para mensajes de longitud mayor a n . Estos modos de operación son: ECB, CBC, CFB y OFB [12].

Electronic CodeBook (ECB)

Cada bloque de texto plano y texto cifrado es cifrado y descifrado por separado. En otras palabras, el cifrado y descifrado de cada bloque es totalmente independiente de otros bloques. Este modo de operación es utilizado principalmente para la transmisión segura de valores únicos.

Cipher Block Chaining (CBC)

Se introduce la retroalimentación. Antes de que cada bloque de texto en claro se cifre, se combina con el texto cifrado del bloque anterior mediante una operación XOR bit a bit. Esto garantiza que, incluso si el texto sin formato contiene muchos bloques idénticos, cada uno cifrará a un bloque de texto cifrado de forma diferente. El vector de inicialización se combina con el primer bloque de texto en claro mediante una operación XOR antes de que el bloque se cifre. Si un solo bit del bloque de texto cifrado es mutilado, el bloque de texto plano

correspondiente también será mutilado. Además, un pedazo en el bloque subsecuente, en la misma posición que el pedazo original mutilado, será mutilado. Este modo de operación es utilizado principalmente para la transmisión orientada a bloques y para la autenticación [15].

Cipher FeedBack (CFB)

El modo CFB opera en segmentos en lugar de bloques. La longitud del segmento (llamada S) está entre un bit y el tamaño del bloque (llamado b) para el algoritmo subyacente (DES o AES). Cada paso de cifrado toma un bloque de entrada, lo cifra con la llave proporcionada para generar un bloque de salida, toma los S bits más significativos del bloque de salida, y luego realiza la operación XOR con el segmento de texto claro. El primer bloque de entrada es el vector de inicialización IV y cada bloque de entrada subsiguiente se forma concatenando los bits menos significativos ($b - S$) del bloque de entrada anterior y el texto cifrado (s bits) del paso anterior para formar un bloque completo. El texto de entrada puede ser de cualquier longitud. El texto de salida tendrá la misma longitud que el texto de entrada [15].

Output Feedback (OFB)

El modo OFB utiliza un vector inicial (IV) en su procesamiento. Este modo requiere que el IV sea único (el IV debe ser único para cada ejecución del modo bajo la clave dada). Cada paso de cifrado toma un bloque de entrada, lo cifra con la clave proporcionada para generar un bloque de salida, para posteriormente realizar XOR con el bloque de texto claro. El primer bloque de entrada es el IV y cada bloque de entrada subsiguiente es el bloque de salida anterior. El texto de entrada puede ser de cualquier longitud. El texto de salida tendrá la misma longitud que el texto de entrada [15].

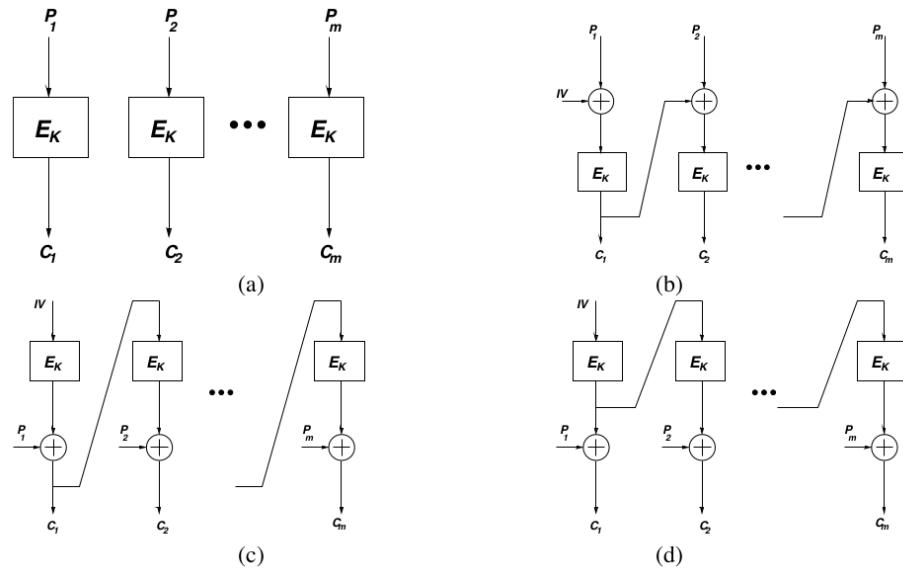


Figura 2.4: Modo de operación a) ECB, b) CBC, c) CFB y d) OFB [14].

2.5. Funciones Hash

Una función de hash es una función computacionalmente eficiente que asigna cadenas binarias de longitud arbitraria a cadenas binarias de cierta longitud fija, denominadas ***hash-value***.

La idea básica es que un ***hash-value*** sirva como una representación compacta de una cadena de entrada. Para ser de uso criptográfico, una función hash h se elige de tal manera que sea imposible encontrar dos entradas distintas que tengan un valor hash común.

Las funciones hash pueden ser utilizadas para la integridad de datos tal como se describe a continuación:

- El ***hash-value*** correspondiente a una entrada particular se calcula en algún momento.
- La integridad de este ***hash-value*** es protegido de alguna manera.
- En un momento posterior, para verificar que los datos de entrada no han sido alterados, el ***hash-value*** es recalculado usando la entrada y comparando que el valor sea igual al hash original.

2.5.1. Propiedades Básicas

Para que resulte útil la autenticación de mensajes, una función Hash H debe poseer las siguientes propiedades: [16]

1. H puede aplicarse a un bloque de datos de cualquier tamaño.
2. H produce una salida de tamaño fijo.
3. $H(x)$ es relativamente fácil de computar para cualquier x .
4. Para cualquier valor h dado, es imposible desde el punto de vista computacional encontrar x tal que $H(x) = h$, lo cual, se conoce como propiedad **unidireccional**.
5. Para cualquier bloque dado x , es imposible desde el punto de vista computacional encontrar $y \neq x$ con $H(y) = H(x)$, lo que se conoce como **resistencia débil a la colisión**.
6. Es imposible desde el punto de vista computacional encontrar un par (x, y) tal que $H(x) = H(y)$, lo que se conoce como **resistencia fuerte a la colisión**

2.5.2. Resistencia a Ataques Criptoanalíticos

Se definen cinco tipos de resistencias a los ataques criptoanalíticos contra las funciones Hash:

1. **Resistencia a Preimágenes:** Una función Hash es resistente a preimágenes si dado el ***hash-value*** z es computacionalmente difícil o imposible hallar algún mensaje x tal que $h(x) = z$.

2. **Resistencia a segundas Preimágenes:** Una función Hash es resistente a segundas preimágenes si dado un mensaje x es computacionalmente imposible hallar un mensaje $x' \neq x$ tal que $h(x) = h(x')$.
3. **Resistente a Colisiones:** Una función Hash es resistente a colisiones si es computacionalmente difícil imposible hallar un par de mensajes distintos x, x' tal que $h(x) = h(x')$. Existen dos variantes con respecto al uso de IV (Vectores de Inicialización), las cuales son con IV's fijos o variables, en este último caso uno de estos vectores puede variar libremente.
4. **Resistente a Seudo-colisiones:** Una función Hash es resistente a seudo-colisiones si es computacionalmente difícil o imposible hallar un par de mensajes distintos x, x' tal que $h(x) = h(x')$ en las cuales se puedan usar distintos IV's para ambos mensajes.
5. **Resistente a Seudo-Preimágenes:** Una función de hashing es resistente a seudo-preimagenes si dado el *hash-value* es computacionalmente difícil o imposible hallar algún mensaje x y algún IV, tal que $h(x) = z$

Dichas formas de resistencia no son equivalentes, lo cual implica que la existencia de la condición 2 no garantiza que se cumpla la condición 1.

2.5.3. Principales Funciones Hash

Función	# Bits	# Rounds	Fuerza Pre-imágen	Fuerza contra Colisión
MD4	128	48	2^{128}	2^{20}
MD5	128	64	2^{128}	2^{64}
RIPEMD128	128	64	2^{128}	2^{64}
SHA-0	160	80	2^{160}	2^{80}
SHA-1	160	80	2^{160}	2^{80}
RIPEMD160	160	80	2^{160}	2^{80}
SHA-224	224	64	2^{224}	2^{112}
SHA-256	256	64	2^{56}	2^{128}
SHA-384	384	80	2^{384}	2^{192}
SHA-512	512	80	2^{512}	2^{256}

Tabla 2.1: Funciones Hash más usadas [17].

Capítulo 3

Esquemas de secreto compartido

Un esquema para compartir secretos es un protocolo criptográfico en el que, como su nombre indica, se divide un determinado secreto en fragmentos que se reparten entre los participantes del esquema.

Se divide un secreto S en n partes S_1, \dots, S_n tal que:

1. El conocimiento de m o más piezas S_i hace a S eficiente.
2. El conocimiento de $m - 1$ o menos S_i piezas deja a S completamente indeterminado.
Esta combinación se denomina umbral (m, n) .

El esquema de Secreto compartido fue inventado de forma independiente por Adi Shamir en 1979 [18] y George Blakley [19].

3.1. Esquema de umbral de Shamir

3.1.1. Proceso de inicialización y compartición de las partes

Sea un secreto $S \in \mathbb{N}$ y $S \in \mathbb{Z}_p$, se planea dividir dicho secreto en n participantes: $S_1, S_2, S_3, \dots, S_n$.

Para lograr esto

1. Se define un umbral t tal que $t \leq n$, este umbral permite establecer con qué cantidad de partes es posible reconstruir el secreto S de manera que contando con $t - 1$ partes no es posible reconstruirlo.
2. Se escoge una función polinomial de grado $t - 1$, es decir: $f(x) = \sum_{i=0}^{t-1} a_i x^i = a_0 + a_1 x + a_2 x^2 \dots + a_{t-1} x^{t-1}$ (mód p) donde: $a_0 = S$, $a_k \in \mathbb{Z}_p$ donde $1 \leq k \leq t - 1$
3. Una vez construido dicho polinomio, se escogen de forma aleatoria $x_i \in \mathbb{Z}_p - \{0\}$ con $1 \leq i \leq n$.
4. Teniendo los x_i se evalúan en la función polinomial $y_i = f(x_i)$ obteniendo los pares ordenados $p(x_i, f(x_i))$ o mejor dicho $p(x_i, y_i)$.
5. Se distribuyen los pares ordenados $p(x_i, y_i)$ entre los n participantes.

3.1.2. Proceso de recuperación

1. Sean $P_0(x_0, y_0), P_1(x_1, y_1), \dots, P_n(x_n, y_n)$ el conjunto de partes entregadas a los n participantes.
2. Contando con t puntos, se puede reconstruir la función polinomial haciendo uso de la interpolación de Lagrange, es decir que $f(x)$ se puede expresar de la siguiente manera:

$$f(x) = \sum_{j=1}^t y_j l_j \text{ donde } l_j = \prod_{1 \leq k \leq t, k \neq j} \frac{x - x_k}{x_j - x_k} \pmod{p}$$

3. Al realizar las operaciones algebraicas de los $y_j l_j \pmod{p}$ y reducir términos se obtendrá la función polinomial de la forma: $f(x) = \sum_{i=0}^{t-1} a_i x^i = a_0 x + a_1 x + a_2 x^2 \dots + a_{t-1} x^{t-1} \pmod{p}$ recuperando así el término $a_0 = S$ y finalmente obteniendo el secreto S .

3.1.3. Ejemplo

Proceso de compartición

Se requiere compartir el secreto $S = 7$ en un conjunto de 5 participantes de tal manera que con al menos 3 de ellos se pueda recuperar el secreto.

1. Sea el secreto $S = 7$ y un conjunto de participantes $n = 5$
2. Se define el conjunto de primos: $P = 17 \implies \mathbb{Z}_p = \mathbb{Z}_{17} = \{0, 1, 3, 5, 7, 9, 11, 13\}$
3. Ya que sólo se necesitan 3 participantes para recuperar el secreto, el umbral será $t = 3$.
4. Definimos la función polinomial con $t = 3$, $a_0 = S = 7 \implies a_0 = 7, a_k \in \mathbb{Z}_p$
 $1 \leq k \leq 2$, elegimos $a_1 = 1, a_2 = 3$ quedando de la siguiente forma: $f(x) = \sum_{i=0}^{3-1} a_i x^i = a_0 + a_1 x + a_2 x^2 = 7 + x + 3x^2 \pmod{17}$
5. Una vez construido dicho polinomio, se escogen de forma aleatoria $x_i \in \mathbb{Z}_{17} - \{0\}$ con $1 \leq i \leq 5$ y se evalúan en la función polinomial obteniendo la siguiente tabla:

i	x_i	$y_i = f(x_i) \pmod{p}$
1	1	$3(1)^2 + (1) + 7 = 3 + 1 + 7 = 11$
2	2	$3(2)^2 + (2) + 7 = 12 + 2 + 7 = 21 = 4$
3	3	$3(3)^2 + (3) + 7 = 27 + 3 + 7 = 37 = 3$
4	4	$3(4)^2 + (4) + 7 = 48 + 4 + 7 = 59 = 8$
5	5	$3(5)^2 + (5) + 7 = 75 + 5 + 7 = 87 = 2$

Tabla 3.1: Tabla que muestra los x_i escogidos y los valores y_i obtenidos

6. Ahora que se han obtenido los pares ordenados $p(x_i, y_i)$, se distribuyen entre los 5 participantes.

Puntos a repartir
$p_1(1, 11)$
$p_2(2, 4)$
$p_3(3, 3)$
$p_4(4, 8)$
$p_5(5, 2)$

Tabla 3.2: Puntos a repartir entre los 5 participantes

Proceso de recuperación

1. Cada uno de los 5 participantes cuenta con su par ordenado de la tabla 3.2
2. El umbral establecido fue $t = 3$. Así que solo se tomarán 3 de esos pares ordenados para la recuperación del secreto:

Puntos a repartir
$p_2(2, 4)$
$p_3(3, 3)$
$p_5(5, 2)$

Tabla 3.3: Puntos a usar para la recuperación del secreto

3. Ahora se procederá a hacer el cálculo de los l_j :

$$\begin{aligned}
 l_2 &= \frac{x - x_3}{x_2 - x_3} \cdot \frac{x - x_5}{x_2 - x_5} = \frac{x - 3}{2 - 3} \cdot \frac{x - 5}{2 - 5} = \frac{x^2 - 8x + 15}{(-1)(-3)} = \frac{x^2 - 8x + 15}{3} \\
 l_3 &= \frac{x - x_2}{x_3 - x_2} \cdot \frac{x - x_5}{x_3 - x_5} = \frac{x - 2}{3 - 2} \cdot \frac{x - 5}{3 - 5} = \frac{x^2 - 7x + 10}{(1)(-2)} = \frac{x^2 - 7x + 10}{-2} \\
 l_5 &= \frac{x - x_2}{x_5 - x_2} \cdot \frac{x - x_3}{x_5 - x_3} = \frac{x - 2}{5 - 2} \cdot \frac{x - 3}{5 - 3} = \frac{x^2 - 5x + 6}{(3)(2)} = \frac{x^2 - 5x + 6}{6} \\
 &\vdots
 \end{aligned}$$

4. Ahora que se han obtenido los l_j , es posible calcular la función polinomial haciendo uso de la interpolación de Lagrange:

$$\begin{aligned}
 f(x) &= \sum y_j \cdot l_j = y_2 \cdot j_2 + y_3 \cdot j_3 + y_5 \cdot j_5 \\
 &= 4 \cdot \left(\frac{x^2 - 8x + 15}{3} \right) + 3 \cdot \left(\frac{x^2 - 7x + 10}{-2} \right) + 2 \cdot \left(\frac{x^2 - 5x + 6}{6} \right) \\
 &= \left(4 \cdot \left(\frac{1}{3} \right) 3 \cdot \left(\frac{1}{-2} \right) 23 \cdot \left(\frac{1}{6} \right) \right) x^2 + \left(4 \cdot \left(\frac{-8}{3} \right) 3 \cdot \left(\frac{-7}{-2} \right) 23 \cdot \left(\frac{-5}{6} \right) \right) x + \left((20) + (-15) + (2) \right) \\
 &= \left(4 \cdot (6) + 3 \cdot (8) + 23 \cdot (3) \right) x^2 + \left(4 \cdot (3) + 3 \cdot (12) + 23 \cdot (2) \right) x + 7 \\
 &= 3x^2 + x + 7 \implies f(x) = 3x^2 + x + 7 \text{ (mód 17).}
 \end{aligned}$$

5. Luego entonces tenemos que $f(x) = 3x^2 + x + 7 \implies a_2 = 3, a_1 = 1, a_0 = 7 \implies a_0 = 7 \implies S = 7$.
6. De tal forma que el secreto $S = 7$.

3.2. Robust Computational Secret Sharing

En el esquema de secreto compartido de Shamir, se supone que los participantes proveen partes (“shares”) correctas. En contraste, el Esquema Computacional Robusto de Secreto Compartido (*Robust Computational Secret Sharing - RCSS*) es un esquema de secreto compartido que puede recuperar correctamente un secreto, incluso en presencia de un número delimitado de “shares” corruptos, mientras mantiene la secrecía de la información [20]. El RCSS fue creado por Krawczyk en 1993.

El RCSS puede ser visto como una poderosa herramienta para construir sistemas de almacenamiento de información distribuidos que sean seguros y confiables [5]. Un conjunto de datos de un usuario (por ejemplo, un archivo), es separado en varias piezas (“shares”) y almacenado en múltiples servidores, de tal manera que es capaz de proteger la privacidad del usuario de servidores “fisgones”, y además permite la recuperación de un conjunto de datos incluso si algunos de los servidores proveen piezas (“shares”) erróneas (de forma accidental o intencional) [5].

El RCSS hace uso de un código de corrección de errores y de un algoritmo de dispersión de información. Un código de corrección de errores es un algoritmo para expresar una secuencia de números tal que cualquier error que se introduce puede ser detectado y corregido (dentro de ciertas limitaciones) tomando como base los números restantes [21].

Los algoritmos de dispersión de información proporcionan un método para almacenar información en fragmentos dispersos en múltiples ubicaciones, de modo que la redundancia proteja la información en caso de corrupción de datos, además de que el acceso no autorizado a cualquier fragmento no proporciona información utilizable. Sólo una entidad cuente con todos los fragmentos y con el algoritmo de dispersión original puede montar correctamente la información completa [22].

El algoritmo de Robust Computational Secret Sharing se puede expresar con el siguiente pseudocódigo.

```

PROCEDURE Share( $X$ )
10  $K \leftarrow \{0, 1\}^k$ ;  $C \leftarrow \text{Encrypt}_K(X)$ 
11  $\mathbf{K} \leftarrow \text{Share}^{\text{PSS}}(K)$ 
12  $\mathbf{C} \leftarrow \text{Share}^{\text{IDA}}(C)$ 
13 FOR  $i \leftarrow 1$  TO  $n$  DO
14      $\mathbf{H}[i] \leftarrow \text{Hash}(\mathbf{K}[i] \mathbf{C}[i])$ 
15      $\mathbf{S}_i \leftarrow \text{Share}^{\text{ECC}}(\mathbf{H}[i])$ 
16 FOR  $i \leftarrow 1$  TO  $n$  DO
17      $\mathbf{X}[i] \leftarrow \mathbf{K}[i] \mathbf{C}[i] \mathbf{S}_1[i] \cdots \mathbf{S}_n[i]$ 
18 RETURN  $\mathbf{X}$ 

PROCEDURE Recover( $\mathbf{X}, j$ )
20 FOR  $i \leftarrow 1$  TO  $n$  DO
21      $\mathbf{K}[i] \mathbf{C}[i] \mathbf{S}_1[i] \cdots \mathbf{S}_n[i] \leftarrow \mathbf{X}[i]$ 
22 FOR  $i \leftarrow 1$  TO  $n$  DO
23      $\mathbf{H}[i] \leftarrow \text{Recover}^{\text{ECC}}(\mathbf{S}_i, j)$ 
24 FOR  $i \leftarrow 1$  TO  $n$  DO
25     IF  $\mathbf{X}[i] \neq \diamond$  AND  $\text{Hash}(\mathbf{K}[i] \mathbf{C}[i]) \neq \mathbf{H}[i]$ 
26         THEN  $\mathbf{K}[i] \leftarrow \diamond$ ;  $\mathbf{C}[i] \leftarrow \diamond$ 
27  $K \leftarrow \text{Recover}^{\text{PSS}}(\mathbf{K}, j)$ 
28  $C \leftarrow \text{Recover}^{\text{IDA}}(\mathbf{C}, j)$ 
29  $X \leftarrow \text{Decrypt}_K(C)$ 
30 RETURN  $X$ 

```

Figura 3.1: Pseudocódigo del algoritmo Robust Computational Secret Sharing [5].

El algoritmo cuenta con dos procesos principales, “Share” y “Recovery”, a continuación se hará una descripción de los pasos que se realiza en la **Figura 3.1**.

Proceso de “Share”

- Línea 10. Se genera una llave K , tomando un vector aleatorio perteneciente al espacio de llaves compuesto por las cadenas binarias de longitud k . Posteriormente se cifra el secreto X haciendo uso de un cifrado por bloques, con la llave generada anteriormente.
- Línea 11. Se genera el vector \mathbf{K} , siendo resultado de aplicar la operación de “Share” (Compartir) bajo el esquema de secreto compartido de Shamir.
- Línea 12. Se genera el vector \mathbf{C} , como resultado de aplicar al secreto cifrado C un algoritmo de dispersión de información (*IDA - Information Dispersal Algorithm*).
- Líneas 13-15. Posteriormente se obtiene el digesto de cada elemento de los vectores \mathbf{K} y \mathbf{C} , almacenándose en el vector \mathbf{H} . Además, se genera el vector \mathbf{S}_i , compuesto por el vector \mathbf{H}_i después de aplicar un código de corrección de errores (*Error-Correcting Code*), esto con el fin de añadir redundancia a la información
- Líneas 16-17. Finalmente, se genera el vector \mathbf{X} , compuesto por los fragmentos de la llave \mathbf{K} , el vector \mathbf{C} y el vector \mathbf{S} . Los fragmentos del vector \mathbf{X} son los que serán distribuidos en el sistema (“shares”).

Proceso de “Recover”

- Líneas 20-21. De cada fragmento (“share”) se extraen sus componentes, obteniendo como resultado los vectores \mathbf{K} , \mathbf{C} , \mathbf{S} .
- Líneas 22-23. Para cada elemento del vector \mathbf{S} , se aplica el proceso de recuperación con el código de corrección de errores.
- Líneas 24-26. Si el vector \mathbf{X} contiene información (es distinto de nulo) y si el digesto de la llave \mathbf{K}_i con el cifrado \mathbf{C}_i no coincide con el digesto \mathbf{H}_i , entonces ese fragmento se considera como corrupto.
- Línea 27. Se recupera la llave aplicando el proceso de “Recover” (recuperación) al vector \mathbf{K} , que es el que contenía la llave.
- Línea 28. De igual forma se recupera el secreto cifrado C , mediante la operación “Recover” del algoritmo de dispersión de información.
- Línea 29. Por último, se descifra el secreto cifrado C , con la llave K , obteniendo el secreto X original.

Capítulo 4

Análisis

En este capítulo se detalla la etapa de Análisis del sistema, mencionando las herramientas tecnológicas que serán utilizadas, la factibilidad del proyecto de acuerdo con los recursos disponibles y las herramientas elegidas. Posteriormente, se hace mención de la metodología de desarrollo de software en la cual estará basado este Trabajo Terminal, además de los requerimientos funcionales, no funcionales, de seguridad y reglas de negocio, acordes con la metodología elegida.

4.1. Herramientas tecnológicas

4.1.1. Sistemas Operativos

GNU/Linux

GNU/Linux es una combinación del sistema operativo GNU, desarrollado por la FSF, y el núcleo (kernel) Linux, desarrollado por Linus Torvalds y la Linux Foundation. Su desarrollo es uno de los ejemplos más prominentes de software libre; todo su código fuente puede ser utilizado, modificado y redistribuido libremente por cualquiera bajo los términos de la GPL.

Se eligió hacer uso del sistema operativo GNU/Linux, debido a sus características de software libre, gratuito, y a su compatibilidad nativa con lenguajes de programación como es C y C++, además de que se puede encontrar una gran cantidad de software y herramientas de desarrollo enfocado a criptografía y sistemas distribuidos en los repositorios del proyecto GNU.

4.1.2. Lenguajes de Programación

Lenguaje de programación C

C es un lenguaje de programación de uso general. Fue desarrollado originalmente por Dennis Ritchie entre 1969 y 1972 en los Laboratorios Bell. Es un lenguaje imperativo con soporte para la programación estructurada, y comúnmente se le asocia con el sistema operativo UNIX, sin embargo, no está atado a algún sistema operativo o computadora [23].

Entre las características de C se encuentran [23]:

- Soporte para diferentes tipos de dato como son: carácter, entero y número de punto flotante.
- Tipos de datos derivados, como son: punteros, matrices, estructuras y uniones.
- Instrucciones de control de flujo, como son: if-else, switch, while, for, do y break.
- Funciones, que pueden devolver valores de tipos básicos, estructuras, uniones o punteros.
- Asignación dinámica de memoria.

Este lenguaje de programación no proporciona operaciones para tratar directamente con objetos compuestos como son cadenas de caracteres, conjuntos, listas o matrices [23].

C++

C++ es un lenguaje de programación de propósito general diseñado a mediados de los años 1980, por Bjarne Stroustrup. La intención de su creación fue extender al lenguaje de programación C con mecanismos que permitan la manipulación de objetos. C++ tiene características de programación imperativas, orientadas a objetos y genéricas, al mismo tiempo que proporciona facilidades para la manipulación de memoria a bajo nivel [24].

Este lenguaje de programación se diseñó con un enfoque hacia la programación de sistemas, centrándose en el rendimiento, eficiencia y flexibilidad. C++ ha sido utilizado en el desarrollo de aplicaciones enfocadas en computadoras de escritorio, servidores, y sistemas críticos [25].

C++ está estandarizado por la Organización Internacional de Estandarización (ISO), siendo la última versión estándar ratificada y publicada por ISO en diciembre del 2014 como ISO/IEC 14882:2014, conocido informalmente como C++ 14 [26].

Java

Java es un lenguaje de programación de propósito general, concurrente, orientado a objetos que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Java fue originalmente desarrollado por James Gosling en Sun Microsystems (ahora Oracle Corporation) y fue lanzado en 1995 como un componente básico de la plataforma Java de Sun Microsystems [27].

Las aplicaciones Java normalmente se compilán a bytecode que puede ejecutarse en cualquier máquina virtual Java (JVM), independientemente de la arquitectura del equipo. Las bibliotecas estándar de Java proporcionan una forma genérica de acceder a características específicas del equipo, como gráficos, procesos, hilos, archivos, y uso de la red.

Java es uno de los lenguajes de programación más populares en uso, particularmente para aplicaciones web cliente-servidor [28]. En la **Tabla 4.1** se muestra una comparativa de los lenguajes de programación más populares actualizada hasta abril del 2017.

Para el desarrollo del proyecto, se decidió hacer uso de los lenguajes de programación C++ y Java. Se eligió el lenguaje C++ debido a su compatibilidad nativa con el sistema operativo GNU/Linux, además de contar con un amplia flexibilidad, rendimiento y funcionalidades, a través de la biblioteca estándar o haciendo uso de bibliotecas externas. Se descartó el uso del

lenguaje C debido a que por si solo no proporciona funciones u operaciones para manipular estructuras complejas.

En el caso de Java, se eligió al ser compatible de igual forma con GNU/Linux, a través de la Máquina Virtual de Java (Java Virtual Machine), y tener capacidad para manejar aplicaciones con interfaz gráfica de cliente-servidor orientadas a Web.

Lugar	Lenguaje de programación	Rating	Cambio respecto al año anterior
1	Java	15.568 %	-5.28 %
2	C	6.966 %	-6.94 %
3	C++	4.554 %	-1.36 %
4	C#	3.579 %	-0.22 %
5	Python	3.457 %	+0.13 %
6	PHP	3.376 %	+0.38 %
7	Visual Basic .NET	3.251 %	+0.98 %
8	JavaScript	2.851 %	+0.28 %
9	Delphi/Object Pascal	2.816 %	+0.60 %
10	Perl	2.413 %	-0.11 %

Tabla 4.1: Popularidad de algunos lenguajes de programación [28].

4.1.3. Bibliotecas a utilizar

OpenSSL

OpenSSL es un proyecto de software libre basado en SSLeay, desarrollado por Eric Young y Tim Hudson. Consiste en un robusto paquete de herramientas de administración y bibliotecas relacionadas con la criptografía, que suministran funciones criptográficas a otros paquetes como OpenSSH y navegadores web (para acceso seguro a sitios HTTPS).

Crypto++

Crypto++ (o Cryptopp) es una biblioteca gratuita y de código abierto desarrollada en C++ de algoritmos de cifrado y esquemas de seguridad, desarrollada por Wei Dai. Crypto++ ha sido ampliamente utilizado en el mundo académico, en proyectos de código abierto y proyectos no comerciales, así como en el ámbito empresarial.

Este proyecto hará uso de ambas bibliotecas, ya que ambas proporcionan una amplia variedad de implementaciones criptográficas a los lenguajes de programación a utilizar, además de que ambas son gratuitas y de código abierto. Entre las implementaciones criptográficas de estas bibliotecas se encuentran: cifradores por bloques con modos de operación, esquemas de autenticación, funciones hash, criptografía de llave pública, etc.

4.1.4. Servidores

Apache Tomcat

Apache Tomcat es una implementación multiplataforma de código abierto de las tecnologías Java Servlet, JavaServer Pages, Java Expression Language y Java WebSocket. Funciona como un contenedor de servlets desarrollado bajo el proyecto Jakarta en la Apache Software Foundation [29].

Entre sus características se encuentran:

- Soporte para Microsoft Windows, GNU/Linux y MacOS.
- “Pool” de conexiones a bases de datos, a través de JDBC, con un alto rendimiento.
- Soporte para la tecnología WebSockets.
- Soporte para conexiones seguras a través de TLS, además permite utilizar los certificados creados con herramientas como OpenSSL [30].
- El número máximo de conexiones concurrentes se encuentra entre 8,192 y 10,000 [31].

Se hará uso de este servidor web debido a su compatibilidad con el lenguaje de programación Java, y a su compatibilidad con el gestor de carga de trabajo, el cual se mencionará más adelante. Además, cuenta con características de seguridad integradas, como es el protocolo TLS y el uso de certificados que pueden ser creados mediante OpenSSL.

4.1.5. Sistemas Gestores de Bases de Datos

MySQL

Es un Sistema Gestor de Bases de Datos relacional de código abierto, desarrollado bajo licencia GPL por Oracle Corporation. MySQL es compatible con una amplia variedad de sistemas operativos, entre los que se encuentran: Linux, MacOS y Microsoft Windows.

Entre algunas de sus características se encuentran [32]:

- Desarrollado en su totalidad en C/C++.
- Soporte multiplataforma.
- Soporte para procedimientos almacenados, utilizando un lenguaje procedural basado en SQL/PSM.
- Soporte para múltiples hilos de ejecución.
- Triggers.
- Cursores.
- Vistas.
- Soporte para conexiones SSL/TLS.
- Soporte Unicode.
- Cumple con las características de ACID (Atomicidad, Consistencia, Aislamiento y Durabilidad).
- Conectores para lenguajes de programación, entre ellos se encuentran: Java, C y C++.

El límite máximo de conexiones en MySQL por defecto es 151, sin embargo, el número máximo de conexiones posibles depende de la cantidad de RAM del equipo, la cantidad de RAM que se utiliza para cada conexión, la carga de trabajo y el tiempo de respuesta deseado.

En el sistema operativo Linux, MySQL debe ser capaz de soportar por lo menos de 500 a 1000 conexiones simultáneas de forma rutinaria, y hasta 10,000 conexiones si el equipo cuenta con suficiente RAM disponible y la carga de trabajo de cada conexión es baja o el tiempo de respuesta es poco exigente [33].

Se eligió MySQL como Sistema Gestor de Base de Datos debido a sus características de software gratuito y libre, además de que es compatible con el sistema operativo GNU/Linux y proporciona características de seguridad como son: autenticación, soporte para SSL/TLS, funciones para cifrado y descifrado, y funciones hash [34]. MySQL también provee conectores para los lenguajes de programación C++ y Java.

4.1.6. Frameworks para sistemas distribuidos

HTCondor

HTCondor es un sistema gestor de carga de trabajo, especializado para trabajos intensivos de cómputo. Proporciona mecanismos de gestión de trabajos como son: colas de trabajos, planificadores de tareas, esquemas de prioridades, supervisión y gestión de los recursos disponibles.

HTCondor es desarrollado por el Centro de Computación de Alto Rendimiento en el Departamento de Ciencias Computacionales, de la Universidad de Wisconsin-Madison. El software HTCondor, el código fuente y la documentación completa están disponibles gratuitamente bajo una licencia de código abierto. Este sistema es compatible con Linux, MacOS y plataformas Windows [35].

En este sistema, los usuarios envían sus trabajos (seriales o paralelos) a HTCondor, el sistema los coloca en una cola, elige cuándo y dónde ejecutarán basándose en reglas internas, monitorea su progreso e informa al usuario cuando finalizan. HTCondor permite aprovechar eficazmente la potencia del CPU de las computadoras disponibles. El software HTCondor puede ser utilizado para administrar un grupo de nodos de computación dedicados [35].

Globus Toolkit

Globus Toolkit es un software libre para construir Grids computacionales desarrollado por la Globus Alliance. Este sistema también incluye servicios y bibliotecas para el monitoreo, descubrimiento y administración de recursos, además de seguridad y administración de archivos.

Este software contiene herramientas para seguridad, infraestructura de información, administración de recursos, administración de datos, comunicación, detección de fallas y portabilidad. Se empaqueta como un conjunto de componentes que pueden utilizarse de forma independiente o conjunta para desarrollar aplicaciones. Globus Toolkit es compatible con Linux, MacOS y Microsoft Windows [36].

En este proyecto se hará uso del software HTCondor, dado que el funcionamiento de compartición y recuperación de los archivos se asemeja más a un gestor de carga de trabajo (Job scheduler). HTCondor se encuentra más enfocado hacia la gestión de trabajos, a diferencia

de Globus Toolkit, el cual ofrece un mayor conjunto de herramientas enfocadas al desarrollo de Grids. Además, HTCondor proporciona soporte nativo para la ejecución de programas desarrollados en C, C++ y Java.

4.2. Estudio de factibilidad

El estudio de factibilidad es un instrumento que sirve para orientar la toma de decisiones en la evaluación de un proyecto y corresponde a la última fase de la etapa pre-operativa dentro del ciclo del proyecto. Se formula con base en información que tiene la menor incertidumbre posible para medir las posibilidades de éxito o fracaso de un proyecto, apoyándose en él se tomará la decisión de proceder o no con su implementación [37]. Este estudio establecerá la viabilidad del proyecto.

- Factibilidad Técnica. Se refiere a los recursos necesarios como herramientas, conocimientos, habilidades, experiencia, etc., que son necesarios para efectuar las actividades o procesos que requiere el proyecto.
- Factibilidad Operativa. Se refiere a todos aquellos recursos donde interviene algún tipo de actividad (Procesos), depende de los recursos humanos que participen durante la operación del proyecto.
- Factibilidad Económica. Se refiere a los recursos económicos y financieros necesarios para desarrollar o llevar a cabo las actividades.

4.2.1. Factibilidad Técnica

En esta sección se describen las herramientas tecnológicas de las que se hará uso. Para la elección de dichas herramientas fue necesario hacer una investigación de las tecnologías más actuales, además de la revisión de las características que ofrecen y como es que se hará uso de ellas. A continuación, se muestra una tabla con las herramientas de software que se han contemplado:

Sistemas operativos
GNU/Linux
Lenguajes de programación
C
C++
Java
Bibliotecas
OpenSSL
Crypto++
Servidores
Apache Tomcat
Gestores de base de datos
MySQL
Frameworks
HTCondor

Tabla 4.2: Herramientas de software a utilizar.

Además de las herramientas tecnológicas de software, se debe considerar las de hardware, la arquitectura del sistema será implementada en equipos de cómputo personales, los cuales cuentan con las siguientes características:

Característica	Descripción
Marca	Samsung
Modelo	NP300V4A
Procesador	Intel Core i5 2450M (3100 MHz)
Tarjeta de vídeo	Intel integrada 1 GB
Memoria	8GB DDR3 (1333 MHz)
Disco Duro	HDD 500GB (5400rpm)
Ethernet	LAN Ethernet Gigabit
Red inalámbrica	Genérica (802.11 b/g/n)

Tabla 4.3: Especificaciones de los equipos Samsung np3v4a.

Característica	Descripción
Marca	Lenovo
Modelo	Lenovo y510p
Procesador	Intel Core i7- 4700MQ 4ta generación (2.40 GHz 1600 MHz 6 MB)
Tarjeta de vídeo	NVIDIA SLI – gráficos duales NVIDIA GeForce GT 750M 2GB
Memoria	16 GB DDR3L
Disco Duro	1 TB
Ethernet	LAN 1 GB
Red inalámbrica	Intel Centrino Wireless N-2230 - 802.11b/g/n

Tabla 4.4: Especificaciones de los equipos Lenovo y510p

Característica	Descripción
Marca	Hp
Modelo	envi-4
Procesador	Intel® Core™ 3MB de caché L3 (i5 a 1,7GHz)
Memoria	RAM DD3 8GB
Disco Duro	SATA de 5400 rpm (500GB)
Ethernet	LAN Ethernet Gigabit
Red inalámbrica	Intel 802.11 b/g/n con WiFi

Tabla 4.5: Especificaciones de los equipos HP envi-4

Además de las herramientas de hardware y software a utilizar, es necesario hacer mención del uso de los servicios básicos para el desarrollo del sistema, como son:

1. Energía eléctrica
2. Agua potable
3. Línea telefónica / Uso de internet *Papelería*

Estos servicios básicos tienen relevancia, dado que son factores necesarios para el desarrollo del sistema e implican un costo, dicho costo será mencionado en el estudio de factibilidad económica.

4.2.2. Factibilidad Operativa

Se refiere a todos aquellos recursos que intervienen en el desarrollo de algún proceso. Éste depende de los recursos humanos que participan durante la operación del proyecto.

Para hacer el análisis del personal que estará en operación, es necesario calcular el tiempo real de esfuerzo y las horas dedicadas al sistema.

En la siguiente tabla se muestra cómo se calcularon las horas de desarrollo del sistema:

Mes	No. Días	Sábados y Domingos	Días no laborales	Días hábiles	Días no hábiles	Horas de trabajo por día (promedio)	Horas totales	Días laborables (8 horas al día)
Enero	31	9	10	19	12	2	24	3
Febrero	28	8	1	9	19	2	38	4.75
Marzo	31	8	1	9	22	2	44	5.5
Abril	30	9	5	14	16	2	32	4
Mayo	31	8	1	9	22	2	44	5.5
Junio	30	8	0	8	22	2	24	3
Agosto	31	8	0	8	23	2	46	5.75
Septiembre	30	9	1	10	20	2	40	5
Octubre	31	9	0	9	22	2	44	5.5
Noviembre	30	8	1	9	21	2	42	5.25

Tabla 4.6: Relación de días laborales periodo de desarrollo

Se ha decidido tomar como días desarrollo solo los días hábiles de la semana, excluyendo los fines de semana y los días festivos en el periodo de **enero-diciembre** de este año en curso, con un periodo de trabajo de alrededor de dos horas por día. Finalmente se convierte el periodo total de horas en días laborales (jornadas de 8 horas) obteniendo un total de **47.25 de días laborales**.

Una vez que se ha obtenido el tiempo estimado de desarrollo, es necesario definir los roles necesarios que se estarán desempeñando, los cuales se presentan en la siguiente tabla.

Rol	Personas por rol
Líder de proyecto	1
Programador	1
Analista de sistemas	1
Diseñador	1
Tester	1

Tabla 4.7: Roles a desempeñar para el desarrollo del sistema

La tabla anterior muestra los roles a desempeñar y el número de personas necesarias para ellos. Es importante mencionar que los roles aquí especificados no serán desempeñados sólo por una persona, según la etapa de desarrollo alguno o varios de los integrantes desempeñarán algún rol por un periodo según la etapa de desarrollo.

La división del trabajo será expresada en porcentajes de la siguiente manera:

Rol	% de participación	Días laborales
Líder de proyecto	100 %	47.25
Analista de sistemas	50 %	23.62
Diseñador	50 %	23.62
Programador	30 %	14.17
Tester	20 %	9.45

Tabla 4.8: Roles a desempeñar para el desarrollo del sistema.

4.2.3. Factibilidad Económica

Después de hacer un análisis de las propuestas hechas en los estudios de factibilidad técnica y factibilidad operativa, se definieron los siguientes recursos materiales y humanos para la etapa de desarrollo del sistema.

- Personal: Se tiene contemplado un tiempo en días laborales de 47.25.
- Mobiliario: Se cuenta con las instalaciones de la escuela, así como las viviendas propias para el desarrollo del trabajo, debido a ello no se generaron gastos de esta índole.

Estimación de los costos de Software y Hardware en caso de no contar con los recursos ya mencionados

A continuación, se muestran los costos monetarios en caso de haber adquirido dichas herramientas para poder trabajar:

- Servicios a utilizar

Servicio	Costo mensual	Costo anual
Luz	\$1,000	\$7,000
Telefonia/Internet	\$600	\$4200
Total:		\$11,200

Tabla 4.9: Servicios

- Software a utilizar

Costo	Costo mensual
Software libre \$599	Servidor Web Apache Tomcat Herramientas de escritorio (Office)
Software libre	Sistema operativo Linux
Software libre	Net-Beans
Software libre	MYSQL
Software libre	Framework HTCondor
Total: 599.00	

Tabla 4.10: Costo del software

- Hardware a utilizar

Equipo	Precio	Descripción
Laptop	\$22,000	Intel Core i7- 4700MQ 4 generación (2.40 GHz)
Laptop	\$5,000	Samsung NP300V4A Intel Core i5 2450M (3100 MHz) 8GB DDR3
Laptop	\$7,000	HP envy-4 i5 a 1,7GHz DD3 8GB
Total: \$34,000.		

Tabla 4.11: Costo del Hardware

Como se mencionó en el estudio de factibilidad, se necesitarán 47.25 días laborales para el desarrollo del proyecto. Suponiendo una jornada laboral de 8 horas diarias por 5 días a la semana el proyecto se terminaría en 2.3 meses es decir 5 quincenas.

Se analizaron varios sitios especializados en los sueldos promedio de TI en México señalando un sueldo quincenal para desarrolladores de software de 9,000\$ MxN [38].

- Para personal

Responsable	Sueldo	Quincenas	Costo del esfuerzo
Armenta García Guadalupe Javier	\$9,000	5	\$45,000
Cárdenas Castillo Víctor Hugo	\$9,000	5	\$45,000
Moreno Zárate Víctor Gibrán	\$9,000	5	\$45,000
Total: \$135,000			

Tabla 4.12: Personal

Realizando la estimación anterior, se observa que el costo total del desarrollo del software es: **\$169,599**.

Conclusión Después de haber realizado el estudio de factibilidad se concluye que el desarrollo de este sistema es “**viable**”, debido a que se cuenta con los recursos necesarios para su desarrollo, señalando que el equipo responsable no recibió el salario mencionado y que además ya se contaba con el software y hardware necesario.

4.3. Metodología

De acuerdo con las características que posee el proyecto, se optó por hacer uso de la metodología SDL (Security Development Lifecycle) de Microsoft puesto que esta se enfoca en la elaboración de software que requiere de la implementación de seguridad dentro de su funcionalidad.

El proceso SDL de Microsoft se basa en tres conceptos básicos [39]:

- Formación: Al contar con un personal dentro de un grupo de desarrollo de software que está en constante actualización las organizaciones se mantienen al día en cuanto a los cambios en las tecnologías y las amenazas que involucran el desarrollo.
- Mejora continua de los procesos: Se realizan constantemente evaluaciones sobre los posibles riesgos y amenazas sobre la seguridad de los sistemas; esto a su vez conlleva a la recopilación de datos para la evaluación de la eficacia de la formación del personal y el uso de métricas de procesos para documentar la conformidad. Además, se hacen métricas posteriores al lanzamiento para ayudar a definir los futuros cambios.
- Responsabilidad: Por último, SDL requiere el archivado de todos los datos necesarios para realizar el mantenimiento de una aplicación en caso de que surjan problemas. Si lo combinan con detallados planes de comunicación y de respuesta en materia de seguridad, las organizaciones podrán orientar de manera concisa y contundente a todas las partes implicadas.

Microsoft recomienda hacer uso de esta metodología siempre y cuando el proyecto a desarrollar cumpla con al menos una de estas características a desarrollar [39]:

- Aplicaciones implementadas en un entorno empresarial.
- Aplicaciones que procesan información de identificación personal (PII) u otro tipo de información confidencial.
- Aplicaciones que se comunican frecuentemente a través de Internet u otras redes.

La metodología está estructurada en torno a cinco áreas de capacidades que, se corresponden con las fases del ciclo de vida de desarrollo de software:

- Formación
- Requisitos
- Diseño
- Implementación
- Comprobación
- Lanzamiento
- Respuesta

Dentro de cada una de estas áreas se definen ciertas tareas que describen el desarrollo de dicha área; Microsoft propone como base algunas de las tareas que deben implementarse para llevar a cabo dicha metodología, esto no significa que el grupo de desarrollo pueda agregar fases dentro de dichas áreas para aumentar los niveles de seguridad según sus necesidades. Dichas tareas se muestran en la siguiente figura:



Figura 4.1: Diagrama de la metodología SDL [39].

4.4. Requerimientos

4.4.1. Requerimientos Funcionales

- **RF1. Subida de archivos.** El usuario podrá subir archivos de cualquier tipo (extensión).
- **RF2. Descarga de archivos.** El usuario podrá descargar sus archivos.
- **RF3. Organización de carpetas.** El usuario podrá organizar sus archivos en carpetas.
- **RF4. Eliminación de archivos.** El usuario podrá eliminar sus archivos.
- **RF5. Eliminación de carpetas.** El usuario podrá eliminar sus carpetas.
- **RF6. Renombrar archivos.** El usuario podrá renombrar sus archivos.
- **RF7. Renombrar carpetas.** El usuario podrá renombrar sus carpetas.
- **RF8. Creación de cuenta.** El usuario podrá crear una cuenta.
- **RF9. Eliminación de cuenta.** El usuario podrá eliminar su cuenta.
- **RF10. Visualización de detalles de cuenta.** El usuario podrá visualizar detalles de su cuenta, como son:
 - Alias.
 - Correo asociado.
 - Espacio disponible.
 - Espacio ocupado.
 - Espacio total.
- **RF11. Modificación de cuenta.** El usuario podrá modificar la contraseña de usuario de su cuenta.

4.4.2. Requerimientos No Funcionales

- **RFN1. Nombre de usuario.** El nombre de usuario será un correo valido.
- **RFN2. Tratamiento de archivos.** Los archivos podrán ser almacenados y recuperados acorde a los umbrales del esquema del secreto compartido.
- **RFN3. Plataforma de desarrollo.** El sistema será desarrollado como aplicación web.
- **RFN4. Límite de almacenamiento.** Límite de almacenamiento de 5 GB por usuario.
- **RFN5. Límite de cargas concurrentes.** Límite de 3 cargas de archivos máxima por usuario, para evitar la saturación del sistema.
- **RFN6. Límite de descargas concurrentes.** Límite de 3 descargas de archivos máxima por usuario, para evitar la saturación del sistema.

- **RFN4. Resistencia de contraseñas** Las contraseñas de usuario serán resistentes a ataques de fuerza bruta.
- **RFN8. Características de usabilidad** Cumplirá con las características de acuerdo a la norma ISO 9126, en la sección de usabilidad.
- **RFN9. Compatibilidad con navegadores.** Será compatible con navegadores que cumplan con el estándar HTML-5.
- **RFN10. Contenedor web.** La aplicación será desarrollada en el contenedor web Apache Tomcat versión 8.5.
- **RFN11. Gestor de base de datos.** Se utilizar MySQL versión 5.7 como gestor de base de datos.
- **RFN12. Tecnologías para interfaz de usuario.** El sistema hará uso de HTML, CSS, Javascript como tecnologías para la interfaz de usuario.
- **RFN13. Framework para balanceo de carga.** El sistema hará uso de HTCondor como gestor de carga de trabajo en la versión 8.6.
- **RFN14. Lenguajes de programación.** El sistema hará uso de C++ para el procesamiento de información en el esquema de secreto compartido y Java como servicio web.
- **RFN15. Entornos de desarrollo.** El sistema hará uso de Netbeans como IDE, debido a que es compatible con las tecnologías a usar.
- **RFN16. Sistema operativo base.** El sistema hará uso de GNU/Linux en su distribución Ubuntu como Sistema Operativo ya que cuenta con:
 - Soporte de Apache Tomcat.
 - Soporte de MySQL.
 - Soporte de G++.
 - Soporte de GCC.
 - Soporte de JVM.
 - Soporte de Java Compiler.
 - Se encuentran de forma nativa los repositorios de HTCondor.
- **RFN17. Caducidad de las sesiones.** Las sesiones del sistema caducarán después de 30 minutos de inactividad, con el fin de evitar que personas ajenas al usuario tengan acceso al sistema.

4.4.3. Requerimientos de Seguridad

- **RS1. Autenticación de entidades.** La autenticación de los usuarios será vía login.
- **RS2. Recuperación de cuentas de usuario.** La recuperación de cuentas de usuario será por medio del correo electrónico con el cual está asociada la cuenta.

- **RS3. Confidencialidad de la información almacenada.** El sistema garantizará la confidencialidad de la información almacenada, dentro de los umbrales del esquema de secreto compartido.
- **RS4. Integridad de la información almacenada.** El sistema garantizará la integridad de la información almacenada, dentro de los umbrales del esquema de secreto compartido.
- **RS5. Disponibilidad de la información almacenada.** El sistema garantizará la disponibilidad de la información almacenada, dentro de los umbrales del esquema de secreto compartido.

4.5. Reglas del negocio

4.5.1. Archivos

- **RN-A1. Confidencialidad de los Archivos:** Los Usuarios sólo podrán tener acceso a sus propios Archivos, esto con el fin de mantener la confidencialidad en el sistema.
- **RN-A2. Longitud del nombre del Archivo:** Los nombres de los Archivos deberán contener una longitud mínima de 1 carácter, y una longitud máxima de 255 caracteres, esto se debe a las limitaciones del sistema de archivos ext-4.
- **RN-A3. Carácter barra inversa:** Los nombres de los Archivos no pueden contener el carácter barra inversa {\}, debido a las limitaciones del sistema de archivos ext-4.
- **RN-A4. Nombre del Archivo:** Los Archivos no podrán llamarse de la siguiente forma: {.} o {..}, dado que se encuentran reservados en el sistema de archivos ext-4.
- **RN-A5. Tipo de Archivo:** Los Usuarios podrán subir cualquier tipo de Archivo (extensión).
- **RN-A6. Cola de subida:** Los Usuarios sólo podrán agregar como máximo 3 archivos a la cola de subida, a fin de que no se sature el sistema.
- **RN-A7. Cola de bajada:** Los Usuarios sólo podrán agregar como máximo 3 archivos a la cola de descarga, a fin de que no se sature el sistema.
- **RN-A8. Capacidad de almacenamiento:** Cada Usuario podrá acumular como máximo 5 GB de Archivos.
- **RN-A9. Unicidad en el nombre del Archivo:** No pueden existir dos Archivos con el mismo nombre dentro de la misma Carpeta.
- **RN-A10. Tamaño máximo de un archivo:** El sistema soportara archivos con un tamaño de hasta 1 GB como máximo.

4.5.2. Nombre de Usuario

- **RN-U1. Unicidad del nombre de Usuario:** El nombre de Usuario es único, no podrá haber dos Usuarios con el mismo nombre de Usuario.
- **RN-U2. Espacios en el nombre de Usuario:** El nombre de Usuario no podrá contener espacios.
- **RN-U3. Longitud del nombre de Usuario:** El nombre de Usuario deberá tener una longitud mínima de 8 caracteres y máxima de 255 caracteres.
- **RN-U4. Caracteres válidos en el nombre de Usuario:** El nombre del Usuario sólo podrá contener caracteres imprimibles definidos en el ASCII, como son: Caracteres en mayúsculas {A-Z}, caracteres en minúsculas {a-z}, números {0-9}, guión bajo {_}, exclamación {!}, arroba {@}, símbolo de número {#}.
- **RN-U5. Caracteres no válidos en el nombre de Usuario:** Los nombres de Usuario no pueden contener los siguientes caracteres: Et {&}, igual {=}, comillas angulares {< >}, más {+}, coma {,} o barra inversa {\}.
- **RN-U6. Caracteres no válidos al inicio en el nombre de Usuario:** El nombre de Usuario no puede iniciar con punto {.} o con coma {,}.
- **RN-U7. Caracteres repetidos no válidos en el nombre de Usuario:** El nombre de Usuario no puede contener más de dos puntos {..} seguidos.

4.5.3. Contraseña del Usuario

- **RN-C1. Caracteres válidos en la contraseña del Usuario:** La contraseña del Usuario sólo podrá contener caracteres imprimibles definidos en el ASCII.
- **RN-C2. Longitud de la contraseña del Usuario:** La contraseña del Usuario deberá tener una longitud mínima de 12 caracteres y máxima de 255 caracteres.
- **RN-C3. Robustez de la contraseña del Usuario:** La contraseña deberá incluir al menos: Un carácter en mayúscula {A-Z}, un carácter en minúscula {a-z}, un carácter numérico {0-9} y un carácter especial. Estas características se determinaron con base en las recomendaciones de Microsoft [40] y Google [41].
- **RN-C4. Contraseña del Usuario diferente del nombre de Usuario:** La contraseña no deberá ser igual al nombre de Usuario.
- **RN-C5. Recuperación de la contraseña del Usuario:** La recuperación de la contraseña se hará vía correo electrónico.

4.5.4. Inicio de sesión

- **RN-I1. Nombre de Usuario existente:** Cuando se inicie sesión, el nombre de Usuario debe estar registrado en el sistema.

- **RN-I2. Contraseña del Usuario válida:** Cuando se inicie sesión, la contraseña del Usuario debe corresponder a la registrada en el sistema.
- **RN-I3. Múltiples intentos de inicio de sesión fallidos:** En caso de 10 intentos de inicio de sesión fallidos, la cuenta del usuario quedará bloqueada y sólo podrá recuperarse mediante la opción de “Recuperar Contraseña” vía correo electrónico.

4.5.5. Carpetas

- **RN-CA1. Unicidad en el nombre de las Carpetas:** No pueden existir dos Carpetas con el mismo nombre dentro de una misma Carpeta.
- **RN-CA2. Longitud del nombre de Carpeta:** Los nombres de las Carpetas deberán contener una longitud mínima de 1 carácter, y una longitud máxima de 255 caracteres, esto se debe a las limitaciones del sistema de archivos ext-4.
- **RN-CA3. Carácter barra inversa:** Los nombres de las Carpetas no pueden contener el carácter barra inversa {\\"}, debido a las limitaciones del sistema de archivos ext-4.
- **RN-CA4. Nombre de la Carpeta:** Las Carpetas no podrán llamarse de la siguiente forma: {.} o {..}, dado que se encuentran reservados estos nombres en el sistema de archivos ext-4.

4.6. Umbrales de calidad y límite de errores

De acuerdo con la metodología SDL [39], se usan umbrales de calidad y límites de errores para establecer niveles mínimos aceptables de calidad en materia de seguridad y privacidad. Al definir estos criterios al comienzo de un proyecto, se comprenderán mejor los riesgos asociados a los problemas de seguridad y los equipos podrán identificar y corregir los errores de seguridad durante el desarrollo.

Un límite de errores es un umbral de calidad que se aplica a todo el proyecto de desarrollo de software. Se usa para definir los umbrales de gravedad de las vulnerabilidades de seguridad; por ejemplo, se puede establecer que en una aplicación no hay vulnerabilidades conocidas que estén clasificadas como “críticas” o “importantes” en el momento de su lanzamiento. Una vez definido, el límite de errores no debe disminuir [39].

En la **Tabla 4.13** se muestran los posibles defectos, fallas o errores que podría tener el sistema del lado del servidor, y en la **Tabla 4.14** del lado del cliente. Posteriormente se menciona cuáles de estos errores estarán definidos en el umbral del sistema. Para la realización de ambas tablas, se usaron las definiciones para la metodología SDL de Microsoft [39].

Nivel	Falla/Error	Descripción	Ejemplo
-------	-------------	-------------	---------

Crítico	Elevación de privilegios	Situación en la cual un adversario puede ejecutar realizar acciones a las cuales no tiene permiso.	<ul style="list-style-type: none"> ■ Acceso, modificación y borrado de archivos pertenecientes a los demás usuarios. ■ Ejecución de código arbitrario.
Crítico	Acceso a la información de los servidores	Ataque en el cual un adversario tiene acceso a información restringida sobre el sistema.	<ul style="list-style-type: none"> ■ Acceso a credenciales del sistema operativo, base de datos, balanceador de carga, etc.
Crítico	Falla total del servidor	En este tipo de errores o fallas se presentan cuando ya sea por problemas físicos o lógicos no se puede proveer el servicio a los usuarios.	<ul style="list-style-type: none"> ■ Memoria defectuosa ■ Malware ■ Sobrecalentamiento ■ Falla CPU ■ Falla de la tarjeta de red
Crítico	Falla en la recuperación de los archivos.	Sucede cuando el servidor es incapaz de recuperar los archivos manteniendo su integridad.	<ul style="list-style-type: none"> ■ Corrupción de los archivos. ■ Falla de los servidores más allá del umbral de recuperación del esquema de Secreto Compartido. ■ Fallas en la implementación del esquema de Secreto Compartido.
Crítico	Falla en autenticación de usuario.	El sistema permite el acceso a cuentas de usuario sin haber realizado un inicio de sesión exitoso.	<ul style="list-style-type: none"> ■ Ataques por fuerza bruta. ■ Bugs en el Login.
Importante	Denegación de servicio	Tipo de ataque en el cual el atacante busca hacer inaccesible el servicio a usuarios legítimos del sistema.	<ul style="list-style-type: none"> ■ Múltiples intentos de inicio de sesión en un periodo de tiempo corto. ■ Múltiples peticiones de bajada de archivos. ■ Múltiples peticiones de subida de archivos.

Importante	Inyección SQL	Ejecución de consultas sobre la base de datos de manera maliciosa, con la cual se pretende alterar la integridad de la información, incrustar e inclusive robar dicha información.	<ul style="list-style-type: none"> ■ Inserción de código malicioso SQL en los formularios. ■ Revelación de información. ■ Robo de credenciales de usuarios.
Importante	Envío de información en claro.	Sucede cuando el servidor envía en cliente información sin ser cifrada.	<ul style="list-style-type: none"> ■ Uso de Sniffer's para interceptar la información confidencial. ■ Uso de protocolos inseguros por parte del sistema.
Moderado	Falla en la carga/descarga de archivos.	<ul style="list-style-type: none"> ■ Situación en la cual los archivos no son capaces de ser almacenados por el servidor. ■ Situación en la cual los archivos no son capaces de ser descargados del servidor. 	<ul style="list-style-type: none"> ■ Sobrepaso de dimensiones permitidas por la aplicación. ■ Mala calidad de conexión. ■ Errores en la red. ■ Mantenimiento del servidor. ■ Servidores desconectados.

Tabla 4.13: Umbrales de calidad y límite de errores (Servidor).

De los posibles errores mencionados del lado del servidor en la tabla anterior, el umbral de errores estará conformado por los siguientes:

- Acceso a la información de los servidores.
- Falla en la recuperación de los archivos (dentro de los límites del esquema de secreto compartido).
- Falla en la autenticación.
- Denegación de servicio.
- Inyección SQL.
- Envío de información en claro.

Nivel	Falla/Error	Descripción	Ejemplo
Crítico	Errores de interacción entre el navegador y el sistema.	Situación en la cual el navegador de cliente deja de responder o se cierra por si solo al hacer uso del sistema.	<ul style="list-style-type: none"> ■ Cierre inesperado de la aplicación. ■ Uso excesivo de recursos, como es RAM o CPU.
Moderado	Interfaz no usable	<p>Dentro de la usabilidad descrita por la ISO-9126 se describen cuatro aspectos de la usabilidad los cuales se enfocan en la calidad del software dichos aspectos son:</p> <ul style="list-style-type: none"> ■ Aprendizaje ■ Comprensión ■ Operatividad ■ Atractividad 	<ul style="list-style-type: none"> ■ La interfaz no realiza correctamente las acciones del usuario. ■ Corrupción en el despliegue de información del usuario. ■ La interfaz no es intuitiva al usuario. ■ La interfaz es difícil de manejar para usuarios "inexpertos". ■ El diseño de la interfaz no es adecuado.
Moderado	Caducidad de las sesiones de usuario	El uso de sesiones dentro de la aplicación permite controlar la actividad del usuario dentro de la misma, guardando la información que es usada frecuentemente.	<ul style="list-style-type: none"> ■ Caducidad corta de sesión. ■ Ausencia de caducidad de sesiones.
Bajo	Mensajes de error en la consola de errores del navegador.	Se presenta cuando el navegador no es capaz de procesar alguna parte de la información proveniente del servidor.	<ul style="list-style-type: none"> ■ Errores relacionados al procesamiento de tecnologías como JavaScript, CSS, HTML, etc. ■ Incompatibilidad del navegador con las tecnologías usadas en el sistema.

Tabla 4.14: Umbrales de calidad y límite de errores (Cliente).

El umbral de errores para el lado del cliente estará conformado por:

- Errores de interacción entre en navegador y el sistema.
- Interfaz no usable.
- Caducidad de las sesiones de usuario.

De tal forma que al finalizar el desarrollo de la aplicación, esta no presentará los errores mencionados en los umbrales establecidos.

Capítulo 5

Diseño

En la etapa de Diseño se especifican las características del sistema necesarias para permitir su interpretación y realización física. A continuación se muestran los casos de uso, la arquitectura, el diseño de la base de datos, los diagramas de clases y los diagramas de secuencia asociados al sistema. También se incluyen bocetos de las interfaces gráficas de usuario y actividades específicas para la metodología SDL.

5.1. Casos de Uso

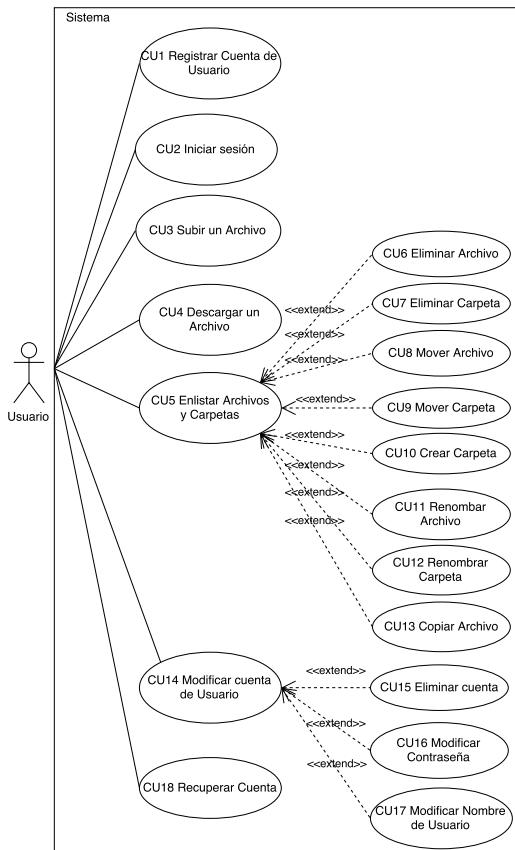


Figura 5.1: Diagrama de Casos de Uso del Sistema.

5.2. Documentación de los Casos de Uso

5.2.1. CU1 Registrar Cuenta de Usuario

Caso de Uso: CU1 Registrar Cuenta de Usuario	
Concepto	Descripción
Actor:	Usuario.
Propósito:	Este caso de uso permite al actor crear una cuenta dentro del sistema para poder tener acceso a sus funcionalidades.
Entradas:	Nombre de Usuario (e-mail). Contraseña del Usuario. Alias.
Salidas:	PP1 Pantalla Principal.
Pre-condiciones:	El Usuario debe encontrarse en la pantalla PI2 Registro de Cuenta de Usuario .
Post-condiciones:	Usuario Registrado.
Reglas de negocio:	RN-C1. Caracteres válidos en la contraseña del Usuario. RN-C2. Longitud de la contraseña del Usuario. RN-C3. Robustez de la contraseña del Usuario. RN-C4. Contraseña del Usuario diferente del nombre de Usuario. RN-U1. Unicidad del nombre de Usuario. RN-U2. Espacios en el nombre de Usuario. RN-U3. Longitud del nombre de Usuario. RN-U4. Caracteres válidos en el nombre de Usuario. RN-U5. Caracteres no válidos en el nombre de Usuario. RN-U6. Caracteres no válidos al inicio en el nombre de Usuario. RN-U7. Caracteres repetidos no válidos en el nombre de Usuario.
Errores:	Contraseña no valida. Nombre de Usuario no válido.

Trayectoria Principal

1.  Ingresa el Alias, Contraseña y Nombre de Usuario a los cuales se asociará la cuenta.
2.  Da clic en el botón “Registrar”.
3.  Verifica RN-C1. Caracteres válidos en la contraseña del Usuario, RN-C2. Longitud de la contraseña del Usuario, RN-C3. Robustez de la contra-

seña del Usuario, RN-C4. Contraseña del Usuario diferente del nombre de Usuario. [Trayectoria Alternativa A].

4. Verifica RN-U1. Unicidad del nombre de Usuario, RN-U2. Espacios en el nombre de Usuario, RN-U3. Longitud del nombre de Usuario, RN-U4. Caracteres válidos en el nombre de Usuario, RN-U5. Caracteres no válidos en el nombre de Usuario, RN-U6. Caracteres no válidos al inicio en el nombre de Usuario, RN-U7. Caracteres repetidos no válidos en el nombre de Usuario. [Trayectoria Alternativa B].
5. Despliega **PMSG2 Mensaje de información** con el mensaje “Usuario registrado correctamente”.
6. Despliega la **PP1 Pantalla Principal**.

Fin de la Trayectoria Principal.

Trayectoria Alternativa A. Contraseña no válida.

1. Despliega **PMSG2 Mensaje de información** con el mensaje “Contraseña no válida” especificando el por qué no se pudo registrar el usuario debido a la regla de negocio que no cumple.
2. Despliega la pantalla **PI2 Registro de Cuenta de Usuario**

Fin de la Trayectoria Alternativa A.

Trayectoria Alternativa B. Nombre de Usuario no válido.

1. Despliega **PMSG2 Mensaje de información** con el mensaje “Nombre de Usuario no válido” especificando el por qué no se pudo registrar el usuario debido a la regla de negocio que no cumple.
2. Despliega la pantalla **PI2 Registro de Cuenta de Usuario**

Fin de la Trayectoria Alternativa B.

5.2.2. CU2 Iniciar sesión

Caso de Uso: CU2 Iniciar sesión.	
Concepto	Descripción
Actor:	Usuario.
Propósito:	Este caso de uso permite al actor iniciar sesión en el sistema para poder hacer uso del mismo.
Entradas:	Nombre de Usuario. Contraseña del Usuario.
Salidas:	PP1 Pantalla Principal
Pre-condiciones:	El Usuario debe encontrarse registrado en el sistema. El Usuario debe encontrarse en la pantalla PI1 Inicio de Sesión .
Post-condiciones:	Ninguna.
Reglas de negocio:	RN-I1. Nombre de Usuario existente. RN-I2. Contraseña del Usuario válida. RN-I3. Múltiples intentos de sesión fallidos.
Errores:	Usuario o Contraseña incorrectos. Bloqueo de cuenta debido a múltiples inicios de sesión fallidos.

Trayectoria Principal

1.  Ingresa el Nombre de Usuario y Contraseña de Usuario.
2.  Da clic en el botón “Iniciar Sesión”.
3.  Verifica **RN-I1. Nombre de Usuario existente**, **RN-I2. Contraseña del Usuario válida**. [Trayectoria Alternativa A].
4.  Verifica **RN-I3. Múltiples intentos de sesión fallidos**. [Trayectoria Alternativa B].
5.  Despliega **PP1 Pantalla Principal**.

Fin de la Trayectoria Principal.

Trayectoria Alternativa A. Credenciales de inicio de sesión no válidas.

1.  Despliega **PMSG2 Mensaje de información** con el mensaje “Usuario o Contraseña incorrectos”.

Fin de la Trayectoria Alternativa A.

Trayectoria Alternativa B. Múltiples intentos de sesión.

1. Despliega **PMSG2 Mensaje de información** con el mensaje “La cuenta ha sido bloqueada debido a múltiples intentos de inicio de sesión fallidos”.

Fin de la Trayectoria Alternativa B.

5.2.3. CU3 Subir un archivo

Caso de Uso:	
Concepto	Descripción
Actor:	Usuario.
Propósito:	Este caso de uso permite al actor subir un archivo al sistema.
Entradas:	El archivo que subirá al sistema.
Salidas:	Desplegar PMSG2 Mensaje de información con el estatus final de la subida del archivo.
Pre-condiciones:	El Usuario debe haber iniciado sesión en el sistema. El Usuario debe encontrarse en PP1 Pantalla Principal .
Post-condiciones:	Ninguna.
Reglas de negocio:	RN-A2. Longitud del nombre del Archivo. RN-A3. Carácter barra inversa. RN-A4. Nombre del Archivo. RN-A5. Tipo de Archivo. RN-A6. Cola de subida. RN-A8. Capacidad de almacenamiento. RN-A9. Unicidad en el nombre del Archivo. RN-A10. Tamaño máximo de un archivo.
Errores:	El usuario subió el archivo con un nombre incorrecto de acuerdo con las reglas del negocio. Su archivo excede el límite de tamaño máximo por archivo El tamaño total de su archivo excedió el límite de espacio disponible de su cuenta.

Trayectoria Principal

1.  Da clic en el botón “Subir Archivo”.
2. Verifica el número de archivos subiéndose **RN-A6. Cola de subida**. [Trayectoria Alternativa A].
3.  Selecciona el archivo que desea subir.

4. Verifica el nombre del archivo de acuerdo con **RN-A2. Longitud del nombre del Archivo**, **RN-A3. Carácter barra inversa**, **RN-A4. Nombre del Archivo**, **RN-A5. Tipo de Archivo**. [Trayectoria Alternativa B].
5. Verifica **RN-A9. Unicidad en el nombre del Archivo** [Trayectoria Alternativa C].
6. Verifica el tamaño del archivo acorde a **RN-A8. Capacidad de almacenamiento**, **RN-A10. Tamaño máximo de un archivo**. [Trayectoria Alternativa D].
7. Despliega **PMSG2 Mensaje de información** con el mensaje “Archivo subido correctamente”, y despliega **PP1 Pantalla Principal**.

Fin de la Trayectoria Principal.

Trayectoria Alternativa A. Nombre de Archivo no válido

1. Despliega **PMSG2 Mensaje de información** con el mensaje “Límite de cola de subida excedido”.

Fin de la Trayectoria Alternativa A.

Trayectoria Alternativa B. Nombre de Archivo no válido

1. Despliega **PMSG2 Mensaje de información** con el mensaje “Nombre de Archivo inválido”, además de indicar las reglas del negocio que no cumple.

Fin de la Trayectoria Alternativa B.

Trayectoria Alternativa C. Existencia del Archivo en la cuenta.

1. Despliega **PMSG2 Mensaje de información** con el mensaje “Existe un archivo con el mismo nombre”.

Fin de la Trayectoria Alternativa C.

Trayectoria Alternativa D. Capacidad excedida.

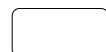
1. Despliega **PMSG2 Mensaje de información** con el mensaje “El Archivo sobrepasa la capacidad del sistema”, además de indicar la causa de la falla en la subida, dependiendo de las reglas del negocio.

Fin de la Trayectoria Alternativa D.

5.2.4. CU4 Descargar un archivo

Caso de Uso: CU4 Descargar un Archivo	
Concepto	Descripción
Actor:	Usuario.
Propósito:	El usuario podrá descargar del sistema el archivo que desee.
Entradas:	Ninguna
Salidas:	Archivo descargado
Pre-condiciones:	El Usuario debe contar con una sesión válida. El Usuario debe encontrarse en PP1 Pantalla Principal .
Post-condiciones:	Ninguna
Reglas de negocio:	RN-A7. Cola de bajada.
Errores:	Se solicitaron más del límite de descargas permitidas por usuario.

Trayectoria Principal

1.  Se ubica dentro de la carpeta donde se encuentra(n) el(los) archivo(s) a descargar(se).
2.  Selecciona el(los) archivo(s) a descargar(se).
3.  Da clic en el botón Descargar.
4.  Verifica **RN-A7. Cola de bajada.** [Trayectoria Alternativa A].
5.  Despliega **PMSG2 Mensaje de información** con el mensaje "Archivo(s) descargado(s) correctamente" y despliega **PP1 Pantalla Principal**.

Fin de la Trayectoria Principal.

Trayectoria Alternativa A. Sobrepasso del límite de descargas permitidas

1.  Despliega **PMSG2 Mensaje de información** con el mensaje "No se pueden descargar más de 4 archivos a la vez".
2.  Despliega **PP1 Pantalla Principal**

Fin de la Trayectoria Alternativa A.

5.2.5. CU5 Enlistar Archivos y Carpetas

Caso de Uso: CU5 Enlistar Archivos y Carpetas.	
Concepto	Descripción
Actor:	Usuario.
Propósito:	Este caso de uso permite al usuario visualizar sus Archivos y Carpetas existentes en el sistema.
Entradas:	Ninguna.
Salidas:	PP1 Pantalla Principal
Pre-condiciones:	El Usuario debe contar con una sesión válida.
Post-condiciones:	Ninguna.
Reglas de negocio:	Ninguna.
Errores:	Ninguno.

Trayectoria Principal

1.  Da clic en el botón “Inicio”.
2.  Despliega **PP1 Pantalla Principal**.

Fin de la Trayectoria Principal.

Casos de uso que extienden:

- CU6 Eliminar Archivo.
- CU7 Eliminar Carpeta.
- CU8 Mover Archivo.
- CU9 Mover Carpeta.
- CU10 Crear Carpeta.
- CU11 Renombrar Archivo.
- CU12 Renombrar Carpeta.

5.2.6. CU6 Eliminar Archivo

Caso de Uso: CU6 Eliminar Archivo	
Concepto	Descripción
Actor:	Usuario.
Propósito:	Eliminar un archivo perteneciente al usuario del sistema.
Entradas:	Ninguna
Salidas:	Archivo eliminado.
Pre-condiciones:	El Usuario debe haber iniciado sesión en el sistema El Usuario debe encontrarse en PP1 Pantalla Principal .
Post-condiciones:	El archivo será eliminado del sistema.
Reglas de negocio:	Ninguna.
Errores:	Ninguno.

Trayectoria Principal

1.  Se ubica en la carpeta donde se encuentra el archivo a borrar.
2.  Selecciona el archivo a eliminar.
3.  Da clic sobre el botón de “Eliminar Archivo”.
4.  Despliega **PMSG1 Mensaje de confirmación**. [Trayectoria Alternativa A].
5.  Confirma la eliminación del archivo.
6.  Elimina el archivo.
7.  Despliega **PMSG2 Mensaje de información** con el mensaje “Archivo eliminado” y despliega **PP1 Pantalla Principal**.

Fin de la Trayectoria Principal.

Trayectoria Alternativa A. Eliminación cancelada.

1.  Cancela la operación de eliminación.
2.  Despliega **PMSG2 Mensaje de información** con el mensaje “Se canceló la eliminación del archivo”.
3.  Despliega **PP1 Pantalla Principal**.

5.2.7. CU7 Eliminar Carpeta

Caso de Uso: CU7 Eliminar Carpeta.	
Concepto	Descripción
Actor:	Usuario.
Propósito:	Este caso de uso permite al actor eliminar una carpeta del sistema, junto con los archivos que contiene.
Entradas:	Carpeta a ser eliminada.
Salidas:	Ninguna.
Pre-condiciones:	El Usuario debe contar con una sesión válida. El Usuario se debe encontrar en PP1 Pantalla Principal .
Post-condiciones:	La Carpeta es eliminada del sistema.
Reglas de negocio:	Ninguna.
Errores:	Ninguno.

Trayectoria Principal

1. ♂ Da clic sobre el botón de “Eliminar Carpeta”, de la Carpeta que desea eliminar.
2. Muestra **PMSG1 Mensaje de confirmación** con el mensaje “¿Desea eliminar la Carpeta?. [Trayectoria Alternativa A].
3. ♂ Da clic sobre el botón de “Aceptar”.
4. Elimina la carpeta.
5. Despliega **PMSG2 Mensaje de información** con el mensaje “Carpeta eliminada” y despliega **PP1 Pantalla Principal**.

Fin de la Trayectoria Principal.

Trayectoria Alternativa A. Eliminación cancelada.

1. ♂ Da clic sobre el botón de “Cancelar”.
2. Despliega **PMSG2 Mensaje de información** con el mensaje “Se canceló la eliminación de la carpeta”.
3. Despliega **PP1 Pantalla Principal**.

Fin de la Trayectoria Alternativa A.

5.2.8. CU8 Mover Archivo

Caso de Uso: CU8 Mover Archivo.	
Concepto	Descripción
Actor:	Usuario.
Propósito:	Este caso de uso permite al Usuario mover un archivo de la lista de archivos que pertenece a dicho Usuario.
Entradas:	Archivo que se moverá.
Salidas:	Archivo movido.
Pre-condiciones:	El Usuario debe contar con una sesión válida. El Usuario se debe encontrar en PP1 Pantalla Principal .
Post-condiciones:	El archivo es movido.
Reglas de negocio:	RN-A9. Unicidad en el nombre del Archivo.
Errores:	El archivo no se puede mover debido a que ya existe uno con el mismo nombre.

Trayectoria Principal

1.  Se ubica sobre algún archivo y da clic sobre el botón de “Mover archivo”.
2.  Despliega ventana para mover el archivo.
3.  Selecciona el directorio a donde se quiere mover el archivo.
4.  Da clic en el botón Aceptar.
5.  Verifica **RN-A9. Unicidad en el nombre del Archivo.** [Trayectoria Alternativa A].
6.  Despliega **PMSG2 Mensaje de información** con el mensaje “Archivo movido correctamente” y despliega **PP1 Pantalla Principal**.

Fin de la Trayectoria Principal.

Trayectoria Alternativa A. Archivo repetido.

1.  Despliega **PMSG2 Mensaje de información** con el mensaje “*No se puede mover el archivo dado que ya existe otro con el mismo nombre*” .
2.  Despliega **PP1 Pantalla Principal**.

Fin de la Trayectoria Alternativa A.

5.2.9. CU9 Mover Carpeta

Caso de Uso: CU9 Mover Carpeta.	
Concepto	Descripción
Actor:	Usuario.
Propósito:	Este caso de uso permite al Usuario mover una carpeta a otra ubicación.
Entradas:	Ninguna.
Salidas:	Carpeta movida.
Pre-condiciones:	El Usuario debe contar con una sesión válida. El Usuario se debe encontrar en PP1 Pantalla Principal .
Post-condiciones:	La carpeta es movida.
Reglas de negocio:	RN-CA1. Unicidad en el nombre de las Carpetas.
Errores:	La carpeta puede ser movida debido que existe otra con otro nombre en la carpeta destino.

Trayectoria Principal

1. ⚒ Se ubica sobre algún archivo y da clic sobre el botón de “Mover archivo”.
2. Despliega ventana para mover la carpeta.
3. ⚒ Se escoge la ubicación donde se desea reubicar la carpeta.
4. ⚒ Da clic en el botón “Aceptar”.
5. Verifica RN-CA1. Unicidad en el nombre de las Carpetas [Trayectoria Alternativa A].
6. Despliega PMSG2 Mensaje de información con el mensaje “Carpeta movida correctamente” y despliega PP1 Pantalla Principal.

Fin de la Trayectoria Principal.

Trayectoria Alternativa A. No se puede mover la carpeta.

1. Despliega PMSG2 Mensaje de información con el mensaje “No se puede mover la carpeta debido a que existe otra con el mismo nombre”.
2. Despliega PP1 Pantalla Principal.

Fin de la Trayectoria Alternativa A.

5.2.10. CU10 Crear Carpeta

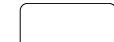
Caso de Uso: CU10 Crear Carpeta.	
Concepto	Descripción
Actor:	Usuario.
Propósito:	Este caso de uso permite al actor crear una carpeta dentro del sistema.
Entradas:	Nombre de la Carpeta.
Salidas:	Ninguna.
Pre-condiciones:	El Usuario debe contar con una sesión válida. El Usuario se debe encontrar en PP1 Pantalla Principal .
Post-condiciones:	La Carpeta es creada en el sistema.
Reglas de negocio:	RN-CA1. Unicidad en el nombre de las Carpetas. RN-CA2. Longitud del nombre de Carpeta. RN-CA3. Carácter barra inversa. RN-CA4. Nombre de la Carpeta.
Errores:	Error en el nombre de la Carpeta.

Trayectoria Principal

1.  Da clic sobre el botón de “Crear Carpeta”, de la **PP1 Pantalla Principal**.
2.  Despliega la pantalla **PPC2 Nueva Carpeta**.
3.  Escribe el nombre de la Carpeta y da clic en el botón “Crear”.
4.  Verifica **RN-CA2. Longitud del nombre de Carpeta**, **RN-CA3. Carácter barra inversa**, **RN-CA4. Nombre de la Carpeta**. [Trayectoria Alternativa A].
5.  Verifica **RN-CA1. Unicidad en el nombre de las Carpetas**. [Trayectoria Alternativa B].
6.  Despliega **PMSG2 Mensaje de información** con el mensaje “Carpeta creada correctamente” y despliega **PP1 Pantalla Principal**.

Fin de la Trayectoria Principal.

Trayectoria Alternativa A. Nombre de Carpeta Inválido

1.  Despliega **PMSG2 Mensaje de información** con el mensaje “Introduce un nombre de carpeta válido”.
2.  Regresa al paso 2 de la Trayectoria Principal.

Trayectoria Alternativa B. Nombre de Carpeta repetido

1.  Despliega **PMSG2 Mensaje de información** con el mensaje “Ya existe una carpeta con el mismo nombre”.
2.  Regresa al paso 2 de la Trayectoria Principal.

5.2.11. CU11 Renombrar Archivo

Caso de Uso: CU11 Renombrar Archivo.	
Concepto	Descripción
Actor:	Usuario.
Propósito:	Este caso de uso permite al Usuario renombrar un archivo de la lista de archivos que pertenece a dicho Usuario.
Entradas:	Archivo a ser renombrado.
Salidas:	Archivo renombrado.
Pre-condiciones:	El Usuario debe contar con una sesión válida. El Usuario se debe encontrar en la PP1 Pantalla Principal .
Post-condiciones:	El archivo es renombrado.
Reglas de negocio:	RN-A2. Longitud del nombre del Archivo. RN-A3. Carácter barra inversa. RN-A4. Nombre del Archivo. RN-A5. Tipo de Archivo. RN-A9. Unicidad en el nombre del Archivo.
Errores:	El archivo no se puede renombrar debido a que no se cumple alguna regla de negocio.

Trayectoria Principal

1. ♂ Se ubica sobre algún archivo y da clic sobre el botón de “Renombrar archivo”.
2. Despliega la pantalla **PPC2 Renombrar Archivo**.
3. ♂ Ingresa el nombre del Archivo.
4. ♂ Da clic en el botón “Aceptar”.
5. Verifica RN-A2. Longitud del nombre del Archivo, RN-A3. Carácter barra inversa, RN-A4. Nombre del Archivo, RN-A5. Tipo de Archivo, RN-A9. Unicidad en el nombre del Archivo. [Trayectoria Alternativa A].
6. Despliega la **PP1 Pantalla Principal**.

Fin de la Trayectoria Principal.

Trayectoria Alternativa A. No se puede renombrar archivo.

1. Despliega **PMSG2 Mensaje de información** con el mensaje “No se puede renombrar el archivo” especificando por qué no se pudo de acuerdo con las reglas de negocio que no se cumplen.

2. Despliega la **PP1 Pantalla Principal**.

Fin de la Trayectoria Alternativa A.

5.2.12. CU12 Renombrar Carpeta

Caso de Uso: CU12 Renombrar carpeta	
Concepto	Descripción
Actor:	Usuario.
Propósito:	Cambiar el nombre de una carpeta.
Entradas:	Nombre de la carpeta
Salidas:	Nombre de la carpeta actualizado
Pre-condiciones:	El usuario deberá haber iniciado sesión. El Usuario se debe encontrar en la PP1 Pantalla Principal .
Post-condiciones:	Carpeta renombrada.
Reglas de negocio:	RN-CA1. Unicidad en el nombre de las Carpetas RN-CA2. Longitud del nombre de Carpeta RN-CA3. Carácter barra inversa RN-CA4. Nombre de la Carpeta
Errores:	Nombre no válido de carpeta debido a una regla de negocio.

Trayectoria Principal

1.  Se ubica en la ruta de la Carpeta a renombrar.
2.  Selecciona la Carpeta a renombrar.
3.  Da clic en el botón “Renombrar”.
4. Despliega la pantalla **PPC1 Renombrar Carpeta**.
5.  Introduce el nuevo nombre de la Carpeta.
6. Verifica **RN-CA1. Unicidad en el nombre de las Carpetas RN-CA2. Longitud del nombre de Carpeta RN-CA3. Carácter barra inversa RN-CA4. Nombre de la Carpeta**[Trayectoria Alternativa A].
7. Despliega la **PP1 Pantalla Principal**.

Fin de la Trayectoria Principal.

Trayectoria Alternativa A. No se puede renombrar carpeta.

1. Despliega **PMSG2 Mensaje de información** con el mensaje “*No se puede renombrar la carpeta*”, especificando por qué no se pudo renombrar de acuerdo con las reglas de negocio.
2. Despliega la **PP1 Pantalla Principal**.

5.2.13. CU13 Copiar Archivo

Caso de Uso: CU13 Copiar Archivo	
Concepto	Descripción
Actor:	Usuario.
Propósito:	Permite al Usuario copiar un archivo en la misma Carpeta.
Entradas:	Archivo a copiar.
Salidas:	Archivo copiado.
Pre-condiciones:	El Usuario debe contar con una sesión válida. El Usuario debe encontrarse en la PP1 Pantalla Principal .
Post-condiciones:	El archivo seleccionado se copia.
Reglas de negocio:	RN-A2. Longitud del nombre del Archivo. RN-A3. Carácter barra inversa. RN-A4. Nombre del Archivo. RN-A5. Tipo de Archivo. RN-A8. Capacidad de almacenamiento. RN-A9. Unicidad en el nombre del Archivo.
Errores:	El Usuario ha alcanzado la capacidad máxima de espacio asignada. El nombre de Archivo es inválido, debido a que no cumple con las reglas del negocio.

Trayectoria Principal

1.  Selecciona el Archivo el cual desea copiar.
2.  Da clic en el botón “Copiar”
3. Verifica **RN-A8. Capacidad de almacenamiento.** **Trayectoria Alternativa A.**
4. Despliega **PPC4 Copiar Archivo**.
5.  Ingresa el nombre del Archivo.

6. Verifica RN-A2. Longitud del nombre del Archivo, RN-A3. Carácter barra inversa, RN-A4. Nombre del Archivo, RN-A5. Tipo de Archivo, RN-A9. Unicidad en el nombre del Archivo. Trayectoria Alternativa B.
7. Crea una copia del Archivo seleccionado en la misma Carpeta, con el nombre ingresado por el Usuario.
8. Despliega la PP1 Pantalla Principal.

Fin de la Trayectoria Principal.

Trayectoria Alternativa A. Espacio insuficiente.

1. Despliega PMSG2 Mensaje de información con el mensaje “No cuenta con suficiente espacio para realizar una copia”, y despliega la PP1 Pantalla Principal.

Fin de la Trayectoria Alternativa A.

Trayectoria Alternativa B. Nombre inválido.

1. Despliega PMSG2 Mensaje de información con el mensaje “No se puede copiar el archivo” especificando por qué no se pudo de acuerdo con las reglas de negocio que no se cumplen.
2. Despliega la PP1 Pantalla Principal.

Fin de la Trayectoria Alternativa B.

5.2.14. CU14 Modificar cuenta de usuario

Caso de Uso: CU14 Modificar cuenta de Usuario.	
Concepto	Descripción
Actor:	Usuario.
Propósito:	Este caso de uso permite al usuario visualizar y modificar su propia cuenta.
Entradas:	Ninguna.
Salidas:	PG1 Visualizar detalles de Usuario.
Pre-condiciones:	El Usuario debe contar con una sesión válida.
Post-condiciones:	Ninguna.
Reglas de negocio:	Ninguna.
Errores:	Ninguno.

Trayectoria Principal

1.  Da clic en el botón “Modificar cuenta de usuario”, de la PP1 Pantalla Principal.

2. Despliega la pantalla **PG1 Visualizar detalles de Usuario**.

Fin de la Trayectoria Principal.

Casos de uso que extienden:

- CU14 Eliminar Cuenta.
- CU15 Modificar Contraseña.

5.2.15. CU15 Eliminar cuenta

Caso de Uso: CU15 Eliminar Cuenta.	
Concepto	Descripción
Actor:	Usuario.
Propósito:	Este caso de uso permite al Usuario eliminar su cuenta del sistema, al eliminar su cuenta se elimina consigo sus Carpetas y sus Archivos.
Entradas:	Ninguna.
Salidas:	Ninguna.
Pre-condiciones:	El Usuario debe contar con una sesión válida. El Usuario se debe encontrar en la pantalla PG2 Modificar Usuario .
Post-condiciones:	El Usuario, sus Carpetas y Archivos son eliminados del sistema.
Reglas de negocio:	Ninguna.
Errores:	Ninguno.

Trayectoria Principal

1.  Da clic sobre el botón “Eliminar cuenta” en la Pantalla de Gestión de Cuenta.
2. Despliega **PMSG1 Mensaje de confirmación** con el mensaje “¿Desea eliminar su cuenta? [Trayectoria Alternativa A].
3. Cierra la sesión y elimina la cuenta del Usuario.
4. Muestra la pantalla de **PI1 Inicio de Sesión**.

Fin de la Trayectoria Principal.

Trayectoria Alternativa A. Eliminación de cuenta cancelada.

1.  Da clic sobre el botón “Cancelar”.
2. Despliega **PMSG2 Mensaje de información** con el mensaje “Se canceló la eliminación de la cuenta”

3. Muestra la pantalla **PG2 Modificar Usuario**.

Fin de la Trayectoria Alternativa A.

5.2.16. CU16 Modificar contraseña

Caso de Uso: CU16 Modificar contraseña.	
Concepto	Descripción
Actor:	Usuario.
Propósito:	Este caso de uso permite al Usuario modificar su contraseña del sistema.
Entradas:	Contraseña Actual. Nueva Contraseña.
Salidas:	Ninguna.
Pre-condiciones:	El Usuario debe contar con una sesión válida. El Usuario se debe encontrar en la pantalla PG1 Visualizar detalles de Usuario .
Post-condiciones:	La contraseña es modificada.
Reglas de negocio:	RN-C1. Caracteres válidos en la contraseña del Usuario RN-C2. Longitud de la contraseña del Usuario RN-C3. Robustez de la contraseña del Usuario RN-C4. Contraseña del Usuario diferente del nombre de Usuario. RN-I2. Contraseña del Usuario válida.
Errores:	La nueva contraseña es errónea, no cumple con las reglas de negocio para contraseñas.

Trayectoria Principal

1.  Da clic sobre el botón “Modificar Cuenta” en la pantalla **PG1 Visualizar detalles de Usuario**.
2. Sigue la trayectoria alternativa A.
3.  Introduce la contraseña actual.
4. Verifica la contraseña actual acorde con **RN-I2. Contraseña del Usuario válida. [Trayectoria Alternativa A]**.
5. Despliega **PG2 Modificar Usuario**.
6.  Introduce la nueva contraseña y su confirmación.

7. Verifica la nueva contraseña acorde con **RN-C1**. Caracteres válidos en la contraseña del Usuario, **RN-C2**. Longitud de la contraseña del Usuario, **RN-C3**. Robustez de la contraseña del Usuario, **RN-C4**. Contraseña del Usuario diferente del nombre de Usuario. [Trayectoria Alternativa B].
8.  El usuario da clic en *Cambiar contraseña*.
9. Se actualiza la contraseña.
10. Muestra **PMSG2 Mensaje de información** con el mensaje “Contraseña cambiada correctamente”.

Fin de la Trayectoria Principal.

Trayectoria Alternativa A. Contraseña actual errónea

1.  Despliega **PMSG2 Mensaje de información** con el mensaje “*Contraseña actual errónea*”.
2. Muestra la pantalla **PG1 Visualizar detalles de Usuario**.

Fin de la Trayectoria Alternativa A.

Trayectoria Alternativa B. Nueva contraseña errónea

1.  Despliega **PMSG2 Mensaje de información** con el mensaje “*Nueva Contraseña errónea*”, especificando las reglas del negocio que no cumple.
2. Muestra la pantalla **PG1 Visualizar detalles de Usuario**.

Fin de la Trayectoria Alternativa B.

5.2.17. CU17 Modificar Nombre de Usuario

Caso de Uso: CU17 Modificar Nombre de Usuario.	
Concepto	Descripción
Actor:	Usuario.
Propósito:	Este caso de uso permite al Usuario modificar su Nombre de Usuario.
Entradas:	Contraseña actual. Nueva Nombre de Usuario.
Salidas:	Ninguna.
Pre-condiciones:	El Usuario debe contar con una sesión válida. El Usuario se debe encontrar en la pantalla PG1 Visualizar detalles de Usuario .
Post-condiciones:	El Nombre de Usuario es modificado.
Reglas de negocio:	RN-U1. Unicidad del nombre de Usuario. RN-U2. Espacios en el nombre de Usuario. RN-U3. Longitud del nombre de Usuario. RN-U4. Caracteres válidos en el nombre de Usuario. RN-U5. Caracteres no válidos en el nombre de Usuario. RN-U6. Caracteres no válidos al inicio en el nombre de Usuario. RN-U7. Caracteres repetidos no válidos en el nombre de Usuario.
Errores:	El nuevo Nombre de Usuario no cumple con las reglas del negocio.

Trayectoria Principal

1. ♂ Da clic sobre el botón “*Modificar Cuenta*” en la pantalla **PG1 Visualizar detalles de Usuario**.
2. Solicita la contraseña actual.
3. ♂ Introduce la contraseña actual.
4. Verifica la contraseña actual acorde con **RN-I2. Contraseña del Usuario válida**. [Trayectoria Alternativa A].
5. Despliega **PG2 Modificar Usuario**.
6. ♂ Introduce el nuevo Nombre de Usuario y su confirmación.
7. Verifica el nombre de Usuario acorde con **RN-U1. Unicidad del nombre de Usuario**, **RN-U2. Espacios en el nombre de Usuario**, **RN-U3. Longitud del**

nombre de Usuario, RN-U4. Caracteres válidos en el nombre de Usuario, RN-U5. Caracteres no válidos en el nombre de Usuario, RN-U6. Caracteres no válidos al inicio en el nombre de Usuario, RN-U7. Caracteres repetidos no válidos en el nombre de Usuario. [Trayectoria Alternativa B].

8.  El usuario da clic en *Cambiar nombre de usuario*.
9.  Se actualiza el Nombre de Usuario.
10.  Muestra **PMSG2 Mensaje de información** con el mensaje “Nombre de Usuario cambiado correctamente”.

Fin de la Trayectoria Principal.

Trayectoria Alternativa A. Contraseña actual errónea

1.  Despliega **PMSG2 Mensaje de información** con el mensaje “*Contraseña actual errónea*”.
2.  Muestra la pantalla **PG1 Visualizar detalles de Usuario**.

Fin de la Trayectoria Alternativa A.

Trayectoria Alternativa B. Nueva Nombre de Usuario erróneo.

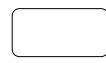
1.  Despliega **PMSG2 Mensaje de información** con el mensaje “*Nuevo Nombre de Usuario erróneo*”, especificando las reglas del negocio que no cumple.
2.  Muestra la pantalla **PG1 Visualizar detalles de Usuario**.

Fin de la Trayectoria Alternativa B.

5.2.18. CU18 Recuperar cuenta.

Caso de Uso: CU18 Recuperar cuenta	
Concepto	Descripción
Actor:	Usuario.
Propósito:	Permite al Usuario recuperar su cuenta, en caso de que haya olvidado su contraseña.
Entradas:	Nombre de Usuario.
Salidas:	Ninguna.
Pre-condiciones:	El Usuario a recuperar se encuentra registrado en el sistema. El actor se encuentra en la pantalla PI1 Inicio de Sesión .
Post-condiciones:	Se envía un enlace al correo asociado para recuperar la contraseña.
Reglas de negocio:	Ninguna.
Errores:	Ninguno.

Trayectoria Principal

1.  Da clic en el “Recuperar contraseña”, de la pantalla **PI1 Inicio de Sesión**.
2.  Despliega la pantalla **PI3 Recuperar contraseña**.
3.  Ingresa el Nombre de Usuario y da clic en el botón “Recuperar Contraseña”.
4.  Despliega **PMSG2 Mensaje de información** con el mensaje “En caso de existir la cuenta, se ha enviado un enlace de recuperación”.
5.  Verifica la existencia del correo y en dado caso, envía el enlace de recuperación al correo especificado.
6.  Despliega la pantalla **PI1 Inicio de Sesión**.

Fin de la Trayectoria Principal.

5.3. Arquitectura del sistema

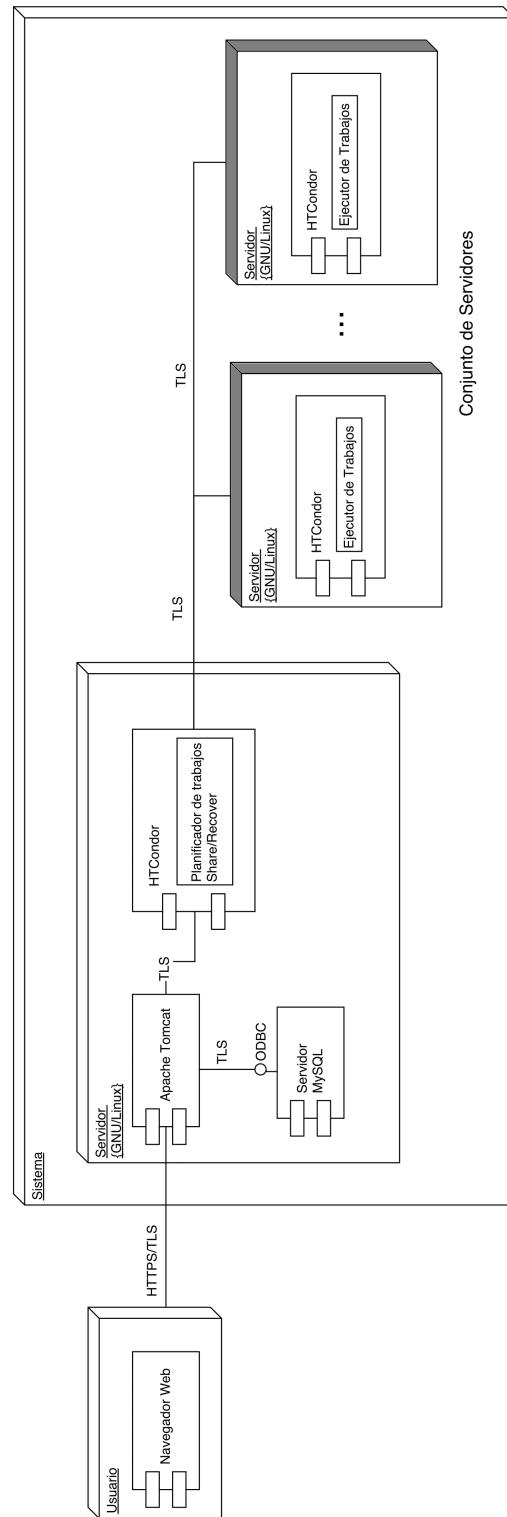


Figura 5.2: Diagrama de la Arquitectura del sistema.

5.4. Diseño de la base de datos

5.4.1. Diagrama Entidad-Relación

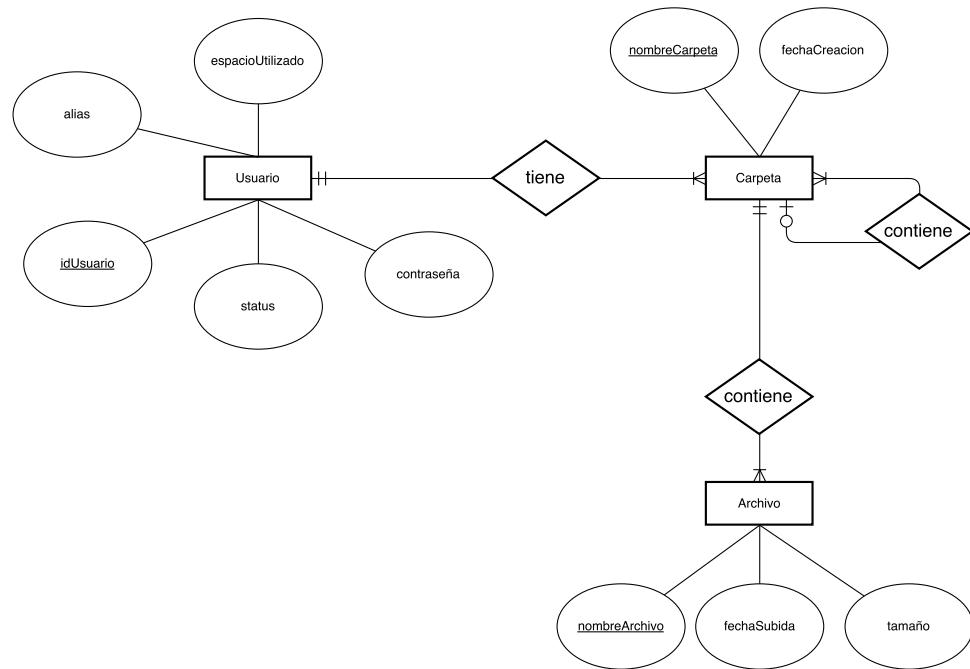


Figura 5.3: Diagrama Entidad-Relación del sistema.

5.4.2. Modelo Relacional

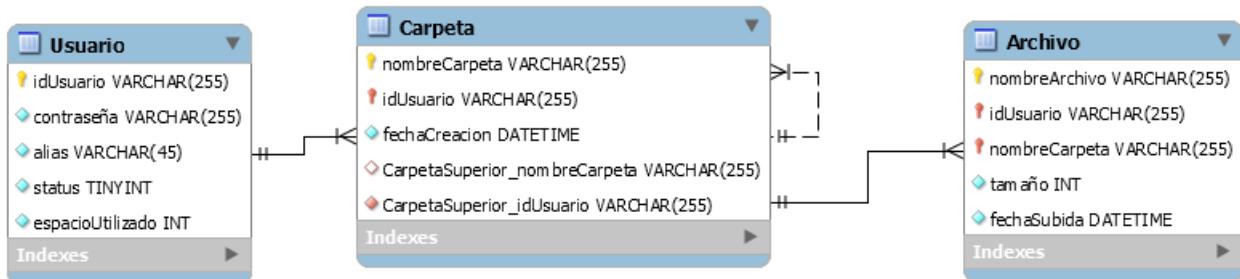


Figura 5.4: Modelo Relacional de la base de datos del sistema.

5.5. Diagrama de clases



Figura 5.5: Diagrama de clases del sistema.

5.6. Diagramas de secuencia

5.6.1. DS1 Registrar Cuenta de Usuario

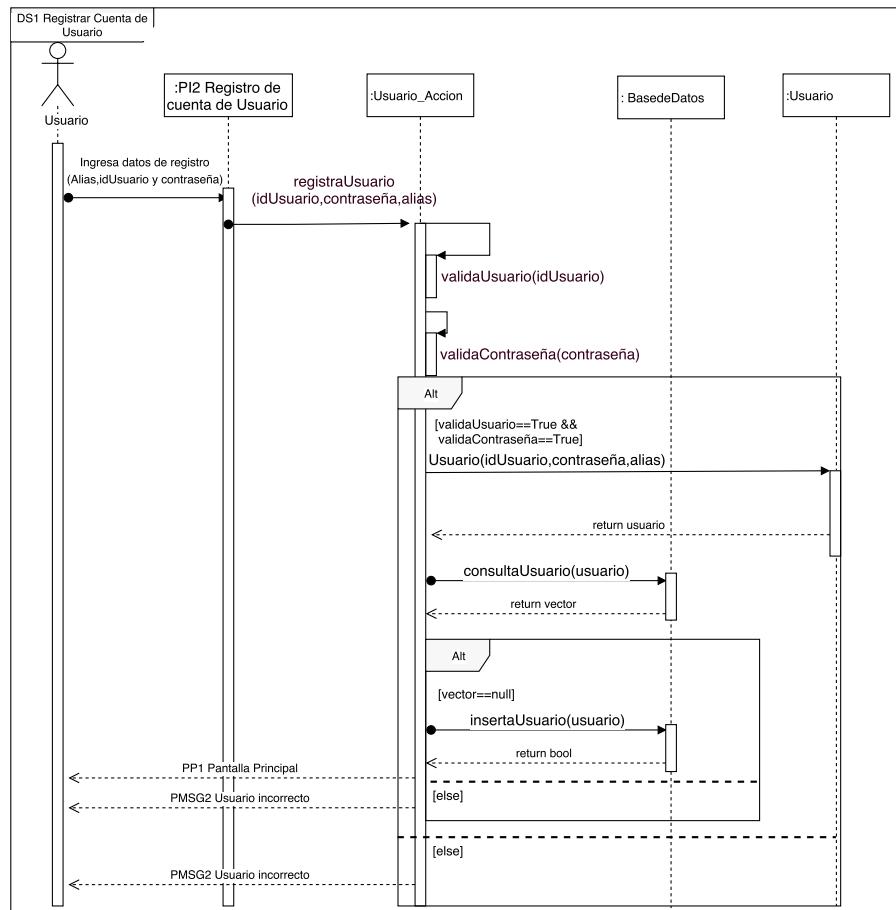


Figura 5.6: Diagrama de secuencia para Registrar Cuenta de Usuario.

5.6.2. DS2 Iniciar sesión

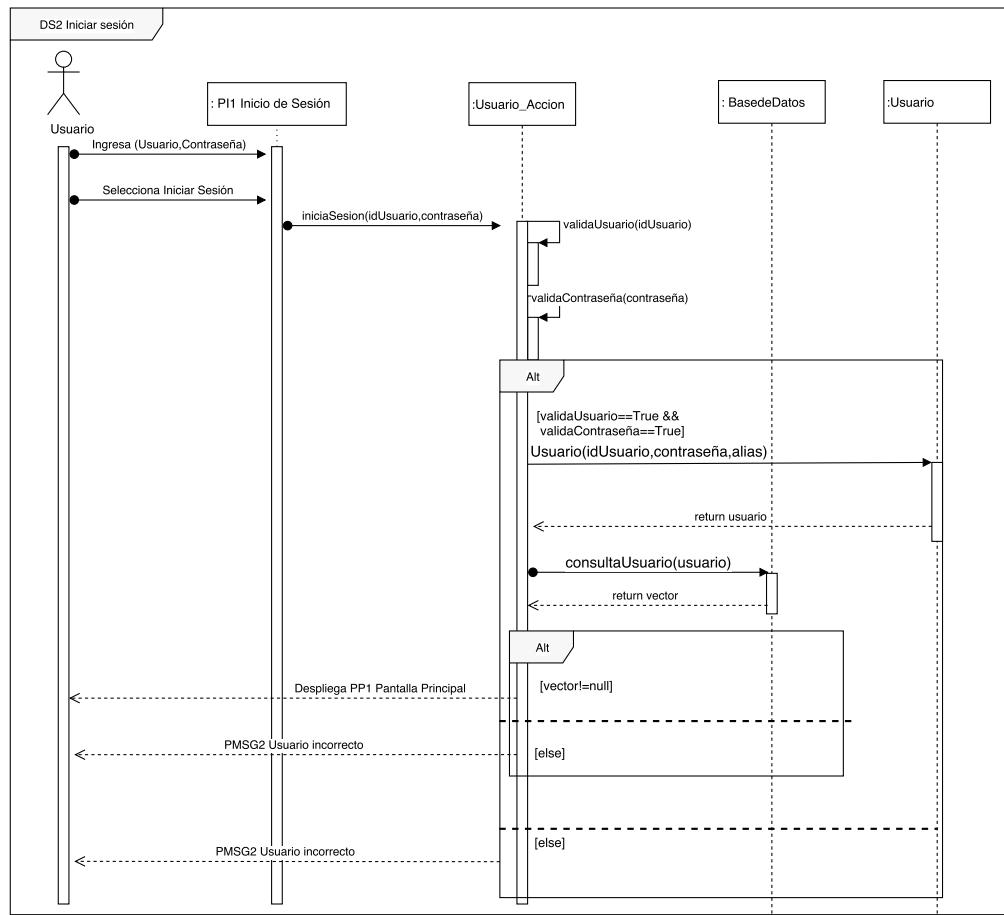


Figura 5.7: Diagrama de secuencia para Iniciar sesión.

5.6.3. DS3 Subir un archivo

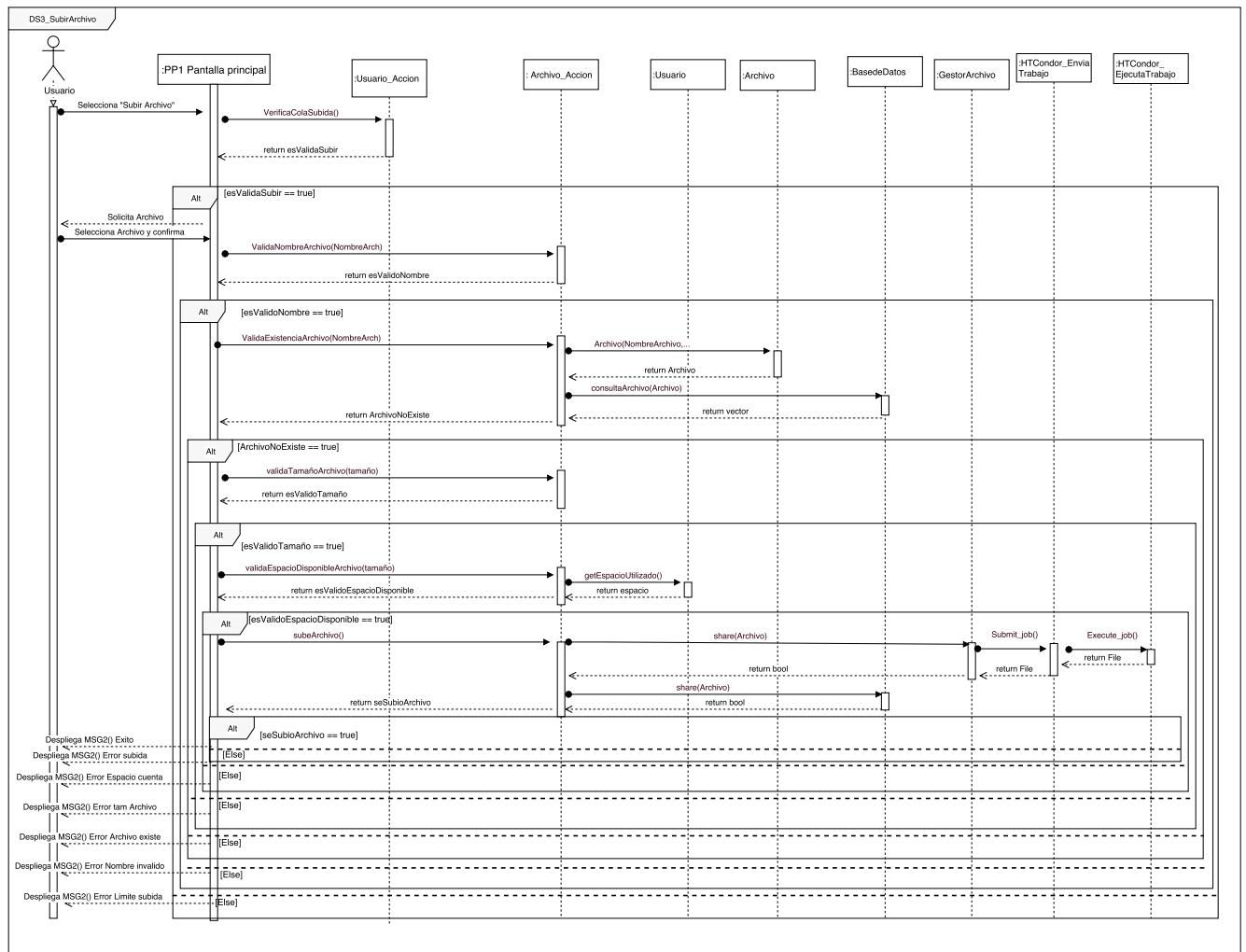


Figura 5.8: Diagrama de secuencia para Subir un archivo.

5.6.4. DS4 Descargar un archivo

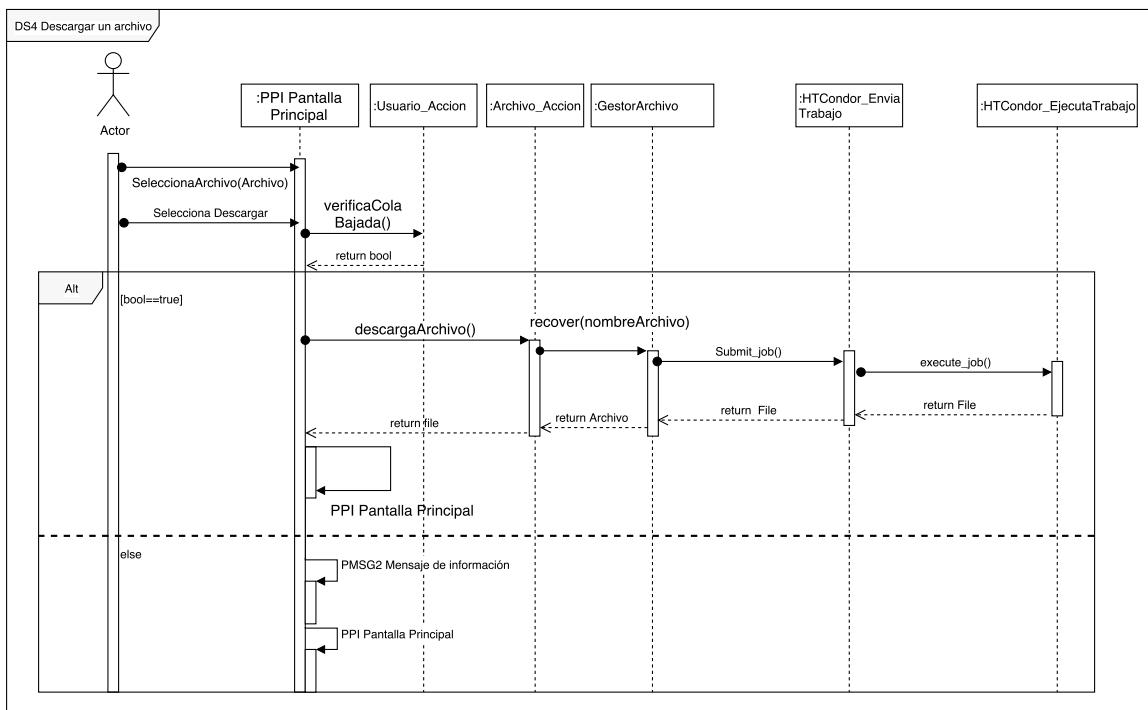


Figura 5.9: Diagrama de secuencia para Descargar un archivo.

5.6.5. DS6 Eliminar Archivo

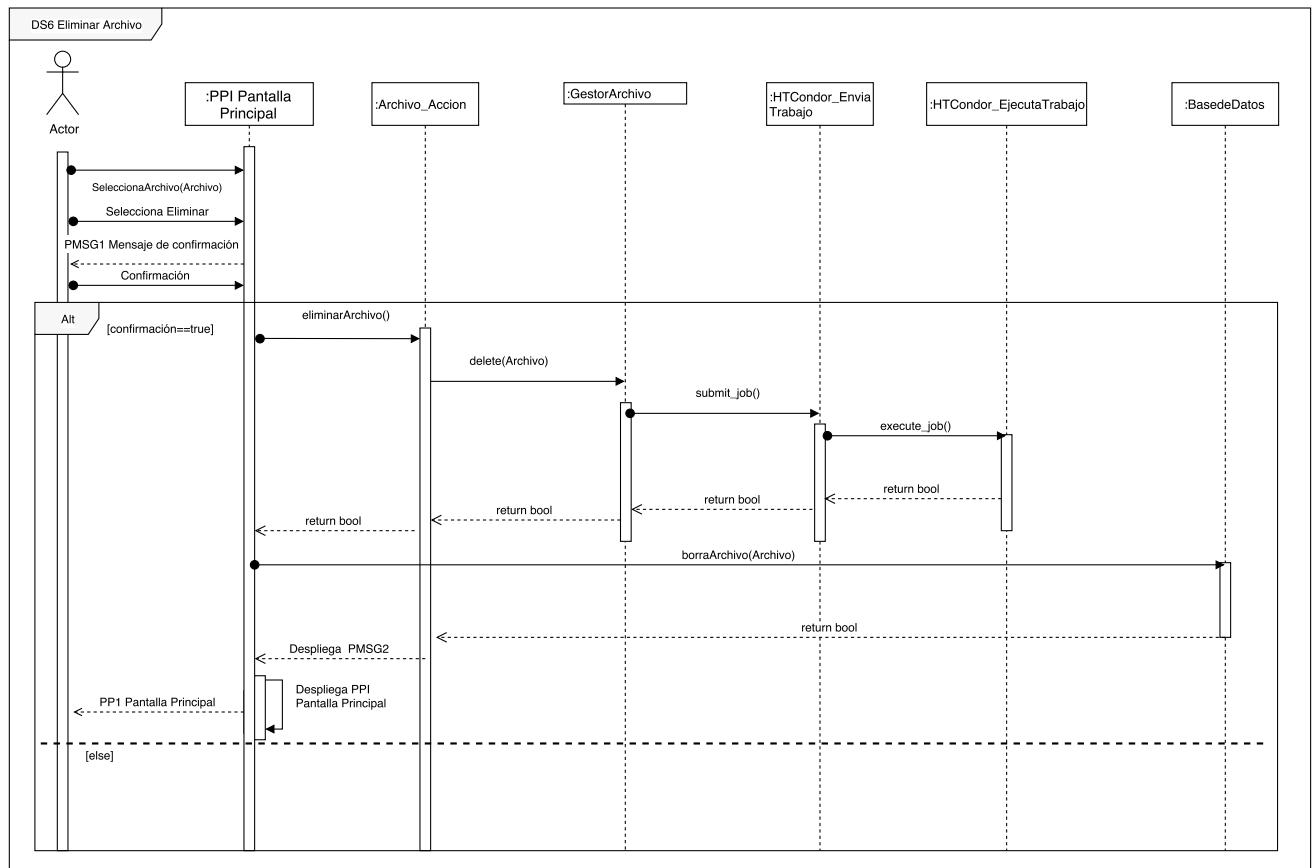


Figura 5.10: Diagrama de secuencia para Eliminar un Archivo.

5.6.6. DS7 Eliminar Carpeta

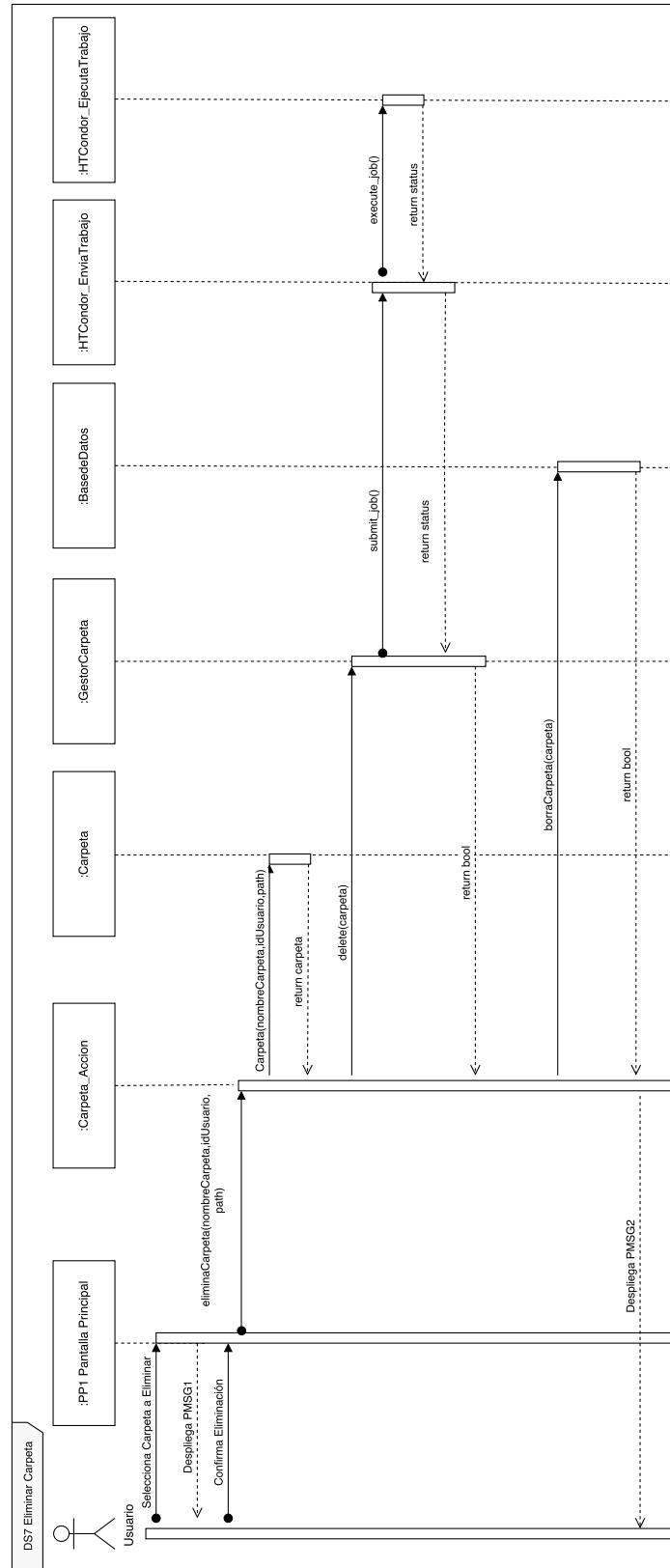


Figura 5.11: Diagrama de secuencia para Eliminar una Carpeta.

5.6.7. DS8 Mover Archivo

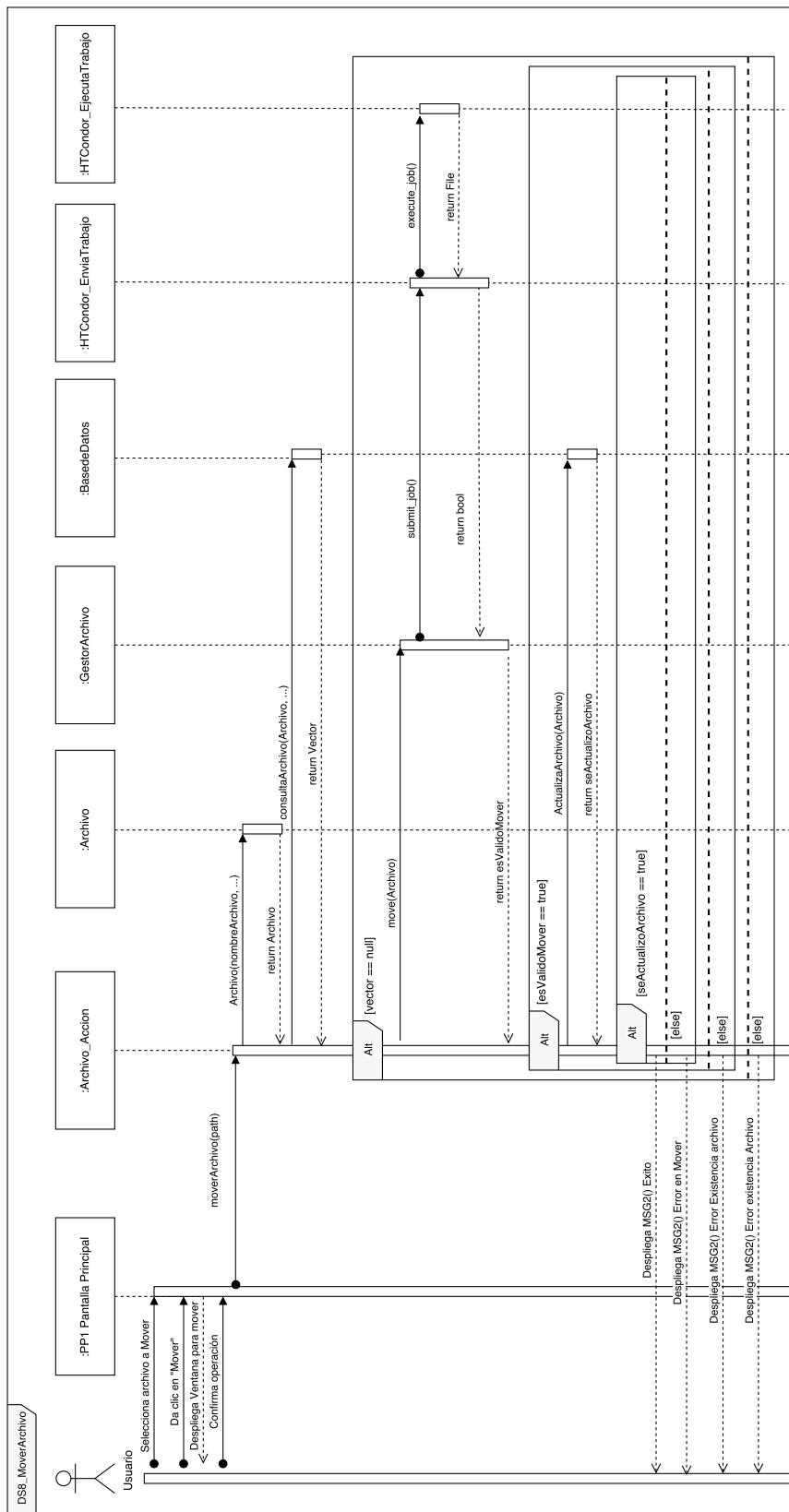


Figura 5.12: Diagrama de secuencia para Mover un Archivo.

5.6.8. DS9 Mover Carpeta

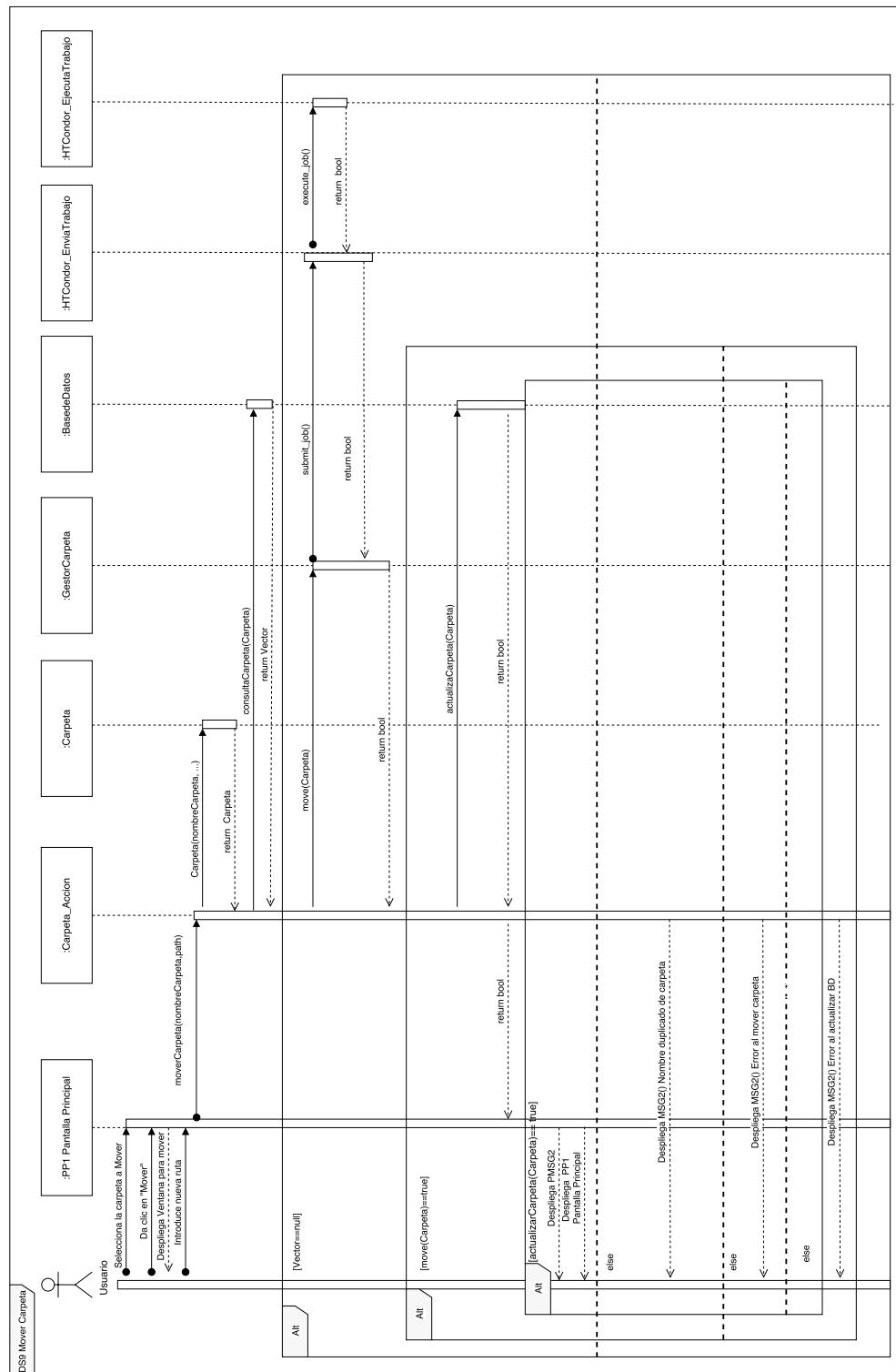


Figura 5.13: Diagrama de secuencia para Mover una Carpeta.

5.6.9. DS10 Crear Carpeta

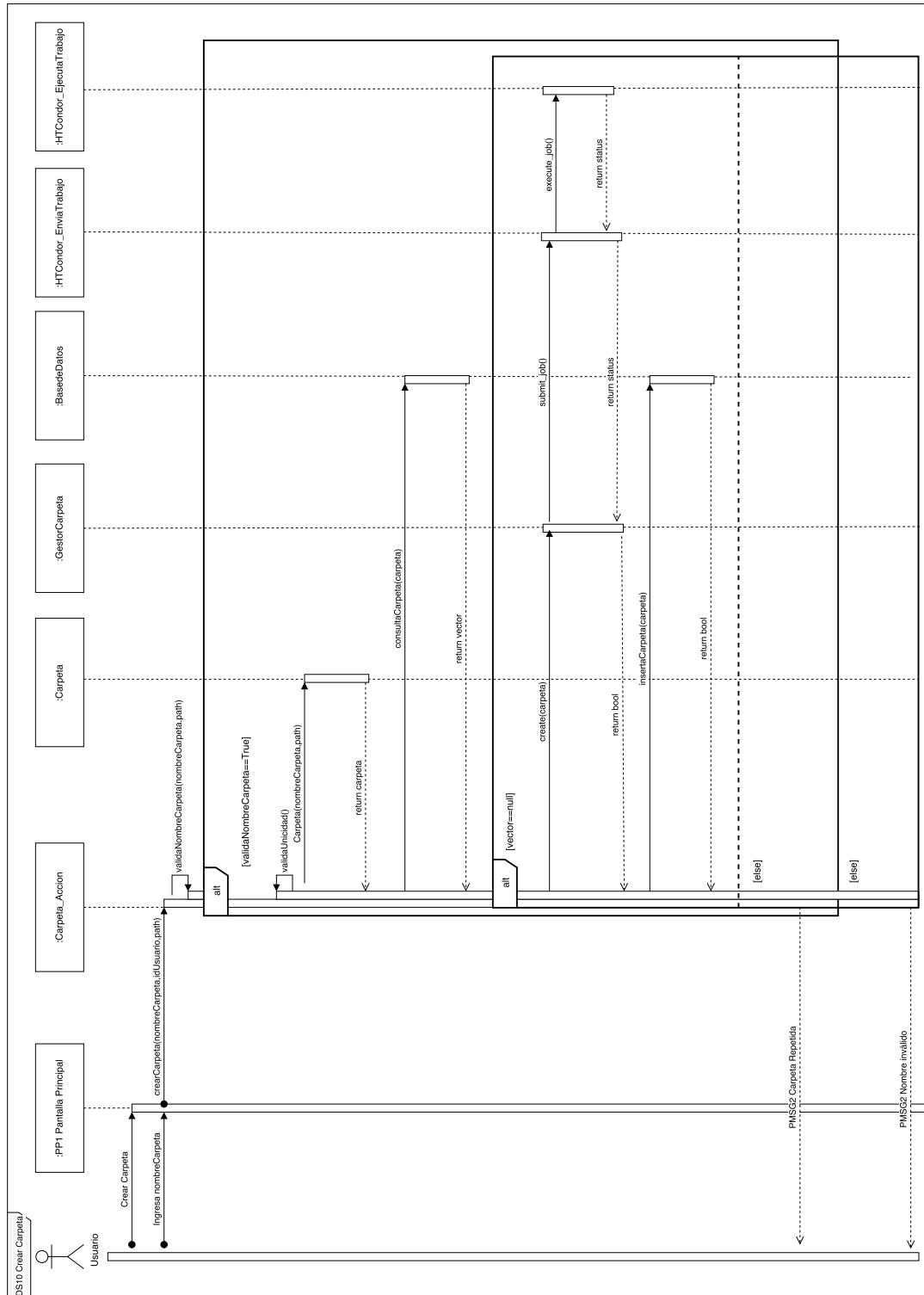


Figura 5.14: Diagrama de secuencia para Crear una Carpeta.

5.6.10. DS11 Renombrar Archivo

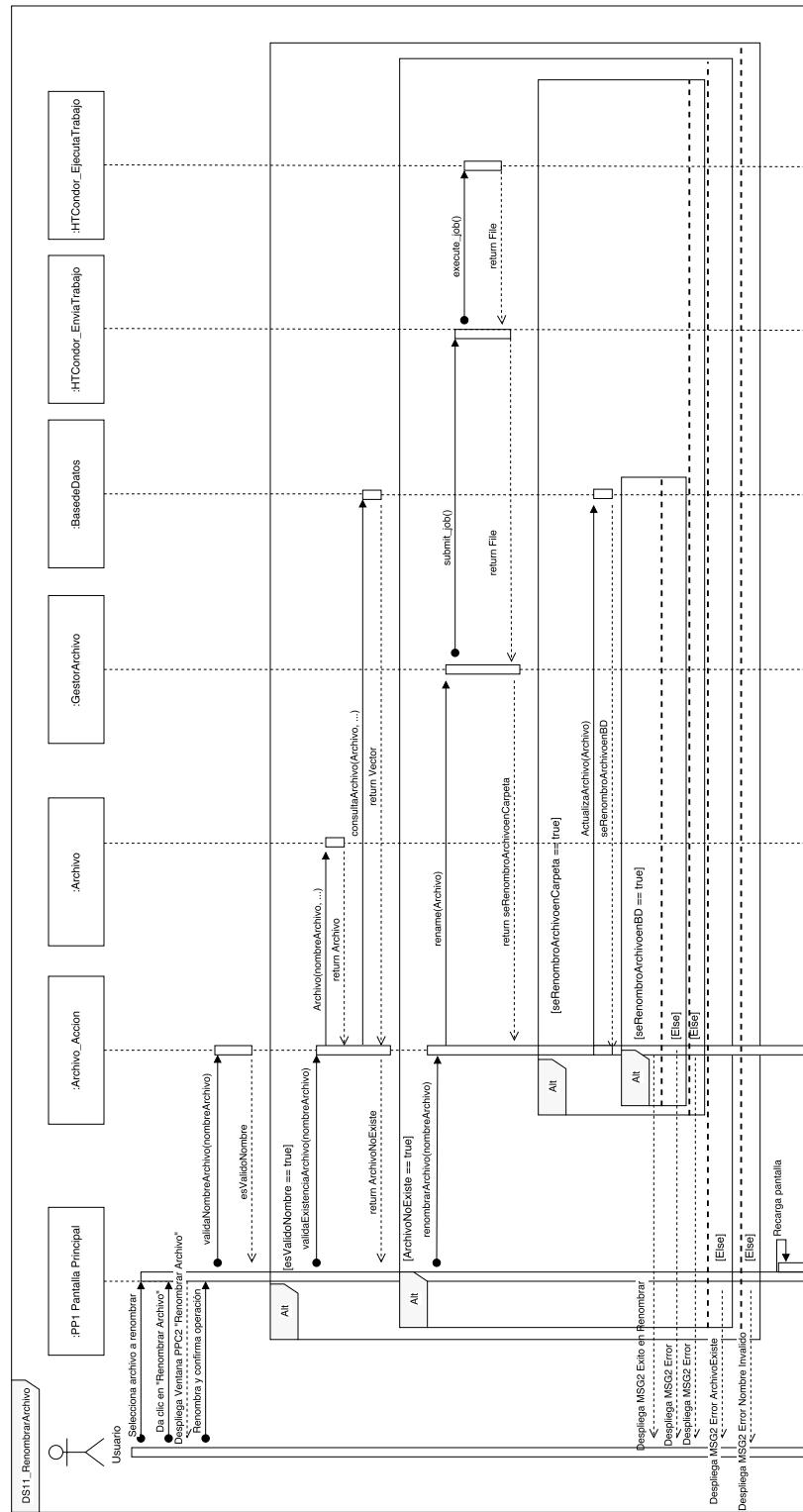


Figura 5.15: Diagrama de secuencia para Renombrar un Archivo.

5.6.11. DS12 Renombrar Carpeta

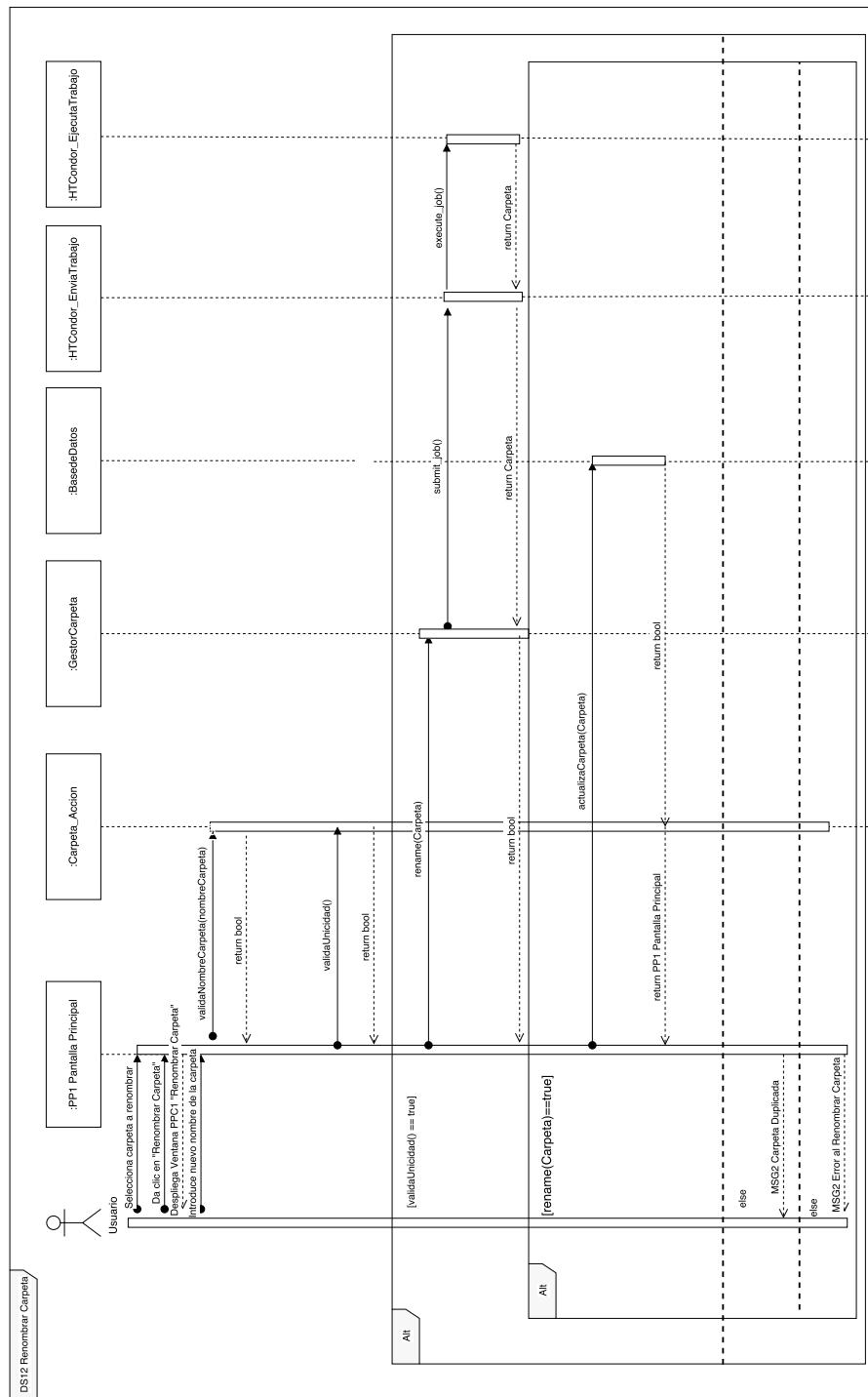


Figura 5.16: Diagrama de secuencia para Renombrar una Carpeta.

5.6.12. DS13 Copiar Archivo

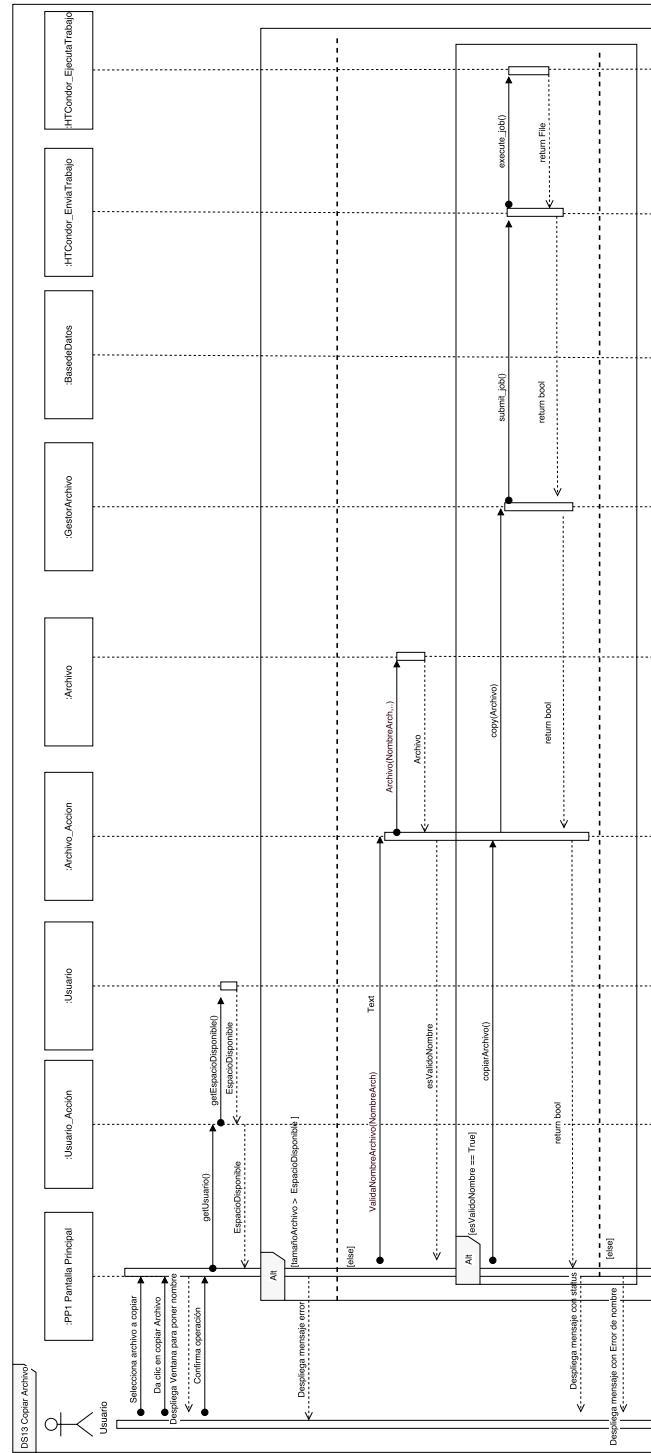


Figura 5.17: Diagrama de secuencia para Copiar un Archivo.

5.6.13. DS15 Eliminar cuenta

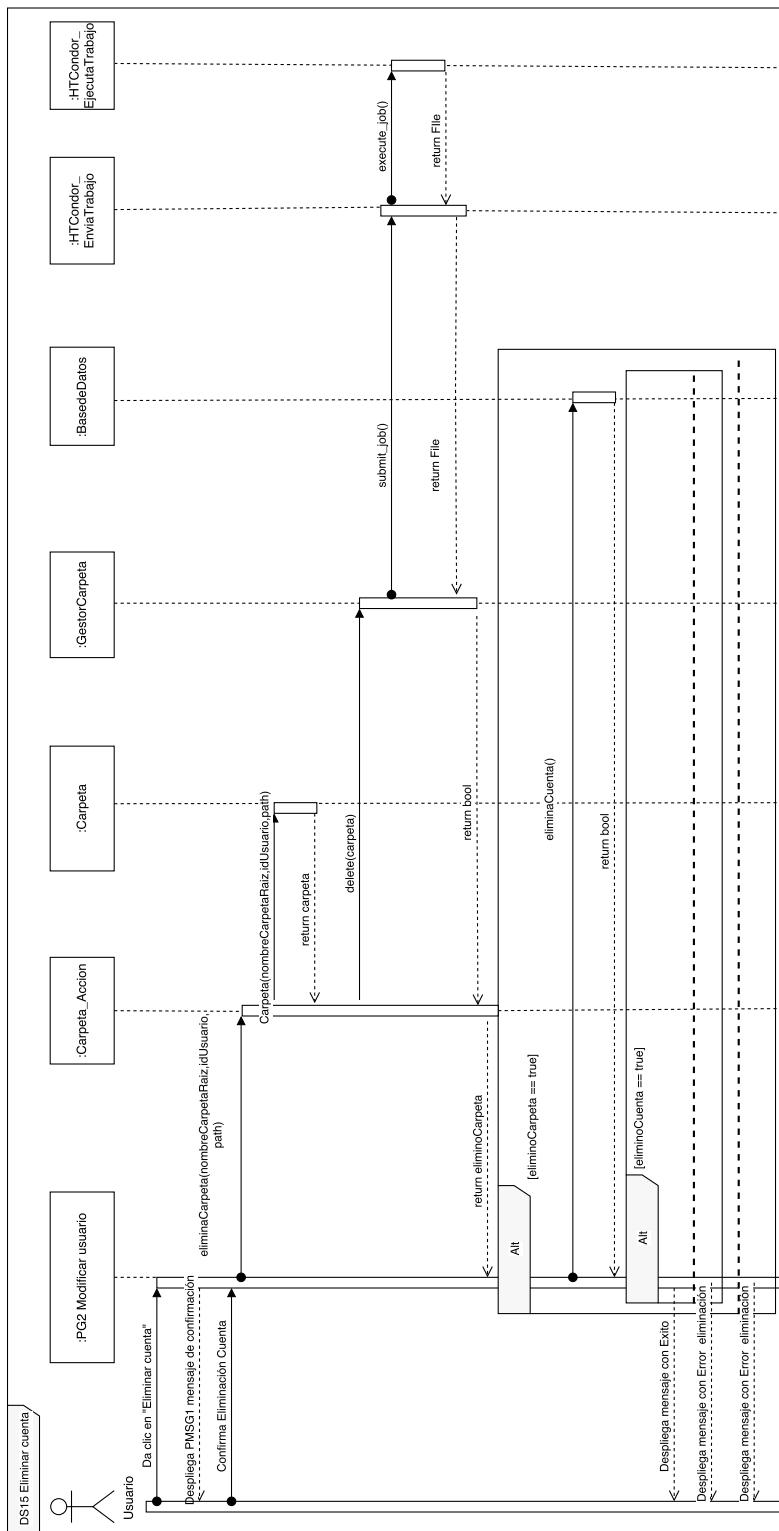


Figura 5.18: Diagrama de secuencia para Eliminar cuenta.

5.6.14. DS16 Modificar contraseña

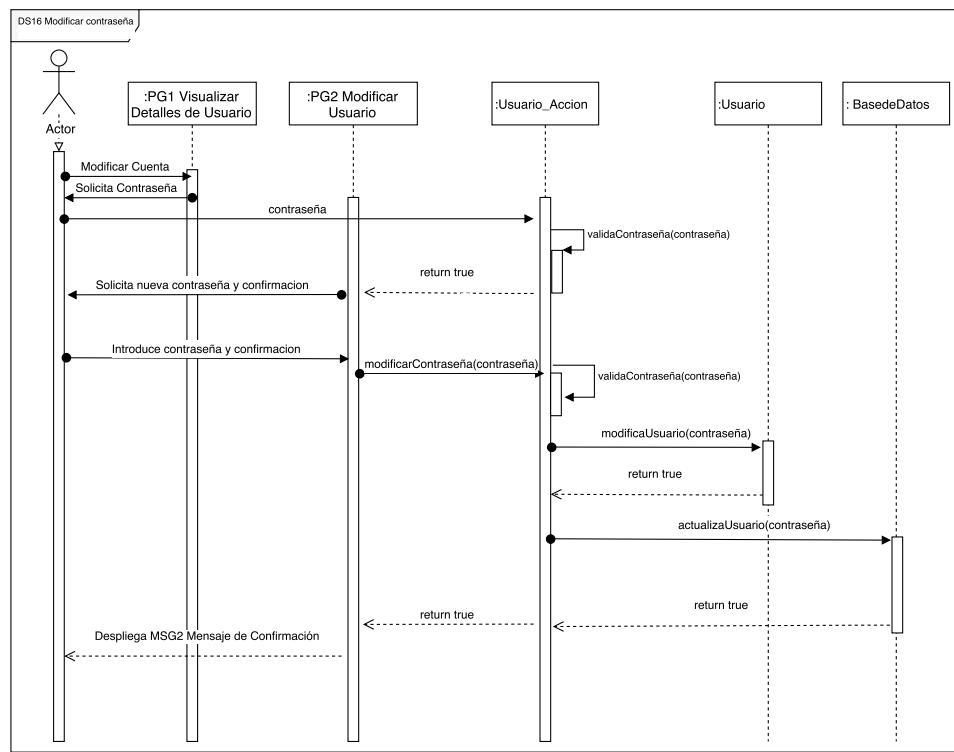


Figura 5.19: Diagrama de secuencia para Modificar contraseña.

5.6.15. DS17 Modificar Nombre de Usuario

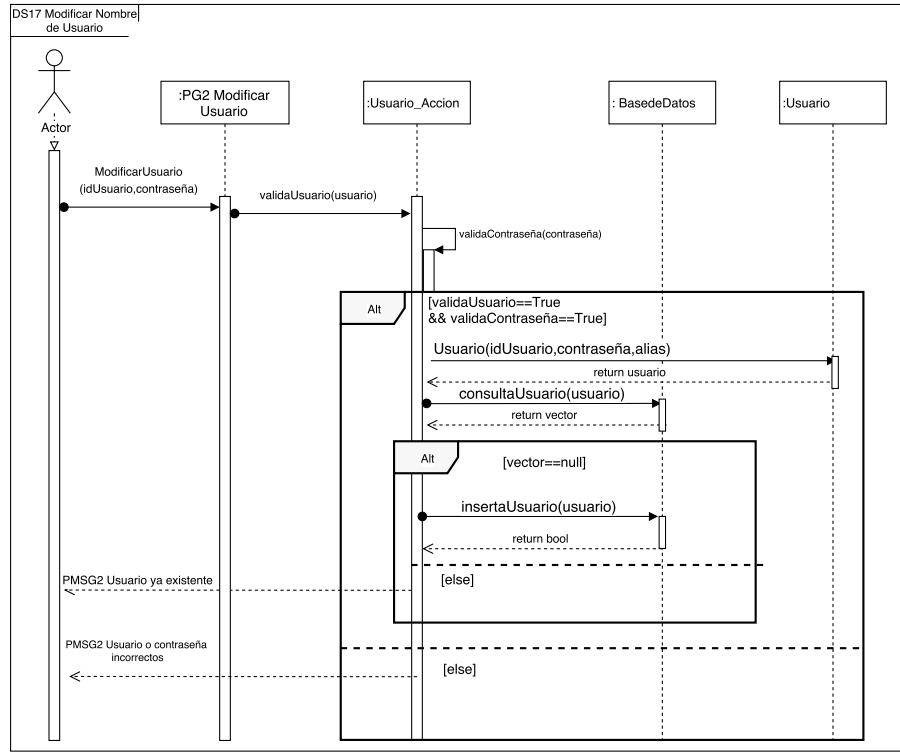


Figura 5.20: Diagrama de secuencia para Modificar Nombre de Usuario.

5.6.16. DS18 Recuperar cuenta.

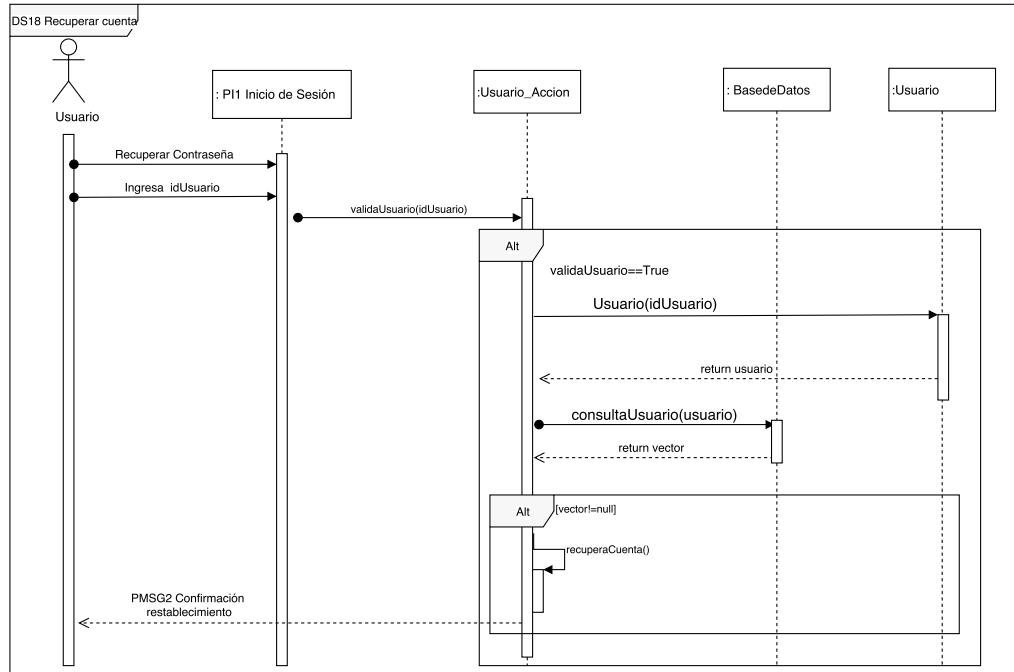


Figura 5.21: Diagrama de secuencia para Recuperar cuenta.

5.7. Interfaz de usuario

5.7.1. PI1 Inicio de Sesión

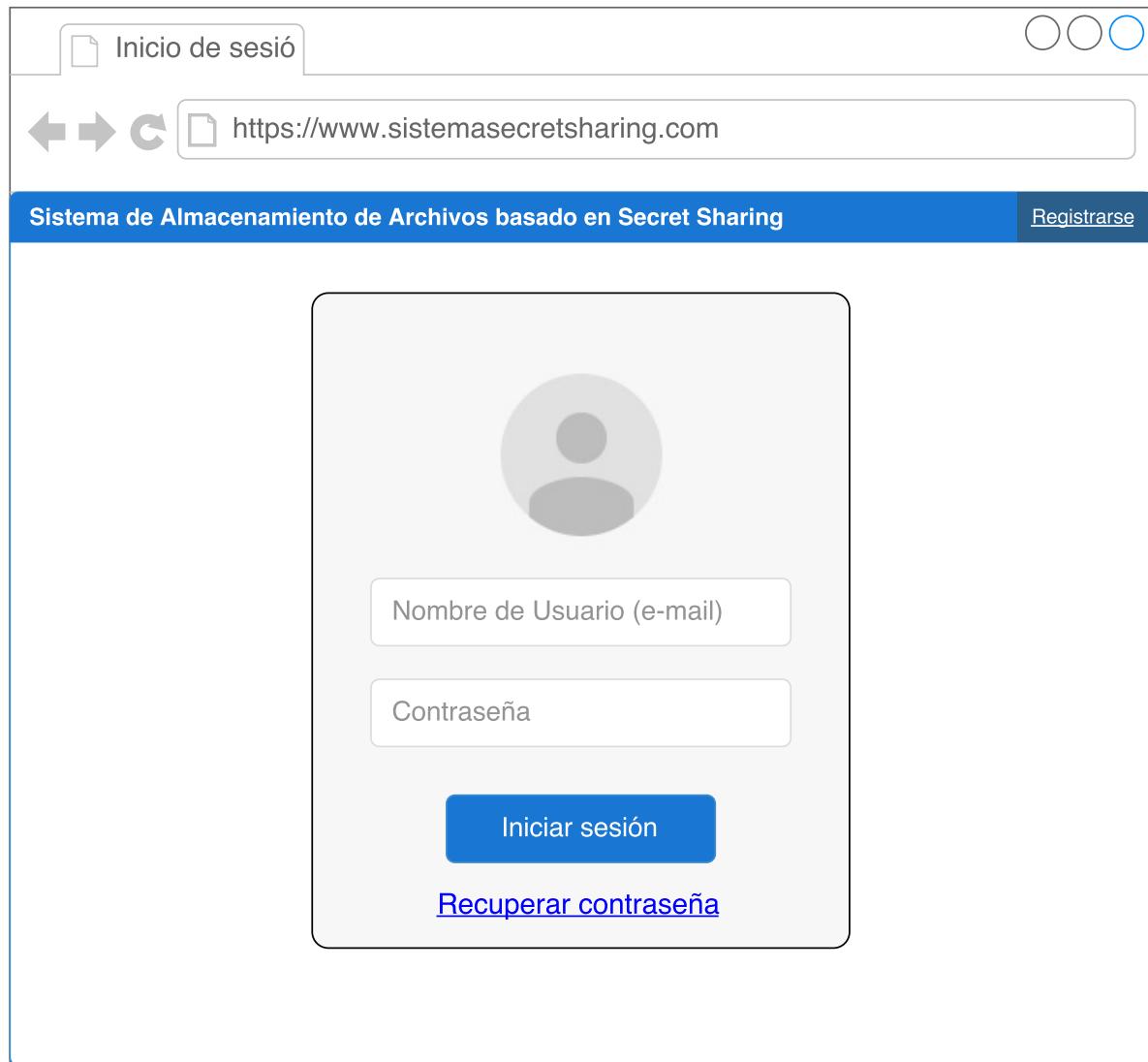


Figura 5.22: Pantalla de Inicio de sesión del sistema.

5.7.2. PI2 Registro de Cuenta de Usuario

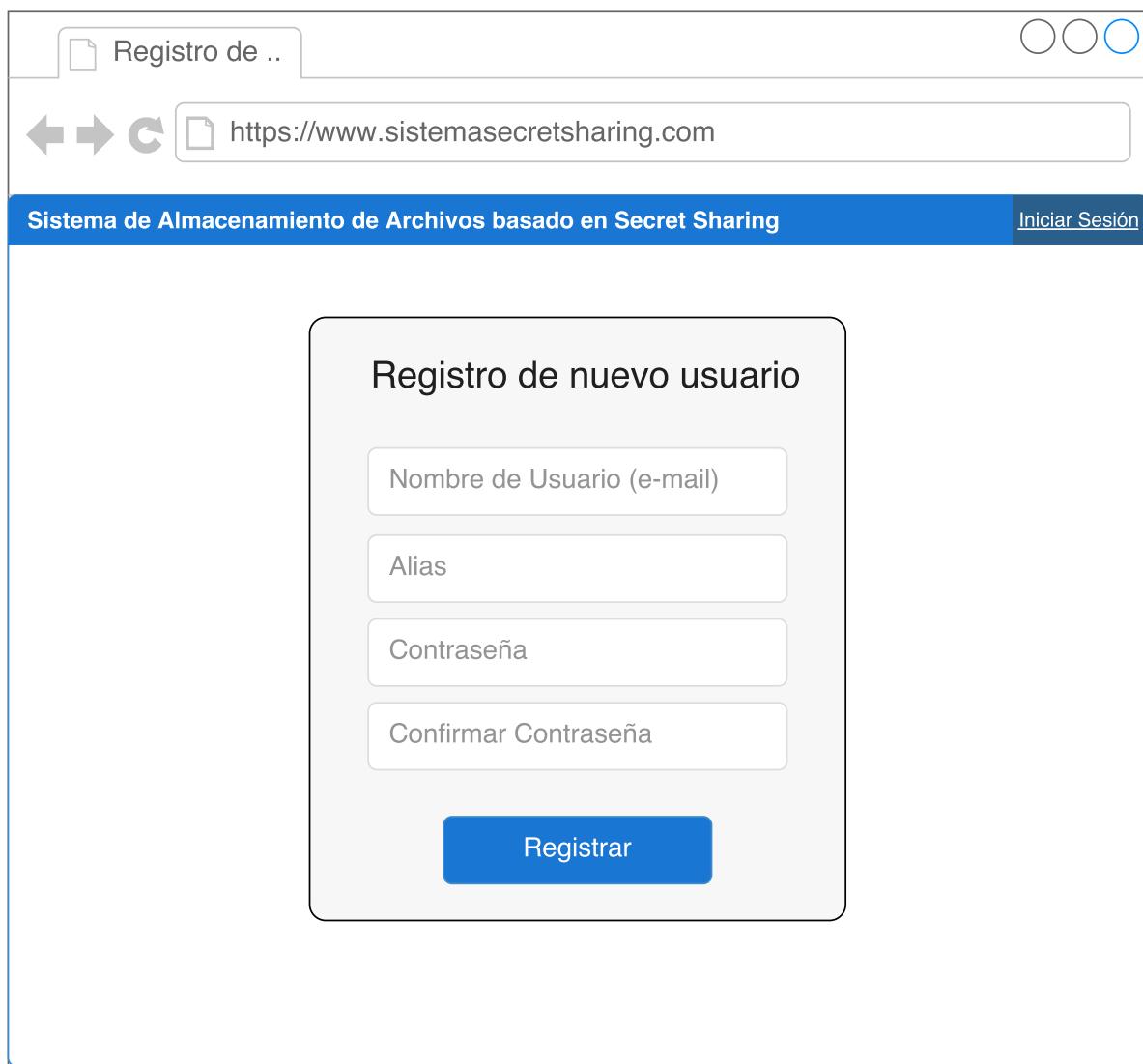


Figura 5.23: Pantalla de registro de una Cuenta de Usuario.

5.7.3. PI3 Recuperar contraseña

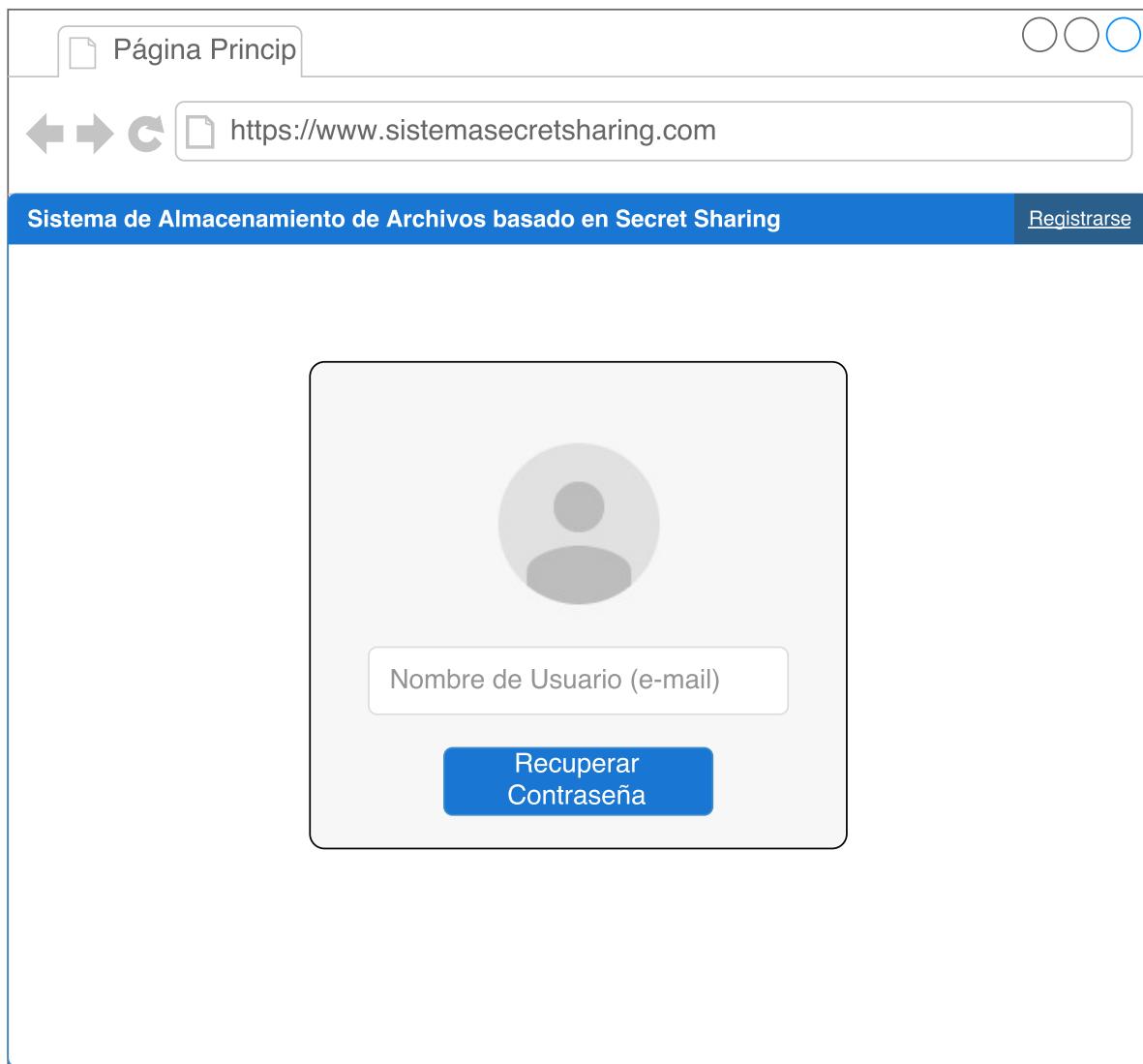


Figura 5.24: Pantalla de recuperación de la contraseña de Usuario.

5.7.4. PP1 Pantalla Principal

The screenshot shows a web-based application interface. At the top, there is a header bar with the URL <https://www.sistemasecretsharing.com>. Below the header, the main content area has a title "MIS DOCUMENTOS". On the left side, there is a sidebar with a user profile picture, the text "@UserAlias", and two buttons: "SUBIR ARCHIVO" and "CREAR CARPETA". The main content area displays a table of documents:

	Nombre	Fecha de Subida	Tamaño	¿Qué deseas hacer?
<input type="checkbox"/>	Notas.txt	16-10-2017	30 KB	<input type="checkbox"/> Copiar
<input type="checkbox"/>	Proyecto.pdf	23-10-2017	784 KB	<input type="checkbox"/> Mover
<input type="checkbox"/>	Avances.doc	24-10-2017	200 KB	<input type="checkbox"/> Eliminar
<input type="checkbox"/>	ADOO.xml	01-11-2017	298 KB	<input type="checkbox"/> Renombrar
<input type="checkbox"/>	Vacaciones.jpg	26-11-2017	1200 KB	
<input type="checkbox"/>	Calificaciones.xls	05-12-2017	10 KB	
<input type="checkbox"/>	Criptografía	01-10-2017	1000 KB	

On the right side of the table, there is a vertical menu with options: "Copiar", "Mover", "Eliminar", and "Renombrar". At the bottom right of the table, there is a blue button labeled "Descargar Archivo".

Figura 5.25: Pantalla Principal del sistema.

5.7.5. PPC1 Renombrar Carpeta



Figura 5.26: Pantalla del Panel para Renombrar Carpeta.

5.7.6. PPC2 Renombrar Archivo



Figura 5.27: Pantalla del Panel para Renombrar Archivo.

5.7.7. PPC3 Nueva Carpeta

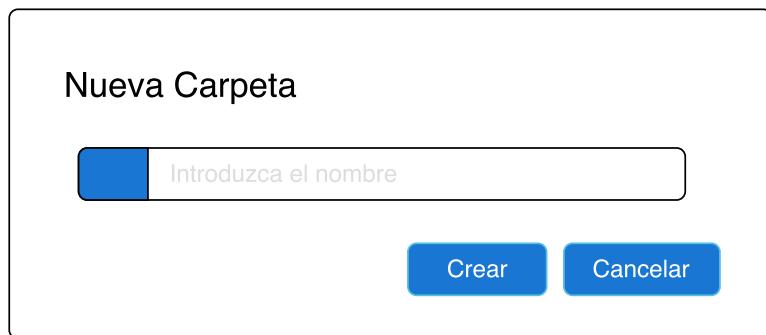


Figura 5.28: Pantalla del Panel para crear una nueva Carpeta.

5.7.8. PPC4 Copiar Archivo

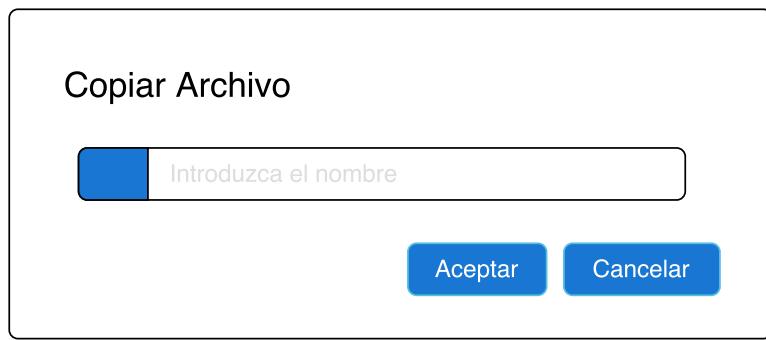


Figura 5.29: Pantalla del Panel para ingresar un nombre al copiar un Archivo.

5.7.9. PG1 Visualizar detalles de Usuario

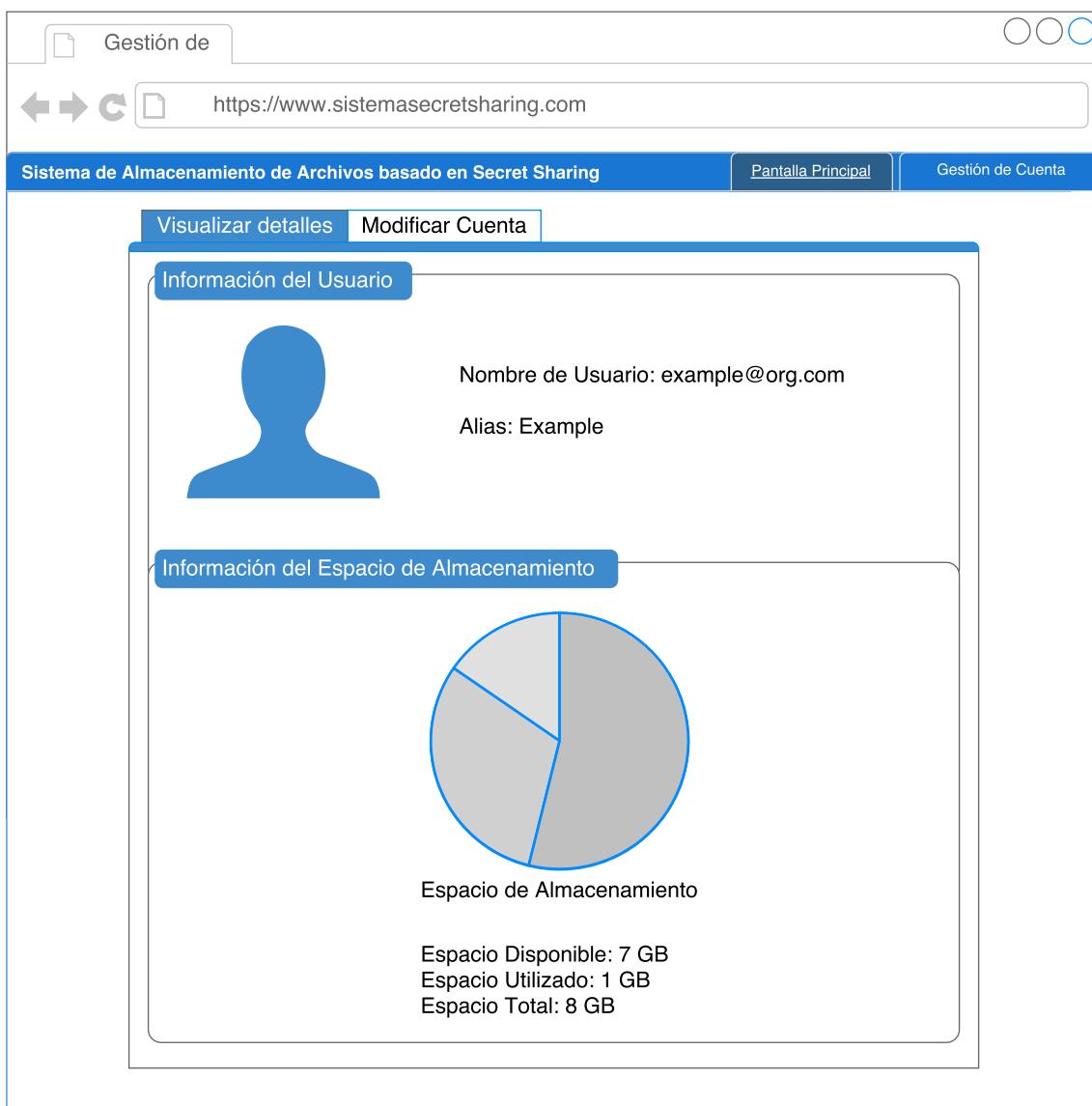


Figura 5.30: Pantalla de Gestión de Cuenta, en la sección de Visualizar detalles.

5.7.10. PG2 Modificar Usuario

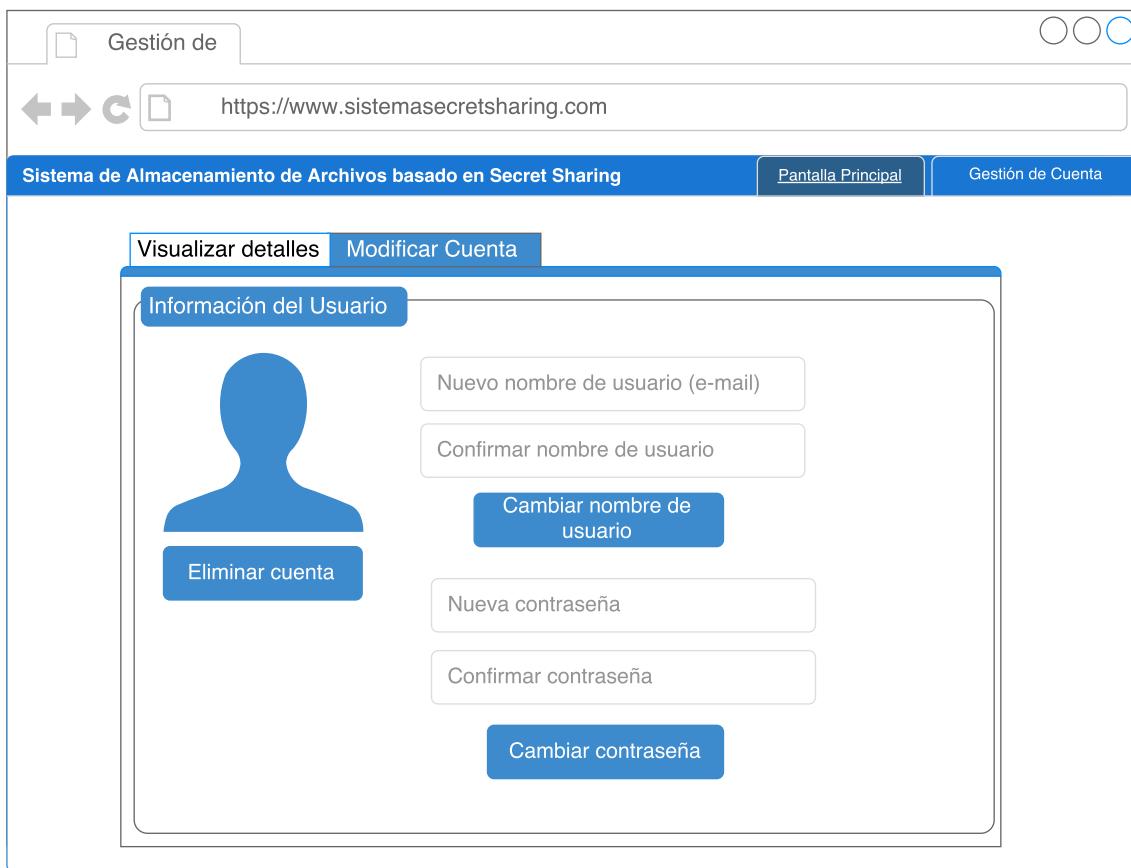


Figura 5.31: Pantalla de Gestión de Cuenta, en la sección de Modificar Cuenta.

5.7.11. PMSG1 Mensaje de confirmación



Figura 5.32: Pantalla del Mensaje de confirmación.

5.7.12. PMSG2 Mensaje de información



Figura 5.33: Pantalla del Mensaje de información.

5.8. Requisitos de diseño

En esta etapa se realiza la creación de especificaciones de diseño en materia de seguridad y privacidad, como es la especificación de los requisitos mínimos de diseño criptográfico.

Para la realización de esta especificación, se consultaron las recomendaciones del NIST (National Institute of Standards and Technology)[42] y las recomendaciones de Microsoft para la metodología SDL[43].

5.8.1. Protocolos de seguridad

- El sistema hará uso del protocolo TLS en la versión 1.2
- El sistema no podrá hacer uso de versiones anteriores de TLS.
- El sistema no podrá hacer uso del protocolo SSL versión 3 ni SSL versión 2.

5.8.2. Cifrado y descifrado utilizando algoritmos de cifrado por bloques

El sistema hará uso del siguiente algoritmo de cifrado por bloques.

- Algoritmo de cifrado por bloques “AES” (Advanced Encryption Standard), con un tamaño mínimo de llave de 128 bits.

El sistema no podrá hacer uso de los siguientes algoritmos de cifrado por bloques:

- DES
- “SKIPJACK”
- Triple DES

5.8.3. Modos de operación

El sistema podrá hacer uso de los siguientes modos de operación:

- CTR
- CBC

El sistema no podrá hacer uso de cualquier otro modo de operación.

5.8.4. Generadores de número pseudoaleatorios

El sistema podrá hacer uso de cualquiera de los siguientes generadores de número aleatorios:

- HASH_DRBG
- HMAC_DRBG
- CTR_DRBG

El sistema no podrá hacer uso de los siguientes generadores de número pseudo-aleatorios:

- Generadores de número aleatorios especificados en FIPS 186-2, ANS X9.31 y ANS X9.62-1998.
- Generador de números aleatorios de doble curva elíptica (DUAL_EC_DRBG).
- Funciones criptográficamente inseguras relacionadas con la generación de números aleatorios, como son rand() (C y C++) o Math.random() (Java).

5.8.5. Funciones Hash

El sistema podrá hacer uso de las siguientes funciones hash:

- Función Hash SHA-2 en cualquiera de sus variantes (SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224 y SHA-512/256).
- Función Hash SHA-3 en cualquiera de sus variantes (SHA3-224, SHA3-256, SHA3-384, y SHA3-512).

El sistema no podrá hacer uso de las siguientes funciones hash:

- Función Hash SHA-1.
- Función Hash MD5.

5.8.6. Otros requisitos

El sistema no reportará errores o fallos específicos en alguna operación criptográfica a los usuarios. Cuando alguno de estos ocurra, el sistema mostrará a los usuarios finales un mensaje de error genérico, sin especificar detalles innecesarios.

5.9. Superficie de ataques

De acuerdo con la metodología SDL[39], la reducción de la superficie de ataques está estrechamente relacionada con los modelos de riesgos, sin embargo aborda los problemas de seguridad desde una perspectiva ligeramente diferente. La reducción de la superficie de ataques es una forma de reducir el riesgo, dando a los atacantes menos oportunidades para aprovechar un posible punto débil o una posible vulnerabilidad.

Para reducir la superficie de ataques, se restringe el acceso a los servicios del sistema, se aplica el principio de privilegios mínimos o se usan en la medida de lo posible defensas por capas.

Para la reducción de la superficie de ataques dentro del sistema de almacenamiento seguro de archivos basado en secreto compartido se hicieron las siguientes consideraciones:

1. Aplicación de cortafuegos a nivel de aplicación que solo permita el flujo de tráfico de los siguientes procesos:
 - Sistema Gestor de Base de Datos MySQL.

- Contenedor web Tomcat.
 - Sistema gestor de carga de trabajo HTCondor.
2. Restricción de permisos para archivos de configuración y archivos de registro (logs) en el Sistema Operativo anfitrión, es decir, sólo se podrá acceder y modificar dichos archivos con permisos de superusuario (root).
 3. Ejecución de los procesos mencionados con permisos de “Usuario estándar”, dentro del Sistema Operativo.

5.10. Modelo de riesgos

Para el análisis de riesgos de este sistema, se aplicará el modelo de amenazas STRIDE, el cual es recomendado por Microsoft para la implementación dentro de la metodología seleccionada (SDL).

5.10.1. Modelo de Amenazas STRIDE

STRIDE (Spoofing, Tampering, Repudiation, Information disclosure, Denial of service & Elevation of privilege) es un acrónimo que representa el espectro de amenazas de seguridad que pueden afectar a la aplicación. Dicho modelo describe 6 categorías de amenazas las cuales son explicadas a grandes rasgos a continuación:

- Suplantación de identidad: Ocurre cuando una persona se hace pasar por alguien que no es.
- Manipular datos: Alteración de la información.
- Repudio: Son aquellas en las que los usuarios niegan la autoría de una acción sin que otras partes puedan probar lo contrario.
- Divulgación de información: Revelación de información a individuos que no deben tener acceso a la misma.
- Denegación de servicio: Los ataques por denegación de servicio (DDoS) ocasionan la pérdida de servicio a los usuarios válidos.
- Elevación de privilegios: Acceso a niveles más altos de permisos dentro de la aplicación por usuarios que no deberían poseer dichos permisos.

La forma más sencilla y adecuada de aplicar este modelo en una aplicación es mediante el análisis de cada una de las amenazas dentro de cada componente y a cada una de sus conexiones o relaciones con los demás componentes, esto con la finalidad de determinar las categorías de amenazas que afectan a cada uno de los componentes. Dicho proceso debe ser repetitivo, con cada una de las etapas que se mencionan a continuación:

1. Enumerar las amenazas conocidas y determinar cómo cada una de ellas afecta al sistema

2. Clasificación de las amenazas por gravedad o impacto y probabilidad, asignando valores entre 1 y 10 siendo 10 el nivel más grave, mientras que para la asignación de probabilidad los valores se toman de forma inversa, siendo el 1 el más probable. Una vez asignados estos valores se calcula el riesgo de la siguiente manera: $Riesgo = Gravedad/Probabilidad$
3. Elegir una técnica o tecnología adecuada para contrarrestar cada amenaza.
4. Repetir durante la evolución del proyecto.

Análisis de Riesgos para Sistema de Almacenamiento Seguro de Archivos Basado en Secreto Compartido

Amenaza	Gravedad	Probabilidad	Riesgo	Técnica o tecnología para contrarrestar la amenaza
Acceso a los servidores mediante el uso de puertos usados por la aplicación.	10	8	1.25	Uso de Firewall
Uso de fuerza bruta al realizar login.	8	9	0.88	Uso de contraseñas robustas. Cifrado de información antes de enviar las peticiones.
Robo de credenciales de usuarios.	10	6	1.67	Cifrado de la base de datos que contiene las credenciales
Alteración de los documentos al momento de ser almacenados.	7	7	1	Cifrado de la información contenida en los documentos. Uso de funciones hash que validen la integridad de la información.
Eliminación intencional de documentos por parte de terceros.	7	5	1.4	Creación de respaldos de información. Uso de logs.
Fallas físicas de los servidores	10	10	1	Mantenimiento correctivo y preventivo de los servidores
Uso de Sniffers por parte de terceros para la obtención de información confidencial	6	4	1.5	Uso de HTTPS/TLS
Alteración de la información almacenada en la base de datos	10	7	1.42	Realizar copias de seguridad (Back-Up/Rollback) Uso de logs
Inyecciones SQL	10	5	2	Acceso a Base de datos con permisos limitados. Uso de funciones seguras provistas por el lenguaje.
Saturaciones de búfer Errores aritméticos de enteros	10	5	2	Uso de funciones/clases seguras provistas por el lenguaje.
Modificación de permisos de los usuarios	10	10	1	Acceso a Base de datos con permisos limitados. Restricciones bien definidas de los privilegios de los usuarios.
Saturación del servicio	10	10	1	Uso de balanceadores de carga.

Tabla 5.1: Detección y clasificación de las Amenazas por Gravedad y Probabilidad.

La evaluación de la gravedad y probabilidad asignada a cada una de las amenazas se basó en el análisis de los documentos provistos por el CERT y por Kaspersky Lab, donde el primero menciona las amenazas detectadas durante los trimestres del año 2016 dentro de la RedUNAM [44]; mientras que el documento provisto por Kaspersky Labs[45] hace un recuento

del número de ataques realizados durante el 2013 a diversas compañías a nivel internacional.

Glosario

SDL Microsoft Security Development Lifecycle. 4, 34, 35, 40, 92–94

SSL Secure Sockets Layer. 2, 26–28, 92

TLS Transport Layer Security. 2, 3, 27, 28, 92, 96

Referencias

- [1] M. Geel, *Cloud Storage: File Hosting and Synchronisation 2.0*, https://www.vis.ethz.ch/de/visionen/pdfs/2012/visionen_2012_3.pdf.
- [2] F. Alsolami y T. Boult, *CloudStash: Using Secret-Sharing Scheme to Secure Data, Not Keys, in Multi-clouds*, https://www.researchgate.net/publication/269304909_CloudStash_Using_Secret_Sharing_Scheme_to_Secure_Data_Not_Keys_in_Multi-clouds, 2014.
- [3] Dropbox, *¿Cuál es el nivel de seguridad de Dropbox?*, <https://www.dropbox.com/help/27>.
- [4] Google, *Verificación en dos pasos*, <https://www.google.com/landing/2step/features.html>.
- [5] M. Bellare y P. Rogaway, «Robust Computational Secret Sharing and a Unified Account of Classical Secret-sharing Goals», en *Proceedings of the 14th ACM Conference on Computer and Communications Security*, ép. CCS '07, Alexandria, Virginia, USA: ACM, 2007, págs. 172-184, ISBN: 978-1-59593-703-2. DOI: 10.1145/1315245.1315268. dirección: <https://eprint.iacr.org/2006/449.pdf>.
- [6] IBM, *IBM Cloud Object Storage*, <https://www.ibm.com/bs-en/marketplace/object-storage>.
- [7] L. Rathnam, *A Look Into IBM's Cloud Object Storage*, <http://cloudnewsdaily.com/ibm-cloud-object-storage/>.
- [8] S. First, *SPx™ TECHNOLOGY*, <https://securityfirstcorp.com/spx-technology/>.
- [9] A. V. S. Tanenbaum, *Sistemas Distribuidos. Principios y paradigmas*. Pearson Educación, 2008.
- [10] J. K. G. Dolimore, *Sistemas Distribuidos Conceptos y Diseño*. Pearson Educación, 2001.
- [11] D. Salomon, *Coding for Data and Computer Communications*. Springer, 2005.
- [12] A. Menezes, P. Oorschot y S. Vanstone, *Handbook of Applied Cryptography*. CRC Press, 1996, ISBN: 0-8493-8523-7.
- [13] IBM, *Cifrado de clave privada*, https://www.ibm.com/support/knowledgecenter/es/SSMKHH_9.0.0/com.ibm.etools.mft.doc/ac55940_.htm.
- [14] D. Chakraborty y P. Sarkar, «HCH: A New Tweakable Enciphering Scheme Using the Hash-Counter-Hash Approach», *IEEE Trans. Information Theory*, vol. 54, n.º 4, págs. 1683-1699, 2008. DOI: 10.1109/TIT.2008.917623. dirección: <http://dx.doi.org/10.1109/TIT.2008.917623>.

- [15] IBM, *Modes of Operation*, https://www.ibm.com/support/knowledgecenter/es/SSLTBW_2.2.0/com.ibm.zos.v2r2.csfb400/mooke.htm.
- [16] W. Stallings, *Fundamentos de Seguridad en Redes: Aplicaciones y Estándares*. Pearson Educación, 2004.
- [17] H. D. Scolnik y J. Hecht, *Impacto de recientes ataques de colisiones contra funciones de hashing de uso corriente*, https://www.certisur.com/sites/default/files/docs/ataques_funciones_hashing.pdf.
- [18] A. Shamir, «How to Share a Secret», *Commun. ACM*, vol. 22, n.º 11, págs. 612-613, nov. de 1979, ISSN: 0001-0782. DOI: 10.1145/359168.359176. dirección: <https://cs.jhu.edu/~sdoshi/crypto/papers/shamirturing.pdf>.
- [19] B. G, «Safeguarding cryptographic keys», *Managing Requirements Knowledge, International Workshop on*, vol. 00, pág. 313, 1899. DOI: 10.1109. dirección: <https://www.computer.org/csdl/proceedings/afips/1979/5087/00/50870313.pdf>.
- [20] E. W. Weisstein, «Secret sharing made short», *Advances in Cryptology – CRYPTO '93*, vol. 773, págs. 136-146, 1993.
- [21] ———, *Error-Correcting Code. From MathWorld—A Wolfram Web Resource*, <http://mathworld.wolfram.com/Error-CorrectingCode.html>.
- [22] Gartner, *Information Dispersal Algorithms*, <https://www.gartner.com/it-glossary/information-dispersal-algorithms>.
- [23] B. Kernighan y D. Ritchie, *The C programming Language*. Prentice-Hall, 1988.
- [24] B. Stroustrup, «The Essence of C++», en *The C++ Programming Language*, 2014.
- [25] M. Stanley, *C++ Applications*, <http://www.stroustrup.com/applications.html>.
- [26] ISO/IEC, *ISO/IEC 14882:2011*, <https://www.iso.org/standard/50372.html>.
- [27] J. Gosling, B. Joy, G. Steele, G. Bracha y A. Buckley, *The Java Language Specification*, <https://docs.oracle.com/javase/specs/jls/se8/jls8.pdf>.
- [28] TIOBE, *TIOBE Index for March 2017*, <https://www.tiobe.com/tiobe-index/>.
- [29] ApacheSF, *Apache Tomcat*, <https://tomcat.apache.org/>.
- [30] ———, *SSL/TLS Configuration HOW-TO*, <https://tomcat.apache.org/tomcat-7.0-doc/ssl-howto.html>.
- [31] ———, *The HTTP Connector*, <https://tomcat.apache.org/tomcat-7.0-doc/config/http.html>.
- [32] Oracle, *The Main Features of MySQL*, <https://dev.mysql.com/doc/refman/5.7/en/features.html>.
- [33] ———, *Too many connections*, <https://dev.mysql.com/doc/refman/5.5/en/too-many-connections.html>.
- [34] ———, *Encryption and Compression Functions*, <https://dev.mysql.com/doc/refman/5.5/en/encryption-functions.html>.
- [35] HTCondor, *What is HTCondor*, <https://research.cs.wisc.edu/htcondor/description.html>.

- [36] G. Alliance, *About the Globus Toolkit*, <http://toolkit.globus.org/toolkit/about.html>.
- [37] GestioPolis.com, *¿Qué es el estudio de factibilidad en un proyecto?*, <https://www.gestiopolis.com/que-es-el-estudio-de-factibilidad-en-un-proyecto/>.
- [38] *Sueldos IT en México 2017*: <https://everac99.wordpress.com/tag/sueldos-it/>.
- [39] Microsoft, *Implementación Simplificada del proceso SDL de Microsoft*, <https://www.microsoft.com/es-mx/download/details.aspx?id=12379>.
- [40] ———, *Create a strong password*, <https://support.microsoft.com/en-us/instantanswers/9bd5223b-efbe-aa95-b15a-2fb37bef637d/create-a-strong-password>.
- [41] Google, *Creating a strong password*, <https://support.google.com/accounts/answer/32040?hl=en>.
- [42] E. Barker y A. Roginsky, *Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths*, <http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-131Ar1.pdf>.
- [43] Microsoft, *Microsoft SDL Cryptographic Recommendations*, <https://goo.gl/su7wVV>.
- [44] CERT, *Estadísticas de incidentes en RedUNAM durante 2016*, <http://www.cert.org.mx/estadisticas.dsc>.
- [45] K. Lab, *Las amenazas más importantes de 2013*, <https://blog.kaspersky.es/las-amenazas-mas-importantes-de-2013/2008/>.