



UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO

CENTRO UNIVERSITARIO VALLE DE CHALCO



SISTEMA FORENSE AUDITOR DE INTEGRIDAD Y RECUPERACIÓN DE SOPORTES DE DATOS

T E S I S

QUE PARA OBTENER EL TÍTULO DE

INGENIERA EN COMPUTACIÓN

P R E S E N T A

YESSICA SARAHI TORRES TORRES

ASESORA:

Dra. en C. MARÍA DE LOURDES LÓPEZ GARCÍA

Revisor: L.C.I HORACIO JESUS TACUEÑO CRUZ

Revisor: Ing. en C. RICARDO JAVIER BUCIO LÓPEZ

VALLE DE CHALCO SOLIDARIDAD, MÉXICO JULIO 2017





OFICIO: FT5




Valle de Chalco, Méx. miércoles, 05 de julio de 2017

M. EN P. J. JUAN CARLOS HERNÁNDEZ HERNÁNDEZ
SUBDIRECTOR ACADÉMICO
DEL CENTRO UNIVERSITARIO UAEM VALLE DE CHALCO

PRESENTE.

Por este conducto, comunico a usted que el trabajo de Tesis con el titulo:

Sistema Forense Auditor de Integridad y Recuperación de Soportes de Datos
Llevado a cabo por Yessica Sarahi Torres Torres con número de cuenta
1226770 de la licenciatura en **Ingeniería en Computación** registrado el 18 de noviembre
de 2016 con Número de Registro ICO/18.11.16/449 ha concluido y estamos de acuerdo
para la impresión definitiva de Tesis

	Nombre	Firma
Asesor	Dra. en C. Maria de Lourdes López Garcia	
Revisor	L.C.I. Horacio Jesús Tacubéño Cruz	
Revisor	I. en C. Ricardo Javier Bucio López	

Sin más por el momento quedo de usted.

ATENTAMENTE



Yessica Sarahi Torres Torres





CARTA DE CESIÓN DE DERECHOS DE AUTOR

El que suscribe Yessica Sarahi Torres Torres Autor del trabajo escrito de evaluación profesional en la opción de Tesis con el título Sistema Forense Auditor de Integridad y Recuperación de Soportes de Datos, por medio de la presente con fundamento en lo dispuesto en los artículos 5, 18, 24, 25, 27, 30, 32 y 148 de la Ley Federal de Derechos de Autor, así como los artículos 35 y 36 fracción II de la Ley de la Universidad Autónoma del Estado de México; manifiesto mi autoría y originalidad de la obra mencionada que se presentó en Valle de Chalco Solidaridad, para ser evaluada con el fin de obtener el Título Profesional de Ingeniera en Computación.

Así mismo expreso mi conformidad de ceder los derechos de reproducción, difusión y circulación de esta obra, en forma NO EXCLUSIVA, a la Universidad Autónoma del Estado de México; se podrá realizar a nivel nacional e internacional, de manera parcial o total a través de cualquier medio de información que sea susceptible para ello, en una o varias ocasiones, así como en cualquier soporte documental, todo ello siempre y cuando sus fines sean académicos, humanísticos, tecnológicos, históricos, artísticos, sociales, científicos u otra manifestación de la cultura.

Entendiendo que dicha cesión no genera obligación alguna para la Universidad Autónoma del Estado de México y que podrá o no ejercer los derechos cedidos.

Por lo que el autor da su consentimiento para la publicación de su trabajo escrito de evaluación profesional.

Se firma la presente en el Estado de México, a los 06 días del mes de Julio de 2017.

Yessica Sarahi Torres Torres



Dedicatoria

A mis padres, hermanos, hijo/sobrino, familia, Amigos (David, Sam, Gabriel, Ismael, Andrea, Celeste, Mishh, Efrain, Sergio, Ignacio) y a todas las personas que tuvieron fe en mi...

A mi asesora: Dra. María de Lourdes López García.

AGRADECIMIENTOS

En el transcurso de mi vida he tenido críticas sobre lo que sería de mi futuro, con el paso del tiempo también dude de lo que puedo llegar a realizar, con el apoyo de personas importantes estoy viviendo este logro el cual me satisface saber que están para apoyarme.

Mi Familia son las personas principales que han estado para brindarme su apoyo, confianza y fe en mí.

A Mis Padres:

Es inefable el agradecimiento que tengo, ya que ambos me apoyan en mis errores, fracasos y logros.

Casi 25 años de que me conocen y han estado para guiarme en cada paso que doy, lo que me ha convertido en la hija, hermana, amiga y mujer que soy. Siempre existieron palabras de apoyo y motivación que me permitieron culminar esta etapa. Es sencillo me siento feliz de que ustedes sean mis padres. Los Amo.

A Mis Hermanos:

Erika, Miguel

Me brindaron su apoyo incondicional en este proceso brindándome cada uno a su manera esas palabras que me impulsaron a concluir este proyecto, les agradezco que sean mis hermanos saben todo el amor que les tengo, sé que los tres estamos para apoyarnos en cada situación que la vida nos coloque.

A Mi Sobrino:

Leonardo

El niño que adoro con todo mi ser, gracias por hacerme reír cuando más lo necesitaba, quien en mis momentos de desesperación estuvo para darme un abrazo, una broma o simplemente un comentario que hacía olvidarme de mi

estrés. Te Amo Sobrino.

He conocido personas que han mejorado la perspectiva que veo de mi vida y les agradezco mucho por hacerme parte de su vida.

Sam: Sin duda alguna eres un increíble ser humano, siempre estuviste para escucharme, aconsejarme y sobre todo hacerme ver de lo que puedo ser capaz, llegaste a mi vida en el mejor momento me diste las palabras de consuelo y aliento para seguir e incluso tu apoyo incondicional, gracias por motivarme a ser mejor persona, eso lo valoro aun más. ¡Te Quiero!

David: Mi mejor amigo, cómplice, hermano... Gracias por creer y tener fe en mí. Eres un gran joven y hemos compartido muchos años de amistad, permitiendo conocernos, apoyarnos y amarnos como personas, gracias por permanecer en mi vida.

Gabriel, Ismael, Andrea: Siempre creyeron en mí, incluso cuando yo no creía y me dieron fuerzas para salir adelante, siempre existieron palabras de aliento, muchas gracias ¡OHANA!

Durante mi trayecto universitario conocí a personas que me apoyaron de manera académica y personal. Sin pensarlo el destino decidió que se diera más que una amistad.

Celeste, Efraín, Juan: Gracias infinitas por todo el apoyo que me brindaron en esta etapa que compartimos juntos, tuvimos problemas, desacuerdos, risas, alegrías y demás situaciones que permitieron conocernos como personas y que a pesar de tener distintas personalidades aprendimos a querernos. Su apoyo incondicional en la parte académica, estuvieron para apoyarme, ayudarme, explicarme durante nuestra instancia en el C.U.

Celes: Mi mejor amiga, mi cómplice durante esta etapa importante en mi vida, el destino esta decidiendo que sigamos manteniendo esa amistad, ser esa persona que impulso a mejorar en la parte academica, gracioso es recordar como fue nuestro acercamiento, pasar tantas aventuras juntas y las que vienen aun serán

mas grandiosas, esperemos el destino sea a nuestro favor y logremos las metas que tenemos. ¡Te Amo Amora!

Juan: Simplemente gracias por todo el apoyo que brindaste durante esta etapa, conocerte desde la secundaria y volver a coincidir en la universidad es agradable, con errores, problemas, excelentes momentos vividos a tu lado. ¡Gracias infinitas!

Efra: Un gran chico, excelente amigo sin pensarlo consolidamos una linda amistad, tantas risas, y grandes momentos que compartimos sin duda alguna gracias por dejarme conocer mas de ti Amigo.

Sergio: Gracias por permitir conocerte fuera de la institución se ha dado una amistad muy agradable y en poco tiempo logre tener mucho cariño por ti, el apoyo que me has brindado y sobre todo por estar para escucharme.

Ignacio: Agradezco por el apoyo, y los consejos que me brindaste, tantos recuerdos que tenemos, un excelente amigo y ser humano, fuiste de los primeros amigos que tuve en la universidad. ¡Te Quiero, Paws Up!

Este trabajo de tesis no hubiese sido posible sin la participación de la Dra. María de Lourdes López García, quien me brindo dedicación, apoyo, enseñanza y confianza. Quien pude convivir más con ella y me permitió conocer sobre su vida fuera del C.U. Gracias por aceptarme como tesista, y por creer en mí, juntas pudimos hacer posible este logro, sin duda alguna un placer ser su asesora de tesis. Le guardo cariño.

A mis revisores los profesores Horacio Tacubeño, Ricardo Bucio les agradezco la atención que me brindaron durante la etapa final de este proyecto, con ambos tuve la oportunidad de que me impartieran clases, me llevo aprendizaje y una bonita amistad.

A mis profesores del C.U UAEM Valle de Chalco que ayudaron en mi formación académica les agradezco por ser parte de este ciclo de mi vida.

¡Gracias infinitas a todos!

ÍNDICE

AGRADECIMIENTOS	5
ÍNDICE	8
ÍNDICE DE TABLAS	9
ÍNDICE DE FIGURAS	10
RESUMEN	13
1. INTRODUCCIÓN	14
1.1. <i>Planteamiento del problema</i>	15
1.2. <i>Hipótesis</i>	16
1.3. <i>Objetivos</i>	17
1.4. <i>Metodología</i>	18
1.5. <i>Organización del documento</i>	19
2. ANTECEDENTES	20
2.1. <i>Respaldo y compresión</i>	24
2.2. <i>Auditoría e integridad</i>	27
3. HERRAMIENTAS UTILIZADAS	29
3.1. <i>Función Hash</i>	35
4. SISTEMA COMPRESOR DE ARCHIVOS UAEMEX	37
4.1. <i>Casos de Usos</i>	37
4.2. <i>Diagrama de Clases</i>	50
4.3. <i>Diagramas de Interacción</i>	50
4.4. <i>Diagrama de Actividades</i>	55
5. IMPLEMENTACIÓN	61
5.1. <i>Pruebas</i>	71
6. CONCLUSIONES Y ALCANCES A FUTURO	75
6.1. <i>Conclusiones</i>	75
6.2. <i>Alcances a Futuro</i>	77
REFERENCIAS	78

ÍNDICE DE TABLAS

Tabla 3–1. Obtención y ordenación de la frecuencia de cada carácter. (Fuente: Propia, 2017).	30
Tabla 3–2. Obtención y ordenación de la frecuencia de cada carácter. (Fuente: Propia, 2017).	31
Tabla 3–3. Código Huffman obtenido después del proceso. (Fuente: Propia, 2017).	33
Tabla 4–1. Especificaciones del caso de uso Agregar Usuario. (Fuente: Propia, 2017).	40
Tabla 4–2. Especificaciones del caso de uso Iniciar Sesión. (Fuente: Propia, 2017).	40
Tabla 4–3. Especificaciones del caso de uso Seleccionar Archivo. (Fuente: Propia, 2017).	43
Tabla 4–4. Especificaciones del caso de uso Generar código Huffman y Hash. (Fuente: Propia, 2017).	44
Tabla 4–5. Especificaciones del caso de uso Respaldo. (Fuente: Propia, 2017).	45
Tabla 4–6. Especificaciones del caso de uso Seleccionar archivo y decodificar Huffman. (Fuente: Propia, 2017).	48
Tabla 4–7. Especificaciones del caso de uso Analizar Archivos. (Fuente: Propia, 2017).	49
Tabla 5–1. Tiempos de Ejecucion de Diferentes Tamaños de Archivos en el Sistema Propuesto. (Fuente: Propia, 2017).	73

ÍNDICE DE FIGURAS

Figura 3.1 Primer árbol generado, que tiene como raíz la suma de los dos primeros valores de la frecuencia en el orden de menor a mayor. (Fuente: Propia, 2017).	30
Figura 3.2 Primer árbol generado, que tiene como raíz la suma de los dos primeros valores de la frecuencia en el orden de menor a mayor. (Fuente: Propia, 2017).	31
Figura 3.3 Segundo árbol generado, que tiene el nodo de la suma de las frecuencias 4 y 2. (Fuente: Propia, 2017).	32
Figura 3.4 Tercer árbol generado, que tiene el nodo de la suma de las frecuencias 6 y 2. (Fuente: Propia, 2017).	32
Figura 3.5 Cuarto árbol generado, que tiene el nodo de la suma de las frecuencias 8 y 2. (Fuente: Propia, 2017).	33
Figura 3.6 Árbol generado con la codificación Huffman. (Fuente: Propia, 2017).	34
Figura 4.1 Diagrama general de casos de usos. (Fuente: Propia, 2017).	38
Figura 4.2 Caso de uso “Registro Usuarios”. (Fuente: Propia, 2017).	39
Figura 4.3 Caso de uso “Comprimir”. (Fuente: Propia, 2017).	42
Figura 4.4 Caso de uso “Descomprimir”. (Fuente: Propia, 2017).	47
Figura 4.5 Diagrama de clases del sistema propuesto. (Fuente: Propia, 2017).	50
Figura 4.6 Diagrama de interacción General del sistema. (Fuente: Propia, 2017).	51
Figura 4.7 Diagrama de interacción del registro de usuario. (Fuente: Propia, 2017).	52
Figura 4.8 Diagrama de interacción de la Compresión. (Fuente: Propia, 2017).	53
Figura 4.9 Diagrama de interacción de la Descompresión. (Fuente: Propia, 2017).	54
Figura 4.10 Diagrama de interacción del análisis de archivos. (Fuente: Propia, 2017).	55
Figura 4.11 Diagrama de Actividades General del sistema propuesto. (Fuente: Propia, 2017).	56
Figura 4.12 Diagrama de Actividades de Registro Usuario. (Fuente: Propia, 2017).	57
Figura 4.13 Diagrama de Actividades Comprimir. (Fuente: Propia, 2017).	58

Figura 4.14 Diagrama de Actividades Descomprimir. (Fuente: Propia, 2017).	59
Figura 4.15 Diagrama de Actividades Analizar Archivo. (Fuente: Propia, 2017).	60
Figura 5.1 Interfaz principal del sistema propuesto. (Fuente: Propia, 2017).	63
Figura 5.2 Interfaz para el registro del usuario. (Fuente: Propia, 2017).	63
Figura 5.3 Interfaz de Menú del sistema propuesto. (Fuente: Propia, 2017).	64
Figura 5.4 Interfaz para la selección del archivo a comprimir. (Fuente: Propia, 2017).	65
Figura 5.5 Ventana de alerta en caso de que el archivo no haya sido seleccionado. (Fuente: Propia, 2017).	66
Figura 5.6 Interfaz que visualiza el contenido del archivo a comprimir. (Fuente: Propia, 2017).	67
Figura 5.7 Ventana de aviso correspondiente a la codificación completa (Fuente: Propia, 2017).	67
Figura 5.8 Visualización de los archivos que se encuentran comprimidos en la carpeta de respaldo. (Fuente: Propia, 2017).	68
Figura 5.9 Ventana de aviso correspondiente a la decodificación completa. (Fuente: Propia, 2017).	69
Figura 5.10 Selección de archivo para el análisis de integridad. (Fuente: Propia, 2017).	69
Figura 5.11 Ventana de aviso correspondiente a la integridad intacta del documento. (Fuente: Propia, 2017).	70
Figura 5.12 Ventana de aviso correspondiente la modificación del documento.	70
Figura 5.13 Ventana de registro del Usuario1. (Fuente: Propia, 2017).	71
Figura 5.14 Ilustración de la carpeta creada por el Usuario1, denominada Usuario1 (Fuente: Propia, 2017).	71
Figura 5.15 Ventana de aviso correspondiente a la bienvenida en el sistema. (Fuente: Propia, 2017).	72

Sistema Forense Auditor de Integridad y Recuperación de Soportes de Datos

En el siguiente trabajo de tesis se explica de manera exacta el desarrollo del sistema propuesto el cual tiene como objetivo la integridad y seguridad de los datos que son almacenados en soportes digitales.

El desarrollo de este sistema sobresalio la importancia que tiene la informática forense en la tecnología, ya que se incurren actos maliciosos poniendo en riesgo la integridad de una persona asi también de un dispositivo tecnológico.

En esta tesis se describe el proceso detallado de la “Códificación Huffman” y “Función Hash” ya que son las herramientas principales que conforman el sistema propuesto, teniendo como resultado realizar respaldos, mantener seguro e integro un documento de texto.

1. INTRODUCCIÓN

“La computadora nació para resolver problemas que antes no existían”.

Bill Gates

Al paso de los años, la tecnología ha evolucionado de una manera intrépida, día tras día conocemos alguna novedad tecnológica innovadora, obteniendo como resultado que las personas de distintas edades sean más dependientes de algún dispositivo tecnológico, tanto para uso laboral, como estudiantil o de entretenimiento. Sin embargo, pocas personas se percatan de los datos que comparten, siendo así que no leen los acuerdos de privacidad. Algunos de ellos como son las redes sociales que con el fin de que toda la información que la persona brinde ya es totalmente regalada a dicha red social, es un simple ejemplo para mostrar de qué manera los datos que proporcionamos pueden tener como consecuencia actos maliciosos provocando ciberdelitos.

En la actualidad es importante la seguridad e integridad de los datos, puesto que las amenazas digitales están muy presentes y como consecuencia se comenten actos maliciosos, haciendo que la integridad de una persona o un dispositivo tecnológico sea expuesto.

Para ello, el sistema auditor de soportes de datos que se está presentado, tiene como objetivo principal de que dicho programa sea implementado en la Universidad Autónoma del Estado de México campus Valle de Chalco. Con el fin de que se evite la pérdida de datos, archivo y programas que se encuentren instalados por el personal autorizado de la institución. Y así mismo tener un respaldo por seguridad ya que las computadoras al tener el uso por alumnos de la universidad pueden tener actos inapropiados para los equipos de cómputo.

1.1. Planteamiento del problema

En la actualidad es importante la seguridad e integridad de los datos, puesto que las amenazas digitales están muy presentes y como consecuencia se comenten actos maliciosos, haciendo que la integridad de una persona o un dispositivo tecnológico sea expuesto.

En el ámbito de la información digital, es necesario realizar auditorías periódicas a los dispositivos computacionales, existen programas gratuitos en la red sobre la compresión de datos, pero algunos de ellos son software malicioso ya que contienen virus.

Al no realizar una instalación correcta, estos elementos se instalan y como consecuencia pueden dañar tanto el software como el hardware de una computadora, generando aún más problemática en la integridad del equipo de cómputo. Los programas que tienen buena calidad en la compresión de datos, suelen tener licenciamientos, de tal forma que permiten que el usuario tenga más opciones de herramientas para poder realizar un respaldo y compresión más seguro.

Como se sabe los licenciamientos de los programas tienen costos elevados e incluso muchos de ellos solo manejan suscripciones mensuales o anuales, lo que dificulta la adquisición de estas aplicaciones, tan necesarias, en nuestra institución.

1.2. Hipótesis

La implementación, de un sistema auditor que identifique el uso malicioso de los documentos que se encuentran en equipos de cómputo permitiendo proteger los derechos de autor en archivos de texto. En este sentido se presentan las siguientes interrogantes:

1. ¿Con un sistema auditor implementado en los equipos de los laboratorios del C.U UAEM Valle de Chalco se logrará proteger en su caso mantener intactos los documentos que se almacenen en las computadoras?
2. ¿El control en la modificación de documentos a través de programas auditores será un proceso eficiente para mantener el control adecuado de los documentos utilizados en nuestra institución?
3. ¿Tener una aplicación que ayude a la recuperación e integridad de información, aportará como evidencia para comprobar en uso inadecuado de los equipos?

En este proyecto de tesis se pretende responder estas cuestiones formuladas a través de la siguiente hipótesis: ¿La implementación un sistema auditor garantizará la reducción del uso inadecuado de los equipos de cómputo?

1.3. Objetivos

Los objetivos de este proyecto son los siguientes.

General: Desarrollar un sistema auditor forense que permita la recuperación y la detección de anomalías en archivos de textos ubicados en el soporte de datos de un equipo de cómputo.

Específicos:

1. Realizar una investigación profunda la cual permita conocer los conceptos básicos que son parte de los soportes de datos.
2. Analizar programas existentes.
3. Comparar las características y la funcionalidad de los programas identificados en el punto 2.
4. Diseñar un sistema que audite la integridad y la recuperación de soportes de datos.
5. Implementar el sistema diseñado.
6. Realizar pruebas al sistema desarrollado.
7. Presentar los resultados.

1.4. Metodología

Para llegar al objetivo de este proyecto, se utilizará una metodología de Proceso Unificado de Desarrollo de Software (PUDS) basado en lenguaje unificado de modelado (UML, por sus siglas en inglés, Unified Modeling Language). (Hernández, R. 2003).

En el caso de la recuperación de información y análisis de los requerimientos se utilizará la investigación documental:

- Recopilación de Información: Se realizará una investigación documental de las principales características y el funcionamiento sobre los sistemas que se encuentren en la red que tienen la función de recuperar información.
- Analizar los requerimientos: Se realizarán diagramas para cubrir los requisitos solicitados en los diagramas UML. Proporcionando una idea clara de cómo funcionara el sistema cuando el usuario haga uso de este.

En el caso de diseño e implementación del sistema se utilizará la investigación experimental:

- Diseñar un sistema: El prototipo de interfaz se realizará mediante diagramas UML en la fase de diseño.
- Implementar el sistema: Se realizará la instalación del sistema desarrollado con la finalidad de que algún usuario haga uso de él.

En la sección de pruebas se hará una investigación de campo:

- Pruebas del Sistema: Un usuario empleara el sistema y se corroborara el correcto funcionamiento del mismo.

1.5. Organización del documento

El resto del documento se organiza como sigue. El capítulo 2 permite conocer antecedentes sobre el tema propuesto, explicando las etapas que tiene la informática forense, conocer más sobre los soportes de datos y los delitos que se pueden cometer con el mismo. Teniendo el apartado de definiciones (respaldo, compresión, auditoría e integridad), ya que son los elementos sobresalientes de este proyecto propuesto. El capítulo 3, se explica de manera detallada los temas de "Código Huffman" y "Función Hash". Permitiendo conocer su funcionamiento de ambos, ya que estos dos elementos son el cerebro del sistema propuesto. En el capítulo 4 describe el proceso de análisis y el diseño de este software, describiendo los distintos diagramas que se necesitan para la elaboración de este proyecto, así como la interacción que permite tener este sistema. En el capítulo 5, se demostrará la Implementación del sistema, mostrando imágenes de su interfaz gráfica. Permitiendo conocer su funcionalidad. Por último, en el capítulo 6 se presentan las conclusiones y trabajo futuro de este proyecto.

2. ANTECEDENTES

“La Seguridad no es solo un proceso Tecnológico...

Es un proceso Organizacional”.

Anónimo.

Hoy en día vivimos en una sociedad en donde los delitos están muy presentes, siendo así que la informática no se salva de ello. Es por ello que se requieren herramientas que apoyen a la protección de la información o que permitan auditar los sistemas informáticos. En este sentido, la “Informática Forense” es una disciplina criminalística que como objetivo principal es realizar una investigación profunda en sistemas informáticos pasando por las fases de adquisición, análisis y presentación (Lázaro, F., 2013):

Adquisición: Es el procedimiento que permite adquirir los medios digitales que serán sometidos a un análisis, en el cual se trabajará con una copia de bajo nivel del soporte, para no dañar la imagen original. Para realizar la imagen de dicho soporte de almacenamiento, existen técnicas para la extracción de esta información. Esta copia de la imagen consiste en hacer una imagen exacta del soporte original tomando a consideración el espacio libre de los sistemas de archivos, los archivos eliminados, así como los datos que pudieron haber existido antes de ser formateado el dispositivo de almacenamiento, copiando la sección de particiones, los espacios libres del disco duro, la tabla de particiones, el sector de arranque e incluso las zonas reservadas, que generalmente son inaccesibles al sistema utilizadas por el fabricante para incluir la información especial.

Análisis: Esta fase consiste en la identificación, el estudio y la interpretación de los elementos que forman parte de la evidencia existente del soporte de datos. En esta fase se realiza un examen detallado de los sistemas de archivos y analiza el

contenido de los mismos. Realizando búsqueda de caracteres con el fin de encontrar anomalías en el soporte de datos, elabora estadísticas y ejecuta otras tareas que forman parte de la investigación. En la etapa final de la investigación consiste en interpretar los resultados y realizar un informe detallado del procedimiento exponiendo las anomalías que vio durante el estudio.

Presentación: Una vez que el investigador concluya el análisis, brinda un informe detallado a las personas que se van a encargar de utilizar esta información con fines prácticos o hacer valer ante tribunales estos actos que pueden ser maliciosos. Usando este reporte informativo como un elemento de evidencia.

En la actualidad los “Soportes de Datos” son elementos clave para la investigación de un acto malicioso ya que es una de las principales fuentes de evidencias durante algún caso de delito, en la informática forense.

El autor Francisco Lázaro define que un soporte de datos es el dispositivo físico de almacenamiento de información como lo son discos duros, memorias USB, Smartphone, reproductores MP3, etc. (Lázaro, F., 2013).

Los soportes de datos se clasifican en internos que son los dispositivos que se encuentran, como su nombre lo indica, en el interior de la computadora; y los externos que son elementos que se conectan a través de un puerto de conexión (por ejemplo: puertos USB). (Hamud, A., 2010).

Los soportes de datos tienen características que los pueden hacer vulnerables o consistentes en su uso, un ejemplo de estos atributos son los siguientes:

Riesgos: Los dispositivos que funcionan como soporte de datos son susceptibles a daños por un mal uso, la pérdida de este dispositivo, o que un virus informático se almacene en el soporte de datos, por mencionar algunos.

Destrucción segura: Los soportes de datos tienen cierto tiempo de vida, todo depende del uso adecuado que le de el usuario, la capacidad de almacenaje que

tenga, o presentar algún fallo técnico, teniendo como consecuencia que el usuario no pueda recuperar su información. Incluso si alguna persona no autorizada trata de acceder al soporte de datos no tendrá dicha información. Inservible el soporte de datos es recomendable realizar una destrucción total del dispositivo de almacenaje.

Borrado seguro: Algunos usuarios reutilizan soportes de datos adquiridos por otras personas, es importante realizar un borrado seguro, existe software libre que permite realizar este proceso.

Cifrados: Conociendo algunos de los riesgos que pueden tener los soportes de datos, es importante cifrar la información ya que una persona no autorizada podría acceder al soporte de datos y hacer un mal uso con la información que contiene el dispositivo.

Respaldos: Es sumamente importante realizar respaldo sobre el soporte de datos puesto que, por un virus, el mal uso del dispositivo de almacenamiento, o la manipulación de la información que almacena en el soporte de datos puede ser dañada, haciendo que el contenido que tiene el dispositivo pueda perderse y no recuperarse.

Existen infinidad de delitos informáticos, donde el soporte de datos es el factor principal para cometer algún acto malicioso, estos son algunos de los delitos que se conocen:

Manipulación de datos de entrada: También conocido como sustracción de datos, es el delito más conocido, ya que no es necesario tener conocimientos sólidos para acceder a las funciones básicas de una computadora siempre y cuando la persona que manipula la computadora tenga el acceso.

Manipulación de programas: La persona que realiza este tipo de acto malicioso debe tener conocimientos sólidos de las computadoras y sus funciones, este delito

consta de configurar los programas que se encuentran en el equipo e instalar software nuevo. Los programadores que realizan este tipo de acciones consiste en poner en cubierta la computadora y se manipule sin que el usuario este enterado de esta acción.

Manipulación de los datos de salida: Este delito se caracteriza por que es común que en las computadoras que contiene el cajero automático, mediante la alteración de las instrucciones que debe ejecutar el sistema, ya que existen programas que codifican la información, falsificándola por medio de las tarjetas que acceden al cajero.

Falsificaciones informáticas: El equipo de computo juega el papel más importante ya que este permite realizar acciones maliciosas y el soporte de datos es versátil para realizar algún acto sospechoso en el equipo, siendo el factor importante la computadora interactúa de dos maneras, como objeto, donde se modifican los documentos o los programas almacenados e instrumento que permite la falsificación de documentos (comerciales, educativos, médicos, etc.).

Sabotaje informático: Consiste en eliminar o alterar funciones o información que contiene el soporte de datos, con la consecuencia de limitar las funciones del sistema, un ejemplo de este tipo de sabotajes son los virus y gusanos.

Hackers: Su acceso lo efectúan a través de la red, el delincuente accede violando las medidas de seguridad de la computadora y de esta manera puede robar la información almacenada en el soporte de datos.

Conociendo estas características sobre los riesgos que se encuentra un soporte de datos es sumamente importante realizar un respaldo del mismo, ya que, si se llega a presentar algún acto de los ya mencionados, podríamos saber si hemos sido parte de algún delito.

2.1. Respaldo y compresión

Un respaldo consiste en la realización de una copia de la información más importante de un soporte de datos, y puede ser almacenada dicha copia en el mismo soporte de datos o en otro dispositivo de almacenaje. Los respaldos tienen como objetivo principal la restauración de archivos individuales o de sistemas de archivos completos.

El usuario al eliminar algún archivo o una carpeta de manera accidental, la opción principal es restaurar la última copia de respaldo que se realizó al equipo, pero se debe considerar con qué frecuencia de tiempo se realiza dicho respaldo ya que puede que el archivo no se encuentre en el momento que se hace la copia de seguridad.

Los datos que cambian con poca frecuencia se pueden respaldar de manera limitada, mientras que los datos que cambian a regularmente deben ser copiados periódicamente.

Existen tres tipos de respaldos:

- a) Respaldo Completo: Su característica principal consiste en realizar una copia total de los documentos que se encuentren en el soporte de datos, es decir si se perdiese la información esta se pueda usar en su totalidad como se mantenía en el equipo.
- b) Respaldo de Incremento: Con el último respaldo completo ya realizado, solo hace una copia de seguridad a los documentos que han sido modificados y solo agregara al respaldo los últimos cambios realizados.
- c) Respaldo diferencial: Únicamente copia los archivos y directorios que han sido creados o modificados desde el último respaldo realizado.

La compresión consiste en una copia de un archivo de menor tamaño que

el archivo original, para realizar este proceso se utiliza software que funciona con métodos y funciones de algoritmos matemáticos, los cuales tienen como finalidad realizar una copia de menor tamaño de un archivo. (Cordova, C., 2010).

Al comprimir algún archivo, la copia que se crea es totalmente ilegible, de esta manera mantiene asegurado el archivo; para visualizar el archivo es necesario realizar una descompresión.

Existe software que permite la compresión de documentos, ejemplo de ello, son:

WinRAR: es un compresor de archivos que permite mayor compresión que otras herramientas, especialmente en ficheros ejecutables, bibliotecas de objetos y grandes archivos de texto, dispone de un algoritmo de compresión altamente optimizado para datos multimedia, así como de compresión sólida. A continuación, algunas características:

- a) Seguro: Protección con contraseña con cifrado Rijndael (AES-128), bloqueo contra modificaciones, protección avanzada contra daños que permite recuperar ficheros en mal estado, verificación anti-virus configurable y eliminación segura para eliminar datos sensibles.
- b) Escalable: RAR es el primer compresor que implementa manejo de ficheros de 64 bits, lo que le permite manejar grandes cantidades de ficheros y tamaños muy grandes (solo limitado por el sistema operativo).
- c) Fácil de usar: Asistente para usuarios noveles, integración con el Explorador de Windows.
- d) Compatible: Programas gratuitos de descompresión de archivos RAR para múltiples plataformas (Windows, OS/2, Mac OSX, BeOS, Linux, FreeBSD, Solaris, BSD Unix, Irix, Pocket PC).
- e) Ideal para copias de seguridad: Todas las operaciones de compresión y

descompresión pueden automatizarse a través de su completa interface de línea de órdenes, puede almacenar varias versiones de cada fichero y además puede añadir información redundante que permite recuperar datos de ficheros dañados (WinRAR©, 2017).

WinZip: es un compresor de archivos comercial que funciona en Microsoft Windows, desarrollado por WinZip Computing (antes conocido como Nico Mak Computing). Es un producto comercial con una versión de evaluación gratuita. En la versión más reciente de WinZip se destacan:

- a) Extracción de archivos RAR y BZ2.
- b) Compresión mejorada de archivos de audio WAV.
- c) Vista de imágenes en miniatura.
- d) Visor interno de imágenes.
- e) Selección automática del método de compresión.
- f) Grabación de archivos Zip existentes en CD o DVD.
- g) Programación del Asistente de WinZip para tareas para enviar por correo electrónico informes y archivos Zip de forma automática.

A partir de la versión 10, ofrece una nueva vista estilo Explorador, permitiendo trabajar con estructuras complejas de archivos comprimidos; búsqueda automática de actualizaciones; y soporte para Administración de Datos Adjuntos, el cual permite alertar a los usuarios sobre archivos potencialmente peligrosos. (Copyright© 2017 Corel Corporation, 2017).

De esta manera nos damos cuenta que estas aplicaciones brindan opciones muy similares para la compresión y descompresión de soportes de datos, considerando la reducción del tamaño del archivo comprimido. Es importante que un soporte de datos este sujeto a auditorias para proteger su integridad.

2.2. Auditoría e integridad

Es el proceso que evalúa y analiza la información de una empresa o una entidad, con el fin de comprobar eficacia y eficiencia sobre su integridad en las áreas, tales como las administrativas, financieras, operativas, crédito, fiscales, informáticas. (Hernández, E., 1993).

Es importante conocer las fases por la cual está conformada la auditoría:

- a) Planificación: El auditor decidirá la información con la cual trabajará, y determinará el objetivo de la realización de la auditoría.
- b) Revisión preliminar: Se comenzará a analizar la información, el auditor realizará una documentación en la cual hará las observaciones necesarias para conocer un estatus en el que se encuentra la empresa o la organización que lo contrato.
- c) Trabajo de campo: El auditor realizará un estudio más profundo sobre la información, las personas que conforma la empresa y comenzará a determinar los procesos y el tipo de operación que hay en la empresa, identificará las áreas que no están cumpliendo con su objetivo.
- d) Informe de auditoría: Durante el proceso de evaluación el auditor tendrá que ir realizando un estudio donde explicará de manera detallada el proceso de la auditoría y expresará que áreas son las que necesitan mejor desempeño.

En el área informática es importante realizar constantes auditorías al soporte de datos y sistema para conocer la integridad que salvaguardara la información.

El objetivo principal de la auditoría informática consiste en analizar la eficiencia de un sistema informático, conocer la integridad de los datos. Con ayuda

de software especiales para realizar la auditoría y tener un resultado exacto sobre el estatus que se encuentra el soporte de datos o el equipo de cómputo.

Los tipos de la auditoría informática son los siguientes:

Auditoría de la gestión: Se refiere a la contratación de servicios y licenciamientos de software.

Auditoría de los datos: Trata sobre la clasificación de los datos, estudio de las aplicaciones y análisis de los diagramas que conforma ese sistema de información.

Auditoría de las bases de datos: Analiza los controles de acceso, actualización, y la seguridad de los datos.

Auditoría de la seguridad: Es referente a la información verificando disponibilidad, integridad, confidencialidad, autenticación, con el fin de mantener protegida dicha información.

Auditoría de la seguridad lógica: Comprende del estudio sobre los métodos de autenticación de los sistemas de información.

Auditoría de la seguridad en producción: Se enfoca a errores, accidentes y fraudes, que puede ser desarrollada en un equipo informático. (Hernández, E., 1993).

La integridad se refiere a la calidad, seguridad sobre un documento, software, dispositivo físico de un dispositivo tecnológico, o un soporte de datos. (Hall, A. & Devoto, M.,2000).

Si esta integridad es modificada por algún usuario no autorizado puede provocar errores, fallos y pérdidas importantes poniendo en riesgo la seguridad y disponibilidad del soporte de datos o el dispositivo tecnológico.

3. HERRAMIENTAS UTILIZADAS

“Una computadora puede ser llamada “inteligente” si logra engañar a una persona haciéndole creer que es humano”

Alan Turing

En 1952, David A. Huffman mientras era estudiante de postgrado en el Instituto Tecnológico de Massachusetts (MIT), propuso un algoritmo de compresión de datos, el cual implicaba el uso de árboles binarios. El método fue publicado en una revista teniendo como título *“Method for the Construction of Minimum-Redundancy Codes”* (*Un método para la construcción de los códigos mínimos de redundancia*). (Sánchez, V., 2010)

También conocido como “código Huffman” se ocupa para la compresión o encriptación de caracteres mediante la frecuencia de repetición de cada uno de ellos, asignando cierta codificación de distinta longitud a cada carácter.

Entre más repetido sea un carácter menor cantidad de bits tendrá en su código, y si un carácter tiene menos repeticiones mayor cantidad de bits tendrá en la codificación que obtendrá. (Morales, G., 2010).

El algoritmo Huffman consiste en realizar el conteo de caracteres en un archivo y realizando una tabla de frecuencia; una vez obtenida la frecuencia de cada carácter se realiza la creación de un árbol binario, donde cada hoja (nodo hijo) tendrá una variante con valor (carácter y frecuencia), obteniendo una raíz (nodo padre), cada rama tendrá un valor binario (rama izquierda 0, rama derecha 1), la cual al comenzar a buscar el camino de un símbolo se obtendrá el código Huffman de cada caracter. (Sayood, K., 2006).

El procedimiento para realizar un árbol binario con el método Huffman es el siguiente:

1. Se tiene la palabra “GENCIENCIA” se comienza a realizar el conteo de la frecuencia de los caracteres, mostrados en la Tabla 1, se ordenan de menor a mayor, conforme al orden en su aparición en la palabra, véase en la Tabla 2.

Tabla 3–1. Obtención y ordenación de la frecuencia de cada carácter. (Fuente: Propia, 2017).

Caracter	Frecuencia		Caracter	Frecuencia
G	1	⇒	G	1
E	2		A	1
N	2		E	2
C	2		N	2
I	2		C	2
A	1		I	2

2. Se suman los primeros dos valores de la tabla dos y se comienza a formar el árbol binario.

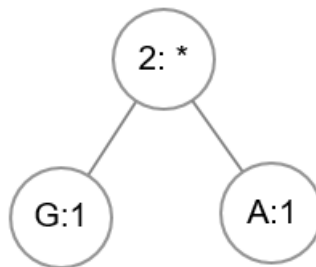


Figura 3.1 Primer árbol generado, que tiene como raíz la suma de los dos primeros valores de la frecuencia en el orden de menor a mayor. (Fuente: Propia, 2017).

3. Los números sumados se sustituyen en la Tabla 1, por el resultado obtenido en la suma del primer árbol generado.

Tabla 3–2. Obtención y ordenación de la frecuencia de cada carácter. (Fuente: Propia, 2017).

Caracter	Frecuencia
*	2
E	2
N	2
C	2
I	2

4. Se repite el paso 3, y se obtiene el siguiente resultado del árbol y tabla de frecuencia.

Caracter	Frecuencia
*	4
N	2
C	2
I	2

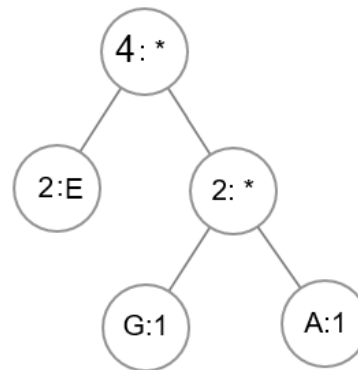


Figura 3.2 Primer árbol generado, que tiene como raíz la suma de los dos primeros valores de la frecuencia en el orden de menor a mayor. (Fuente: Propia, 2017).

5. De manera sucesiva, se repite el proceso sumando los dos primeros valores de la tabla e insertándolo el resultado en el árbol. El proceso completo se muestra a continuación.

Caracter	Frecuencia
*	6
C	2
I	2

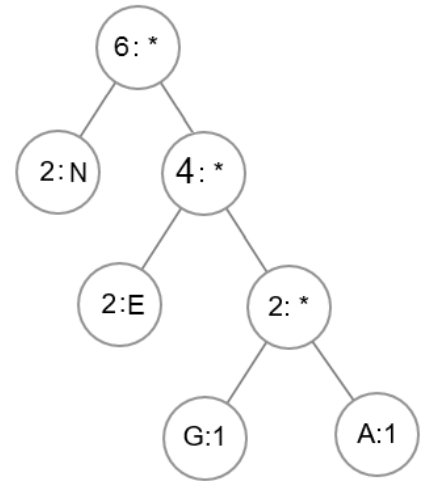


Figura 3.3 Segundo árbol generado, que tiene el nodo de la suma de las frecuencias 4 y 2.
(Fuente: Propia, 2017).

Caracter	Frecuencia
*	8
I	2

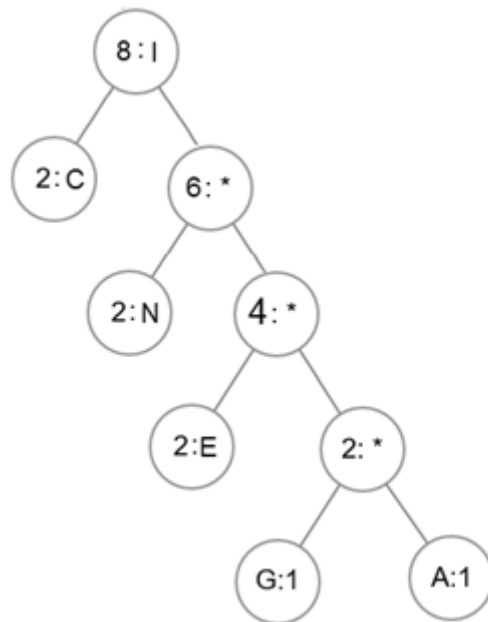


Figura 3.4 Tercer árbol generado, que tiene el nodo de la suma de las frecuencias 6 y 2.
(Fuente: Propia, 2017).

Caracter	Frecuencia
*	10

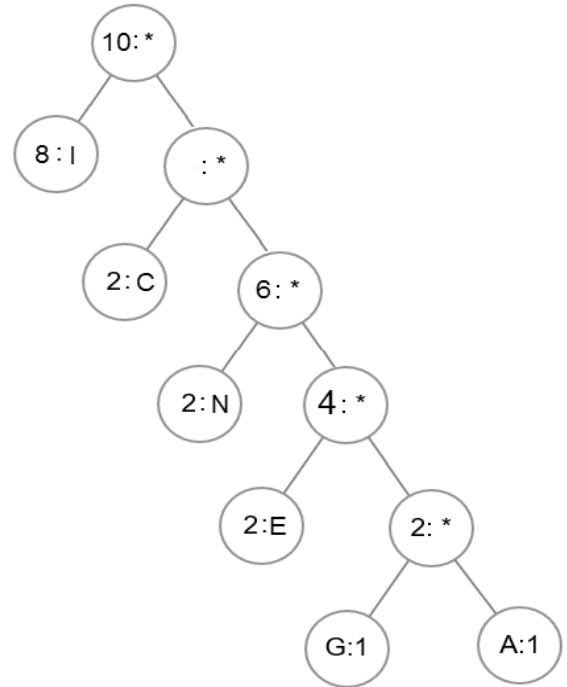


Figura 3.5 Cuarto árbol generado, que tiene el nodo de la suma de las frecuencias 8 y 2.
(Fuente: Propia, 2017).

- Al término de sumar las frecuencias de los caracteres se coloca un 0 a los nodos que se encuentren a la izquierda y un 1 a los nodos ubicados de lado derecho (veáse en la Fig. 6), para obtener el código de cada carácter, que se muestra en la Tabla 3.

Tabla 3–3. Código Huffman obtenido después del proceso. (Fuente: Propia, 2017).

Caracter	Frecuencia
G	11110
E	1110
N	110
C	10
I	0
A	1111

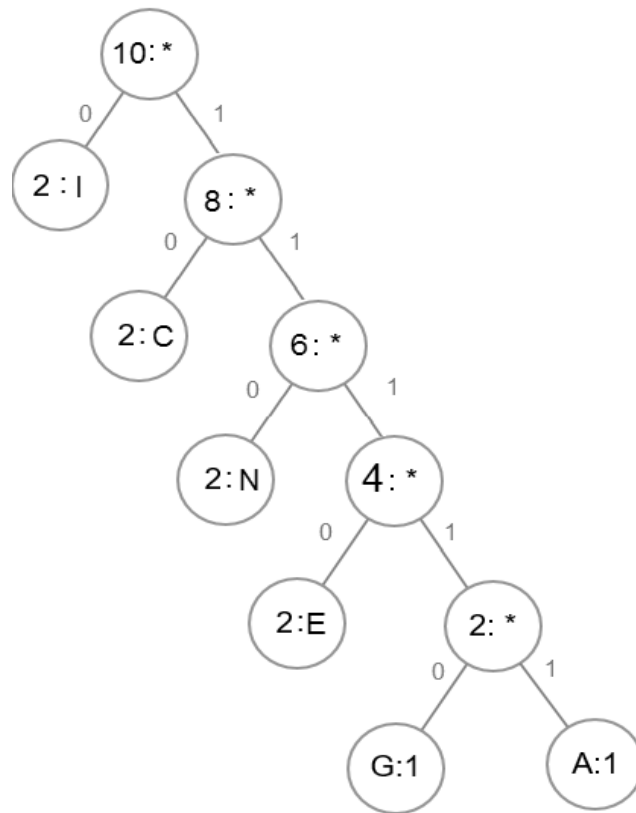


Figura 3.6 Árbol generado con la codificación Huffman. (Fuente: Propia, 2017).

Los códigos permiten comprimir la palabra “GENCIENCIA”, al sustituir la codificación de cada letra en la posición en la que se encuentra en la palabra. Cada letra está formada por 8 bits y al codificar resulta un valor distinto en la cantidad de bits.

No Codificado:

GENCIENCIA = 10 caracteres x 8 bits (1 byte) = 80 bits

Codificado

11110 1110 110 10 0 1110 110 10 0 11111 = 25 bits
 G E N C I E N C I A

3.1. Función Hash

La función hash (picadillo), es un algoritmo de reducción criptográfico de 128 bits representado por un número de 32 números hexadecimales, teniendo como objetivo principal hacer que un archivo no sea modificado. A partir de una entrada de caracteres de un archivo binario, alfanumérico, entre otros, teniendo una salida alfanumérica que servirá de protección para conocer la modificación que se pueda realizar a un archivo. (Medina, J., 2009).

En el año 1995 Ronald Rivest desarrollo la función MD5 (Algoritmo de Resumen del Mensaje 5), al estudiar versiones antiguas de otros hashes, se percató de que mostraban problemas, teniendo el riesgo de que esas funciones fueran inseguras.

Las funciones picadillo soy muy utilizadas, ya que una de sus características principales es proteger la integridad de un archivo que se almacene en un soporte de datos.

El procedimiento para conocer si un archivo ha sido modificado consiste en realizar un hash del archivo original, obteniendo una copia y para terminar se vuelve a extraer el hash de la copia. Se hace una comparación de los hashes y si es que el archivo es modificado al menos por un carácter, el hash cambiará y se sabrá si es que ha sido alterado ese archivo.

Según el autor Douglas R. Stinson (2006), la familia del hash se compone de cuatro características importantes (X, Y, K, H), las cuales tienen las siguientes condiciones

X: es un conjunto de posibles mensajes.

Y: es un conjunto finito de posibles claves o etiquetas de autenticación.

K: es el conjunto de datos asociados o las diversas claves finitas de autenticación que puede tener un mensaje.

H: $K \in K$, y tiene una función hash, $h_k \in H$ cada $h_k: X \rightarrow Y$

En la definición anterior X podría ser infinita mientras que Y siempre es conjunto finito. Si X es conjunto finito, la función hash es algunas veces llamada como “Funcion de compresión” en esta situación se asume que (X, Y) se denomina par válido bajo la llave k si $hk(X) = Y$.

Las principales características de las funciones hash son las siguientes:

a) Resistencia a la Preimagen:

- Significa que dado cualquier valor hash “ y ”, es computacionalmente imposible encontrar el valor de la preimagen “ x ” usada tal que $h(x)$.
- Otra forma como se conoce esta propiedad es que la función hash sea de un solo sentido.

b) Resistencia a 2° preimagen:

- Significa que dado “ x ”, es computacionalmente imposible encontrar una x' tal que $h(x)=h(x')$. Otra forma de conocer esta propiedad es que “ h ” sea resistente a una colisión suave.

c) Resistencia a colisión:

- Significa que es computacionalmente imposible encontrar dos mensajes “ x ”, x' tal que $h(x)=h(x')$

4. SISTEMA COMPRESOR DE ARCHIVOS UAEMEX

“La tecnología no es nada. Lo importante es que tengas fe en la gente, que sean básicamente buenas e inteligentes, y si les das herramientas, harán cosas maravillosas con ellas.”

Steve Jobs.

En este capítulo se desarrolla el análisis y diseño del sistema propuesto, permitiendo conocer la estructura, la función y la interfaz que integra este software.

4.1. Casos de Usos

En los diagramas de casos de uso que se presentan a continuación, se visualiza cómo el usuario manipula el sistema propuesto, lo complementara una tabla donde se explica de manera más explícita el funcionamiento interno del caso de uso. (Fowler, M., & Scott, K., 1999).

La Fig. 4.1 muestra el diagrama general de casos de usos, que contiene tres subcasos llamados: Registro Usuarios, Comprimir y Descomprimir; y un actor denominado “Usuario”.

La funcionalidad general del sistema es que el Usuario seleccione un archivo, lo codifique usando el caso de uso “Comprimir” y los decodifique con el caso de uso “Descomprimir”.

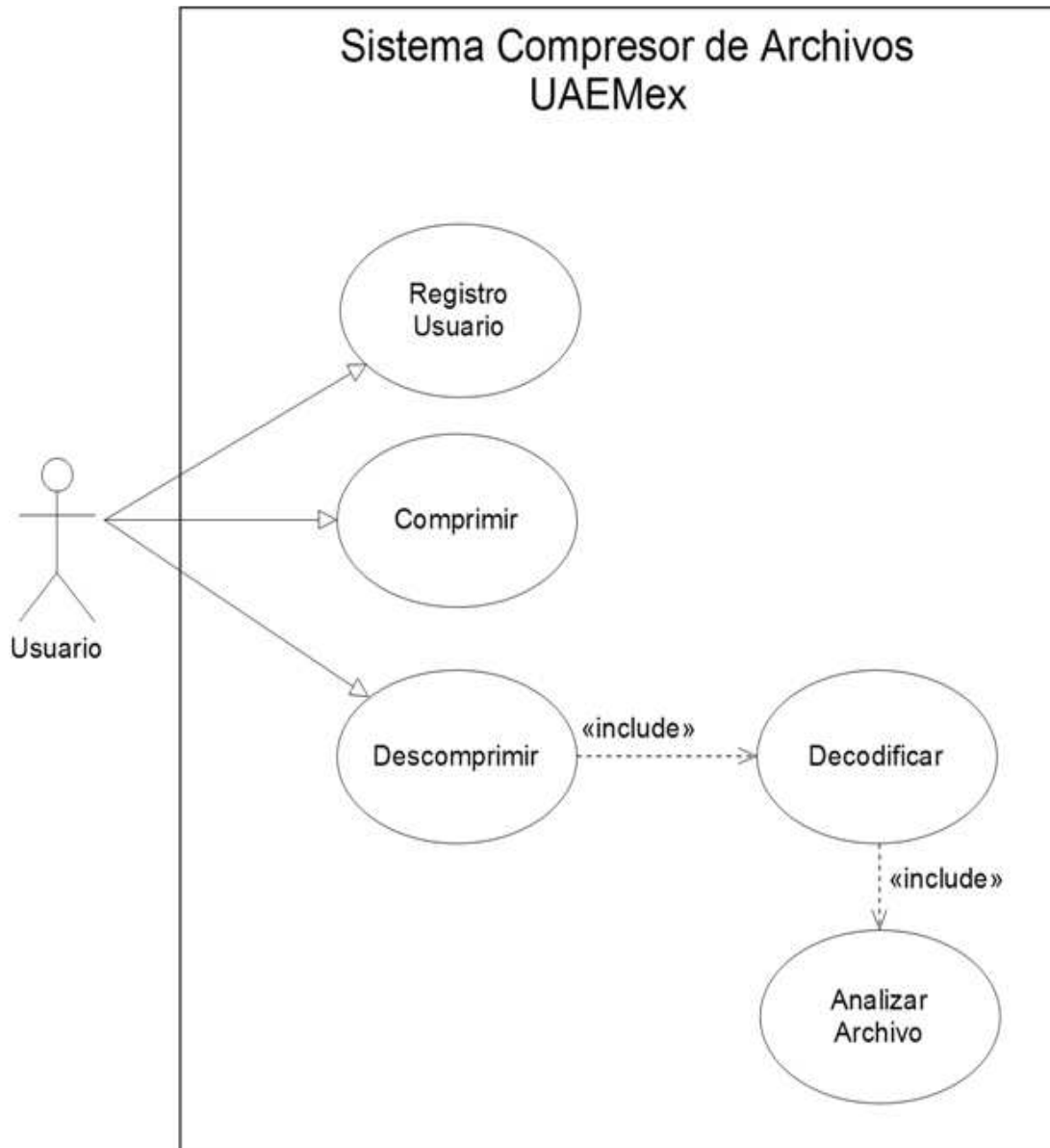


Figura 4.1 Diagrama general de casos de usos. (Fuente: Propia, 2017).

La primera actividad es el caso de uso “Regisro Usuarios”, que se divide en “Agregar Usuario” e “Iniciar Sesión”. El actor dispara ambos casos, las Tablas 4.1 y 4.2 describen la funcionalidad de ambos casos, respectivamente.

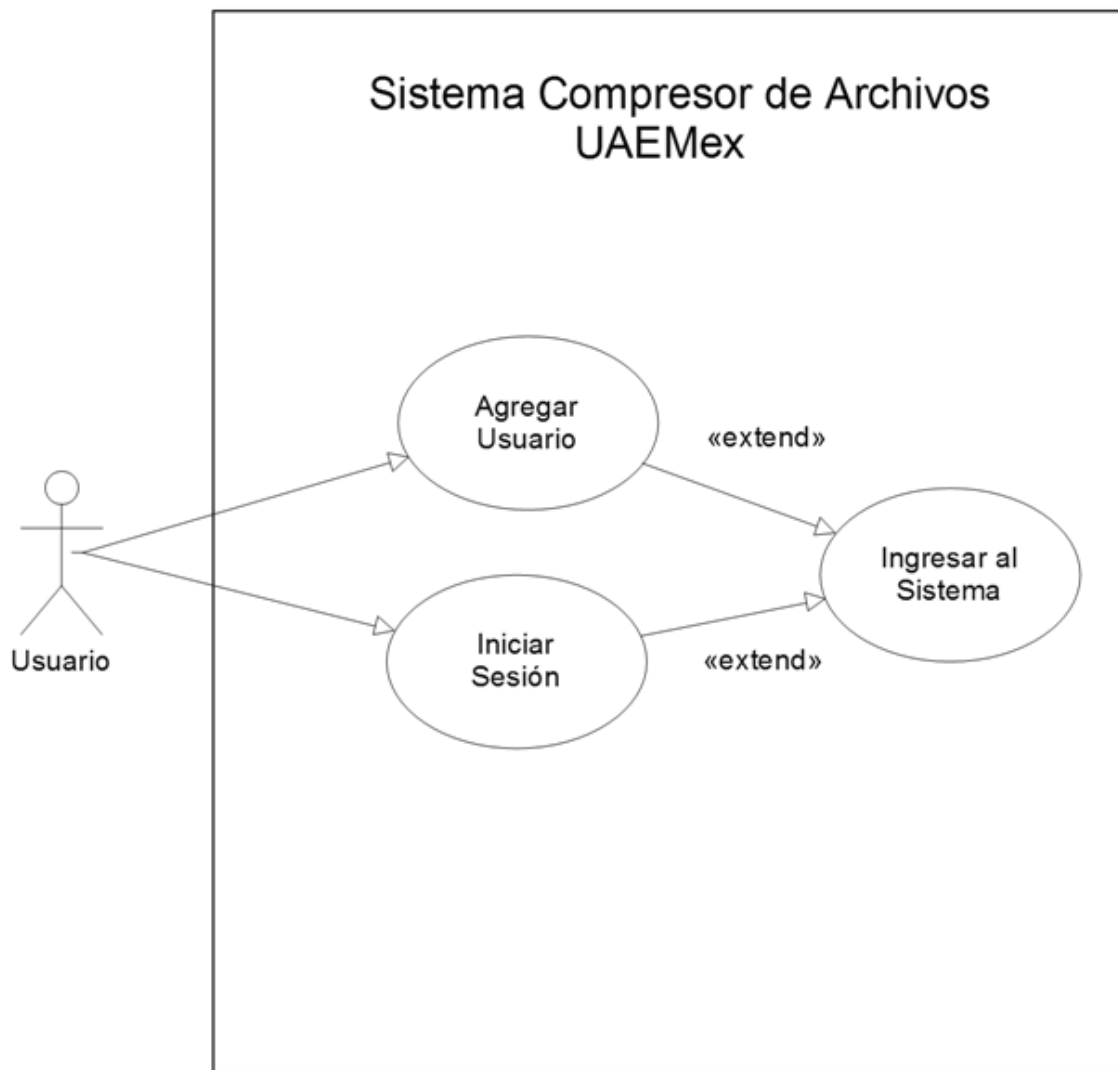


Figura 4.2 Caso de uso "Registro Usuarios". (Fuente: Propia, 2017).

Tabla 4–1. Especificaciones del caso de uso Agregar Usuario. (Fuente: Propia, 2017).

Nombre: Agregar Usuario

Actor: Usuario

Descripción: El usuario realiza un registro en el sistema para poder acceder.

Disparador: Seleccionar botón “Nuevo Usuario”

Precondiciones: Seleccionar el botón Nuevo Usuario

Postcondiciones: Se tiene un registro del usuario, con un login, un password y una carpeta de respaldo.

Flujo Normal:

1. Seleccionar el botón Nuevo usuario. (S.1)
2. Agregar un nombre de usuario.
3. Colocar una contraseña. (S.2)
4. Dar nombre de la carpeta que se creará.
5. Seleccionar el botón Registrar. (S.3) (E.1)

Flujo alternativo:

- S.1 El usuario no da clic en el botón Nuevo Usuario, regresa al paso 1.
- S.2 El usuario cierra el sistema.
- S.3 Si el usuario coloca sus datos y son inválidos el sistema mandará un mensaje de error. Regresa al paso 1.

Excepciones:

E1: Si el usuario ya está registrado, el sistema mandará un mensaje de que el registro ya ha sido realizado.

Prioridad: Alta

Frecuencia de uso: Alta

Reglas: El usuario debe tener un nombre de usuario y una contraseña que proporcionar.

Tabla 4–2. Especificaciones del caso de uso Iniciar Sesión. (Fuente: Propia, 2017).

Nombre: Iniciar Sesión

Actores: Usuario

Descripción: El usuario accede al sistema con un nombre de sesión y una contraseña

Disparador: Seleccionar botón "Iniciar Sesión"

Precondiciones: El usuario debe estar previamente agregado

Postcondiciones: Se genera un archivo con una reducción de tamaño en bits que el original y se coloca en la carpeta de respaldo.

Flujo Normal:

1. Agregar un nombre de usuario.
2. Colocar una contraseña. (S.1)
3. Seleccionar el botón Iniciar sesión. (S.2) (E.1)

Flujo alternativo:

- S.3 Si el usuario al iniciar sesión coloca sus datos y son erróneos el sistema mandara un mensaje de error. Regresa al paso 1.
- S.2 El usuario cierra el sistema.

Excepciones:

E1: Si el usuario no se encuentra, el sistema mostrará un mensaje de error. Regresa al paso 1.

Prioridad: Alta

Frecuencia de uso: Alta

Reglas: El usuario debe tener un nombre de usuario y una contraseña que proporcionar, para acceder al sistema.

Al acceder al sistema, el usuario está en posibilidad de realizar las tareas tanto de compresión como de descompresión. En la primera de ellas, el usuario debe seleccionar el archivo que puede estar ubicado en cualquier parte del disco duro, mientras que, para descomprimir, es necesario que se seleccione directamente de la carpeta de respaldo. La Fig. 4.2 muestra el caso de uso de "Comprimir" que está compuesto por cuatro subcasos de uso. Las Tablas 4.3 a la 4.5 muestran a detalle la funcionalidad de este caso de uso.

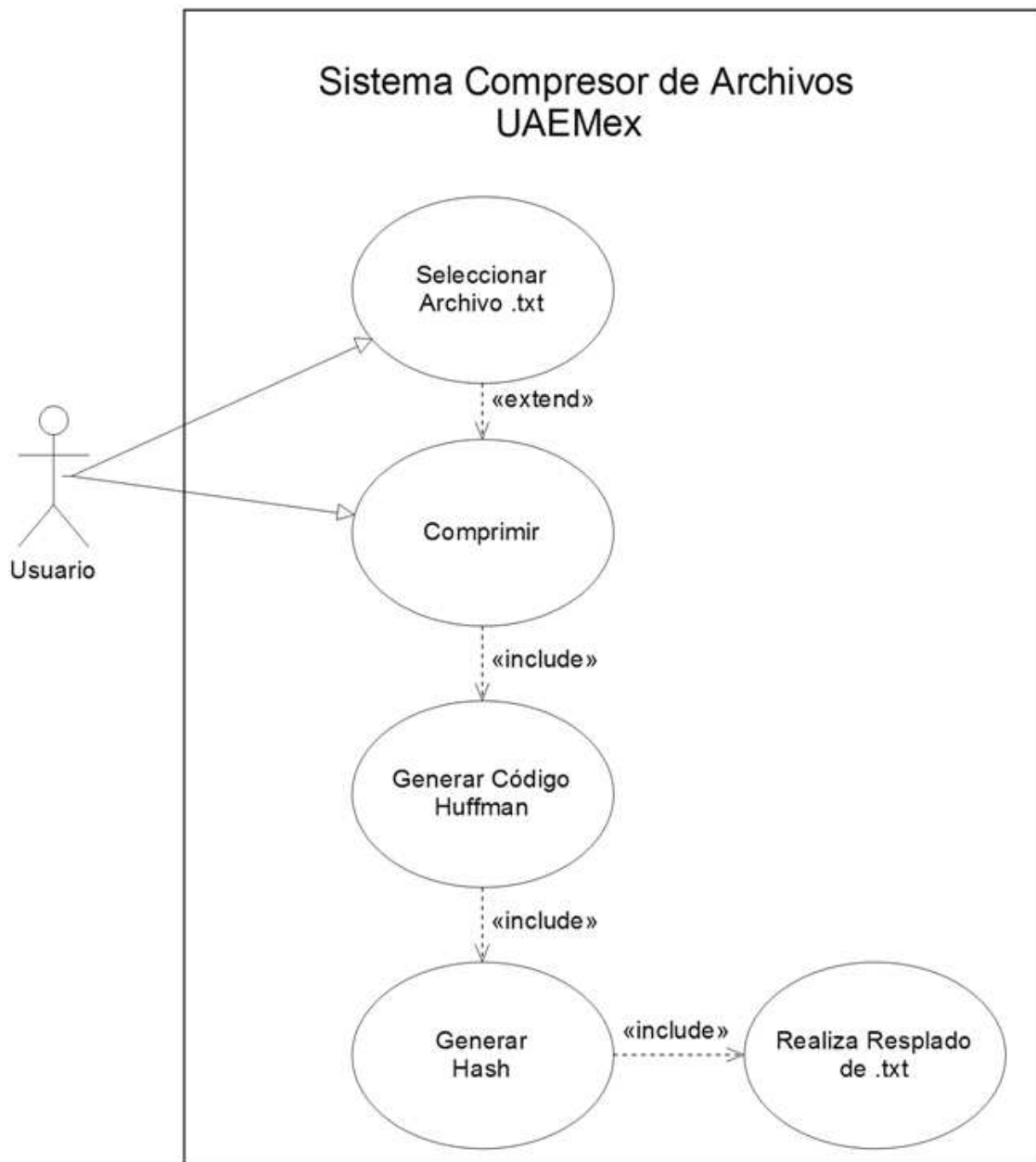


Figura 4.3 Caso de uso "Comprimir". (Fuente: Propia, 2017).

Tabla 4–3. Especificaciones del caso de uso Seleccionar Archivo. (Fuente: Propia, 2017).

Nombre: Seleccionar Archivo

Actor: Usuario

Descripción: El usuario examinará la ruta del archivo

Disparador: Presionar el botón Examinar

Precondiciones: El sistema debe estar abierto, el archivo debe existir y tiene que ser .txt

Postcondiciones: El archivo quedará seleccionado, ya seleccionado el archivo se activará la opción Comprimir

Flujo Normal:

1. El usuario abre el sistema (E2)
2. Da clic en el botón Examinar (E1)
3. Busca el archivo (S1)
4. Selecciona el archivo
5. El sistema verifica si el archivo es txt (S2)
6. El archivo quedara seleccionado (S3)

Flujo alternativo:

- S1. El archivo no existe
 - S.1.1 El usuario cancela la búsqueda
 - S1.2 El usuario cierra el sistema
- S2. El archivo no es txt
 - S2.1 Envía un mensaje de que el archivo no es .txt
 - S2.2 El sistema pregunta si quiere buscar de nuevo y regresa a paso 3
- S3 El archivo .txt queda seleccionado
 - S3.1 Finaliza el caso de usos de Selección de archivo

Excepciones:

E1: El usuario se equivocó de documento y cancela el proceso y regresa a paso 2
E2: El usuario cierra accidentalmente el sistema y regresa al paso 1.
E3: El usuario no selecciona el documento y el sistema no realizara ninguna acción
E4: El documento seleccionado tiene extensión .txt, pero no tiene contenido, el archivo aun así quedara seleccionado y continuara al proceso de codificación

Prioridad: Alta

Frecuencia de uso: Alta

Reglas: Seleccionar documento .txt

Tabla 4–4. Especificaciones del caso de uso Generar código Huffman y Hash.
(Fuente: Propia, 2017).

Nombre: Generar Código Huffman y Hash

Actores: Sistema

Descripción: El archivo .txt ya seleccionado se le realizara la codificación Huffman

Disparador: Seleccionar el botón comprimir

Precondiciones: El archivo .txt debe estar seleccionado

Postcondiciones: Archivo comprimido que contiene el árbol y código Huffman

Flujo Normal:

1. Analizar Frecuencia (S1)
2. Generar Tabla de frecuencia (S2)
3. Almacenar frecuencia
4. Generar árbol
5. Realizar recorrido del árbol para obtener su codificación
6. Almacenara recorrido
7. Obtiene Código Huffman
8. Calcula la Función Hash (S4)

Flujo alternativo:

S1. El archivo no está seleccionado y regresa a “Selección de Archivo” caso de usos 1
S.1.1 El usuario no da clic en el botón comprimir y regresa al paso 2
S1.2 El usuario cierra el sistema
S2. El usuario cancela la acción, y el sistema manda mensaje de error y se cierra el sistema
S3 Si el archivo seleccionado tiene extensión .txt, pero no contiene contenido realiza todos los pasos del flujo normal (2 al 8)
S4 Se finaliza caso de usos de “Generar Código Huffman”

Excepciones:

E1: Si el documento no contiene información, generara un archivo de Código Huffman vacío, y también un archivo de Función Hash
E2: Si el documento vacío no contiene información por lógica no tendrá frecuencia, y no generará árbol.
E3: El archivo de la Función Hash tendrá un código porque solamente contara un espacio que viene definido ya en los procesadores de textos y se puede apreciar porque es donde se encuentra el cursor.

Prioridad: Alta

Frecuencia de uso: Alta

Reglas: Seleccionar el botón comprimir

Tabla 4–5. Especificaciones del caso de uso Respaldo. (Fuente: Propia, 2017).

Nombre: Respaldo

Actores: Sistema

Descripción: El sistema realiza el respaldo del documento .txt original, guardándolo en una carpeta oculta en el sistema.

Disparador: El sistema tendrá una carpeta definida en la cual se realizará el respaldo.

Precondiciones: El archivo debe estar comprimido, tener código Huffman y

Hash.

Postcondiciones: El documento .txt original se quedará en su ubicación de origen, los archivos generados Huffman y Hash se almacenarán en la misma carpeta de respaldo.

Flujo Normal:

1. Teniendo los códigos de recorrido se generará el archivo del código Huffman y se realiza la compresión
2. Teniendo el archivo del código Huffman a este se le realiza la Función Hash

Flujo alternativo:

- S1. El archivo no existe
 - S.1.1 El usuario cancela la búsqueda
 - S1.2 El usuario cierra el sistema
- S2. El archivo no es txt
 - S2.1 Envía un mensaje de que el archivo no es .txt
 - S2.2 El sistema pregunta si quiere buscar de nuevo y regresa a paso 3
- S3 El archivo .txt queda seleccionado
 - S3.1 Finaliza el caso de usos de Selección de archivo.

Excepciones:

- E1: El usuario se equivocó de documento y cancela el proceso y regresa a paso 2
- E2: El usuario cierra accidentalmente el sistema y regresa al paso 1
- E3: El usuario no selecciona el documento y el sistema no realizará ninguna acción.
- E4. El documento seleccionado tiene extensión .txt, pero no tiene contenido, el archivo aún así quedará seleccionado y continuará al proceso de codificación.

Prioridad: Alta

Frecuencia de uso: Alta

Reglas: Seleccionar documento txt

Por último, el caso de uso descomprimir que tiene tres sub casos de uso. En esta actividad se requiere que el usuario ya tenga información en la carpeta de respaldo, de lo contrario, no será posible descomprimir un archivo que no ha sido comprimido.

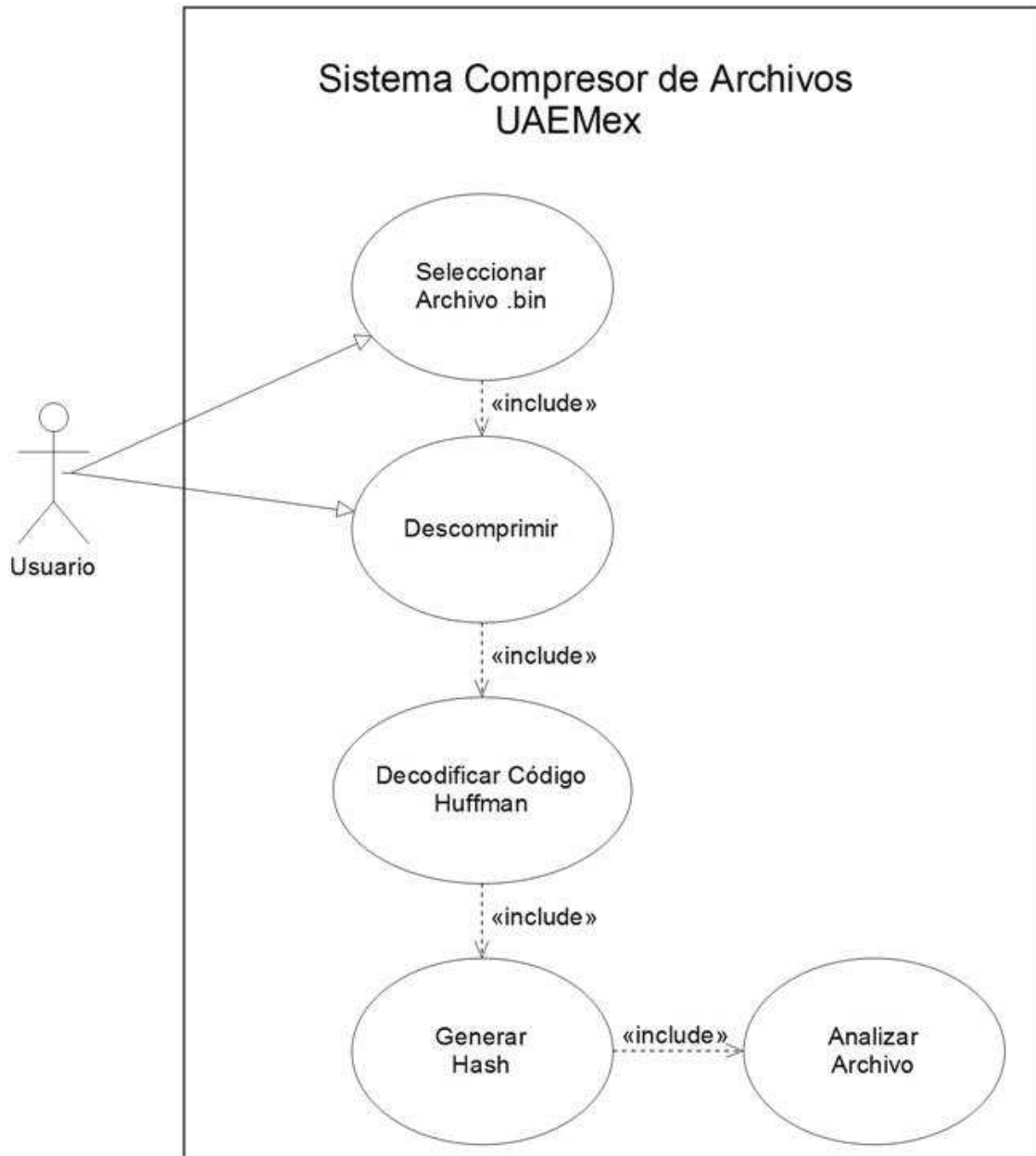


Figura 4.4 Caso de uso "Descomprimir". (Fuente: Propia, 2017).

Las Tablas 4.6 y 4.7, muestran el funcionamiento del caso de uso “Descomprimir”.

Tabla 4–6. Especificaciones del caso de uso Seleccionar archivo y decodificar Huffman.

(Fuente: Propia, 2017).

Nombre: Seleccionar archivo y Decodificar Huffman

Actores: Usuario

Descripción: El usuario examina la carpeta de respaldo

Disparador: Presiona el botón Descomprimir

Precondiciones:

El sistema debe estar abierto

El archivo de Código Huffman estará en la carpeta de Respaldo

Postcondiciones:

El archivo quedará seleccionado

Ya seleccionado el archivo se activará la opción Descomprimir

Flujo Normal:

1. El usuario abre el sistema
2. Da clic en el botón Examinar
3. Selecciona el archivo Código Huffman con extensión .bin del documento .txt seleccionado (S2) (S3)
4. Se activa el botón Descomprimir
5. El usuario da clic en el botón Descomprimir (S1)
6. El sistema genera Código Huffman para la descompresión y la función Hash

Flujo alternativo:

S.1.El usuario no selecciona la opción del menú correcta y cancela la acción, cierra sistema y regresa al paso 1

S2. El usuario no selecciona el archivo correcto

S2.1 Envía un mensaje de que el archivo no es .bin, cierra mensaje y regresa al

Paso 2.

S3 El archivo .bin queda seleccionado

S3.1 Finaliza el caso de usos de Selección de archivo

Excepciones:

E1: El usuario se equivocó de archivo y cancela el proceso y regresa a paso 1
E2: El usuario cierra accidentalmente el sistema y regresa al paso 1
E3: El usuario no selecciona el documento y el sistema no realiza ninguna acción.

Prioridad: Alta

Frecuencia de uso: Alta

Reglas: Seleccionar archivo .bin

Tabla 4–7. Especificaciones del caso de uso Analizar Archivos. (Fuente: Propia, 2017).

Nombre: Analizar Archivos

Actores: Usuario

Descripción: El usuario realizar un análisis sobre el archivo que comprimió para conocer si fue modificado.

Disparador: Seleccionar botón Analizar Archivos

Precondiciones: Los archivos deben estar en la carpeta de respaldo y debe ser extensión .md5

Postcondiciones: ninguna.

Flujo Normal:

1. Clic en el botón Mis Archivos
2. Seleccionar el archivo con extensión .md5
3. Dar clic en Analizar Archivos.

Flujo alternativo:

- S.1 El usuario selecciona archivo. Regresa al paso 1.
S1.2 El usuario no selecciona el archivo con extensión .md5.

Excepciones:

E1: Si el usuario no selecciona el botón “Analizar Archivos” (elige la opción codificar o decodificar). Regresa al paso 1.

Prioridad: Alta

Frecuencia de uso: Alta

Reglas:

- Clic el botón Analizar Archivos.
Seleccionar archivo con extensión .md5.

4.2. Diagrama de Clases

En este apartado se describe la estructura del sistema propuesto, mostrando las clases del programa, sus atributos y métodos; permitiendo conocer las relaciones que existen entre ellos. La Fig. 4.5 muestra el diagrama general de clases, que tendrá mejor entendimiento con los diagramas de interacción.

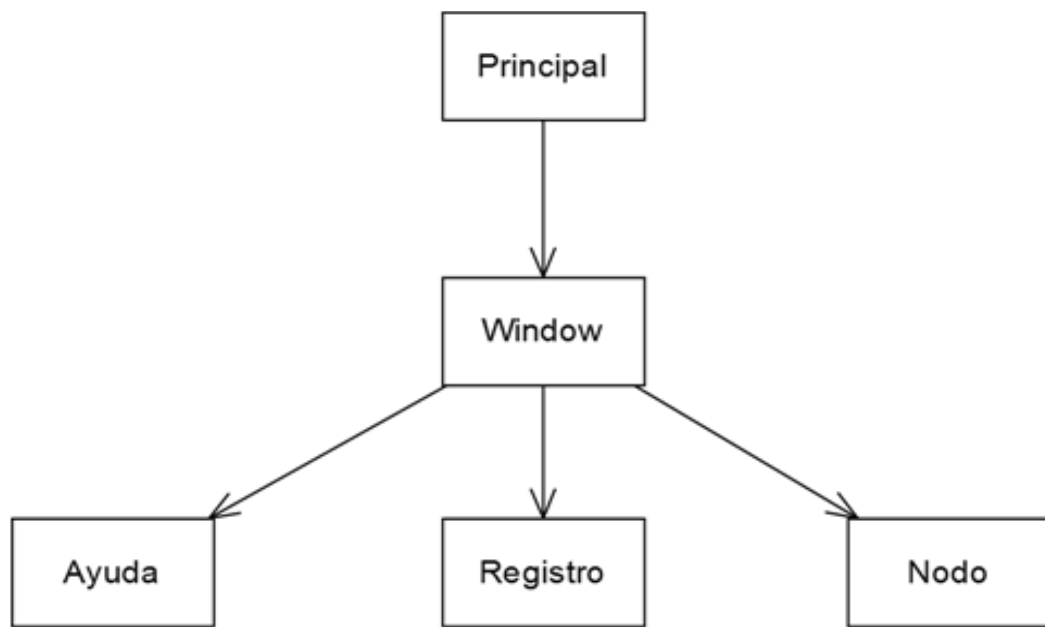


Figura 4.5 Diagrama de clases del sistema propuesto. (Fuente: Propia, 2017).

4.3. Diagramas de Interacción

Las Fig. 4.5 a la 4.10, se muestra el paso de los mensajes entre las clases a través de los diagramas de interacción. Siguiendo el paradigma de orientado a objetos, estos diagramas visualizan el llamado de los métodos entre las instancias generadas para cada clase. Así, se describe la manera en cómo se comunican y se comportan para cada caso de uso visto en el subtema anterior. Permitiendo conocer de manera exacta como el usuario interactúa con el sistema.

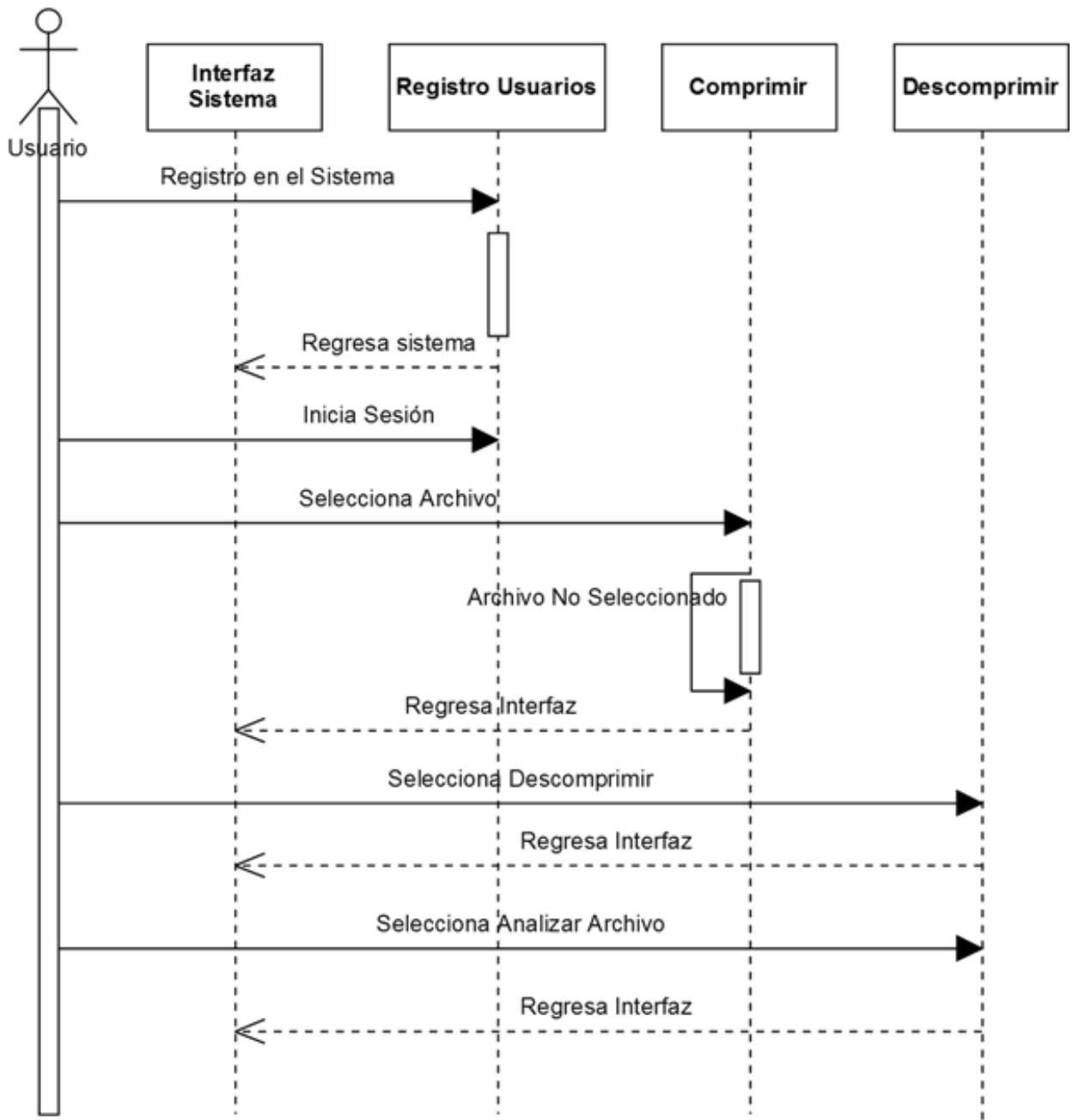


Figura 4.6 Diagrama de interacción General del sistema. (Fuente: Propia, 2017).

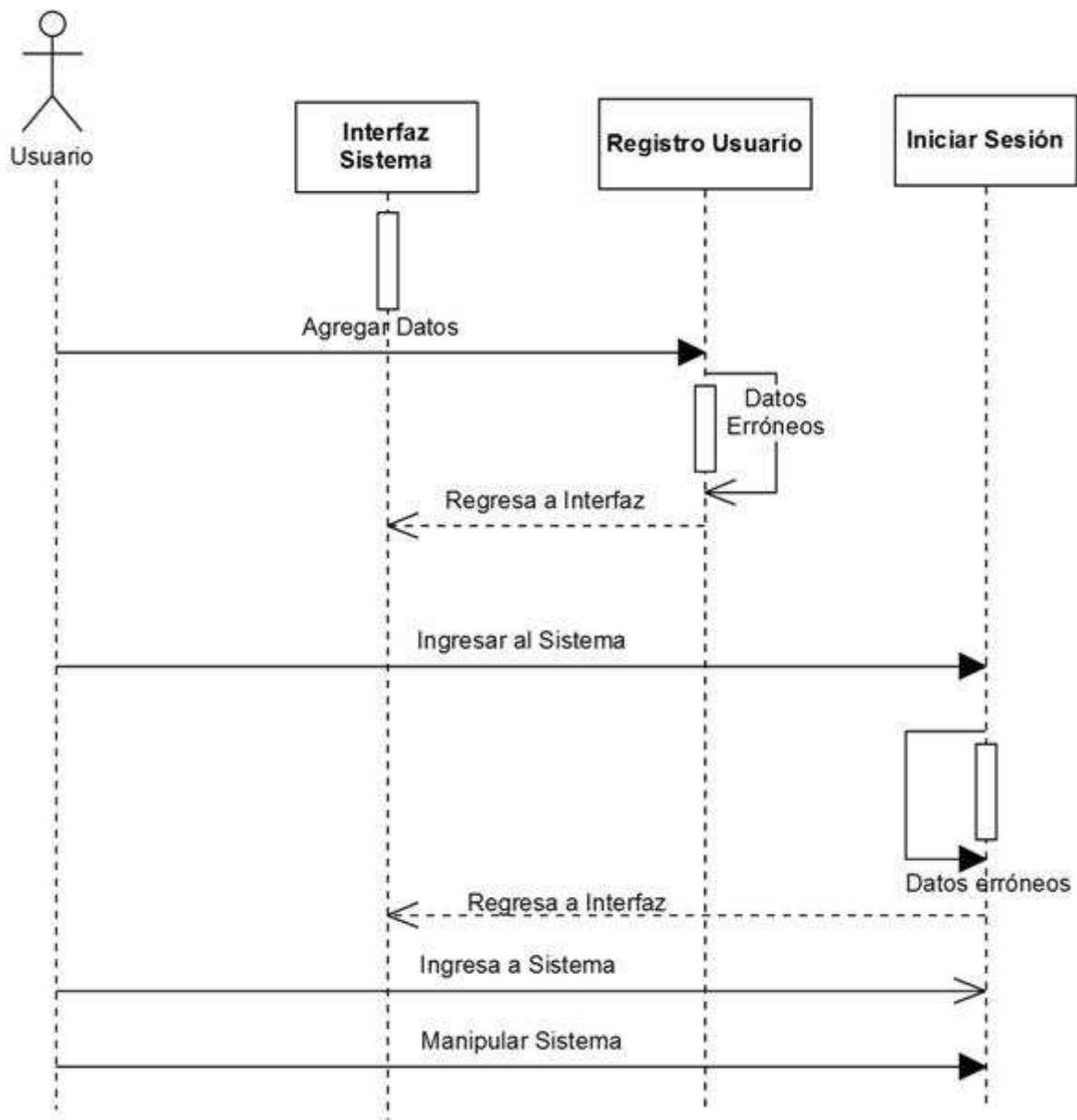


Figura 4.7 Diagrama de interacción del registro de usuario. (Fuente: Propia, 2017).

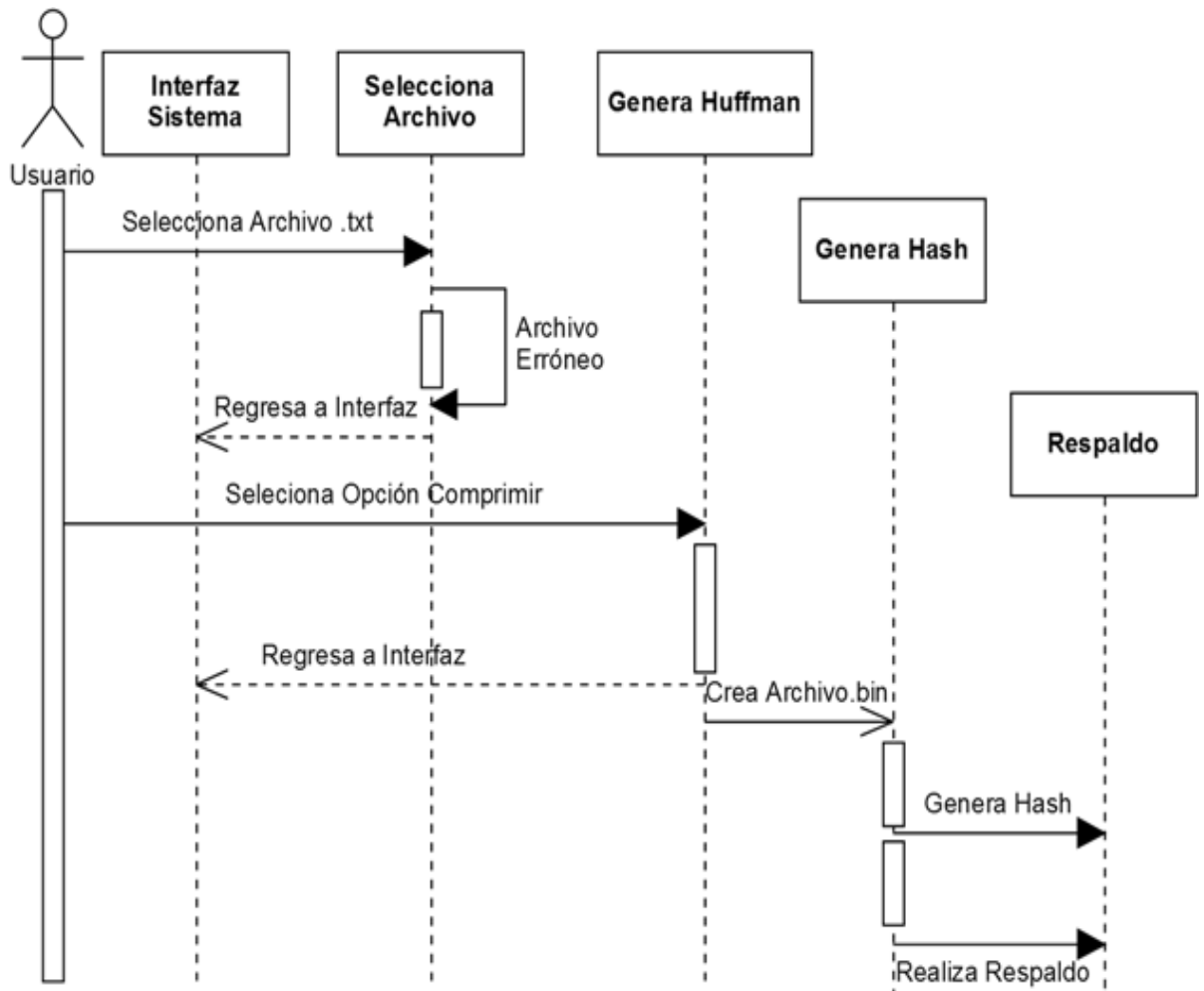


Figura 4.8 Diagrama de interacción de la Compresión. (Fuente: Propia, 2017).

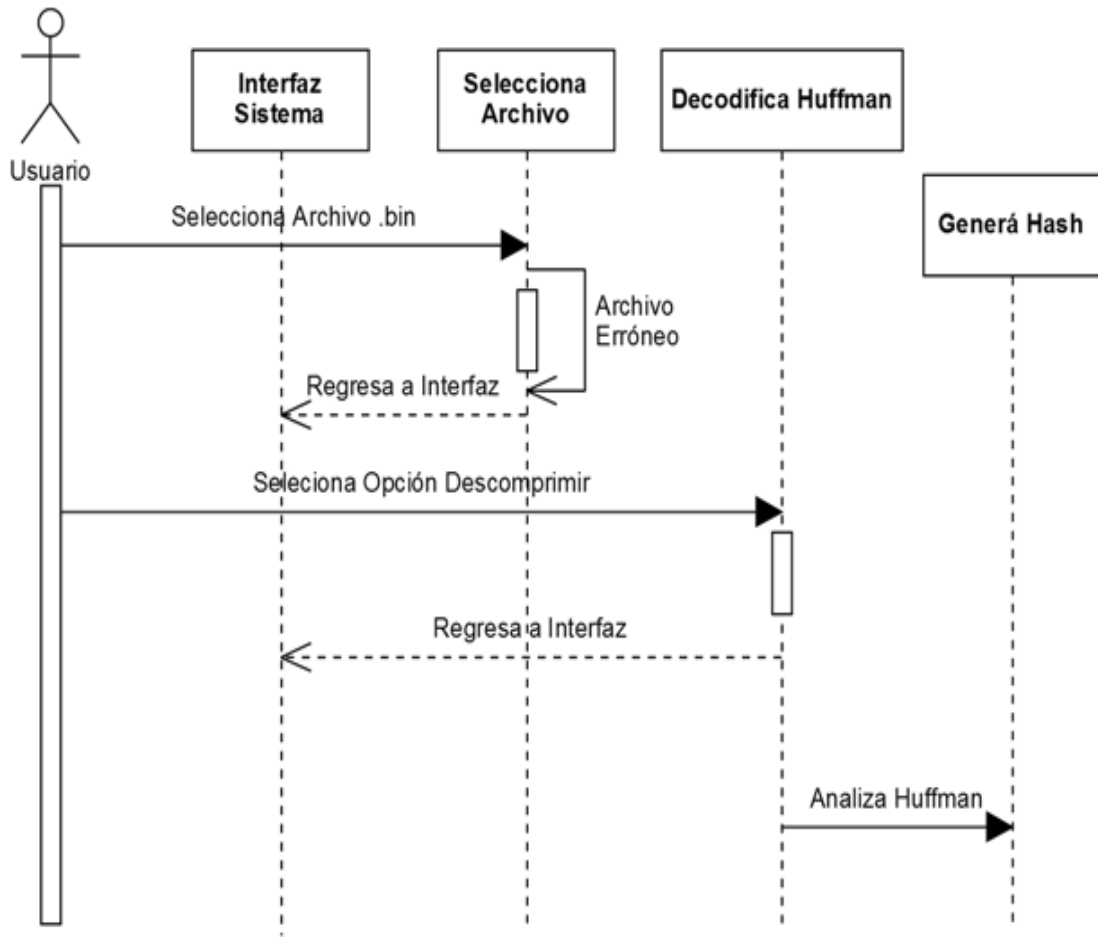


Figura 4.9 Diagrama de interacción de la Descompresión. (Fuente: Propia, 2017).

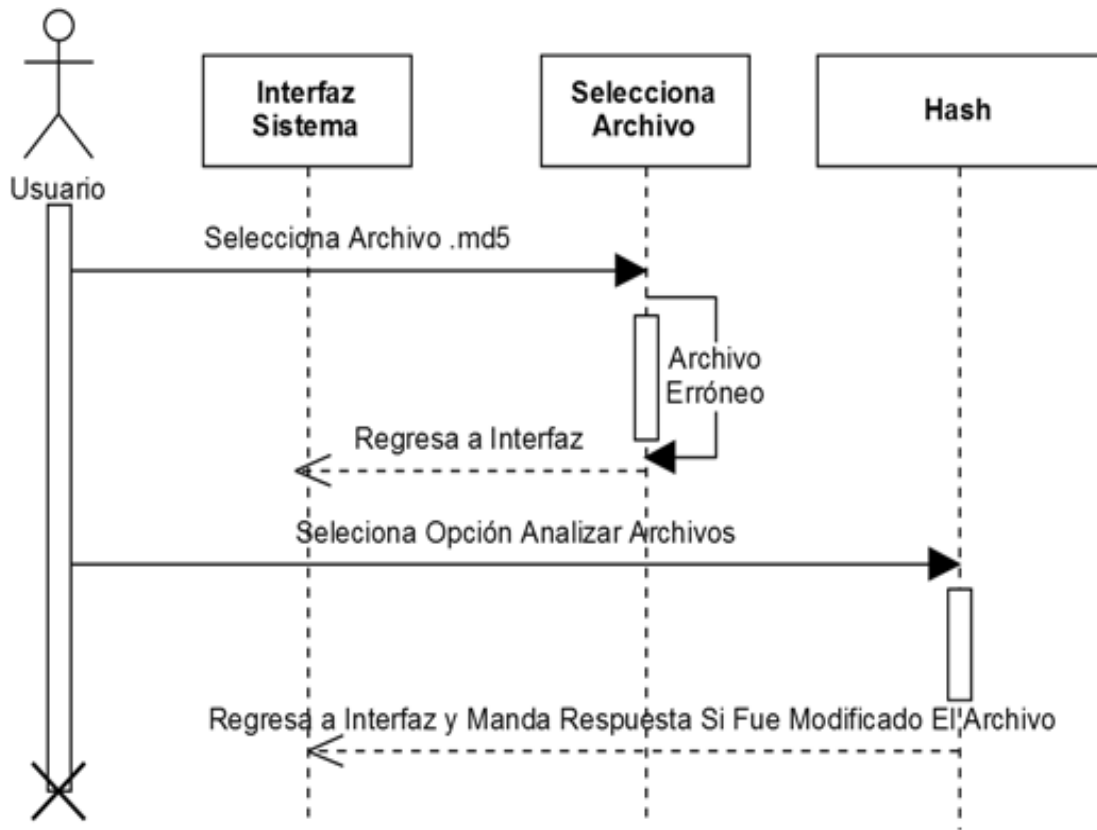


Figura 4.10 Diagrama de interacción del análisis de archivos. (Fuente: Propia, 2017).

4.4. Diagrama de Actividades

Los diagramas de actividades permiten conocer las relaciones y acciones que tiene el usuario al manipular un software. Por ello es importante conocer el correcto uso de este sistema que se está proponiendo. En esta subsección se presentan los diagramas de actividades. La Fig. 4.11 muestra cómo es la interacción del usuario al realizar su registro en el sistema, y como debe iniciar sesión. La Fig. 4.12 presenta el diagrama de actividades general del sistema propuesto y las Fig. 4.13 a 4.15, muestran las actividades para los restantes casos de uso.

Diagrama General de Actividades

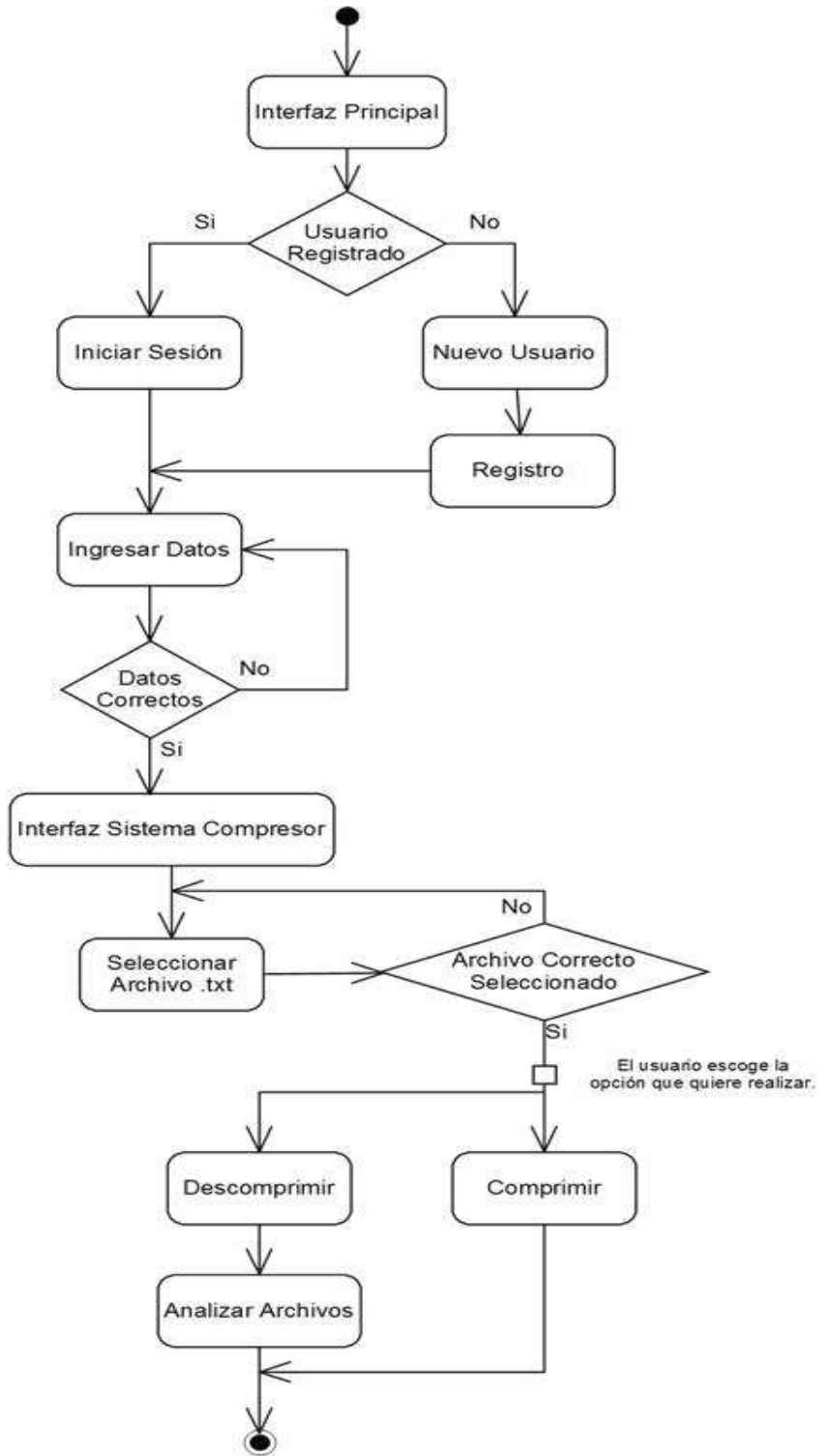


Figura 4.11 Diagrama de Actividades General del sistema propuesto. (Fuente: Propia, 2017).

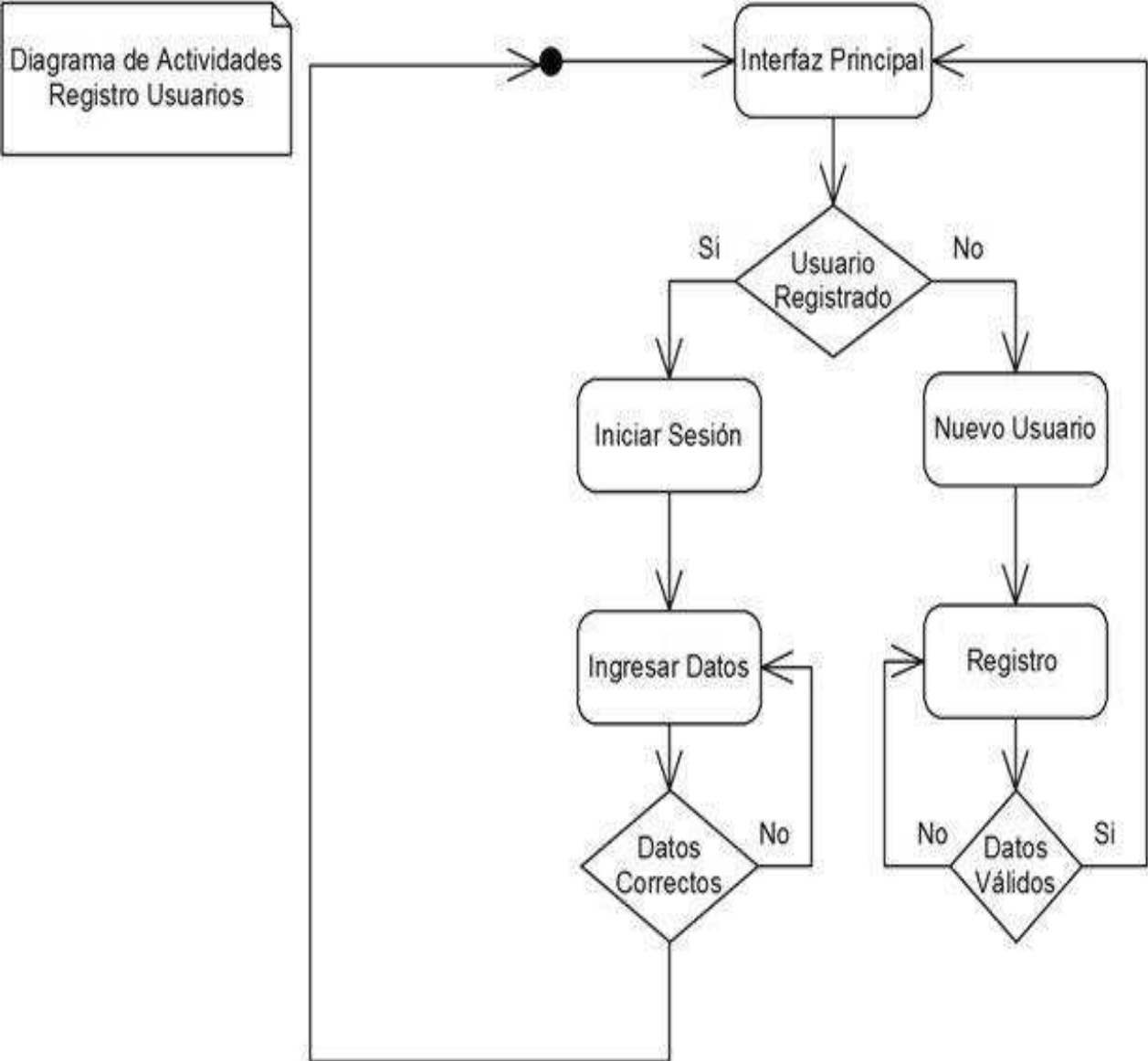


Figura 4.12 Diagrama de Actividades de Registro Usuario. (Fuente: Propia, 2017).

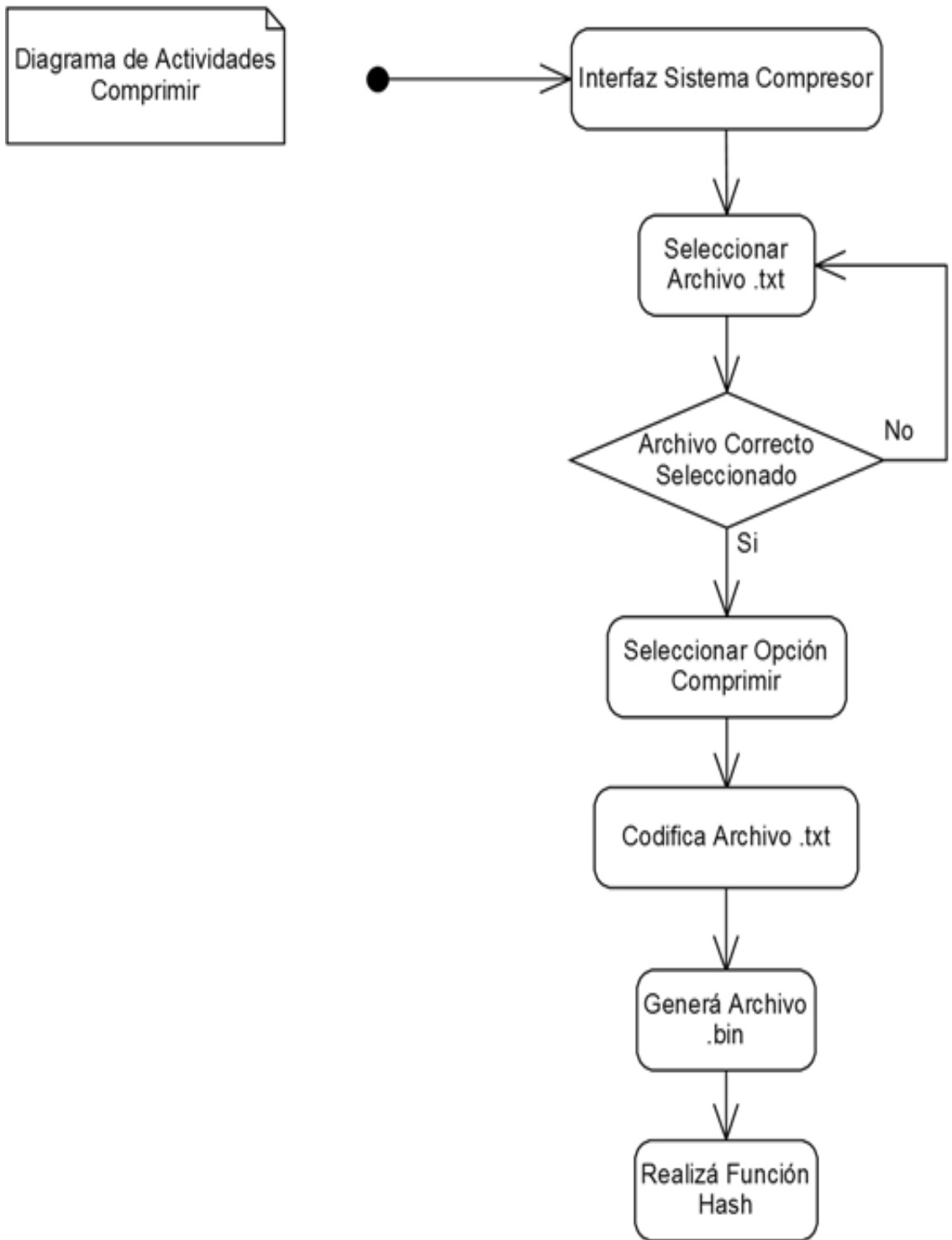


Figura 4.13 Diagrama de Actividades Comprimir. (Fuente: Propia, 2017).

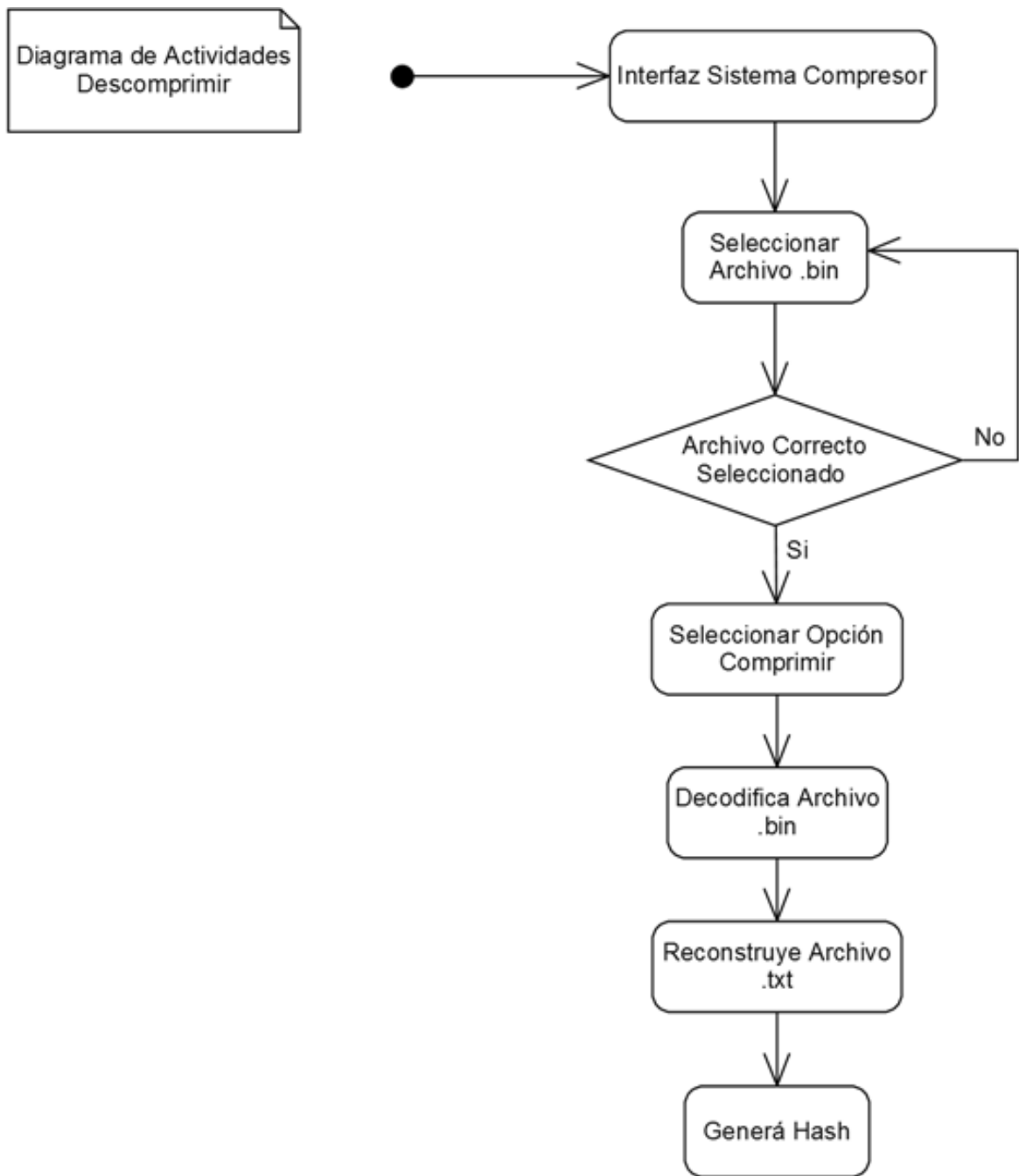


Figura 4.14 Diagrama de Actividades Descomprimir. (Fuente: Propia, 2017).

Diagrama de Actividades
Analizar Archivo

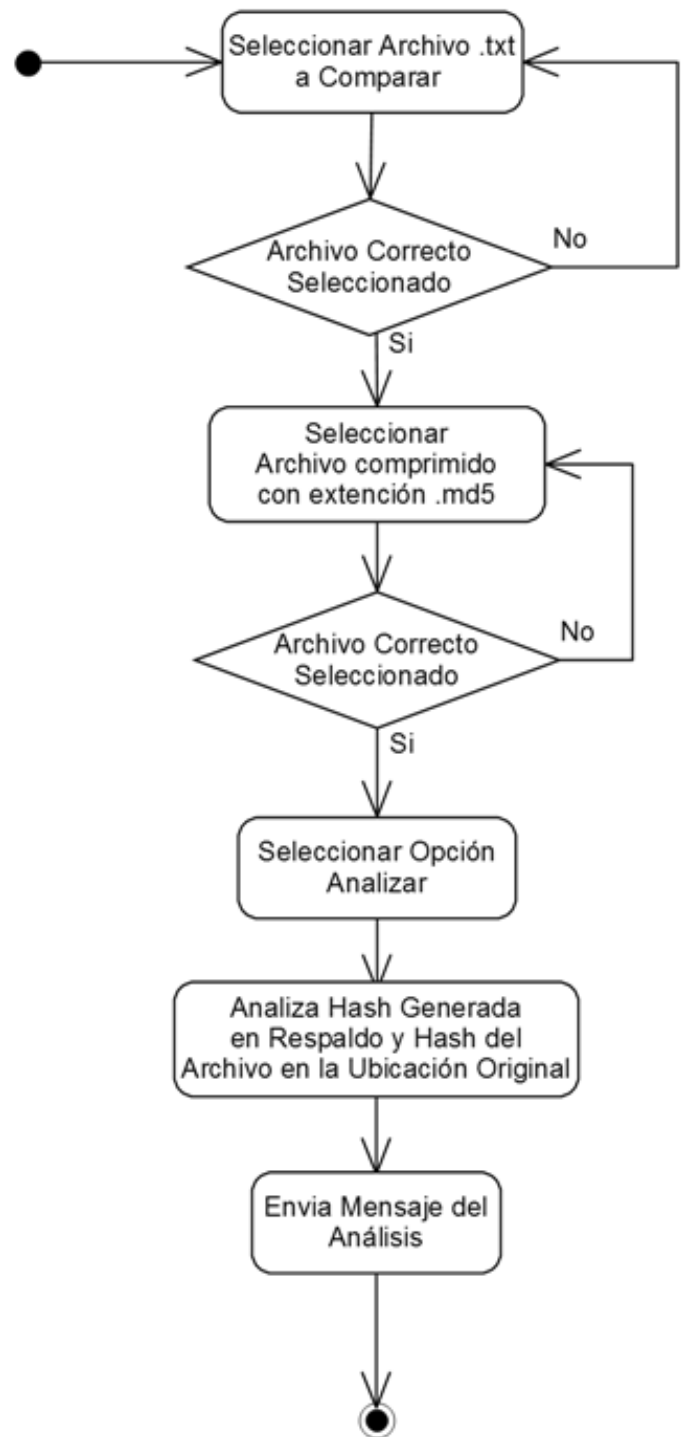


Figura 4.15 Diagrama de Actividades Analizar Archivo. (Fuente: Propia, 2017).

5. IMPLEMENTACIÓN

"La mejor forma de predecir el futuro es implementarlo"
David Heinemeier Hansson

En este capítulo se muestra la interfaz del sistema. La Fig. 5.0 presenta el diagrama por bloques de todas las posibles rutas que a través de la ventana inicial se pueden acceder. Como puede observarse, existen dos posibilidades para iniciar el sistema, que son agregar usuario o iniciar sesión.

La Fig. 5.1 presenta la interfaz principal del sistema, donde el usuario proporcionará la información de inicio de sesión o en su defecto se agregará para generar su información de acceso y su carpeta de respaldo. Las opciones de esta primera interfaz son las siguientes:

- a) Iniciar Sesión: el usuario colocará el nombre con el cual se registró en el sistema, y también su contraseña, para poder tener acceso a la interfaz de compresión.
- b) Registrar: en esta opción el usuario realizará el llenado de un formulario donde se le pedirá un nombre, contraseña y nombre de la carpeta en donde se almacenará el respaldo, la función hash y la codificación, teniendo como objetivo principal que solo él sepa estos datos para su acceso al sistema.

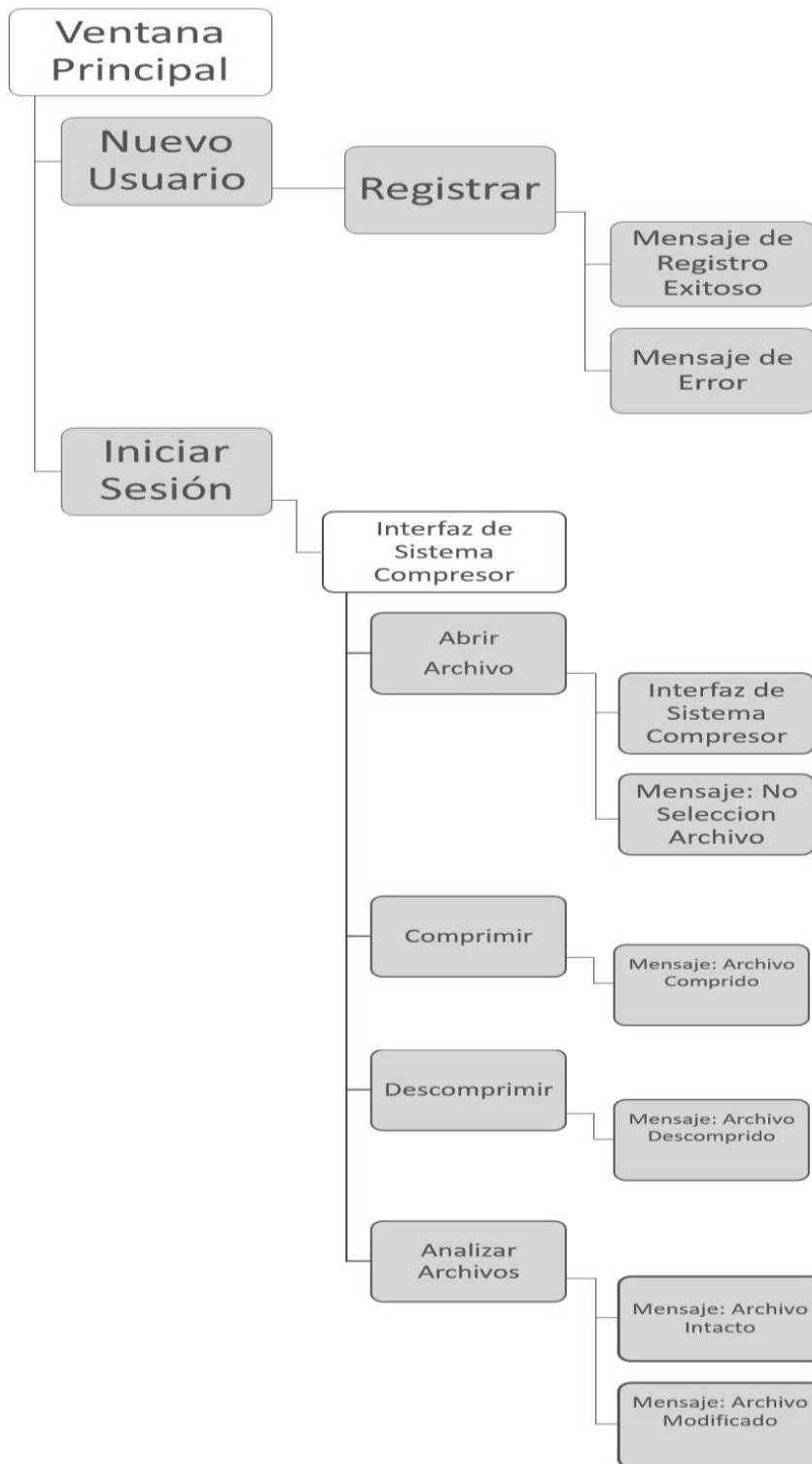


Figura 5.0 Estructura de la interfaz del sistema propuesto. (Fuente: Propia, 2017).



Figura 5.1 Interfaz principal del sistema propuesto. (Fuente: Propia, 2017).



Figura 5.2 Interfaz para el registro del usuario. (Fuente: Propia, 2017).

La interfaz que permitirá el manejo total del sistema tiene por nombre “Sistema Compresor”, donde tendrá el menú para poder realizar la compresión de los documentos .txt, que el usuario desee reducir su tamaño y realizar el respaldo del mismo. La primera interacción que el usuario realiza será escoger el archivo .txt a comprimir como se muestra en la Fig. 5.3.



Figura 5.3 Interfaz de Menú del sistema propuesto. (Fuente: Propia, 2017).

Al seleccionar la opción de selección de archivo se mostrará un panel de navegación, permitiendo que el usuario navegue en su equipo de cómputo para poder seleccionar el documento a comprimir (Fig. 5.4).



Figura 5.4 Interfaz para la selección del archivo a comprimir. (Fuente: Propia, 2017).

Si el usuario no selecciona un archivo de manera intencional o accidentalmente, el sistema enviara un mensaje alertando que no se seleccionó algún archivo. Teniendo como reacción que el sistema envíe al usuario a la interfaz “Sistema Compresor”, y tendrá que realizar nuevamente la selección del archivo.

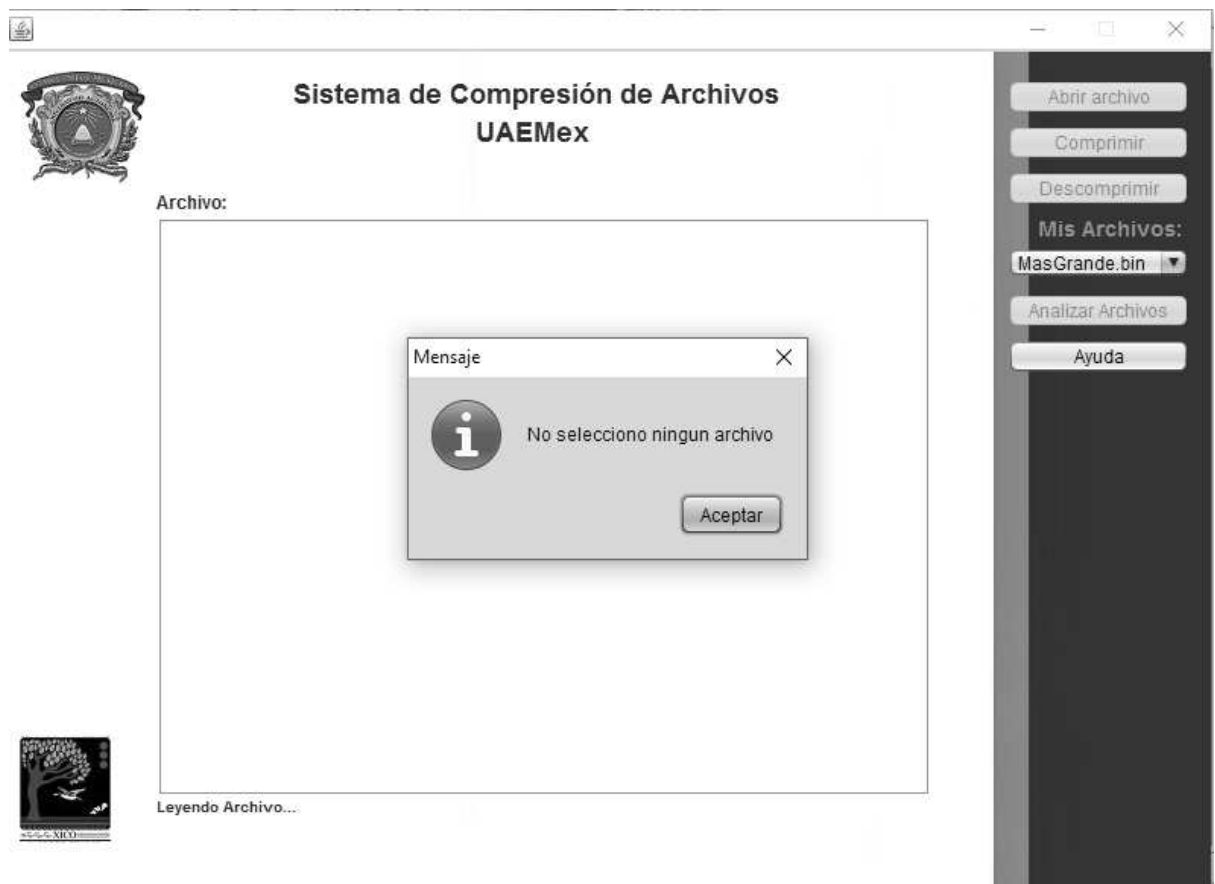


Figura 5.5 Ventana de alerta en caso de que el archivo no haya sido seleccionado.(Fuente: Propia, 2017).

Si el usuario realiza la correcta selección del archivo, el sistema activara el botón de comprimir.

Al seleccionar el botón comprimir, el sistema realiza el respaldo, almacenándolo en la carpeta que el usuario creó. Y el programa enviara un mensaje informando que se realizó la codificación, como se muestra en las Fig. 5.6 y 5.7.

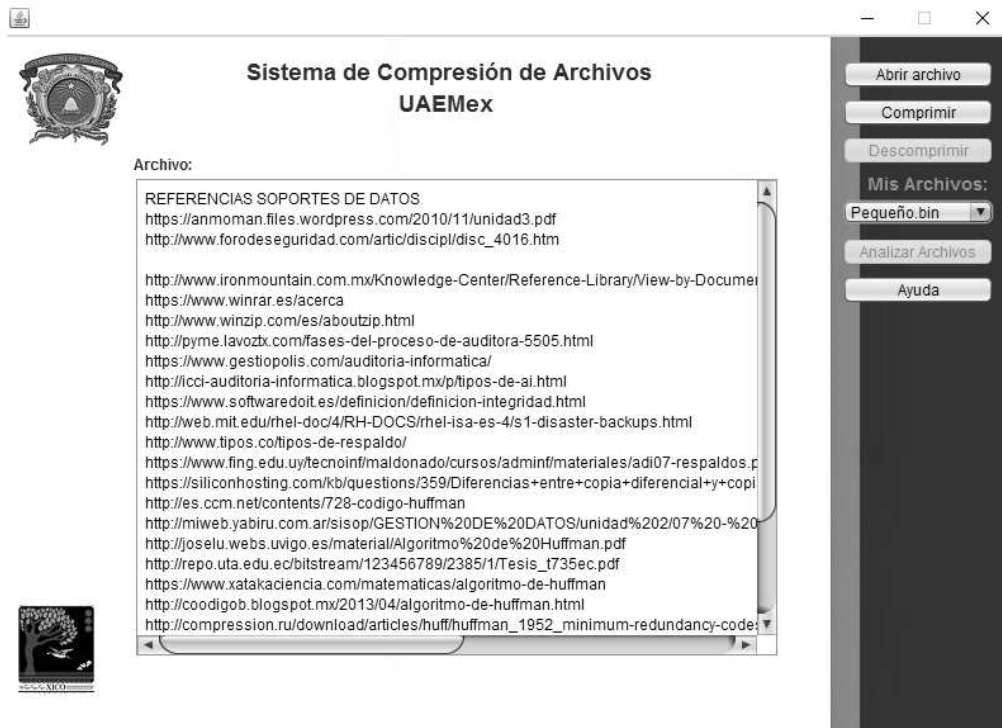


Figura 5.6 Interfaz que visualiza el contenido del archivo a comprimir.

(Fuente: Propia, 2017).

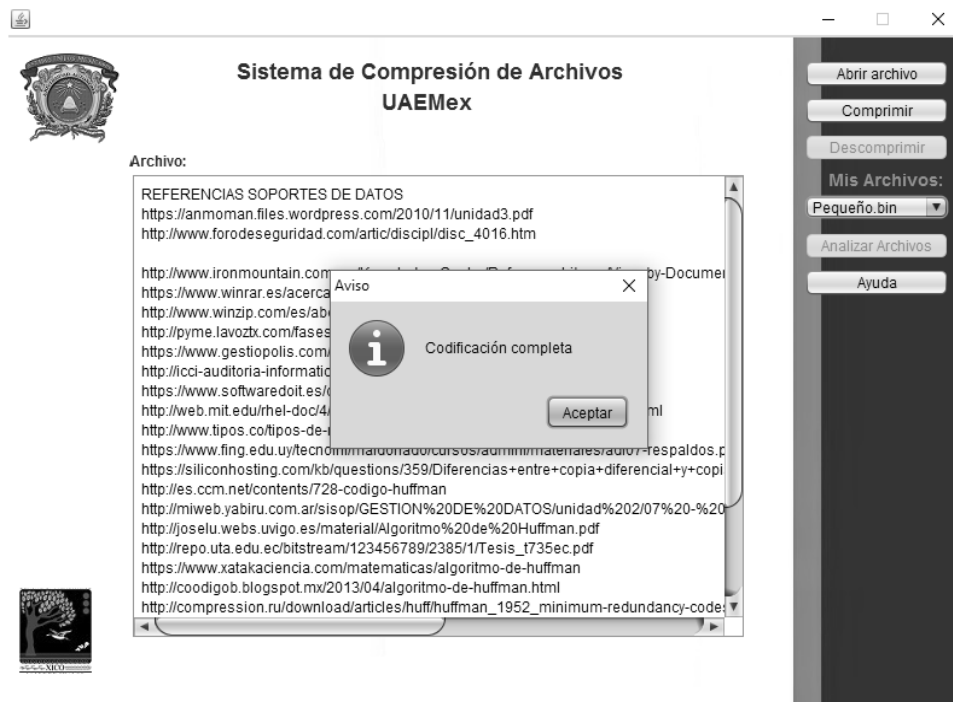


Figura 5.7 Ventana de aviso correspondiente a la codificación completa (Fuente: Propia, 2017)..

Y en el menú, en la sección de “Mis Archivos”, se observará el nuevo archivo creado con la extensión .bin, al igual que el archivo .md5 y el respaldo (Fig. 5.8).

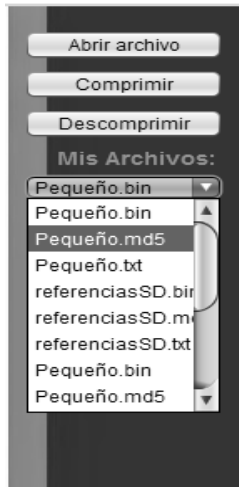


Figura 5.8 Visualización de los archivos que se encuentran comprimidos en la carpeta de respaldo. (Fuente: Propia, 2017).

El usuario seleccionara el archivo con extensión .bin. Y dará click al botón habilitado que será “Descomprimir”. Si el usuario selecciono el archivo correcto se envía un mensaje de que el archivo ha sido descomprimido. La Fig. 5.9 muestra lo comentado.

Para conocer si el archivo ha sido modificado el sistema realiza una comparación entre el hash generada al comprimir, y el documento que se quedó en la ubicación original. El usuario realizara la selección del archivo a analizar, y el archivo que se generó con extensión .md5, paso siguiente dará clic en el botón de “Analizar Archivos” (Fig. 5.10).

El sistema a hará la comparación enviará un mensaje indicando el estado en el que se encuentra el archivo. Como se puede observar en las Figs. 5.11 y 5.12.



Figura 5.9 Ventana de aviso correspondiente a la decodificación completa. (Fuente: Propia, 2017).

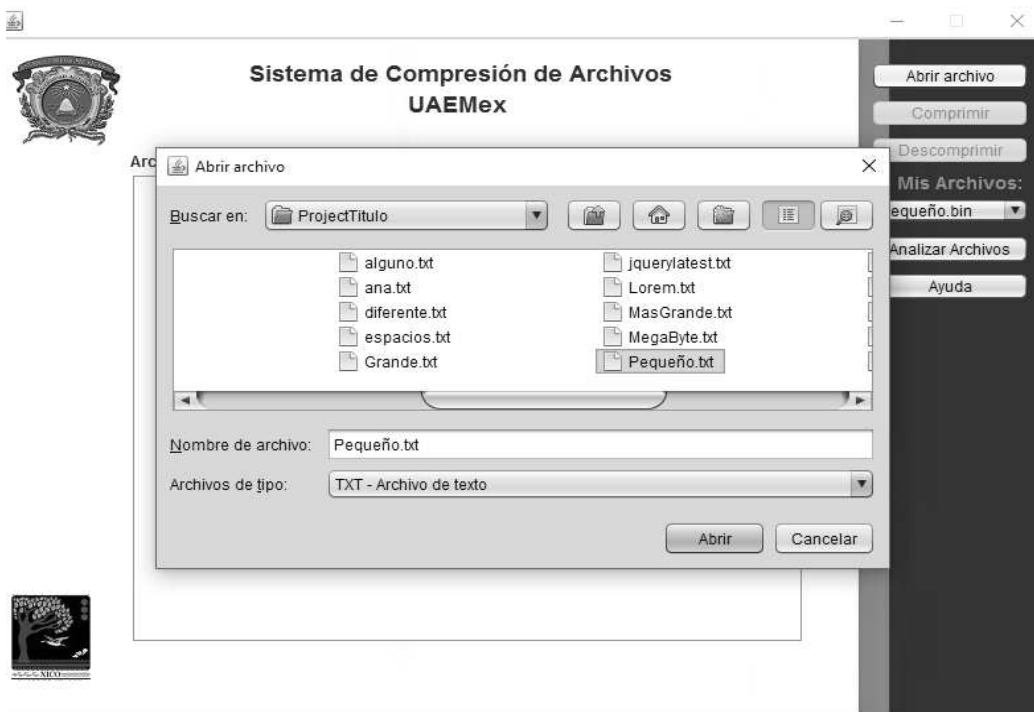


Figura 5.10 Selección de archivo para el análisis de integridad. (Fuente: Propia, 2017).

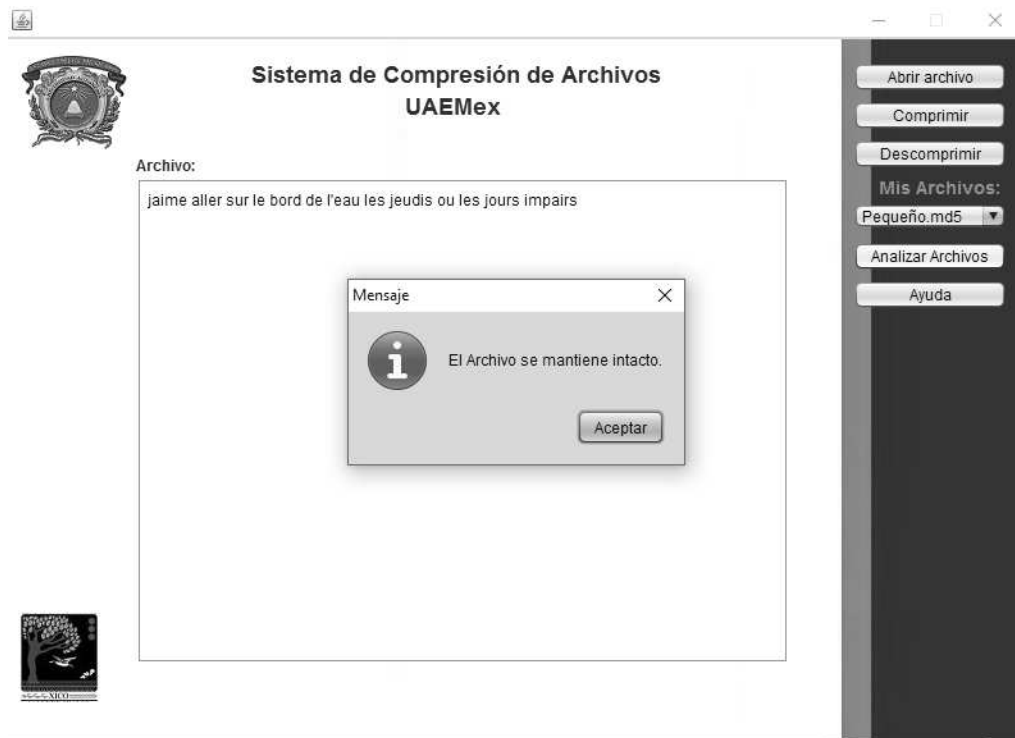


Figura 5.11 Ventana de aviso correspondiente a la integridad intacta del documento.
(Fuente: Propia, 2017).



Figura 5.12 Ventana de aviso correspondiente la modificación del documento.

5.1. Pruebas

En esta sección se pone en ejecución el sistema que se propone, para visualizar la correcta función de este software. En la interfaz principal se selecciona la opción de nuevo usuario. Paso siguiente, se realiza el registro del nuevo usuario que accederá al sistema, colocando un nombre, una contraseña y el nombre de la carpeta en la cual se almacenarán los archivos que comprima. Como se ilustra en las Fig. 5.13 y 5.14. Al dar clic en “Registrar” se creará la carpeta personal en el proyecto.



Nombre de Usuario:

Contraseña:

Nombre de la carpeta:

Figura 5.13 Ventana de registro del Usuario1. (Fuente: Propia, 2017).

Nombre	Fecha de modifica...	Tipo	Tamaño
bdFile.bd	20/06/2017 08:18 ...	Archivo BD	1 KB
Usuario1	20/06/2017 08:12 ...	Carpeta de archivos	
build	31/05/2017 08:19 ...	Carpeta de archivos	
dist	31/05/2017 08:19 ...	Carpeta de archivos	
src	31/05/2017 01:47 ...	Carpeta de archivos	
Test	31/05/2017 12:52 ...	Carpeta de archivos	
bin	09/05/2017 06:30 ...	Carpeta de archivos	
nbproject	09/05/2017 06:30 ...	Carpeta de archivos	

Figura 5.14 Ilustración de la carpeta creada por el Usuario1, denominada Usuario1 (Fuente: Propia, 2017).

El sistema envía un mensaje donde informa que el registro se hizo de manera correcta, y regresara a la interfaz principal, donde se coloca el nombre de usuario y la contraseña recién creados. El sistema responde con un mensaje de bienvenida, como se muestra en la Fig. 5.15.



Figura 5.15 Ventana de aviso correspondiente a la bienvenida en el sistema.
(Fuente: Propia, 2017).

Se dará clic en la opción "Abrir Archivo", y se selecciona el documento al que se le aplicará la compresión. Al término de la compresión, aparecerá un mensaje de éxito en la codificación, para que se active la descompresión y el análisis de integridad.

Para garantizar el buen funcionamiento del sistema, se realizaron 30 pruebas de compresión, descompresión y análisis de integridad. La Tabla 5.1 presenta los tiempos de ejecución con distintos tamaños de archivos de texto.

Como puede observarse, los resultados aumentan según el tamaño del archivo, debido a que se realiza la protección de la información a través de la función picadillo. Es importante mencionar que los tiempos obtenidos pueden ser mejorados, optimizando la implementación del código Huffman y de la función picadillo.

Tabla 5–1. Tiempos de Ejecucion de Diferentes Tamaños de Archivos en el Sistema Propuesto.

(Fuente: Propia, 2017).

Archivo	Tamaño	Tiempo Ejecución
1	2.8k	01:15 min
2	7.9k	01:27 min
3	15k	01:55 min
4	42k	02:24 min
5	80k	03:02 min
6	85k	03:05 min
7	120k	03:25 min
8	145 k	03:38 min
9	350 k	03:41 min
10	500k	04:02 min
11	1 Mb	05:01 min
12	1.5 Mb	05:25 min
13	3.3 Mb	06:40 min
14	4.2 Mb	07:33 min
15	4.8 Mb	07:52 min
16	5 Mb	07:59 min
17	5.6 Mb	08:06 min
18	6.2 Mb	08:18 min
19	7 Mb	08:57 min
20	8.2 Mb	09:06 min

21	9 Mb	09:35 min
22	10 Mb	10:05 min
23	12.4 Mb	11:07 min
24	15 Mb	13:05 min
25	16.7 Mb	13:55 min
26	18 Mb	15:02 min
27	20 Mb	16:08 min
28	26 Mb	18:58 min
29	26.8Mb	19:10 min
30	30 Mb	21:58 min

6. CONCLUSIONES Y ALCANCES A FUTURO

*“Solo podemos ver poco del futuro, pero lo suficiente para darnos cuenta que hay mucho que hacer”
Anónimo*

A continuación, se presentan las conclusiones y el trabajo futuro de este proyecto de tesis.

6.1. Conclusiones

En el presente trabajo de tesis tuvo como finalidad conocer sobre la importancia en la seguridad e integridad de la información almacenada en los dispositivos tecnológicos, ya que hoy en día se cometen actos maliciosos arriesgando la estabilidad de los soportes de datos.

Al conocer la problemática del tema, se decidió hacer una investigación sobre el análisis forense, el cual permitió definir que es un proceso, el cual es sumamente importante trabajar con copias sobre la información almacenada en un dispositivo, para que este salvaguardada.

La integridad y seguridad es el objetivo principal al desarrollar el sistema propuesto, ya que al tener un archivo en una computadora cualquier persona lo puede modificar para un mal uso de este, es por ello que el software diseñado realiza un respaldo, donde el usuario elegirá la carpeta en donde será almacenada esta copia, y podrá realizar la codificación de la misma manteniéndola segura, y someterla a un análisis para conocer si ha sido alterado ese archivo.

Se estima que con el programa desarrollado brinda la seguridad necesaria para que los respaldos que realiza se mantengan íntegros. Al realizar pruebas se pudo observar que es eficaz, y hace un análisis fiable en los archivos.

La codificación de la información consiste en aplicar el algoritmo Huffman que su objetivo principal es realizar árboles binarios de acuerdo a la frecuencia del texto que contiene un archivo, obteniendo un código que represente ese carácter.

En el análisis para conocer la integridad de los archivos fue utilizada la función Hash ya que, al tener una entrada de información, la transforma en un valor numérico y es más sencillo su estudio ya que es más fácil analizar dos series de números que carácter por carácter, al implementar esta función, se pudo observar que es eficaz y seguro en su resultado.

La auditoría sobresalió en esta investigación e implementación del sistema propuesto ya que, para realizar un respaldo, codificación, decodificación, recuperación y análisis, estudia de manera precisa y exacta la información que se está protegiendo, y si nota alguna deficiencia en algún proceso de los ya mencionados, mandará un mensaje alertando al usuario de que hay alguna anomalía en el archivo.

6.2. Alcances a Futuro

Al realizar el diseño del sistema, se comenzó a pensar en muchas mejoras que permitirá que el sistema propuesto tenga un mejor rendimiento en su uso.

- Variedad de tipos de archivos,
- Mejores técnicas de seguridad en los archivos,
- Mejoría en tiempos de ejecución,
- Implementar más funciones que puedan hacer esta herramienta más eficaz entre estas características se encuentra la recuperación de datos de hasta un 90% de información en un soporte de datos.

Estos son solo algunos objetivos a futuro, ya que es un trabajo que se puede prestar para mejorar y sea un software que se implemente en distintas áreas.

REFERENCIAS

- [1] Cegarra, J. (2014). Metodología de la Investigación Científica y tecnológica. España: Díaz Santos.
- [2] Cordova, C. (2010). Desarrollo de una Herramienta Educativa, para la Compresión y Reducción de Imágenes en Escala Grises, usando Transformada de Wavelet (Licenciatura). Escuela Superior Politécnica del Litoral.
- [3] Diagramas de secuencia UML: Referencia. (2017). Msdn.microsoft.com. Retrieved 10 Jun 2017, from <https://msdn.microsoft.com/eses/library/dd409377.aspx>
- [4] Fowler, M., & Scott, K. (1999). UML gota a gota. Bogotá: Pearson Education.
- [5] Función hash criptográfica. (2017). Es.wikipedia.org. Retrieved 16 June 2017, from https://es.wikipedia.org/wiki/Funci%C3%B3n_hash_criptogr%C3%A1fica
- [6] Hall, A. and Devoto, M. (2000). La Firma digital: herramienta habilitante del comercio electrónico. 1st ed., Argentina: La Ley.
- [7] Hamud, A. (2010). El Funcionamiento De Los Discos Duros, Memorias USB y Como Trabajan Las Herramientas de Informática Forense en la Recuperación de Datos en Estos Dispositivos (Licenciatura). Universidad Nacional Autónoma de México.
- [8] Hankerson, R., D. (2003). Introduction to Information Theory and Data Compression (2nd ed.). Chapman.

- [9] Hernández Sampieri Roberto, "Metodología de la investigación", Editorial McGraw-Hill, 2003, tercera edición. México DF
- [10] Hernández, E. (1993). Auditoría Informática (Un Enfoque Metodológico) (Maestría). Universidad Autónoma de Nuevo León.
- [11] Hernández, I. (2014). Buenas Prácticas para la Implementación de la Seguridad en un Centro de Cómputo (licenciatura). Universidad Nacional Autónoma de México.
- [12] IBM Knowledge Center. (2017). Ibm.com. Retrieved 10 Jun2017, from https://www.ibm.com/support/knowledgecenter/es/SS6RBX_11.4.3/com.ibm.sa.tutorial.doc/topics/Less2.1_Bu
- [13] Landeros, R., & Gonzáles, M. (2016). Estadística con SPSS y Metodología de la investigación. México: Trillas.
- [14] Lázaro, F. (2015). Introducción a la Informática Forense. Madrid: RA-MA.
- [15] López, C., & Veiga, M. (2002). Teoría de la Información y Codificación. Biblioteca Uvigo. Retrieved 14 May 2017, from <http://www.investigacion.biblioteca.uvigo.es/xmlui/bitstream/handle/11093/188/mybook.pdf?sequence=1>
- [16] Medina, J. (2009). Funciones Hash Criptográficas (Licenciatura). Universidad Nacional Autónoma de México. México: Limusa.
- [17] Morales, G. (2010). Introducción a la Teoría de Códigos. Retrieved 15 April 2017, from <http://delta.cs.cinvestav.mx/~gmorales/TeoriaDeCodigos/Incd.pdf>
- Morán, G., & Alvarado, D.(2010). Métodos de Investigación. México: Pearson.

- [18] On-Line Services 2000, S. (2017). Soporte WinRAR - Que es WinRAR. Winrar.es. Retrieved 11 feb 2017, from <https://www.winrar.es/soporte/articulo/30>
- [19] PUD (Proceso Unificado de Desarrollo) - Business-Control. (2017). Sites.google.com. Retrieved 3 Feb 2017, from <https://sites.google.com/site/businesscontrolesi/productossoftware/metodologia-del-trabajo/pud>
- [20] Red Hat Enterprise Linux 4. (2004). Web.mit.edu. Retrieved 7 May 2017, from <http://web.mit.edu/rhel-doc/4/RH-DOCS/rhel-isa-es-4/index.html>
- [21] Respaldos de información (backups). (2017). ingen.unam.mx. Retrieved 4 Jun 2017, from <http://www.iingen.unam.mx/es-mx/Publicaciones/GacetaElectronica/GacetaMayo2014/Paginas/Respaldosdeinformacion.aspx>
- [22] Salomon, D. (2002) Data A Guide to Data Compression Methods. Springer. USA.
- [23] Sánchez, V. (2010). Criptoanálisis de la Grafía Informática (Licenciatura). Universidad Autónoma del Estado de México.
- [24] Stinson, Douglas R. "Cryptography - Theory and Practice", Discrete Mathematics and its Applications. Chapman & Hall / CRC Press, Boca Raton~FL, 3rd edition
- [25] Sayood, K. (2006) Introduction to Data Compression. Elsevier. USA.
- [26] Tamayo, M. (2014). El proceso de la investigación Científica. (5th ed.),
- [27] Un manual sobre compresión de datos. (2012). Ibm.com. Retrieved 8 May 2017, from <https://www.ibm.com/developerworks/ssa/library/lcompr/index.html>

[28] WinZip® - Información Básica. (2017). Winzip.com. Retrieved 11 feb 2017,
from <http://www.winzip.com/win/es/aboutzip.html>