Universidad de las Ciencias Informáticas

Facultad 10

SISTEMA OFIMÁTICO ENCRIPTADO

Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas.

Autores

Caridad Menéndez Samé

Heileen Martínez Ortega

Tutor

Ing. Daniel Hernández Garrigó

Co-Tutor

Ing. Yordan Vélez Rodríguez

Ciudad Habana, julio, 2010

"Año 52 de la Revolución"

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.
Para que así conste firmamos la presente a los ____ días del mes de _____ del año _____

Caridad Menéndez Samé

_____

Firma del Autor

Heileen Martínez Ortega

_____

Firma del Autor

Daniel Hernández Garrigó

_____

Firma del Tutor

Yordan Vélez Rodríguez

_____

Firma del Co-Tutor

DATOS DE CONTACTO

Ing. Daniel Hernández Garrigó.

Graduado en el 2009 de Ingeniero en Ciencias Informáticas en la Universidad de las Ciencias Informáticas (UCI). Especialista general del bloque 8 de la Infraestructura Productiva de la UCID. Actualmente desempeña las funciones de Jefe de línea de Seguridad Informática.

e-mail: dgarrigo@uci.cu

Ing. Yordan Vélez Rodríguez.

Graduado en el 2008 de Ingeniero en Ciencias Informáticas en la Universidad de las Ciencias Informáticas (UCI). Se ha desempeñado como jefe de Línea de Gestión de Redes y Seguridad Informática así como Jefe de Centro de Fortalecimiento a la Seguridad Informática, donde fue tutor de las tesis "Aplicación para la administración centralizada de servicios y servidores" y "Entorno seguro de desarrollo de software para la UCID". Actualmente se desempeña como jefe de Centro de Datos y ejerce funciones de administrador de redes.

e-mail: yvelez@uci.cu

RESUMEN

La seguridad de la información es un elemento de mucha importancia para todas las instituciones de Las Fuerzas Armadas Revolucionarias (FAR), ya que cuentan con información confidencial que debe ser debidamente protegida, ya sea contenida en documentos impresos o en formato digital. Actualmente las FAR llevan a cabo un proceso de informatización para mejorar sus actividades y fortalecer su seguridad, ejemplo de ello es la migración que realizan hacia el sistema operativo GNU/Linux. Diariamente los oficiales y trabajadores civiles de las entidades militares realizan actividades que quedan registradas en documentos en formato digital, elaborados con la herramienta OpenOffice.org de dicho sistema operativo. A pesar de estar diferenciados según la clasificación de la información contenida, no todos los documentos son archivados en las computadoras con la seguridad requerida. Una vez guardados, el contenido queda en texto plano, siendo esto una amenaza a la confidencialidad, integridad y disponibilidad de los mismos. Teniendo en cuenta esta problemática, el objetivo fundamental del presente trabajo es añadir a las funcionalidades básicas Guardar y Abrir de los documentos elaborados con el Procesador de textos y la Hoja de cálculos, de la herramienta OpenOffice.org, las técnicas y métodos de encriptación necesarios para cifrar los objetos de tipo texto, permitiendo así que la información esté siempre encriptada, y por tanto, protegida.

INTRODUCCIÓN

La escritura es un medio utilizado por el hombre para comunicar a sus semejantes aquello que piensa y transmitir conocimientos. Permite plasmar inquietudes y experiencias dejando constancia de lo que se dice y hace en un proceso determinado. Actualmente es utilizada en todas las esferas de la sociedad para documentar los adelantos científicos, tecnológicos, culturales y sociales.

El conjunto organizado de datos, recogidos a través de la escritura, es llamado información. Esta puede existir y ser transmitida en diferentes formas, impresa o utilizando medios digitales. Cualquiera sea la forma adquirida es un recurso que posee un gran valor, por tal motivo debe ser debidamente protegido. Dada la velocidad en que los individuos y organizaciones se apropian de más tecnologías para todas las actividades humanas, esta protección se ve comprometida, puesto que los problemas de seguridad y confiabilidad son cada vez mayores al igual que la proliferación de fuentes y técnicas invasivas.

A partir de esta situación el concepto de seguridad informática adquiere un papel relevante para mantener un control estricto en el acceso a la información. El punto de partida para ello es fomentar la confianza y salvaguardar la seguridad, en un mundo cada vez más interconectado por redes. Por tal motivo es necesario que las tecnologías se adapten a las nuevas necesidades y se transformen en herramientas útiles para la innovación económica y social.

Cuba apuesta hoy por el desarrollo de la Informática enfrentando un nuevo reto: el logro de la seguridad y protección de los recursos informativos (1). Para apoyar este objetivo todos los ministerios, empresas e instituciones, deben confeccionar y aplicar medidas que reduzcan el riesgo de afectaciones a sus recursos.

Las Fuerzas Armadas Revolucionarias (FAR), como parte indisoluble del pueblo, se insertan en este proceso de informatización, automatizando sus actividades y creando planes de contingencia. Integrada con la Universidad de Ciencias Informáticas (UCI), producen sistemas para perfeccionar su seguridad, compatibilizando lo que está hecho, integrando soluciones y desarrollando productos propios para la defensa del país. Dentro de este proceso se encuentra la migración que hacen hacia el sistema operativo GNU/Linux[1].

Una de las actividades informáticas principales llevadas a cabo por los oficiales y trabajadores civiles de las FAR es el trabajo con documentos en formato digital, elaborados con la herramienta OpenOffice.org[2] de dicho sistema operativo, en especial el Procesador de textos y la Hoja de cálculos. A pesar de estar

diferenciados según la clasificación de la información contenida, no todos los documentos son archivados en las computadoras con la seguridad requerida. Una vez guardados, el contenido queda en texto plano, siendo esto una amenaza a la confidencialidad, integridad y disponibilidad de los mismos. Aunque las FAR cuenten con sus mecanismos de seguridad algún usuario puede violarlos de manera fortuita o provocada, poniendo en riesgo esta información. Esta **situación problemática** lleva a la siguiente interrogante: ¿Cómo añadir al Procesador de textos y la Hoja de cálculos, las funcionalidades de guardar y abrir documentos con la seguridad requerida?

Para dar solución al planteamiento anterior queda definido como **objeto de estudio** el proceso de encriptación de datos digitales, enmarcando el **campo de acción** en el proceso de encriptar objetos de tipo texto, elaborados en el Procesador de textos y la Hoja de cálculos, de la herramienta OpenOffice.org.

Este trabajo tiene como **objetivo general** añadir a las funcionalidades básicas Guardar y Abrir del Procesador de textos y la Hoja de cálculos, las técnicas y métodos de encriptación necesarios para que el texto de los documentos sea guardado con la seguridad requerida por los oficiales y trabajadores civiles de las FAR.

Para ello se dará cumplimiento a los siguientes **objetivos específicos:**

- Definir el marco teórico que sustenta el estudio del tema en cuestión.
- Identificar soluciones existentes sobre dichas funcionalidades.
- Modelar la solución propuesta.
- Implementar y validar la solución.

La **Idea a defender** queda definida de la siguiente manera**:** Si se añaden las técnicas y métodos de encriptación a las funcionalidades básicas Guardar y Abrir, del Procesador de textos y la Hoja de cálculos, se evitará el riesgo de comprometer la confidencialidad, integridad y disponibilidad del texto de los documentos.

El presente trabajo de diploma está conformado por 3 capítulos:

Capítulo 1. Fundamentación teórica.

En este capítulo se realiza un estudio sobre el uso de los documentos digitales y la seguridad de los mismos. Se analizan aspectos relacionados con la criptografía y el uso de las macros[3] del OpenOffice.org. Se selecciona la metodología, herramienta de desarrollo y lenguaje de programación y de modelado

necesarios para el desarrollo de una nueva propuesta, la cual persigue garantizar la seguridad de los documentos utilizados por los oficiales y trabajadores civiles de las FAR.

Capítulo 2. Características de la aplicación.

En este capítulo se describe el flujo actual del proceso utilizado en las oficinas de las FAR para guardar los documentos. Se proponen nuevas funcionalidades para que sean guardados con la seguridad requerida y se dan a conocer las personas relacionadas con la aplicación. Se realizan las actividades referentes a los flujos de trabajo de Modelación de Negocio, Especificación de los Requisitos de Software y Diseño, generándose los artefactos correspondientes.

Capítulo 3. Implementación y Prueba.

En este capítulo se realizan las actividades de implementación a partir de los resultados obtenidos en el diseño y se describe el código de las funcionalidades propuestas. Concluido este flujo se realizan los casos de pruebas para garantizar que los programas cumplan con las especificaciones requeridas y que sean eliminados los posibles defectos.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

En este capítulo se realiza un estudio sobre las tendencias actuales del trabajo con documentos digitales y la seguridad de los mismos. Se analizan los conceptos fundamentales de la criptografía así como algunos de los métodos criptográficos más extendidos mundialmente.

Se selecciona la metodología, herramienta de desarrollo y lenguaje de programación y de modelado, necesarios para el desarrollo de este trabajo de diploma. También se abordan los métodos de seguridad aplicados en las FAR y se propone una solución que garantizará mayor seguridad de los documentos utilizados por los oficiales y trabajadores civiles de las FAR.

1.2 Trabajo con documentos

A diario se genera en las organizaciones una gran cantidad de documentos, ya sea en papel o formato electrónico. Estos contienen aspectos referentes a personas o temáticas determinadas. Conforman la base histórica de una organización. Ayudan a tomar decisiones apropiadas para planificar el futuro y lograr más eficiencia, productividad y coherencia. Reducen los riesgos relacionados con la falta de pruebas pues sirven como evidencia de las actividades realizadas. Apoyan los requisitos legales, las regulaciones vigentes y protegen los intereses de la organización, así como los de su personal. Ninguna institución podría sobrevivir sin documentar sus actividades, por lo que el trabajo ofimático ha cobrado gran relevancia.

1.2.1 Documentación digital

La masiva implantación de herramientas informáticas ha ocasionado que la mayoría de los documentos se elaboren de manera electrónica y sean almacenados y procesados en formato digital. Esta digitalización constituyó un avance trascendente que dio origen a los documentos electrónicos. Estos son un contenedor de información en formato digital. Tienen igual importancia que los tradicionales y presentan características que los diferencian de estos, como son:

- Se puede cambiar el contenido de una línea, de un párrafo o una página, sin que por ello halla que cambiar el documento entero.
- Permite su almacenamiento en grandes cantidades en objetos de tamaño reducido.

- Se conservan los originales, evitando su desgaste, deterioro o extravío.
- Se evita la pérdida de tiempo profesional en la búsqueda manual de datos, pues poseen distintos tipos de extracción de información como los índices de búsqueda, entre otros, que permiten obtener la información al instante.
- Permite acceso *on-line* a los documentos desde cualquier lugar del mundo a través de *Internet*.
- Se elimina la necesidad de distribuir múltiples copias de un mismo documento.

1.2.1.1 Seguridad de la documentación digital

La seguridad de los documentos se define como la habilidad para proteger la información y garantizar la confidencialidad, integridad y disponibilidad de la misma. Los sistemas que procesen, almacenen o transmitan información deben preservarla frente a alteraciones, de modo que el contenido no sea modificado por personas no autorizadas, deben evitar que sea conocida por personal ajeno a la institución y garantizar que esté disponible cuando sea necesario. Si alguna de estas características no existe el documento dejaría de ser seguro.

Mantener la seguridad es un proceso continuo, en el cual se conocen siempre los agentes capaces de explotar los fallos de seguridad (vulnerabilidades) y las amenazas que pudiesen afectar una información, así como la probabilidad de que ocurran y el impacto que puede tener.

Las vulnerabilidades pueden ser de varios tipos:

- Físicas: ambiente en el que se almacena o maneja la información.
- Hardware: defectos de fabricación, desactualización, mantenimiento inadecuado.
- Naturales: condiciones de la naturaleza que pueden provocar riesgo.
- Humanas: daños que las personas pueden causar a la información (*hackers*, virus, empleados descontentos).
- *Software*: aplicaciones que permiten accesos indebidos.
- Almacenamiento: soportes físicos utilizados para almacenar información.
- Comunicación: fallos en la transmisión de la información.

Para evitar o eliminar las vulnerabilidades se llevan a cabo un conjunto de medidas que pueden ser **preventivas** (evitan los puntos débiles), **perceptivas** (encuentran actos que supongan un riesgo) y

correctivas (corrección de problemas cuando ocurren). Es decir, se desarrollan políticas y planes que establecen los estándares de seguridad que deberán cumplir todos los usuarios.

1.2.2 Métodos de seguridad aplicados en las FAR

En las FAR existen políticas de seguridad informática como herramientas organizacionales para concientizar a los trabajadores sobre la importancia y sensibilidad de la información que ellos manejan. Estas determinan cómo debe relacionarse el personal con los recursos y servicios informáticos existentes en la entidad. La seguridad que presentan los ordenadores de los trabajadores es la que proveen los sistemas operativos en sí, además de permisos establecidos según la jerarquía de los oficiales. Por su parte, *Windows* combina técnicas que aseguran la protección ante accesos no deseados, permite proteger la red, el sistema operativo y los datos a través de la autentificación de acceso, seguridad a nivel de objeto y derechos de usuarios, proceso realizado únicamente por los administradores del sistema. Por otro lado, existen los ordenadores que utilizan GNU/Linux los cuales pueden estar expuestos a ataques, a pesar de ser un sistema robusto, flexible y potente. Por tanto, se puede decir que en todo momento los documentos son guardados en texto claro que puede ser leído o modificado por intrusos.

1.3 Fundamentos Teóricos de la Criptografía

1.3.1 Conceptos básicos

Texto plano: es el texto del documento original, que se quiere proteger mediante el uso de técnicas criptográficas. El conjunto de todos estos textos es denotado como “M”.

Criptografía (2):
La Real Academia Española (RAE) define criptografía (del griego: oculto+ escritura) como: "el arte de escribir con clave secreta o de modo enigmático". Esta definición es interesante y llamativa pero resulta poco ajustada para los tiempos actuales. Las imprecisiones que presenta son:

- Arte: hace años dejó de ser un arte para convertirse en una ciencia. Es un conglomerado de técnicas que tratan de ocultar cierta información frente a observadores no autorizados.
- Escritura de documentos: los mensajes además de escribirse pueden ser enviados o guardados en un computador. Hoy en día se aplica a diversos tipos de documentos y formatos.

- Se supone una clave: los sistemas actuales usan una o dos, incluso pueden entrar en juego cuatro claves.
- Clave secreta: aunque existen sistemas de clave secreta que usan una sola clave, existen además sistemas de clave pública que usan dos: una privada y otra pública.
- Representación enigmática: la representación binaria de la información podría ser enigmática para los humanos pero no para los computadores debido a que ese es su lenguaje natural.

Por tanto, una definición más técnica sería: "Rama inicial de las Matemáticas y en la actualidad también de la Informática y la Telemática, que hace uso de métodos y técnicas con el objeto principal de cifrar, y Por tanto, proteger un mensaje o archivo por medio de un algoritmo, usando una o más claves" (2).

Criptograma: es el texto transformado mediante alguna técnica criptográfica. Este texto resulta ilegible a no ser que se conozca la clave para recuperarlo. El conjunto de todos estos textos es denotado como "C".

Cifrar: es la técnica que protege o autentica a un documento o usuario al aplicar un algoritmo criptográfico, del cual se debe conocer una clave específica o secreta para poder descifrarlo o recuperarlo. La RAE define cifrar como "Transcribir en guarismos[4], letras o símbolos, de acuerdo con una clave, un mensaje cuyo contenido se quiere ocultar (2)".

Clave: son las llaves privadas o públicas que permiten cifrar un documento y descifrar el correspondiente criptograma. El conjunto de todas las claves utilizadas para encriptar un mensaje es denotado como "K".

Criptosistema: se define un criptosistema como una quíntupla (M;C;K;E;D), donde:
M es el conjunto de todos los mensajes sin cifrar, es decir, el texto plano.
C representa el conjunto de todos los posibles mensajes cifrados o criptogramas.
K representa el conjunto de claves que se pueden emplear en el criptosistema.
E es el conjunto de transformaciones de cifrado o familia de funciones que se aplica a cada elemento de M para obtener un elemento de C.
D es el conjunto de transformaciones de descifrado, análogo a E.
Todo criptosistema ha de cumplir la condición de que si tenemos un mensaje m, y es cifrado empleando la clave k y luego descifrado empleando la misma clave, se obtendrá de nuevo el mensaje original m.

$$D_k(E_k(m)) = m$$

Los criptosistemas se clasifican en dos grupos generales:

Criptosistemas simétricos o de clave privada: Son aquellos que emplean la misma clave k tanto para cifrar como para descifrar y es compartida por el emisor y el receptor, por lo que la seguridad reside en mantener dicha clave en secreto.

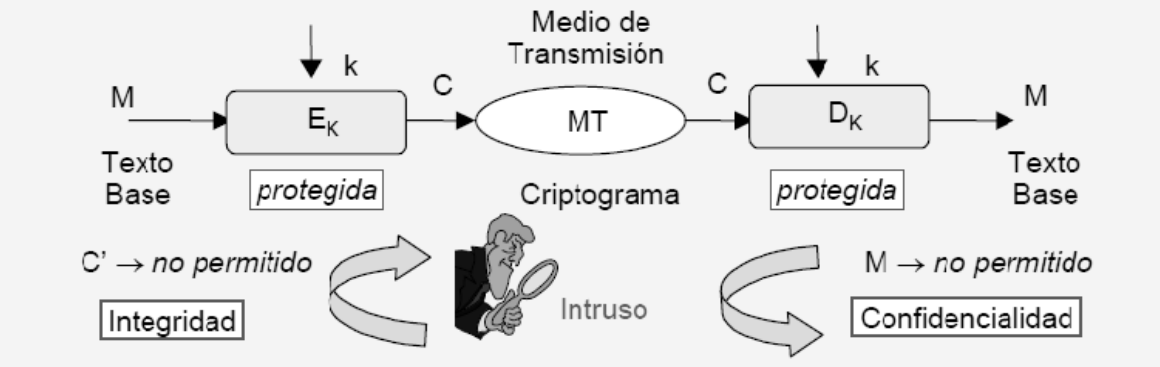

Figura 1. Cifrado con clave privada

Criptosistemas asimétricos o de clave pública: Son aquellos que emplean una doble clave, una privada y otra pública. Una de ellas sirve para la transformación E de cifrado y la otra para la transformación D de descifrado, es decir, lo que se cifra en emisión con una clave, se descifra en recepción con la clave inversa. En muchos casos son intercambiables, esto es, si empleamos una para cifrar la otra sirve para descifrar y viceversa. Para garantizar la seguridad los criptosistemas deben cumplir que a través de la clave pública no se pueda obtener la privada.

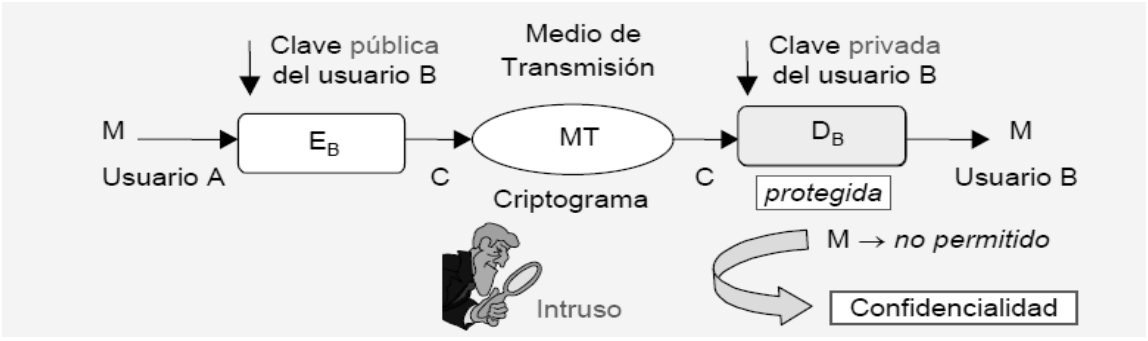

Figura 2. Cifrado con clave pública del receptor.

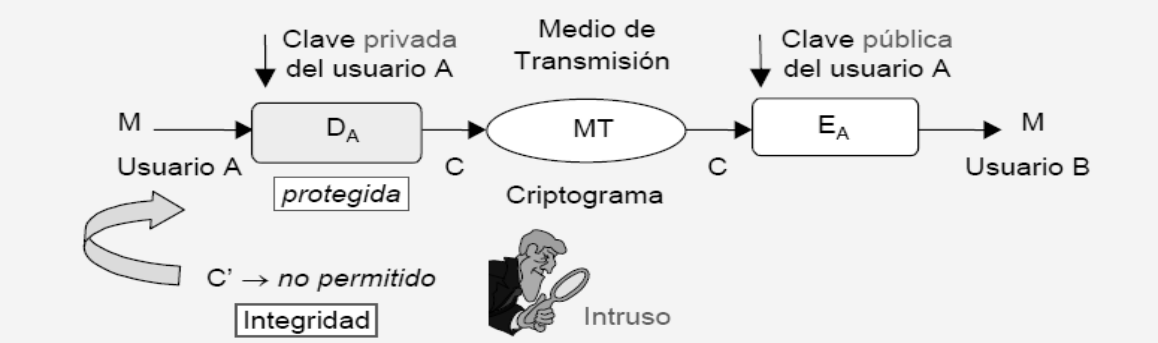

Figura 3. Cifrado con clave privada del emisor.

1.3.2 Algoritmos Criptográficos

DES

Sus siglas están dadas por su nombre en inglés (Data Encryption Standard), lo cual significa en español, Estándar de Encriptación de Datos. Es un algoritmo de cifrado de datos desarrollado originalmente por la compañía IBM (International Business Machines) y posteriormente modificado y adoptado por los Estados Unidos de América (EE.UU) en 1977, como estándar de cifrado de todas las informaciones sensibles no clasificadas. Realiza combinaciones, sustituciones y permutaciones entre el texto a cifrar y la clave, permitiendo que las operaciones puedan realizarse en ambas direcciones. Se trata de un sistema de cifrado simétrico por bloques de 64 *bits*, de los que 8 *bits* (un *byte*) se utilizan como control de paridad (para la verificación de la integridad de la clave). Cada uno de los *bits* de la clave de paridad (1 cada 8 *bits*) se utiliza para controlar uno de los *bytes* de la clave por paridad impar, es decir, que cada uno de los *bits* de paridad se ajusta para que tenga un número impar de "1" dentro del *byte* al que pertenece. Por lo tanto, la clave tiene una longitud "útil" de 56 *bits*, es decir, realmente sólo se utilizan 56 *bits* en el algoritmo (3).

Aunque se demostró que era viable un ataque por fuerza bruta, debido a la escasa longitud que emplea en su clave, no presenta ninguna debilidad grave desde el punto de vista teórico, por lo que se ha convertido en el algoritmo simétrico más extendido mundialmente.

IDEA

Este algoritmo data de 1992. Sus siglas están dadas por su nombre en inglés (International Data Encryption Algorithm), lo cual significa en español, Algoritmo Internacional de Cifrado de Datos. Es más joven que DES y al igual que este usa el mismo algoritmo tanto para cifrar como para descifrar. Opera con bloques de 64 *bits* usando una clave de 128 *bits*. Es un algoritmo seguro que ha mostrado ser resistente ante multitudes de ataques. Para muchos constituye el mejor y más seguro de los simétricos disponibles en la actualidad. Su fortaleza se basa en que dada la longitud de su clave es imposible en la práctica atacar mediante la fuerza bruta, ya que sería necesario probar 10^{38} claves. Además, su seguridad deriva del intercalado de operaciones de distintos grupos que son algebraicamente incompatibles como la adición, la multiplicación modular y el O-exclusivo (XOR[5]) *bit* a *bit*.

Rijndael (AES)

En octubre de 2000 fue adoptado por el Instituto Nacional de Normas y Tecnología de los EE.UU el algoritmo Rijndael, acrónimo derivado de los nombres de sus dos autores; los belgas Joan Daemen y Vincent Rijmen. Fue empleado en aplicaciones criptográficas no militares por ser un algoritmo de cifrado potente, eficiente, y fácil de implementar. Para referirse a él también se emplea la denominación AES (Advanced Encryption Standard) como Estándar Avanzado de Cifrado.

AES es un sistema de cifrado por bloques, que soporta diferentes tamaños de bloque y clave, utiliza una de las tres fortalezas de clave de cifrado: una clave de encriptación (contraseña) de 128, 192, o 256 *bits*. Cada tamaño de la clave de cifrado hace que el algoritmo se comporte ligeramente diferente, por lo que el aumento de tamaño de clave no sólo ofrece un mayor número de *bits* con el que se pueden cifrar los datos, sino que también aumenta la complejidad del algoritmo de cifrado (4).

El proceso de selección, revisión y estudio de este algoritmo se efectúa de forma pública y abierta por toda la comunidad criptográfica mundial, lo cual lo convierte en un algoritmo perfectamente digno de la confianza de todos. Se ha comprobado que es resistente al criptoanálisis y se considera como uno de los más seguros en la actualidad. AES es rápido tanto en *software* como en *hardware* y requiere poca memoria. Como nuevo estándar de cifrado, se está utilizando actualmente a gran escala.

RSA

RSA es el algoritmo más sencillo de comprender e implementar de todos los algoritmos asimétricos. Debe su nombre a sus tres inventores: Ronald Rivest, Adi Shamir y Leonard Adleman, y no fue comercial su uso hasta el 20 de septiembre de 2000. Sus claves sirven indistintamente tanto para codificar como para autentificar.

Como en todo sistema de clave pública, cada usuario posee dos claves de cifrado: una pública y otra privada. Cuando se quiere enviar un mensaje, el emisor busca la clave pública del receptor, cifra su mensaje con esa clave, y una vez que el mensaje cifrado llega al receptor, este se ocupa de descifrarlo usando su clave privada. Los mensajes enviados se representan mediante números, y el funcionamiento se basa en el producto conocido de dos números primos grandes elegidos al azar y mantenidos en secreto. Por tanto, su dificultad consiste en factorizar grandes números, por lo que el atacante si quiere recuperar un texto claro a partir del criptograma y la clave pública debe enfrentarse a un problema de factorización. Aunque existen algunos puntos débiles en la forma de utilizarlo que pueden ser aprovechados por los atacantes, se le tiene como uno de los algoritmos asimétricos más seguros.

1.3.3 Herramientas de encriptación

Las herramientas de encriptación posibilitan a los usuarios mantener la seguridad y privacidad de sus documentos confidenciales. Existen diferentes versiones tanto para *Windows* como para GNU/Linux.

Steganos LockNote

Es un editor de textos, similar al Bloc de Notas de *Windows*, capaz de cifrar y guardar con contraseña los documentos. El programa utiliza la tecnología de cifrado AES de 256 *bits* más recientes. Su principal característica es que además de protegerlo con contraseña encripta el contenido del fichero, convirtiéndolo en un .exe muy especial, y prácticamente inexpugnable. En caso de que se viole la seguridad que proporciona la contraseña, el contenido seguirá encriptado.

CryptoForge

Es un programa de encriptación creado para *Windows*. Es usado en el campo personal y profesional para proteger la privacidad de archivos, carpetas y mensajes, utilizando hasta cuatro algoritmos de encriptación robustos como Blowfish[6] de 448 *bits*, Rijndael (AES) de 256 *bits*, Triple DES[7] de 168 *bits*, y Gost[8] de 256

bits. Una vez encriptada la información puede ser almacenada en un medio inseguro, o transmitida por *Internet*, y aún así permanecer secreta. Luego, puede ser descifrada a su formato original. Es fácil de usar debido a que está diseñado para ocultar la complejidad de la tecnología de encriptación y su instalación incluye los idiomas español e inglés. La versión más actual es la 3.3.0 que salió a la luz el 21 de noviembre de 2009.

Codenigma

Es una utilidad diseñada para *Windows* que permite difundir mensajes de texto sin que nadie pueda entenderlos a no ser que conozca la clave de comprensión. El programa permite escoger entre cuatro claves ya predefinidas o crear, guardar y recuperar tantas claves propias como se desee. Los mensajes de texto pueden ser creados manualmente o pegados desde cualquier otra aplicación y solo podrán acceder a ellos aquellos que conozcan la clave con la que se ha encriptado. La última versión que data del 16 de enero de 2007 es CODENIGMA 3.0. Permite la introducción de texto por teclado o copiarlos y pegarlos de algún lugar. Presenta una nueva estética más funcional y más potencial de encriptación (hasta 50.000 veces más difícil de decodificar). Proporciona gran seguridad porque aunque se conozca la clave, el texto no se podrá decodificar si no se tiene el *software*.

PDFTK

Es una herramienta libre que permite trabajar con la extensión pdf. Puede unir varios documentos en uno solo, separar uno en varios, encriptarlos, desencriptarlos y crear nuevos documentos sin cifrar a partir de PDFs cifrados, siempre que se disponga de la clave. Es posible comprimirlos, descomprimirlos y hasta reparar los que sean dañados. Funciona en una gran variedad de plataformas por lo que es una herramienta útil para todas las personas que trabajen con documentos PDF.

Seahorse

Es una interfaz gráfica de GnuPG[9] (GNU Privacy Guard) que se integra en el escritorio GNOME[10]. Es una herramienta para realizar comunicaciones seguras y almacenamiento de datos seguros. Se pueden crear fácilmente cifrados de datos y firmas digitales mediante la interfaz gráfica. Las operaciones de gestión de claves se pueden realizar mediante una interfaz intuitiva. Además, permite gestionar los datos del portapapeles y posee un agente para el almacenamiento de contraseñas privadas, así como un gestor de claves GnuPG y OpenSSH[11].

GPG

GNU Privacy Guard es una alternativa libre al programa de encriptado PGP[12] (Pretty Good Privacy). Se maneja con un sistema de claves públicas y privadas. Esta última se genera mediante una serie de datos como correo electrónico, nombre y apellidos. Solo se deben recordar los mismos y el programa genera ambas claves. La pública se distribuye libremente para que las personas que la posean puedan escribir mensajes o enviar archivos encriptados que el receptor podrá descifrar con su clave privada.

1.4 Análisis de las metodologías de desarrollo de software

Todo desarrollo de *software* es riesgoso y difícil de controlar, por lo que es necesario regirse por una metodología para no obtener clientes insatisfechos con el resultado y desarrolladores aún más insatisfechos. Las metodologías de desarrollo de *software* son un conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a los desarrolladores a realizar un nuevo *software* (5). Indican cómo hay que obtener los distintos productos parciales y finales durante todo el ciclo de desarrollo.

No hay metodología que funcione de manera universal. Debido a que existen numerosas propuestas metodológicas es necesario ajustarlas según la organización y tipo de proyecto. Dentro de estas se encuentran las tradicionales o clásicas y las ágiles. Las primeras se caracterizan por establecer rigurosamente las actividades involucradas, los artefactos que se deben producir, las herramientas y notaciones que se usarán. Hacen mayor énfasis en la planificación y control del proyecto, en la especificación precisa de requisitos y modelado, mientras que las segundas se dirigen a equipos de desarrollo pequeños y dan mayor valor al individuo, a la colaboración con el cliente y al desarrollo incremental del *software* con iteraciones muy cortas.

1.4.1 Metodologías Tradicionales

Las Metodologías Tradicionales o Pesadas, llamadas también Metodologías Clásicas, están guiadas por una fuerte planificación y control del proyecto durante todo el proceso de desarrollo. Estas metodologías imponen una disciplina de trabajo que consiste en realizar una intensa etapa de análisis y diseño antes de la construcción de un sistema. Definen rigurosamente los roles, actividades, artefactos, herramientas y notaciones para el modelado y una documentación detallada. Son eficientes en proyectos de gran

envergadura, pero no resultan eficientes en proyectos donde el entorno del sistema es muy cambiante y se exige reducir drásticamente los tiempos de desarrollo.

1.4.2 Metodologías Ágiles

En proyectos donde el entorno del sistema es muy cambiante, y se exige reducir drásticamente los tiempos de desarrollo pero manteniendo una alta calidad, es conveniente utilizar metodologías ágiles. Estas constituyen un grupo de prácticas tradicionales pero llevadas al extremo, de las cuales se toma la esencia y se aplican buscando la calidad en el desarrollo desde el inicio hasta la entrega final del sistema, teniendo en cuenta el soporte, mantenimiento, auditoría y capacitaciones al usuario final. Permiten hacer pequeñas entregas de *software*, con ciclos rápidos. Tanto el cliente como los desarrolladores trabajan juntos constantemente, manteniendo una buena comunicación.

Según el Manifiesto Ágil, en el cual se recogen los principios fundamentales de estas metodologías se valora:

- **Al individuo y las interacciones del equipo de desarrollo sobre el proceso y las herramientas**. Los integrantes del equipo son el principal factor de éxito de un proyecto. Las herramientas mejoran la eficiencia, pero son las personas con conocimiento técnico y actitud adecuada las que producen los resultados observables.
- **Desarrollar software que funcione, más que conseguir una buena documentación**. En este punto se resalta que aunque los documentos son soporte de documentación no se deben producir a menos que sean realmente necesarios para tomar una decisión importante. En caso de producirse deben ser cortos y centrarse en lo fundamental.
- **La colaboración con el cliente más que la negociación de un contrato**. Se debe valorar al cliente como un miembro más del equipo para que mantenga una interacción constante con el grupo de trabajo. Esta colaboración será la que marque la marcha del proyecto y asegure su éxito.
- **Responder a los cambios más que seguir estrictamente un plan**. Este punto plantea que la respuesta dada ante los cambios que puedan surgir a lo largo del proyecto determina el éxito o fracaso del mismo. Por lo tanto, la planificación no debe ser estricta sino flexible y abierta.

1.4.3 Proceso de Desarrollo y Gestión de la UCID

El Proceso de Desarrollo y Gestión de la Unidad de Compatibilización, Integración y Desarrollo de Software para la Defensa (UCID) es una metodología creada por sus especialistas. Recoge características de las metodologías tradicionales y ágiles para la producción eficiente de productos de *software* que satisfagan los requisitos de un cliente, llevando a cabo una estimación y planificación de recursos predecibles. Tiene elementos y relaciones que responden Quién debe hacer Qué, Cuándo y Cómo. Esto se logra modelando las interacciones y relaciones que suceden entre las personas (roles), las actividades que estas desarrollan y los artefactos que se crean o actualizan durante el proceso (6).

Figura 4. Relación entre elementos del proceso de software.

Esta metodología propone que el modelo de desarrollo de *software* describa la secuencia de actividades de alto nivel para construir y desarrollar las soluciones, para lo cual se combinan los modelos basados en componentes con el iterativo e incremental. Sus características son:

Desarrollo iterativo e incremental: el ciclo de vida está compuesto por iteraciones, es decir, pequeños procesos compuestos de varias actividades en las cuales se obtiene un sistema parcialmente completo, probado, integrado y estable. Cada iteración se va uniendo hasta obtener el producto de *software* completo.

Desarrollo basado en componentes: pone a prueba cada uno de los componentes por separado, antes de ensamblar el conjunto completo. Esto posibilita que en caso de existir un débil acoplamiento, el

desarrollador pueda actualizar y/o agregar componentes sin afectar otras partes del sistema. Cada vez que se construya o mejore un componente la calidad de la aplicación mejorará con el paso del tiempo.

Ciclo de Vida del Proyecto

El ciclo de vida de un proyecto consiste en una serie de fases, muchas veces secuenciales, qué definen el trabajo técnico que se hará en cada una de ellas, cuándo se generarán los entregables, cómo se revisarán, verificarán y validarán, quién está involucrado en cada fase y cómo controlar y aprobar cada una de ellas.

Las principales características del ciclo de vida expuesto en esta metodología son (6):

- Las fases son secuenciales y su transferencia debe ser precedida por un proceso de revisión o liberación del Centro de Calidad y su aprobación en el Consejo Técnico Formal.
- El nivel del personal es bajo al comienzo, alcanza su nivel máximo en la fase de construcción y decae rápidamente cuando el proyecto se aproxima a su conclusión.
- La participación de los interesados es alta en las etapas de Inicio y Modelación, baja en la etapa de Construcción y vuelve a subir en las etapas finales del proyecto.

Fases del Ciclo de Vida:

El ciclo de vida de un proyecto de *software* desarrollado en la UCID se descompone en cinco fases secuenciales: Inicio, Modelación, Construcción, Explotación experimental y Despliegue. Al final de cada fase se realiza una evaluación para determinar si los objetivos se cumplieron y si se pasa o no a la siguiente fase.

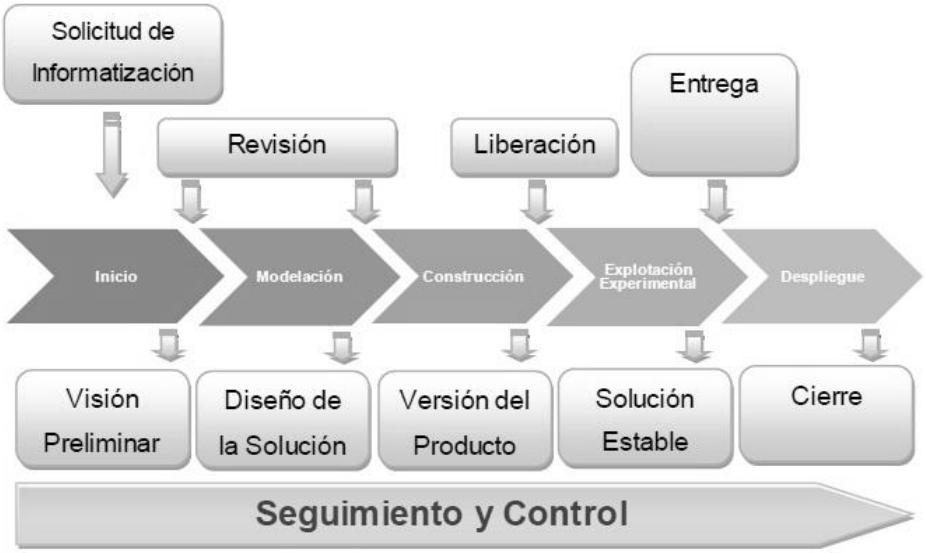

Figura 5. Etapas del ciclo de vida del proyecto.

Descripción de las Fases

Inicio: en esta fase se recibe la solicitud de informatización y se le asigna una prioridad. Se logra una visión de la problemática a resolver, se identifica el alcance del proyecto, se especifican los involucrados y se estiman los recursos necesarios. Luego se establece la estrategia a seguir para realizar la captura de requisitos y se realiza la planificación detallada de la fase de Modelación.

Modelación: en esta fase se definen los procesos del dominio del problema, se estiman los principales riesgos que presenta el proyecto y se especifica la forma de mitigarlos. Se identifican las necesidades del usuario, de las que se derivan los requerimientos del producto a desarrollar, se determina la factibilidad operativa, técnica y/o económica de continuar el proyecto y se define la arquitectura del *software*.

Construcción: en esta fase se aclaran los requisitos restantes y se completa el desarrollo del sistema sobre una base estable de la arquitectura. Todas las características, componentes y requerimientos son integrados, implementados y probados en su totalidad, obteniéndose una versión estable del producto, llamada beta.

Explotación experimental: esta fase asegura que el *software* esté disponible para que un grupo de usuarios realicen las pruebas de aceptación para ser entregado, y se hagan los ajustes necesarios. En este punto del ciclo de vida, la retroalimentación de los usuarios debe enfocarse fundamentalmente en ajustes específicos y de corto alcance al producto junto a otros temas como configuración, instalación, y usabilidad.

Despliegue: En esta fase se realiza la generalización del producto en las entidades y órganos. Durante el proceso de implantación por lo general no es necesaria la participación de los integrantes del equipo de desarrollo.

1.5 Análisis de las herramientas de desarrollo

Las herramientas de desarrollo de *software* son la base para implementar los disímiles tipos de *software*. Permiten automatizar acciones bien definidas y mejorar la calidad y productividad en el diseño y desarrollo. Proporcionan importantes ventajas para el equipo de trabajo, debido a que apoyan a las metodologías y métodos, integrando actividades y propiciando una visión de continuidad entre las fases metodológicas.

1.5.1 Herramientas CASE

CASE son las siglas que corresponden a las iniciales de: Computer Aided Software Engineering; y en su traducción al español significa Ingeniería de Software Asistida por Computación.

Las herramientas CASE son un conjunto de programas que ayudan a automatizar el diseño de un *software* y el proceso de desarrollo durante todo el ciclo de vida del mismo. Estos programas permiten elaborar los modelos que describen la empresa y sus actividades, pasando por el análisis y diseño de sistemas, hasta la generación del código y la documentación; además, de llevar a cabo la planificación. Mejoran la calidad del *software* desarrollado, así como la gestión y dominio del proyecto reduciendo tiempos y costos de desarrollo.

1.5.1.1 Visual Paradigm Suite 3.4

Visual Paradigm for UML (por sus siglas en inglés, Unified Modeling Language) es una herramienta profesional que soporta el ciclo de vida completo del desarrollo de *software*: análisis y diseño orientados a objetos, construcción, pruebas y despliegue (7). Permite construir aplicaciones de gran calidad a un menor

coste, dibujar los diagramas de clases y generar documentación y código base para diferentes lenguajes de programación a partir de estos. Su notación es muy parecida a la estándar, permite configurar las líneas de redacción, el modelado de base de datos, de requerimientos y del proceso de negocio; además, de posibilitar la integración con herramientas de desarrollo IDE (acrónimo en inglés de Integrated Development Environment), lo que en español significa Entorno de Desarrollo Integrado.

Posee gran éxito entre los usuarios ya que presenta un ambiente gráfico agradable y no requiere de plataforma única, es decir, se puede ejecutar sobre diferentes sistemas operativos.

1.5.1.2 Rational Rose

Rational Rose es una poderosa herramienta para el análisis y diseño de sistemas basados en objetos, perteneciente a la familia de productos UML de IBM. Utiliza un proceso de desarrollo iterativo controlado, que cubre todo el ciclo de vida de un proyecto. Cada iteración comienza con una primera aproximación del análisis, diseño e implementación, para identificar los riesgos del diseño. Una vez identificados se prueba la aplicación para que éstos se hagan mínimos. Muchas de sus versiones soportan generación para C++, Java, J2EE y CORBA, y están integradas en IDEs populares tales como Borland, JBuilder y Microsoft Visual Studio.

1.5.1.3 Selección de la herramienta CASE a utilizar

Se decide utilizar Visual Paradigm para UML debido a que es una herramienta muy potente, fácil de utilizar y presenta un entorno amigable para los usuarios. Para los desarrolladores de *software* es una plataforma puntera para construir con rapidez aplicaciones baratas y de gran calidad. Su interoperabilidad con otras herramientas CASE y muchos de los entornos IDE líderes del mercado es excelente. Además, no requiere de plataforma única como la necesita Rational Rose, lo que posibilita que la realización de la solución propuesta sea llevada a cabo en el sistema operativo GNU/Linux.

1.5.2 Entorno de desarrollo a utilizar

Los Entornos de Desarrollo Integrado son un conjunto de herramientas para el programador, incluidos en una misma suite, un buen editor de código, administrador de proyectos y archivos, enlace transparente a compiladores y debuggers e integración con sistemas controladores de versiones o repositorios (8).

Se decide usar como entorno de desarrollo el OpenOffice.org debido a que cuenta con un editor de textos en el cual se pueden crear y manipular macros que permiten añadir nuevas funcionalidades a los documentos. Dicho editor admite los lenguajes OpenOffice.org Basic, JavaScript y Python. Es por ello que puede ser usado para añadir a los eventos[13] de guardar y abrir documentos, diferentes métodos de encriptación, proporcionando seguridad a los mismos.

El OpenOffice.org, como entorno, posibilita manipular documentos en diferentes formatos como: .doc, .xls y .ppt, además de su formato nativo el odf, el cual es estándar y abierto para documentos de oficina, y fue certificado por la Organización Internacional de Estandarización (denominada ISO por las siglas en inglés de International Organization for Standardization) en 2006. Puede manejar otros tipos de ficheros como pdf o html. Está disponible para diversos sistemas operativos como Mac OS X, Microsoft *Windows*, Linux, FreeBSD y Solaris, y se puede encontrar en múltiples idiomas.

1.5.2.1 Introducción a las macros de la suite OpenOffice.org

Las operaciones tradicionales que se pueden realizar en el OpenOffice.org pueden ser automatizadas, al igual que aquellas funciones que se hacen repetidamente. Esto se logra a través de macros que son guardadas en un registro, al que posteriormente se hace referencia para la ejecución de estas tareas. Una macro es un conjunto de operaciones que son memorizadas para repetirlas cada vez que se desee. Estas se pueden activar con alguna tecla, un botón o manualmente. Las macros se alojan en unos contenedores llamados módulos, los cuales pueden pertenecer a las carpetas My macros, Macros de OpenOffice.org o Untitled 1. En el primer nivel se crean o modifican las macros que pueden personalizar el OpenOffice instalado en la computadora. En el segundo se encuentran las definidas ya en el sistema y en el último nivel, están las macros que han sido asignadas a un documento en específico.

1.5.2.2 Crear una macro

Para acceder al entorno de programación de la suite OpenOffice.org es necesario ingresar a cualquiera de sus programas, luego se selecciona en la Barra de Menús la opción **Herramientas>Macros>Organizar macros>OpenOffice.org Basic.** Esto abrirá la ventana de macros de OpenOffice.org Basic (Figura 6). Para poder personalizar el OpenOffice.org los módulos que contendrán las macros serán creados en el nivel My Macros. Para ello se accede al botón **New** y se podrá conservar el nombre que trae por defecto o cambiarlo. Suponiendo que se llama **Module1** se pulsa el botón **Aceptar** y quedará creado el mismo

dentro de una subcarpeta de My Macros llamada **Standard**. El programa crea una macro llamada **Main** que puede ser editada presionando el botón **Edit.**

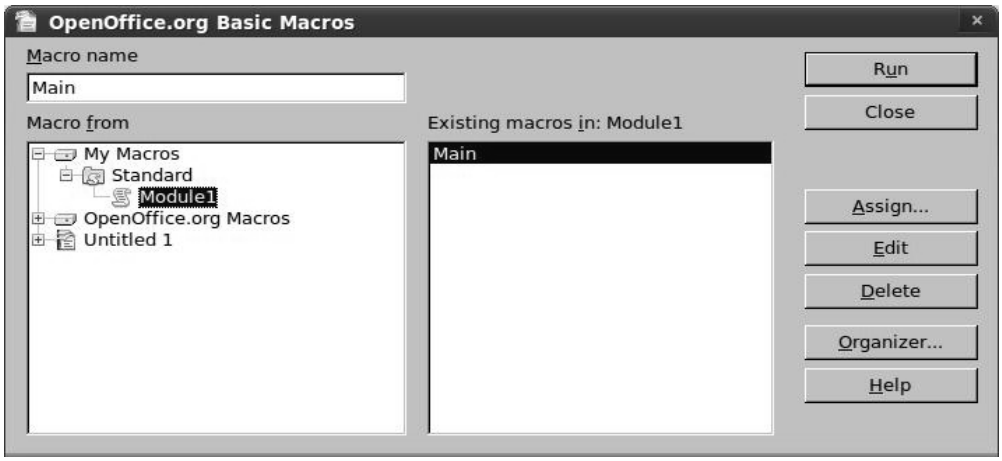

Figura 6. Ventana de macros de OpenOffice.org Basic.

Esta opción despliega el entorno de trabajo de la suite donde se escribe el procedimiento principal y las funciones auxiliares o subrutinas (Figura 7).

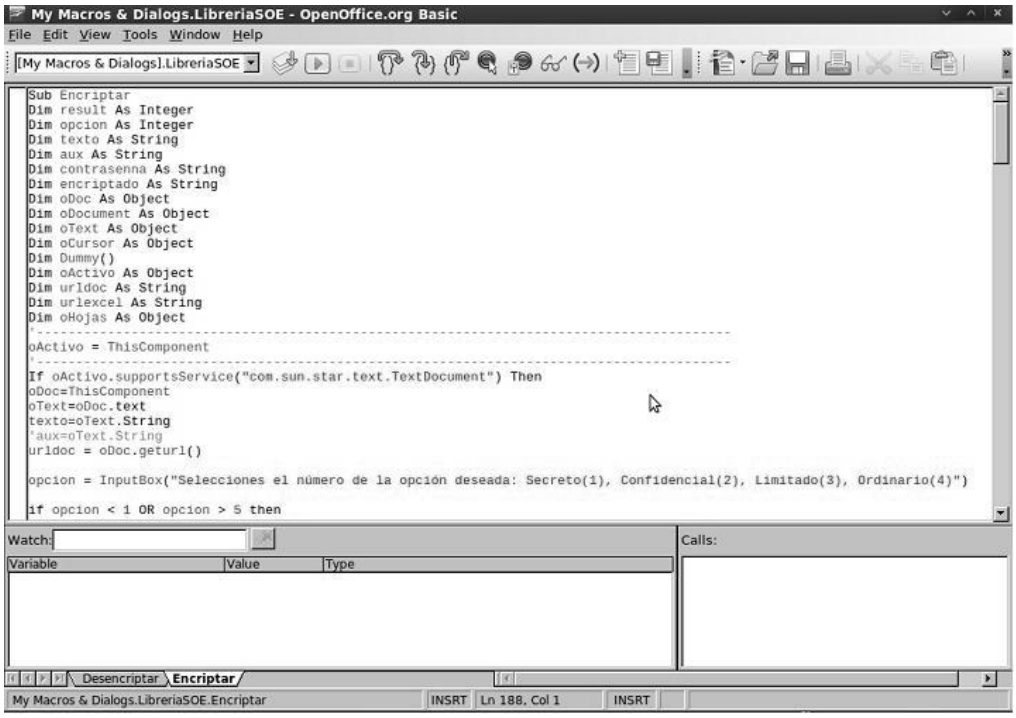

Figura 7. Entorno de trabajo de la suite.

1.5.2.3 Asignar una macro a un objeto

Las macros se pueden ejecutar mediante la secuencia **Herramientas>Macros>Organizar macros>Openoffice.org Basic** y seleccionando la opción **Run,** pero esto tomaría mucho tiempo. Sería preferible ejecutarlas con una sola pulsación de ratón. Para ello basta con asignar la macro a un botón de la Barra de Herramientas, algún menú o evento. Para realizar esto se selecciona el módulo deseado en **My Macros** y se accede al botón **Assign** como se muestra en la figura 6, esto mostrará la siguiente ventana (Figura 8). Aquí se puede crear un nuevo menú o un nuevo botón en la Barra de herramientas a los que se le puede poner el nombre y el ícono deseado. (Para cambiar el nombre y el ícono ver Manual de usuario). Si no se desea crear uno nuevo se puede asignar la macro a un botón o menú existente. Una vez seleccionada la opción a la cual se le añadirá la macro se accede al botón **Add** y se selecciona en **My Macros** el módulo deseado (Figura 9).

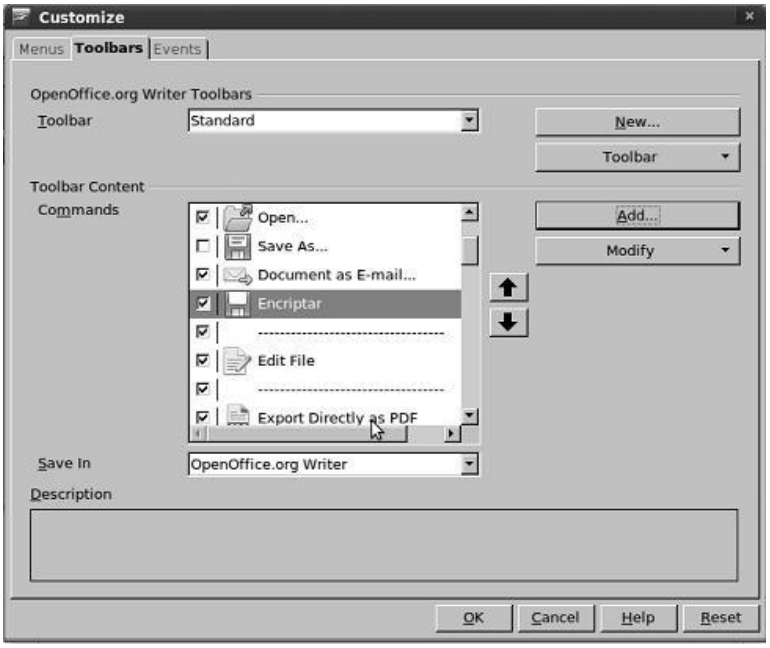

Figura 8. Ventana para asignar las macros a un botón.

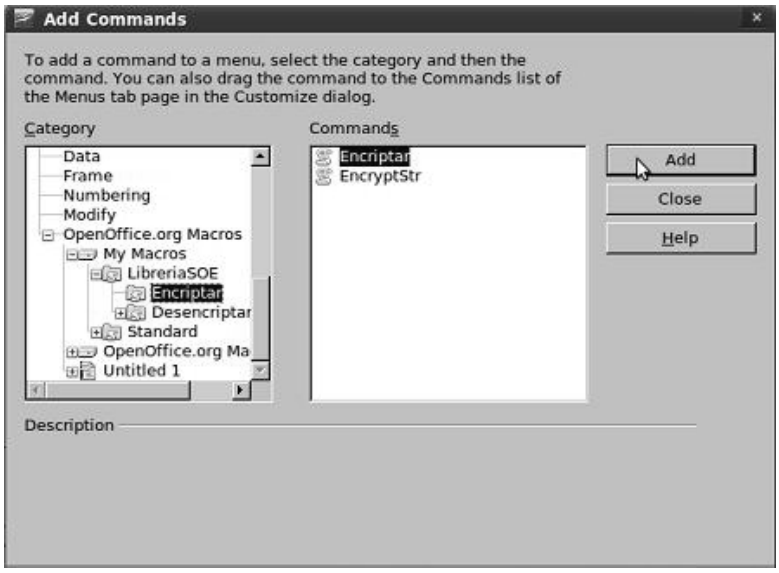

Figura 9. Ventana para seleccionar la macro a asignar.

En caso de querer seleccionar un evento, por ejemplo, si se desea que se ejecute la macro cada vez que se cierre o se abra el documento, se activa la pestaña **Events**, se selecciona uno de ellos y se le asigna con el botón **Macro**. Se debe cambiar el nombre que aparece en **Save in** por OpenOffice.org para garantizar que se haga en todos los documentos. Si se mantiene el nombre que aparece inicialmente solo se hará para ese documento. Una vez hecho esto, aparecerá el nombre del módulo a la derecha del evento **(Figura 10).**

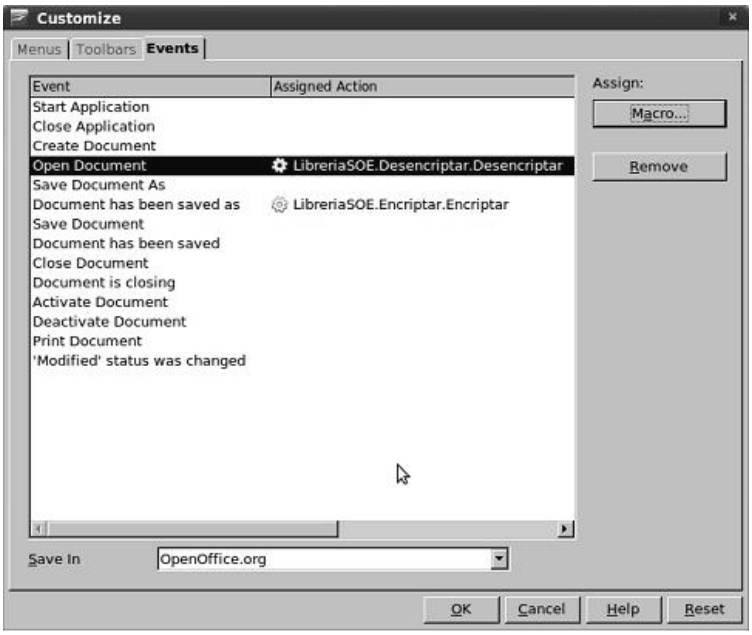

Figura 10. Ventana para asignar las macros a un evento.

1.5.2.4 Exportar Librería

Todos los módulos creados anteriormente tienen como desventaja que sólo se ejecutarán en ese ordenador. Si el OpenOffice.org es desinstalado o se borra la carpeta donde se guardan sus configuraciones se perderán todas las tareas que habían sido automatizadas a través de las macros. Para evitar esto, se crea una librería que contendrá todos los módulos necesarios. Es decir, en lugar de añadirlos a **Standar**, se crea una nueva de la siguiente forma: se selecciona en la ventana de macros de OpenOffice.org Basic (Figura 6) el botón **Organizer**, en la ventana mostrada se selecciona la pestaña **Libraries** y se crea la librería con el nombre que se desee presionando **New**. Una vez creada se procede a añadir los módulos que sean necesarios pero esta vez seleccionando la pestaña **Modules**. Estos se asignan a los botones o eventos deseados de la misma forma en que se hizo anteriormente para la librería **Standar**. Posteriormente se puede exportar para cualquier lugar de la comptadora elegido por el usuario. Para ello se selecciona con el cursor, se accede a la opción Export, se activa **Export as BASIC library** y se escoge el lugar en el equipo (Figura 11). En caso de que haya habido cambios en el OpenOffice.org se puede importar accediendo al botón **Import**, buscando su ubicación en el equipo y señalando **script.xlb.**

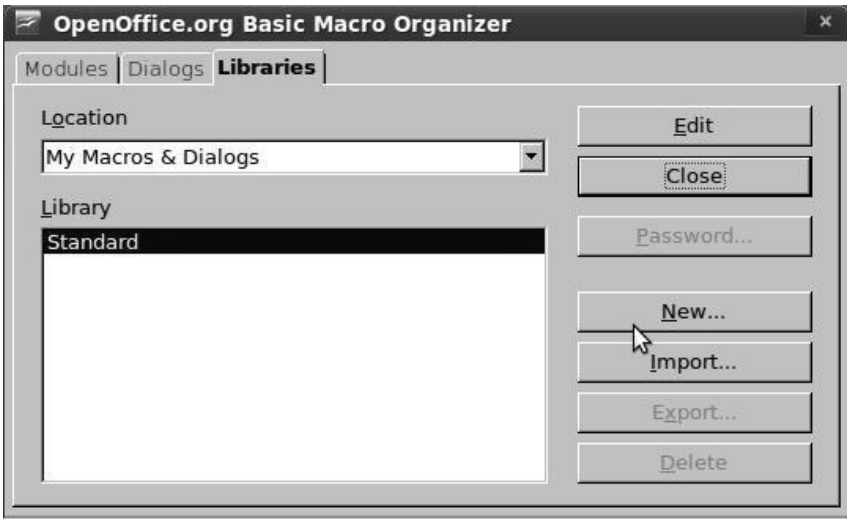

Figura 11. Ventana para crear una nueva librería.

1.5.2.5 Proteger librería con contraseña

Una ventaja de las macros es que se pueden proteger con contraseñas y evitar de este modo que su código sea modificado por personas sin los permisos necesarios. Para ello se selecciona la librería deseada en la pestaña **Libraries** (Figura 11) y se presiona el botón **Password**.

La aplicación debe cerrarse para que guarde los cambios y una vez que se abra nuevamente solo se podrán hacer modificaciones si se conoce la contraseña (Figura 12).

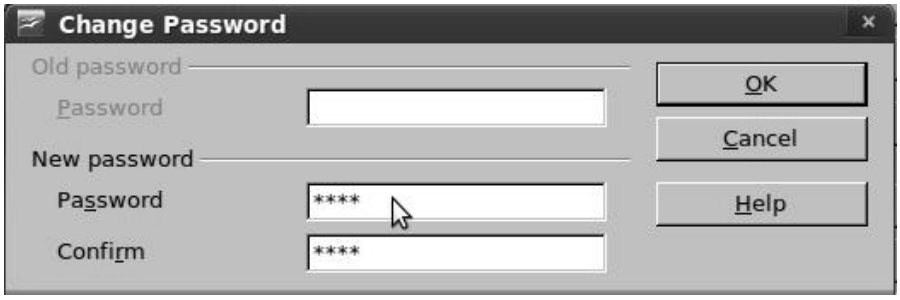

Figura 12. Ventana para proteger la librería con contraseña.

1.6 Posibles lenguajes a utilizar

JavaScript

Es un lenguaje de programación sencillo y pensado para hacer cosas con rapidez. Su uso posibilita interactuar con el navegador de manera eficaz para crear páginas web con dinamismo. Los programas JavaScript van incrustados en los documentos HMTL, permiten crear efectos especiales y definir interactividades con el usuario. Técnicamente, es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos, es decir, se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios. Toda esta potencia se pone a disposición del programador; verdadero dueño y controlador de todo lo que ocurre en la página.

Python

Es un lenguaje interpretado, interactivo, fuertemente tipado, orientado a objetos y que presenta características de los lenguajes de alto nivel. Su sintaxis sencilla, el gestor de memoria, la gran cantidad de librerías disponibles y la potencia, entre otros aspectos, hacen que desarrollar una aplicación en Python sea sencillo, rápido y eficiente.

Algunas de sus características son:

- Muy legible y elegante.
- Simple y poderoso.
- Soporta objetos y estructuras de alto nivel como string, listas, diccionarios, etc.
- No se tienen que declarar variables o constantes antes de utilizarlas.
- Alta velocidad de desarrollo y buen rendimiento.
- Se puede utilizar en múltiples plataformas.

Para programar en Python sólo se necesita un editor de texto. Aunque se puede utilizar el Anjuta, WingIDE, Eclipse, entre otros. Sin embargo, no es adecuado para la programación de bajo nivel o para aplicaciones en las que el rendimiento sea crítico.

OpenOffice.org Basic: el lenguaje de macros

OpenOffice.org (OOo) Basic, es el lenguaje de programación de macros de la suite de aplicaciones ofimáticas OpenOffice.org. Pertenece a la familia de lenguajes Basic, manteniendo muchas de sus partes idénticas al Visual Basic y Microsoft Visual Basic for Applications. Al igual que estos, su código es

fácilmente migrable y muy extendido, por lo que se pueden encontrar fuentes de información y documentación para los proyectos.

Permite a los usuarios programar pequeñas aplicaciones usando la potencia, versatilidad y facilidad de uso que ofrecen todos los componentes disponibles en la suite, como Writer, Calc, Impress, Draw y el Gestor de Base de Datos integrado en la misma.

Es un lenguaje interpretado, es decir, necesita un entorno de ejecución para ejecutar línea a línea el código del programa. Desde dicho entorno se pone en marcha el intérprete de OOo Basic. Este intérprete, comprueba que la sintaxis del código sea correcta y posteriormente lo ejecuta. Para programar las macros se utiliza el editor de desarrollo integrado (IDE) en OpenOffice.org, al que se accede desde cualquier programa de la suite pulsando en Herramientas/Macros/Organizar Macros/OpenOffice.org Basic.

1.6.1 Selección del lenguaje a utilizar

Se decide utilizar OpenOffice.org Basic porque es un lenguaje fácil de aprender, que posee un manejo muy sencillo de cadenas de caracteres. Es soportado por el editor de textos del OpenOffice.org, por lo que permite que las funcionalidades propuestas puedan ser implementadas. Tiene sentencias propias del lenguaje Visual Basic for Applications (VBA) que permite codificar módulos, o como también se le conoce: macros, para las aplicaciones del OpenOffice.org.

1.7 Lenguaje de Modelado Unificado

UML es un lenguaje de modelado unificado es un lenguaje que permite visualizar, especificar, construir y documentar el modelado de sistemas (9). Se utiliza para describir métodos o procesos y para definir un sistema, es decir, es el lenguaje en el que está descrito el modelo. Tiene diferentes tipos de diagramas categorizados jerárquicamente en Diagramas de Estructura y Diagramas de Comportamiento. Dentro de los Diagramas de Comportamiento se encuentran los Diagramas de Transición de Estados (DTE).

El DTE es utilizado para identificar los caminos o cada una de las rutas que puede tomar un flujo de información luego de ejecutarse cada proceso. Además, identifica bajo qué argumentos se ejecuta cada uno de los procesos y en qué momento podrían tener una variación. Permite visualizar de una forma secuencial la ejecución de cada uno de los procesos. Sus componentes son:

- Estados: comportamiento del sistema que es observable en el tiempo (tiene un estado inicial, pero puede tener múltiples estados finales).
- Cambios de estados: condiciones y acciones.

Cada rectángulo representa un estado en el que se puede encontrar el sistema.

Figura 13. Diagrama de transición de estado.

1.8 Conclusiones

Luego de un análisis de las tecnologías y herramientas existentes para el desarrollo de *software* se escogieron las que presentaron mayores ventajas. Se decide modificar la aplicación sobre la base del Proceso de Desarrollo y Gestión de la UCID, con el apoyo de la herramienta Visual Paradigm para el modelado de las clases, ya que es multiplataforma. Se seleccionó el lenguaje de programación OpenOffice.org Basic, como lenguaje de modelado UML y como entorno de desarrollo el OpenOffice.org.

CAPÍTULO 2. CARACTERÍSTICAS DE LA APLICACIÓN

2.1 Introducción

En el presente capítulo se describe el flujo actual del trabajo realizado en las oficinas de las FAR con los documentos. Se propone la modificación de las funcionalidades Guardar y Abrir del Procesador de textos y la Hoja de cálculos, para que el texto de los documentos sea guardado con la seguridad requerida y se dan a conocer las personas relacionadas con el OpenOffice.org. Se realizan las actividades referentes a los flujos de trabajo de Modelación de negocio, Especificación de los requisitos de software y Diseño, generándose los artefactos correspondientes.

2.2 Objetivos estratégicos de las FAR

Con la realización del Sistema Ofimático Encriptado las FAR persiguen como objetivo cifrar el texto contenido en los documentos digitales elaborados con el Procesador de textos y la Hoja de cálculos, de modo que no quede guardado en el disco duro en texto plano y que pueda ser leído por personas sin la debida autorización.

Con la encriptación de estos documentos se obtiene un conjunto de datos ilegibles, imposibles de entender si no se cuenta con la contraseña correspondiente al algoritmo de cifrado. Esto resulta ser de gran importancia para los oficiales y trabajadores civiles de las FAR debido a la confidencialidad y seguridad de los documentos que ellos manipulan en sus ordenadores. Por tanto, el OpenOffice.org encriptado garantizará en gran medida una estricta protección de los archivos de tipo texto, ocultándolos de intrusos que quieran hacer algún daño utilizando esta información.

2.3 Flujo actual del trabajo ofimático llevado a cabo en las FAR

En las instituciones de las FAR toda la información relacionada con la entidad, sus trabajadores civiles y oficiales, así como de las actividades que se realizan en la misma, queda evidenciada en documentos impresos o en formato digital. Muchas veces esta información requiere de cierta seguridad por lo que es clasificada en diferentes niveles. En dependencia del rango de los oficiales y trabajadores civiles, unos pueden acceder a unos documentos pero a otros no, garantizando de esta forma que no sean vistos por personas sin los permisos necesarios, o que sean modificados o eliminados.

Actualmente, dada la migración que se lleva a cabo en las FAR hacia el sistema operativo GNU/Linux, el trabajo ofimático se realiza con la suite OpenOffice.org. Por tanto, los documentos elaborados solo

cuentan con la seguridad que brinda el sistema operativo instalado en las computadoras de los oficiales y trabajadores, además de las políticas de seguridad establecidas en cada entidad para garantizar la confidencialidad, integridad y disponibilidad de los mismos. Una vez archivados en el disco duro quedan en texto plano, es decir, cualquier persona sin los debidos permisos puede consultar la información contenida. Es necesario que las funcionalidades "Guardar" y "Abrir" documentos permitan hacerlo utilizando métodos de encriptación, para mejorar la seguridad y posibilitar un acceso estricto a la información.

Actualmente no existen soluciones de este tipo en las instituciones de las FAR, por lo que el Sistema Ofimático Encriptado será superior al OpenOffice.org usado hoy en día y fortalecerá la restricción que necesitan los documentos secretos.

2.4 Información que se maneja

Los documentos que se manejan en las entidades militares están relacionados con todas las actividades que se realizan dentro de la misma, contienen información sobre sus oficiales y trabajadores civiles. Ejemplo de estos son las actas de sus reuniones, de los eventos que se realizan, las órdenes que se emiten dentro y fuera de la institución, las orientaciones que reciben de otras unidades, los estatutos, reglamentos y directivas. Según el grado de confidencialidad necesario están clasificados en diferentes niveles, que regulan el acceso de los oficiales y trabajadores civiles según el rango y cargo de los mismos. Para el desarrollo de las funcionalidades se trabajará con las siguientes clasificaciones:

- **Confidencialidad máxima:** Estos documentos requieren de una máxima protección.
- **Confidencialidad media:** Estos documentos no necesitan una protección tan estricta, pero si deben mantener en gran medida su confidencialidad.
- **Confidencialidad mínima:** Estos documentos requieren poca protección.

2.5 Propuesta de las funcionalidades a diseñar

Como funciones que serán objeto de automatización se plantean las siguientes:

Encriptar texto: Codifica la información contenida en los documentos de manera que no pueda ser leída.

Desencriptar texto: Descifra la información contenida en los documentos para que pueda ser leída.

Estas funcionalidades serán dos opciones que le permitirán a las personas que trabajan con el OpenOffice.org guardar y abrir los documentos utilizando algoritmos criptográficos, para que no puedan ser leídos por individuos sin la debida autorización.

Para encriptar el texto, una vez que el usuario tiene el documento abierto, debe guardar el mismo mediante la opción Guardar como. Esta acción solicitará al usuario la clasificación deseada, que puede ser Máxima, Media o Mínima. Seguidamente se le solicita la contraseña. Una vez realizado estos procedimientos el documento es guardado en el disco duro de la computadora de forma encriptada, pero permanece abierto para que el usuario pueda seguir trabajando en el mismo. Estos datos solo se solicitan la primera vez que es guardado el documento, por lo que en caso de existir cambios en el mismo, solo se accederá al botón Guardar para actualizar el texto en el disco duro del ordenador. Si el usuario decide cerrar el documento y este no ha sido guardado, se realiza entonces el procedimiento explicado anteriormente.

Para desencriptar el texto del documento, una vez que se accede al mismo, es necesario volver a escribir la clasificación y la contraseña correcta para que se muestre el texto en un formato legible, de lo contrario, sólo se mostrará en pantalla un conjunto de datos sin valor para el usuario.

2.6 Personas relacionadas con la aplicación

Las personas relacionadas con la aplicación son aquellas que se benefician del resultado de uno o varios procesos que se ejecutan en la misma. En este caso las personas vinculadas serían todos los oficiales y trabajadores civiles de las FAR que trabajan en diferentes oficinas con documentos digitales. Estas personas tienen permisos para guardar y actualizar la información de los mismos.

Otra persona relacionada con la herramienta es el administrador, quien es el encargado de velar por la seguridad de los datos de la aplicación. Para ello debe proteger con contraseña las librerías existentes, así como asegurar que cualquier trabajador no pueda modificar la información contenida en las carpetas del OpenOffice.org. Debe actualizar las macros en caso que sea necesario y posteriormente, actualizar la aplicación en todas las máquinas de los oficiales y trabajadores civiles.

2.7 Modelado de los procesos de negocio

Para comprender un sistema y gestionar su complejidad es necesario dividirlo en piezas. Estas se pueden representar mediante modelos que permiten extraer sus características esenciales. Una técnica para la

especificación de los requisitos más importantes del sistema, es el modelo del negocio, el cual dependiendo de la situación que se presente, tiene varias alternativas de desarrollo. Si los procesos están claramente definidos y no se van a introducir cambios, solo es necesario modelar el negocio propuesto. Si se determina que no es necesario el modelo completo del negocio se realizará lo que se conoce como un modelo de dominio.

Por tanto, teniendo en cuenta las características del OpenOffice.org, se decide realizar un modelo de dominio para capturar los tipos más importantes de objetos que existen y los eventos que suceden en el entorno donde estará la aplicación.

2.7.1 Identificación de conceptos en el dominio del problema

Tabla 2.1 Concepto Macro

Nombre de la entidad	Macro					
Descripción de la entidad	Operación que es memorizada para repetirse cada vez que se desee.					
Nombre del atributo	**Descripción**	**Tipo**	**¿Puede ser nulo?**	**¿Es único?**	**Restricciones**	
					Clases válidas	**Clases no válidas**
Subrutinas	Contienen el código de la macro.	-	NO	NO	--	--
Funciones	Métodos que se ejecuta cuando se corre la macro.	-	SI	NO	--	--

Tabla 2.2. Concepto Trabajador de oficina

Nombre de la entidad	Trabajador de oficina					
Descripción de la entidad	Persona que manipula los documentos.					
Nombre del atributo	**Descripción**	**Tipo**	**¿Puede ser nulo?**	**¿Es único?**	**Restricciones**	
					Clases válidas	**Clases no válidas**

Permiso de acceso	Regula el acceso a los documentos.	-Cadena de caracteres.	NO	SI	-Letras. -Números.	-Caracteres extraños.

Tabla 2.3. Concepto Documento

Nombre de la entidad	Documento					
Descripción de la entidad	En el documento se encuentra contenida la información que se desea cifrar.					
Nombre del atributo	**Descripción**	**Tipo**	**¿Puede ser nulo?**	**¿Es único?**	**Restricciones**	
					Clases válidas	**Clases no válidas**
Nombre	Es el nombre del documento.	-Cadena de caracteres.	NO	SI	-Letras. -Números. -Caracteres. extraños.	\ / : * ? “ < > |
Contenido	Es el texto escrito en el documento.	-Letras. -Números. -Caracteres extraños. -Imágenes.	NO	SI	-Letras. -Números. -Caracteres extraños. -Imágenes	--
Tipo de documento	Se refiere al formato del documento.	-Cadena de caracteres.	NO	SI	-Letras.	-Números. -Caracteres extraños.
Clasificación	Se define según el nivel de confidencialidad del texto de un documento.	-Cadena de caracteres.	NO	SI	-Letras.	-Números. -Caracteres extraños.

Tabla 2.4. Concepto OpenOffice.org

Nombre de la entidad	OpenOffice.org					
Descripción de la entidad	Es la herramienta ofimática con la cual trabajan los oficiales y trabajadores civiles de las FAR.					
Nombre del atributo	**Descripción**	**Tipo**	**¿Puede ser nulo?**	**¿Es único?**	**Restricciones**	
					Clases válidas	**Clases no válidas**
Programas	Es el conjunto de módulos que trae la suite.	-Dibujo -Hoja de cálculo -Base de datos -Presentación -Procesador de texto	-	NO	--	--

2.7.2 Modelo conceptual

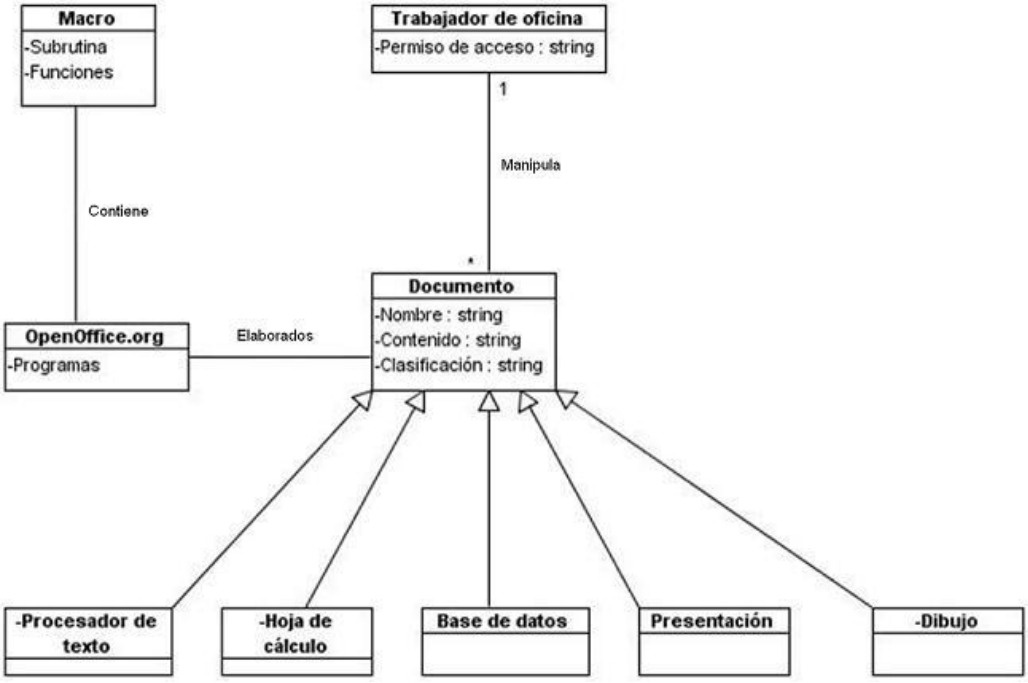

Figura 14. Modelo conceptual.

2.8 Especificación de los requisitos de software

El propósito de la definición de requisitos es definir las condiciones o capacidades que la aplicación debe cumplir y las restricciones bajo las cuales debe operar, logrando un entendimiento con los trabajadores de las oficinas de las entidades militares, y especificando las necesidades reales de forma que satisfagan sus expectativas.

2.8.1 Requisitos funcionales

Los requerimientos funcionales se definen como las condiciones o capacidades que el sistema debe cumplir. Se desea que la aplicación permita:

RF1: Encriptar texto

RF2: Desencriptar texto.

2.8.2 Requisitos no funcionales

Los requerimientos no funcionales son las cualidades o propiedades que la aplicación debe tener, estos representan las características de la misma.

Usabilidad

La aplicación propuesta debe ser fácil de usar por personas con muchas o pocas habilidades en informática, de manera que encriptar los documentos no sea una dificultad. El botón Guardar tanto en la Barra de herramientas como en el menú Archivo, deben brindar la posibilidad de encriptar el texto una vez que se dé clic sobre él.

Confiabilidad

Debe resolver de manera satisfactoria los problemas de seguridad de los oficiales, garantizando que el texto quede realmente encriptado.

Portabilidad

Debe ser multiplataforma, es decir, debe correr sobre el sistema operativo GNU/Linux o *Windows*.

Adquisición de componentes

Se requiere trabajar sobre el OpenOffice.org para poder modificarlo a través de su editor de textos y las macros creadas.

Legales

La aplicación debe ser reconocida y autorizada por instancias superiores tales como la UCI. Además, se debe basar en el Manual de Normas y Principios establecidos por el Ministerio de las Fuerzas Armadas Revolucionarias (MINFAR).

Confidencialidad

Para encriptar el documento la aplicación debe solicitar una contraseña y sólo con ella se podrá desencriptar el mismo.

Software

- Se requiere trabajar sobre el OpenOffice.org para poder dar solución al problema planteado.
- Cuando se instale el programa, debe estar seleccionado la herramienta de Ooo Basic, así como el asistente para macros en las casillas de verificación.
- Instalación de glibc2 en versión 2.1.3 o superior.
- XServer (resolución mínima 800x600 y 256 colores) con administrador de ventanas (p.ej. GNOME).
- Linux Kernel 2.2.13 o superior.

Hardware

- Computadora con procesador Pentium o compatible.
- 128 MB de memoria RAM como mínimo (256 ó 512 para las más recientes).

2.8.3 Descripción de los requisitos funcionales

Tabla 2.5. Descripción RF "Encriptar texto"

	Conceptos	Atributos
Conceptos tratados	OpenOffice.org	-Programas
	Macros	-Subrutina -Función

	Documento	-Nombre -Contenido -Tipo de documento -Clasificación
	Clasificación	-Confidencialidad máxima. -Confidencialidad media. -Confidencialidad mínima.
Precondiciones	**Precondiciones**	**Pre-requisito**
	El documento no debe estar vacío.	No procede.
Descripción	1. Se selecciona la opción Guardar. 2. Se muestra un submenú con los tipos de clasificación de la información. 3. Se selecciona el tipo de clasificación de la información para encriptar el texto del documento. 4. Se oprime el botón Aceptar. 5. Se muestra una ventana solicitando la clave para encriptar. 6. Se introduce la clave. 7. Se oprime el botón Aceptar. 8. La aplicación encripta el texto del documento.	
Flujos Alternos		
1.1: Se oprime el botón Guardar sin haber guardado el documento.	1. Se muestra un mensaje indicándole al usuario que debe usar la opción Guardar Como.	
3.1 Se introduce una clasificación incorrecta.	1. Se muestra un mensaje de error al usuario. 2. Se solicita nuevamente la opción deseada.	
4.1 Se orpime el botón Cancelar.	1. Se muestra un mensaje de error al usuario. 2. Se solicita nuevamente la opción deseada.	

6.1 Se deja el campo clave vacío o es incorrecto.	1. Se muestra un mensaje de error al usuario. 2. Se solicita nuevamente la clave deseada.
Validaciones	Se validan los datos según lo establecido en el Modelo Conceptual.
Post-condiciones	El texto del documento queda encriptado en el lugar donde ha sigo guardado.
Post-requisito	No procede.

Tabla 2.6. Descripción RF "Desencriptar texto"

Conceptos tratados	**Conceptos**	**Atributos**
	OpenOffice.org	-Programas
	Macros	-Subrutina -Función
	Documento	-Nombre -Contenido -Tipo de documento -Clasificación
	Clasificación	-Confidencialidad máxima. -Confidencialidad media. -Confidencialidad mínima.
Precondiciones	**Precondiciones**	**Pre-requisito**
	El texto del documento debe estar encriptado.	RF Encriptar texto.
Descripción	1. Se procede a abrir el documento. 2. Se muestra una ventana con los tipos de clasificacion de la información. 3. Se selecciona la clasificación deseada. 4. Se oprime el botón Aceptar. 5. Se muestra una ventana solicitando la clave para desencriptar. 6. Se introduce la clave. 7. Se oprime el botón Aceptar. 8. Se desencripta el texto del documento y se muestra el contenido en texto plano.	
Flujos Alternos		

3.1 Se introduce una clasificación incorrecta.	1. Se muestra un mensaje de error al usuario. 2. Se solicita nuevamente la opción deseada.
4.1 Se orpime el botón Cancelar.	1. Se muestra un mensaje de error al usuario. 2. Se solicita nuevamente la opción deseada.
6.1 Se deja el campo clave vacío o es incorrecto.	1. Se muestra un mensaje de error al usuario. 2. Se solicita nuevamente la clave deseada.
Validaciones	Se validan los datos según lo establecido en el Modelo Conceptual.
Post-condiciones	El documento queda abierto mostrando el contenido en texto plano.
Post-requisito	No procede.

2.8.4 Prototipos de interfaz de usuario

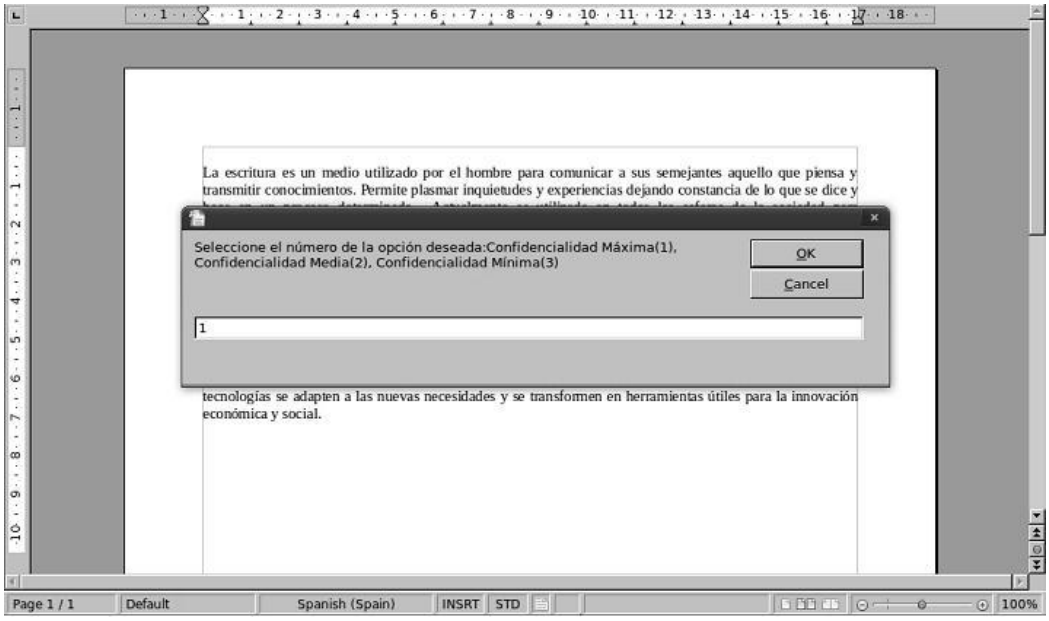

Figura 15. Solicitud de la clasificación deseada para encriptar el documento.

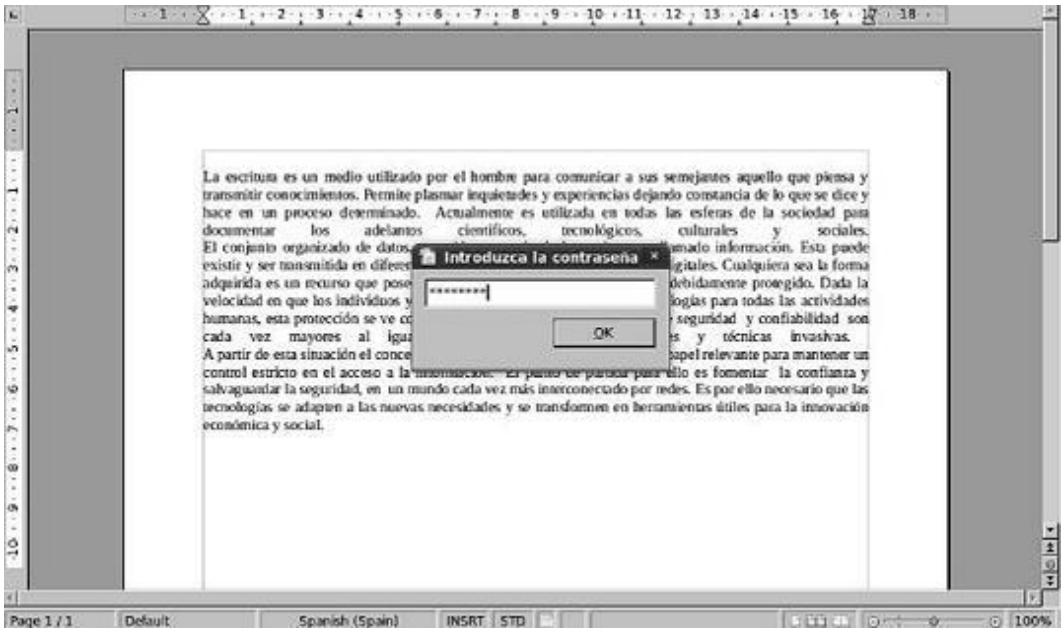

Figura 16. Solicitud de la contraseña deseada para encriptar el documento.

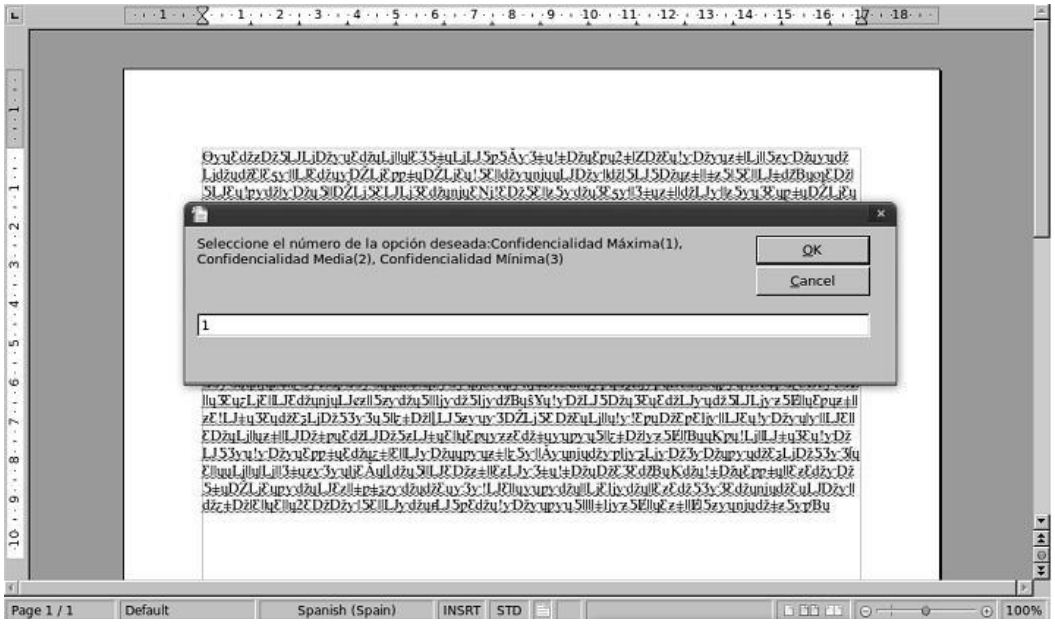

Figura 17. Solicitud de la clasificación deseada para desencriptar el documento.

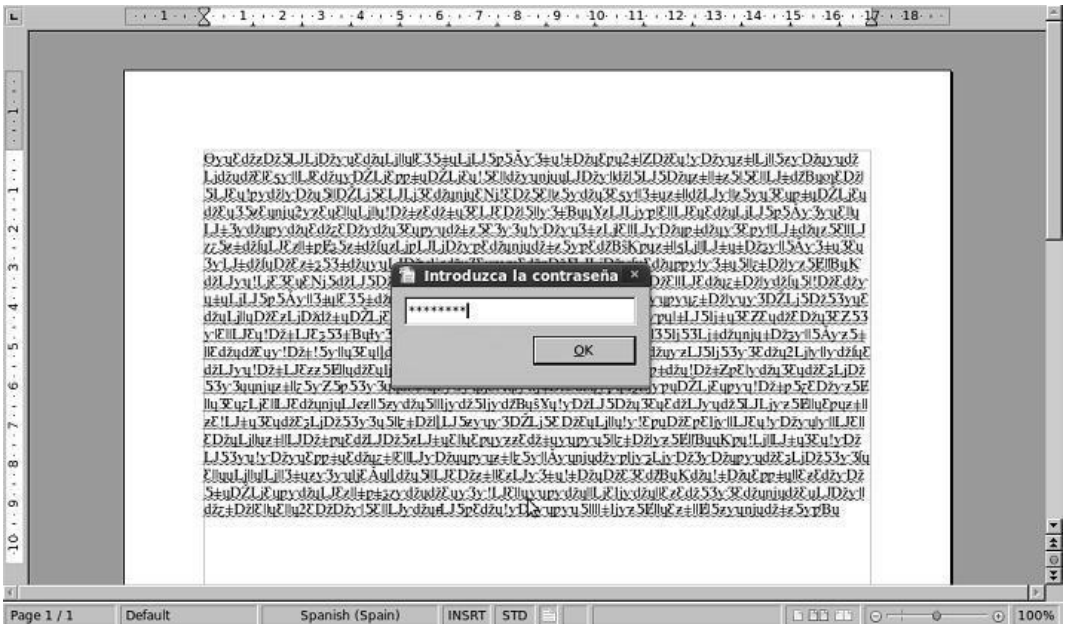

Figura 18. Solicitud de la contraseña deseada para desencriptar el documento.

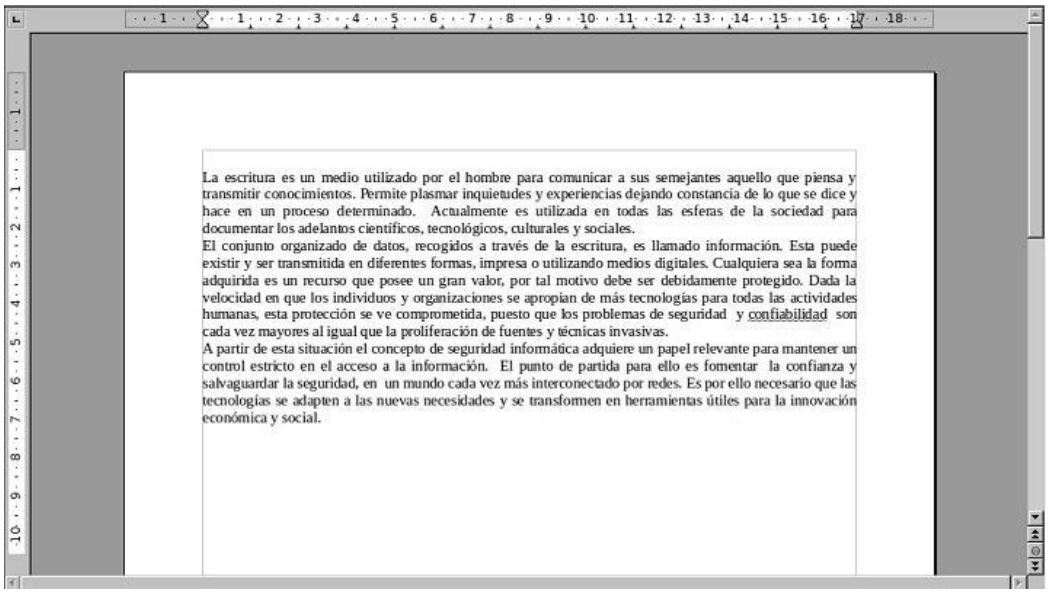

Figura 19. Documento desencriptado.

2.9 Diseño de la aplicación

El diseño del *software* permite obtener a partir de los requisitos funcionales y no funcionales de un sistema, una representación del mismo, que luego de varios refinamientos es utilizada como una entrada fundamental de las actividades de implementación. Durante este proceso se pueden capturar los requisitos o subsistemas individuales, interfaces y clases, descomponer los trabajos de implementación en partes más manejables y reflexionar sobre el diseño utilizando una notación común. Para obtener un mayor entendimiento de las funcionalidades a implementar se describen los diagramas de transición de estados correspondientes a cada requisito funcional.

2.9.1 Diagramas de transición de estados

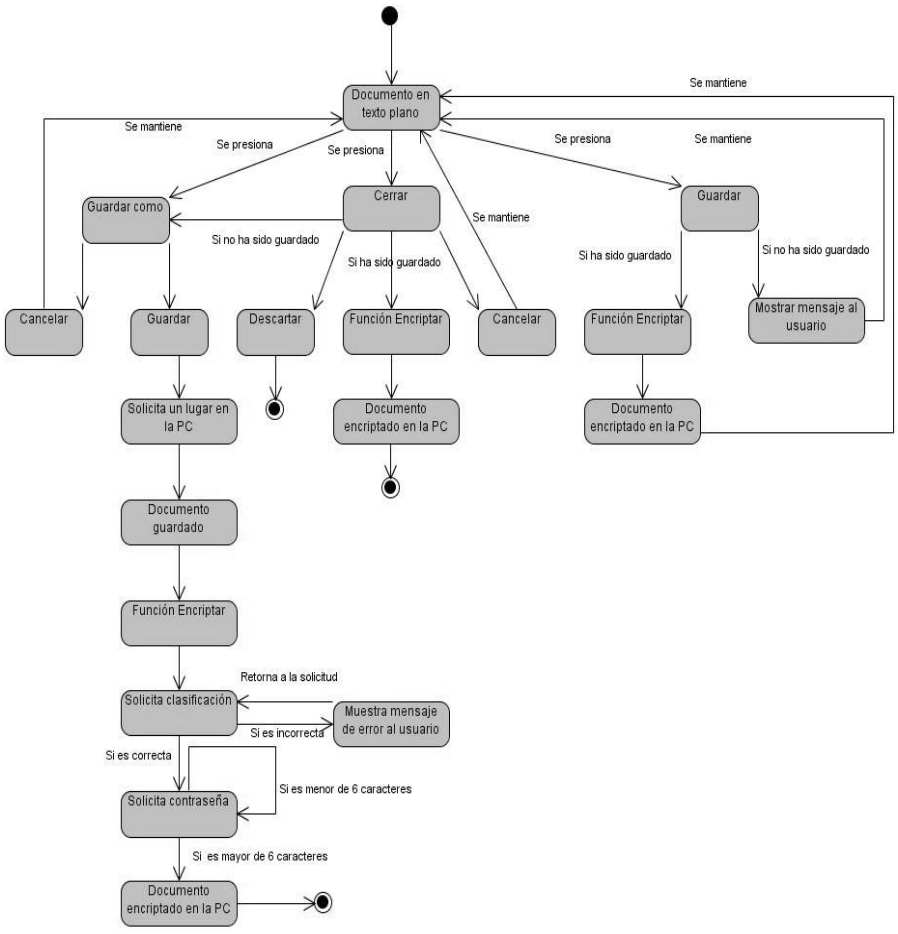

Figura 20. Diagrama de transición de estado del RF ¨Encriptar texto¨.

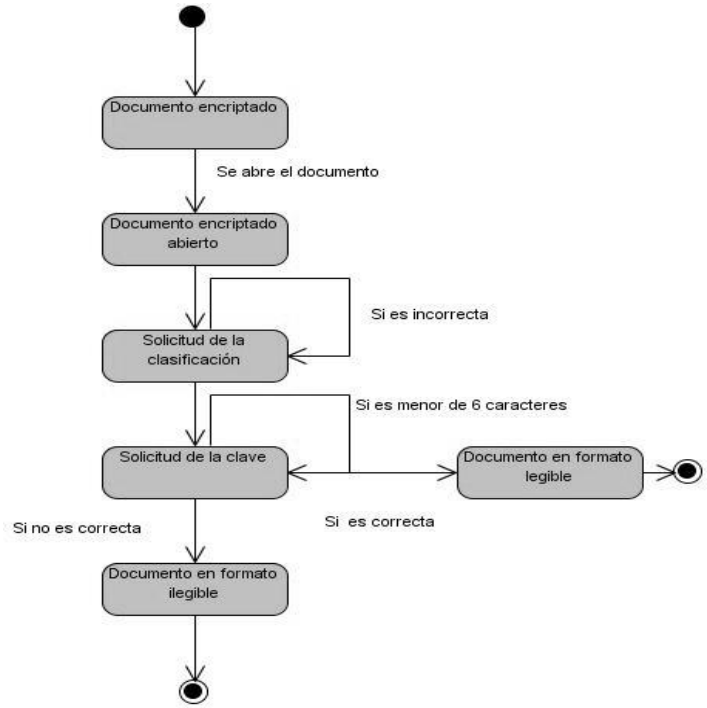

Figura 21. Diagrama de transición de estado del RF ¨Desencriptar texto¨.

2.10 Conclusiones

Durante el desarrollo del capítulo se describió el trabajo actual que se realiza con los documentos en las oficinas de las FAR. Este análisis permitió conocer que a pesar de existir en las entidades militares políticas de seguridad establecidas, se corre el riesgo de que la información sea consultada, modificada o eliminada de los ordenadores. Para lograr que los documentos sean guardados con la seguridad requerida se propuso la automatización de nuevas funcionalidades que se le agregarán al OpenOffice.org a través del lenguaje de macros OpenOffice.org Basic. Se realizaron las actividades correspondientes a los flujos de trabajo de Modelación de negocio, Especificación de los requisitos de software y Diseño, las cuales permitieron conocer las necesidades reales de los trabajadores de oficina y obtener los artefactos correspondientes.

CAPÍTULO 3. IMPLEMENTACIÓN Y PRUEBA

3.1 Introducción

Una vez obtenidos los resultados del diseño detallado se comienza con la implementación, cuyo objetivo principal es desarrollar la arquitectura y el sistema como un todo. Concluido este flujo se procede a probar las funcionalidades. Las pruebas constituyen una etapa imprescindible durante el proceso de desarrollo del *software*. A través de estas se pueden detectar y corregir el máximo de errores posibles antes de que el sistema desarrollado sea entregado al cliente. De esta forma se asegura que el *software* cumpla con las especificaciones requeridas y que sean eliminados los posibles defectos que este pudiera tener. Esta seguridad no es absoluta, debido a que no se garantiza que el sistema esté libre de errores, sino que se detecten la mayor cantidad posibles para su debida corrección.

3.2 Representación de las macros implementadas

Para un mejor entendimiento del código se utilizaron diagramas para describir la secuencia lógica de las subrutinas y funciones implementadas. Las funcionalidades que deben ser sustituidas por los métodos de encriptación utilizados en las FAR son:

Para encriptar texto:

- EncConfMaxima(texto, contrasenna)
- EncConfMedia (texto, contrasenna)
- EncConfMinima (texto, contrasenna)
- DesConfMaxima(texto, contrasenna)
- DesConfMedia (texto, contrasenna)
- DesConfMinima (texto, contrasenna)

Para desencriptar texto:

- DesConfMaxima(texto, contrasenna)
- DesConfMedia (texto, contrasenna)
- DesConfMinima (texto, contrasenna)

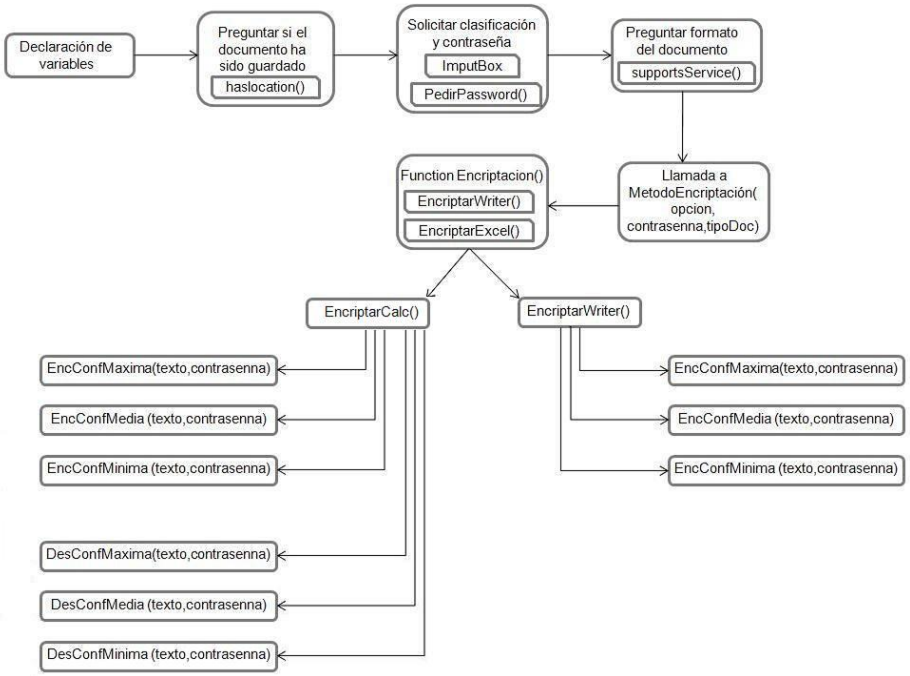

Figura 22. Diagrama de macros implementadas para encriptar texto.

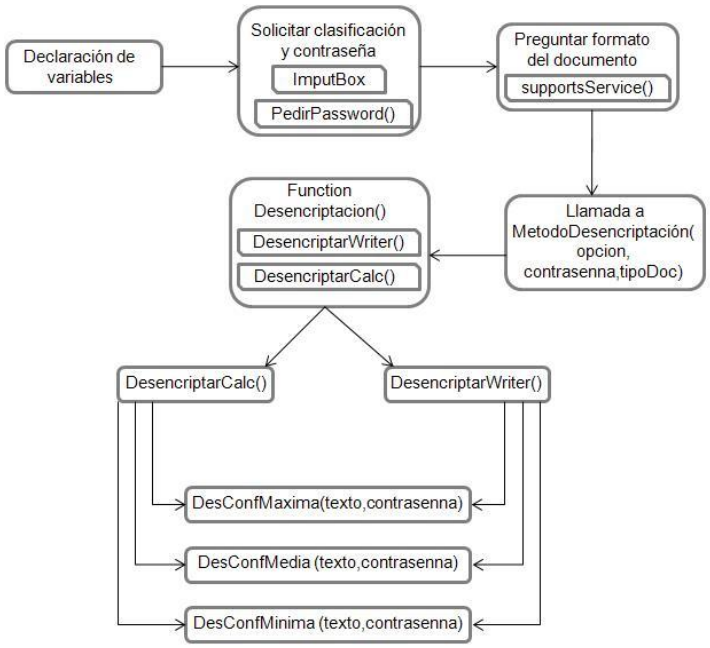

Figura 23. Diagrama de macros implementadas para desencriptar texto.

3.3 Pruebas de la aplicación

Entre los principales métodos de prueba se encuentran las pruebas de Caja Blanca y de Caja Negra. Las primeras se centran en la revisión del código fuente del *software* mientras que las segundas se refieren a las pruebas que se realizan a la interfaz del sistema basándose en los requerimientos funcionales del mismo. Los diseños de casos de prueba son una técnica para probar el sistema mediante el método de Caja Negra y su objetivo fundamental es ofrecer al usuario una guía detallada de cómo realizar la prueba, mostrando los pasos a seguir y los datos a introducir. Estos comprueban que el sistema desarrollado realiza las funciones para las que ha sido creado en base a los requerimientos del usuario. A continuación se describe el diseño de casos de prueba de la aplicación.

3.3.1 Descripción del diseño de casos de prueba

Condiciones de ejecución

1 El documento no debe estar vacío.

Tabla 3.1. Diseño de casos de prueba Primera iteración.

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
1. Encriptar texto	Se cifra el texto del documento actual una vez que es guardado en el disco duro de la computadora.	**EP 1.1** Se selecciona la opción **Guardar como** y se introducen los datos correctamente.	-Se selecciona la opción Guardar como. -La aplicación muestra una ventana para escoger un lugar en el equipo. -Se escoge el lugar en el equipo. -Se oprime el botón Aceptar. -La aplicación muestra un submenú con los tipos de clasificación de la información. -Se escribe el tipo de clasificación de la información para encriptar el texto del documento. -Se oprime el botón Aceptar. -La aplicación muestra una ventana

			solicitando la clave para encriptar. -Se introduce la clave. -Se oprime el botón Aceptar. -La aplicación encripta el texto del documento en el disco duro y lo mantiene abierto en un formato legible.
		EP 1.2 Se selecciona la opción **Guardar como** y se introduce una clasificación incorrecta.	-Se selecciona la opción Guardar como. -La aplicación muestra una ventana para escoger un lugar en el equipo. -Se escoge el lugar en el equipo. -Se oprime el botón Aceptar. -La aplicación muestra un submenú con los tipos de clasificación de la información. -Se deja el campo de clasificación vacío o se introduce una opción distinta. -Se oprime el botón Aceptar. -La aplicación muestra un mensaje de error y solicita nuevamente la opción deseada.
		EP 1.3 Se selecciona la opción **Guardar como** y se introduce una clave incorrecta.	-Se selecciona la opción Guardar como. -La aplicación muestra una ventana para escoger un lugar en el equipo. -Se escoge el lugar en el equipo. -Se oprime el botón Aceptar. -La aplicación muestra un submenú con los tipos de clasificación de la

			información. -Se escribe el tipo de clasificación de la información para encriptar el texto del documento. -Se oprime el botón Aceptar. -La aplicación muestra una ventana solicitando la clave para encriptar. -Se deja el campo clave vacío o es menor de 6 caracteres. -Se oprime el botón Aceptar. -La aplicación muestra un mensaje de error y solicita nuevamente la clave.
		EP 1.4 Se selecciona la opción **Guardar como** y se oprime el botón Cancelar.	-Se selecciona la opción Guardar como. -La aplicación muestra una ventana para escoger un lugar en el equipo. -Se escoge el lugar en el equipo. -Se oprime el botón Aceptar. -La aplicación muestra un submenú con los tipos de clasificación de la información. -Se oprime el botón Cancelar. -La aplicación muestra un mensaje de error y solicita nuevamente la clasificación.
		EP 1.5 Se selecciona la opción **Guardar** y el documento no ha sido guardado.	-Se selecciona la opción Guardar. -La aplicación muestra un mensaje de advertencia solicitando al usuario que debe usar la opción Guardar Como.

		EP 1.6 Se selecciona la opción **Guardar** y el documento ha sido guardado.	-Se selecciona la opción Guardar. -La aplicación encripta el texto del documento en el disco duro y lo mantiene en un formato legible.
		EP 1.7 Se selecciona la opción **Cerrar** y se introducen los datos correctamente.	-Se selecciona la opción Cerrar. -La aplicación muestra un submenú con los tipos de clasificación de la información. -Se escribe el tipo de clasificación de la información para encriptar el texto del documento. -Se oprime el botón Aceptar. -La aplicación muestra una ventana solicitando la clave para encriptar. -Se introduce la clave. -Se oprime el botón Aceptar. -La aplicación encripta el texto del documento en el disco duro y lo cierra.
		EP 1.8 Se selecciona la opción **Cerrar** y se introduce una clasificación incorrecta.	-Se selecciona la opción Cerrar. -La aplicación muestra un submenú con los tipos de clasificación de la información. -Se deja el campo de clasificación vacío o se introduce una opción distinta. -Se oprime el botón Aceptar. -La aplicación muestra un mensaje de error y solicita nuevamente la opción deseada.

		EP 1.9 Se selecciona la opción **Cerrar** y se introduce una clave incorrecta.	-Se selecciona la opción Cerrar. -La aplicación muestra un submenú con los tipos de clasificación de la información. -Se escribe el tipo de clasificación de la información para encriptar el texto del documento. -Se oprime el botón Aceptar. -La aplicación muestra una ventana solicitando la clave para encriptar. -Se deja el campo clave vacío o es menor de 6 caracteres. -Se oprime el botón Aceptar. -La aplicación muestra un mensaje de error y solicita nuevamente la clave.
		EP 1.10 Se selecciona la opción **Cerrar** y se oprime el botón Cancelar.	-Se selecciona la opción Cerrar. --La aplicación muestra un submenú con los tipos de clasificación de la información. -Se deja el campo de clasificación de la información vacío o se introduce una opción distinta. -Se oprime el botón Cancelar. -La aplicación muestra un mensaje de error y solicita nuevamente la clasificación.

2. Desencriptar texto	Se descifra el texto del documento actual.	**EP 2.1** Se procede a abrir el documento y se introducen los datos correctamente.	-Se procede a abrir el documento. -La aplicación muestra un submenú con los tipos de clasificación de la información. -Se escribe el tipo de clasificación de la información para encriptar el texto del documento. -Se oprime el botón Aceptar. -La aplicación muestra una ventana solicitando la clave para encriptar. -Se introduce la clave. -Se oprime el botón Aceptar. -La aplicación desencripta el texto del documento y lo mantiene abierto en un formato legible.
		EP 2.2 Se procede a abrir el documento y se introduce una clasificación incorrecta.	Se procede a abrir el documento. -La aplicación muestra un submenú con los tipos de clasificación de la información. -Se deja el campo de clasificación de la información vacío o se introduce una opción distinta. -Se oprime el botón Aceptar. -La aplicación muestra un mensaje de error y solicita nuevamente la opción deseada. El texto permanece encriptado.

		EP 2.3 Se procede a abrir el documento y se introduce una clave incorrecta.	-Se procede a abrir el documento. -La aplicación muestra un submenú con los tipos de clasificación de la información. -Se escribe el tipo de clasificación de la información para encriptar el texto del documento. -Se oprime el botón Aceptar. -La aplicación muestra una ventana solicitando la clave para desencriptar. -Se deja el campo clave vacío o es menor de 6 caracteres. -Se oprime el botón Aceptar. El documento queda abierto de forma encriptada.
		EP 2.4 Se procede a abrir el documento y se oprime el botón Cancelar.	-Se procede a abrir el documento. -La aplicación muestra un submenú con los tipos de clasificación de la información. -Se escribe la clasificación correcta o incorrectamente. -Se oprime el botón Cancelar. -La aplicación muestra un mensaje de error y solicita nuevamente la opción deseada.

3.3.2 Descripción de variables

Tabla 3.2. Descripción de variables Primera iteración

No	Nombre de campo	Clasificación	Puede ser nulo	Descripción
1	Clasificación	InputBox	NO	El campo sólo acepta números.
2	Contraseña	Odialog	NO	El campo acepta cualquier carácter.

3.3.3 Juego de datos a probar

Tabla 3.3. Juego de datos a probar Primera iteración

Id del escenario	Escenario	Clasificación	Contraseña	Respuesta del sistema	Resultado de la prueba
EP 1.1	Se selecciona la opción **Guardar como** y se entran los datos correctamente.	V (Confidencialidad Máxima (1), Confidencialidad Media (2), Confidencialidad Mínima (3)).	V (******)	El documento es encriptado en el disco duro y se mantiene abierto.	Se obtuvo el resultado adecuado.
EP 1.2	Se selecciona la opción **Guardar como** y se entra una clasificación incorrecta.	I (número distinto de 1, 2, 3, o dejar el campo vacío).	V (******)	La aplicación muestra un mensaje de error informando de que debe seleccionar un número dentro del rango y solicita nuevamente la opción deseada.	Se obtuvo el resultado adecuado.

EP 1.3	Se selecciona la opción **Guardar como** y se entra una clave incorrecta.	V (Confidencialidad Máxima (1), Confidencialidad Media (2), Confidencialidad Mínima (3)).	I (campo vacío o menor de 6 caracteres)	La aplicación muestra un mensaje de error informando de que la contraseña debe tener al menos 6 caracteres y solicita nuevamente la contraseña deseada.	Se obtuvo el resultado adecuado.
EP 1.4	Se selecciona la opción **Guardar como** y se oprime el botón Cancelar.	NA	NA	La aplicación muestra un mensaje de error y solicita nuevamente la clasificación.	Se obtuvo el resultado adecuado.
EP 1.5	Se selecciona la opción **Guardar** y el documento no ha sido guardado.	NA	NA	La aplicación muestra un mensaje de advertencia solicitando al usuario que debe usar la opción Guardar Como.	Se obtuvo el resultado adecuado.
EP 1.6	Se selecciona la opción **Guardar** y el documento ha sido guardado.	NA	NA	Solicita opción y contraseña para encriptar el texto del documento.	Se obtuvo una no conformidad.
EP 1.7	Se selecciona la opción **Cerrar** y se entran los datos correctamente.	V (Confidencialidad Máxima (1), Confidencialidad Media (2), Confidencialidad Mínima (3)).	V (******)	El documento es encriptado en el disco duro y se cierra.	Se obtuvo una no conformidad.

EP 1.8	Se selecciona la opción **Cerrar** y se entra una clasificación incorrecta.	I (número distinto de 1, 2, 3, o dejar el campo vacío).	V (******)	La aplicación muestra un mensaje de error informando de que debe seleccionar un número dentro del rango y solicita nuevamente la opción deseada.	Se obtuvo el resultado adecuado.
EP 1.9	Se selecciona la opción **Cerrar** y se deja el campo clave vacío o se introduce una clave incorrecta.	V (Confidencialidad Máxima (1), Confidencialidad Media (2), Confidencialidad Mínima (3)).	I (campo vacío o menor de 6 caracteres)	La aplicación muestra un mensaje de error informando de que la contraseña debe tener al menos 6 caracteres y solicita nuevamente la contraseña deseada.	Se obtuvo el resultado adecuado.
EP 1.10	Se selecciona la opción **Cerrar** y se oprime el botón Cancelar.	NA	NA	La aplicación muestra un mensaje de error y solicita nuevamente la clasificación.	Se obtuvo el resultado adecuado.
EP 2.1	Se procede a abrir el documento y se entran los datos correctamente.	V (Confidencialidad Máxima (1), Confidencialidad Media (2), Confidencialidad Mínima (3)).	V (******)	El documento es desencriptado en el disco duro y se mantiene abierto en un formato legible.	Se obtuvo el resultado adecuado.

EP 2.2	Se procede a abrir el documento y se entra una clasificación incorrecta.	I (número distinto de 1, 2, 3, o dejar el campo vacío).	V (******)	La aplicación muestra un mensaje de error informando de que debe seleccionar un número dentro del rango y solicita nuevamente la opción deseada.	Se obtuvo el resultado adecuado.
EP 2.3	Se procede a abrir el documento y se deja el campo clave vacío o se introduce una contraseña incorrecta.	V (Confidencialidad Máxima (1), Confidencialidad Media (2), Confidencialidad Mínima (3)).	I (campo vacío)	El texto permanece encriptado.	Se obtuvo el resultado adecuado.
EP 2.4	Se procede a abrir el documento y se oprime el botón Cancelar.	NA	NA	La aplicación muestra un mensaje de error y solicita nuevamente la clasificación.	Se obtuvo el resultado adecuado.

3.3.4 Registro de no conformidades

Una vez realizadas las pruebas y detectadas las no conformidades en la aplicación se hizo necesaria la respuesta por parte de los desarrolladores a las mismas. Se corrigieron los defectos y luego se realizaron las pruebas nuevamente en una segunda iteración para asegurar que los programadores hayan solucionado los problemas encontrados.

Tabla 3.4. No conformidades

Elemento	Etapa de detección	No	No conformidad	Aspecto correspondiente	Tipo	Sig	No Sig	Recomendación
EP 1.6	Validación	1	La aplicación pide clasificación y contraseña cada vez que se oprime el botón Guardar, aunque el documento ya haya sido guardado.	**Tabla** Diseño de casos de prueba Primera iteración. **RF**: Encriptar Texto, **EP** 1.6	Interfaz	X		La aplicación debe pedir clasificación y contraseña solo la primera vez que se desee guardar.
EP 1.7	Validación	2	Cuando se oprime el botón Cerrar, y ya el documento ha sido guardado, la aplicación vuelve a pedir los datos para encriptarlo nuevamente.	**Tabla** Diseño de casos de prueba Primera iteración. **RF**: Encriptar Texto, **EP** 1.7	Interfaz	X		La aplicación debe pedir clasificación y contraseña solamente si el documento no ha sido guardado.

Tabla 3.5. Diseño de casos de prueba Segunda iteración

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
1. Encriptar texto	Se cifra el texto del documento actual una vez que es guardado en el disco duro de la computadora.	**EP 1.6** Se selecciona la opción **Guardar** y el documento ha sido guardado.	-Se selecciona la opción Guardar. -La aplicación encripta el texto del documento en el disco duro y lo mantiene en un formato legible.

		EP 1.7 Se selecciona la opción **Cerrar** y se introducen los datos correctamente.	-Se selecciona la opción Cerrar. -La aplicación muestra un submenú con los tipos de clasificación de la información. -Se escribe el tipo de clasificación de la información para encriptar el texto del documento. -Se oprime el botón Aceptar. -La aplicación muestra una ventana solicitando la clave para encriptar. -Se introduce la clave. -Se oprime el botón Aceptar. -Se encripta el texto del documento y se cierra.

Tabla 3.6. Descripción de variables Segunda iteración

No	Nombre de campo	Clasificación	Puede ser nulo	Descripción
1	Clasificación	InputBox	NO	El campo sólo acepta números.
2	Contraseña	Odialog	NO	El campo acepta cualquier carácter.

Tabla 3.7. Juego de datos a probar Segunda iteración

Id del escenario	Escenario	Clasificación	Contraseña	Respuesta del sistema	Resultado de la prueba
EP 1.6	Se selecciona la opción **Guardar** y el documento ha sido guardado.	NA	NA	Solicita opción y contraseña para encriptar el texto del documento.	Se obtuvo el resultado adecuado.
EP 1.7	Se selecciona la opción **Cerrar** y se entran los datos correctamente.	V (Confidencialidad Máxima (1), Confidencialidad Media (2), Confidencialidad Mínima (3)).	V (******)	El documento es encriptado en el disco duro y se cierra.	Se obtuvo el resultado adecuado.

3.4 Resultados de las pruebas

Para determinar si existen errores en la aplicación se realizó el diseño de casos de prueba teniendo en cuenta una serie de datos a probar para cada escenario. En la primera iteración realizada, las pruebas arrojaron como resultado dos inconformidades que fueron registradas para que los desarrolladores pudiesen corregirlas. Estos errores fueron eliminados para la segunda iteración. Como resultado general de las pruebas se obtuvo que el texto del documento queda cifrado una vez que se usaba la opción Guardar o Cerrar y en el caso de abrir, solo se desencripta si se conoce la clave con la cual fue cifrado. En caso contrario el documento queda abierto pero en un formato ilegible que será inentendible para los intrusos. La aplicación fortalece la confidencialidad, disponibilidad e integridad que requieren los oficiales y trabajadores civiles de nuestras Fuerzas Armadas Revolucionarias.

3.5 Conclusiones

Durante el desarrollo de este capítulo se realizaron las actividades de implementación y prueba de los requisitos funcionales propuestos para la suite OpenOffice.org. Se describieron además los casos de pruebas, teniendo en cuenta una serie de datos a probar para abarcar la mayor cantidad de opciones que

pueda usar el usuario y encontrar los posibles errores. Las mismas arrojaron como resultado que las modificaciones realizadas a la aplicación fortalecen la seguridad de los documentos puesto que una vez que son guardados en el disco duro de la computadora el texto de los mismos queda encriptado.

CONCLUSIONES GENERALES

Luego de realizar un análisis acerca de la problemática presente en las oficinas de las FAR en cuanto a la seguridad con que son guardados los documentos elaborados en el Procesador de textos y la Hoja de cálculos, se determinó añadir a las funcionalidades de Guardar y Abrir de los mismos, las macros necesarias para lograr que el contenido quede en un formato ilegible al ser guardados, y que al abrirlos se solicite una contraseña para desencriptarlos. Al terminar la confección del presente trabajo de diploma se arribó a las siguientes conclusiones:

- Se investigó sobre el trabajo ofimático realizado en las entidades de las FAR y se estudiaron temas relacionados con la criptografía y las macros del OpenOffice.org.
- Se modeló la solución propuesta obteniéndose los artefactos correspondientes y se implementaron las funciones de encriptar y desencriptar objetos de tipo texto en los documentos.
- Se validó la implementación realizada mediante el diseño de casos de prueba para verificar el correcto funcionamiento de los requisitos funcionales.

De manera general, en el documento se recoge todo el trabajo realizado para la elaboración de dichas funcionalidades, se aborda sobre las actividades ofimáticas realizadas en la actualidad y aspectos relacionados con la criptografía. Se definen además las características de estas funcionalidades, así como la modelación, diseño, implementación y pruebas de la aplicación, y se obtuvo como resultado un producto informático que fortalecerá la seguridad con que son guardados los documentos utilizados en las oficinas de las FAR.

RECOMENDACIONES

A consecuencia del proceso de investigación y realización de las funcionalidades, han surgido recomendaciones a tener en cuenta para continuar perfeccionando la suite OpenOffice.org, estas son las siguientes:

- Añadir los algoritmos de encriptación al resto de los objetos de los documentos.
- Añadir las funcionalidades de Guardar y Abrir documentos utilizando algoritmos de encriptación al resto de los programas de la suite.
- Brindar la posibilidad al usuario de que los documentos puedan ser enviados de forma encriptada.
- Añadir a las funcionalidades de Guardar y Abrir los métodos de encriptación necesarios a través de un Plugin.

BIBLIOGRAFÍA REFERENCIADA

1. *Protección contra los recursos informáticos en Cuba.* **Arregoitía.** 4, La Habana: Revista “Giga”, 2002.

2. **AGUIRRE, JORGE RAMIÓ.** *Libro Electrónico de Seguridad Informática y Criptografía.* España: Madrid: Universidad de LLeida, 2006.

3. Kioskea.net. *Introducción al cifrado mediante DES.* [En línea] 16 de octubre de 2008. [Citado el: 20 de enero de 2010.] http://es.kioskea.net/contents/crypto/des.php3.

4. Bitzipper. *Encriptación AES - seguridad de datos.* [En línea] Bitberry Software, 2000. [Citado el: 20 de enero de 2010.] http://www.bitzipper.com/es/aes-encryption.html.

5. Scribd. *Metodologías-de-desarrollo-software.* [En línea] 2 de mayo de 2008. [Citado el: 25 de enero de 2010.] http://www.scribd.com/doc/2050925/metodologías-de-desarrollo-software.

6. **UCID.** *Proceso de Desarrollo y Gestión de Proyectos de Software.* s.l.: Ciudad Habana, 2009. Primera versión.

7. Sitio de descargas de software. *Visual Paradigm for UML (ME).* [En línea] 5 de marzo de 2007. [Citado el: 26 de enero de 2010.] http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%28M%C3%8D%29_14720_p/.

8. **LUCIANO.** LuAuf.com. *Entornos de Desarrollo Integrado para Java.* [En línea] 13 de marzo de 2008. [Citado el: 26 de enero 2010.] http://luauf.com/2008/05/13/entornos-de-desarrollo-integrado-para-java/.

9. **REARTE, EMILIO.** ABCdatos.com . *Base de Datos y UML.* [En línea] 18 de noviembre de 2009. [Citado el: 26 de enero de 2010.] http://www.abcdatos.com/tutoriales/tutorial/l7157.html.

BIBLIOGRAFÍA CONSULTADA

1. **PÉREZ, ALEXEIS GARCÍA.** SciELO - Scientific Electronic Library Online. *La gestión de documentos electrónicos como respuesta a las nuevas condiciones del entorno de información.* [En línea] 2001. [Citado el: 1 de diciembre de 2009.] http://scielo.sld.cu/pdf/aci/v9n3/aci03301.pdf

2. MSINFO-Sistemas de Información. *Documentos Digitales.* [En línea] 1987. [Citado el: 1 de diciembre de 2009.] http://www.msinfo.info/propuestas/documentos/documentos_digitales.html

3. **VEGA, MARISOL MARTÍNEZ.** Slideshare. *Introducción a la administración de documentos y su importancia.* [En línea] enero de 2008. [Citado el: 1 de diciembre de 2009.] http://www.slideshare.net/marisolmartinezvega/introduccion-a-la-administracion-de-documentos-y-su-importancia

4. Documentos Electrónicos Digitales. *Servicios:Digitalización.* [En línea] 2004. [Citado el: 1 de diciembre de 2009.] http://www.documentoselectronicos.com/serv_digit.htm

5. **CLARA BAONZA**. Electrónica y Comunicaciones. *Integridad y Confidencialidad de la Información.* [En línea] 2005. [Citado el: 1 de diciembre de 2009.] http://www.cypsela.es/especiales/pdf206/confidencialidad.pdf

6. **CARLOS LÓPEZ.** GestioPolis. *La seguridad informática y el control interno en Cuba. Experiencias de la división Copextel Villa Clara.* [En línea] Bogotá D.C, 2008. [Citado el: 1 de diciembre de 2009.] http://www.gestiopolis.com/administracion-estrategia/seguridad-informatica-y-su-control.htm

7. **MARTÍNEZ, MAGISTER DAVID LUIS LA RED.** Universidad Nacional del Nordeste. Argentina. *Seguridad en Linux.* [En línea] abril de 2009. [Citado el: 1 de diciembre de 2009.] http://exa.unne.edu.ar/depar/areas/informatica/SistemasOperativos/MonogSO/SEGLIN00.html

8. **PERDOMO, MARÍA ELECTA TORRES.** saber.ula.ve. *La escritura y su importancia en la construcción del conocimiento.* [En línea] 18 de enero de 2002. [Citado el: 25 de enero de 2010.] http://www.saber.ula.ve/bitstream/123456789/17528/2/maria_torres.pdf.

9. **AGUIRRE, JORGE RAMIÓ.** *Libro Electrónico de Seguridad Informática y Criptografía.* Madrid: Universidad Politécnica de Madrid, España, 2006. 84-86451-69-8 (2006).

10. WordReference.com. WordReference.com. *Diccionario de la lengua española.* [En línea] 2010. [Citado el: 10 de febrero de 2010.] http://www.wordreference.com/definicion/Cantidad%20expresada%20por%20medio%20de%20dos%20o%20m%C3%A1s%20cifras.

11. **PÉREZ, JOSÉ ANGEL DE BUSTOS.** TLDP-ES/LuCAS: servicios editoriales para la documentación libre en español. *Criptografía.* [En línea] 20 de mayo de 2003. [Citado el: 20 de enero de 2010.] http://es.tldp.org/Presentaciones/200203jornadassalamanca/jadebustos/conferencia-criptografia.pdf

12. Instituto de Investigación Tecnológica. *El algoritmo criptográfico AES para protección de Datos.* [En línea] Septiembre de 2007. [Citado el: 20 de enero de 2010.] http://www.iit.upcomillas.es/pfc/resumenes/46ea7511774d8.pdf

13. **ARRIAZU, JORGE SÁNCHEZ.** tierradelazaro.com. *Descripción del algoritmo DES.* [En línea] diciembre de 1999. [Citado el: 20 de enero de 2010.] http://www.tierradelazaro.com/public/libros/des.pdf

14. **VERA DELGADO, RAFAEL PALACIOS.** Asociación/Colegio Nacional de Ingerieros del ICAI. *Introducción a la Criptografía: tipos de algoritmos.* [En línea] enero-febrero de 2006. [Citado el: 20 de enero de 2010.] https://www.icai.es/publicaciones/anales_get.php?id=1210

15. EMVI:Enciclopedia Multimedia Interactiva y Biblioteca Virtual de las Ciencias Sociales, Económicas y Jurídicas. *Funciones Resumen (HASH).* [En línea] Universidad de Málaga, 2010. [Citado el: 20 de enero de 2010.] http://www.eumed.net/cursecon/ecoinet/seguridad/resumenes

16. Programas Grastis.net. *Steganos LockNote 1.0.4.* [En línea] 12 de marzo de 2008. [Citado el: 21 de enero de 2010.] http://steganos-locknote.programas-gratis.net/

17. **KELLMAN, JANET.** rbytes reviews. *Steganos LockNote v1.0.3 Descargar.* [En línea] 2004-2010. [Citado el: 21 de enero de 2010.] http://rbytes.net/descargar/steganos-locknote-descargar

18. SOFTONIC. *CryptoForge 3.3.0.* [En línea] Equipo de Softonic, 2 de marzo de 2010. [Citado el: 21 de marzo de 2010.] http://cryptoforge.softonic.com

19. **JOSÉ DOMÍNGUEZ, LUIS HERNÁNDEZ.** Uptodown.com. *CryptoForge 3.2.5.* [En línea] Media Ingea S.L, 4 de marzo de 2008. [Citado el: 21 de enero de 2010.] http://cryptoforge.uptodown.com

20. PortalProgramas. *CryptoForge.* [En línea] Redaccenir, S.L, 01 de Noviembre de 2005. [Citado el: 21 de enero de 2010.] http://gratis.portalprogramas.com/CryptoForge.html.

21. **JOSÉ DOMÍNGUEZ, LUIS HERNÁNDEZ.** Uptodown.com. *Codenigma 3.0.* [En línea] Media Ingea S.L, 29 de octubre de 2007. [Citado el: 21 de enero de 2010.] http://codenigma.uptodown.com

22. ABCdatos.com. *Codenigma v3.0.* [En línea] 16 de enero de 2007. [Citado el: 21 de enero de 2010.] http://www.abcdatos.com/programas/programa/z3738.html.

23. PortalProgramas. *Codenigma.* [En línea] Redaccenir, S.L, 14 de Enero de 2007. [Citado el: 21 de enero de 2010.] http://gratis.portalprogramas.com/CODENIGMA.html.

24. **LÓPEZ, JOSÉ MARÍA.** SOFTONIC. *Unidades virtuales cifradas para proteger archivos confidenciales.* [En línea] 26 de noviembre de 2009. [Citado el: 21 de enero de 2010.] http://truecrypt.softonic.com/.

25. Visual Beta. *TrueCrypt 6.3, ahora con soporte para Windows 7.* [En línea] 23 de octubre de 2009. [Citado el: 21 de enero de 2010.] http://www.visualbeta.es/13859/software/truecrypt-63-ahora-con-soporte-para-windows-7/.

26. Kriptópolis. *PDFTK: Manejo de PDFs con herramientas libres.* [En línea] 1 de agosto de 2006. [Citado el: 21 de enero de 2010.] http://www.kriptopolis.org/pdftk-manejo-de-pdfs-con-herramientas-libres.

27. PROGRAMAS-GRATIS.NET. *PDFTK.* [En línea] 16 de abril de 2009. [Citado el: 21 de enero de 2010.] http://pdftk.programas-gratis.net/.

28. **SOGO, JOSE CARLOS GARCÍA.** Debian. *Seahorse.* [En línea] 2010. [Citado el: 21 de enero de 2010.] http://packages.debian.org/es/lenny/seahorse.

29. **SOGO, JOSE CARLOS GARCÍA.** Ubuntu. *seahorse.* [En línea] 2010. [Citado el: 21 de enero de 2010.] http://packages.ubuntu.com/es/jaunty/seahorse.

30. **JOSÉ H. CANÓS, PATRICIO LETELIER, Mª CARMEN PENADÉS.** WillyDev. *Métodologías Ágiles en el Desarrollo de Software.* [En línea] 2 de mayo de 2008. [Citado el: 25 de enero de 2010.] http://www.willydev.net/descargas/prev/TodoAgil.pdf.

31. **BECK, M. BEEDLE.** Instituto de Investigación de Tecnología Educativa de la UNITEC. *Manifiesto ágil.* [En línea] 2006. [Citado el: 25 de enero de 2010.] http://www.cenitec.com.mx/Manifiesto.pdf.

32. **PADRÓN, JASMÍN FUENTES.** Biblioteca UCI. *Propuesta de marco de Fábricas de Software basado en las buenas prácticas de Métodos Agiles.* [En línea] junio de 2008. [Citado el: 25 de enero de 2010.] http://bibliodoc.uci.cu/TD/TD_1355_08.pdf.

33. **ACUÑA, KARENNY BRITO.** enumed.net. *Selección de Metodologías de Desarrollo para Aplicaciones Web en la facultad de Informática de la Universidad de Cienfuegos.* [En línea] 2009. [Citado el: 25 de enero de 2010.] http://www.eumed.net/libros/2009c/584/index.htm.

34. El prisma. *Ingeniería de sistemas.* [En línea] 2005. [Citado el: 25 de enero de 2010.] http://www.elprisma.com/apuntes/curso.asp?id=13324.

35. **PEREYRA, MARTIN TUESTA.** El prisma. *Herramientas CASE.* [En línea] 2005. [Citado el: 26 de enero de 2010.] HYPERLINK "http://unaduni.wikispaces.com/file/view/Herramientascase.doc" http://unaduni.wikispaces.com/file/view/Herramientascase.doc.

36. Técnica Administrativa ejournal. Ciencia y Técnica Administrativa. Herramientas CASE. [En línea] Oficina en Buenos Aires, 2008. [Citado el: 26 de enero de 2010.] http://www.cyta.com.ar/biblioteca/bddoc/bdlibros/proyectoinformatico/libro/c5/c5.htm.

37. Scribd. *Capitulo I Herramientas CASE.* [En línea] 23 de mayo de 2008. [Citado el: 26 de enero de 2010.] http://www.scribd.com/doc/3062020/Capitulo-I-HERRAMIENTAS-CASE**}**.

38. Equipo Danysoft. Danysoft. *Modelado de bases de datos.* [En línea] 26 de septiembre de 2006. [Citado el: 26 de enero de 2010.] http://www.danysoft.info/free/model2.pdf.

39. Sitio de descargas de software. *Visual Paradigm for UML (ME).* [En línea] 5 de marzo de 2007. [Citado el: 26 de enero de 2010.] http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%28M%C3%8D%29_14720_p/.

40. **PABLO VALDESPINO GONZÁLEZ, JOSÉ RAÚL SEOANE RODRÍGUEZ.** *Desarrollo de un Marco de Trabajo para Aplicaciones Web Empresariales de las FAR.* Ciudad de La Habana: s.n, 2008.

41. **HDEZ, DENNYS J.** *Sistema Automatizado cubano para el control de equipos médicos.* La Habana, 2009: Universidad de Ciencias Informáticas.

42. **RASMUSSEN, ANDERS INGEMAN.** Osalt.com. Código abierto como alternativa*. IBM Rational Rose.* [En línea] 2009. [Citado el: 26 de enero de 2010.] http://www.osalt.com/es/acerca-de.

43. **SlideShare Inc.** SlideShare. *Rational Rose.* [En línea] 2009. [Citado el: 26 de enero de 2010.] http://www.slideshare.net/vivi_jocadi/rational-rose.

44. **LORNEL A. RIVAS, MARÍA PÉREZ, LUIS E. MENDOZA, ANNA GRIMÁN.** Laboratorio de Investigación en Sistemas de Información. *Herramientas de Desarrollo de Software.* [En línea] 2007. [Citado el: 26 de enero de 2010.] http://www.lisi.usb.ve/publicaciones/05%20herramientas/herramientas_25.pdf.

45. **DANIELE, MARCELA.** Facultad de Ciencias Exactas, Físico-Químicas y Naturales. *Análisis y Diseño de Sistemas.* [En línea] 2007. [Citado el: 26 de enero de 2010.] http://dc.exa.unrc.edu.ar/nuevodc/materias/sistemas/2007/TEORICOS/TEORIA_1_Introduccion_AyDS2007.pdf.

46. **TORRES, JOSÉ ERNESTO.** Entorno Virtual de Aprendizaje. *Curso de Python.* [En línea] 2010. [Citado el: 27 de enero de 2010.] http://eva.uci.cu.

47. **DUQUE, RAÚL GONZÁLEZ** *Python para todos.* España: s.n.

48. **LOLA CÁRDENAS, JOAQUIN GRACIA.** WebEstilo. *Dónde y cómo incluir JavaScriptSiguiente página.* [En línea] enero de 2003. [Citado el: 27 de enero de 2010.] http://www.webestilo.com/javascript/js00.phtml.

49. **PÉREZ, JAVIER EGUÍLUZ.** librosweb.es. *Introducción a JavaScript.* [En línea] 2010. [Citado el: 27 de enero de 2010.] http://librosweb.es/javascript/.

50. **ALVAREZ, MIGUEL ANGEL.** Desarrolloweb.com. *Qué es JavaScript y las posibilidades que nos ofrece con respecto al HTML.* [En línea] 16 de julio de 2001. [Citado el: 27 de enero de 2010.] http://www.desarrolloweb.com/articulos/490.php.

51. Infierno Hacker. *Visual Basic.* [En línea] Simple Machines LLC, 10 de Diciembre de 2007. [Citado el: 27 de enero de 2010.] http://foro.infiernohacker.com/index.php?action=printpage;topic=1176.0.

52. Grupo de Usuarios de Software Libre de Córdoba**.** GrULiC. *¿Qué es GNU/Linux?* [En línea] 16 de abril de 2010. [Citado el: 25 de abril de 2010.] http://www.grulic.org.ar/node/10.

53. **GARRO, ARTURO.** OpenOffice.org. *Como hacer Macros en OOo.* [En línea] 5 de Agosto de 2003. [Citado el: 10 de febrero de 2010.] http://es.openoffice.org/servlets/ProjectDocumentView?documentID=1223&showInfo=true.

54. **NOELSON ALVES DUARTE, ISMAEL FANLO.** OpenOffice.org. *Introducción a OpenOffice.org Basic.* [En línea] 23 de agosto de 2003. [Citado el: 10 de febrero de 2010.] http://es.openoffice.org/servlets/ProjectDocumentView?documentID=1249&showInfo=true.

55. **RAMOS, ÁLVARO E. PRIETO.** superalumnos.net. *Tutorial deOpenOffice.org Base.* [En línea] 22 de Noviembre de 2007. [Citado el: 10 de febrero de 2010.] http://superalumnos.net/docs/tutorialOOoBase.pdf.

56. Oficina de Software Libre. *OpenOffice.org.* [En línea] Universidad de Zaragoza, 2010. [Citado el: 10 de febrero de 2010.] http://osluz.unizar.es/aplicacion/openoffice.

57. Microsoft Live. *¿Dónde se guardan las macros?* [En línea] Microsoft Corporation, 23 de febrero de 2006. [Citado el: 14 de mayo de 2010.] http://optimoffice.spaces.live.com/Blog/cns!8F2B7F8F54CC11DA!647.entry.

58. **PROF LAURO SOTO.** MiTecnologico.com. *Diagramas De Transicion.* [En línea] Mexico, 2010. [Citado el: 14 de mayo de 2010.] http://www.mitecnologico.com/Main/DiagramasDeTransicion.

59. Departamento de Sistemas y Computación. *Diagramas de transición de estados.* [En línea] Instituto Tecnológico de La Paz, 20 de febrero de 2006. [Citado el: 14 de mayo de 2010.] http://sistemas.itlp.edu.mx/tutoriales/analisis/43.htm.

60. Universidad Nacional de Colombia. *Analísis y diseño de sistemas de información.* [En línea] Dirección Nacional de Servicios Académicos Virtuales, 2010. [Citado el: 14 de mayo de 2010.] http://www.virtual.unal.edu.co/cursos/sedes/manizales/4060030/lecciones/Capitulo%204/estado.htm

GLOSARIO DE TÉRMINOS

1. **GNU/Linux**: sistema operativo de libre distribución para computadoras personales, servidores, y estaciones de trabajo.
2. **OpenOffice.org**: suite ofimática de *software* libre, distribución gratuita y código abierto con herramientas como procesador de textos, hoja de cálculo, presentaciones, herramientas para el dibujo vectorial y base de datos.
3. **Macros**: conjunto de operaciones que son memorizadas para repetirlas cada vez que se desee activando un botón del OpenOffice.org.
4. **Guarismos**: cantidad expresada por medio de dos o más cifras.
5. **XOR**: expresión física de un operador booleano en la lógica de conmutación.
6. **Blowfish**: codificador de bloques simétricos.
7. **Triple DES**: algoritmo que hace triple cifrado del DES.
8. **Gost**: abreviatura de Gosudarstvenny Standart en español Estándar del estado es un conjunto de estándares internacionales para la Estandartización, Meteorología y Certificación.
9. **GnuPG**: *software* libre alternativo, que permite encriptar y firmar sus datos y la comunicación.
10. **GNOME**: entorno de escritorio e infraestructura de desarrollo para sistemas operativos Unix.
11. **OpenSSH**: conjunto de aplicaciones para realizar comunicaciones cifradas en la red, usando el protocolo SSH.
12. **PGP**: es un programa cuya finalidad es proteger la información distribuida a través de *Internet* mediante el uso de criptografía de clave pública.
13. **Eventos**: acción ejecutada por la suite OpenOffice.og, como son: abrir documento, guardar documento, imprimir documento, entre otros.