Universidad de las Ciencias Informáticas

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

Automatización del proceso de descontaminación de Programas Malignos.

Autor:

Omar Pimentel Alfonso.

Tutor:

Ing. Alberto Arce Martínez.

Ing. Edgar José Guadis Salazar.

Ciudad de la Habana, junio de 2009.

...Su cuerpo encorvado sobre el teclado, denotaba cierto cansancio, pero a pesar de todo tenía ganas de seguir allí, encorvado sobre el teclado, enfrascado en el monitor, en medio de una cadena de datos sin sentido, desplazándose de abajo a arriba a una velocidad de vértigo. Tenía acaso relación con algún tipo de gusano o acaso se trataba de un generador de datos Pseudo aleatorio de encriptado?...

Creo que los virus de computadoras deberían contar como vida. Creo que dicen algo acerca de la naturaleza humana, la única forma de vida que hemos creado hasta el momento es puramente destructiva. Hemos creado vida a nuestra imagen. (1)

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo a la Empresa de Consultoría y Seguridad Informática Segurmática para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmamos la presente a los ____ días del mes de _____ del año _____.

_____

Omar Pimentel Alfonso

Autor

_____ _____

Alberto Arce Martínez Edgar José Guadis Salazar

Tutor Tutor

OPINIÓN DEL USUARIO DEL TRABAJO DE DIPLOMA

El Trabajo de Diploma, titulado Automatización del Proceso de Descontaminación de Programas Malignos. Fue realizado en la Universidad de las Ciencias Informáticas. Esta entidad considera que, en correspondencia con los objetivos trazados, el trabajo realizado le satisface

□ Totalmente

□ Parcialmente en un ____ %

Los resultados de este Trabajo de Diploma le reportan a esta entidad los beneficios siguientes (cuantificar):

_____

_____

_____

_____

_____

_____

Como resultado de la implantación de este trabajo se reportará un efecto económico que asciende a:

Y para que así conste, se firma la presente a los ____ días del mes de _____ del año _____.

_____	_____
Representante de la entidad	Cargo
_____	_____
Firma	Cuño

OPINIÓN DEL TUTOR DEL TRABAJO DE DIPLOMA

Título: Automatización del Proceso de Descontaminación de Programas Malignos.

Autor: Omar Pimentel Alfonso.

Por todo lo anteriormente expresado considero que el estudiante está apto para ejercer como Ingeniero en Ciencias Informáticas; y propongo que se le otorgue al Trabajo de Diploma la calificación de _____.

_____ _____

Alberto Arce Martínez Fecha

_____ _____ Edgar José

Guadis Salazar Fecha

Agradecimientos

Obtenga este agradecimiento desde lo más profundo del alma a mi madre por brindarme su ayuda incondicional y su más sencillo e importante apoyo al servirme de guía y de soporte para luchar por mis sueños.

Mi más profundo agradecimiento a mis tutores Edgar por enseñarme tanto y a Arce por ayudarme con tantas cosas que es imposible mencionarlas además de ser ambos mis amigos gracias por apoyarme en todo momento.

Agradezco también a Raydel, Juan Carlos, Manuel, Yuliesky, Thompson y Anglada por enseñarme tanto, ayudarme siempre que he tenido problemas y por ser amigos míos.

Agradezco de igual forma a toda mi familia.

A todos los que han sido colegas de estudio, por el apoyo ofrecido y por haber tenido la dicha de haberlos conocido y compartir con ellos.

A Lila por ser mi novia mi amiga y consejera de siempre

Las gracias a nuestro comandante en jefe Fidel Castro Ruz por el optimismo depositado en nosotros los estudiantes de la UCI al permitirnos participar en este proyecto y permitirnos formar parte de los profesionales que engrandecen y honran a esta Revolución Socialista Cubana.

En fin a toda nuestra familia y a todos que contribuyeron a la realización de este anhelo.

Muchas Gracias.

A mi Madre, mis Abuelos, mis Hermanos Sarita y Yuri a mi Papa y a toda mi familia por la ayuda y el apoyo que me dieron y por su confianza y por permitir ser el protagonista de este sueño.
A mi novia por estar a mi lado en todo momento y ayudarme a lograr mi anhelo.
A todos aquellos que de una forma u otra me ayudaron.

Omar.

Resumen:

Hoy en día la descontaminación de programas malignos en el mundo se ha convertido en un problema serio para las empresas antivirus, pues la cifra de muestras a analizar diariamente es superior a las 10000. En la práctica sucede, que de manera general, las empresas antivirus no parecen estar haciendo mucho hincapié en este aspecto y como resultado las PC no quedan bien descontaminadas e incluso los sistemas quedan inestables.

En el caso del la Empresa de Consultoría y Seguridad Informática SEGURMATICA, perteneciente al MIC, siempre ha sido un objetivo de primer orden dar solución, con la mayor calidad y eficiencia posible, al número creciente de todas las muestras de códigos malignos reportados nacionalmente a la empresa y a aquellas reportadas internacionalmente como de mayor nivel de propagación o más dañinas, Por lo que se hace necesario la automatización del proceso de análisis que a diario hacen los especialistas con muchas herramientas, tarea muy tediosa pues todas las herramientas tienen un uso distinto y salida en un formato también distinto. (2)

Para darle solución a este problema se ha decidido automatizar este proceso creando una herramienta, que conste de distintos módulos de análisis, capaz de ejecutar las muestras malignas y analizarlas siguiendo el procedimiento normal que haría un especialista en el laboratorio de la empresa pero de forma automática, así como interpretar los resultados brindados en los reportes, generar una descripción de los efectos de la muestra maligna sintética orientada al usuario y evaluar la descontaminación genérica implementada en el producto Segurmática Antivirus.

Índice

OPINIÓN DEL USUARIO DEL TRABAJO DE DIPLOMA ..... **I**

AGRADECIMIENTOS ..... **I**

RESUMEN: III

INTRODUCCIÓN **1**

ESTRUCTURA DEL DOCUMENTO: 2

CAPITULO 1. FUNDAMENTACIÓN TEÓRICA **3**

INTRODUCCIÓN. 3
1.1 CONTAMINACIÓN DE PM EN CUBA. 4
1.2 SISTEMAS DE ANÁLISIS DE PROGRAMAS MALIGNOS EN EL MUNDO. 4
1.2.3 Sistemas de Análisis de Programas Malignos en Segurmática. ***5***
1.3 LENGUAJES Y TECNOLOGÍAS UTILIZADAS. 5
1.4 METODOLOGÍA DE DESARROLLO Y LENGUAJE DE MODELADO. 5
1.4.1 Herramienta de Modelado ***5***
1.4.2 UML ***6***
CONCLUSIONES 8

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA. **9**

INTRODUCCIÓN. 9
2.1 OBJETO DE ESTUDIO 9
2.1.1 Problema y situación problémica: ***9***
2.1.2 Objeto de automatización ***10***
2.2 PROPUESTA DE SISTEMA. 10
2.3 MODELO DEL DOMINIO. 11
FIG 2.1: MODELO DE DOMINIO "ANALIZAR PROGRAMA MALIGNO" 12
2.3.1 Definición de las clases del modelo del dominio. ***13***
2.4 ESPECIFICACIÓN DE LOS REQUISITOS DE SOFTWARE. 16
2.4.1 Requerimientos Funcionales ***16***
2.4.2 Requerimientos no funcionales ***17***
2.5 MODELO DE CASOS DE USO DEL SISTEMA. 19
2.5.1 Definición de los actores del sistema a automatizar. ***19***
FIG 2.2 : DIAGRAMA DE CASOS DE USO DEL SISTEMA. 19
CONCLUSIONES. 21

CAPITULO 3. ANÁLISIS Y DISEÑO DEL SISTEMA **22**

INTRODUCCIÓN. 22
3.1 ANÁLISIS. 22
3.1.2 Diagrama de clases de análisis. ***22***
3.2 DISEÑO. 25
3.2.1 Diagramas de interacción ***25***
3.2.2 Diagrama de clases del diseño. ***26***
CONCLUSIONES. 30

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS. **31**

4.1 INTRODUCCIÓN 31
4.2 DIAGRAMA DE COMPONENTES 31

Fig: 4.1 Diagrama de Componentes caso de usos "Analizar Muestra" *32*
Fig: 4.2 Diagrama de componentes Caso de uso "Descontaminación Genérica" *33*
Fig. 4.3 Diagrama de componentes Caso de uso "Ejecutar Muestra" *34*
Fig. 4.4 Diagrama de componentes Caso de uso "Generar Descripciones" *35*
CONCLUSIONES 36

CAPÍTULO 5: ESTUDIO DE FACTIBILIDAD. **37**

5.1. INTRODUCCIÓN 37
5.2. PLANIFICACIÓN 37
5.2.1. Cálculo de Puntos de Casos de Uso sin ajustar *37*
5.2.2. Cálculo de Puntos de Casos de Uso ajustados *39*
5.2.3 Calcular esfuerzo de FT Implementación *42*
CONCLUSIONES. 44

CONCLUSIONES FINALES. **45**

RECOMENDACIONES. **46**

REFERENCIAS BIBLIOGRÁFICAS. **48**

BIBLIOGRAFÍA **48**

ANEXO_1: MENSAJES CON MUESTRAS DE CÓDIGOS MALIGNOS, REPORTADOS INTERNACIONALMENTE, RECIBIDOS DIARIAMENTE EN LAS ÚLTIMAS SEMANAS **49**

ANEXO_2: ESTADÍSTICAS PROGRAMAS MALIGNOS EN CUBA HASTA 15 DE MAYO 2009. ..... **50**

ANEXO_ 3: DESCRIPCIÓN TEXTUAL DE LOS CASOS DE USOS DEL SISTEMA. **51**

ANEXO_6: DIAGRAMAS DE SECUENCIA. 66

ANEXO_ 4: DIAGRAMA DE SECUENCIA CU_ANALIZAR_MUESTRA_PC_CONTAMINADA **66**

ANEXO_ 5: DIAGRAMA DE SECUENCIA CU_ANALIZAR_MUESTRA_PC_LIMPIA **67**

ANEXO_ 6: DIAGRAMA DE SECUENCIA CU_DESCONTAMINACIÓN_G **68**

ANEXO_ 7: DIAGRAMA DE SECUENCIA CU_EJECUTAR_MUESTRA **69**

ANEXO_ 7: DIAGRAMA DE SECUENCIA CU_GENERAR_DESCRIPCIONES **70**

GLOSARIO DE TÉRMINOS. **71**

Introducción

El análisis de la gran cantidad de nuevas muestras de programas malignos que llegan diariamente a los Laboratorio antivirus se ha convertido en un gran reto. A la par es necesario desarrollar métodos de identificación y descontaminación seguros y eficaces, los cuales se encuentran en continua evolución. Todo lo anterior obliga a diseñar e implementar herramientas que auxilien en el proceso de automatización de las tareas repetitivas a la vez que ayuden a evaluarlas y mejorarlas, tomando como base la experiencia de los analistas de estos códigos nocivos.

Por tanto el **objeto de estudio** de este trabajo está relacionado con la Automatización del Proceso de análisis para la descontaminación de programas malignos.

El **campo de acción** queda enmarcado específicamente en el proceso de actualización de la descontaminación de programas malignos para el antivirus desarrollado en Segurmatica. La idea que se persigue como objetivo general es Automatizar el proceso de análisis de códigos malignos. Para cumplir el objetivo trazado, se desarrollaron las siguientes tareas:

- ✓ Estudiar los procesos de análisis de programas malignos, obtención de la información necesaria para la descontaminación, actualización de los antivirus y control de calidad que llevan a cabo los analistas de la empresa.
- ✓ Estudiar la descontaminación genérica implementada en el antivirus.
- ✓ Evaluar herramientas empleadas en el análisis de programas malignos que permiten automatizar el proceso de análisis, incluyendo el estudio de reportes generados por las herramientas empleadas en el análisis de programas malignos.
- ✓ Estudiar el comportamiento de los distintos tipos de programas malignos en los diferentes sistemas operativos.
- ✓ Proponer algoritmos que permitan automatizar procesos de análisis de programas malignos, obtención de la información para la descontaminación, actualización de descontaminación en Antivirus y su control de calidad.

- ✓ Proponer algoritmos que permitan evaluar la efectividad de la descontaminación genérica implementada en el Antivirus.
- ✓ Implementar algoritmos y desarrollar un conjunto de herramientas que interactuando entre sí logren el objetivo final automatizar el proceso de análisis y la obtención de la información para la descontaminación.

Estructura del documento:

Capitulo 1: Describe cómo se realiza el proceso de análisis y descontaminación actualmente en Segurmática. Refleja el estado del arte del tema, tendencias, técnicas y metodologías. Trata la situación de las tecnologías a utilizar en el desarrollo de las herramientas y se explican los conceptos principales que se van a tratar.

Capítulo 2: Describe el problema, la situación problémica y los procesos que serán objeto de automatización. Además aborda aspectos esenciales del dominio de los trabajadores que intervienen y los requerimientos a tener en cuenta.

Capítulo 3: Detalla el análisis y diseño del sistema a desarrollar, siguiendo la metodología RUP.

Capitulo 4: Define cómo se organizan las clases y objetos en componentes.

Capitulo 5: Hace referencia a la planificación y estimación del proyecto.

Capitulo 1. Fundamentación Teórica.

Introducción.

Hoy en día la descontaminación de programas malignos en el mundo se ha convertido en una problema serio para las empresas antivirus, pues la cifra de muestras a analizar por día es superior a las 10000. En la práctica, lo que sucede es que de manera general los productos antivirus no parecen estar haciendo mucho hincapié en este aspecto y como resultado las PC no quedan bien descontaminadas e incluso los sistemas quedan inestables. En el caso del la Empresa de Consultoría y Seguridad Informática Segurmática, perteneciente al MIC, siempre ha sido un objetivo de primer orden dar solución, con la mayor calidad y eficiencia posible, a todas las muestras de códigos malignos reportados nacionalmente a la empresa y a aquellas reportadas como de amplia difusión o más dañinas. El número de muestras ha ido también en ascenso por lo que es necesario automatizar el proceso de análisis de los códigos malignos y la actualización de las descontaminaciones correspondientes en los antivirus. (3)

La inmensa mayoría de dichos programas están destinados a los sistemas operativos Windows, y según el informe esa cifra ha crecido debido al incremento del número de cibercriminales que se dedican a la creación de virus de forma profesional y que son pagados por ello. Parece que programar virus ya no es sólo para curiosos, y ahora buena parte de esos lanzamientos se utilizan con todo tipo de motivos por ejemplo los económicos.

Algunos datos:

- ✓ **KAV** detecta más de 2 200 000 P.M. (4)
- ✓ **Nod32**, Panda, AVG, McAfee rondan 90 **000.** (5)
- ✓ **SegaV** detecta más de 46 500. (2)

1.1 Contaminación de PM en Cuba.

Cuba con el desarrollo de las nuevas tecnologías no se ha quedado aislada de esta situación, ya que han aumentando vertiginosamente la cantidad de programas malignos nuevos desde 63 que se reportaron a Segurmatica en el año 2002, hasta 116 en el 2003 lo que representó un crecimiento de un 184%. En el 2004 se reportaron 305 y el crecimiento esta vez fue de un 262%, en el año 2005, 2006, 2007 se reportaron 296, 268 y 290 respectivamente estabilizándose e incluso disminuyendo esté índice de crecimiento, pero en el pasado año 2008 se reportaron a la empresa 1725 lo que representó un número nunca visto en el país hasta el momento para un crecimiento de un 594% de la cantidad de muestras reportadas. En el año en curso se han reportado (estadísticas actualizadas del 15 de Mayo de 2009) 510 PM lo que representa un incremento proporcional respecto a años anteriores. En resumen en los 2 últimos años 2008 y 2009 se ha reportado el 53% de todos los PM reportados a Segurmatica desde el año 2002. (2) (6)

1.2 Sistemas de Análisis de Programas Malignos en el Mundo.

En el mundo cada empresa Antivirus tiene su propio sistema de análisis, pero este representa parte de su "**KnowHow**" por lo que no son difundidos, no obstante, existen varias herramientas que pueden ser útiles para el análisis de algunos aspectos referentes al funcionamiento de los programas malignos como son:

- ✓ **IntCtrl(**herramienta para el análisis del registro y el disco**)**
- ✓ **Listdll(**herramienta para el análisis de procesos y las dlls que usa**)**
- ✓ **Catchme(**herramienta para el análisis de procesos ocultos**)**
- ✓ **Handle(**herramienta para el análisis de procesos**)**
- ✓ **RootkitRevealer(**herramienta para el análisis de procesos ocultos y ficheros ocultos en el disco**)**
- ✓ **Fport(**herramienta para el análisis de puertos abiertos**)**
- ✓ **Pv(**herramienta para el análisis de procesos**)**

1.2.3 Sistemas de Análisis de Programas Malignos en Segurmática.

En la empresa cada especialista tiene una suite de herramientas y hace un análisis manual de los efectos del PM tomando en cuenta una gran base de conocimiento formada a partir de su propia experiencia en el análisis. A partir de este resultado se obtiene la información necesaria para actualizar la descontaminación en el antivirus y se elabora una descripción con los aspectos principales, la cual se libera a los usuarios. Además se valora si es necesario implementar nuevos métodos de descontaminación.

1.3 Lenguajes y tecnologías utilizadas.

Para el desarrollo de esta herramienta se decidieron utilizar varios lenguajes de programación debido a la interacción que tendrían muchas herramientas en el proceso de análisis. C++ fue el lenguaje escogido para el trabajo con ficheros, acceso al registro y para el intercambio de resultado entre las demás herramientas. VBS para el monitoreo de procesos, servicios y puertos de la forma más simple posible. Bat para la configuración de las múltiples herramientas de análisis y finalmente AUTOIT para el control de todo el proceso.

1.4 Metodología de desarrollo y lenguaje de modelado.

1.4.1 Herramienta de Modelado.

Visual Paradigm es la herramienta CASE de modelación visual propuesta. “Las Herramientas CASE (de su siglas en inglés: Computer Aided Software Engineering; en español: Ingeniería de Software Asistida por Ordenador) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero. Estas herramientas nos pueden ayudar en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores entre otras.”

La herramienta Visual Paradigm está diseñada para una amplia gama de usuarios, incluyendo Ingenieros de Software, Analistas de Sistemas, Analistas de Negocios y Arquitectos de Sistemas que estén interesados en la creación de grandes sistemas de software de manera confiable a través de la POO. Las transiciones del análisis al diseño, y de este a la implementación, están adecuadamente integradas dentro de la herramienta CASE, de manera que reduce significativamente los esfuerzos de todas las etapas del ciclo de desarrollo de software.

Visual Paradigm ofrece:

Diseño centrado en casos de uso y enfocado al negocio, lo cual permite generar un software de mayor calidad.

- ✓ Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- ✓ Capacidades de ingeniería directa (en su versión profesional) e inversa.
- ✓ Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.
- ✓ Disponibilidad de múltiples versiones, para cada necesidad.
- ✓ Disponibilidad de integrarse en los principales IDE (Integrated Development Environment).
- ✓ Disponibilidad en múltiples plataformas.
- ✓ Generación de código fuente en varios lenguajes de programación.

El sistema que se propone será desarrollado con las técnicas de POO, apoyándose en RUP y Visual Paradigm y usando como notación el lenguaje UML, que es la base del funcionamiento visual de RUP.

1.4.2 UML.

El Lenguaje Unificado de Modelado es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software. Es importante recalcar que UML no es una guía para realizar el análisis y diseño orientado a objetos, es decir, no es un proceso. UML es un lenguaje que permite la modelación de sistemas con tecnología orientada a objetos. También intenta solucionar el problema de propiedad de código que se da con los desarrolladores, al implementar un lenguaje de modelado común para todos los desarrollos se crea una documentación también común,

que cualquier desarrollador con conocimientos de UML será capaz de entender, independientemente del lenguaje utilizado para el desarrollo.

UML también contiene construcciones organizativas para agrupar los modelos en paquetes, lo que permite a los equipos de software dividir grandes sistemas en piezas de trabajo, para entender y controlar las dependencias entre paquetes, y para gestionar las versiones de las unidades del modelo, en un entorno de desarrollo complejo.

El modelo gráfico de UML tiene un vocabulario en el que se identifican:

- ✓ Elementos (abstracciones que constituyen los bloques básicos de construcción).
- ✓ Relaciones: Enlazan los elementos.
- ✓ Diagramas: Es la representación gráfica de un conjunto de elementos. Visualizan un sistema desde diferentes perspectivas.

Conclusiones

En este capítulo se ha hecho un análisis del auge de los programas malignos en el mundo en los últimos años y como Cuba no ha quedado expensa de esta situación, también se da una perspectiva de algunas de las soluciones líderes en el mundo en la identificación y descontaminación de programas malignos.

Además se hace un análisis de los lenguajes, metodologías herramientas y soluciones existentes para darle solución al proceso de automatización de la descontaminación de Programas Malignos. Tras él se llega a la conclusión de la importancia de realizar un conjunto de aplicaciones que no solo sean capaces de interactuar entre ellas sino también con las demás aplicaciones utilizadas por los especialistas en el Laboratorio y que su funcionamiento conjunto sea capaz de automatizar el análisis, teniendo en cuenta el grado de complejidad que puede presentar el análisis de una muestra maligna. Es de vital importancia el dominio de todas las herramientas así como del proceso de análisis para poder desarrollar un sistema de máxima calidad que cumpla con los requisitos propuestos y brinde al cliente un producto que satisfaga sus intereses.

Capítulo 2. Características del sistema.

Introducción.

En este capítulo se realiza un análisis de las características del sistema a desarrollar, haciendo hincapié en la situación problémica que da origen al mismo.

2.1 Objeto de estudio

2.1.1 Problema y situación problémica:

Este trabajo forma parte del proceso de automatización de algunos de los procesos realizados diariamente en el Laboratorio Antivirus de Segurmatica. Está dirigido a desarrollar una herramienta que sea capaz de realizar un análisis de una muestra maligna emulando las acciones que ejecutan los analistas en el Laboratorio.

En la empresa el proceso de análisis de programas Malignos se está efectuando de forma manual y con muchas herramientas a partir de las cuales se obtienen resultados en diferentes formatos. El analista para realizar esta tarea debe obtener información del estado del registro del sistema, procesos en memoria, servicios, puertos y sistema de archivos, en un ambiente limpio. Seguidamente ejecuta las muestras de Programas Malignos a analizar y compara la información anterior con la obtenida a posteriori. En dependencia de ello, localiza los archivos creados y/o modificados, las llaves, valores y datos de registro alterados, procesos y servicios en memoria, así como puertos abiertos. Esta información es “filtrada” y se descarta la que no es importante tener en cuenta para la descontaminación, a la vez que se procesa la información útil para la actualización del antivirus y la elaboración de la descripción.

De manera general se comprueban los ficheros creados y modificados en el sistema de archivos, haciendo énfasis en los ejecutables (por ejemplo: (.EXE, SCR, PIF, SYS, BAT, CMD, OCX, Dll, VBE, VBS, HTA, JS, COM, HTML) se revisan las subllaves, valores y datos de registro adicionados o modificados localizando aquellos que son importantes considerar en la descontaminación, como pueden ser las relacionadas con la ejecución de los componentes malignos desde el proceso de inicio del

sistema. Además, se analizan los procesos y servicios en memoria relacionados con los ficheros creados o modificados, dll inyectadas, servicios y se comprueba la existencia de puertos abiertos asociados con ellos.

A partir de esta información se decide el tipo de descontaminación que se asigne o implemente para combatir al código maligno y se elabora la descripción destinada al usuario final.

También se evalúa si es necesario mejorar la descontaminación genérica desarrollada e implementada en la empresa.

2.1.2 Objeto de automatización.

En Segurmatica no se cuenta con una herramienta que automatice todo este proceso. La herramienta tiene como objetivo la automatización desde el análisis de la muestra maligna hasta la generación de la descripción. Actualmente los analistas del laboratorio analizan las muestras con todas las herramientas de análisis e interpretan estos resultados con distintos formatos y se determina si la descontaminación genérica es suficiente. Además se genera las descripciones y resúmenes del análisis de forma manual.

2.1.3 Información que se maneja.

La información que se maneja son los ficheros fuentes de la descontaminación genérica asi como la base de conocimiento.

2.2 Propuesta de sistema.

Se propone hacer una herramienta que haga un análisis completo de una muestra Maligna con las mismas herramientas que trabajan los analistas en el laboratorio, integrándole otras que automaticen los procesos de análisis que hoy se ejecutan manualmente, que permita interpretar todos estos resultados y generar un resumen, una descripción. Además compruebe que la descontaminación genérica es suficiente para cubrir los efectos de la muestra analizada.

2.3 Modelo del Dominio.

El primer paso en el proceso de desarrollo de este software es precisamente alcanzar cierto nivel de conocimientos sobre el proceso de análisis de muestras malignas que se realiza en el laboratorio. Algo que solo se pudo lograr con una comunicación efectiva entre los especialistas y el equipo del proyecto, con el objetivo de llegar a un entendimiento de lo que se debía hacer, esta fue la clave del éxito en el desarrollo. El siguiente modelo de dominio captura los tipos más importantes de objetos que existen o los eventos que suceden en el entorno donde se desarrolla todo el proceso de análisis y elaborándose porque se determinó que los procesos del negocio no estaban lo suficientemente definidos. Por lo que no fue necesario un modelo completo del negocio. Las claves para la realización de este modelo fueron.

- ✓ Comprender la estructura y la dinámica del laboratorio de análisis de Segurmática.
- ✓ Comprender los problemas actuales del proceso de análisis e identificar las mejoras potenciales.
- ✓ Asegurar que los especialistas y el equipo de desarrollo tengan un entendimiento.
- ✓ Derivar los requerimientos del sistema que va a soportar el proceso a automatizar.

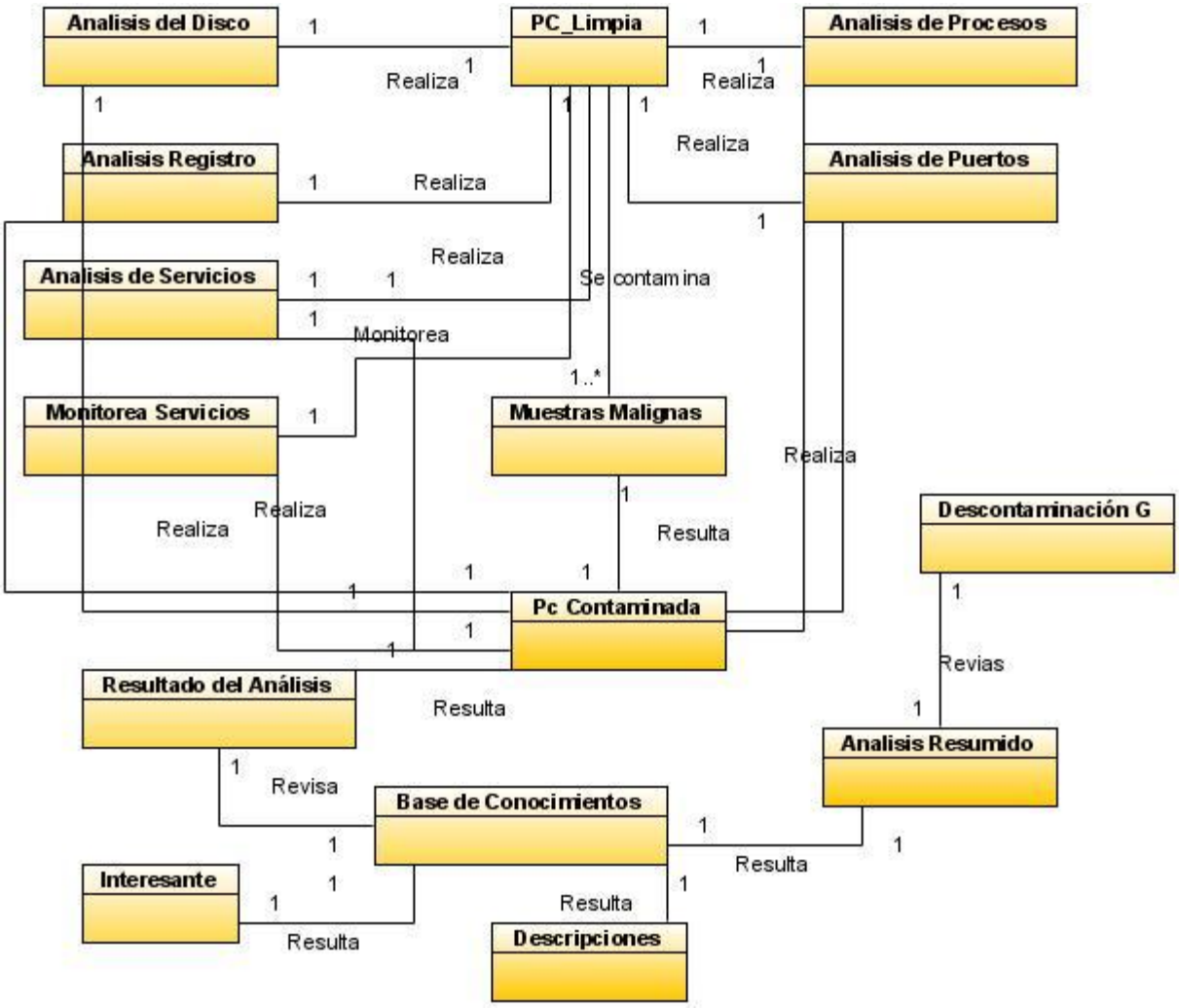


Fig 2.1: Modelo de Dominio “Analizar Programa Maligno”

2.3.1 Definición de las clases del modelo del dominio.

2.3.1.1 Análisis del Disco:

Análisis de disco define las aplicaciones encargadas de realizar un análisis de los cambios en el sistema de archivos de las unidades de soporte de almacenamiento de la PC fundamentalmente en cuanto a ficheros adicionados y a ficheros modificados primero en un estado limpio y después de que la PC haya sido infectada.

2.3.1.2 PC Limpia:

PC Limpia define el modelo de la PC cuando aún no ha sido ejecutado el programa maligno.

2.3.1.3 Análisis de Procesos:

Análisis de Procesos define la acción de un conjunto de aplicaciones que se dedican a hacer un análisis del estado de los procesos en memoria.

2.3.1.4 Análisis de Registro:

Análisis de Registro define la acción de un conjunto de aplicaciones que son las encargadas de analizar el estado del registro del sistema en cuanto a subllaves y valores adicionados y/o modificados.

2.3.1.5 Análisis de Servicios:

Análisis de Servicios define la acción de un conjunto de herramientas que se encargan de analizar el estado de los servicios en la PC y sus cambios antes y después de que las muestras malignas han sido ejecutadas en la PC.

2.3.1.6 Monitoreo de servicios:

Monitoreo de servicios define la acción de un conjunto de herramientas que son las encargadas de realizar un monitoreo del estado de los servicios del sistema antes de que la muestra a analizar sea ejecutada.

2.3.1.7 Análisis de Puertos:

Análisis de Puertos define la acción de un conjunto de herramientas que son las encargadas de analizar los cambios en las peticiones por determinados puertos y su cambio antes y después de ejecutada la muestra maligna.

2.3.1.8 Muestra Maligna:

Muestra Maligna define un conjunto de programas malignos que son el objetivo de todo el proceso de análisis. Estas muestras son un caso especial cuando son dlls y su ejecución se procederá con run32dll.

2.3.1.9 PC Contaminada:

PC contaminada define el modelo de la PC después de que ella fue ejecutada un programa maligno.

2.3.1.10 Resultado del Análisis:

Resultado del Análisis define el conjunto de resúmenes resultantes de todos los análisis hechos a la PC y comprobada su valides con la base de conocimientos.

2.3.1.11 Base de Conocimientos:

Base de Conocimientos define el conjunto de ficheros resultado de la experiencia del análisis de los especialistas que sirven para darle valides al análisis hecho a la muestra maligna.

2.3.1.12 Interesante:

Interesante se define como los comportamientos que son nuevos y que no se habían visto antes en ninguna muestra maligna y que es necesario separarlos para que un especialista los investigue y actualice la base de conocimientos.

2.3.1.13 Descripciones:

Descripciones se define como el conjunto de referencias que se tienen en el Laboratorio de las muestras malignas y sus funcionamientos.

2.3.1.14 Análisis Resumido:

Análisis Resumido se define como el resultado de la revisión del resultado del análisis con la base de conocimiento.

2.3.1.15 Descontaminación G:

Descontaminación Genérica se define como una representación de las funcionalidades de la descontaminación implementadas en el antivirus las cuales son suficientes para descontaminar genéricamente el programa maligno.

2.4 Especificación de los requisitos de software.

2.4.1 Requerimientos Funcionales

R1. Analizar Registro

1.1. Reporte de Llaves Adicionadas.

1.2. Reporte de Laves Modificadas.

1.3. Reporte de Valores Adicionados

1.4. Reporte de Valores Modificados.

R2. Analizar Disco.

2.1. Analizar Ficheros Adicionados y Modificados.

2.1.1. Reporte de ficheros adicionados.

2.1.2. Reporte de ficheros Modificados.

2.2. Copiar Ficheros Adicionados y Modificados.

R3. Analizar procesos

3.1. Reporte de procesos.

3.2. Reporte Procesos Ocultos.

R4.Analizar Servicios.

4.1. Reporte de Servicios.

4.1.1. Reporte de Monitoreo de Servicios.

R5. Generar descripciones.

5.1 Reporte de Análisis de la Muestra Maligna.

R6. Comprobar Genérica.

6.1. Reporte fuente de la Genérica.

R7.Obtener reportes Puertos.

7.1. Buscar Puertos abiertos.

R8. Ejecutar muestras.

2.4.2 Requerimientos no funcionales

En muchos casos los requerimientos no funcionales son fundamentales en el éxito del producto. Normalmente están vinculados a requerimientos funcionales. Hemos seleccionado los siguientes requisitos no funcionales:

Requerimientos de Apariencia o interfaz externa

- ✓ Interfaz con un diseño sencillo que se genera de acuerdo con los resultados del análisis.

Requerimientos de Usabilidad.

- ✓ El sistema estará diseñado para ser usado por cualquiera de los especialistas del Laboratorio, por lo que es necesario que cuente con un diseño de interfaz de fácil uso,
- ✓ Documentar bien la aplicación para poder hacer mejor uso de todos los servicios que este ofrece.

Requerimientos Software.

- ✓ La computadora donde se realice el análisis debe estar ejecutando el sistema operativo Windows 2000 o XP.

Requerimientos de Hardware.

- ✓ Se necesitan como requerimientos mínimos para una PC con procesador Pentium II o superior.
- ✓ 256 megabytes (MB) de memória RAM o más.

Requerimientos de Soporte

- ✓ La herramienta debe ser de fácil configuración.

Requerimientos de diseño

- ✓ Los lenguajes de programación que se usarán serán el C++, VBS, Bat, AUTOIT.

- ✓ Para el análisis y el diseño del sistema debe ser utilizada la metodología RUP, usando el lenguaje de modelación UML y como herramienta para llevarlo a cabo el Visual Paradigm.

Requerimientos de Seguridad:

- ✓ El sistema debe conservar su integridad y disponibilidad en todo momento.

Elementos de confiabilidad

- ✓ Todas las salidas del sistema tienen que tener el 100% de veracidad y precisión.
- ✓ Capaz de recuperarse a las fallas del sistema.

Requerimientos legales

- ✓ La herramienta de Análisis de Programas Malignos, así como la documentación de la misma; pertenecen a la UCI y a la empresa SEGURMATICA y solo a ellos se les permite su uso.

2.5 Modelo de Casos de Uso del Sistema.

2.5.1 Definición de los actores del sistema a automatizar.

Actores	**Justificación**
Sistema	Es en encargado de controlar la ejecución de todo el proceso de análisis.

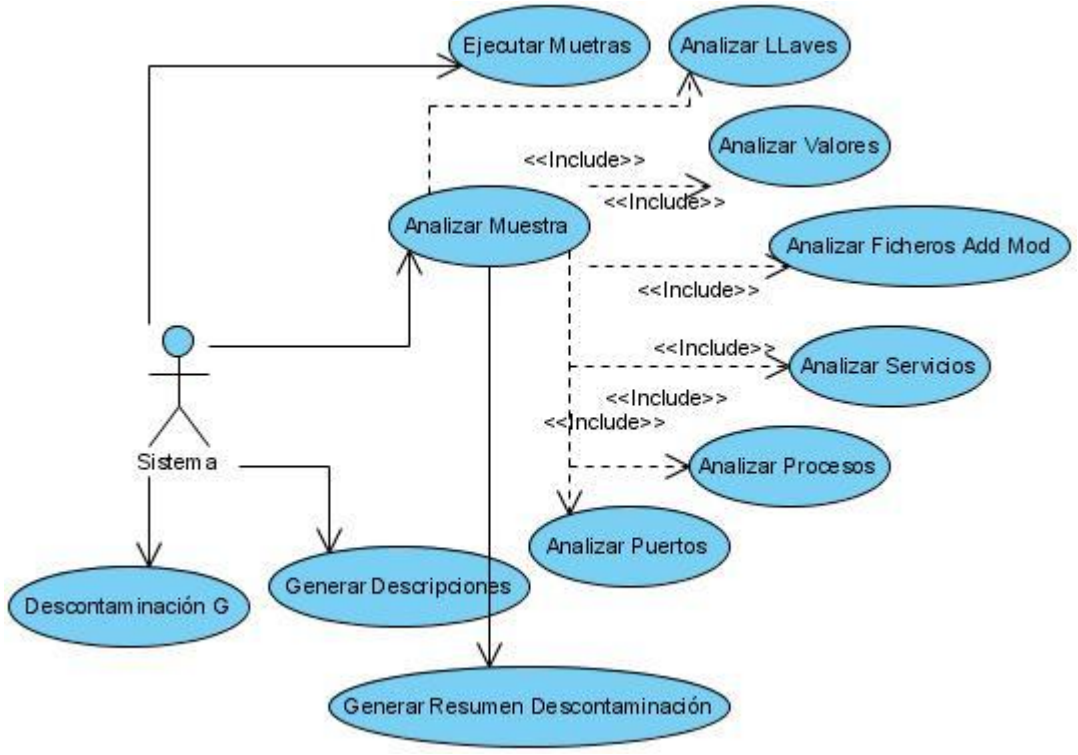


Fig 2.2 : Diagrama de casos de uso del sistema.

2.5.3 Descripción de los Casos de uso.

En el **Anexo_2** se muestran las descripciones textuales de los casos de usos descritos en el diagrama.

Conclusiones.

En este capítulo se realizó un análisis de los objetivos estratégicos de Segurmática empresa para la cual se va a desarrollar el producto y del flujo actual de los procesos involucrados en el campo de acción. También se realizo un análisis crítico de cómo se ejecutan actualmente estos procesos que nos conducen finalmente a las causas que originan la situación problémica .Después se describieron los procesos que serán objeto de automatización y se mencionaron algunas de las herramientas que existen en la empresa y que están vinculadas al proceso de análisis de muestras malignas.
Se mencionaron los documentos específicos que se procesen, se realizo una propuesta de sistema y como debe funcionar y los requerimientos (funcionales, no funcionales).En resumen en este capítulo se han tratado las ideas que se desarrollaran para automatizar el proceso de análisis de Programas Malignos en la Empresa. Además se realizó un estudio detallado del proceso de análisis, y finalmente se diseñó una propuesta de sistema que satisface las especificaciones de automatización de los procesos de análisis actuales y se completó el levantamiento de los requerimientos esperados.

Capitulo 3. Análisis y diseño del sistema.

Introducción.

En este capítulo se abordará el tema del análisis y diseño de la propuesta del sistema. Se propone en función de satisfacer los requerimientos esperados, se confeccionan los diagramas de clases del análisis y los diagramas de interacción para los casos de usos críticos en el diseño, se completa el modelo de clases de diseño para las especificaciones de la aplicación. Se realiza además el diseño del modelo de datos y el prototipo de interfaz para el sistema.

3.1 Análisis.

El análisis consiste en obtener una visión del sistema que se preocupa de ver QUÉ hace, de modo que sólo se interesa por los requisitos funcionales.

En el análisis se debe profundizar en los casos de usos, detallándolos de manera que permitan reflejar una vista interna del sistema descrita con el lenguaje de los desarrolladores. En esta vista interna se especifican mejor los casos de uso y se determinan las clases necesarias para llevar a cabo las funcionalidades en ellos contenidos.

Este proceso se desarrolla fundamentalmente dentro de la fase de elaboración y se corresponde principalmente con el flujo de trabajo de análisis y diseño.

3.1.2 Diagrama de clases de análisis.

Un Diagrama de clases del análisis es un artefacto en el que se representan los conceptos en un dominio del problema. Representa las cosas del mundo real, no de la implementación automatizada de estas.

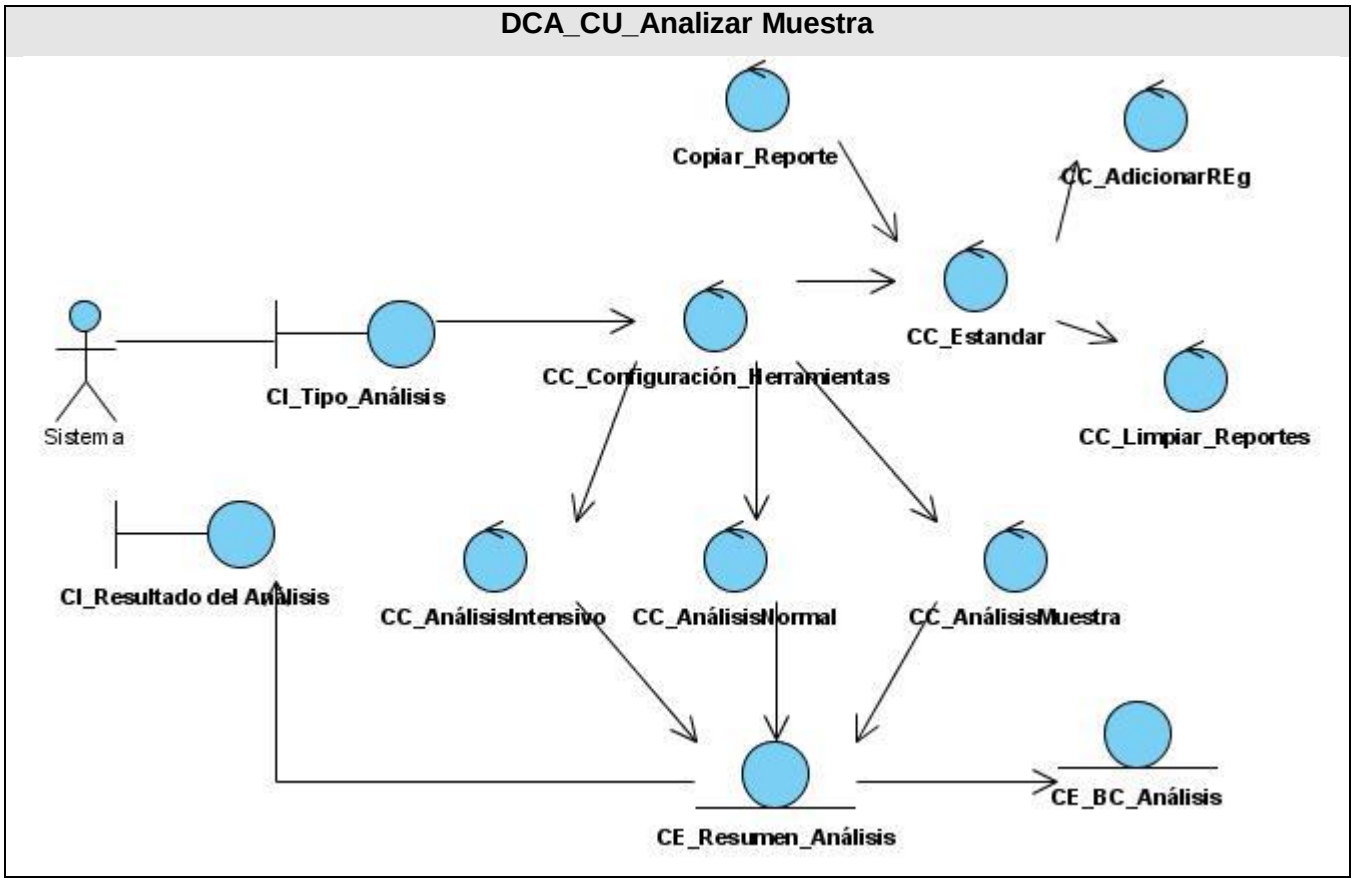


Figura 3.1*:* Diagrama de clase del análisis “Analizar Muestra”.

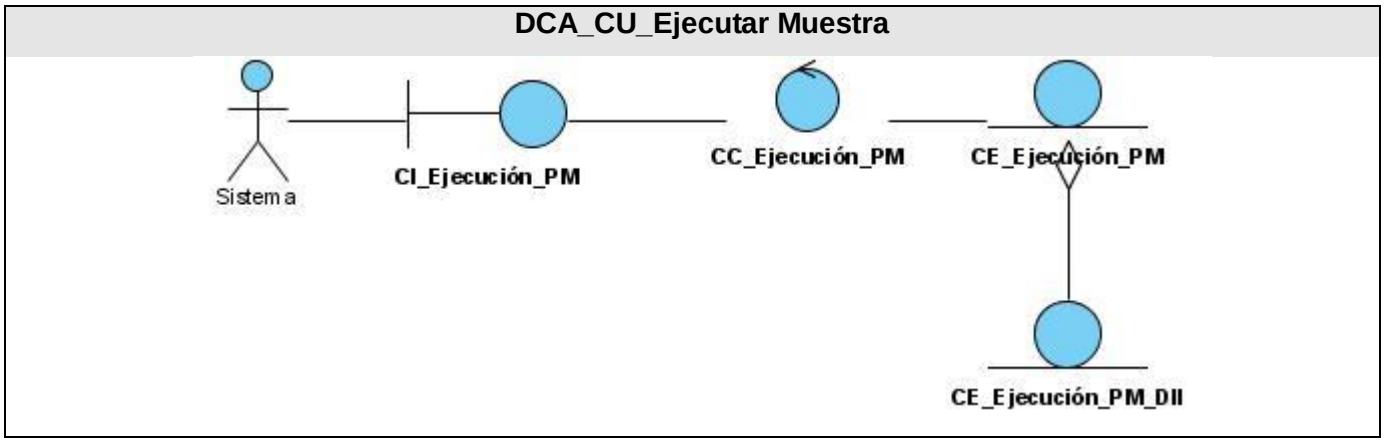


Figura 3.2*:* Diagrama de clase del análisis “Ejecutar Muestra”.

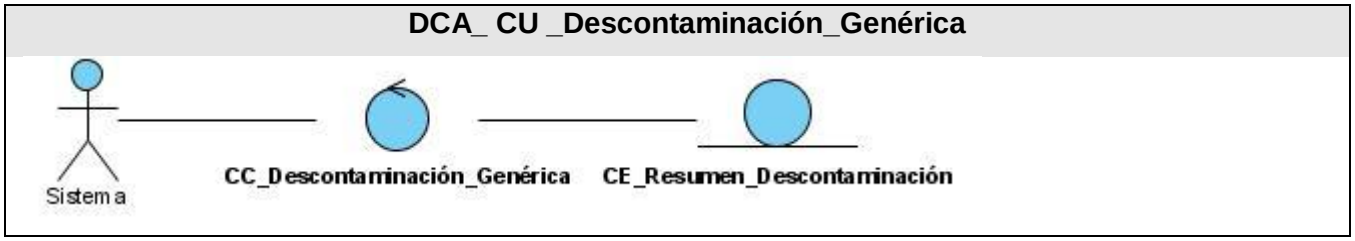


Figura 3.3*:* Diagrama de clase del análisis “Descontaminación Genérica”.

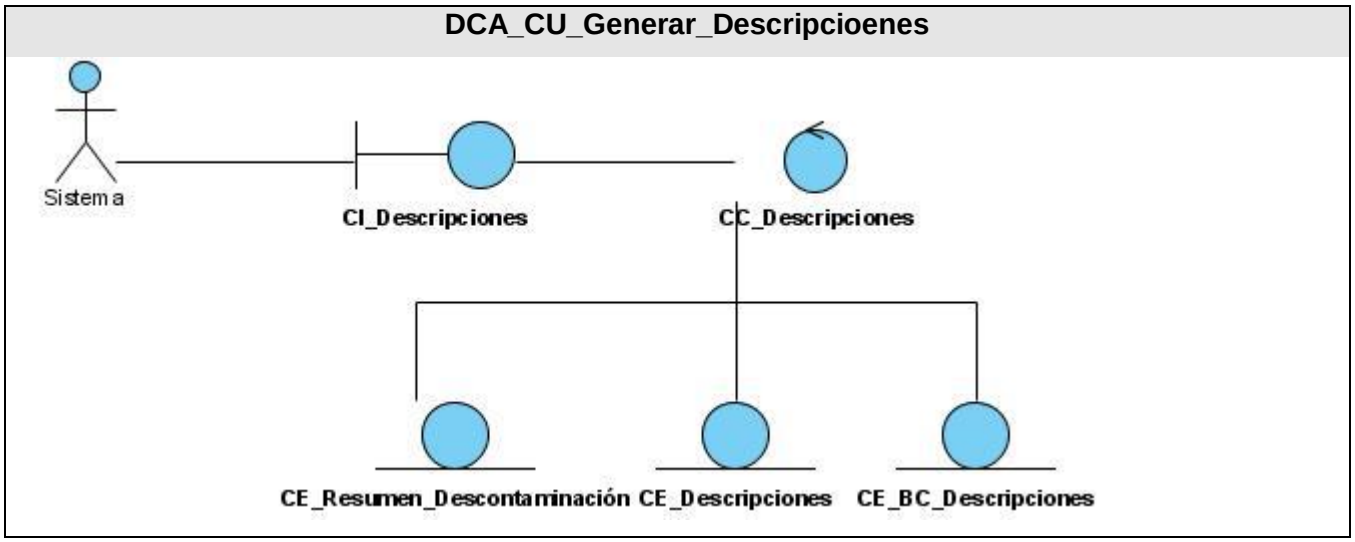


Figura 3.4: Diagrama de clase del análisis “Generar Descripciones”.

3.2 Diseño.

Después de realizar un estudio más profundo sobre el funcionamiento del sistema y todas las clases definidas en el diagrama de clases del análisis, se pasa a la fase de diseño.

3.2.1 Diagramas de interacción.

Los diagramas de interacción se utilizan para modelar los aspectos dinámicos de un sistema. La mayoría de las veces, esto implica modelar instancias concretas o prototípicas de clases, interfaces, componentes y nodos, junto con los mensajes enviados entre ellos, todo en el contexto de un escenario que ilustra un comportamiento. Los diagramas de interacción pueden utilizarse para visualizar, especificar, construir y

documentar la dinámica de una sociedad particular de objetos, o se pueden utilizar para modelar un flujo de control particular de un caso de uso.

En el Anexo_6 se muestran los diagramas de secuencia asociados al caso de uso crítico del sistema.

3.2.2 Diagrama de clases del diseño.

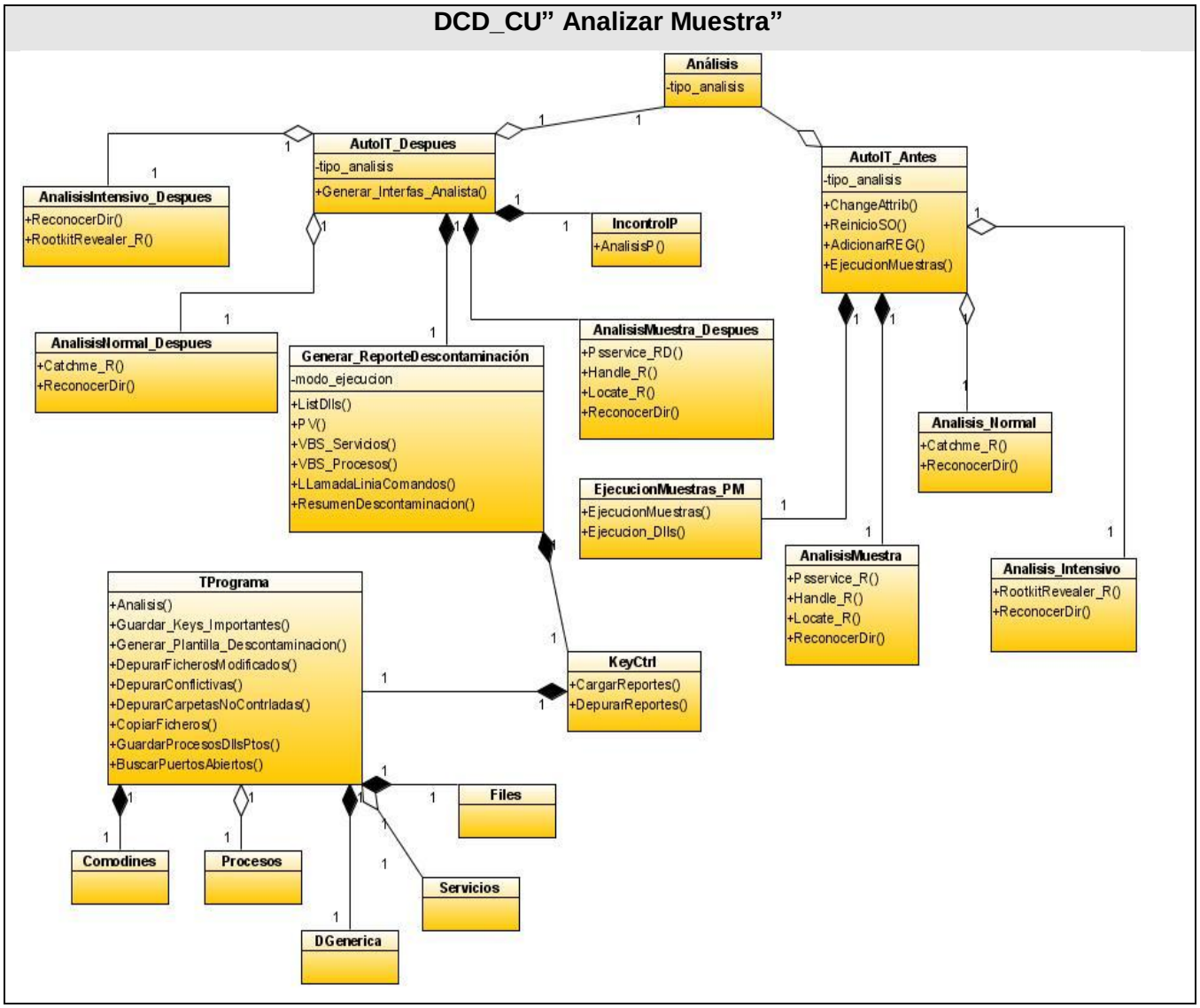


Figura 3.5: Diagrama de clase del diseño "Analizar Muestra".

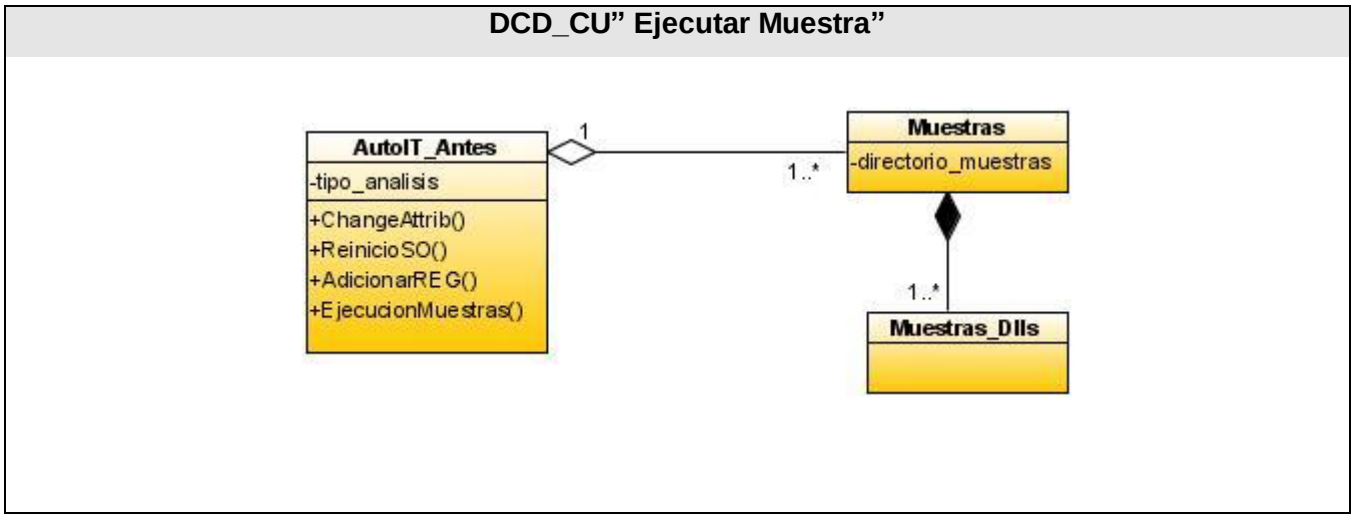


Figura 3.6: Diagrama de clase del diseño "Ejecutar Muestra".

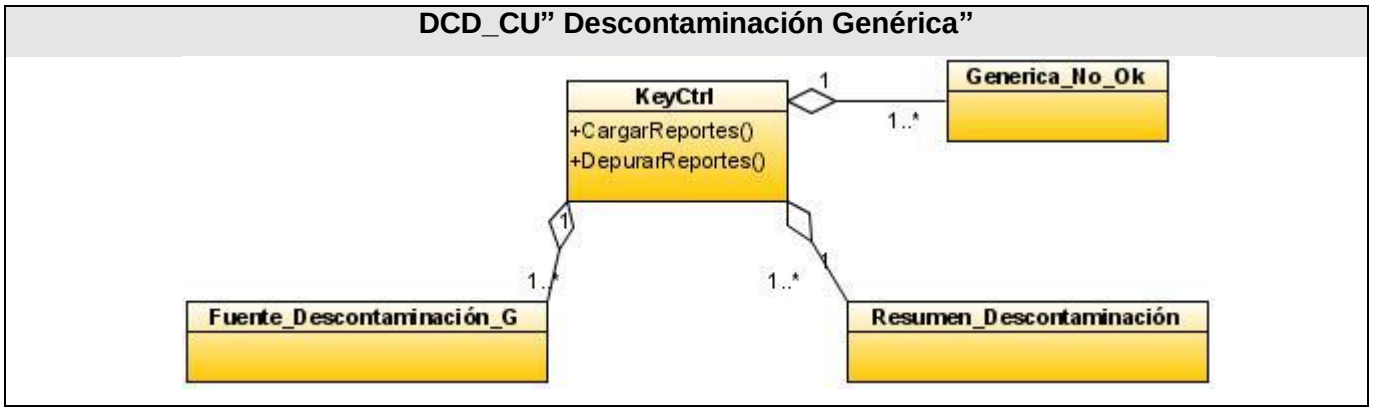


Figura 3.7: Diagrama de clase del diseño "Descontaminación Genérica".

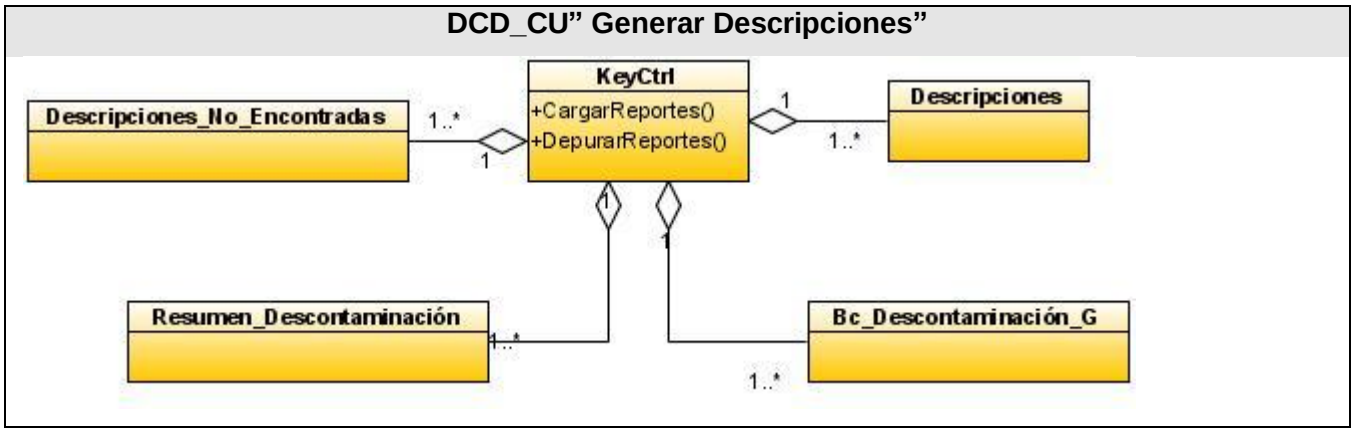


Figura 3.8: Diagrama de clase del diseño "Generar Descripciones".

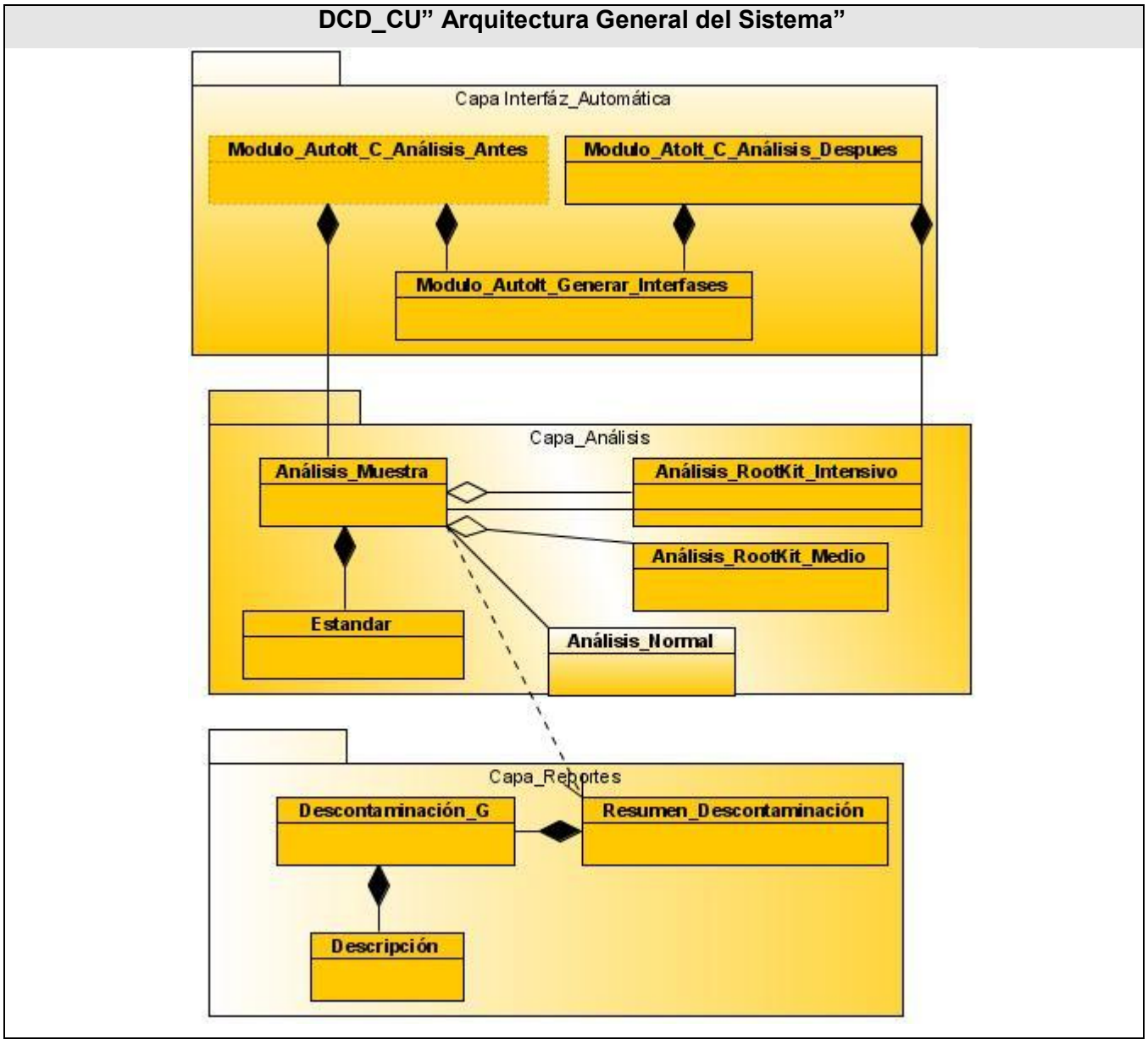


Figura 3.9: Diagrama de clase del diseño Arquitectura General del Sistema".

Conclusiones.

En este capítulo se abordó el análisis y el diseño sistema propuesto, se desarrollaron los diagramas de clases del análisis como una primera aproximación y el modelo del diseño. Luego de haber desarrollado este conjunto de actividades se procedió a automatizar el proceso de análisis, cuidando de garantizar el cumplimiento de las condiciones que se puntualizaron en este capítulo y que no pueden faltarle al producto.

Capítulo 4: Implementación y Pruebas.

4.1 Introducción

El presente capítulo tiene como objetivo desarrollar los artefactos correspondientes a la implementación del sistema, comenzando con el resultado más importante del capítulo anterior: el modelo de diseño. A partir de este artefacto se realizan los diagramas de componentes y despliegue que conforman lo que se conoce como: Modelo de implementación; de esta forma se describen cómo los elementos del modelo del diseño se implementan en términos de componentes y se organizan de acuerdo a los nodos específicos en el modelo de despliegue.

4.2 Diagrama de componentes

Los diagramas de componentes muestran la separación de un sistema de software en componentes físicos y las dependencias entre ellos.

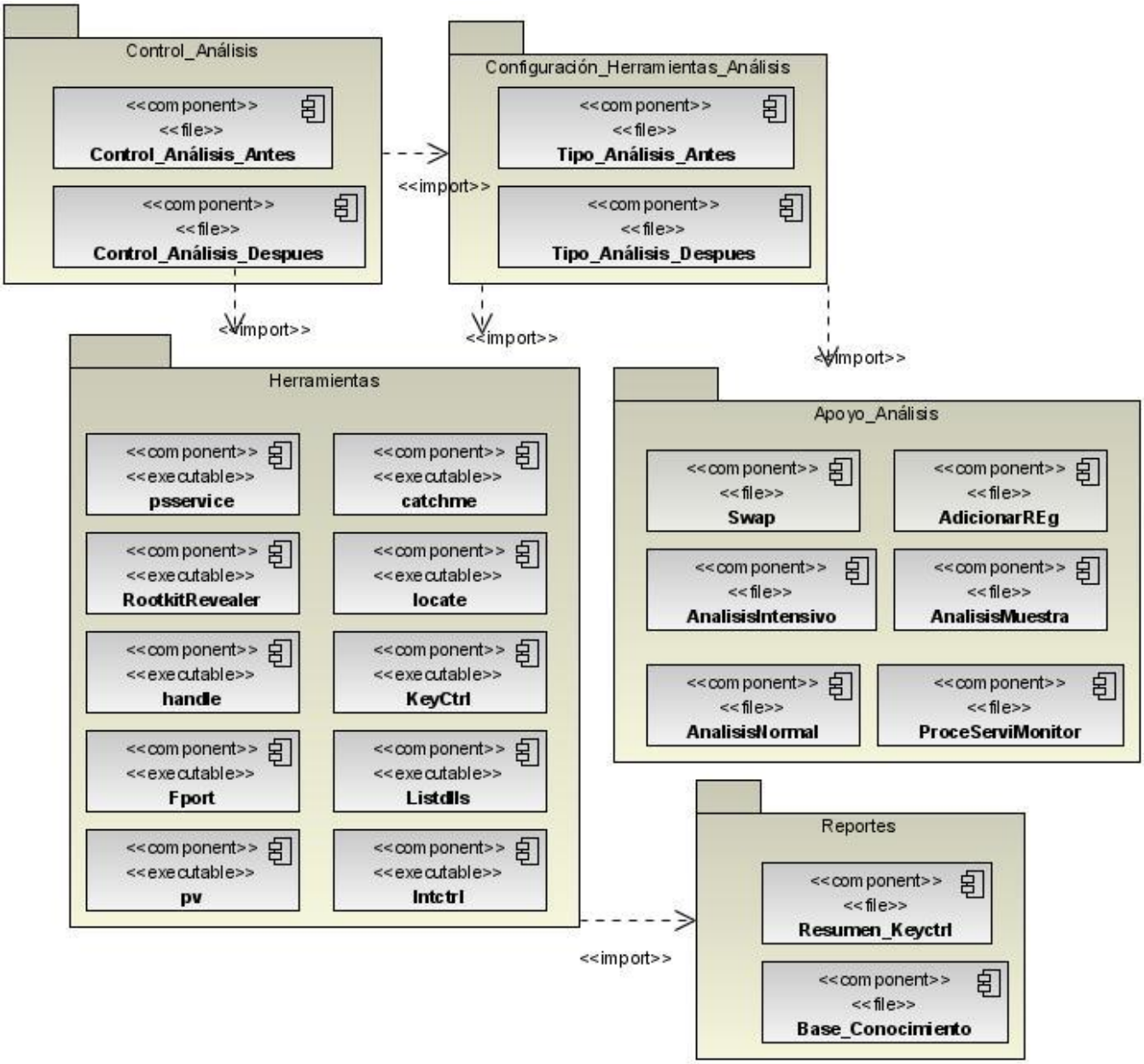


Fig: 4.1 Diagrama de Componentes caso de usos “Analizar Muestra”

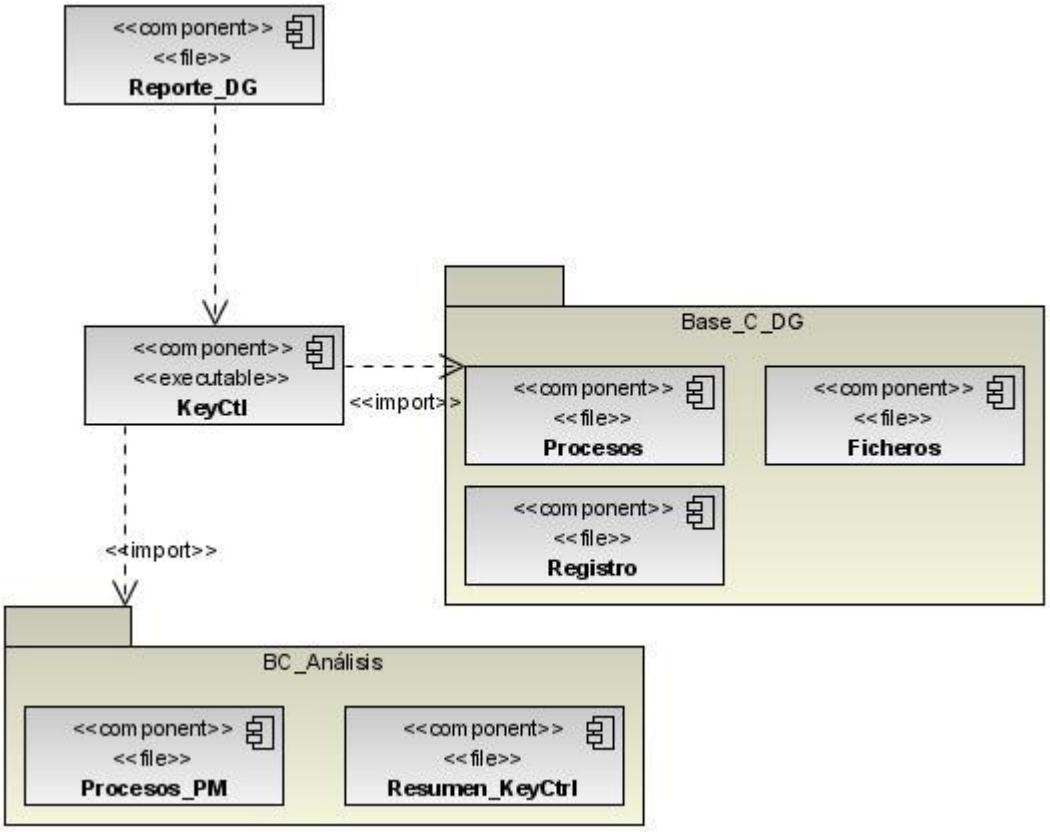


Fig: 4.2 Diagrama de componentes Caso de uso “Descontaminación Genérica”

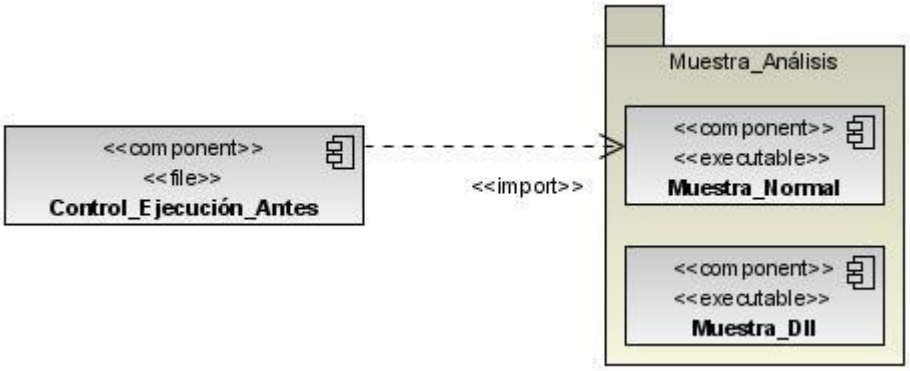


Fig. 4.3 Diagrama de componentes Caso de uso “Ejecutar Muestra”

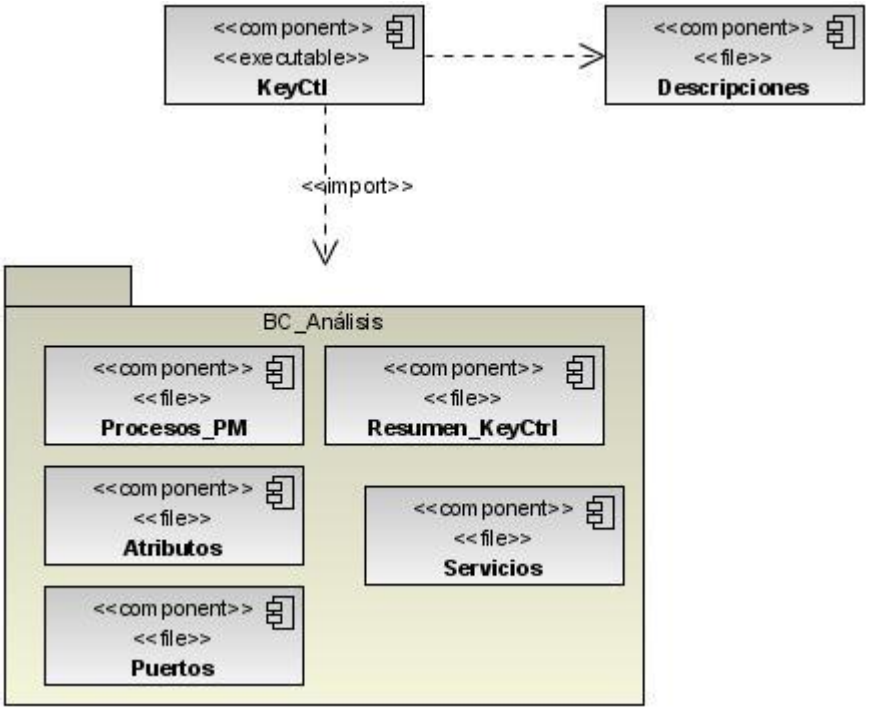


Fig. 4.4 Diagrama de componentes Caso de uso “Generar Descripciones”

Conclusiones

En este capítulo se desarrollaron los elementos correspondientes al modelo de implementación: los diagramas de componentes por cada caso de uso crítico. Al concluir se completaron los artefactos fundamentales que constituyen la base de la aplicación para el análisis de Muestras Malignas, dando cumplimiento de esta forma, a los requerimientos y necesidades del cliente.

Capítulo 5: Estudio de Factibilidad.

5.1. Introducción

En este capítulo se hace una breve descripción a una de las actividades más importantes llevadas a cabo en el proceso de gestión de proyecto de software sobre la planificación del proyecto, la estimación. Entre las diversas técnicas útiles para la estimación de costos y de tiempos se ha seleccionado la estimación mediante Casos de Uso. En su desarrollo se ofrecen una serie de pasos a seguir para lograr excelentes resultados, entre los que se encuentran:

- ✓ Cálculo de Puntos de Casos de Uso sin ajustar.
- ✓ Cálculo de Puntos de Casos de Uso ajustados.
- ✓ Calcular esfuerzo del Flujo de Trabajo de Implementación.

5.2. Planificación

Para la realización de una excelente planificación se ha utilizado la estimación mediante el análisis de Puntos de Casos de Uso. Este es un método propuesto de estimación del tiempo de desarrollo de un proyecto mediante la asignación de "pesos" a un cierto número de factores que lo afectan, para finalmente, contabilizar el tiempo total estimado para el proyecto a partir de esos factores.
A continuación se muestra el cálculo para la aplicación de este método:

5.2.1. Cálculo de Puntos de Casos de Uso sin ajustar

Se calcula a partir de la siguiente ecuación:

UUCP = UAW + UUCW

Donde:

- ✓ UUCP: Puntos de Casos de Uso sin ajustar
- ✓ UAW: Factor de Peso de los Actores sin ajustar

- ✓ UUCW: Factor de Peso de los Casos de Uso sin ajustar

- ✓ **Factor de Peso de los Actores sin ajustar (UAW)**

Tipo Actor	Descripción	Factor	#Actores	Resultado
Simple	Otro sistema que interactúa con el sistema a desarrollar mediante una interfaz de programación (API, Application Programming Interface).	1	40	40
Medio	Otro sistema que interactúa con el sistema a desarrollar mediante un protocolo o una interfaz basada en texto.	2	5	10
Complejo	Una persona que interactúa con el sistema mediante una interfaz gráfica.	3	1	3

Tabla 5.1 Factor de Peso de los Actores sin ajustar (UAW).

Total 17 **UAW=17**

- ✓ **Factor de Peso de los Casos de Uso sin ajustar (UUCW)**

Tipo Caso Uso	Descripción	Factor	#Casos Uso	Resultado
Simple	1-3 Transacciones	5	6	30
Medio	4-7 Transacciones	10	3	30
Complejo	+8 Transacciones	15	1	15

Tabla 5.2 Factor de Peso de los Casos de Uso sin ajustar (UUCW).

Total = 60

UUCW = 60

UUCP: Puntos de Casos de Uso sin ajustar

UUCP = **UAW+UUCW.**

UUCP = 17 + 60
UUCP= 77

5.2.2. Cálculo de Puntos de Casos de Uso ajustados

Después de tener los Puntos de Casos de Uso sin ajustar, se procede a ajustar mediante la siguiente

Ecuación: **UCP = UUCP x TCF x EF**

Donde:

UCP: Puntos de Casos de Uso ajustados.

UUCP: Puntos de Casos de Uso sin ajustar.

TCF: Factor de complejidad técnica.

EF: Factor de ambiente.

Factor de complejidad técnica: Este coeficiente se calcula mediante un conjunto de Factores que determinan la complejidad del sistema, cada factor se cuantifica con un valor de 0 a 5, donde 0 significa un aporte irrelevante y 5 un aporte muy importante. La ecuación para su cálculo es:

TCF = 0.6 + 0.01 * Σ (Peso * Valor Asignado).

- ✓ **Factor de complejidad técnica (TCF)**

Factor	Descripción	Peso	Valor asignado	Comentario	Σ(Peso * Valor)
T1	Sistema distribuido.	2	1		2
T2	Objetivos de performance o tiempo de respuesta.	1	3		3
T3	Eficiencia del usuario final.	1	3		3
T4	Procesamiento interno complejo.	1	5		5
T5	El código debe ser reutilizable.	1	2		2
T6	Facilidad de instalación.	0.5	4		2
T7	Facilidad de uso.	0.5	4		2
T8	Portabilidad.	2	4		8
T9	Facilidad de cambio.	1	5		5
T10	Concurrencia.	1	1		1
T11	Incluye objetivos especiales de seguridad.	1	2		2
T12	Provee acceso directo a terceras partes.	1	0		0
T13	Se requieren facilidades especiales de entrenamiento a los usuarios.	1	1		1
	TOTAL		36		

Tabla 5.3 Factor de Complejidad Técnica (TCF).

TCF = 0.6+0.01 x Σ (Peso x Valor asignado)

TCF = 0.6+0.01*36

TCF = 0.216

Factor de ambiente: Contempla las habilidades y el entrenamiento del grupo de desarrollo por su importancia en las estimaciones de tiempo. Al igual que el factor de complejidad técnica se cuantifican con valores de 0 a 5. La ecuación para su cálculo es:

EF = 1.4 - 0.03 * Σ (Peso i * Valor Asignado i)

- ✓ **Para calcular Factor Ambiente (EF)**

Factor	Descripción	Peso	Valor asignado	Σ(Peso * Valor)
E1	Familiaridad con el modelo de proyecto utilizado.	1.5	2	3
E2	Experiencia en la aplicación.	0.5	2	1
E3	Experiencia en orientación a objetos.	1	4	4
E4	Capacidad del analista líder.	0.5	5	2.5
E5	Motivación.	1	4	4
E6	Estabilidad de los requerimientos.	2	2	4
T7	Personal part - time.	-1	4	-4
T8	Dificultad del lenguaje de programación.	-1	3	-3
	TOTAL			11.5

Tabla 5.4 Cálculo del Factor Ambiente (EF).

EF = 1.4 - 0.03 x Σ (Peso x Valor asignado)

EF = 1.4 - 0.03 *11.5.

EF = 1.4 - 0.345

EF = 1,055

Finalmente, los Puntos de Casos de Uso ajustados resultan:

UCP = UUCP x TCF x EF

UCP = 77 * 0.216 * 1,055

UCP = 85,29675

5.2.3 Calcular esfuerzo de FT Implementación

E = UCP * CF

Donde:

E: Esfuerzo estimado en horas-hombre.

UCP: Puntos de Casos de Uso ajustados.

CF: Factor de conversión.

- **Para calcular CF**

Para calcular el Factor de conversión se contabilizan cuántos factores de los que afectan el factor ambiente están por debajo del valor medio, para los factores E1 a E6.
También se contabilizan cuántos factores de los que afectan el factor ambiente están por encima del valor medio, para los factores E7 y E8.

Entonces, si:

(Total EF ≤ 2): CF = 20 horas-hombre

(Total EF = 3 ó Total EF = 4): CF = 28 horas-hombre

(Total EF ≥ 5) CF = Hacer cambios en proyecto ya que el riesgo de fracaso es alto.

Para el caso del módulo hay un total de 2 que cumplen con las condiciones planteadas por lo que el Factor de conversión es 20 horas – hombre.

CF = 20 Horas-Hombres.

E = UCP * CF

E = 17.54676 * 20 Horas-Hombres.

E = 350.9352

Por lo que el esfuerzo en el desarrollo de las funcionalidades de los casos de uso es 350.9352 horas – hombre. Se considera que este esfuerzo representa un porcentaje del esfuerzo total del proyecto, de

acuerdo a los valores porcentuales de la tabla para la distribución del esfuerzo entre las diferentes actividades de un proyecto se obtiene:

Actividad	Porcentaje	Horas-Hombre
Análisis	15%	87
Diseño	5%	29
Programación	60%	350
Prueba	10%	58
Sobrecarga(otras actividades)	10%	58
Total	100%	583

Tabla 5.5 Cálculo del Factor de Conversión (CF).

Esfuerzo Total (horas-hombres) 583
Si una personas trabaja 20 horas promedio en una semana el proyecto se puede terminar en Aproximadamente 29 semanas, que equivale a 7 meses de trabajo.
Pero teniendo en cuenta que:
CH (Cantidad de hombres): 1 ----- Salario promedio mensual: $50
CHM (Costo por hombre/mes) = CH * Salario Promedio = 1 * 50 = $50 / mes
Costo = CHM * ET (meses) /CH = 50 * 23/1 = $1150
Tiempo = ET (meses)/ CH = 7 / 1 = 7 meses.
A partir de los resultados obtenidos se puede demostrar que con un hombre trabajando en la herramienta, esta se puede desarrollar en aproximadamente en 7 meses para un costo total asociado de $1150.

Conclusiones.

En este capítulo se realizó el estudio de factibilidad, analizando el esfuerzo, costo del proyecto, los beneficios tangibles e intangibles y el análisis de costo, teniendo en cuenta en este último los beneficios, costo y tecnología, deduciendo la factibilidad de la realización de la aplicación.
Este sistema aportará beneficios a la UCI y Segurmática, específicamente al laboratorio de análisis de muestras malignas.

Conclusiones Finales.

Como resultado del trabajo de diploma se ha investigado sobre el análisis de los programas malignos y se ha obtenido una herramienta capaz de automatizar de manera integrada un conjunto de herramientas y acciones que realizan diariamente los analistas del laboratorio antivirus de Segurmática la cual permite mejorar la eficiencia en el trabajo, permitiendo obtener resultados con calidad en un menor tiempo.

Recomendaciones.

Se recomienda ampliar la herramienta de forma tal que sea capaz de automatizar el proceso de análisis en varias computadoras conectadas en red, en ambiente de Windows, así como la posibilidad de brindar una propuesta de plantilla de descontaminación compleja para programas malignos específicos, cuando sea necesario.

- ✓ Elaborar una versión portable para el ambiente de Linux, con el fin de automatizar el análisis de códigos malignos en esta plataforma.
- ✓ Mejorar la información que se brinda al usuario como parte de la descripción del funcionamiento del código maligno, creando textos homogéneos claros y educativos.
- ✓ Continuar analizando el comportamiento de la herramienta en ambiente comprometido por un programa maligno.
- ✓ Investigar sobre el uso de herramientas alternativas que permitan obtener información sobre el funcionamiento de códigos malignos ante desactivaciones o mal funcionamiento de las empleadas en la actualidad en el proceso de análisis.

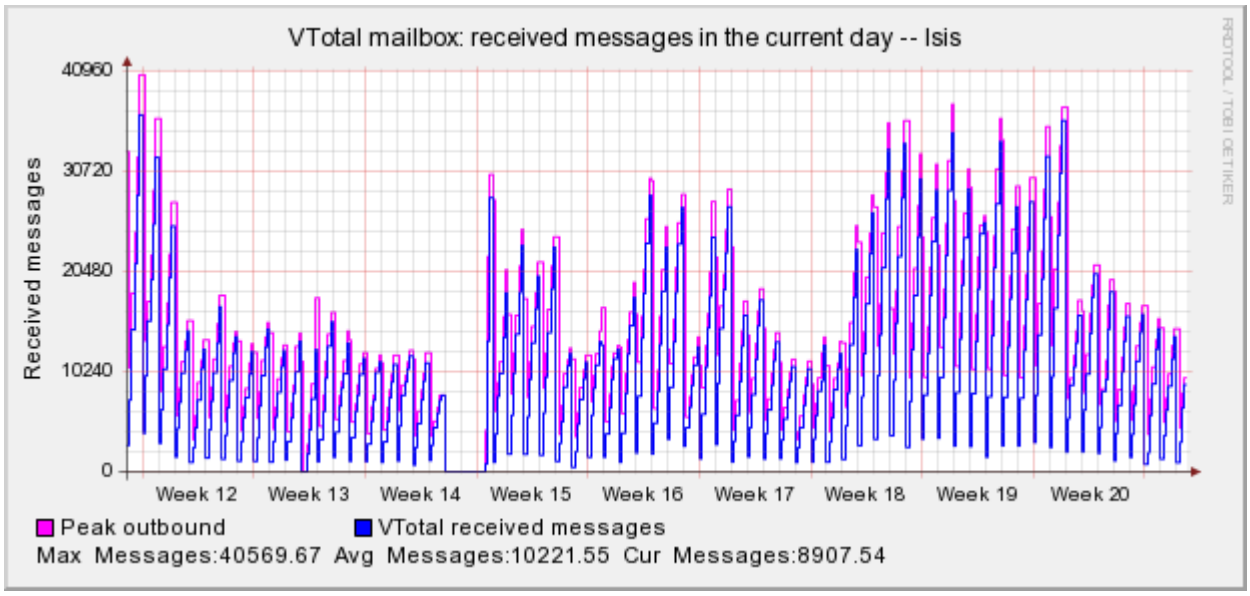
Referencias bibliográficas.

1. **HAWKING, STEPHEN.** ¿Juega Dios a los dados? *http://ciencia.astroseti.org.* 2005.
2. **Guadis, Edgar.** Segurmática. [En línea] 1 de 3 de 2002. [Citado el: 2 de 1 de 2009.] http://www.segurmatica.co.cu/.
3. **Neitzel, Michael St.** Sunbelt Malware Research Labs. [En línea] 1 de 2 de 2009 . [Citado el: 1 de 3 de 2009.] http://www.sunbeltsecurity.com.
4. **Kaspersky, Eugene.** Kaspersky Lab Web. [En línea] 3 de 1997. [Citado el: 1 de 4 de 2009.] http://www.kaspersky.com.
5. **ESET.** PC a Salvo. [En línea] 2008. [Citado el: 12 de 2 de 2009.] http://www.pcasalvo.com/.
6. **ThreatExpert.** ThreatExpert. [En línea] 2007. http://www.threatexpert.com.
7. **Anubis.** Analyzing Unknown Binaries. [En línea] 3 de 1 de 2005. [Citado el: 1 de 2 de 2009.] http://anubis.iseclab.org/.
8. Descripciones de Programas Malignos. [En línea] 2007. http://alerta-antivirus.inteco.es/.

Bibliografía

HAWKING, STEPHEN. ¿Juega Dios a los dados? *http://ciencia.astroseti.org.* 2005.
Guadis, Edgar. Segurmática. [En línea] 1 de 3 de 2002. [Citado el: 2 de 1 de 2009.] http://www.segurmatica.co.cu/.
Neitzel, Michael St. Sunbelt Malware Research Labs. [En línea] 1 de 2 de 2009 . [Citado el: 1 de 3 de 2009.] http://www.sunbeltsecurity.com.
. **Kaspersky, Eugene.** Kaspersky Lab Web. [En línea] 3 de 1997. [Citado el: 1 de 4 de 2009.] http://www.kaspersky.com.
ESET. PC a Salvo. [En línea] 2008. [Citado el: 12 de 2 de 2009.] http://www.pcasalvo.com/.
ThreatExpert. ThreatExpert. [En línea] 2007. http://www.threatexpert.com.
Anubis. Analyzing Unknown Binaries. [En línea] 3 de 1 de 2005. [Citado el: 1 de 2 de 2009.] http://anubis.iseclab.org/.
Descripciones de Programas Malignos. [En línea] 2007. http://alerta-antivirus.inteco.es/.

Anexo_1: Mensajes con muestras de códigos malignos, reportados internacionalmente, recibidos diariamente en las últimas semanas.



Anexo_2: Estadísticas Programas Malignos en Cuba hasta 15 de mayo 2009.

Clasificación	2002	2003	2004	2005	2006	2007	2008	2009
VIRUS	9	11	12	8	10	9	17	1
Troyanos	23	62	146	192	175	209	1283	401
GUSANOS	25	43	126	91	76	72	421	108
JOKE	6	0	9	3	1	0	1	0
EXPLOIT	0	0	12	2	6	0	0	0
Total:	63	116	305	296	268	290	1725	510

Anexo_ 3: Descripción textual de los casos de usos del Sistema.

Caso de Uso:	Ejecutar Muestra
Actores :	Sistema (Inicia)
Resumen :	El caso de uso se inicia cuando el Especialista ejecuta las muestras malignas.
Propósito:	Ejecutar las muestras de programas malignos.
Referencias :	
Precondiciones :	Las muestras deben estar en la PC donde se va a hacer el análisis y se debe haber hecho un análisis previo de la PC en su estado limpio.
Flujo de Eventos	
Acción del Actor	Respuesta del Sistema
1. Selecciona cada una de las muestras	1.1 El en caso de ser una dll debe mostrar la opción de ejecutarla con run32dll.
2 .Espera 5 min y reinicia la PC.	2.1 El sistema adiciona la ejecución posterior al registro.
Flujo Alternos	
1.1. En caso de que la muestra maligna reinicie la PC cuando inicie se debe esperar entonces los 5 min para que haga todos sus efectos.	

Caso de Uso:	Analizar Muestra
Actores :	Sistema (Inicia)
Resumen :	El caso de uso se inicia cuando comienza el proceso de análisis de la muestra maligna.
Propósito:	Obtener los efectos de la Muestra maligna en el registro.
Referencias :	
Precondiciones :	El análisis debe de iniciarse sin haber ejecutado las muestras malignas.

Flujo de Eventos	
Acción del Actor	Respuesta del Sistema
1.1 El sistema Inicia la ejecución del proceso de análisis.	1.1 Se llama al caso de uso incluido “Analizar Llaves” 1.2 Se llama al caso de uso incluido “Analizar Valores”. 1.3 Se llama al caso de uso incluido “Analizar Servicios”. 1.4 Se llama al caso de uso incluido “Analizar Procesos” 1.5 Se llama al caso de uso incluido “Analizar puertos” 1.6 Se llama al caso de uso incluido Ejecutar Muestras. 1.7 Se hace referencia a 1.1, 1.2, 1.3, 1.4, 1.5 después de haberse reiniciado la PC. 1.8 Se Busca en ficheros creados y modificados que no se encuentren en: "Ficheros_no_controlados" ni en "Carpetas_no_controladas". 1.9 Buscar en los registros: (El objetivo es localizar

	llaves de registros no tenidas en cuenta) 1.9.1) las referencias a ellos: - Si se encuentran en: "Llaves_conflictivas" copiar los datos de registros y ficheros en fichero "interesante". - Si no se encuentran en: "Llaves_conflictivas", ni "Llaves_a_controlar", ni "Llaves_de_servicio", ni "Valores_a_controlar" copiar los datos de registros y ficheros a "interesante_registos_Ficheros". 1.10 Buscar en los registros referencias a otros ficheros ejecutables, localizándolos por la extensión (.EXE, .SCR, .PIF, .SYS, .BAT, .CMD, .OCX, .Dll, .VBE, .VBS, .HTA, .JS, COM, .HTML) (El objetivo es localizar ficheros ejecutables cuya creación o modificación no ha sido reportada por la herramienta empleada al efecto y son refrenciados desde el registro del Sistema) 1.10.1 las referencias a ellos: - Si se encuentran en: "Llaves_conflictivas" copiar los datos de registros y ficheros a "interesante"". - Si no se encuentran en: "Llaves_conflicticas", ni "Llaves_a_controlar", ni "Llaves_de_servicio", ni "Valores_a_controlar" copiar los datos de registros y ficheros en "interesante_registros_Ejecutables".

	1.11 Buscar en los registros referencias a otros ficheros no ejecutables, es decir, su extensión sea diferente a: .EXE, .SCR, .PIF, .SYS, .BAT, .CMD, .OCX, .Dll, .VBE, .VBS, .HTA, .JS, COM, .HTML) (El objetivo es localizar ficheros no ejecutables cuya creación o modificación no ha sido reportada por la herramienta empleada al efecto y son refrenciados desde el registro del Sistema)) 1.11.1 Las referencias a ellos: - Si se encuentran en: "Llaves_conflictivas" copiar los datos de registros y ficheros a "interesante"". - Si no se encuentran en: "Llaves_conflictivas", ni "Llaves_a_controlar", ni "Llaves_de_servicio", ni "Valores_a_controlar" copiar los datos de registros y ficheros en "interesante_registros_No_Ejecutables". 1.12 En el reporte del InCtrl: 1.12.1 No tener en cuenta: "Llaves_conflictivas", "Ficheros_no_controlados", "Carpetas_no_controladas". 1.12.2 Se tendrán en cuenta: 1.12.3) Llaves añadidas, que no se encuentren en "Llaves_conflictivas" y sí sean hijas de "Llaves_a_controlar" o de "Llaves_de_Servicios".

	- Dentro de ellas se tendrán en cuenta todos las valores que sean añadidos. 1.12.4 Valores añadidos que no se encuentren en "Llaves_conflictivas", ni en "Valores_no_controlados" y sí en "Valores_a_controlar". 1.12.5 La información referente de estos valores añadidos que no se encuentren en "Valores_a_borrar" se copirán en "interesante_valores_creados. 1.12.6 Valores modificados que no se encuentren en "Llaves_conflictivas, "ni en "Valores_no_controlados" y sí en "Valores_a_controlar". La información referente de estos valores modificados que no se encuentren en "Valores_a_restaurar" se copiarán en "interesante_valores_modificados". 1.12.7 Valores modificados que no se encuentren en "Llaves_conflictivas", "ni en "Valores_no_controlados" y sí en subllaves hijas de "Llaves_de_servicio" La información referente de estos valores modificados se copiará en "interesante_valores_servicios_modificados". 1.13 Copiar las muestras malignas hacia la carpeta Muestras Modificadas y Muestras adicionadas.

Caso de Uso:	Descontaminación Genérica.
Actores :	Sistema (Inicia)
Resumen :	El caso de uso se inicia después de haber terminado el proceso de análisis y se resume a comprobar si todos los efectos de la muestra maligna los cubre la descontaminación genérica.
Propósito:	Determinar la Suficiencia de la Descontaminación Genérica .
Referencias :	
Precondiciones :	El proceso de análisis debe de haber terminado.

Flujo de Eventos	
Acción del Actor	Respuesta del Sistema
1. Inicia el proceso de comprobación de la genérica.	1.1 Se trata de revisar información referente a los procesos relacionada con los ficheros creados o modificados 1.2 Lo obvio es el proceso aislado cargado con el mismo nombre del fichero 1.3 Proceso wscript.exe (Microsoft Windows Script Host) para los que la secuencia de comando en ejecución se corresponde con el programa maligno, el nombre del fichero aparece en la línea de comandos del proceso wscript.exe y tendrá esta manera $SYSTEMDIR$\wscript.exe $CURRENTMALWAREFILENAME$ 1.4 Proceso cscript.exe (Microsoft Windows Script Host) para los que la secuencia de comando en ejecución se corresponde con el programa maligno El nombre del fichero aparece en la línea de comandos del proceso wscript.exe

	$SYSTEMDIR$\cscript.exe $CURRENTMALWAREFILENAME$ 1.5 Procesos rundll32.exe (Service host) para los que la dll que se carga se corresponde con el programa maligno. El nombre del fichero aparece en la línea de comandos del proceso rundll32.exey tendrá la siguiente forma: $SYSTEMDIR$\rundll32.exe $CURRENTMALWAREFILENAME$ 1 1.6 Proceso svchost.exe (Service host) para los que la dll que se carga se corresponde con el programa maligno indica que se carga como dll $SYSTEMDIR$\svchost.exe $CURRENTMALWAREFILENAME$ 1.7 Procesos rundll32.exe (Service host) para los que la dll que se carga se corresponde con el programa maligno. $SYSTEMDIR$\rundll32.exe $CURRENTMALWAREFILENAME$ 1.9 Se comprueban valores desconocidos del registro a borrar cuyo dato es el nombre del archivo. 1.10 Se comprueban valores conocidos del registro a borrar cuyo dato es el nombre del archivo. 1.11 Se comprueban valores de registro a restaurar en los que en el dato aparece el nombre del archivo. 1.12 Se comprueban valores desconocidos del registro a borrar cuyo nombre es el del archivo 1.13 Se comprueban valores del registro a restaurar en los que en el dato aparece el nombre del archivo. 1.14 Se comprueban valores de Registro conocidos a limpiar cuyo dato tiene el nombre del archivo. 1.15 Se comprueban valores del Registro Predeterminado a

	limpiar dentro de HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\numero de CLSD variable\InprocServer32 // Aqui se contemplan: Generica_Ficheros 1.16 Se comprueban llaves de registros a Borrar cuyo dato tiene el nombre del archivo. 1.17 Se comprueban valores del Registro a restaurar relacionadas con Políticas. 1.18 Se comprueban valores del registro a borrar relacionadas con Políticas. 1.19 Se comprueban ficheros creados o alterados por el código maligno que se encuentren en Generica_Ficheros.
Flujo Alternos	

Caso de Uso:	Generar Descripciones.
Actores :	Sistema (Inicia)
Resumen :	El caso de uso se inicia después de haber realizado el análisis y se tiene un resumen de los efectos de la Muestras maligna por cada uno de los efectos se realiza una descripción si la descripción no aparece en la base de conocimientos de la genérica se hace un reporte con los efectos no encontrados para una actualización de la Base de conocimiento por parte de los especialistas.
Propósito:	Obtener descripciones de los efectos de la muestra maligna.
Referencias :	
Precondiciones :	El análisis de la muestras ya se debe de haber realizado.

Flujo de Eventos	
Acción del Actor	Respuesta del Sistema
1. El sistema inicia la generación de las descripciones.	1.1. El busca en las base de conocimiento las referencias a las descripciones de cada uno de los efectos de la muestra maligna sobre el sistema a partir del resumen obtenido del análisis.

Flujo Alternos	
Para el caso de que no se encuentren las descripciones el sistema guarda un reporte de las descripciones no encontradas en la base de conocimiento.	
Pos condiciones :	
Prioridad:	Primario

Caso de Uso:	Analizar Llaves
Actores :	Sistema (Inicia)
Resumen :	El caso de uso se inicia antes de ejecutar la muestra maligna y después de haberla ejecutado y reiniciado la PC.
Propósito:	Obtener los cambios en las llaves del registro antes y después de haber ejecutado la muestra maligna.
Referencias :	
Precondiciones :	Las muestras no se deben de haber ejecutado.

Flujo de Eventos	
Acción del Actor	Respuesta del Sistema
1. Ejecuta el IntCtrl(/G) para iniciar el análisis del estado de la PC limpia en cuanto a registro y a ficheros en disco.	1.2 Al ejecutar el IntCtrl comienza en análisis del registro este proceso debe de realizarse antes y después de haber ejecutado la muestra maligna obteniendo para obtener una visión del estado del registro con la PC en un estado libre de contaminación alguna. 1.3 El análisis de llaves del registro debe tenerse en cuenta las llaves adicionadas modificadas y las eliminadas. 1.4 Después de haberse tomado los 2 estados del registro de la PC primeramente limpia y después de haber sido contaminada se comparan los 2 resultados y se guardan en el reporte EXTRARPT.txt
Flujo Alternos	

Pos condiciones :	
Prioridad:	Primario

Caso de Uso:	Analizar Valores
Actores :	Sistema (Inicia)
Resumen :	El caso de uso se inicia antes de ejecutar la muestra maligna y después de haberla ejecutado y reiniciado la PC.
Propósito:	Obtener los cambios en los valores en la PC antes y después de haber ejecutado la muestra maligna.
Referencias :	
Precondiciones :	Las muestras no se deben de haber ejecutado.

Flujo de Eventos	
Acción del Actor	Respuesta del Sistema
1. Ejecuta el IntCtrl(/G) para iniciar el análisis del estado de la PC limpia en cuanto a registro y a ficheros en disco.	1.2 Al ejecutar el IntCtrl comienza en análisis del registro este proceso debe de realizarse antes y después de haber ejecutado la muestra maligna obteniendo para obtener una visión del estado del registro con la PC en un estado libre de contaminación alguna en el análisis debe de tenerse en cuenta los valores adicionado modificados y lo valores cambiados. 1.2 Después de haberse tomado los 2 estados del registro de la PC primeramente limpia y después de haber sido contaminada se comparan los 2 resultados y se guardan en el reporte EXTRARPT.txt
Flujo Alternos	

Pos condiciones :	
Prioridad:	Primario

Caso de Uso:	Analizar Ficheros Adicionados y Modificados.
Actores :	Sistema (Inicia)
Resumen :	El caso de uso se inicia antes de ejecutar la muestra maligna y después de haberla ejecutado y reiniciado la PC se analizan los cambios que hizo la muestra maligna en el disco en cuanto a ficheros adicionados y a modificados.
Propósito:	Obtener los cambios en el disco antes y después de haber ejecutado la muestra maligna.
Referencias :	
Precondiciones :	Las muestras no se deben de haber ejecutado.

Flujo de Eventos	
Acción del Actor	Respuesta del Sistema
1. Ejecuta el IntCtrl(/G) para iniciar el análisis del estado de la PC limpia en cuanto a registro y a ficheros en disco.	1.1 Al ejecutar el IntCtrl comienza en análisis de los cambios en el disco este proceso debe de realizarse antes y después de haber ejecutado la muestra maligna obteniendo para obtener una visión del estado del disco y los cambios que ocurrieron fundamentalmente los ficheros que fueron adicionado y los que fueron modificados 1.2 Después de haberse tomado los 2 estados del disco de la PC primeramente limpia y después de haber sido contaminada se comparan los 2 resultados y se guardan en el reporte EXTRARPT.txt
Flujo Alternos	
Pos condiciones :	
Prioridad:	Primario

Caso de Uso:	Analizar Servicios.
Actores :	Sistema (Inicia)
Resumen :	El caso de uso se inicia antes de ejecutar la muestra maligna y después de haberla ejecutado y reiniciado la PC se analizan los servicios y se inicia un monitoreo de los cambios que van ocurriendo.
Propósito:	Obtener los cambios en los servicios como resultado de la comparación de cómo estaban antes y como están después y de su monitoreo.
Referencias :	
Precondiciones :	Las muestras no se deben de haber ejecutado para iniciar el proceso.

Flujo de Eventos	
Acción del Actor	Respuesta del Sistema
1. Se ejecuta el script de VBS que hace un listado de los servicios.	1.1 Al ejecutarse el script se hace una consulta al sistema para que devuelva todos los servicios que existen en la PC en ese momento, el proceso se repite después de haberse ejecutado la muestra y reiniciado la PC. 1.2 Después de que la lista de servicios este completa en un primer momento se comienza un monitoreo de la creación de los servicios en el sistema operativo los resultados de ambos reportes se guardan como reporte_monitoreo.txt reporte_servicios_antes.txt reporte_servicios_despues.txt
Flujo Alternos	

Pos condiciones :	
Prioridad:	Primario

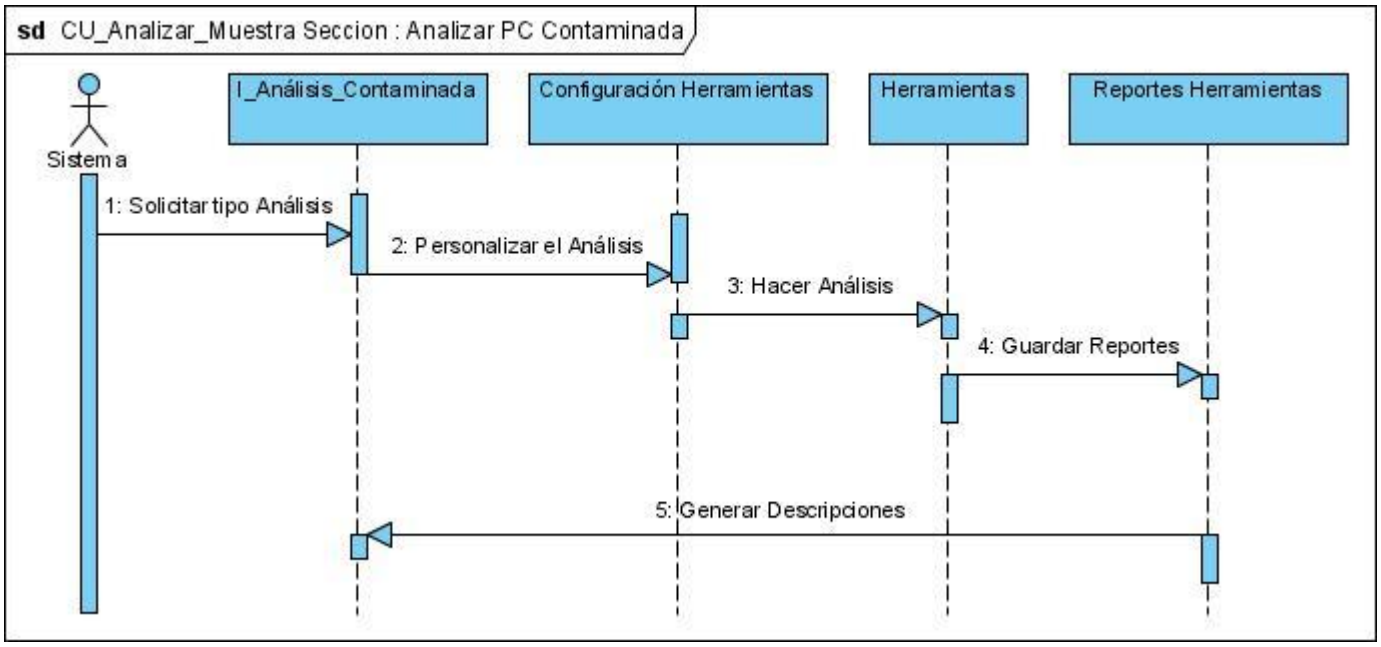
Caso de Uso:	Analizar Procesos.
Actores :	Sistema (Inicia)
Resumen :	El caso de uso se inicia antes de ejecutar la muestra maligna y después de haberla ejecutado y reiniciado la PC se analizan los procesos y se inicia un monitoreo de los cambios que van ocurriendo.
Propósito:	Obtener los cambios en los servicios como resultado de la comparación de cómo estaban antes y como están después y de su monitoreo.
Referencias :	
Precondiciones :	Las muestras no se deben de haber ejecutado para iniciar el proceso.

Flujo de Eventos	
Acción del Actor	Respuesta del Sistema
1. Se ejecuta el script de VBS que comienza el monitoreo de los procesos.	1.1 Por cada proceso que se cree en el sistema se llevara un registro con el nombre del procesos el id (PID). 1.2 Después de que la muestras se ejecute y la PC se halla reiniciado lista de procesos de almacenan en el directorio reportes de la herramienta encargada de crear un resumen de la descontaminación. 1.3 Después de que se halla reiniciado la PC también se hace un análisis de los procesos en memoria con un conjunto de herramienta que varían según la muestra maligna.
Flujo Alternos	
Pos condiciones :	

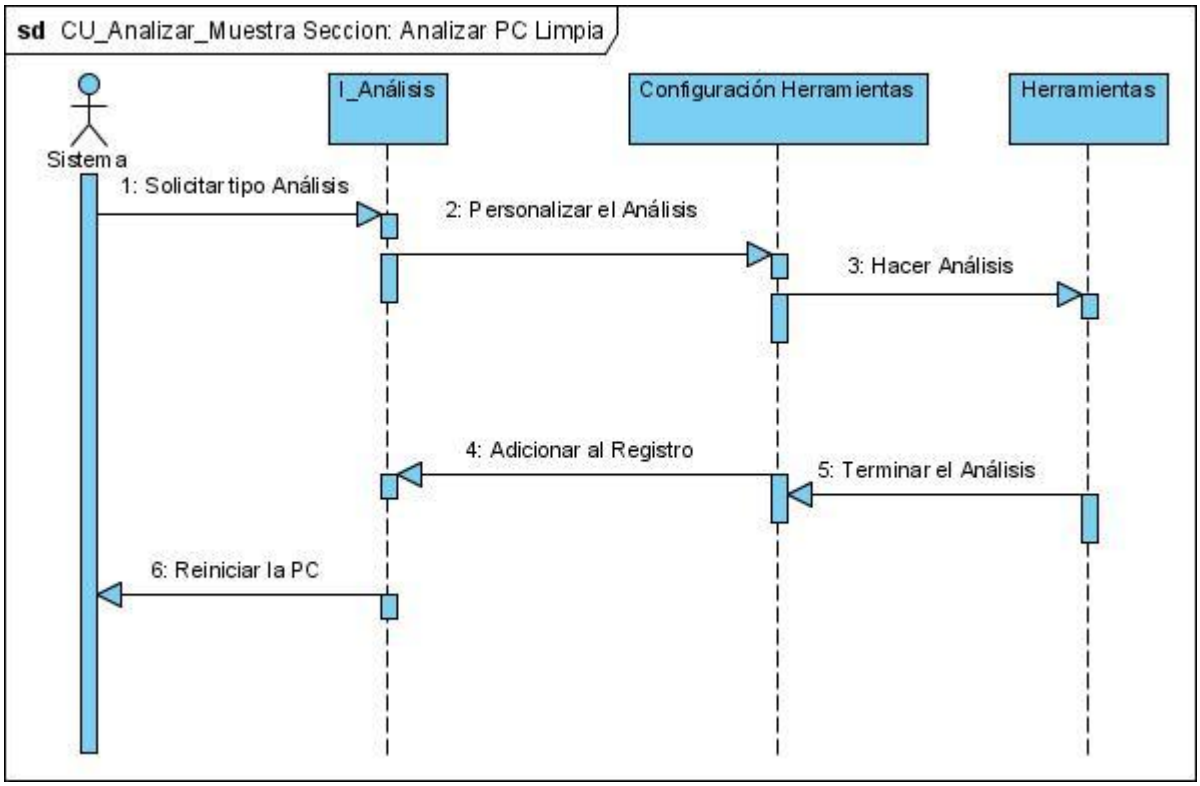
Prioridad:	Primario

Caso de Uso:	Analizar Puertos.	
Actores :	Sistema (Inicia)	
Resumen :	El caso de uso se inicia después de haber ejecutado la muestra maligna y reiniciado la PC se analizan los puertos y las peticiones.	
Propósito:	Obtener las peticiones a los distintos puertos y determinar si están relacionado con la muestra maligna.	
Referencias :		
Precondiciones :	Las muestras se deben de haber ejecutado para iniciar el proceso.	
Flujo de Eventos		
Acción del Actor	Respuesta del Sistema	
1. Se ejecuta las herramientas que consultan las peticiones por puertos.	1.1 Por cada puerto que haga peticiones en el sistema se llevara un registro con el id (PID) del procesos que hace la petición. 1.2 Después de que la muestras se ejecute y la PC se halla reiniciado lista de peticiones se almacenan en el directorio reportes de la herramienta encargada de crear un resumen de la descontaminación.	
Flujo Alternos		
Pos condiciones :		
Prioridad:	Primario	

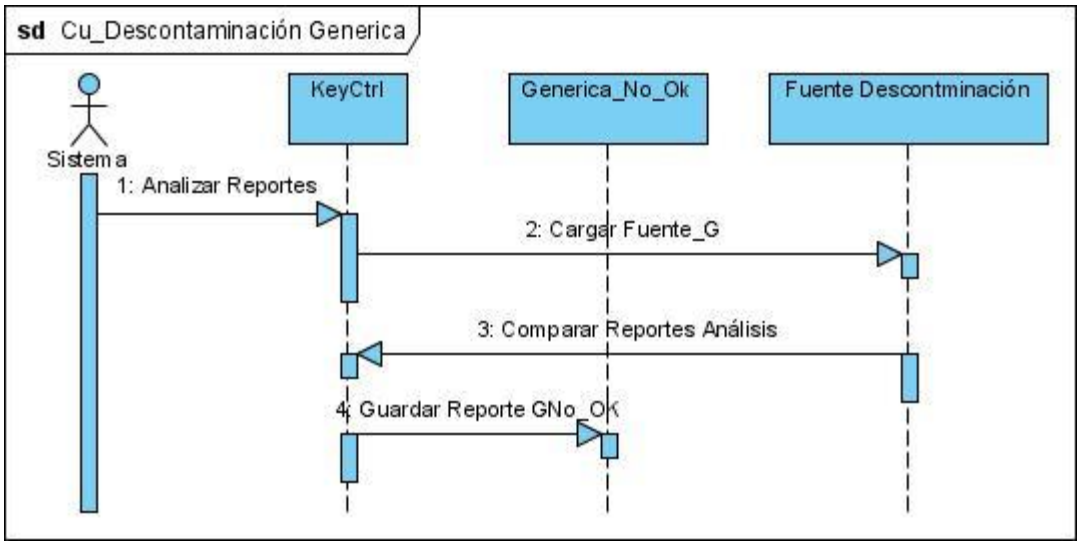
Anexo_6: Diagramas de secuencia.



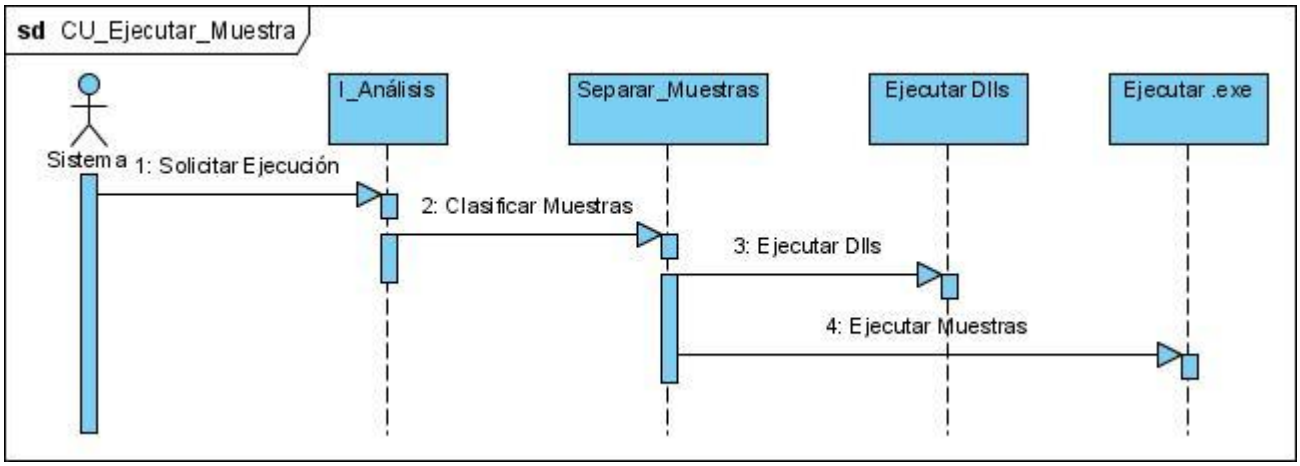
Anexo_ 4: Diagrama de Secuencia Cu_Analizar_Muestra_Pc_Contaminada



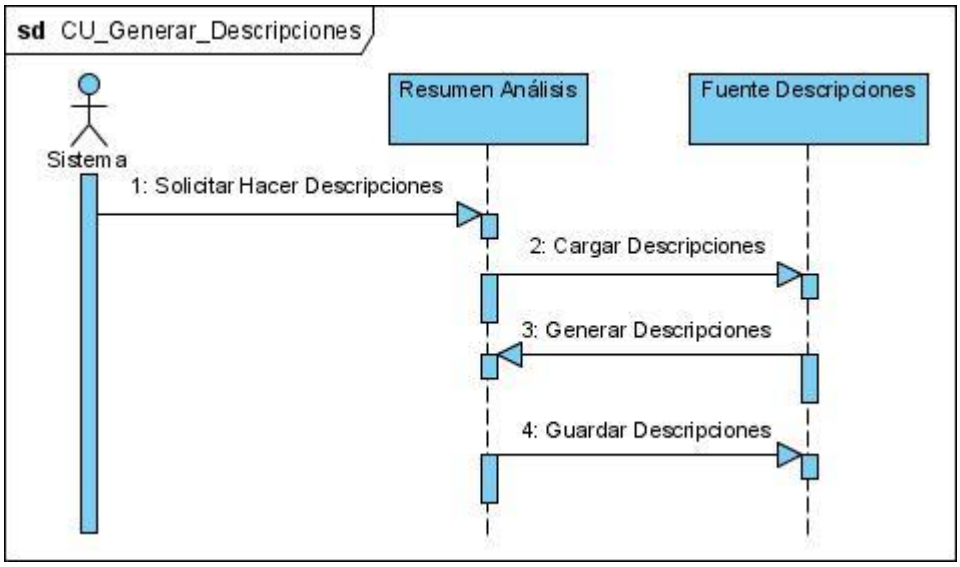
Anexo_ 5: Diagrama de Secuencia Cu_Analizar_Muestra_Pc_Limpia



Anexo_ 6: Diagrama de Secuencia Cu_Descontaminación_G



Anexo_ 7: Diagrama de Secuencia Cu_Ejecutar_Muestra



Anexo_ 7: Diagrama de Secuencia Cu_Generar_Descripciones

Glosario de términos.

API.

Es la abreviatura de Aplication Programming Interface. Un API no es más que una serie de servicios o funciones que el Sistema Operativo ofrece al programador, como por ejemplo, imprimir un carácter en pantalla, leer el teclado, escribir en un fichero de disco, etc.

PM.

Programas Malignos.

C++

Es un lenguaje de programación diseñado a mediados de los años 1980 por Bjarne Stroustrup. La intención de su creación fue el extender al exitoso lenguaje de programación C con mecanismos que permitan la manipulación de objetos. En ese sentido, desde el punto de vista de los lenguajes orientados a objetos, el C++ es un lenguaje híbrido.

AutoIt

Es un sistema de programación gratuito, código libre y abierto, tipo Visual Basic usando más opciones, un Visual Basic Killer. Además de automatizar tareas usando combinaciones de teclas simuladas, clic de ratón, comandos de Windows y ficheros Script cuenta con ActiveX, dlls y plugins, como cualquier lenguaje dinámico al estilo PHP.

BAT

Extensión de un fichero formado por un lote (batch) de órdenes de Dos .

VBScript:

Lenguaje de programación para WWW desarrollado por Microsoft. VBScript y JavaScript, de Netscape, son muy similares.