Universidad de las Ciencias Informáticas

Facultad 2

Título: "Sistema de Reportes de Programas Malignos capturados en Segurmática".

Trabajo de Diploma

Para optar por el título de Ingeniería en Ciencias Informáticas

Autores: Jean Reina Lugo

Liliam Balmaseda Acosta

Tutor: Orlando Cabrera Báez

Ciudad de la Habana, 2007.

"Año 49 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 firmo la presente a los ____ días del mes de _____ del año _____.

_____ _____

Jean Reina Lugo Liliam Balmaseda

_____

Orlando Cabrera Báez

Opinión del usuario del Trabajo de Diploma

El Trabajo de Diploma, titulado Sistema de Reportes de Programas Malignos capturados en Segurmática, 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: Sistema de Reportes de Programas Malignos capturados en Segurmática.

Autor: Jean Reina Lugo
Liliam Balmaseda Acosta

El tutor del presente Trabajo de Diploma considera que durante su ejecución el estudiante mostró las cualidades que a continuación se detallan.

Por todo lo anteriormente expresado considero que los estudiantes están aptos para ejercer como Ingenieros en Ciencias Informáticas; y propongo que se le otorgue al Trabajo de Diploma la calificación de _____.

_____ _____

Orlando Cabrera Báez Fecha

Firma

Agradecimientos

Agradecemos a nuestros padres por todo su apoyo en los momentos difíciles y por dedicarnos su vida entera.

A nuestros hermanos por querernos tanto, darnos la fuerza y el deseo de seguir adelante para cumplir nuestras metas.

A los amigos de la universidad: Yeitel, Clavijo, Arce, Yosleinis, Zadi, Hanillilian, Lisbettes, Yamisleidis, Liudmila, Líber, Roberto Carlos, Yan Ayata, Yonger Cala, Wiliam Sánchez, por permitirnos conocerlos y ser parte de su vida. Por ayudarnos y estar con nosotros en muchas ocasiones en el transcurso de la carrera.

A Pavel, Giorbi, Juan Carlos, Yuri, Elvis que emplearon parte de su tiempo para brindarnos su colaboración.

Al tutor por asesorarnos a lo largo de la tesis y ayudarnos.

A nuestras familias que nos ofrecieron ayuda en tantas ocasiones.

A nuestros profesores.

A los compañeros de aula.

A Fidel por hacer posible este sueño.

Agradecemos a todos por ayudarnos a lograrlo.

Dedicatoria

Dedicamos este trabajo a nuestros padres y hermanas.

A nuestra familia en general que de una forma u otra nos han apoyado todo este tiempo.

A nuestros compañeros.

Pensamiento

"En Cuba nadie ha hecho tanto en tan poco tiempo"

Fidel Castro Ruz.

Resumen

La implantación de nuevas tecnologías ha sido clave en la transformación de la sociedad y de los sistemas que se utilizan en las empresas. Debido a esto las instituciones desarrollan aplicaciones que le permitan manipular grandes volúmenes de información.

El presente trabajo consiste en un sistema para generar reportes con información los de ataques y amenazas que se obtienen a través de herramientas en Segurmática.

En esta empresa se utilizan herramientas que contribuyen en gran medida, a la captura de datos relacionados con los programas malignos y ataques de intrusos. Datos que luego de ser obtenidos van a ser analizados .A pesar de esto la empresa tiene la necesidad de desarrollar un sistema que le permita ver reportes de la información obtenida de una forma legible, puesto que actualmente el proceso de análisis de la información se lleva a cabo mediante la lectura de los ficheros Log que tienen mucha información irrelevante que no es usada por los analistas, y que hace que el proceso sea lento y muy engorroso.

El objetivo fundamental de este trabajo es desarrollar una aplicación Web, segura y de interfaz amigable, que garantice en todo momento, ofrecer a los usuarios autorizados una información detallada, para con esto hacer más cómodo el trabajo de estas personas.
En el documento se muestra un estudio de las herramientas mas apropiadas para el desarrollo de la aplicación y todos los resultados de cada flujo por los que transitó nuestro sistema hasta su culminación.

Índice

Introducción

Dentro del inminente mundo de la Informática, la seguridad juega un papel fundamental ya que esta, desarrolla técnicas para proteger los equipos informáticos individuales y los conectados en una red, frente a daños accidentales o intencionados.

Lo más importante es que un sistema se pueda definir como seguro y para esto debemos dotar de estas características al mismo tiempo:

- Integridad: La información no puede ser modificada por quién no está autorizado
- Confidencialidad: La información solo debe ser legible para los autorizados
- Disponibilidad: Debe estar disponible cuando se necesita

En los equipos informáticos sin darnos cuenta podemos tener muchas vulnerabilidades .Una vulnerabilidad es un fallo que compromete la seguridad de un programa o sistema. Generalmente son errores de programación que pueden ser utilizados para ejecutar código arbitrario, detener el sistema o aprovecharse del mismo para sacar cualquier tipo de beneficio.

Por esto se puede decir que debido a una vulnerabilidad, estamos expuestos a un riesgo .Existen varios riesgos tales cómo: ataque de virus, códigos malignos, gusanos, caballos de Troya y hackers.

Con el surgimiento diario de nuevos ataques y programas malignos la Seguridad informática utiliza disímiles herramientas para detectar y analizar estos programas, con el objetivo de atacarlos antes de que causen daños. De estas herramientas podemos mencionar a Honeypots que como su traducción indica son “tarros de miel para atraer abejas”, pero en el terreno de la seguridad informática, son servidores, que simulan tener muchas vulnerabilidades para atraer ataques informáticos y registrarlos para su posterior estudio. Estos se usan sobre todo para estudiar ataques en una red, son también muy usados para la detección temprana de ataques. Pero hay que tener bien claro que un Honeypot no se puede usar como elemento de protección contra ataques

Dentro de los Honeypots están los Honeynet que son un tipo especial de alta interacción, los cuáles actúan sobre una red entera, diseñada para ser atacada y recobrar así mucha más información sobre

posibles atacantes. Se usan equipos reales con sistemas operativos reales y corriendo aplicaciones reales .Este tipo de Honeypots se usa principalmente para la investigación de nuevas técnicas de ataque y para comprobar el modo en que operan los intrusos.

Como experiencia de lo antes planteado se tiene el caso de la empresa de Segurmática en la que se utilizan varias herramientas que capturan de forma automática programas malignos y ofrecen datos para el análisis de los ataques recibidos durante su propagación. Hay ataques que no propagan programas malignos y ataques que si. Los ataques que propagan programas malignos pueden no tener efecto en el sistema que funciona en Segurmática y en ese caso se necesita actualizarlo. Para actualizarlo es necesario tener información relativa al ataque. Por esta razón es necesario tener una correlación de los flujos de tráfico de los ataques y los efectos que este causo en el sistema de captura.
En la actualidad los datos relativos a las capturas deben analizarse desde ficheros log, donde existe información irrelevante que retrasa el proceso de análisis de los mismos. Con el surgimiento diario de nuevos ataques y programas malignos es necesario realizar un sistema que genere información con estos datos devueltos por los sistemas antes mencionados, por ejemplo, distintos tipos de reportes, que permita a los analistas de seguridad informática dar una repuesta eficiente a las incidencias. Dicha respuesta va desde el reporte y aviso de redes comprometidas hasta la actualización del sistema para que capture nuevos programas malignos automáticamente.

Por esta razón se plantea el siguiente problema:
¿Cómo procesar la información más específica referente a los ataques y programas malignos capturados, mediante la implementación de un sistema de reportes?

El objeto de estudio lo constituye el proceso de reportes en la empresa de Segurmática, profundizando aún más en los procesos de reportes de los datos referentes a las capturas de programas malignos, en dicha empresa.

El objetivo general que se ha trazado, es un sistema diseñado para obtener reportes .Que estos reportes den una información clara del estado de los ataques para que el usuario final solucione el problema anteriormente planteado .Como objetivo específico tenemos, la obtención de estos reportes mostrando información de las capturas de acuerdo a sus fechas .así como la creación de gráficas que mostrarán un estado general de la información que ha sido capturada.

Para el cumplimiento de los objetivos trazados, se han elaborado un conjunto de tareas:

- Estudio de la metodología que vamos a utilizar para realizar el modelamiento del sistema.
- Estudio de las tecnologías más actuales utilizadas para el diseño y desarrollo de aplicaciones Web en Cuba y el mundo.
- Investigación de la existencia de sistemas que capturan información de programas malignos para conocer más detalladamente del contenido de los datos capturados.
- Estudio del funcionamiento actual del proceso de análisis de la información.

EL documento ha quedado estructurado de la siguiente manera:

Capitulo 1: Fundamentación teórica, se mostrará un estudio de los sistemas que simulan vulnerabilidades con el objetivo de capturar información de ataques, para comprender como estos funcionan actualmente en el mundo. También se presentarán las herramientas y tecnologías utilizadas, para la solución del problema planteado.
Capitulo 2: Se representan las características del sistema al que nos enfrentaremos, se explicara en este las funcionalidades del sistema y los requisitos que deben tenerse en cuenta.
Capitulo 3: En el análisis y diseño del sistema realizaremos diagramas que mostrarán los conceptos en un dominio del problema, los objetos que participan en una interacción. Además los diagramas para diseñar la base de datos.
Capitulo 4: En el flujo de implementación representamos, el diagrama de despliegue del sistema que muestra la situación física de la aplicación, así como los diagramas de componentes que representan cada parte modular del sistema y sus relaciones entre ellas.

Capitulo 5: Planificación, se realiza un análisis del costo de la misma, para hacer un estudio sobre la factibilidad de nuestra aplicación .Para ello aplicaremos la técnica de estimación por caso de uso que nos servirá para calcular el costo, tiempo de desarrollo, el esfuerzo y la cantidad de personas que se necesitan para desarrollar el sistema.

Capítulo 1. Fundamentación Teórica.

1.1 Introducción.

En el presente capítulo, se describen los conceptos principales, relacionados con el tema de la captura de información de programas malignos. Se explicará como algunos sistemas simulan vulnerabilidades, para atraer a los atacantes con el objetivo de aprender cuánto sea posible de las amenazas y del comportamiento de los mismos. Además se explicará una herramienta utilizada a nivel internacional para el proceso de captura de estos programas que contienen código maligno. Por último se presentan las diferentes herramientas y tecnologías utilizadas para darle solución al problema planteado.

1.2 Sistemas y herramientas de captura de programas malignos.

1.2.1 Honeypots.

Los Honeypots, como su traducción al castellano indica son “tarros de miel para atraer abejas”, pero en el terreno de la seguridad informática, son servidores que emulan tener muchas vulnerabilidades para atraer ataques informáticos y registrarlos para su posterior estudio. [1]

Los Honeypots se usan sobre todo para estudiar ataques en una red. Los tarros de miel son también muy usados para la detección temprana de ataques. Ellos pueden distraer a los atacantes de las máquinas más importantes del sistema, y advertir rápidamente al administrador del sistema de un ataque, además de permitir un examen en profundidad del atacante, durante y después del ataque al honeypot. Una de las herramientas que aplica en el concepto de Honeypot es Nepenthes que captura información que es almacenada en ficheros log.

Clasificación de los Honeypots

1ra Clasificación. Es basada en la funcionalidad que se desea asignar al Honeypot:

- **Honeypot de producción:** Su principal objetivo es el de mitigar el riesgo de un ataque informático a la red de una institución o empresa. De esta forma, un Honeypot de producción simula diferentes servicios con el único objetivo de ser atacado. Una vez descubierto el atacante se “avisa” al resto del sistema para que tome las medidas oportunas (denegar

cualquier acceso con un origen determinado, limitar las capacidades de un servicio, paralizar varios servicios momentáneamente.

- **Honeypot de investigación:** Su principal objetivo es recoger información sobre los distintos atacantes, así como sus comportamientos y técnicas asociados. También han sido diseñadas para ser comprometidas al igual que los de producción, sin embargo no añaden ninguna capacidad extra de seguridad o mitigación de los ataques.
  Suelen ofrecer servicios reales (no los simula) e incluso pueden llegar a permitir que el atacante tome el control total de la máquina [2]

2da Clasificación. Hace referencia al grado de compromiso o riesgo que esta introduzca en nuestra red:

- **Compromiso bajo:** El sistema Honeypot simplemente simula la existencia de algún servicio común (WWW, FTP, TELNET...) escucha y almacena todas las peticiones recibidas en ficheros de logs. De esta forma, se tiene un sistema totalmente pasivo que simplemente registra peticiones de acceso, ya que no responde a ninguna de ellas o interacción con el atacante. El riesgo que introduce esta variante es mínimo puesto que el atacante nunca podrá acceder a la máquina, lo que hace perder la posibilidad de investigar y analizar sus técnicas [2]

- **Compromiso medio:** En este grupo los sistemas simulan la existencia de uno o varios servicios de forma más sofisticada. Con este tipo de Honeypots se pretende captar la atención del atacante y lograr un grado mayor de interacción, que permita analizar mínimamente el comportamiento del atacante.
  El grado de riesgo aumenta moderadamente, ya que por un lado el servicio sigue siendo una simulación, lo que permite tener acotado/enjaulado la interacción entre el atacante y el servicio. Por otro lado, si existe un fallo en la implementación del servicio simulado, el atacante puede aprovecharlo para atacar el sistema real.

- **Compromiso alto**: En este grupo se encuentran aquellos sistemas que no simulan diferentes servicios, sino que utilizan un entorno real con servicios de verdad.

Este tipo de Honeypots son muy atractivas para los atacantes y permiten un estudio completo de su comportamiento. Deben estar constantemente monitorizadas ya que su peligro consiste en que si un atacante logra acceso a ella, puede disponer de todo el sistema como le plazca. Esto significa que ya no podemos considerarlo como un lugar con logs "fiables" y puede ser utilizado para atacar otros sistemas de nuestra red o incluso de otras conectadas a Internet.

1.2.2 Los Honeynet.

Una vez definido el concepto de Honeypot y realizadas las primeras pruebas con éxito, se propuso la extensión de este concepto que sería honeynet.
Podemos definir una Honeynet como un tipo concreto de Honeypot. Específicamente es un Honeypot altamente interactivo diseñado para la investigación y la obtención de información sobre atacantes. Una Honeynet es una arquitectura, no un producto concreto o un software determinado.
El nuevo enfoque consiste no en falsear datos o engañar a un posible atacante como suelen hacer algunos Honeypot, sino que el objetivo principal es recoger información "real" de cómo actúan los atacantes en un entorno de verdad.

Para conseguir este entorno real con sistemas reales, no con simples emulaciones de servicios y altamente interactivo, se dispone una configuración de red típica con todos sus elementos.
Obviamente, esta red ha sido diseñada para ser comprometida, por lo que debe estar separada de forma segura y controlada de la de producción. Por otro lado, como nuestro objetivo es el de hacer creer al atacante que está ante una red "real", debemos añadir los distintos elementos que conforman una arquitectura "normal" en cualquier red, distintas máquinas, distintos sistemas operativos.

Requerimientos básicos de una Honeynet:

- **Control del flujo de datos:** Siempre que interactuamos con un atacante, el peligro aumenta exponencialmente. Aunque el objetivo de la Honeynet es ser comprometida y atacada,

debemos mantener siempre un control del flujo de datos para evitar que el atacante la utilice contra terceros o contra nuestra propia red [2].

Si bien es cierto que cuánta más interacción se logre con el exterior, más datos reales se pueden obtener del atacante, evaluando los riesgos que conlleva. Análogamente, una Honeynet que no permita ningún tipo de actividad con el exterior no dejará de ser atractiva para un atacante y perderá toda su utilidad. Como siempre, la búsqueda de un equilibrio nos debe guiar en este aspecto.

- **Captura de datos:** La captura de todos los movimientos y acciones que realice el atacante en la Honeynet, revelará sus técnicas y motivaciones. Si bien es esencial que el nivel de vigilancia y captura sea alto, si este es excesivo o detectado por el atacante dejará de ser efectivo. Obviamente, la captura de datos debe hacerse sigilosamente y sin despertar ningún tipo de sospecha, por lo que debe planificarse cuidadosamente.

El lugar dónde se almacena esta información debería encontrarse fuera de la Honeynet, ya que compromete al sistema, sí algún atacante llega a encontrarla y hace cambios o la elimina. Esto eliminaría cualquier utilidad a la Honeynet.

1.2.3 Nepenthes

Nepenthes es una herramienta que simula vulnerabilidades conocidas para descargar el malware que intenta explotar estas debilidades. Es un tipo de honeypot de baja interacción que se puede utilizar para alertar rápidamente a un administrador de un compromiso en la red. Estos dispositivos proveen de binarios maliciosos que son transferidos y almacenados bajo mecanismos de seguridad en el disco duro para su posterior análisis, y nunca se ejecutan.

Un honeypot de bajo-interacción como Nepenthes es fácil de instalar y requiere mantenimiento mínimo. Nepenthes tiene módulos para simular las vulnerabilidades que requieren del conocimiento de las debilidades, de tal forma que se pueda examinar cómo el virus se aprovechará de la debilidad, entonces recolectar toda esta información para descargar el archivo y enviar al asaltante sólo bastante información para que no note que esta siendo engañado. Nepenthes captura la información y la almacenada en ficheros log.

Ficheros Log

Cuando se accede a una página, esta nos registra pero no en forma de estadísticas, sino que lo hace en una hoja parecida a la del bloc de notas poniendo todos nuestros datos, los cuáles contienen nuestra dirección IP, el navegador que utilizamos, sistema operativo y tiempo de permanencia, entre otros datos. Esta clase de archivos no son públicos, solo para los administradores y son llamados Log.

Definición de Log

Registro oficial de eventos durante un período de tiempo en particular. Para los profesionales en seguridad informática un log es usado para registrar datos o información sobre quién, qué, cuándo, dónde y por qué un evento ocurre para un dispositivo en particular o aplicación. [3]
La mayoría de los logs son almacenados o desplegados en el formato estándar ASCII, el cuál es un conjunto de caracteres para dispositivos comunes y aplicaciones. De esta forma logs generados por un dispositivo en particular pueden ser leídos y desplegados en otro diferente.

Propósito de los logs

Si un log tiene la capacidad de registrar los eventos, entonces el propósito de un log es proveer a los profesionales de seguridad informática la habilidad de monitorear las actividades de la aplicación o dispositivo. Revisando las salidas de los archivos de logs, se puede obtener una buena oportunidad para determinar los eventos, y tomar la acción necesaria para corregir el problema o iniciar una investigación en caso de un incidente de seguridad.

1.3 Lenguajes y Tecnologías de desarrollo para la aplicación.

1.3.1 PHP como lenguaje de programación.

PHP es un lenguaje de programación usado frecuentemente para la creación de contenido para sitios Web, con el este se puede programar las páginas html y los códigos de fuente. PHP es un acrónimo recursivo que significa "PHP Hypertext Pre-processor", y se trata de un lenguaje interpretado usado

para la creación de aplicaciones para servidores, o creación de contenido dinámico para sitios Web. Últimamente también para la creación de otro tipo de programas incluyendo aplicaciones con interfaz gráfica usando las librerías GTK.

Ventajas de PHP

- Es un lenguaje multiplataforma.
- Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL
- Lee y manipula datos desde diversas fuentes, incluyendo datos que pueden ingresar los usuarios desde formularios HTML.
- Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados extensiones).
- Posee una amplia documentación en su página oficial, entre la cuál se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite las técnicas de Programación Orientada a Objetos.
- Permite crear los formularios para la Web.
- Biblioteca nativa de funciones sumamente amplia e incluida.
- No requiere definición de tipos de variables ni manejo detallado del bajo nivel.

1.3.2 MySQL como Gestor de Bases de Datos.

Como gestor de base datos se escogió MySQL porque es muy rápido, confiable, robusto y fácil de usar, tanto para volúmenes de datos grandes como pequeños. Además tiene un conjunto muy práctico de características desarrolladas en cooperación muy cercana con los usuarios. Sin embargo bajo constante desarrollo, MySQL hoy en día ofrece un rico y muy útil conjunto de funciones.

Desde sus inicios MySQL se ha convertido en el Gestor de Bases de datos de código abierto más popular de Internet, es principalmente por su simpleza, y a la vez robustez, que ha llamado la atención de los desarrolladores Web. Inicialmente, MySQL carecía de elementos considerados esenciales en

las bases de datos relacionales, tales como integridad referencial y transacciones. A pesar de ello, atrajo a los desarrolladores de páginas Web con contenido dinámico, justamente por su simplicidad; aquellos elementos faltantes fueron llenados por la vía de las aplicaciones que la utilizan [4].

Poco a poco los elementos faltantes en MySQL están siendo incorporados por desarrolladores internos, o por desarrolladores de software libre.

Entre las características disponibles en las últimas versiones se puede destacar.

- Amplio subconjunto del lenguaje SQL. Algunas extensiones son incluidas igualmente.
- Disponibilidad en gran cantidad de plataformas y sistemas.
- Diferentes opciones de almacenamiento según si se desea velocidad en las operaciones o el mayor número de operaciones disponibles.
- Transacciones y claves foráneas.
- Conectividad segura.
- Replicación.

1.4 Herramientas utilizadas para el desarrollo de la aplicación

1.4.1 Dreamweaver.

Adobe Dreamweaver es un editor de páginas web, creado por Adobe. Es el programa de este tipo más utilizado en el sector del diseño y la programación web, por sus funcionalidades, su integración con otras herramientas. Su principal competidor es Microsoft FrontPage. Tiene soporte tanto para edición de imágenes como para animación a través de su integración con otras herramientas. Dreamweaver ha tenido un gran éxito desde finales de los 90 y actualmente mantiene el 90% del mercado de editores HTML. Esta aplicación está disponible tanto para la plataforma MAC como Windows [5].

Dreamweaver permite utilizar la mayoría de los navegadores Web instalados en su ordenador para previsualizar las páginas web. También dispone de herramientas de administración de sitios dirigidas a principiantes como, por ejemplo, la habilidad de encontrar y reemplazar líneas de texto y código por cualquier tipo de parámetro especificado, hasta el sitio web completo. El panel de comportamientos también permite crear JavaScript básico sin conocimientos de código [5].

Un aspecto de alta consideración es su arquitectura extensible. Es decir, permite el uso de "Extensiones". Las extensiones, tal y como se conocen, son pequeños programas, que cualquier desarrollador web puede escribir (normalmente en HTML y Javascript) y que cualquiera puede descargar e instalar, ofreciendo así funcionalidades añadidas a la aplicación.

1.4.2 Rational Rose 2003.

Rose es una herramienta con plataforma independiente que ayuda a la comunicación entre los miembros de equipo, a monitorear el tiempo de desarrollo y a entender el entorno de los sistemas. Una de las grandes ventajas de Rose es que utiliza la notación estándar en la arquitectura de software(UML), la cuál permite a los arquitectos de software y desarrolladores visualizar el sistema completo utilizando un lenguaje común, además los diseñadores pueden modelar sus componentes e interfaces en forma individual y luego unirlos con otros componentes del proyecto.

Se decidió utilizar el Rational Rose Enterprise Edición 2003, para respaldar la documentación, como modelador visual de la notación UML (Unified Modeling Language) para la confección de los diagramas que se ilustran en este documento. Esta herramienta es muy completa y ofrece amplias potencialidades.

Funciones:

- Posibilidad de publicar en las Web modelos e informes para mejorar la comunicación entre los miembros.
- Modelado en UML para diseñar bases de datos, que integran los requisitos de datos y aplicaciones mediante diseños lógicos y analíticos.
- Funciones de análisis de calidad de código.

Complemento de modelado Web que incluye funciones de visualización, modelado y herramientas para desarrollar aplicaciones Web.

1.4.3 SQL Manager 2005 para MySQL.

El EMS para MySQL es una herramienta para administrar una base de datos en MYSQL, te proporciona un completo conjunto de eficaces y potentes herramientas para administrar un servidor MySQL.

A través de su clara interfaz gráfica podrás crear y editar parámetros de tu base de datos de forma sencilla. Ofrece la posibilidad de otorgar y administrar privilegios de usuarios, ejecutar scripts SQL, consultas visuales integradas, extraer o imprimir meta data, importar y exportar datos, etc [6].

1.5 Metodologías y lenguajes de modelado.

1.5.1 RUP. El Proceso Unificado de Desarrollo.

En un proceso de desarrollo de software existen un conjunto de actividades que guían los esfuerzos de las personas implicadas en el proyecto, a modo de plantilla explica los pasos necesarios para terminar el proyecto, dando como resultado final un producto terminado.

RUP es el resultado de varios años de desarrollo y uso práctico, en el que se han unificado técnicas de desarrollo a través de UML y trabajo de muchas metodologías utilizadas por los clientes. La versión que se ha estandarizado vío la luz en 1998 y se conoció en sus inicios como Proceso Unificado de Rational 5.0; de ahí las siglas con las que se identifica a este proceso de desarrollo.

En RUP se han agrupado las actividades en grupos lógicos definiéndose 9 flujos de trabajo principales de las cuales los 6 primeros son conocidos como flujos de ingeniería y los tres últimos como de apoyo.

Flujos de trabajo:

- Modelamiento del negocio: Describe los procesos de negocio, identificando quiénes participan y las actividades que requieren automatización.

- Requerimientos: Define qué es lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen.

- Análisis y diseño: Describe cómo el sistema será realizado a partir de la funcionalidad prevista y las restricciones impuestas (requerimientos), por lo que indica con precisión lo que se debe programar.

- Implementación: Define cómo se organizan las clases y objetos en componentes, cuáles nodos se utilizarán y la ubicación en ellos de los componentes y la estructura de capas de la aplicación.

- Prueba (Testeo): Busca los defectos a los largo del ciclo de vida.

- Instalación: Produce release del producto y realiza actividades (empaque, instalación, asistencia a usuarios, etc. para entregar el software a los usuarios finales.

- Administración del proyecto: Involucra actividades con las que se busca producir un producto que satisfaga las necesidades de los clientes.

- Administración de configuración y cambios: Describe cómo controlar los elementos producidos por todos los integrantes del equipo de proyecto en cuanto a: utilización, actualización concurrente de elementos, control de versiones, etc.

- Ambiente: Contiene actividades que describen los procesos y herramientas que soportarán el equipo de trabajo del proyecto; así como el procedimiento para implementar el proceso en una organización.

RUP tiene además tiene 4 fases por la que obligatoriamente tiene que transitar todo proyecto: Conceptualización o Inicio es la fase donde se describe el negocio y se delimita el proyecto

describiendo sus alcances con la identificación de los casos de uso del sistema, elaboración es en la que se define la arquitectura del sistema y se obtiene una aplicación ejecutable que responde a los casos de uso que la comprometen, construcción donde se obtiene un producto listo para su utilización que está documentado y tiene un manual de usuario. La última fase es la de transición en la que el release ya está listo para su instalación en las condiciones reales.

1.5.2 Lenguaje Unificado de Modelo. UML.

UML es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. Además ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables.

Es importante remarcar que UML es un "lenguaje" para especificar y no un método o un proceso, se utiliza para definir un sistema de software, para detallar los artefactos en el sistema, para documentar y construir el lenguaje en el que está descrito el modelo. Se puede aplicar en una gran variedad de formas para soportar una metodología de desarrollo de software (tal como el Proceso Unificado de Rational) pero no especifica en sí mismo qué metodología o proceso usar. UML cuenta con varios tipos de diagramas, los cuales muestran diferentes aspectos de las entidades representadas.

1.6 Patrón Modelo-Vista-Controlador.

En este trabajo utilizamos el patrón Modelo-Vista-Controlador porque nos ofrece una guía para el diseño de la arquitectura de la aplicación de tal forma que brinde una fuerte interactividad con los usuarios. Este patrón organiza la aplicación en tres modelos separados, un modelo para representar los datos de la aplicación y sus reglas de negocio, otro para el conjunto de vistas que representa los formularios de entrada y salida de información, y por ultimo uno, para el conjunto de controladoras que procesan las peticiones de los usuarios y controlan el flujo de ejecución del sistema.

Escogimos este porque nos brinda beneficios como son:

- Menor acoplamiento
- Mayor cohesión ya que cada elemento del patrón esta altamente especializado en su tarea, la vista en mostrar datos al usuario, el controlador en las entradas y el modelo en su objetivo del negocio.
- Las vistas proveen mayor flexibilidad y agilidad pues se pueden crear múltiples vistas de un modelo, las vistas pueden anidarse, se pueden sincronizar las vistas y otras más opciones.
- Más claridad de diseño
- Facilita el mantenimiento.
- Mayor escalabilidad

1.7 Conclusiones

En este capítulo se abordaron los temas más actuales relacionados con el estado del arte del tema tratado en el trabajo, los sistemas que se utilizan para la captura de información de ataques, así como la clasificación de los mismos, para una mejor comprensión de la información que va a ser manejada por el sistema que debemos diseñar. Se definen las tecnologías y herramientas usadas para de realizar el proyecto con calidad.

Capítulo 2. Características del sistema.

2.1 Introducción

El capítulo presentará un análisis de la situación problémica a la que se va enfrentar el presente trabajo .Se expondrá el modelo del negocio a través de diagramas y descripciones que harán fácil su comprensión. Contendrá los requerimientos funcionales y no funcionales especificados por el cliente. Se realiza el modelo del sistema, especificando sus características, basándose en los actores, procesos de casos de uso vinculados al sistema.

2.2 Objeto de estudio.

2.2.1 Situación problémica.

En Segurmática se utilizan herramientas para capturar información de ataques y programas malignos .En la actualidad estos datos capturados, referentes a los ataques y programas malignos se obtienen en ficheros bitácoras los cuáles tienen mucha información irrelevante que no es necesaria, y conlleva a que el proceso de análisis de esta información capturada sea complejo y engorroso. Esto a la vez retrasa a los analistas de seguridad informática a la hora de dar una respuesta eficiente ante las incidencias.

El problema existente está dado por la carencia, en Segurmática de un sistema que genere reportes con la información relativa a los ataques y programas malignos, para agilizar el proceso de análisis y permitir a éstos una solución en menos tiempo.

2.2.2 Objeto de automatización.

Nuestro objeto de automatización es desarrollar un sistema generador de reportes con el objetivo de mantener un flujo constante de datos con información de ataques y programas malignos, facilitando al usuario final poder acceder a una información mucho más clara, pudiendo buscar por fechas las estadísticas de los ataques.

Además es de interés automatizar la generación de gráficos en cuanto a:

- Cantidad total de programas malignos
- Cantidad total de sesiones de ataques
- Cantidad total de IP fuentes de los ataques.

En la empresa existe un sistema automatizado, llamado Nepenthes que está vinculado al campo de acción de nuestro proyecto, a través de esta herramienta que simula tener vulnerabilidades, es que se obtienen los datos relativos a las capturas en ficheros log.

2.2.3 Información que se maneja.

La información que se maneja es la relacionada con los datos de las capturas y datos de los ataques, específicamente de ellos el tiempo en que ocurrieron, desde que dirección IP, país y de las capturas su dirección de descarga, el tiempo de la descarga, la cantidad de veces que ha sido descargada, tamaño, fecha .Para ello se dispondrá de una base de datos donde estará registrada toda esta información necesaria.

2.3 Propuesta de sistema.

Con el objetivo de mejorar la calidad del trabajo relacionado con la generación de reportes en la empresa Segurmática, tenemos como propuesta un sistema que garantice en todo momento a los usuarios autorizados, una información detallada sobre los datos de las capturas que propagaron programas malignos y las que no. También debe garantizar información de las sesiones de ataques y desde que IP se realizó el ataque.

El sistema permitirá buscar toda la información relacionada con los aspectos antes mencionados de forma rápida, a partir de un conjunto de reportes que serán de gran importancia para los analistas. Además generará gráficas con la cantidad total de programas malignos capturados, sesiones de ataques e IP fuentes de ataques obtenidos en tiempo real por los sistemas que están funcionando.

2.4 Modelo del negocio

En el modelo del negocio se describirán los procesos, existentes u observados, con el propósito de comprenderlos. Se especifican aquí qué procesos del negocio soportará el sistema, se identificarán los objetos del negocio, implicados. También este modelo establece las competencias que se requieren de cada proceso: sus trabajadores, sus responsabilidades y las operaciones que llevan a cabo.
En Segurmática se utilizan herramientas para capturar información de toda la actividad realizada por los atacantes ,para ello en la empresa , se utilizan sistemas que obtienen de forma automática la información necesaria en ficheros log, para lograr entender la naturaleza de los ataques y conocer nuevos programas malignos que estos pueden propagar, para ello se lleva a cabo en la organización el proceso de análisis de estos datos recibidos que son analizados desde ficheros logs los cuales guardan gran cantidad de eventos que hace mas engorroso el proceso de análisis de la información.
En el negocio actual proponemos que estos datos utilizados por el analista se muestren a través de reportes para lograr una mejor comprensión de los datos que le permita al analista agilizar el proceso de análisis.

2.4.1 Definición de actores y trabajadores del negocio.

El término actor del negocio significa el rol que algo o alguien juega cuando interactúa con el negocio.

Actores del negocio	**Justificación**
Jefe	Recibe los reportes de las capturas y basándose en ellos, toma decisiones.

Tabla 2.1: Descripción de los actores del negocio.

Un trabajador del negocio es una abstracción de una persona (o grupo de personas), una máquina o un sistema automatizado,que actúa en el negocio realizando una o varias

actividades, interactuando con otros trabajadores del negocio y manipulando endidades del negocio.

Trabajadores del negocio	Justificación
Analista	Crear los reportes y analizar la información capturada.

Tabla 2.2: Descripción de los trabajadores del negocio.

2.4.2 Diagrama de casos de uso del negocio.

En este diagrama se representa gráficamente los procesos que existen en el negocio y su relación con los actores.

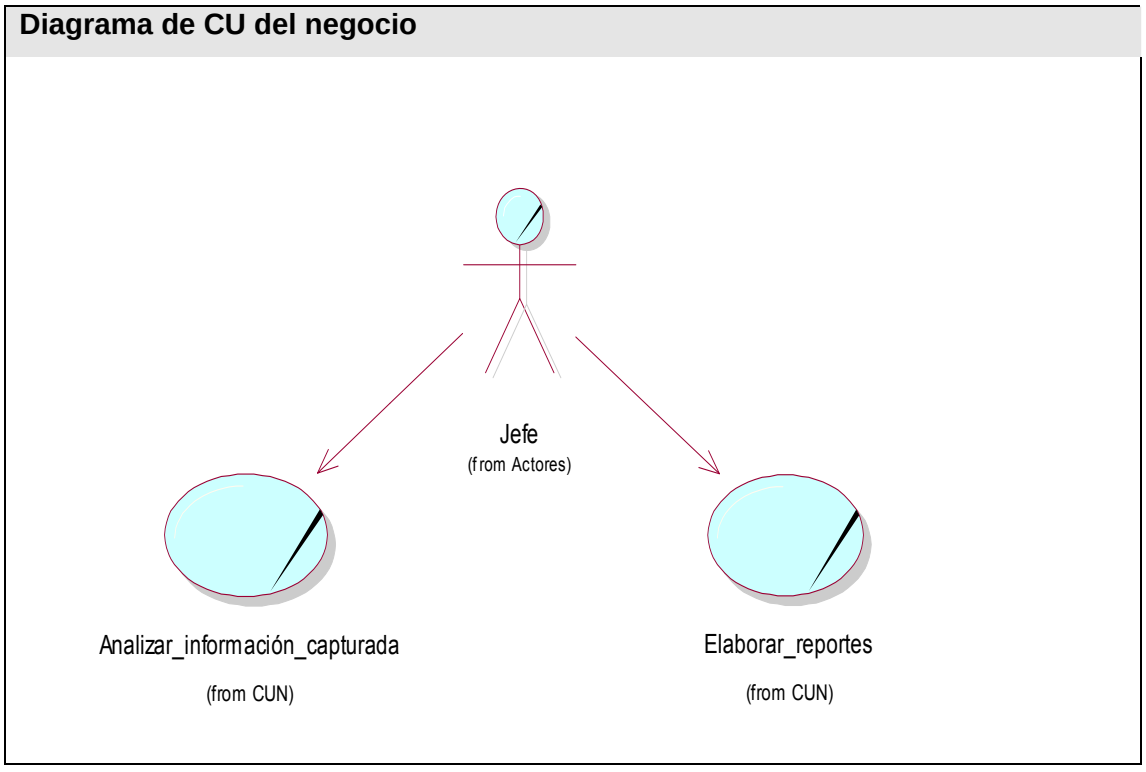


Figura 2.1: Diagrama de Casos de Usos del Negocio

2.4.3 Descripción textual de los Casos de Uso del Negocio.

La descripción de un caso de uso del negocio muestra cómo colaboran los trabajadores y entidades del negocio para ejecutar el proceso. Son reseñas textuales del caso de uso y explica los procesos o actividades que tienen lugar en el mismo.
En el Anexo 1 se encuentran las descripciones textuales de los CUN representados en el diagrama anterior. (Ver anexo 1).

2.4.4 Diagramas de Actividades.

Un diagrama de actividad es un grafo, que describe un proceso que explora el orden de las tareas o actividades que logran los objetivos del negocio.
Hemos desarrollados los diagramas de actividades correspondientes a cada uno de los Casos de Usos del Negocio (ver anexo 2).

2.4.5 Modelo de objetos.

En este modelo mostramos la participación de los trabajadores y entidades implicados en nuestro negocio y la relación que existe entre ellos.
En nuestro caso se ha construido un modelo de objetos general que involucra a todos los trabajadores del negocio y sus relaciones con las entidades correspondientes (ver anexo 3)

2.5 Relación de Requerimientos.

Los requerimientos son las condiciones o capacidades que tiene que tener un sistema para satisfacer un contrato o documento. Por tanto es aquí donde definiremos que es lo que el sistema debe hacer.

2.5.1 Listado de los requerimientos funcionales.

R1 Obtener reportes

- R 1.1 Obtener reportes por captura de programas malignos
- R 1.2 Obtener reportes por sesiones de ataques
- R 1.3 Obtener reportes por IP fuentes de los ataques
- R 1.4 Obtener reportes por contenido de ataques no propagaron programas malignos.

R2. Autenticar usuario.

R 2.1 Comparar usuario y contraseña con los ya existentes en el sistema.

R3 Generar gráficas

R 3.1 Generar gráficas con la cantidad total de programas malignos capturados

R 3.2 Generar gráficas con la cantidad total de sesiones de ataques

R 3.3 Generar gráficas con la cantidad total de IP fuentes.

R4 Administrar usuario

R 4.1 Buscar y eliminar usuarios

R 4.2 Insertar usuarios

R 4.3 Cambiar contraseña de usuarios

2.5.2 Definición de los requerimientos no funcionales

Los requerimientos no funcionales son fundamentales, ya que estos son propiedades que debe tener el producto, que lo hacen atractivo, usable, rápido o confiable .Se determinaron los siguientes requisitos no funcionales.

Requerimientos de apariencia o interfaz externa

- En la interfaz se presentan funcionalidades específicas del sistema. Para el caso de los reportes se limitará la cantidad de resultados que se mostrarán en una misma página.
- Interfaz legible, fácil de usar, profesional con gráficos que muestran el comportamiento de algunos parámetros importantes para el usuario.

Requerimiento de Seguridad

- Garantiza que la información sea actualizada únicamente por quién tiene acceso a trabajar con el sistema.
- Protección contra acciones no autorizadas o que puedan afectar la integridad de los datos.

- La información del sistema esta protegido de acceso no autorizado.
- Los usuarios autorizados a trabajar con la aplicación se les ha garantizado el acceso a la información solicitada en todo momento.

Requerimientos de rendimiento.

- El tiempo de respuesta en una transacción o traspaso de información será en el menor tiempo posible para que la aplicación sea eficiente, rápida y precisa, esta implementada sobre una tecnología Web.

Requerimientos de soporte.

- El sistema debe ser de fácil instalación
- El sistema debe estar bien documentado de forma tal que el tiempo de mantenimiento sea mínimo en caso de necesitarse.

Requerimientos de Portabilidad.

- Al sistema se debe acceder desde cualquier plataforma, la aplicación debe correr sobre una plataforma Web, codificada en PHP y el sistema de Base datos MYSQL.

Requerimientos de software.

- Se utilizará cualquier sistema operativo de la familia de Windows superior a Windows 98 o cualquier versión de Linux.

Requerimientos de Hardware.

Las computadoras que se van a utilizar requieren de las siguientes características:

- PC con procesador Pentium II o superior.
- 256 megabytes (MB) de memória RAM o más.

Requerimientos de diseño.

- Se utilizara como patrón de arquitectura el modelo vista controlador (MVC).
- Como gestor de base de datos será usado MYSQL
- El lenguaje de programación que se usará es el PHP.

Requerimientos de Usabilidad:

- El sistema podrá ser usado por cualquier usuario que posea conocimientos básicos en el manejo de la computadora y de un ambiente Web en sentido general.
- La instalación del sistema conseguirá una mayor rapidez del trabajo de los analistas.
- Documentación adecuada de la aplicación, que brindará un mejor entendimiento de ella.

2.6 Modelo de Caso de Uso del Sistema.

2.6.1 Definición de los actores del sistema a automatizar.

Los trabajadores del sistema son nuestros actores en el negocio, pero en caso de que algún actor del negocio interactué con el sistema, entonces también seria un actor de este. De ellos podemos destacar las siguientes características:

- No son parte de él.
- Pueden intercambiar información con él.
- Pueden ser un recipiente pasivo de información.

Actores	**Justificación**
Analista	Se encarga de elaborar los reportes al interactuar con la aplicación

Tabla 2.3: Listado de Actores del Sistema

2.6.2 Diagrama de Caso de Uso del Sistema a automatizar.

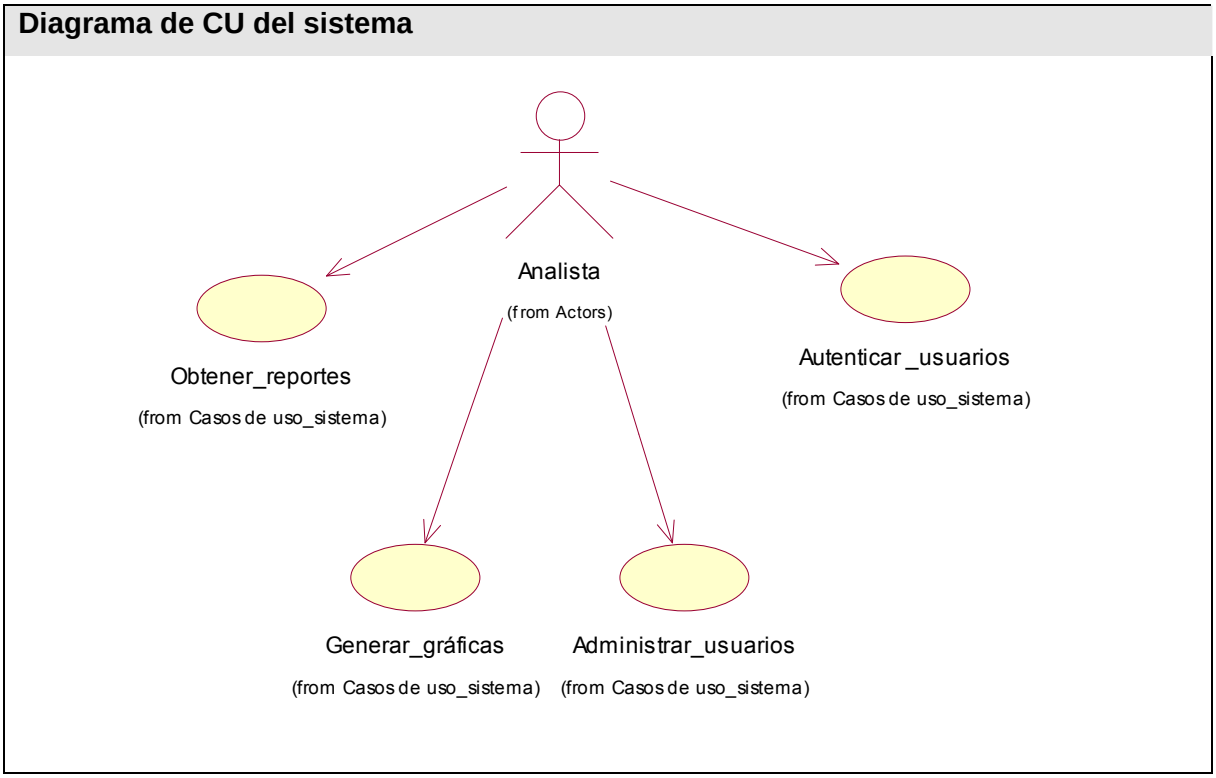


Figura 2.2: Casos de Usos del Sistema

2.6.3 Descripción de los casos de uso.

Los casos de uso son fragmentos de funcionalidad que el sistema ofrece para aportar un resultado de valor para sus actores.

En el Anexo 4 se encuentran las descripciones textuales de los Casos de Usos representados en el diagrama anterior. (Ver anexo 4).

2.7 Conclusiones

El estudio del capítulo proporcionó una mejor visión del problema al que nos enfrentamos, sacando de aquí una propuesta de un sistema generador de reportes, que se trazará como objetivo mejorar la calidad de los procesos que se realizan actualmente en la empresa. Para esto se definieron las condiciones o especificaciones que tendrá el sistema que se desea desarrollar. Además se representó como desglosar la aplicación a través de los artefactos que se realizan en el modelo del negocio y sistema.

Capítulo 3. Análisis y diseño del sistema.

3.1 Introducción

En el presente capítulo abordaremos el análisis y diseño, son etapas importantes en el desarrollo de nuestra aplicación ya que se usan para transformar los requisitos en el diseño del futuro sistema. De ellas se representarán los elementos más significativos como diagramas de clases así como diagramas de interacción. Además se contendrá los diagramas del modelo de datos lógicos y el modelo de datos físicos para una mejor comprensión del diseño de la base de datos.

3.2 Análisis

El análisis consiste en transformar los requisitos funcionales en un diseño de clases en el cuál se ven las relaciones e interacción que existe entre ellas, teniendo en cuenta en este proceso una arquitectura robusta que permita adaptar el sistema al entorno de implementación que se esta desarrollando. Además en este flujo se obtiene una visión del sistema que se preocupa de ver que hace, de tal forma que se preocupa sólo por los requisitos funcionales.

3.2.1 Diagramas de clases del análisis.

El diagrama de clases del análisis es un artefacto en el que se representan los conceptos en un dominio del problema. Seguidamente se muestran los diagramas del análisis por cada caso de uso del sistema.

.

Diagrama de clases de análisis: CU_ Obtener reportes

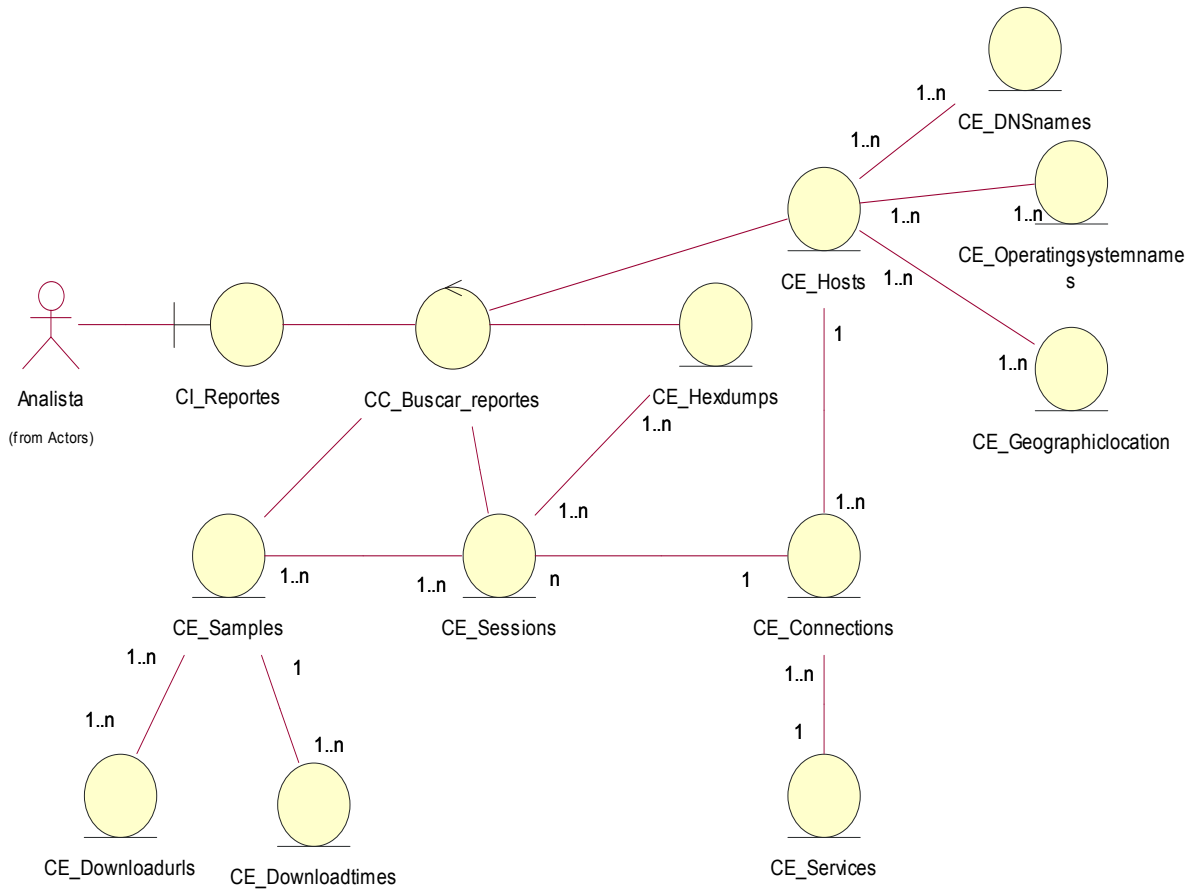


Figura 3.1: Diagrama de clases de análisis.: CU_ Obtener Reportes

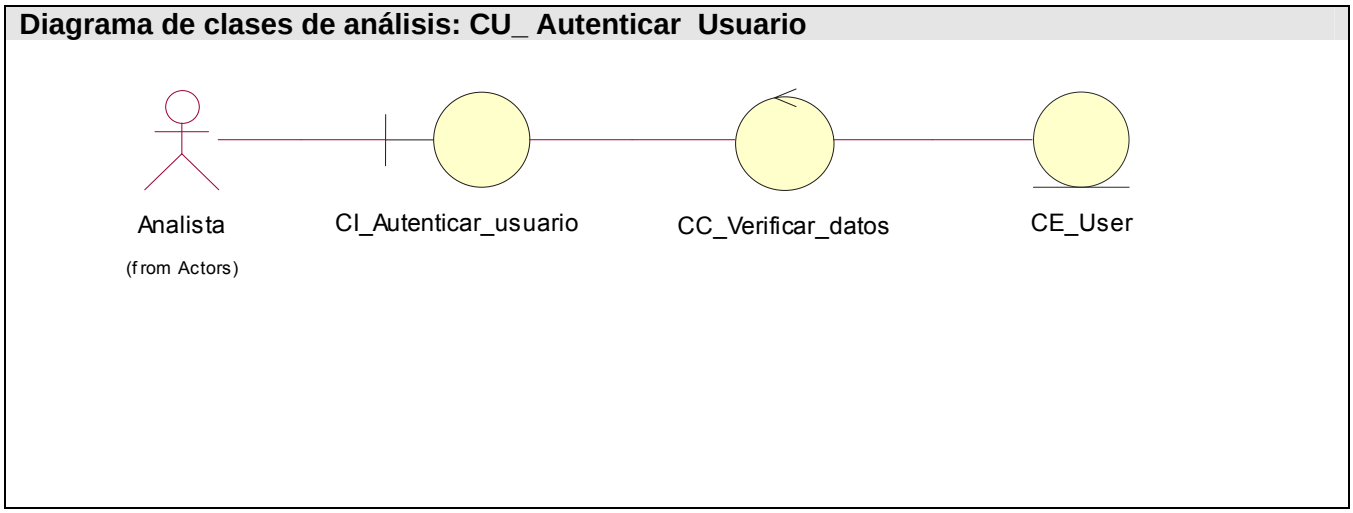


Figura 3.2: Diagrama de clases de análisis.: CU_ Autenticar Usuario

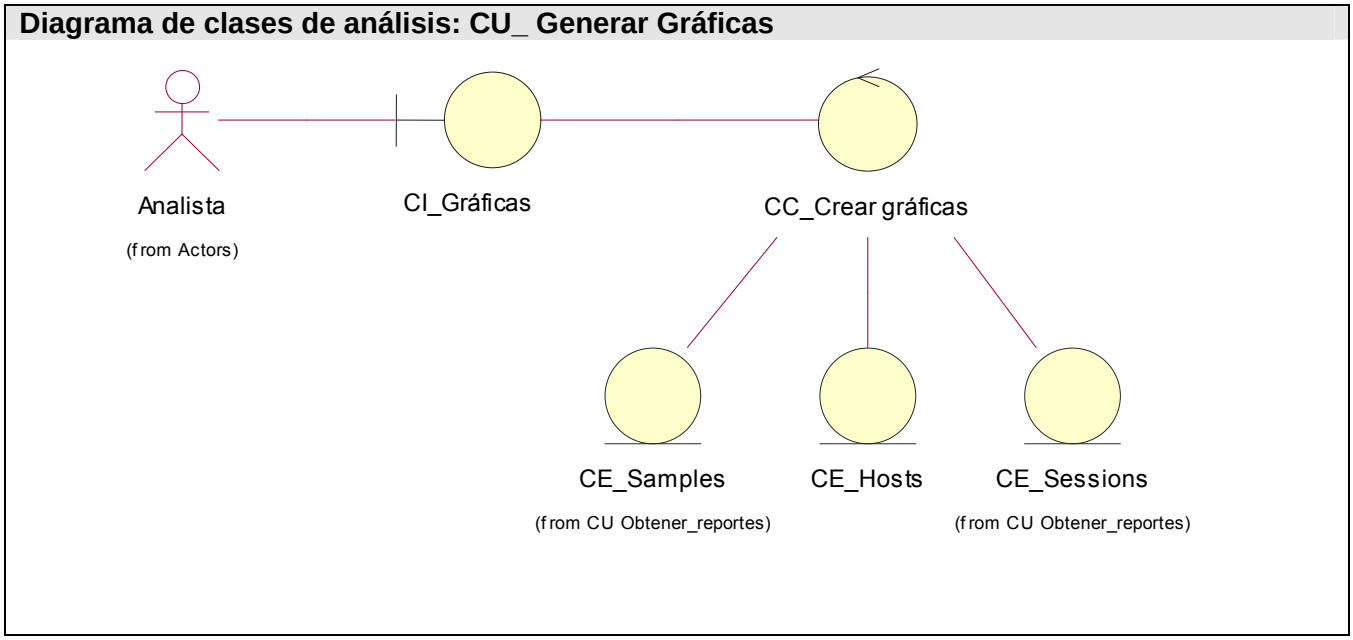


Figura 3.3: Diagrama de clases de análisis.: CU_ Generar Gráficas

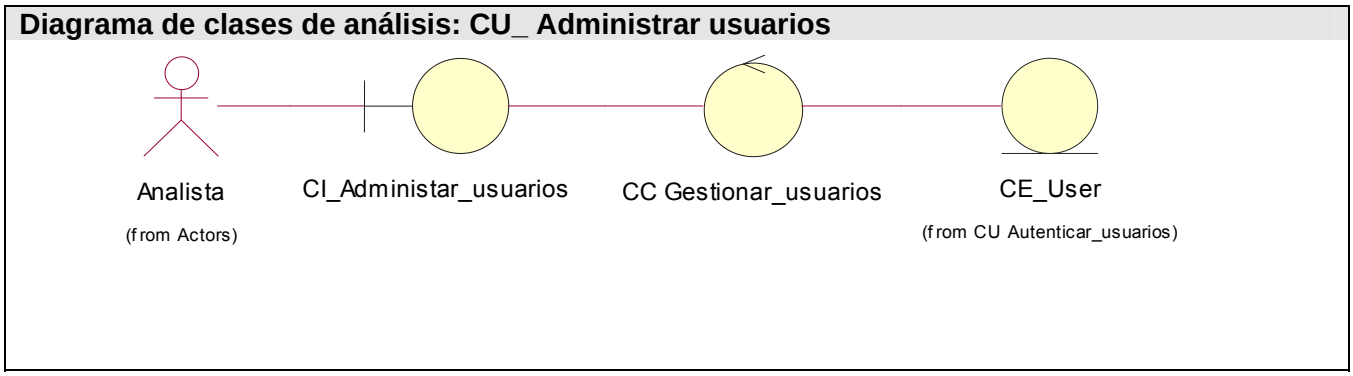


Figura 3.4: Diagrama de clases de análisis.: CU_ Administrar usuarios

3.3 Diseño

El modelo del Diseño pretende esencialmente:

- Adquirir una comprensión de los aspectos relacionados con los requisitos no funcionales y restricciones relacionadas con los lenguajes de programación, componentes reutilizables, sistemas operativos, tecnologías de distribución, concurrencia y tecnologías de interfaz de usuario.

- Crear una entrada apropiada y un punto de partida para actividades de implementación, capturando los requisitos o subsistemas individuales, interfaces y clases.

- Descomponer los trabajos de implementación en partes más manejables que puedan ser llevadas a cabo por diferentes equipos de desarrollo.

- Capturar las interfaces entre los subsistemas antes en el ciclo de vida del software, lo cual es muy útil cuando utilizamos interfaces como elementos de sincronización entre diferentes equipos de desarrolladores, sistemas operativos y otras especificaciones que representan las características del producto.

3.3.1 Diagramas de interacción.

Los diagramas de interacción no son más que una descripción del modo en el que cada operación detectada en los diagramas de secuencia lleva a cabo sus responsabilidades y modifica el estado del sistema. En UML los diagramas de interacción pueden representarse a través de los diagramas de Colaboración y/o de los diagramas de Secuencia.

El tipo de diagrama seleccionado para construir los diagramas de interacción fue el de Secuencia, debido a que muestra cómo los objetos se comunican unos con otros en una secuencia de tiempo, qué sucede en cada momento, y para ello contienen objetos con sus ciclos de vida y los mensajes que se envían entre ellos ordenados secuencialmente.

El diagrama de secuencias es el núcleo de un modelo dinámico, y muestra todos los cursos alternos que pueden tomar los casos de uso. Los diagramas de secuencias se componen de 4 elementos que son: el curso de acción, los objetos, los mensajes y los métodos.

En el Anexo 5 se encuentran los diagramas de secuencia correspondientes a cada caso de uso. (Ver anexo 5).

3.3.2 Diagramas de clases del Diseño.

En el diagrama de clases del diseño se especifica la estructura de las clases del sistema, incluyendo las relaciones entre clases. Durante el análisis del sistema, el diagrama se desarrolla buscando una solución ideal. Durante el diseño, se usa el mismo diagrama, y se modifica para satisfacer los detalles de la implementación.

Los diagramas de clases del diseño elaborados se representan a continuación:

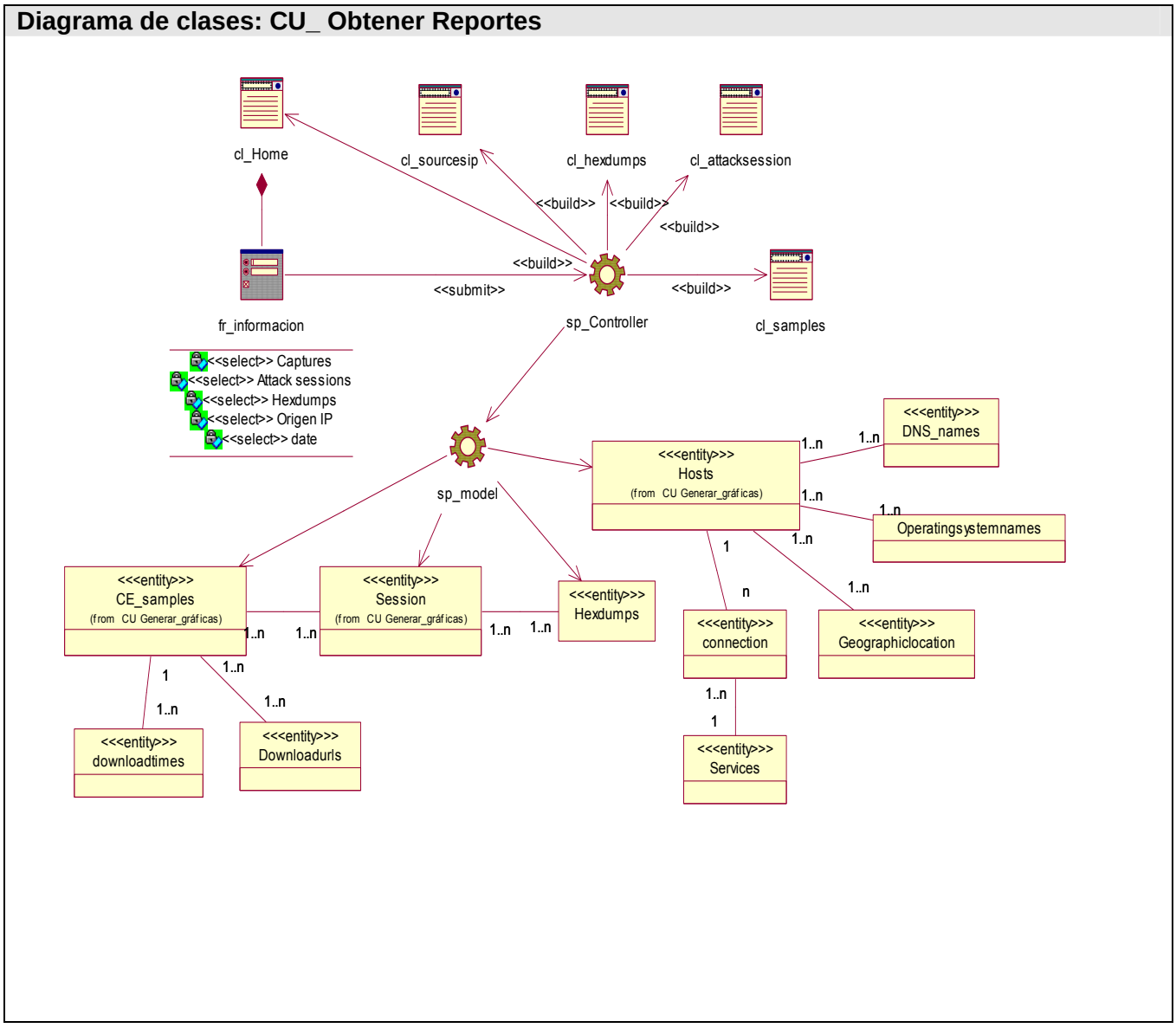


Figura 3.5: Diagrama de clases: CU_ Obtener Reportes

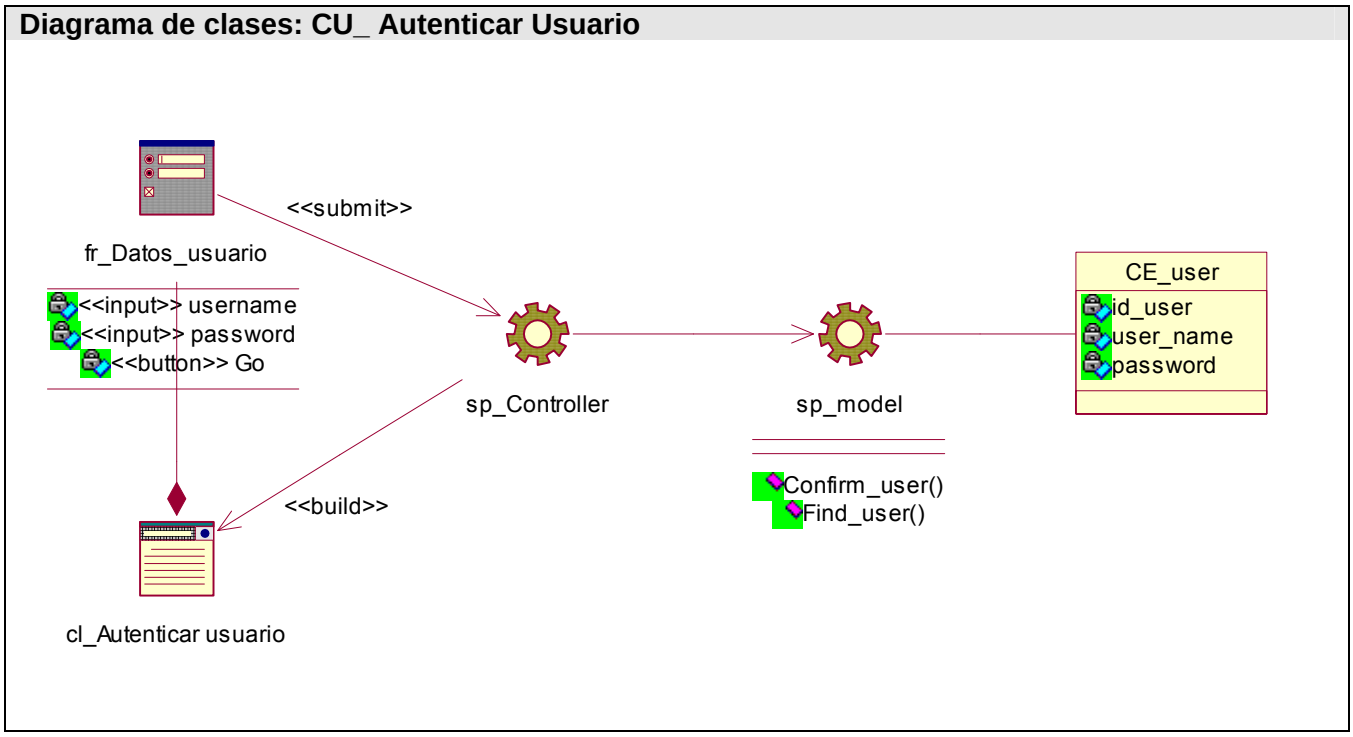


Figura 3.6: Diagrama de clases: CU_ Autenticar Usuario

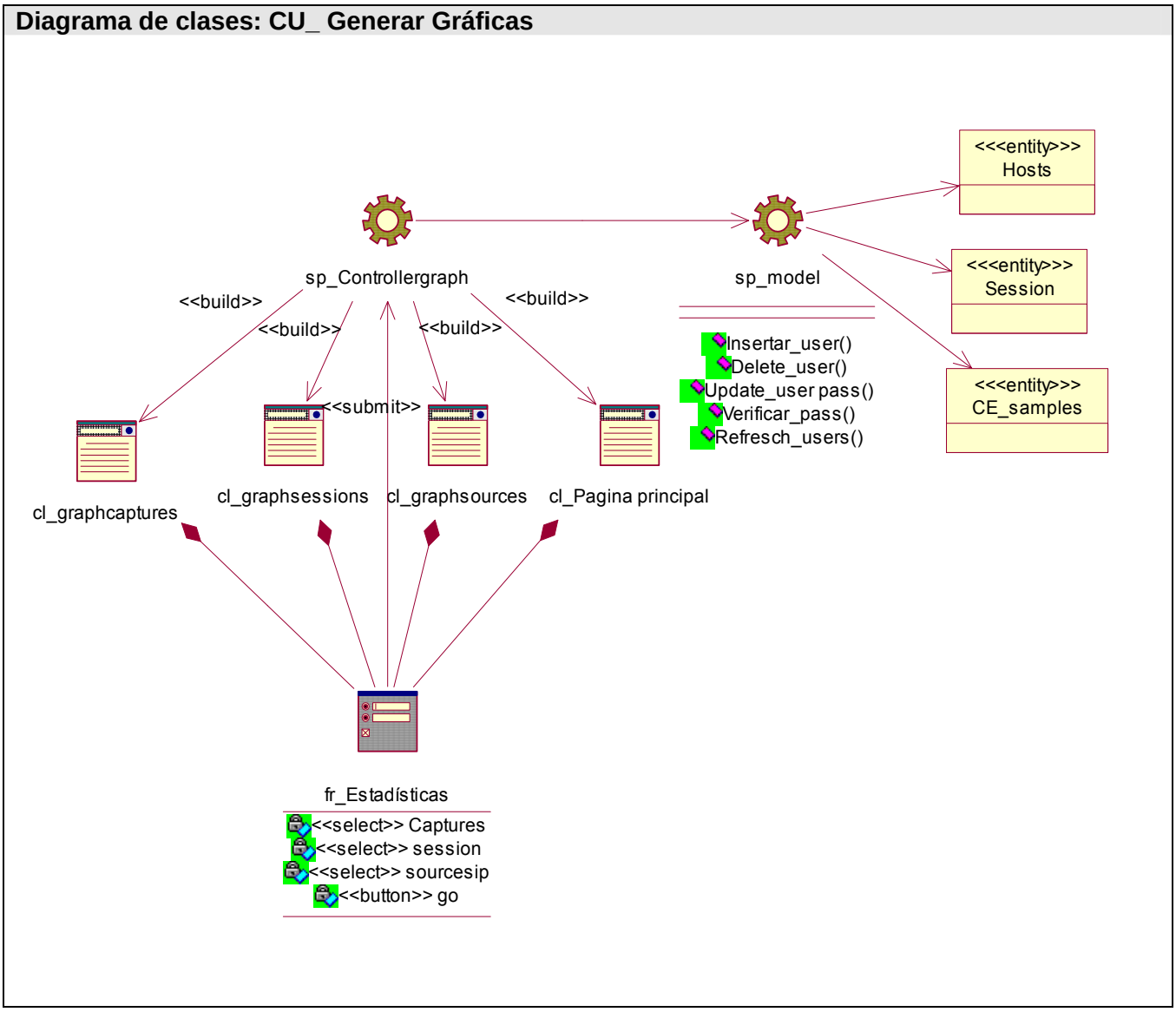


Figura 3.7: Diagrama de clases: CU_ Generar Gráficas

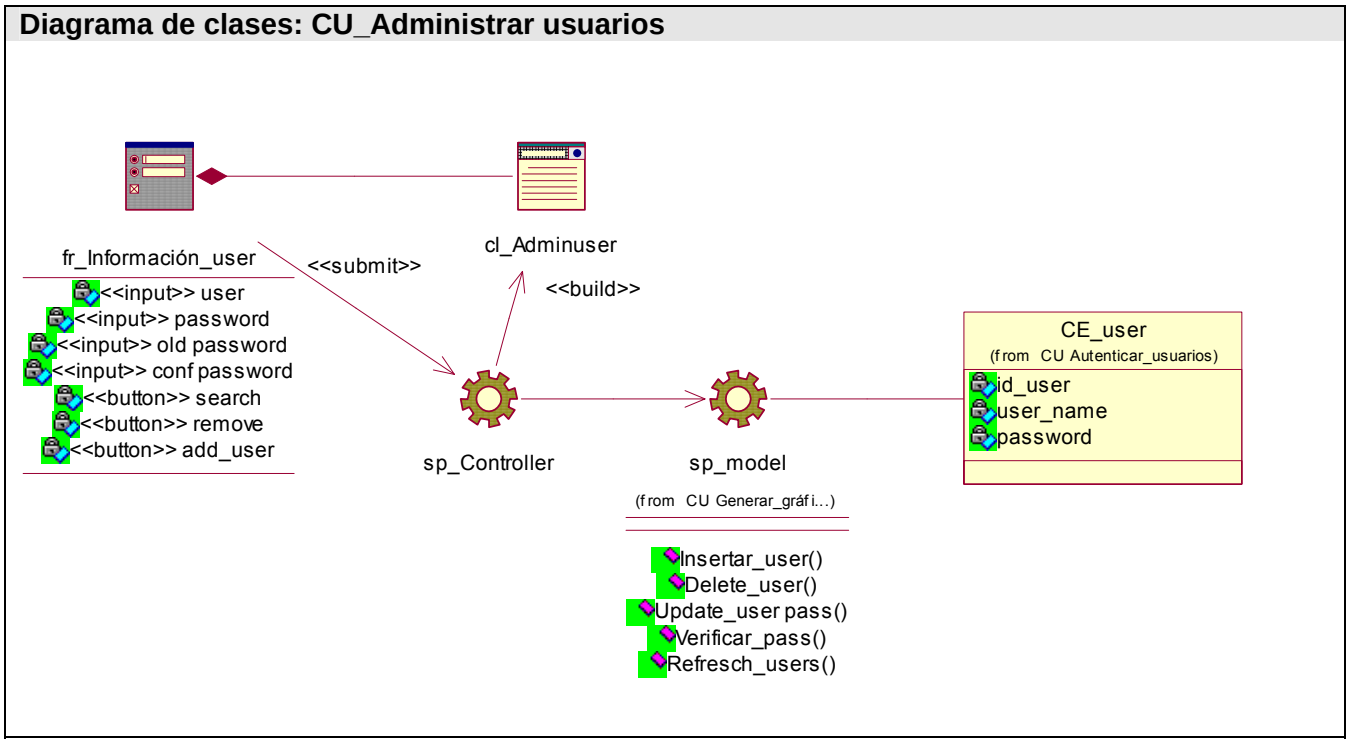


Figura 3.8: Diagrama de clases: CU_ Generar Gráficas

3.4 Diseño de la BD

3.4.1 Modelo lógico de datos.

Todas las clases identificadas en el dominio del análisis no son persistentes. La persistencia es la capacidad de un objeto de mantener su valor en el espacio y en el tiempo. Se han seleccionado las siguientes clases persistentes.

Diagrama de clases persistentes

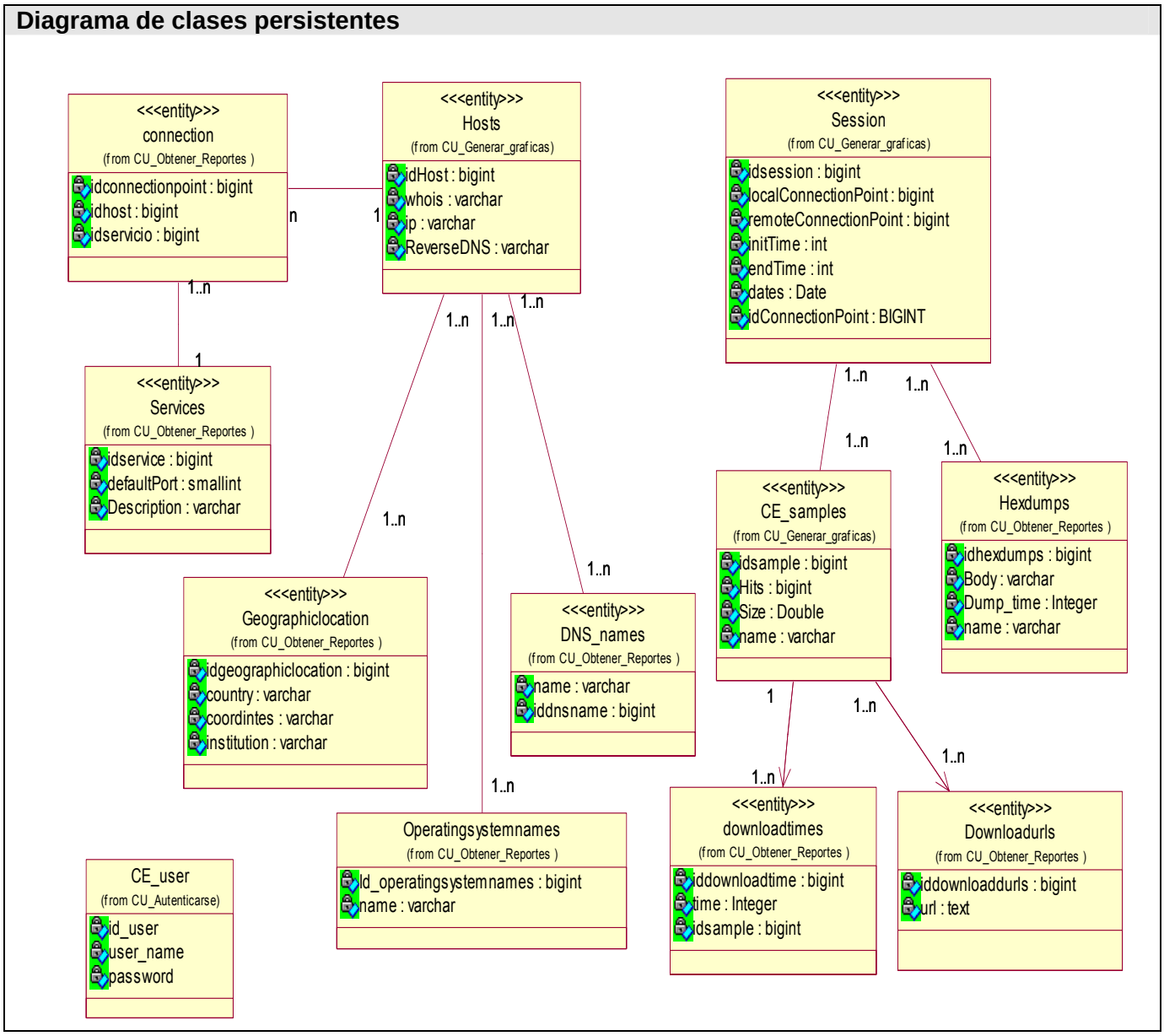


Figura 3.9: Diagrama de clases persistentes.

3.4.2 Modelo físico de datos.

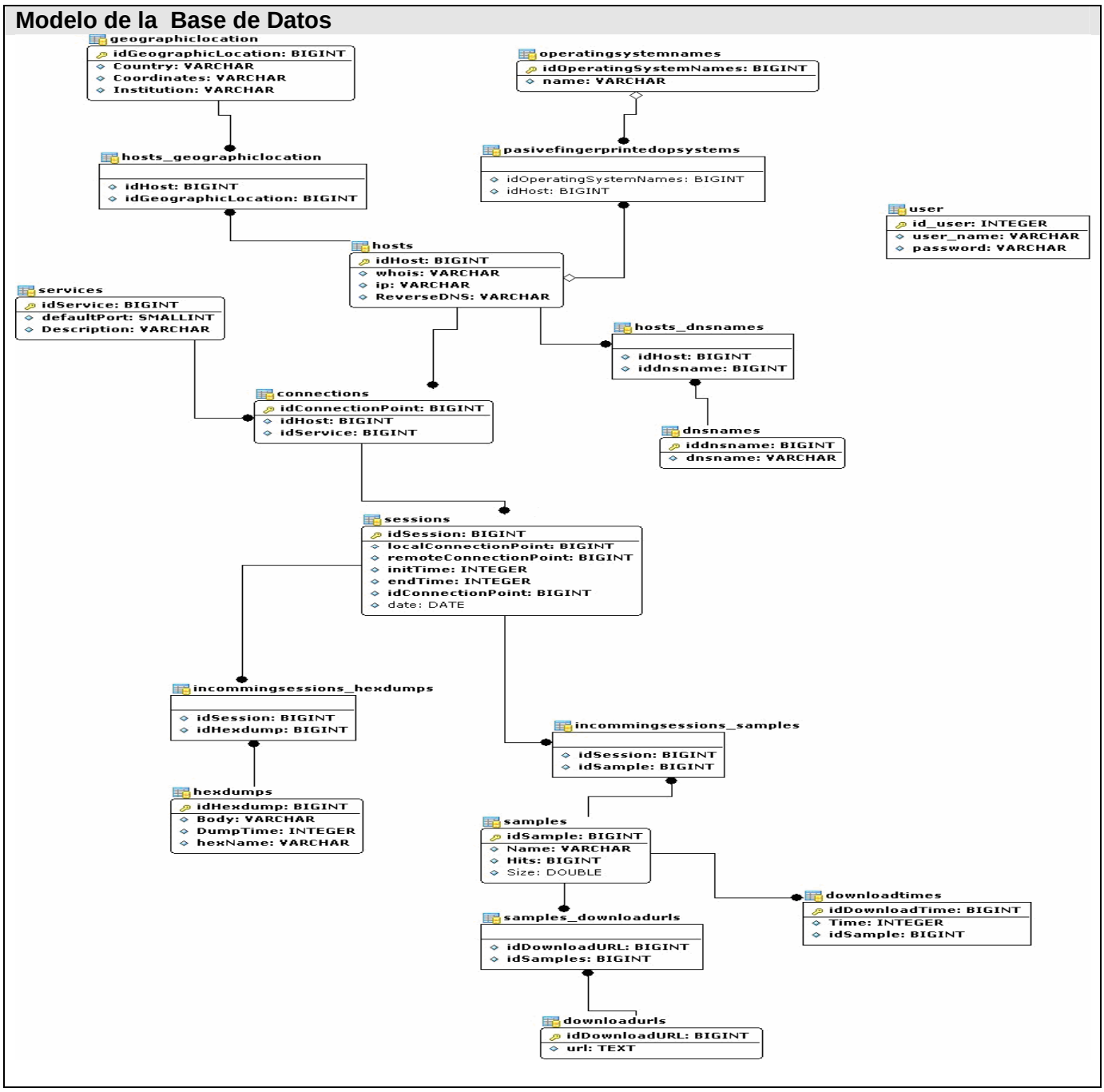


Figura 3.10: Modelo Físico de la Base de Datos.

3.5 Conclusiones.

En este capítulo se desarrollaron todos los aspectos referentes al análisis y diseño de una aplicación, modelando a través de diagramas los requerimientos funcionales y no funcionales de forma tal que estos diagramas describan como implementar el sistema. Con la utilización de la herramienta Rational, se realizaron diagramas de análisis y diseño web, diagramas de interacción, así como los diagramas para representar el diseño de la base datos cumpliendo así el objetivo trazado en el capitulo

Capítulo 4. Implementación.

4.1 Introducción

En este capítulo se explicará como es el comportamiento del flujo de implementación aplicado al sistema de reportes de programas malignos que se desea realizar. Se describirán los resultados anteriormente obtenidos en el diseño, en diagramas de componentes que modelan la vista estática del sistema, en un grafo con los componentes de software unidos a través de relaciones de dependencias. Así como también el diagrama de despliegue que indica la situación física de la aplicación, donde cada hardware es un nodo y su relación entre ellos.

4.2 Diagrama de Despliegue.

Los diagramas de despliegue muestran las relaciones físicas entre los componentes de hardware y software en el sistema final, es decir la configuración de los elementos de procesamiento en tiempo de ejecución y los componentes software. A continuación se muestra el diagrama de despliegue correspondiente:

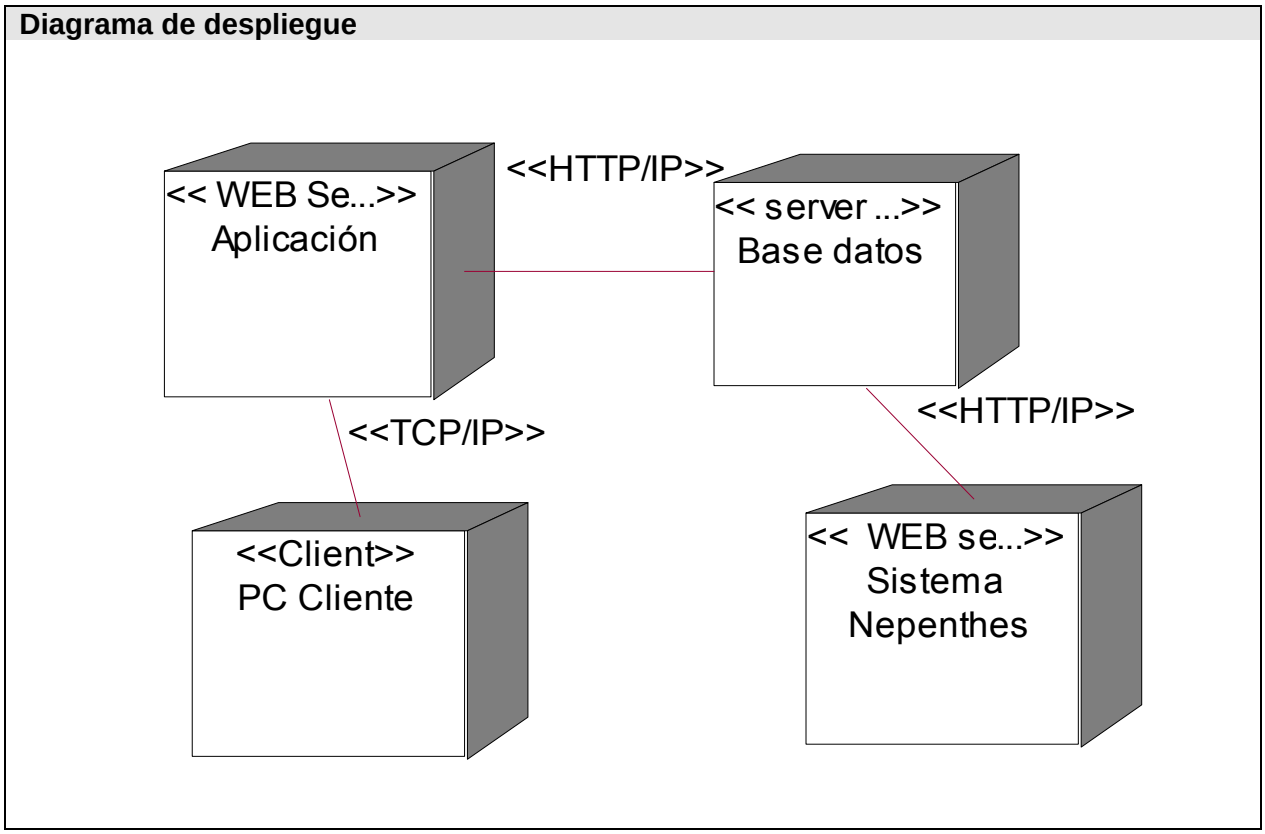


Figura 4.1: Diagrama de Despliegue

4.3 Diagrama de Componentes.

Un diagrama de componentes muestra las dependencias lógicas entre los componentes de software, sean estos componentes, fuentes, binarios o ejecutables. Para este caso representaremos un diagrama general de la aplicación y uno para cada paquete en que agrupamos.

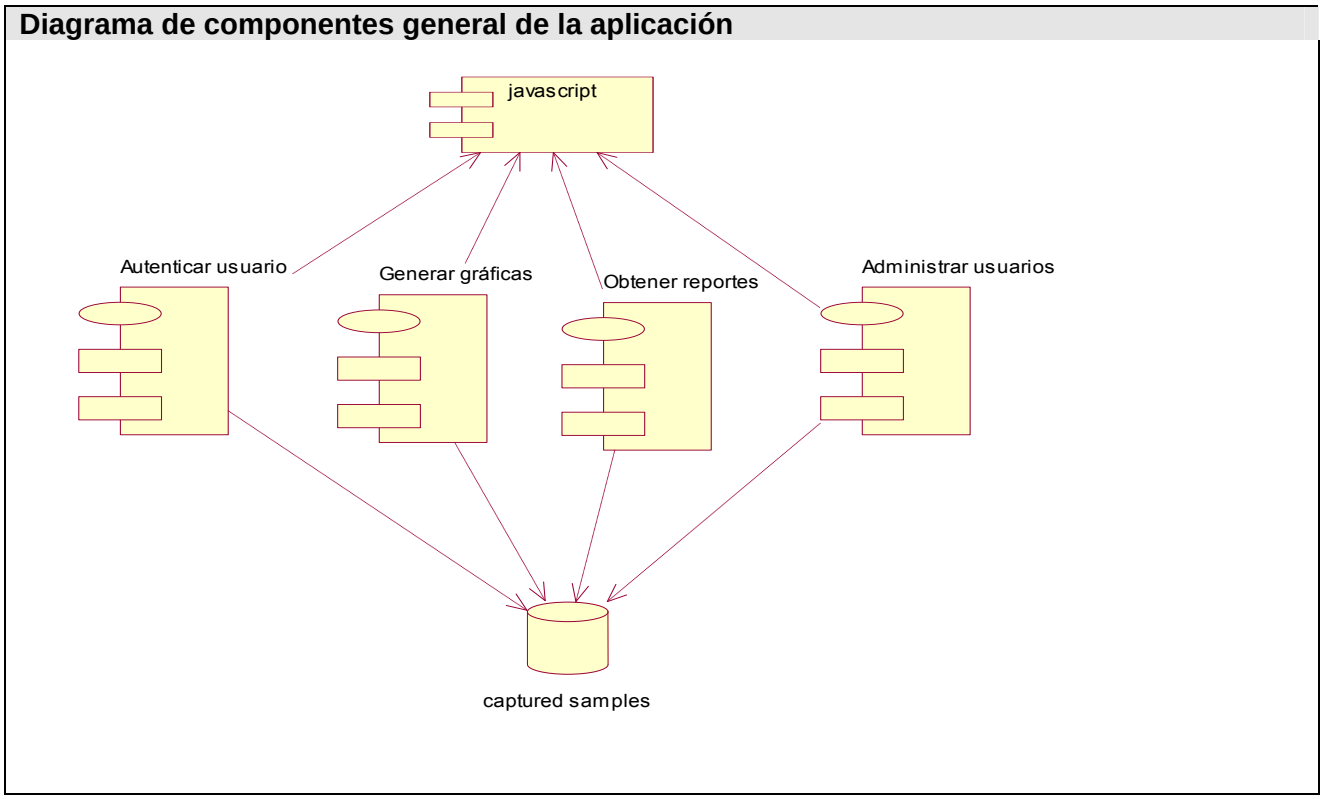


Figura 4.2: Diagrama de componentes general de la aplicación.

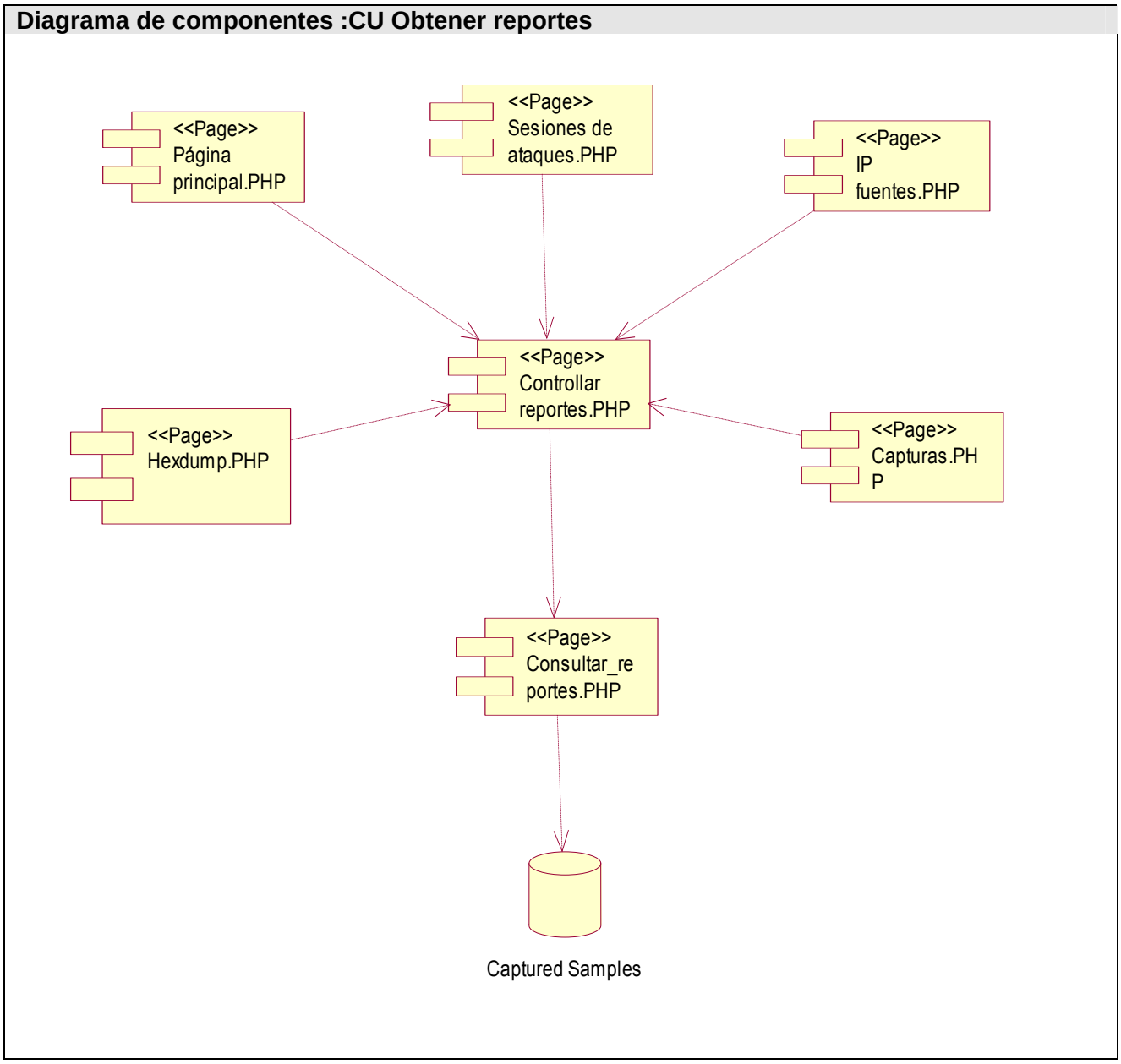


Figura 4.3: Diagrama de componentes: CU Obtener reportes

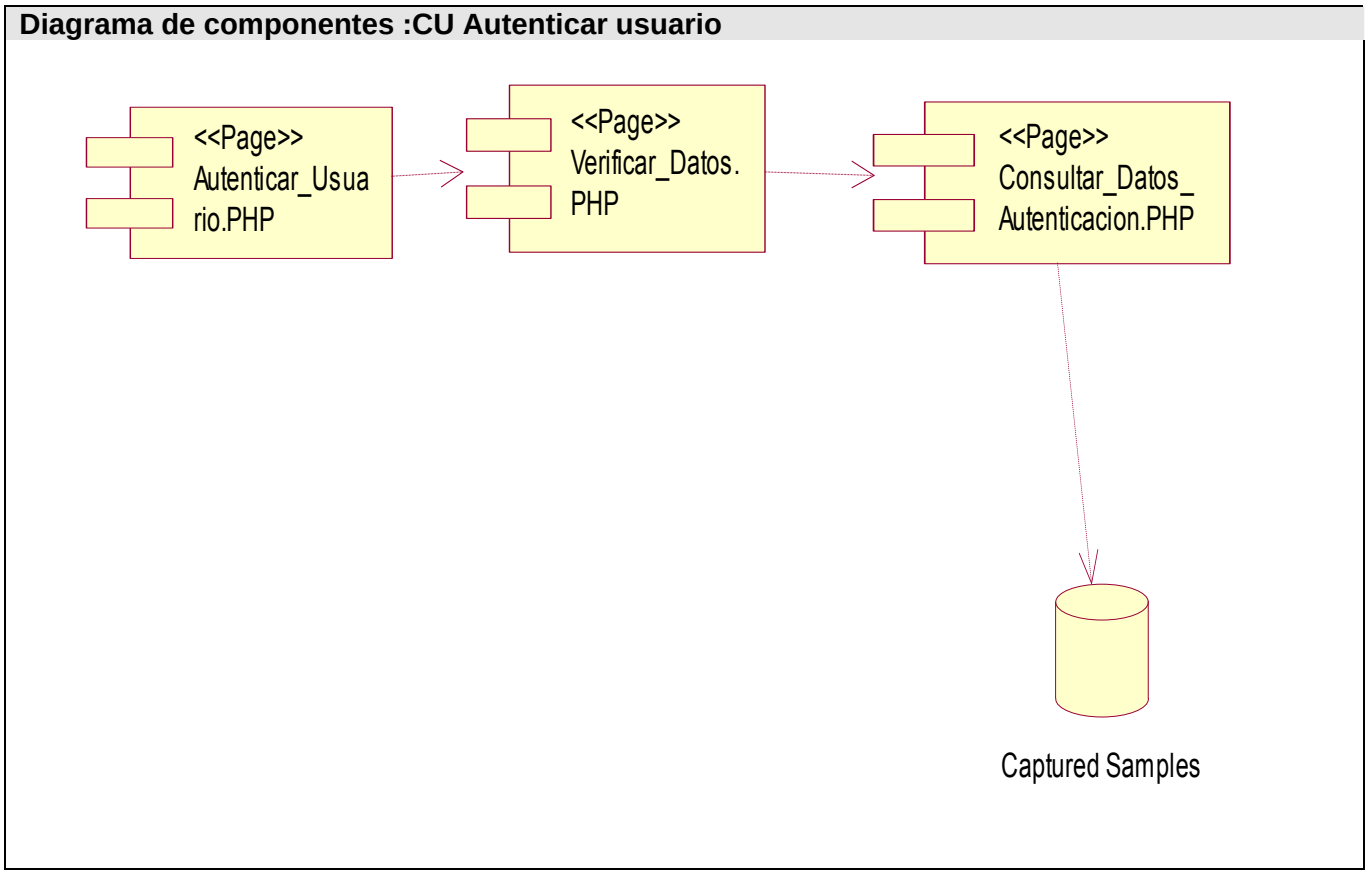


Figura 4.4: Diagrama de componentes: CU Autenticar Usuario

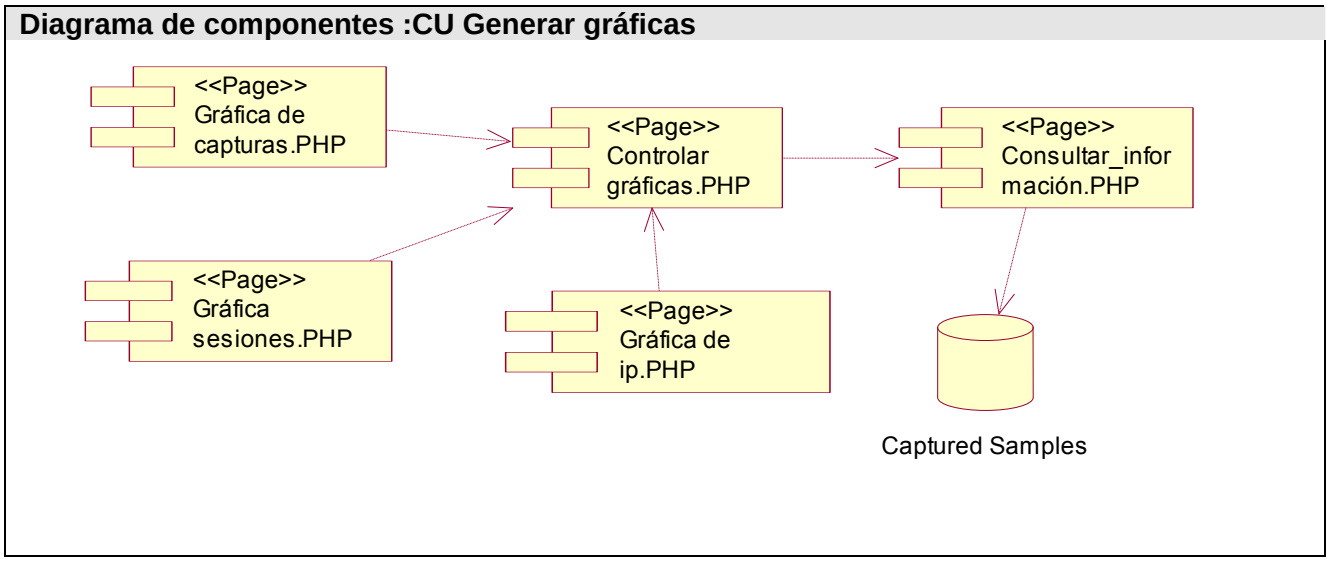


Figura 4.5: Diagrama de componentes: CU Generar Gráficas

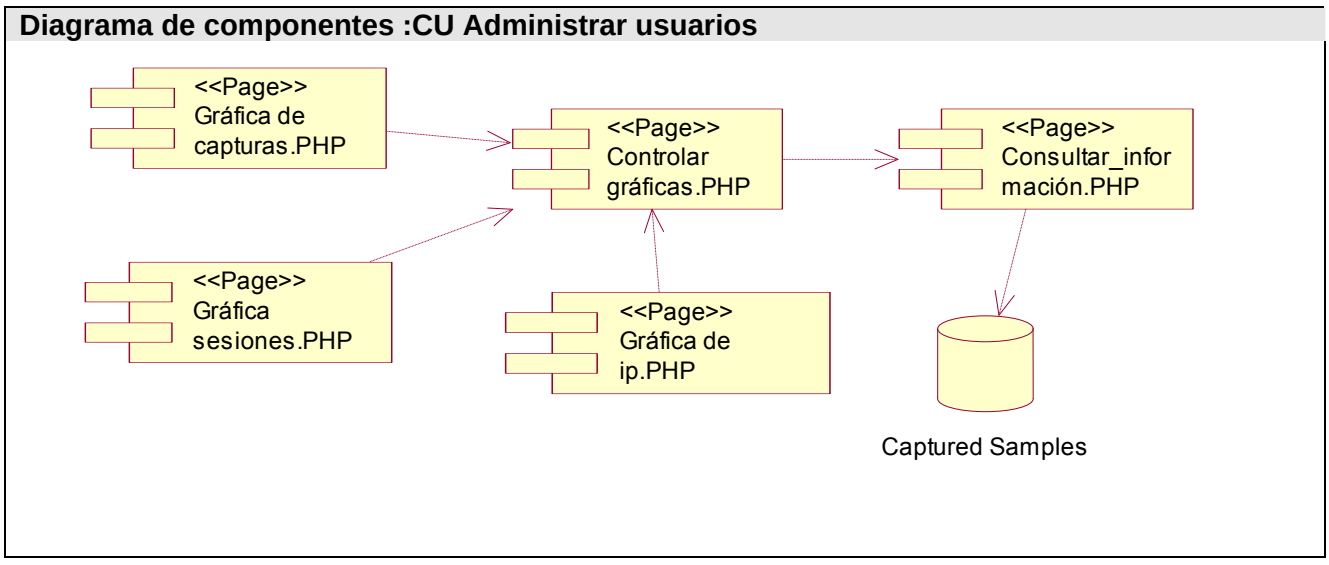


Figura 4.6: Diagrama de componentes: CU Administrar usuarios

4.5 Conclusiones

En este capitulo obtuvimos una representación del flujo de implementación a través del diagrama de despliegue del sistema que muestra la situación física de la aplicación, así como los diagramas de componentes que representan cada parte modular del sistema y las relaciones entre ellas .

.

Capítulo 5. Estudio de Factibilidad.

5.1 Introducción

Para la realización de una aplicación es muy importante hacer un análisis del costo de la misma, por tanto, en el presente capítulo se hará un estudio de la factibilidad de la aplicación .Para ello se aplicará la técnica de estimación por caso de uso que nos servirá para calcular el costo, tiempo de desarrollo, el esfuerzo y la cantidad de personas que se necesitan para desarrollar el sistema.

5.2 Planificación basada en casos de uso.

Aplicación de una técnica de estimación de esfuerzo y tiempo de desarrollo por Puntos de Casos de Uso

Paso 1. Identificar los Puntos de casos de uso Desajustados.

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

Tipo de actor	Descripción	Factor de peso	Actores	Total
Simple	Sistema con sistema a través de interfaz de programación.	1	0	0
Medio	Sistema con sistema mediante protocolo de interfaz basada en texto.	2	0	0
Complejo	Persona que interactúa con el sistema mediante interfaz gráfica.	3	1	3

Tabla 5.1 Factor de peso de los actores sin ajustar.

UAW = Σ (Factor* Actores)

UAW = 1*3

UAW = 3

Tipo de CU	Descripción	Peso	Cant. de CU	Total
Simple	El caso de uso tiene de 1 a 3 transacciones.	5	1	5
Medio	El caso de uso tiene de 4 a 7 transacciones.	10	3	30
Complejo	El caso de uso tiene más de 8 transacciones.	15	0	0

Tabla 5.2 Factor de peso de los casos de uso sin ajustar.

UUCW = Σ (Factor *Cant CU)

UUCW = 5+30

UUCW = 35

Luego:

UUCP = UAW + UUCW

UUCP = 3+35

UUCP = 38

Paso 2. Cálculo de los Puntos de casos de uso ajustados.

UCP = UUCP *TCF * 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

El factor de complejidad técnica (**TCF**) se calcula mediante la cuantificación de un conjunto de factores que determinan la complejidad técnica del sistema. Cada factor se cuantifica en un valor desde 0 (aporte irrelevante) hasta 5 (aporte muy relevante).

Factor	**Descripción**	**Peso**	**Valor asignado**	**Total**
T1	Sistema distribuido	2	0	0
T2	Tiempo de respuesta	1	3	3
T3	Eficiencia del usuario final	1	3	3
T4	Funcionamiento Interno complejo	1	4	4
T5	El código debe ser reutilizable	1	2	2
T6	Facilidad de instalación	0.5	5	2.5
T7	Facilidad de uso	0.5	5	2.5
T8	Portabilidad	2	5	10
T9	Facilidad de cambio	1	3	3
T10	Concurrencia	1	2	2
T11	Incluye objetivos especiales de seguridad	1	3	3
T12	Provee acceso directo a terceras partes	1	0	0
T13	Se requieren facilidades especiales de entrenamiento de usuarios	1	1	1
Total				36

Tabla 5.3 Factor de complejidad técnica.

TCF = 0.6 +0.01 * Σ (Peso* Valor)

TCF = 0.6 + 0.01 *36

TCF = 0.96

Factor	Descripción	Peso	Valor asignado	Total
E1	Familiaridad con el modelo de proyecto utilizado	1.5	3	4.5
E2	Experiencia en la aplicación	0.5	2	1
E3	Experiencia en la orientación a objetivos.	1	3	3
E4	Capacidad del analista líder.	0.5	4	2
E5	Motivación.	1	3	3
E6	Estabilidad de requerimientos	2	5	10
E7	Personal Part–Time	-1	2	-2
E8	Dificultad del lenguaje de programación	-1	4	-4
Total				17.5

Tabla 5.4 Factor de ambiente.

EF = 1.4 – 0.03 * Σ (Peso*Valor)

EF = 1.4 – 0.03 * 17.5

EF = 0.875

Luego:

UCP = UUCP * TCF * EF

UCP =38 * 0.96 * 0.875

UCP = 31,92

Paso 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 obtener el factor de conversión (CF) se cuentan cuántos valores de los que afectan el factor ambiente (E1...E6) están por debajo de la media (3), y los que están por arriba de la media para los restantes (E7, E8). Si el total es 2 o menos se utiliza el factor de conversión 20 Horas-Hombre / Punto de Casos de uso. Si el total es 3 o 4 se utiliza el factor de conversión 28 Horas-Hombre / Punto de Casos de uso. Si el total es mayor o igual que 5 se recomienda efectuar cambios en el proyecto ya que se considera que el riesgo de fracaso del mismo es demasiado alto.

CF = 20 horas-hombre (si Total EF ≤ 2)
CF = 28 horas-hombre (si Total EF = 3 ó Total EF = 4)
CF = abandonar o cambiar proyecto (si Total EF ≥ 5)

$Total_{EF}$ = Cant EF < 3 (entre E1 –E6) + Cant EF > 3 (entre E7, E8)

Como Total EF = 1 + 1
Total EF = 2
CF = 20 horas-hombre (porque Total EF ≤ 2)

Luego:
E = UCP * CF
E = 31,92 * 20 horas-hombre
E = 638,4 horas-hombre

Paso 4. Calcular esfuerzo de todo el proyecto.

Actividad	Porcentaje %	Horas-Hombres
Análisis	20	425,6
Diseño	35	744,8
Implementación	30	638,4
Pruebas	5	106,4
Sobrecarga (otras actividades)	10	2128
Total	100	2128

Tabla 5.5 Esfuerzo total de todo el proyecto.

Suponiendo que una persona trabaje 8 horas por día, y un mes tiene como promedio 24 días laborables; la cantidad de horas que puede trabajar una persona en 1 mes es 192 horas

Si E_T = 2128 horas-hombre y por cada 192 horas para una persona me daría un E_T = 11 mes-hombre.

Costo del Proyecto.

Se asume como salario promedio mensual $100.00

CH: Cantidad de hombres.

CHM: Costo hombres-mes

CHM = CH * Salario Promedio

CHM = 2 * 100

CHM = $ 200.00

Costo Total
Costo = CHM * E / CH
Costo = 200.00 * 11 / 2
Costo = $1100

De lo anteriormente calculado se puede concluir que con 1 hombre se realiza el proyecto en 11 meses, con dos personas en la mitad de este tiempo y su costo esta estimado en 1100 pesos.

5.3 Beneficios Tangibles e intangibles.

El Sistema realizado para crear reportes, tiene como objetivo principal solucionar el problema al que se enfrentan los analistas de Segurmática cuando tienen que examinar las capturas.
El beneficio fundamental de este sistema es poseer una aplicación Web, manejable y de interfaz agradable que le permita conocer de una forma más precisa y en el menor tiempo posible los datos de interés de los analistas.

Por tanto, podemos plantear los siguientes beneficios:

- Disminución del tiempo y esfuerzo que invierten los analistas a la hora de examinar nuevas capturas e información de nuevos ataques, que hasta estos momentos era un proceso muy engorroso.
- Disminución de la cantidad de información irrelevante con la que tenían que trabajar.
- Fácil y rápido acceso a la información actualizada.
- Fácil procesamiento de la información y obtención, dinámica, de reportes.

5.4 Análisis de costo.

Antes de desarrollar una aplicación es muy importante que primero hagamos un análisis de su costo , ya que este nos dirá si los beneficios que ella nos reportará se corresponden con su costo y de acuerdo a esto se considerará si se continua o no con su desarrollo.
El sistema que se propone está dirigido fundamentalmente apoyar uno de los métodos que se utilizan en la seguridad informática. Una vez implementado el sistema contribuirá en gran mediada con el trabajo de los analistas de seguridad Informática.

Analizando el costo del proyecto, los numerosos beneficios que reporta, detallados con anterioridad, se puede concluir que su implementación es realmente factible.

5.5 Conclusiones

En este capítulo se describió el estudio de factibilidad realizado al sistema propuesto a través de la técnica de puntos de casos de uso, mediante el cuál se obtuvo el costo y tiempo de desarrollo, cantidad de personas, es decir datos con los que podemos medir la viabilidad del proyecto. Se analizaron además los beneficios que este nos reportará al ser implantado.
Por todo esto se llegó a la conclusión que es factible implementar el sistema propuesto.
.

Conclusiones

La seguridad informática involucra gran cantidad de actividades, para mantener los sistemas protegidos contra amenazas y ataques, de ellos los más comunes son los relacionados con la aparición de códigos malignos y vulnerabilidades del software, a la vez, son los más fáciles de enfrentar, puesto que se requiere principalmente de aplicar las medidas preventivas inmediatamente al descubrimiento de la amenaza o falla, en estos casos, la cooperación del usuario facilita la labor de los encargados de mantenimiento.

Después de realizado el tema tratado en este trabajo, profundizado en las herramientas de captura de programas malignos, se hizo más fácil la comprensión de los datos con los que trabaja el sistema.

Para el cumplimiento de los objetivos trazados y el desarrollo de esta aplicación, se hizo un estudio de las técnicas y tecnologías más convenientes. Posteriormente se hizo la captura de los requisitos funcionales y no funcionales, y la agrupación de los de los casos de uso, según las funcionalidades del sistema. A continuación se transitó por las etapas de negocio, requerimientos, análisis y diseño utilizando los artefactos de RUP para modelar toda la ingeniería del software, donde surgieron los diagramas que están presentes en el documento. Seguidamente se desarrolló la implementación concluyendo así las etapas por las que transitó el sistema.

Por todo lo anterior se concluye que el objetivo principal propuesto en el presente trabajo, se cumplió satisfactoriamente: desarrollar un sistema informático robusto, de interfaz amigable que genere reportes de las amenazas capturadas en Segurmática.

Recomendaciones

Realizadas las conclusiones y logrado el objetivo del trabajo, se recomienda lo siguiente:

- Dar seguimiento al estudio de los sistemas que capturan programas malignos, para garantizar mejoras, en futuras versiones del sistema.
- Continuar el desarrollo de este sistema perfeccionándolo aún más, de forma tal que se le añadan nuevas funcionalidades, que puedan surgir según las necesidades de los usuarios de la empresa de Segurmática.
- Poner a prueba el sistema durante un período de tiempo significativo, para comprobar su desempeño y que las funcionalidades del sistema se correspondan con la necesidad del usuario.
- Que se le añadan nuevos requerimientos al caso de uso del sistema Generar gráfica, por ejemplo, que además de las gráficas que muestran la cantidad de programas malignos, sesiones de ataques y IP fuentes, se creen otras que contengan otro tipo de información que sea relevante y útil para el cliente.

Bibliografía

Álvarez, M. A. DesarrolloWeb.com, 2006 [Disponible en: http://www.desarrolloweb.com/articulos/25.php]

Ferrari, L. G. Usar Nepenthes, Honeypots, para detectar malware común, 2007 [Disponible en: http://talsoft.com.ar/weblog/?p=66]

Gabriel Verdejo Álvarez. Honeypots y Honeynets, 2004 [Disponible en: *http://tau.uab.es/~gaby*]

Gilfillan, L. La Biblia de MySQL, SYBEX, 2003. 880 páginas

Honeypot, 2007 [Disponible en: http://es.wikipedia.org/wiki/Honeypot]

Ivar Jacobson, Grady Booch, James Rumbaugh. El proceso unificado de desarrollo de software , Ciudad de la Habana, Cuba, Félix Varela, 2004. 438 páginas.

Matthew Norman. Database Design Manual using MySQL for Windows, Estados Unidos de América, Springer, 2004. 221 páginas.

Michael Kofler. The Definitive Guide to MySQL5, New York, Estados Unidos, APRESS, 2005. 785 páginas.

Nepenthes, 2006 [Disponible en: http://nepenthes.mwcollect.org/]

Remko Lodder.Conoce a tu amigo Honeynet, 2003 [Disponible en: http://project.honeynet.org]

Rubén Aquino Luna, D. J. D. Experiencias Académicas, Honeynet DSC/UNAM-CERT, 2006 [Disponible en: http://www.seguridad.unam.mx/honeynet]

Referencias Bibliográficas.

[1]. Paz, A. Guru de la informática, 2007. [Disponible en: http://vtroger.blogspot.com/]

[2]. Alvarez, G.V. Trabajo de investigación para el DEA ,2004. [Disponible en: http://tau.uab.es/~gaby]

[3]. Rondón, J.C.T.y.R.G. Control, Administración e Integridad de los logs, 2007. [Disponible en: http://www.wikilearning.com/que_es_un_log-wkccp-3485-2.htm]

[4]. Wikipedia. Bases de datos | Sistemas de gestión de bases de datos libres, 2007.[Disponible en:http://es.wikipedia.org/wiki/HTML.

[5]. Wikipedia. Esbozo software | Editores de páginas web, 2007.[Disponible en: http://es.wikipedia.org/wiki/Dreamweaver]

[6]. Softonic. Herramientas para administrar una base de datos MySQL, 2007 [Disponible en: http://ems-mysql-manager.softonic.com/]

Anexos

Anexo 1. Descripción textual de los Casos de Uso del Negocio

Caso de Uso:	Analizar información capturada	
Actores:	Jefe	
Trabajadores:	Sistema de captura de programas , Analista	
Resumen:	El caso de uso se inicia cuando el jefe solicita información de los ataques y programas malignos capturados por un sistema. Esta información antes de ser entregada al jefe debe ser analizada por el analista. Finalizando así el caso de uso.	
Flujo Normal de Eventos		
Sección "1"		
Acción del Actor		**Respuesta del Negocio**
1. El jefe solicita información al analista de los programas malignos capturados.		1.1. El analista atiende la solicitud del jefe. 1.2. Revisa los programas malignos que fueron entregados al laboratorio. 1.3. Identifica los ataques para ver cuáles propagan programas malignos .y cuáles no. 1.4 Si el ataque propagó programas malignos, se actualiza el sistema para que capturar estos nuevos programas. 1.5 Se envía información al jefe
2. El jefe obtiene la información solicitada		
Flujos Alternos		
Acción del Actor		**Respuesta del Negocio**
		1.2. No han sido entregados programas malignos al laboratorio.

	1.4. El ataque no propago programas malignos no se actualiza el sistema.

Caso de Uso:	**Elaborar reportes**	
Actores:	Jefe (inicia)	
Trabajadores:	Analista	
Resumen:	El caso de uso se inicia cuando el jefe solicita información de los ataques que han sucedido en un determinado período de tiempo entonces el analista elabora el reporte de los ataques y lo entrega al jefe. Finalizando así el caso de uso.	
Flujo Normal de Eventos		
Sección "1"		
Acción del Actor		**Respuesta del Negocio**
1. El Jefe solicita información de los ataques sucedidos en un período de tiempo.		1.1. El analista recibe la solicitud del jefe. 1.2. El analista busca los ataques que sucedieron en el período de tiempo. 1.3 El analista elabora el reporte y lo envía al jefe.
2. El jefe obtiene el reporte para conocer las tendencias de los ataques y capturas.		
Flujos Alternos		
Acción del Actor		**Respuesta del Negocio**
		1.2 No hay ataques en ese período de tiempo, entonces se le informa al jefe

Anexo 2. Diagramas de Actividades.

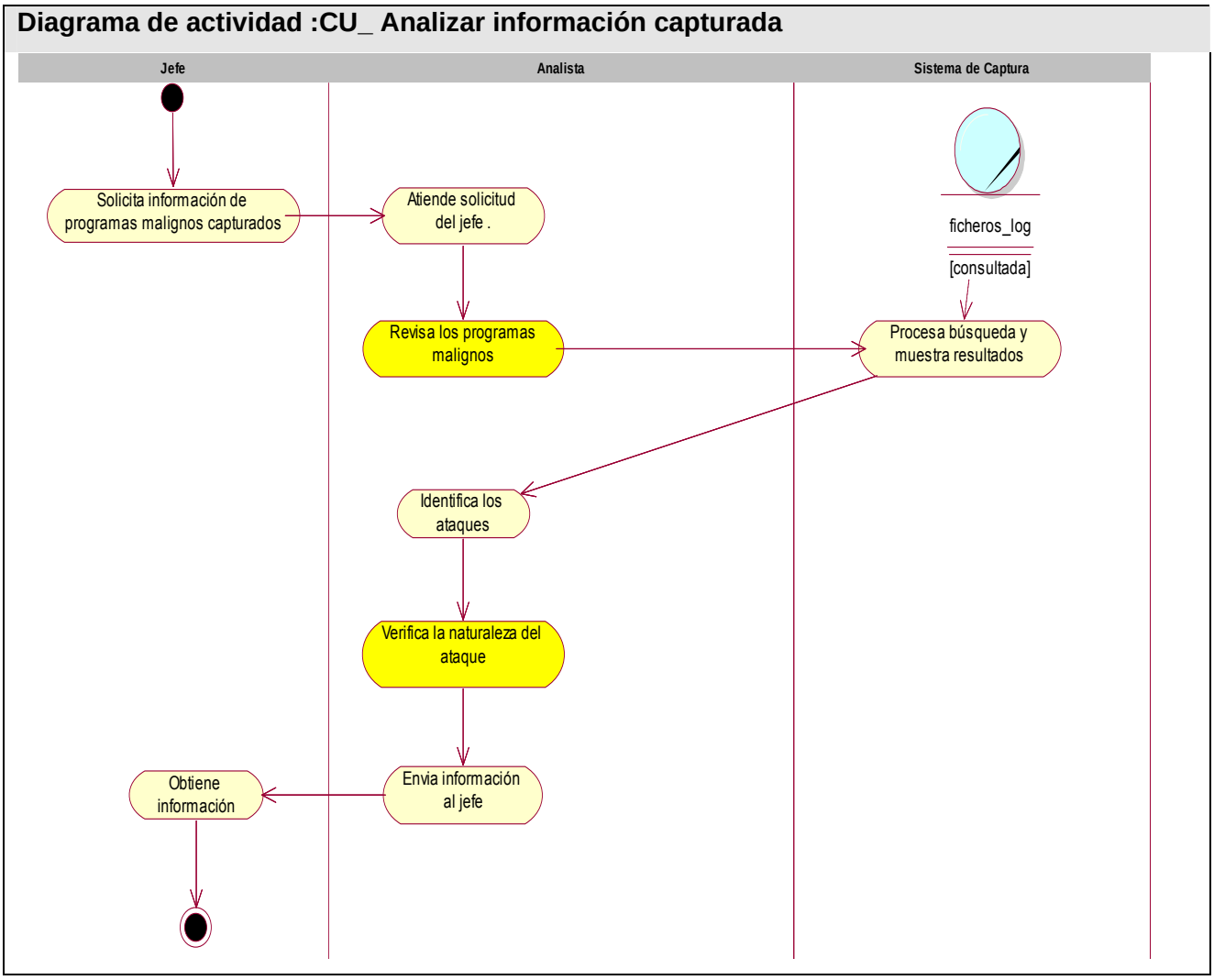
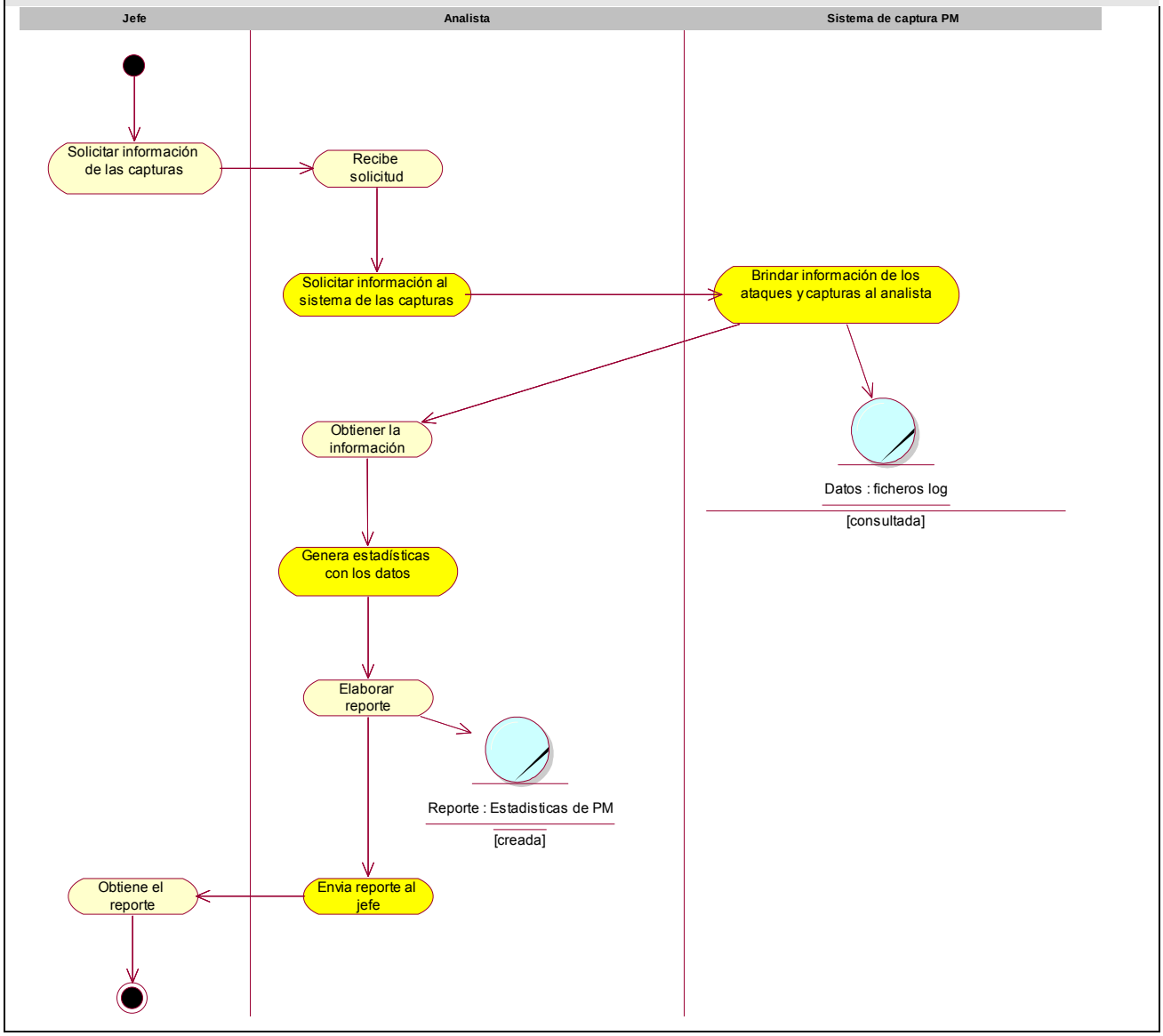
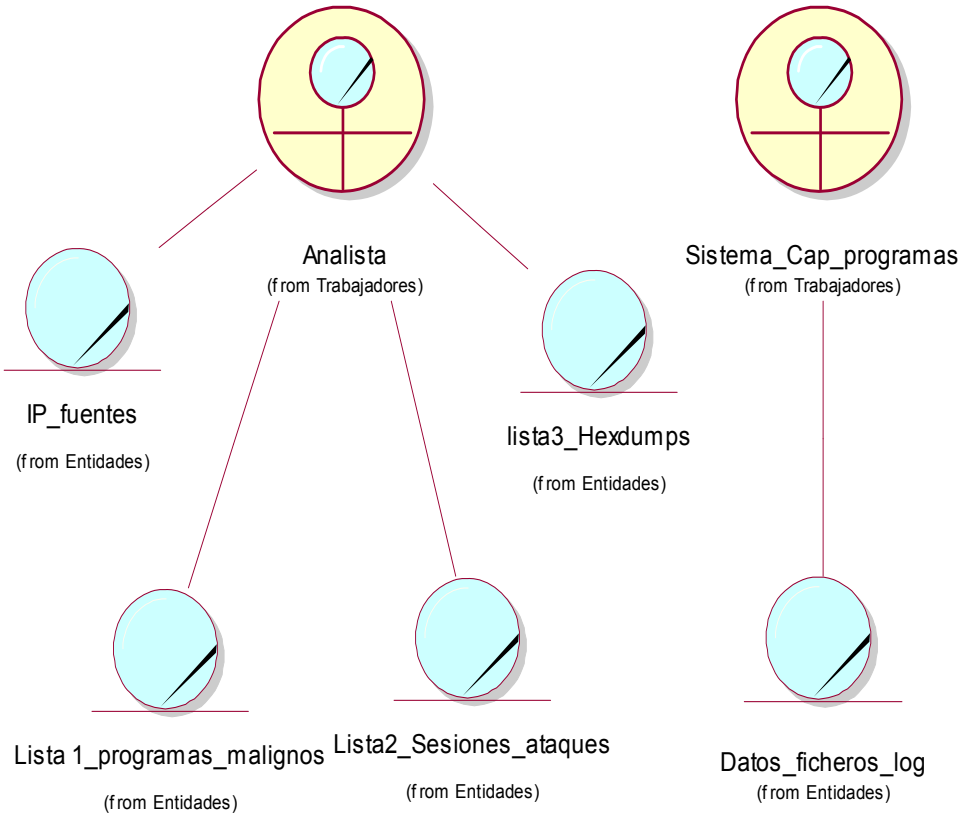


Diagrama de actividad :CU_ Elaborar reportes



Anexo 3 Modelo de objetos

Modelo de objetos

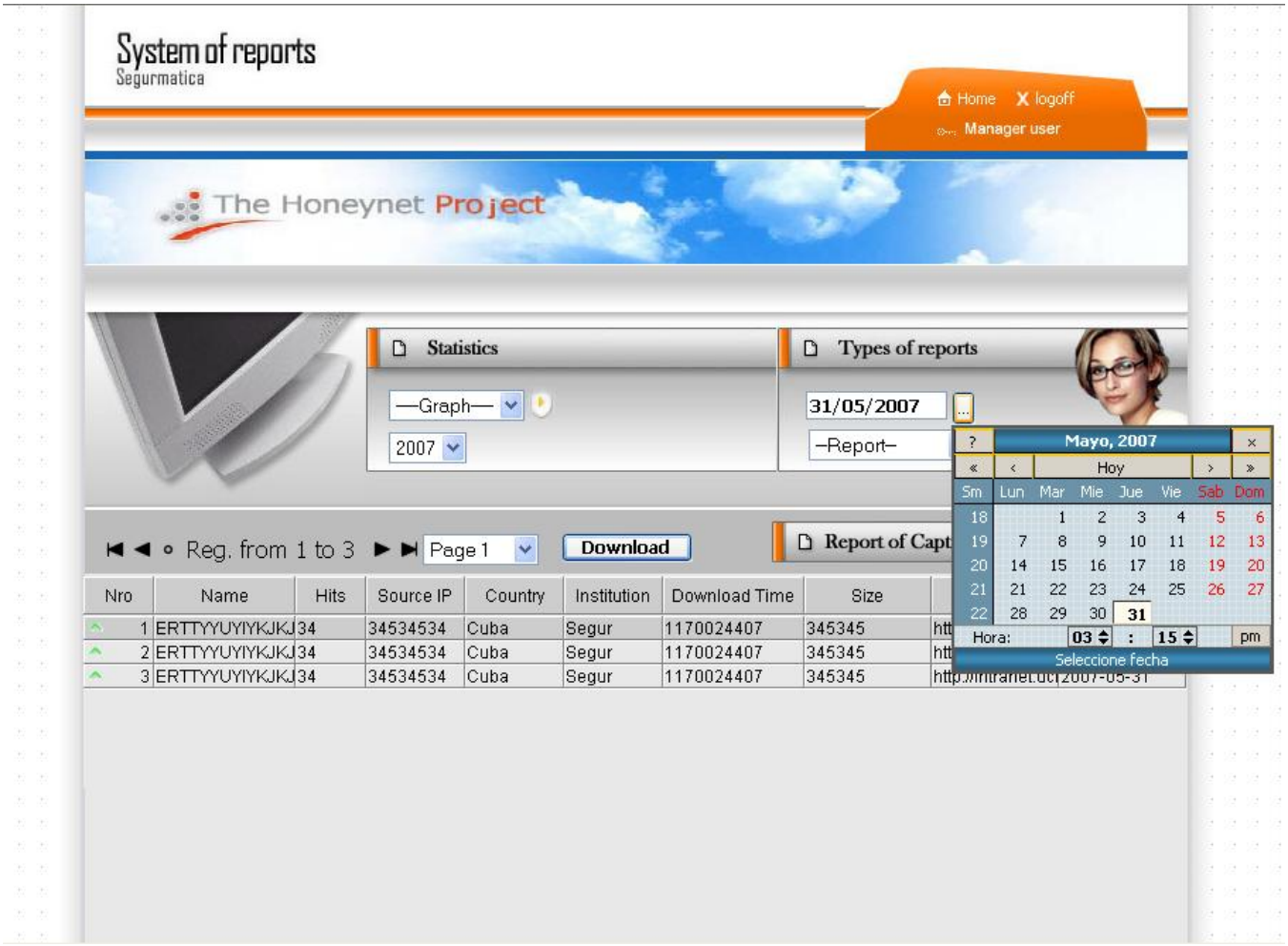


Anexo 4 Descripción de los Casos de uso.

<table>
<tr><td colspan="2">Caso de uso:</td><td colspan="2">Obtener reportes</td></tr>
<tr><td colspan="2">Actores :</td><td colspan="2">Analista (inicia)</td></tr>
<tr><td colspan="2">Propósito:</td><td colspan="2">Permitir al analista registrado en el sistema obtener reportes específicos sobre programas malignos capturados, IP fuentes, sesiones de ataques y que ataques que no propagaron programas malignos.</td></tr>
<tr><td colspan="4">Resumen:
El CUS se inicia cuando el analista selecciona una de las opciones existentes para obtener un reporte, el sistema realiza la acción seleccionada por el usuario y termina el CUS.</td></tr>
<tr><td colspan="4">Referencias: RF 1.1, RF 1.2, RF 1.3, RF 1.4</td></tr>
<tr><td colspan="4">Precondiciones: 1. El analista debe seleccionar un tipo de reporte</td></tr>
<tr><td colspan="4">Poscondiciones: Se crean reportes</td></tr>
<tr><td colspan="4">Curso normal de los eventos:</td></tr>
<tr><td colspan="2">Acción del actor:</td><td colspan="2">Respuesta del proceso del Sistema:</td></tr>
<tr><td>1</td><td>El analista selecciona una fecha y un tipo de reporte.</td><td>1.1</td><td>El sistema muestra la información referente al reporte seleccionado por el analista.</td></tr>
<tr><td colspan="4">Escenario 1: Obtener reportes de Capturas</td></tr>
<tr><td>1</td><td>El analista selecciona la opción de Captura y fecha.</td><td>1.1</td><td>El sistema muestra un listado con todas las capturas en un periodo de tiempo seleccionado.</td></tr>
<tr><td>2</td><td>El analista selecciona la captura que desea analizar y presiona el botón para ir a la dirección de descarga.</td><td>2.1</td><td>El sistema muestra la página desde donde se descargo esta captura.</td></tr>
<tr><td colspan="4">Escenario 2: Obtener reportes de IP Fuentes.</td></tr>
<tr><td>1</td><td>El analista selecciona la</td><td>1.1</td><td>El sistema muestra un listado con</td></tr>
</table>

	opción de IP fuentes		las IP fuentes en un período de tiempo seleccionado
Escenario 3: Obtener reportes de Sesiones de ataques			
1	El analista selecciona la opción de Sesiones de ataques	**1.1**	El sistema muestra un listado con las sesiones de ataques.
Escenario 4:Obtener reportes por ataques que no propagaron programas malignos			
1	El analista selecciona la opción de ataques que no propagaron programas malignos	**1.1**	El sistema muestra un listado de estos ataques.
2	El analista selecciona un ataque en específico y presiona un botón para ver el cuerpo del ataque		El sistema muestra información que contiene el cuerpo del ataque.
Prioridad	Crítico		

Prototipo del caso de uso Obtener reportes.

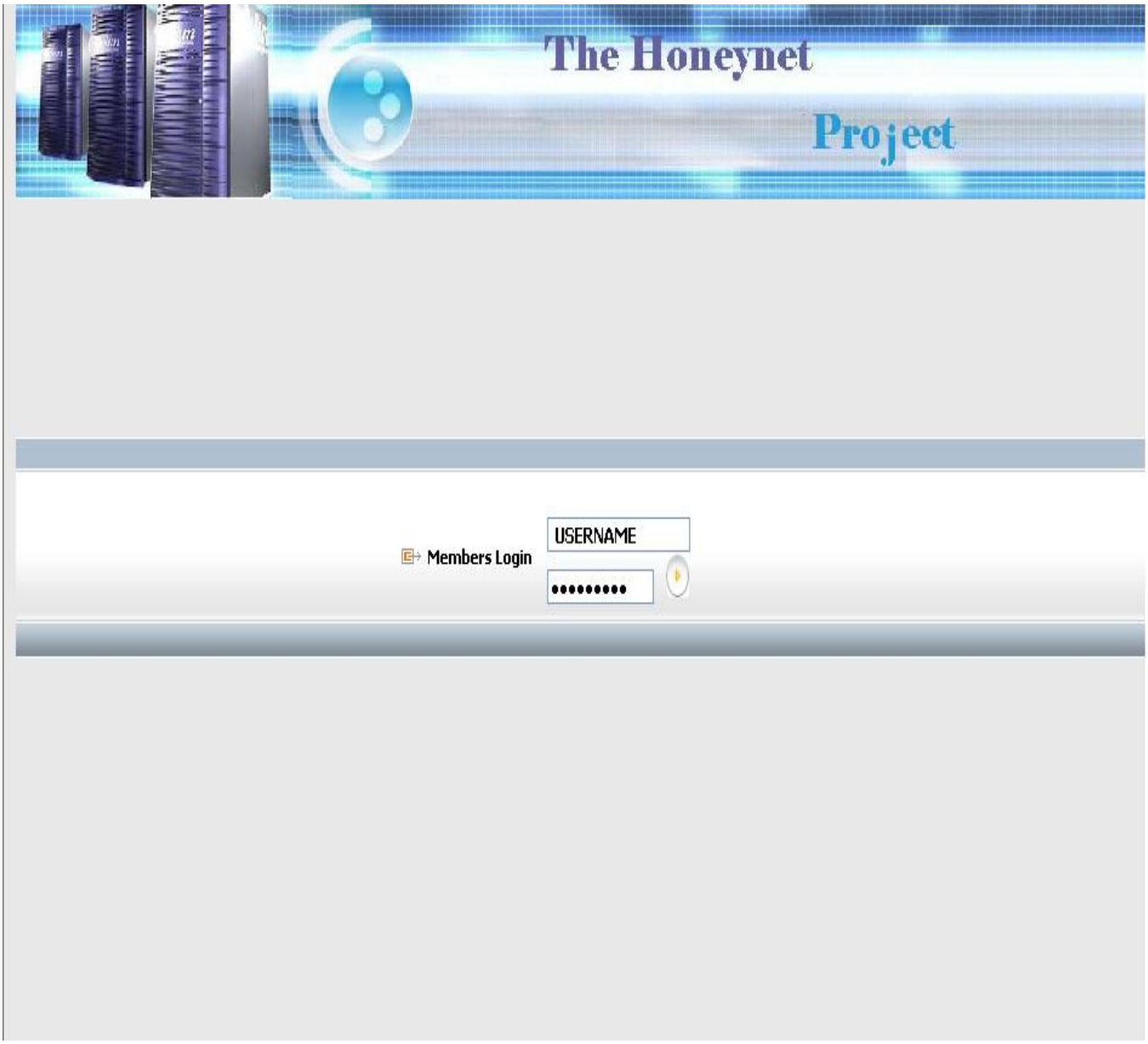


<table>
<tr><td colspan="2">Caso de uso:</td><td colspan="3">Autenticar usuario</td></tr>
<tr><td colspan="2">Actores :</td><td colspan="3">Analista (inicia)</td></tr>
<tr><td colspan="2">Propósito:</td><td colspan="3">Permitir al analista trabajar con la aplicación.</td></tr>
<tr><td colspan="5">Resumen: El CUS se inicia cuando el analista entra sus datos para acceder al sistema, el sistema verifica que los datos entrados estén correctos, para habilitarle la entrada y termina el CUS.</td></tr>
<tr><td colspan="5">Referencias: RF 2, RF 2.1</td></tr>
<tr><td colspan="5">Precondiciones: El analista debe ser usuario del sistema.</td></tr>
<tr><td colspan="5">Poscondiciones: El analista se conecta al sistema</td></tr>
<tr><td colspan="5">Curso normal de los eventos:</td></tr>
<tr><td colspan="2">Acción del actor:</td><td colspan="3">Respuesta del proceso del Sistema:</td></tr>
<tr><td>1</td><td>El analista entra usuario y contraseña.</td><td>1.1

1.2</td><td colspan="2">El sistema valida los datos entrados

Si son correctos habilita la entrada al sistema</td></tr>
<tr><td colspan="5">Flujo alternativo</td></tr>
<tr><td colspan="2">Acción del actor:</td><td colspan="3">Respuesta del proceso del Sistema:</td></tr>
<tr><td></td><td></td><td>1.3</td><td colspan="2">Si no son correctos envía un mensaje de error al usuario.</td></tr>
<tr><td colspan="2">Prioridad :</td><td colspan="3">Crítico</td></tr>
</table>

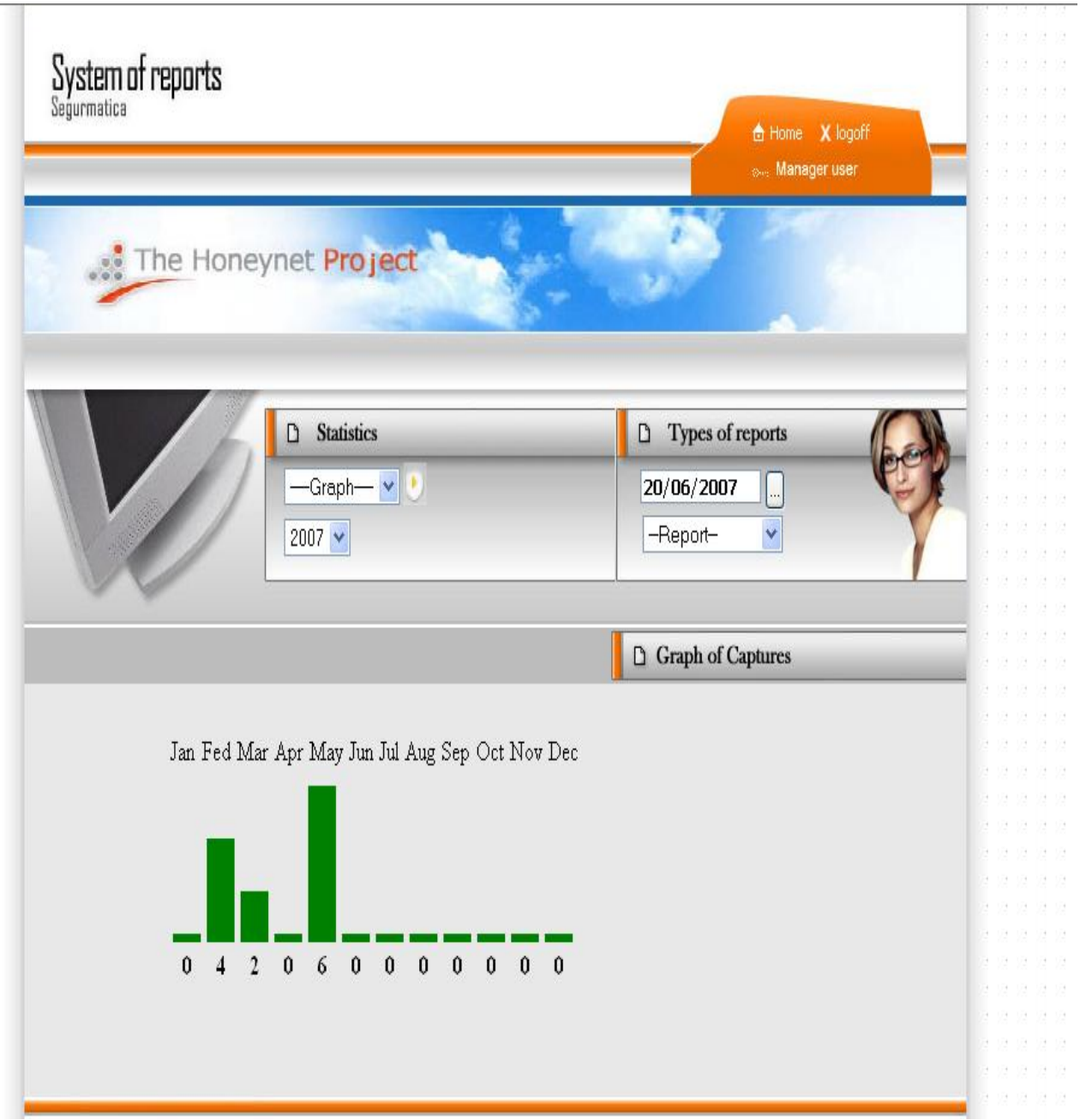
Prototipo del caso de uso Autenticar usuario.



<table>
<tr><td colspan="2">Caso de uso:</td><td colspan="2">Generar gráficas</td></tr>
<tr><td colspan="2">Actores :</td><td colspan="2">Analista (inicia)</td></tr>
<tr><td colspan="2">Propósito:</td><td colspan="2">Permitir al analista registrado en el sistema ver información a través de gráficas</td></tr>
<tr><td colspan="4">Resumen:
El CUS se inicia cuando el analista selecciona en la página principal una opción para ver gráficas, el sistema realiza la acción seleccionada por el usuario y termina el CUS.</td></tr>
<tr><td colspan="4">Referencias: RF 3.1, RF 3.2, RF 3.3</td></tr>
<tr><td colspan="4">Precondiciones: El analista debe seleccionar la opción para ver un tipo de gráfica.</td></tr>
<tr><td colspan="4">Poscondiciones: Quedan generadas las gráficas</td></tr>
<tr><td colspan="4">Curso normal de los eventos:</td></tr>
<tr><td colspan="2">Acción del actor:</td><td colspan="2">Respuesta del proceso del Sistema:</td></tr>
<tr><td>1</td><td>El analista selecciona un año y un tipo de gráfica.</td><td>1.1</td><td>El sistema muestra la gráfica con las estadísticas correspondientes.</td></tr>
<tr><td colspan="4">Escenario 1: Generar gráficas de Capturas</td></tr>
<tr><td>1</td><td>El analista selecciona capturas y año.</td><td>1.1</td><td>El sistema muestra la gráfica correspondiente con la cantidad de capturas por meses, del año seleccionado.</td></tr>
<tr><td colspan="4">Escenario 2: Generar gráficas de Sesiones de ataques</td></tr>
<tr><td>1</td><td>El analista selecciona sesiones de ataques y año.</td><td>1.1</td><td>El sistema muestra la gráfica correspondiente con la cantidad de sesiones de ataques por meses del año seleccionado.</td></tr>
<tr><td colspan="4">Escenario 3: Generar gráficas de IP fuentes</td></tr>
<tr><td>1</td><td>El analista selecciona IP fuentes y año.</td><td>1.1</td><td>El sistema muestra la gráfica correspondiente con la cantidad de IP fuentes por meses del año seleccionado.</td></tr>
</table>

Prioridad	Secundario

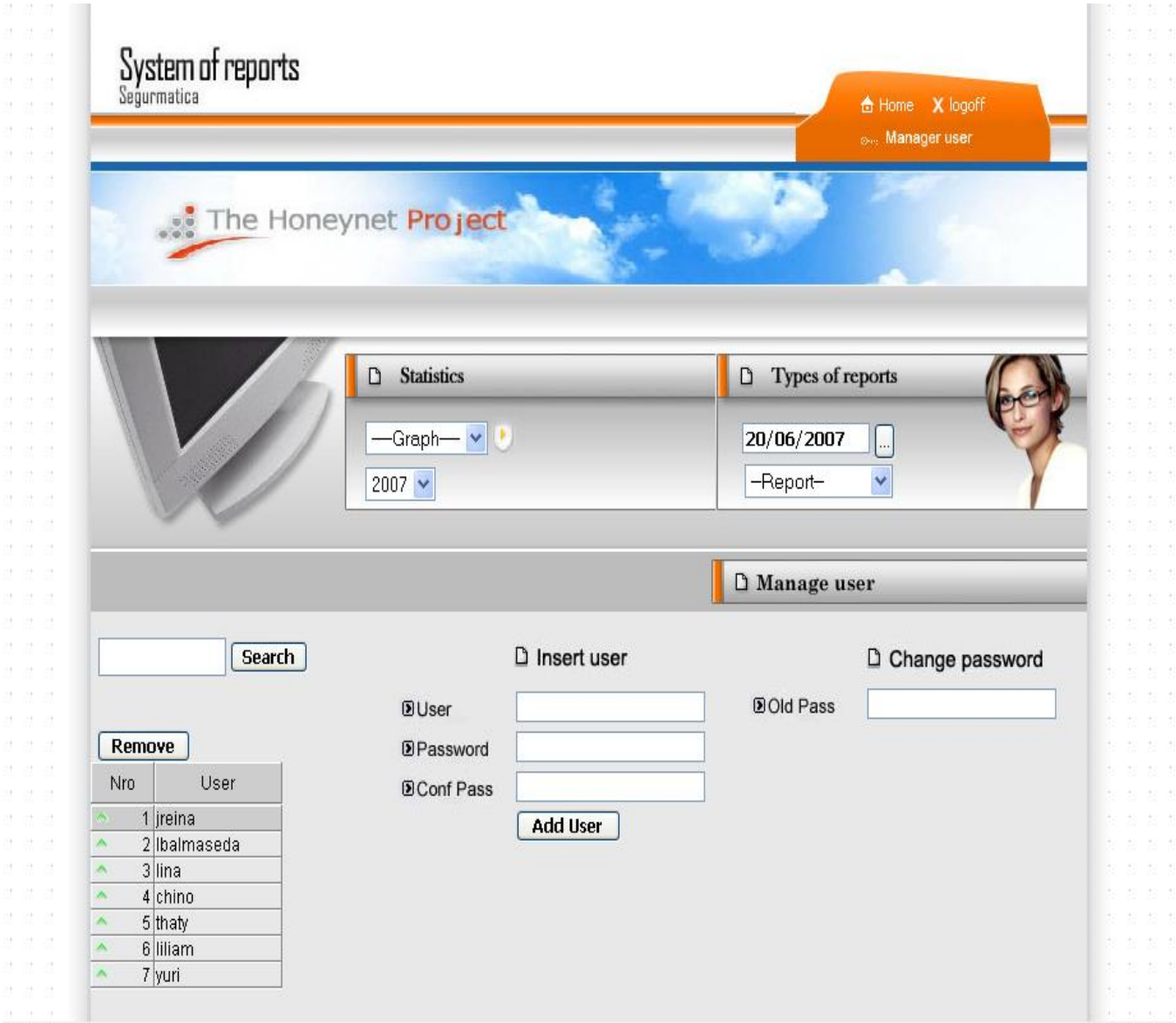
Prototipo del caso de uso Generar gráficas.



Caso de uso:	Administrar usuarios		
Actores :	Analista (inicia)		
Propósito:	Permitir al analista registrado en el sistema realizar una serie de opciones para administrar usuarios.		
Resumen: El CUS se inicia cuando el analista selecciona la opción existente para administrar usuario, el sistema realiza la acción seleccionada por el usuario y termina el CUS.			
Referencias: RF 4.1, RF 4.2, RF 4.3			
Precondiciones: El analista debe ser usuario del sistema.			
Poscondiciones: Se agregan nuevos usuarios al sistema.			
Curso normal de los eventos:			
Acción del actor:		***Respuesta del proceso del Sistema:***	
1	El analista selecciona la opción administrar usuario.	**1.1**	El sistema muestra la opción de buscar, eliminar, adicionar y cambiar contraseña de usuario.
Escenario 1: Buscar y eliminar usuarios			
1	El analista selecciona la opción de buscar usuario.	**1.1**	El sistema muestra todos usuarios
2	El analista selecciona el usuario que desea eliminar y presiona el botón para eliminarlo.	**2.1**	El sistema elimina el usuario seleccionado.
Escenario 2: Adicionar usuarios.			
1	El analista llena los campos correspondientes para crear un nuevo usuario.	**1.1**	El sistema adiciona un nuevo usuario y muestra un listado con los usuarios del sistema.
Escenario 3: Cambiar contraseña de usuarios			
1	El analista llena los campos correspondientes para cambiar la contraseña.	**1.1**	El sistema cambia la contraseña de l usuario.

Prioridad	Crítico

Prototipo del caso de uso Administrar usuarios.



Anexo 5 Diagramas de interacción

Diagrama de secuencia :CU Obtener reportes(Escenario Obtener reporte de capturas)

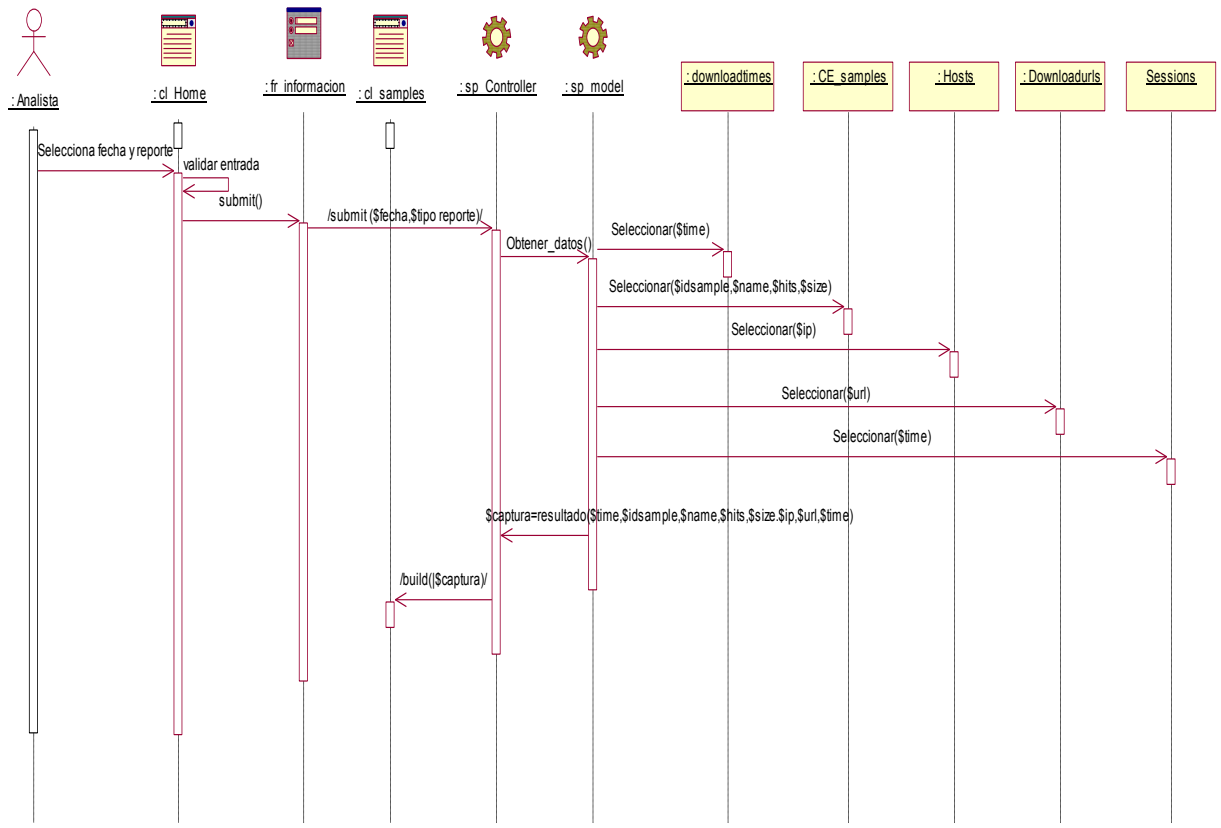


Diagrama de secuencia :CU Obtener reportes(Escenario Obtener reporte de sesiones de ataques)

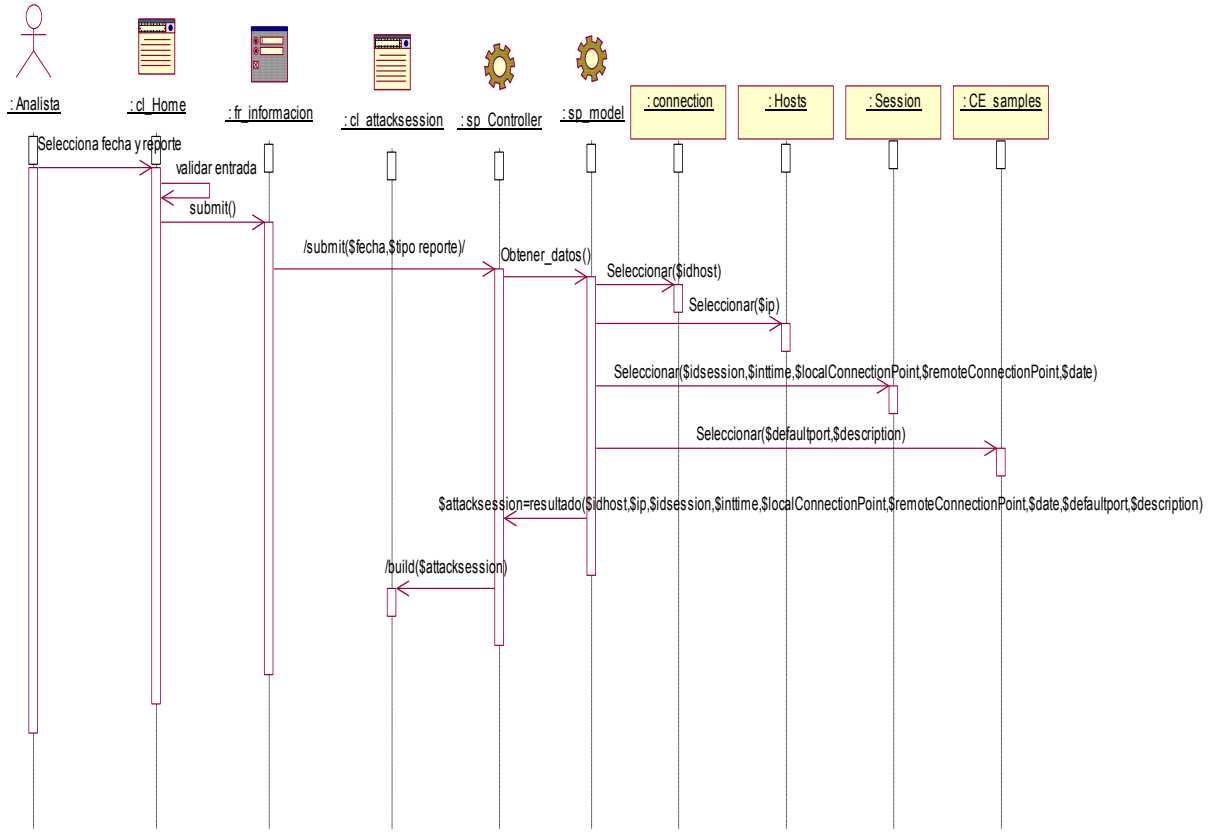


Diagrama de secuencia :CU Obtener reportes(Escenario Obtener reporte de Hexdumps)

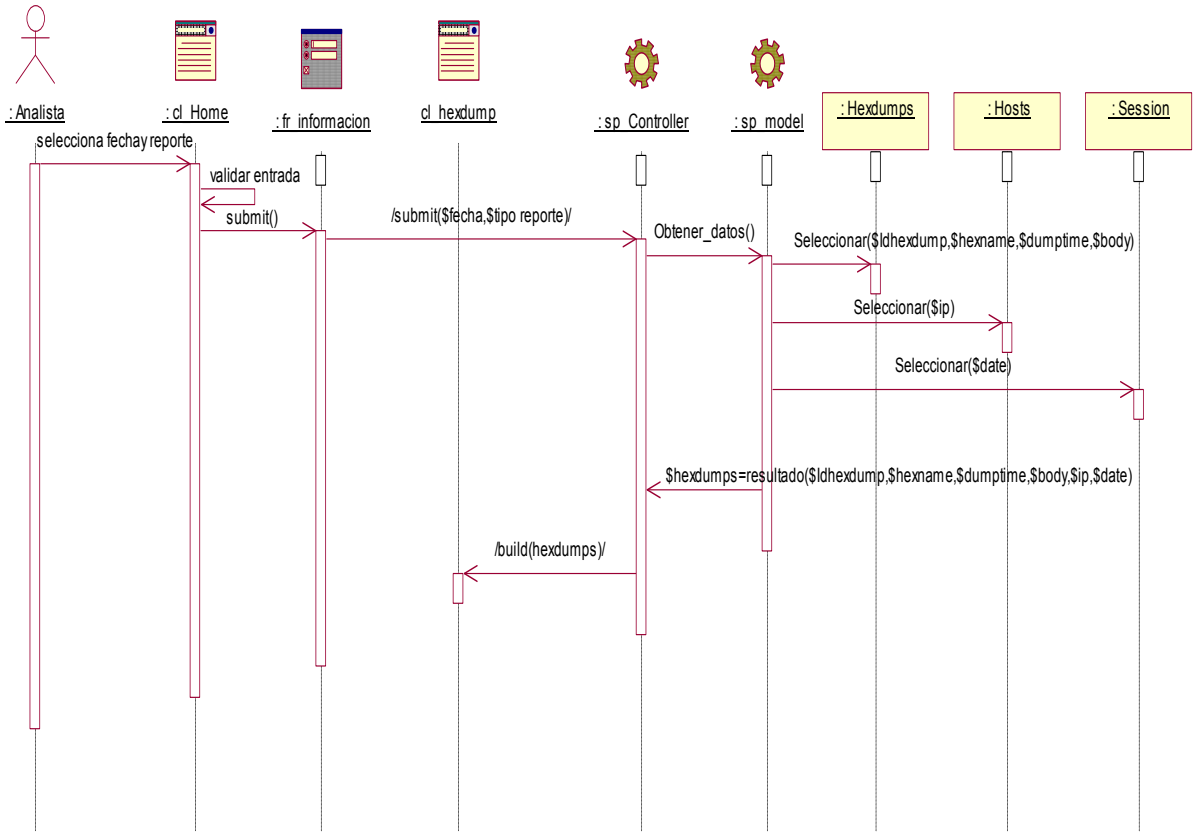


Diagrama de secuencia :CU Obtener reportes(Escenario Obtener reporte de IP fuentes)

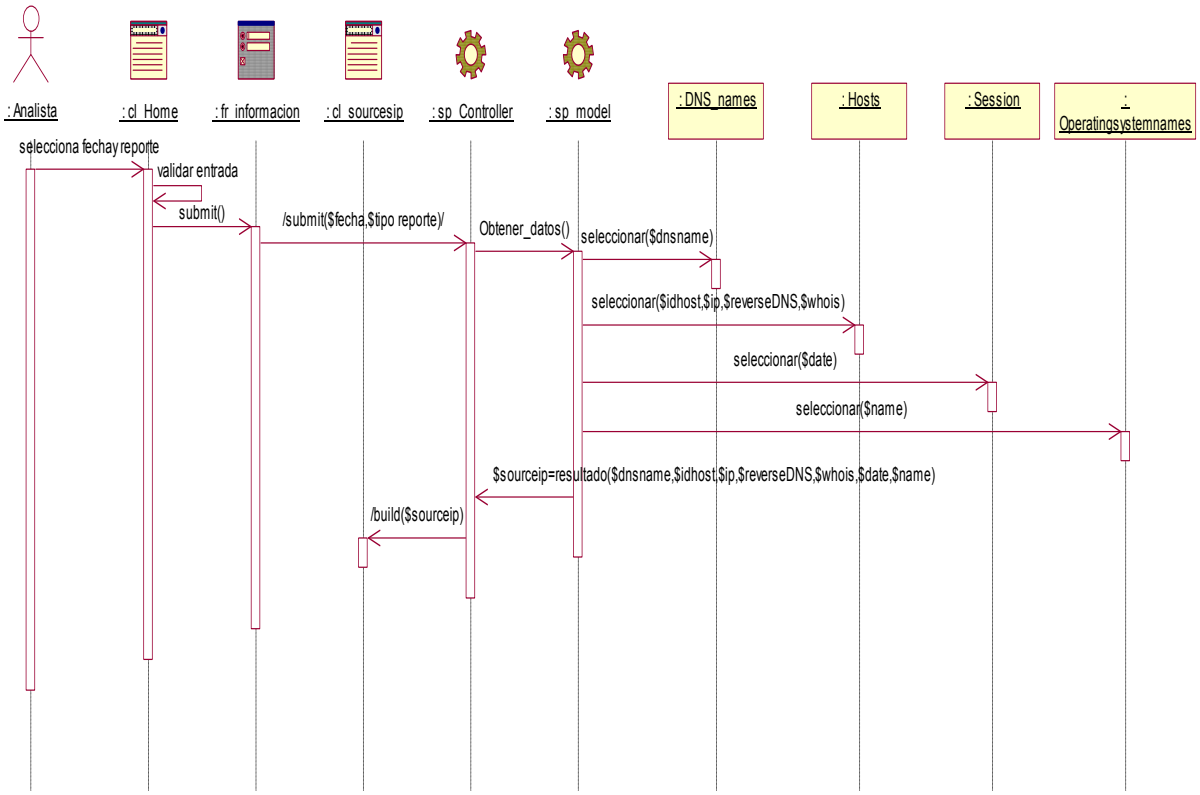


Diagrama de secuencia :CU Autenticar Usuario (Flujo normal)

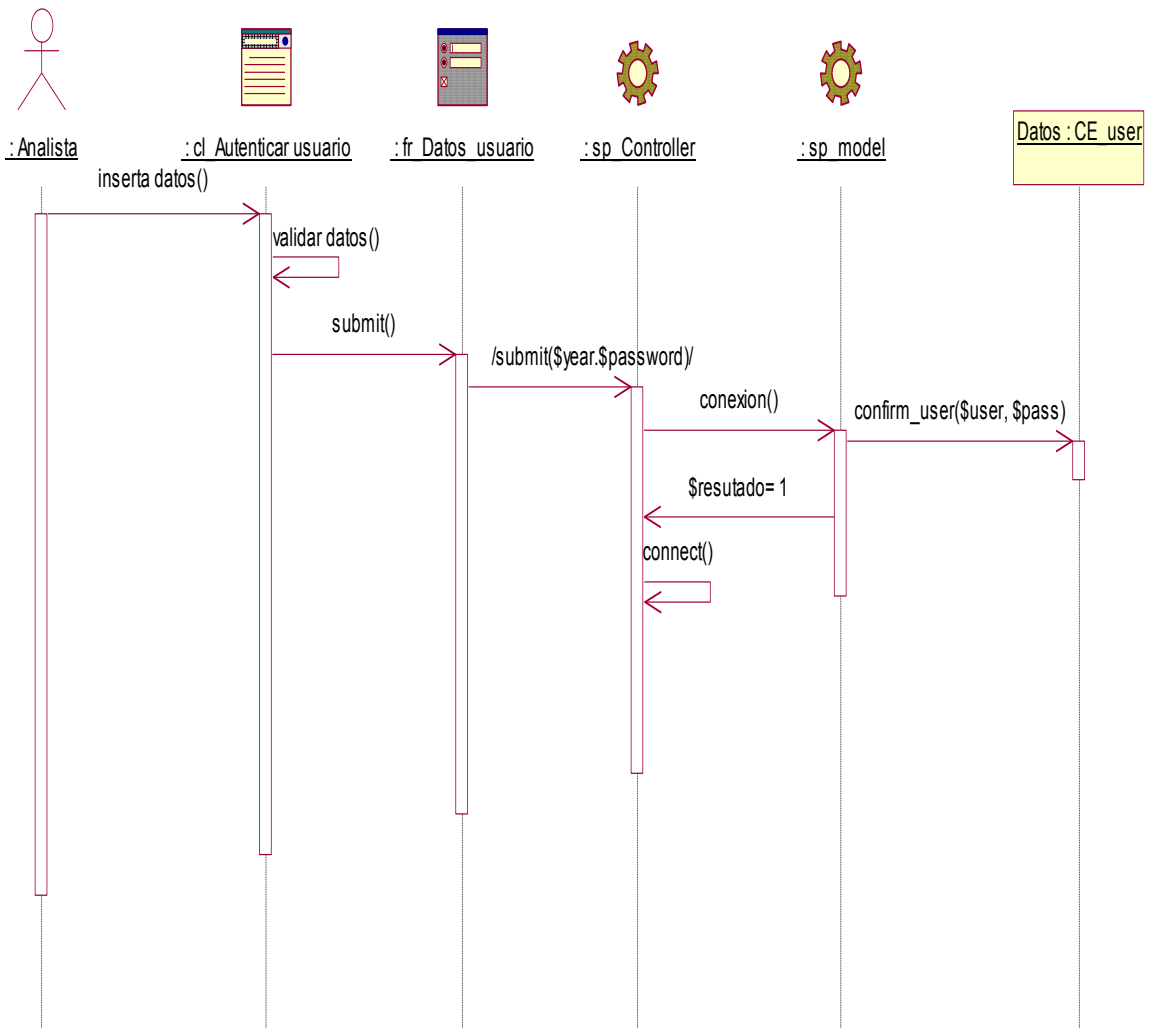


Diagrama de secuencia :CU Autenticar Usuario(Flujo alterno)

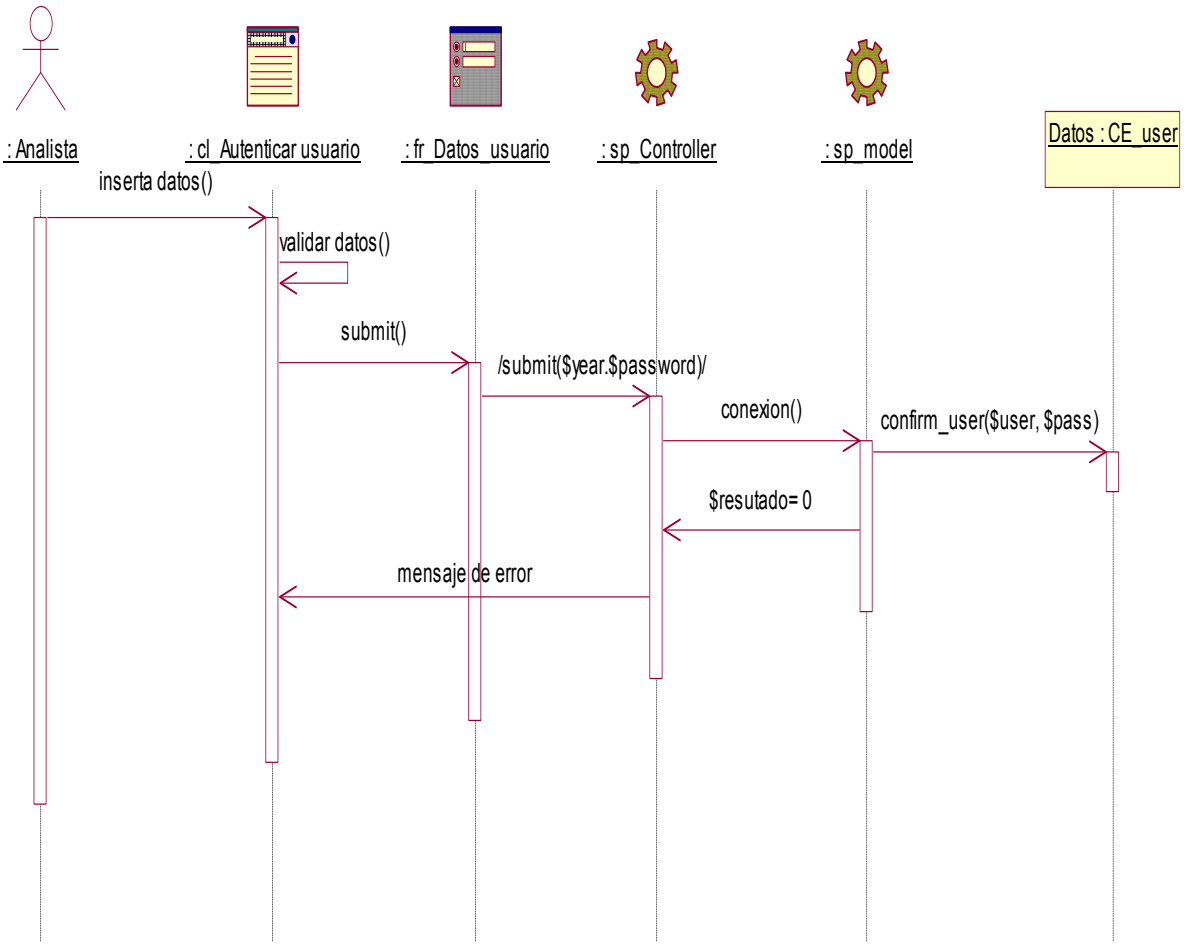


Diagrama de secuencia :CU Generar gráficas (Escenario Generar gráficas de las capturas)

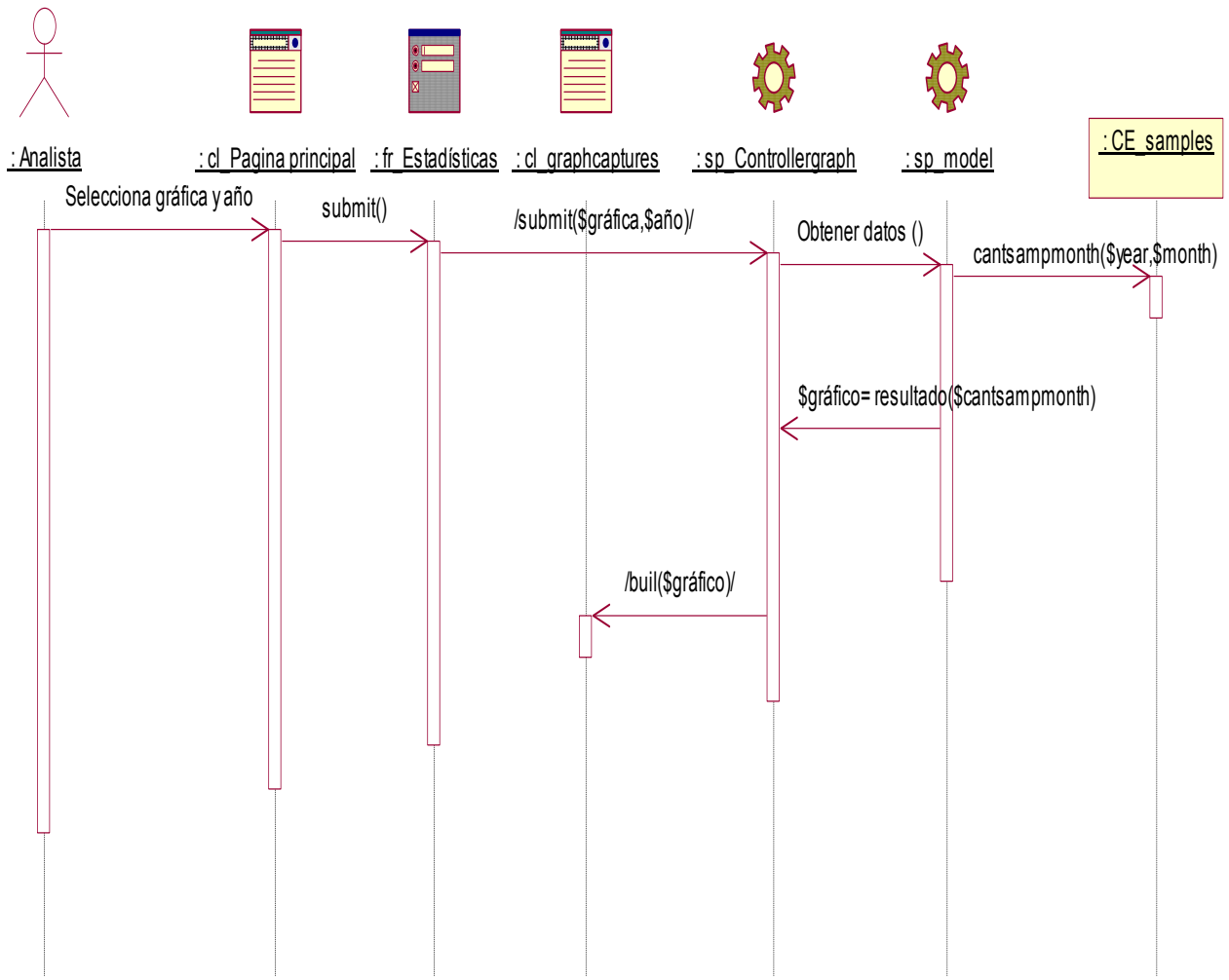


Diagrama de secuencia :CU Generar gráficas (Escenario Generar gráficas de las sesiones de ataques)

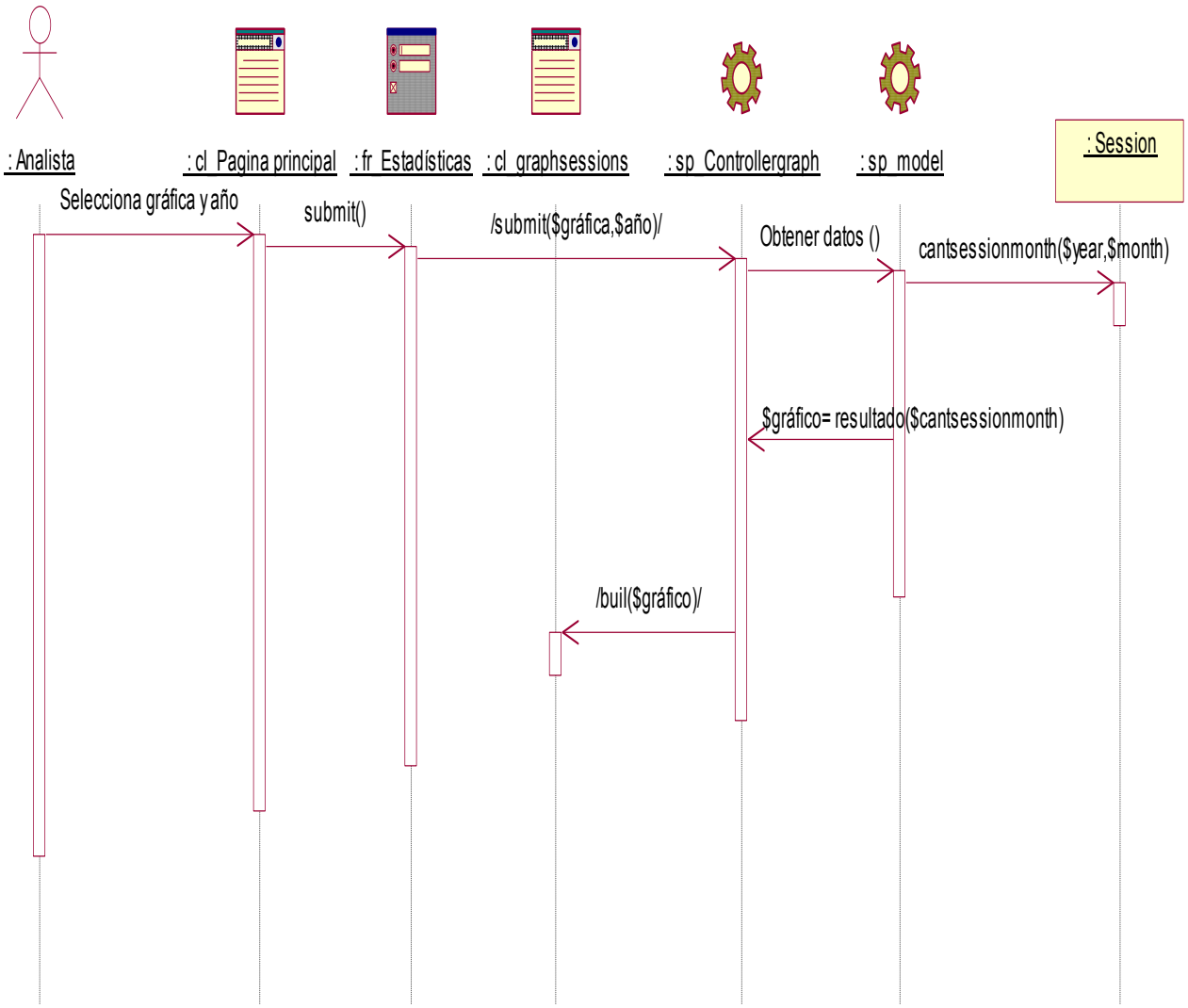


Diagrama de secuencia :CU Generar gráficas (Escenario Generar gráficas de los IP fuentes)

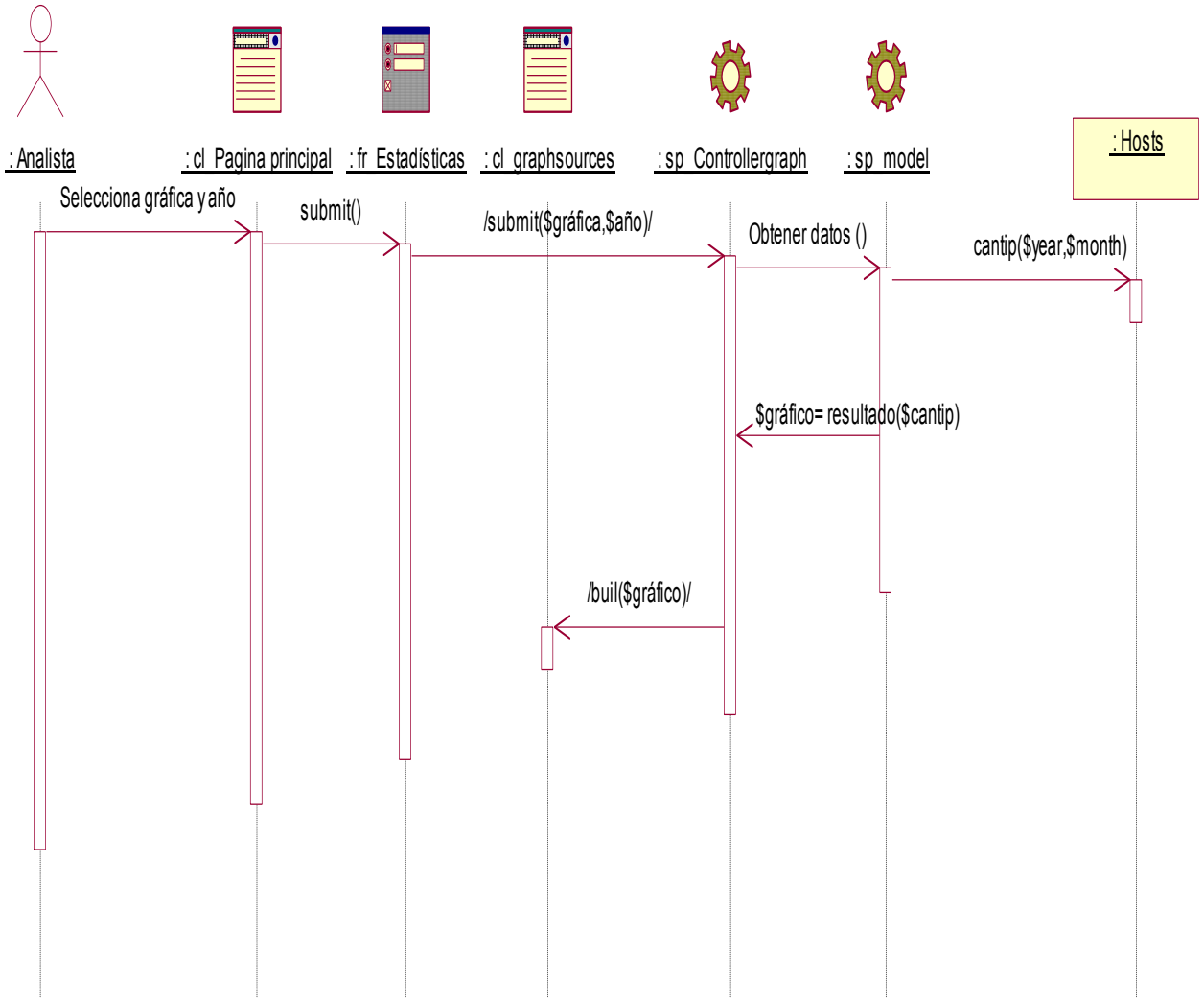


Diagrama de secuencia :CU Administrar usuarios (Escenario Eliminar usuario)

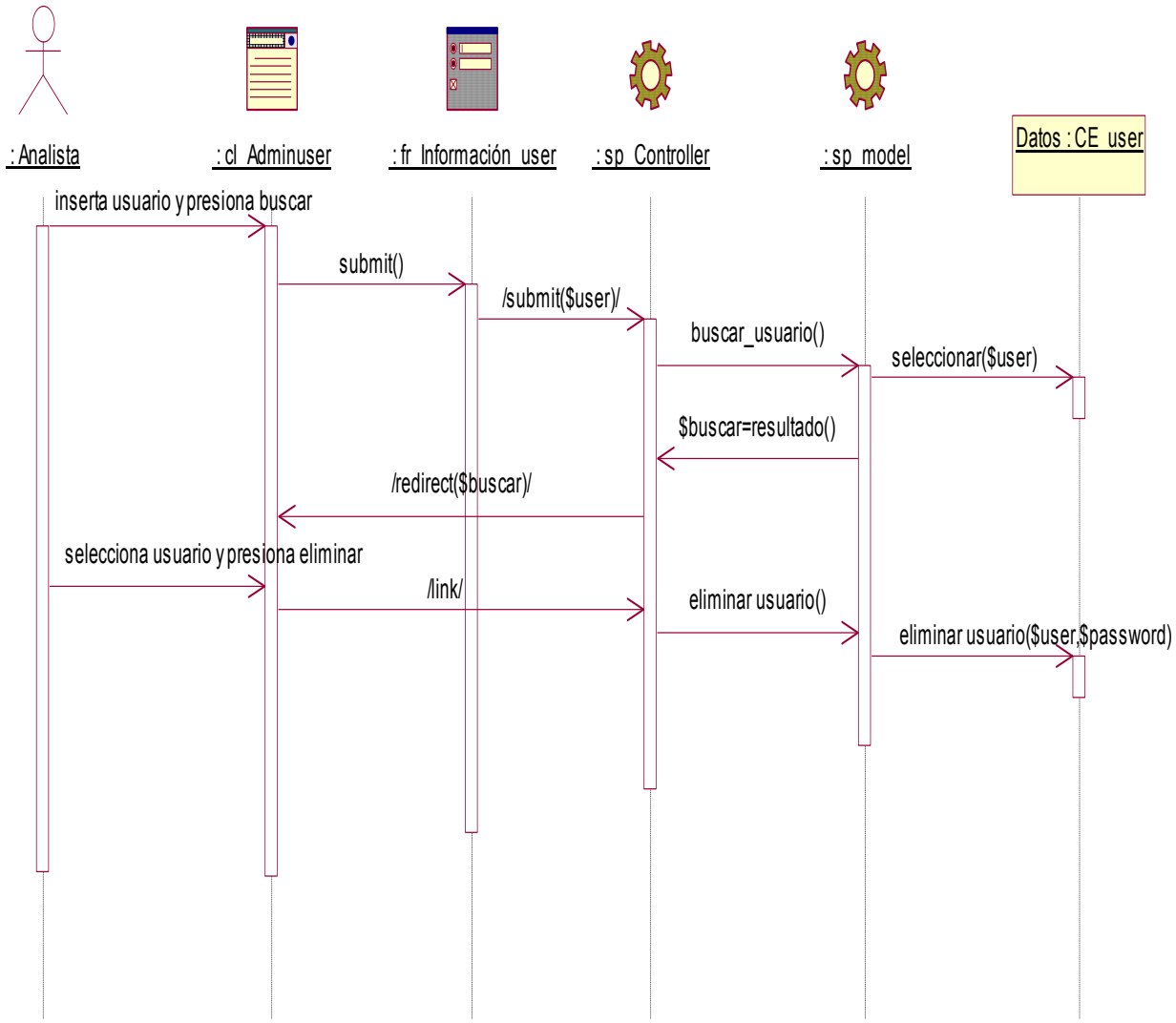


Diagrama de secuencia :CU Administrar usuarios (Escenario Adicionar usuario)

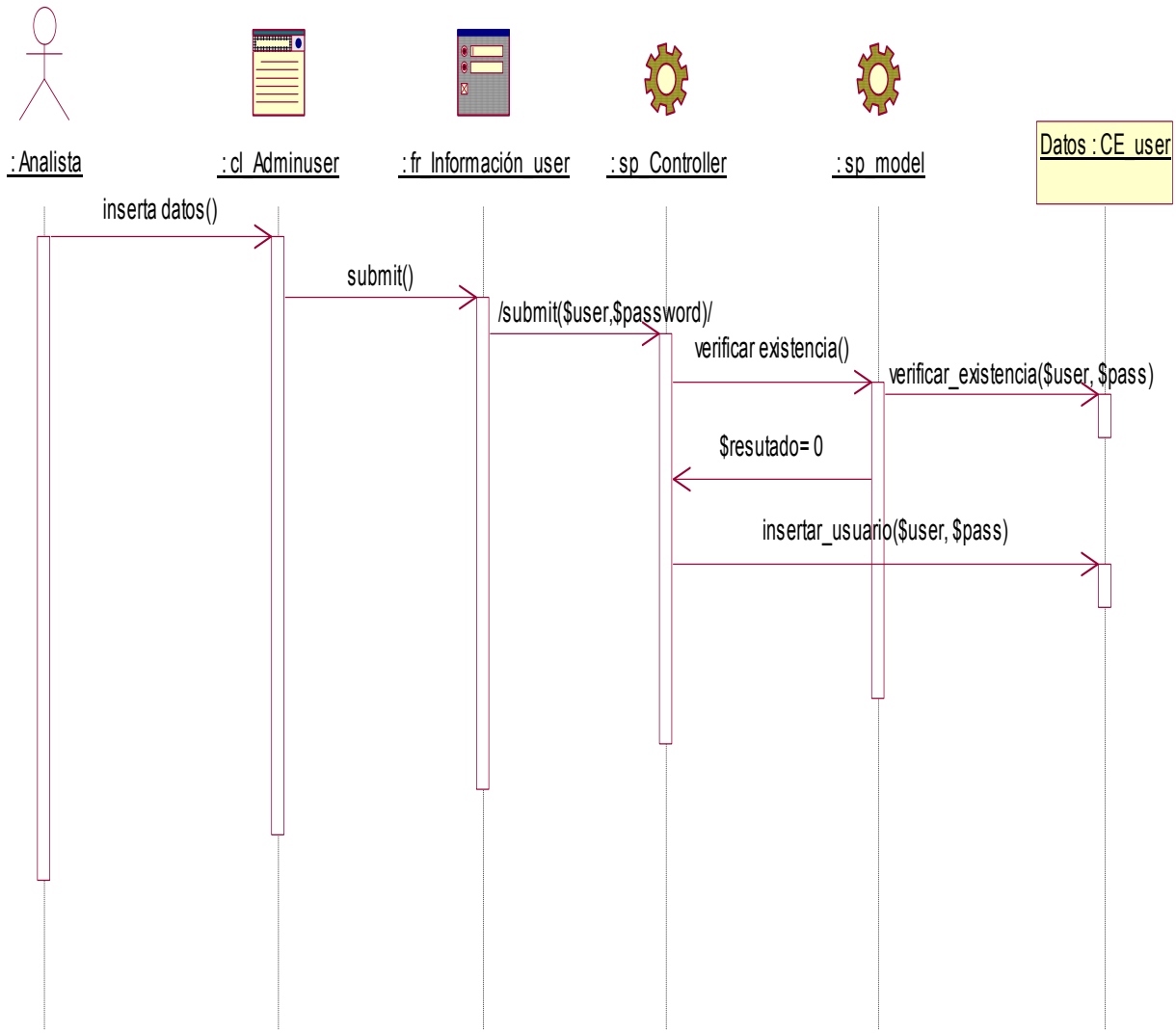
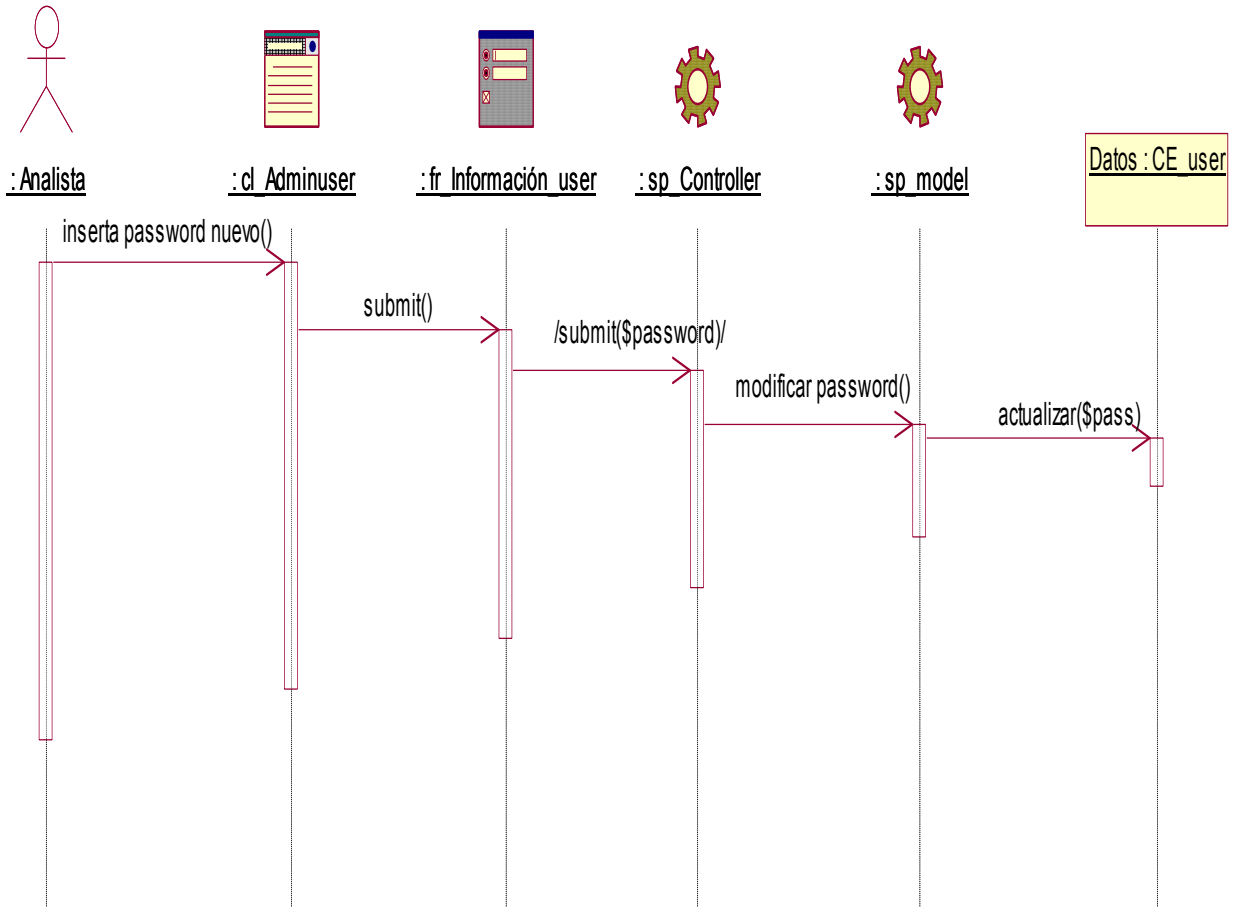


Diagrama de secuencia :CU Administrar usuarios (Escenario Cambiar contraseña del usuario)



Glosario de Términos

A

ASCII: (American Standard Code of Information Interchange). Es un largo código que define caracteres alfanuméricos, para compatibilizar procesadores de texto y programas de comunicaciones.

D

Dirección IP: La dirección IP es una serie de números asociadas a un dispositivo (una computadora), con la cuál es posible identificarlo ,dentro de una red configurada específicamente para utilizar este tipo de direcciones (una red configurada con el protocolo I.P. - Internet Protocol). Internet es un ejemplo de una red basada en protocolo IP.
Como Internet es una red basada en el protocolo IP, por lo tanto, toda computadora o dispositivo conectados a esta deben ser asociados a una dirección IP. Esta dirección identifica a ese dispositivo unívocamente y puede permanecer invariable en el tiempo o cambiar cada vez que se reconecte a la red. Una dirección IP es estática cuando no varía, y es dirección IP dinámica cuando cambia en cada reconexión.

F

Ficheros logs: Registro oficial de eventos durante un período de tiempo en particular, son ficheros que contienen mensajes sobre el sistema, servicios y las aplicaciones que se ejecutan en dicho sistema.

FTP: FTP son las siglas de File Transfer Protocol, es decir, Protocolo de Transferencia de Archivos. Es un sistema que permite enviar y recibir ficheros entre computadores a través de la red Internet. Con el fin de facilitar la creación de tu web, los servidores comerciales disponen de un sistema de FTP, mediante el que puedes enviar rápidamente y de una sola vez todos los ficheros que desees publicar en tu página u otros ficheros: imágenes, archivos de audio, etc.

H

Hacker: Una persona que disfruta aprendiendo de los detalles de programación y cómo extender sus capacidades, intensamente .Alguien que programa con entusiasmo, o que disfruta programando más que teorizando acerca de la programación. Una persona capaz de apreciar el valor de la tajada (*hack*). Un experto programando rápidamente. No todo lo que un *hacker* produce es una tajada. También un hacker es un entrometido inquilino que trata de descubrir información haciendo trampas."

Honeynet: Son un tipo especial de Honeypots de alta interacción que actúan sobre una red entera, diseñada para ser atacada y recobrar así mucha más información sobre posibles atacantes.

Honeypots: Software o conjunto de computadores cuya intención es atraer a crackers o spammers, simulando ser sistemas vulnerables o débiles a los ataques. Es una herramienta de seguridad informática utilizada para recoger información sobre los atacantes y sus técnicas.

HTML: Es el lenguaje con el que se definen las páginas web. Básicamente se trata de un conjunto de etiquetas que sirven para definir la forma en la que se presenta el texto y otros elementos de la página.

I

Interfaz: Dispositivo que permite la conexión de dos elementos para que puedan intercambiar información. En cuanto a la interfaz de usuario tiene que ver con la apariencia visual y modo de presentación de mensajes, así como con la forma de actuar de un programa o un sistema operativo.

J

Javascript: Javascript es un lenguaje de programación utilizado para crear pequeños programitas encargados de realizar acciones dentro del ámbito de una página web.

Se trata de un lenguaje de programación del lado del cliente, porque es el navegador el que soporta la carga de procesamiento. Gracias a su compatibilidad con la mayoría de los navegadores modernos, es el lenguaje de programación del lado del cliente más utilizado.

Con Javascript podemos crear efectos especiales en las páginas y definir interactividades con el usuario. El navegador del cliente es el encargado de interpretar las instrucciones Javascript y ejecutarlas para realizar estos efectos e interactividades, de modo que el mayor recurso, y tal vez el único, con que cuenta este lenguaje es el propio navegador.

L

Librerías GTK: Es un grupo importante de bibliotecas o rutinas para desarrollar interfaces gráficas de usuario.

M

MySQL: Es un sistema de gestión de base de datos relacional, multihilo y multiusuario, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan.

N

Nepenthes: Es una herramienta que simula vulnerabilidades conocidas para descargar el malware que intenta explotar estas vulnerabilidades.

T

TELNET: Telnet es una aplicación que permite desde nuestro sitio y con el teclado y la pantalla de nuestra computadora, conectarnos a otra remota a través de la red. Lo importante, es que la conexión puede establecerse tanto con una máquina multiusuario, que está en nuestra misma habitación o al otro lado del mundo. Una conexión mediante Telnet permite acceder a cualquiera de los servicios que la máquina remota ofrezca a sus terminales locales. De esta manera se puede abrir una sesión (entrar y ejecutar comandos) o acceder a otros servicios especiales: como por ejemplo consultar un catálogo de una biblioteca para buscar un libro, leer un periódico electrónico, buscar información sobre una persona, etc.

U

UML: Lenguaje Unificado de Modelado, utiliza un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos, y describe la semántica esencial de lo que estos diagramas y símbolos significan.

W

WWW:(World Wide Web) La WWW es en esencia, un servicio, que proporciona el Internet y sin embargo, muchos de nosotros constantemente pensamos que ambos son sinónimos.

X

XML: Lenguaje de Extensible de Marcado. No es un lenguaje sino un metalenguaje, esto es que, sirve para crear lenguajes. No se trata de una extensión ni un componente de HTML.