

Universidad de las Ciencias Informáticas
Facultad 4



Título: Sistema de control de procesos en los ordenadores

Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas

Autores: Yunek Pereira Salazar

Tutores: Ing. Luís Orlando Martín Álvarez

Junio de 2008

DECLARACIÓN DE AUTORÍA

Declaro que soy la única autora de este trabajo y autorizo a la Dirección de Redes y Seguridad Informática y a la Dirección de los laboratorios de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

Yunek Pereira Salazar

Ing. Luís Orlando Martín Álvarez

Firma del Autor

Firma del Tutor

DATOS DE CONTACTO

Ing. Luís Orlando Martín Álvarez graduado de Ingeniero en Ciencias Informáticas en el curso 2006-2007. Especialista de la Dirección de Redes y Seguridad Informática



AGRADECIMIENTOS

El que haya logrado alcanzar todas las metas que me propuse al culminar estos 5 años con la tesis se lo agradezco en primera instancia a mi mamá, gracias por haberme ayudado a más de 800 Kms. de distancia día a día.

A mi padre, hermano y toda mi familia, que de una forma u otra siempre se preocupó por mí y me mantuvo presente.

A Roberto Ferrer, Rafael Espinosa, Osmar Melo, Dionner Polanco, Bismark Castilla, Kiuver Ibáñez y Amary Jackson. por dedicarme un minuto de su apretado tiempo.

A Ariel Milian, Daniel Ríos e Ilen Ruiz por escucharme en mis momentos de desesperación.

A mi novio Angel Cabrera por sobrarle la paciencia para lidiar conmigo.

A cada persona que me preguntó cómo iba mi tesis, porque inconcientemente me recordaban la razón por la que estaba aquí y me daban fuerzas para seguir.

Gracias a todos y a mi misma

Yunek



DEDICATORIA

*A mis padres y a mi abuela Lidia
, mis principales razones de existir.*

Yunek Pereira Salazar.



RESUMEN

La Universidad de las Ciencias Informáticas ha acortado la brecha que existía entre nuestro país y el mundo referente al desarrollo de aplicaciones tanto Web como de escritorio de suficiente calidad como para ser utilizadas tanto en Cuba como en cualquier lugar del mundo, insertando a nuestro país en este ámbito en el cual todavía no ha alcanzado un completo desarrollo. Para ello ha vinculado miles de estudiantes y trabajadores a esta tarea productiva.

Como toda buena idea en su fase inicial, pues no pueden considerarse 5 años como un rango para definir que se posee una gran experiencia, presenta sus pros y contras, pues el poner en manos de estudiantes esta diversa cantidad de recursos se corre el riesgo de que no siempre se utilicen como se espera. En este caso específico se ha definido que además de la producción de software las computadoras se utilizan para jugar a gran escala, además de instalar keylogger y utilizarla para Proxy, lo cual no solo afecta la privacidad y el tiempo de uso que le puedan los estudiantes interesados a la PC, sino que también puede afectar la seguridad del país.

El sistema permite el control de los procesos que se ejecutan en el ordenador, para definir si se inicia alguno considerado prohibido. También obtiene una lista de procesos prohibidos desde una base de datos que detiene automáticamente. Permite además gestionar la lista de procesos prohibidos dadas las necesidades de los clientes y realiza diversos tipos de modificaciones que van desde el cambio de los parámetros de conexión a la base de datos hasta el cambio de la hora de accesos a la misma.

Aunque el entorno de desarrollo permite que el software sea multiplataforma solo se ha definido que funcione en Linux y aunque ha sido realizado para ser aplicado en la Universidad de las Ciencias Informáticas, por sus características se adaptaría a cualquier organización.



PALABRAS CLAVE

Procesos

Reportes

Directorios

Linux

Automático

Servicios

Ficheros



TABLA DE CONTENIDOS

INTRODUCCION.....	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	4
1.1. INTRODUCCIÓN	4
1.2. ESTADO DEL ARTE	6
1.2.1. <i>Synaptic</i>	6
1.2.2. <i>Definición de Proceso</i>	8
1.2.3. <i>Administrador de tareas</i>	8
1.2.4. <i>KDE</i>	8
1.2.5. <i>GNOME</i>	9
1.3. SOFTWARES INTERNACIONALES QUE SE ENCARGAN DEL CONTROL DE LOS PROCESOS	9
1.3.1. <i>Process Tamer</i>	9
1.3.2. <i>Process Control Task Manager 2.0</i>	9
1.3.3. <i>RunAlyzer</i>	10
1.3.4. <i>Faronics Anti-Executable</i>	10
1.3.5. <i>Kill Process</i>	11
1.4. HERRAMIENTAS PROPUESTAS PARA EL DESARROLLO DEL SISTEMA.....	11
1.4.1. <i>Lenguaje de programación: C++</i>	11
1.4.2. <i>Sistemas gestores de base de datos (SGBD)</i>	12
1.4.2.1. <i>Gestor de base de datos: MySQL</i>	12
1.4.3. <i>Lenguaje de Modelación: UML (Unified Modeling Language)</i>	13
1.4.4. <i>Entornos de desarrollo integrado</i>	13
1.4.4.1. <i>Eclipse</i>	14
1.4.4.2. <i>Gambas</i>	14
1.4.4.3. <i>Anjuta</i>	15
1.4.4.4. <i>MonoDevelop</i>	15
1.4.4.5. <i>KDevelop</i>	16
1.5. BIBLIOTECAS, LIBRERÍAS Y COMPILADORES	17
1.5.1. <i>Biblioteca Qt</i>	17
1.5.2. <i>GCC 4.2.0</i>	17
1.5.3. <i>Librería Time.h</i>	18
1.6. CONCLUSIONES	20
CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA.....	21
2.1. INTRODUCCIÓN.....	21
2.2. CARACTERÍSTICAS DEL MODELAMIENTO DEL NEGOCIO (MODELO DE DOMINIO).....	21
2.2.1. <i>Desarrollo del Dominio</i>	22
2.2.2. <i>Representación del modelo de Dominio</i>	24
2.3. PROPUESTA DEL SISTEMA DE CONTROL DE PROCESOS DE LOS ORDENADORES DE LA UCI.....	24
2.3.1. <i>Sistema de control de procesos en los ordenadores:</i>	25
2.3.2. <i>Sistema de manejo de reportes</i>	25



2.4. ESPECIFICACIÓN DE LOS REQUERIMIENTOS DEL SOFTWARE.....	25
2.4.1. <i>Requerimientos no funcionales</i>	25
2.4.1.1. Requerimientos de apariencia o interfaz externa:.....	26
2.4.1.2. Requerimiento de Usabilidad:.....	26
2.4.1.3. Requerimiento de rendimiento:.....	26
2.4.1.4. Requerimiento de Portabilidad:.....	26
2.4.1.5. Requerimiento de Soporte:	26
2.4.1.6. Requerimiento de Seguridad:.....	26
2.4.1.7. Requerimiento de Software:.....	27
2.4.1.8. Requerimiento de Hardware:	27
2.4.2. <i>Requerimientos funcionales</i>	27
2.5. DEFINICIÓN DE ACTOR DEL SISTEMA.....	28
2.6. CASOS DE USO DEL SISTEMA	28
2.7. DIAGRAMAS DE CASO DE USO DEL SISTEMA:.....	31
2.7.1. <i>Diagrama de caso de uso del Sistema de control de procesos</i>	31
2.7.2. <i>Diagrama de caso de uso del sistema de gestión de reportes</i>	32
2.8. CASOS DE USO EXPANDIDOS.....	32
2.8.1. <i>CU_Modificar_Conexion_Base_Datos</i>	32
2.8.2. <i>CU_Detener_Procesos</i>	33
2.8.3. <i>CU_Actualizar_Lista_Proceso</i>	34
2.8.4. <i>CU_Mostrar_Reportes</i>	36
2.8.5. <i>CU_Modificar_Intervalo</i>	38
2.8.6. <i>CU_Modificar_Hora_Acceso_Base_Datos</i>	39
2.9 CONCLUSIONES:.....	40
CAPÍTULO 3: ANÁLISIS Y DISEÑO.....	41
3.1. INTRODUCCIÓN.....	41
3.2. DIAGRAMAS DE CLASES DEL ANÁLISIS DEL SISTEMA DE CONTROL DE PROCESOS EN LOS ORDENADORES	42
3.3. DIAGRAMA DE CLASES DEL SISTEMA DE MANEJO DE REPORTES.....	45
3.4. DIAGRAMAS DE SECUENCIA	45
3.5. DIAGRAMAS DE CLASES DE DISEÑO	51
3.5.1. <i>Estilo arquitectónico</i>	51
3.5.2. <i>Diagrama de clases de diseño del sistema de control de procesos</i>	52
<i>CU_ACTUALIZAR_LISTA_PROCESO</i>	52
3.5.2.1. Descripción de las clases de diseño del CU_Actualizar_Lista_Proceso	53
3.5.2.2. Descripción de las clases de diseño del CU_Modificar_Conrxion_Base_Datos	59
3.5.2.3. Descripción de las clases de diseño del CU_Modificar_Timer.....	61
3.5.2.4 Descripción de las clases de diseño del CU_Detener_Procesos_Automaticamente	63
3.6. DIAGRAMA DE CLASES DE DISEÑO DEL SISTEMA DE MANEJO DE REPORTES.....	64
3.6.1. <i>Descripción de las clases de diseño del sistema de gestión de reportes</i>	65



3.7. DISEÑO DE LA BASE DE DATOS.....	68
3.8.1. <i>Descripción de las tablas de la Base de Datos</i>	69
3.9. CONCLUSIONES	70
CAPÍTULO 4: IMPLEMETACION Y PRUEBAS DEL SISTEMA.....	72
4.1. INTRODUCCIÓN.	72
4.2. IMPLEMENTACIÓN	72
4.2.1. <i>Diagrama de despliegue</i>	72
4.2.2. <i>Diagrama de componentes</i>	72
4.2.2.1. <i>Diagrama de componentes del sistema de control de procesos</i>	73
4.2.2.2. <i>Diagrama de componentes del sistema de gestión de reportes</i>	73
4.3. MODELO DE PRUEBAS.....	74
4.3.1. <i>Casos de prueba del sistema de Control de procesos</i>	74
4.3.2. <i>Casos de prueba del sistema de gestion de reportes</i>	79
4.4. CONCLUSIONES	80
CONCLUSIONES.....	81
RECOMENDACIONES.....	82
BIBLIOGRAFIA.....	83
BIBLIOGRAFIA REFERENCIADA.....	84
ANEXOS	86
GLOSARIO DE TERMINOS.....	88



INTRODUCCION

En la Universidad de las Ciencias Informáticas se localiza una de las principales redes del país, en ella existen estudiantes y profesores vinculados a la producción de software en proyectos productivos cuya principal herramienta de trabajo son las computadoras. También existen laboratorios asignados a la docencia en el cual los estudiantes utilizan los ordenadores para investigaciones con el objetivo de realizar actividades docentes. Pese a todas estas facilidades algunos estudiantes no utilizan su tiempo en los laboratorios de forma óptima y utilizan las computadoras para hacer uso de juegos en redes, ver series y películas, mientras otros no pueden utilizar este recurso; además de la afectaciones a la eficiencia de transmisión de la red. Otras de las violaciones del código de ética que se cometen es la instalación keylogger en las computadoras de los laboratorios para la adquisición indebida de contraseñas, así como la configuración de proxy, actividad que rompen con algunas restricciones y logra de cierta forma el acceso a Internet. Estas violaciones atentan contra la privacidad de servicios que les brinda la universidad a los estudiantes ya sean de correo o mensajería e incluso su navegación por Internet debido al irregular consumo de su cuota de navegación.

Ante estas circunstancias existe la siguiente **situación problémica**. ¿Cómo lograr disminuir el robo de contraseñas y un mejor aprovechamiento de los recursos de las computadoras en los laboratorios de la Universidad de las Ciencias Informáticas? Analizando la anterior interrogante se llegó a concluir la siguiente **hipótesis** que defiende este trabajo:

- Si se desarrolla un sistema que controle los procesos en las computadoras de los laboratorios de la Universidad de las Ciencias Informáticas entonces se logrará disminuir el robo de contraseñas y condicionar a un mejor aprovechamiento de los recursos que brindan las mismas.

Definida la anterior hipótesis de alguna forma entonces podemos deducir el **objetivo general** de este trabajo que se expresa a continuación:

- Desarrollar un servicio de control de procesos de las computadoras en la Universidad de las Ciencias Informáticas que contribuya a eliminar el robo de contraseñas y aprovechar mejor los recursos de las mismas.
- Además de haber propuesto como **Objetivos Específicos**:



- Desarrollar una solución de software libre y código abierto aplicable en distribuciones de Linux.
- Mostrar a los administradores reportes estadísticos relacionados con los procesos que han sido controlados.

Para poder desarrollar el sistema en cuestión se tuvieron en cuenta un grupo de investigaciones que permitieron cumplir con cabalidad la creación del mismo. Las mismas se mencionan a continuación:

- Analizar el funcionamiento de otros softwares controladores de procesos para tomar experiencias del funcionamiento de los mismos.
- Estudiar el funcionamiento y estructura de los procesos en distribuciones de Linux.
- Investigar lenguajes de programación idóneos para la implementación del software valorando el sistema operativo sobre el que se desarrollará.
- Investigar librerías que controlen el ciclo de vida de los procesos.
- Investigar y definir la arquitectura idónea para desarrollo del sistema.
- Profundizar en el diseño de un modelo de datos relacional eficiente para un alto volumen de información provocado por todas las operación de control sobre los procesos en todas las computadoras donde esta funcionando el sistema.
- Investigar lenguajes idóneos para la implementación del software vinculado con el sistema operativo que voy a utilizar.

Auxiliados por estas tareas, definidas es pos de alcanzar un resultado satisfactorio en este trabajo, se pudo estructurar el mismo como a continuación se vera:

Capitulo I: En este capitulo se realiza un estudio sobre diversos softwares con características similares al que se va a desarrollar, además de definir las herramientas que se van a utilizar para implementar el software.

Capitulo II: En este capitulo se define el negocio mediante un estudio de los procesos que se realizan en el mismo, además de presentar una propuesta del sistema que se va a encargar de automatizar este negocio.

Capitulo III. Se representan los enfoques que en cuestión de análisis y diseño se van a tener en cuenta esbozados a través de diversos diagramas.



Capitulo IV: Se exponen las principales características del flujo de implementación y se realizan casos de prueba para definir los posibles fallos.



CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1. Introducción

La Informática es una disciplina que surge producto de la aplicación-interacción sinérgica de varias ciencias, como la computación, la electrónica, la cibernética, las telecomunicaciones, la matemática, la lógica, la lingüística, la ingeniería, la inteligencia artificial, la robótica, la biología, la psicología, las ciencias de la información, etc... La cultura alcanzada mundialmente cada día va en aumento, las personas utilizan las redes para la comunicación interpersonal mediante un medio que ha sufrido desde su origen muchas mejoras: las computadoras.

Las computadoras están compuestas por hardware y software y trabajan mediante procesos que facilitan su funcionamiento y estos a su vez están controlados por sistemas operativos que mejoran el encuentro usuario –maquina.

¿Que es un sistema operativo (SO)? entre sus acepciones:

Un sistema operativo es un conjunto de programas y utilidades básicas que hacen que su computadora funcione. El centro de un sistema operativo es el núcleo (kernel). El núcleo es el programa más importante en la computadora, realiza todo el trabajo básico y le permite ejecutar otros programas.

Como ejemplos de estos tenemos el renombrado Windows, que es un sistema operativo amigable usado por millones de personas con la característica negativa de que su uso está restringido por licencias que no permiten que ningún usuario tenga derecho a cambiar su código fuente, como contrapartida a este SO ha surgido otro que gracias a su peculiaridad de acceso al código fuente ha venido a cambiar el rumbo de la informática para los usuarios: Linux, sistema operativo con código abierto siempre dispuesto a cambios cuyas principales ventajas con respecto a Windows se verán a continuación mediante una breve comparación.

La instalación:

En Linux a pesar de todos los esfuerzos la instalación no resulta sencilla siempre, pero te permite personalizar totalmente los paquetes que quieras instalar.

En Windows la instalación es mínimamente configurarle aunque es muy sencilla.



La compatibilidad: Ninguno de los dos sistemas operativos son totalmente compatibles con el Hardware, a pesar de que Windows se acerca más, los dos están cerca de conseguirlo.

Aunque Linux no esta detrás de ninguna casa comercial gracias a su elevada popularidad ofrece una alta compatibilidad ofreciendo, además, actualizaciones frecuentes.

Windows al ser parte de Microsoft intenta ofrecer una gran cantidad de drivers ya que su gran poder económico hace que las empresas mismas de hardware creen sus propios drivers

Software:

Linux al tener menos software en algunos campos sufre una menor aceptación por parte de las empresas, aunque gracias a los apoyos de empresas como Sun Microsystems o IBM se ha logrado muchos avances.

Windows al ser el más fácil de usar en las empresas, posee una gran cantidad de software.

Robustez:

Linux se ha caracterizado siempre por la robustez de su sistema ya que pueden pasar meses e incluso años sin la necesidad de apagar o reiniciar el equipo, también si una aplicación falla simplemente no bloquea totalmente al equipo.

En Windows siempre hay que reiniciar cuando se cambia la configuración del sistema, se bloquea fácilmente cuando ejecuta operaciones aparentemente simples por lo que hay que reiniciar el equipo. A estas características se les puede sumar:

- Es software libre, lo que quiere decir que no hay que pagar nada por el sistema en sí.
- Es un sistema operativo muy fiable ya que hereda la robustez de UNIX.
- Ideal para las redes ya que fue diseñado en Internet y para Internet
- No es cierto que tenga pocos programas, solo en algún campo muy específico.
- Es 100% configurable.



- Es el sistema más seguro, ya que al disponer del código fuente cualquiera puede darse cuenta de algún fallo, se puede decir que decenas de miles de personas velan por tu seguridad.
- Existe muchísima documentación, también en español gracias a los proyectos como LUCAS.
- Cuenta con el soporte de muchas grandes empresas como IBM, Corel, Lotus, Siemens, Motorola, Sun, etc.
- Puedes encontrar ayuda en millones de sitios en Internet como los foros.
- Es muy portable, si tienes un Mac un Alpha o un SPARC puedes usar Linux sin problemas(1)

La Universidad de las Ciencias Informáticas, surgida al calor de la batalla de ideas ha venido a formar parte de este mundo de procesamiento de datos, sistematizaciones y automatizaciones y se ha insertado en la creación de software para uso mundial y debido a cuestiones de diversas aristas ,por ejemplo la política , el sistema operativo Linux facilita el trabajo en este tipo de producción , pero existen muchas maneras de malgastar el tiempo y por eso se están estudiando formas de que se optimicen el funcionamiento de los ordenadores verificando esencialmente los procesos que se están manejando en los diversos clientes y manteniendo solo los necesarios para un uso adecuado de los mismos mediante un sistema que controlará los procesos que se inician en el ordenador ,por ejemplo los juegos. En el presente capítulo se muestran los resultados de un estudio de diversos softwares encargados del control y gestión de los procesos en el mundo, además de una panorámica sobre las herramientas con las que se va a desarrollar este software.

1.2. Estado del arte

Linux es un sistema operativo cuya principal novedad es su autonomía y facilidad de acción, en él todos los softwares con los que se desea trabajar están por separados y vienen empaquetados, para acceder a los mismos es necesario al instalarlo, actualizar su lista de repositorios mediante un paquete principal conocido como Synaptic, cada repositorio tiene paquetes definidos según sus funcionalidades por lo cual cuando se necesita realizar una aplicación se debe actualizar los disponibles.

1.2.1. Synaptic



Synaptic es un programa informático que es una interfaz gráfica GTK+ de APT, para el sistema de gestión de paquetes de Debian GNU/Linux. Generalmente se utiliza Synaptic para sistemas basados en paquetes .deb pero también puede ser usado en sistemas basados en paquetes RPM.

Synaptic utiliza repositorios Debian, y permite su gestión mediante un menú interactivo. Mediante un clic, es posible actualizar la lista, de paquetes disponibles o marcar automáticamente todas las actualizaciones de los paquetes ya instalados.

Dispone de un avanzado filtro de búsqueda, es capaz de reparar las dependencias rotas de paquetes y permite deshacer y rehacer las últimas selecciones de paquetes.

Su interfaz gráfica, además, es personalizable mediante el menú de configuración

En Linux ya se han desarrollado diversas distribuciones entre la cuales se encuentran, Debian, Ubuntu, RedHat e incluso en nuestra universidad se esta desarrollando la distribución Nova Linux. En Linux también existe una estrecha vinculación entre el SO y sus comandos pues pese a existir el Synaptic también se puede instalar un programa ejecutando simplemente el comando `apt-get install /nombre del paquete`, lo que facilita mucho el trabajo en el mismo. Este es sólo un comando de ejemplo.

Aunque todavía no ha alcanzado muchas de las funcionalidades que posee Windows desde el punto de vista de la programación, pues tiene muchas funciones definidas que todavía están por implementarse, es un sistema muy flexible del cual se puede obtener mucha información a través de los ficheros y directorios, que en nuestro caso fueron especialmente útiles para desarrollar el sistema, pues para controlar los procesos tuvo que acceder a detalles específicos del sistema operativo por la escasez de funciones que a veces presenta.

En este caso se utilizará Debian GNU/Linux o simplemente Debian. Los sistemas Debian utilizan hasta el momento el núcleo de Linux pero se debe aclarar que existe otros núcleos, tal es el caso de Hurd. Debian viene con más de 15490 paquetes (software precompilado y empaquetado en un formato amigable para una instalación sencilla en su máquina) — todos ellos de forma gratuita.

En la actualidad diversas personas se han detenido a pensar como detener los procesos que están ejecutándose en una PC o ajustarlos a las necesidades de las propias características del entorno donde se desarrolla tanto formando parte de una red como una PC personal específica.

Ahora cabe la pregunta: ¿Qué se define como proceso?



1.2.2. Definición de Proceso

Un proceso no es más que un conjunto de threads (hilos) que ejecutan el mismo código, junto con las zonas de memoria asociadas a ellos y los ficheros que tienen abiertos.

Un programa consta, al menos, de un proceso, y un proceso, al menos, de un thread. Cuando un programa tiene varios procesos, lo normal es que cada uno ejecute un código distinto, los cuales se encuentran en ficheros ejecutables separados. Dos procesos solo pueden compartir una zona de memoria si esta es definida expresamente como tal.(3) Además tiene sus propias características como estado, prioridad y nombre con el cual los reconoce el CPU.

Ante todo puntualizar que el software controlador de procesos por excelencia es el Administrador de Tareas.

1.2.3. Administrador de tareas

Un administrador de tareas es un programa informático que se utiliza para proporcionar información sobre los procesos y programas que se están ejecutando en una computadora y su situación general. Puede emplearse para finalizar procesos, comprobar el uso de CPU de éstos, así como terminar programas y cambiar la prioridad entre procesos. Posee diversas acepciones según el sistema operativo que se enunciaran a continuación:

- Administrador de tareas de Microsoft Windows
- GKrellM en BSD, Linux, Solaris y Mac OS X.
- KDE System Guard y KTop en KDE
- GNOME System Monitor en GNOME.

1.2.4. KDE

KDE (*K Desktop Environment* o *Entorno de Escritorio K*, es un entorno de escritorio e infraestructura de desarrollo para sistemas Unix/Linux. En otras palabras se puede decir que KDE es un entorno de escritorio contemporáneo para estaciones de trabajo Unix. KDE llena la necesidad de un escritorio amigable para estas estaciones de trabajo, similar a los escritorios de MacOS o Windows. La «K», originariamente, representaba la palabra «Kool», pero su significado fue abandonado más tarde.



Actualmente significa simplemente «K», la letra inmediatamente anterior a la «L» (inicial de Linux) en el alfabeto.(5)

1.2.5. GNOME

GNOME: posee dos características fundamentales, la primera: es un entorno de escritorio para sistemas operativos de tipo Unix bajo tecnología X Windows. Forma parte oficial del proyecto GNU. Nació como una alternativa a KDE.GNOME también es una plataforma de desarrollo compuesta por diversas librerías y bibliotecas como la Gtk.(6)

En el capítulo en cuestión se hará mención de algunos de los software más usados en la actualidad para el control y gestión de los procesos y servicios que se ejecutan en las computadoras después de haber dejado claro que el administrador de tareas no se puede utilizar por que nuestra idea es que el sistema funcione como un servicio que ningún usuario sin privilegio de superusuario pueda detener. Además que tenga como principal característica ser automático, es decir, que para detener los procesos no haya que acceder directamente a él.

1.3. Softwares internacionales que se encargan del control de los procesos

1.3.1. Process Tamer

Hay ocasiones en las que un solo proceso activo de tu sistema consume, literalmente, todos los recursos de la computadora, como por ejemplo cuando realizas conversiones de formato, editas vídeo, grabas CD o cualquier otra tarea que requiera una gran carga de procesador. Este pequeño programa se encarga de controlar esas situaciones, monitorizando el uso que hace de los recursos cada uno de los procesos activos, y cambiando temporalmente la prioridad de aquellos que sobrecarguen en exceso el sistema. Su funcionamiento está totalmente automatizado, poniéndose en marcha sólo cuando sea necesario. Además, puedes activarlo y desactivarlo fácilmente desde un icono en la bandeja de sistema. (7; 8)

De esta aplicación se puede tomar como útil y factible su característica de monitorizar automáticamente los procesos, pues es lo mismo que se persigue con el software a desarrollar, pero no es totalmente adaptable a nuestras condiciones debido a que no se están buscando detener procesos por el gasto que hagan del CPU.

1.3.2. Process Control Task Manager 2.0



Descripción: Controla y decide los procesos ocultos que se efectúan en tu PC. Process Control Task Manager te permite ver todos los procesos que están actualmente en funcionamiento en tu PC, incluyendo los procesos invisibles. Te permite finalizar un proceso o simplemente cambiar la prioridad de ejecución. Los procesos se muestran bajo nombres comprensibles, completos con el nombre del programa y ubicación exacta. Limitaciones de la versión shareware: Caduca a los 30 días de uso.

1.3.3. RunAlyzer

RunAlyzer es un excelente gestor de procesos para Windows que nos permite controlar los servicios que se inician con el sistema operativo. Muestra una exhaustiva información sobre los servicios iniciados, las tareas programadas, los añadidos para Internet Explorer, los procesos en uso y la lista de programas autoarrancables. Es compatible con Windows Vista tanto en las versiones de 32 y 64 bits. Además es gratuito (9)

Este software, pese a controlar los procesos y servicios que se inician en el sistema no es eficiente para servir de solución al problema que se nos plantea principalmente porque no puede ser utilizado en Linux, que el SO para el cual se está diseñando esta solución, además de que los procesos y servicios se controlan manualmente y lo que se espera es que la aplicación controle los procesos prohibidos sola.

1.3.4. Faronics Anti-Executable

Te ayudará a mejorar el control que tienes sobre tu sistema, restringiendo la capacidad de los usuarios de ejecutar o instalar nuevos programas. Al instalarlo, el programa realiza una revisión profunda y autoriza cualquier ejecutable que ya se encuentre instalado. A partir de ese punto, cualquier otro fichero ejecutable es señalado como desautorizado y no es posible ejecutarlo o instalarlo. Al prevenir la ejecución o instalación de nuevos programas, Faronics Anti-Executable funciona muy bien contra programas espías y de código maliciosos en general.

Características:

Previene la ejecución o instalación de nuevos programas.

Crea automáticamente una lista blanca (programas autorizados) al instalarse.

Detecta más de 80 tipos diferentes de ficheros ejecutables.



Acceso total a todas las funciones del ordenador.

Reportes detallados de eventos.

Restringe la copia, borrado o renombramiento de ficheros ejecutables.

Protección por contraseña.

Interfaz amigable y de fácil uso(10)

No es necesario abundar mucho para definir la principal desventaja que presenta este software. Simplemente suprime la instalación de cualquier posterior a su instalación lo cual no es desde ningún punto de vista lo que se persigue.

1.3.5. Kill Process

Práctica herramienta que muestra todos los procesos en ejecución de tu sesión para que puedas terminar cualquiera de ellos al instante.

Kill Process indica también el ID (identificador) de cada proceso y, gracias a su sistema de identificación, la aplicación a la que pertenece cada uno. Kill Process puede ser utilizado como un buen complemento del Administrador de tareas del sistema, puesto que la función de identificación anteriormente descrita no se encuentra en dicha utilidad nativa de Windows.(11)

Kill Process, es simplemente un complemento del administrador de tareas por lo que esta claro que no es adaptable a la idea que se quiere desarrollar. Se espera detener procesos con esta aplicación pero no manualmente sino automáticamente.

1.4 Herramientas propuestas para el desarrollo del sistema

Con el objetivo de desarrollar un sistema óptimo se llevo a cabo un exhaustivo estudio de las diversas metodologías, tendencias y tecnologías que se utilizan en la actualidad para desarrollar aplicaciones de escritorio en el mundo que se ejecuten en el Sistema Operativo Linux, definiéndose las siguientes propuestas.

1.4.1 Lenguaje de programación: C++

C++ es un lenguaje que abarca tres paradigmas de la programación: la programación estructurada, la programación genérica y la programación orientada a objetos. Las principales características del



C++ son las facilidades que proporciona para la programación orientada a objetos y para el uso de plantillas o programación genérica (templates). Además posee una serie de propiedades difíciles de encontrar en otros lenguajes de alto nivel.

C++ está considerado por muchos como el lenguaje más potente, debido a que permite trabajar tanto a alto como a bajo nivel. También se caracteriza por ser uno de los pocos lenguajes que no necesita una máquina virtual para leer su código y ser compilado por lo que se considera un lenguaje optimizado. Linux tiene programada una parte fundamental de su código en este lenguaje por lo que interactúa con su Kernel, facilitando la programación.

1.4.2 Sistemas gestores de base de datos (SGBD)

Un Sistema Gestor de base de datos (SGBD) es un conjunto de programas que permiten crear y mantener una Base de datos, asegurando su integridad, confidencialidad y seguridad. Por tanto debe permitir:

- Definir una base de datos: especificar tipos, estructuras y restricciones de datos.
- Construir la base de datos: guardar los datos en algún medio controlado por el mismo SGBD
- Manipular la base de datos: realizar consultas, actualizarla, generar informes. (12)

1.4.2.1 Gestor de base de datos: MySQL

El principal objetivo que se tuvo en cuenta para su diseño fue la velocidad., por lo que sacrificó algunas características que eran esenciales para sistemas llamados más “serios”. Otra funcionalidad que lo vuelve útil es que consume muy pocos recursos, tanto de memoria como de CPU.

Ventajas:

1. Mayor rendimiento: Mayor velocidad tanto al conectar con el servidor como al servir diversas consultas
2. Mejores utilidades de administración (backup, recuperación de errores, etc.).
3. Aunque en ocasiones suele colgarse con grandes volúmenes de información, no suele perderla ni corromper los datos.
4. No hay límites en el tamaño de los registros.



5. Mejor control de acceso, en el sentido de qué usuarios tienen acceso a qué tablas y con qué permisos.
6. MySQL se comporta mejor que otros gestores de Base de datos, como por ejemplo, Postgres, a la hora de modificar o añadir campos a una tabla "en caliente". (13)

1.4.3. Lenguaje de Modelación: UML (Unified Modeling Language).

UML ("Unified Modeling Language") está definido como el lenguaje estándar en el análisis y diseño de sistemas automatizados. Gracias a UML es posible establecer las principales características que va a tener nuestro sistema, en especial los requerimientos y estructuras a los que se les va a dar solución en la organización antes de adentrarse en el código con el que se le dará vida y funcionalidad al mismo, por tener la característica de detectar las dependencias y dificultades implícitas del sistema y permitir que los cambios realizados en una etapa inicial, como por el ejemplo el análisis, sean más fáciles de realizar que cuando se entra en la etapa de codificación.

En otras palabras, con UML se realizan los diseños previos a la codificación del sistema ya que este se encarga de especificar, visualizar y documentar los componentes de un sistema siempre y cuando posea desarrollo orientado a objetos. Pese a que UML es un lenguaje este posee más características visuales que características de programación, las cuales facilitan a los integrantes del equipo mantener una constante comunicación. Las herramientas CASE, tales como el Visual Paradigm y el Rational Rose lo han adoptado como lenguaje de modelado. En este trabajo se utilizará el Visual Paradigm por ser una herramienta desarrollada en software libre.

RUP es una metodología que se utiliza para el análisis, diseño, implementación y documentación de los sistemas realizados cuya característica fundamental es que son orientados a objetos. Es una metodología que se adapta al entorno, es decir sus utilidades se moldean de acuerdo al contexto y la necesidad de cada organización en específico. UML y RUP están estrechamente relacionados, ya que el primero define los diagramas y el segundo los pasos.

1.4.4. Entornos de desarrollo integrado

Un **entorno de desarrollo integrado** o en inglés Integrated Development Environment ('**IDE**') es un programa compuesto por un conjunto de herramientas para un programador. Puede dedicarse en exclusiva a un sólo lenguaje de programación o bien, poder utilizarse para varios.



Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica GUI. Los IDEs pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes. El lenguaje Visual Basic por ejemplo puede ser usado dentro de las aplicaciones de Microsoft Office, lo que hace posible escribir sentencias Visual Basic en forma de macros para Microsoft Word.

Los IDEs proveen un marco de trabajo amigable para la mayoría de los lenguajes de programación tales como C++, Java, C#, Delphi, Visual Basic, etc. En algunos lenguajes, un IDE puede funcionar como un sistema en tiempo de ejecución, en donde se permite utilizar el lenguaje de programación en forma interactiva, sin necesidad de trabajo orientado a archivos de texto, como es el caso de Smalltalk u Objective-C.

1.4.4.1. Eclipse

Todo esto es posible gracias a que Eclipse se basa fuertemente en el uso de plug-ins. Hoy en día, existen plug-ins para hacer casi cualquier cosa, desde crear tests automáticos y generar código, hasta usar un CVS. Si bien se puede programar en distintos lenguajes, el que se utiliza más es, sin dudas, Java. Este sistema se distribuye bajo una licencia Open Source. Sin embargo, hay que considerar que está íntegramente desarrollado en Java, que no es libre, de modo que Eclipse tampoco lo es. Esta plataforma de desarrollo es una de las más utilizadas en la actualidad, ya que no sólo tiene una gran calidad, sino que también es inherentemente útil para programar en Java, que es un lenguaje multiplataforma. Esto nos permite integrar distintos tipos de ambientes (GNU/Linux, Windows, Mac, etc.) de manera casi transparente. El punto más fuerte de Eclipse es, sin duda, su capacidad para extenderse mediante plugins. La idea es, entonces, contar con una herramienta que pueda ser empleada por todos los usuarios, de modo tal que el pasaje de información entre distintas aplicaciones sea transparente. Si todos corren sobre la misma herramienta, es sencillo que dos aplicaciones compartan datos sin mucha dificultad. Dado que este entorno tiene una gran aceptación y una gran cantidad de usuarios, existen plugins para numerosas necesidades. (14)

1.4.4.2. Gambas

Es el único entorno en el cual la creación de GUIs está totalmente integrada y acoplada a la codificación; para decirlo de otra manera, funciona al mejor estilo Visual Basic. Simplemente, debemos “dibujar” los formularios y, luego, codificar los eventos que nos interesen. Aclaremos que



Gambas no pretende, ni pretenderá, ser un clon de Visual Basic, por lo que nunca se podrá “copiar y pegar” un proyecto de éste en aquél y esperar que funcione. Sin embargo, ya que ambos son Basic, existen muchísimas similitudes entre ellos. (14)

1.4.4.3. Anjuta

Funciona sobre GNOME. Se trata de un entorno de programación orientado a programar en C/C++. Si bien es posible utilizar otros lenguajes (tiene la sintaxis resaltada para Perl, Python, Java, PHP, SQL, JavaScript, Pascal, etc.), el punto fuerte es la programación en el lenguaje madre.

Otra característica que está integrada es la posibilidad de manejar un CVS. Si bien no hay una vista para ver gráficamente todo el árbol del CVS, sí tenemos opciones tendientes a ejecutar todos los comandos necesarios para administrarlo. El depurador incorporado es muy completo: nos permite definir breakpoints y watchpoints, y ver los estados de los registros, las señales, las variables, la memoria, etc. De manera análoga a los entornos anteriores, esta herramienta nos permite generar un TarGZ con todo nuestro proyecto. Un punto muy destacable es que Anjuta es altamente configurable: podemos desde configurar el tipo de indentación que queremos usar; hasta definir, para distintos lenguajes (los más populares y algunos no tan populares ya vienen configurados por defecto), cuáles son las formas de compilarlos, ejecutarlos o hacer un make. (14)

1.4.4.4. MonoDevelop

MonoDevelop es un entorno de desarrollo integrado libre y gratuito, diseñado primordialmente para C# y otros lenguajes .NET como Nemerle, Boo, y Java (vía IKVM.NET). **MonoDevelop** originalmente fue una adaptación de SharpDevelop para Gtk#, pero desde entonces se ha desarrollado para las necesidades de los desarrolladores de Mono. El IDE incluye manejo de clases, ayuda incorporada, completamiento de código, Stetic (diseñador de GUI), soporte para proyectos, y un depurador integrado.

Aunque **MonoDevelop** solo puede ejecutarse en Mac y distintas distribuciones de Linux, no es posible hasta el momento ejecutarlo sobre Windows. Sin embargo, un IDE llamado SharpDevelop facilita la compilación y depuración (desde su versión 2.0) de aplicaciones Mono sobre ambiente Windows.

De todos los proyectos, el único dedicado a la programación de C# es MonoDevelop. Digamos que éste sería el entorno de desarrollo oficial para programar en Mono (una implementación libre de la



plataforma .NET). Al igual que Mono, esta herramienta se distribuye libremente bajo la licencia GPL. La versión actual de MonoDevelop está alrededor de la 0.6.X, es decir que todavía está en sus inicios, y las versiones que circulan son inestables. Sin embargo, el proyecto Mono sí está muy avanzado, pero la creación de su IDE va mucho más lentamente. Para comenzar, notamos que está en sus inicios porque, simplemente, su instalación no es una tarea sencilla. Primero, es necesario tener instalado Mono (aunque si no lo tenemos, tampoco queremos trabajar con MonoDevelop, obviamente). Luego se requiere de varias librerías que deben instalarse en orden para que todo funcione bien. La única excepción es si lo instalamos en Debian, ya que haciendo el **apt-get** correspondiente, todo saldrá marchando solo. El sistema es totalmente elemental. Posee un editor de texto para escribir código, en el que sólo podemos realizar las operaciones básicas. (14)

1.4.4.5. KDevelop

Kdevelop es un entorno de desarrollo integrado libre y gratuito que combinado con Qt permite hacer aplicaciones con interfaz gráfica de gran calidad. KDevelop usa por defecto el editor de texto Kate. Se distribuye bajo la licencia GPL. Sin embargo, permite que el software generado utilizando esta herramienta se distribuya bajo la licencia que el usuario desee (ya sea GPL, Open Source o privativa). Para utilizar KDevelop no es necesario tener una máquina tan potente como en el caso de Eclipse, pero de todos modos, no es el entorno más liviano de todos los disponibles. Las características que se mencionan a continuación son específicas del entorno de desarrollo:

- Editor de código fuente con destacado de sintaxis e indentado automático (Kate).
- Gestión de diferentes tipos de proyectos, como Automake, qmake (para proyectos basados en la biblioteca Qt y Ant (para proyectos basados en Java).
- Navegador entre clases de la aplicación.
- Frontend para gcc, el conjunto de compiladores de GNU.
- Frontend para el depurador de GNU.
- Asistentes para generar y actualizar las definiciones de las clases y el framework de la aplicación.
- Completado automático del código en C y C++.



- Compatibilidad nativa con Doxygen.
- Permite control de versiones.

Otra característica muy práctica de KDevelop es que permite generar automáticamente un paquete con todas las fuentes. Luego, podremos generar un RPM para distribuir nuestra aplicación, tanto de las fuentes como de los binarios. Si el usuario recibe un paquete binario, entonces sólo le hará falta tener instaladas las bibliotecas extra que utilice el programa. Si recibe las fuentes, precisará tanto un compilador como todas las bibliotecas necesarias. Sin embargo, KDevelop generará todo el sistema de auto-configurar para que el usuario, en caso de que necesite compilar, sólo deba hacer el clásico `configure`, `make`, `make install`.(14)

Para dar solución a este sistema se utiliza KDevelop por que es un IDE más completo para Linux que trabaja con KDE, que es un entorno de escritorio casi completamente programado en C++, lo que facilita bastante el desarrollo de las funciones que ese quieren llevar a cabo pese a que Linux no ha alcanzado todavía las características referentes a la programación que posee Windows. MonoDevelop es una versión del .net para Linux pero en estos momentos todavía le faltan muchas funcionalidades por ser añadidas ante de poder desarrollar software de esta complejidad. También se debe recalcar las facilidades que brinda la ayuda del KDevelop y, pese a que es inglés, pues con una buena lógica de programación permite crear aplicaciones de calidad sin un conocimiento previo del entorno.

1.5. Bibliotecas, librerías y compiladores

1.5.1. Biblioteca Qt

Qt es una biblioteca multiplataforma para desarrollar interfaces gráficas de usuario. Fue creada por la compañía noruega Trolltech. Qt es utilizada en KDE, un entorno de escritorio para sistemas como GNU/Linux o FreeBSD, entre otros. Utiliza el lenguaje de programación C++ de forma nativa y además existen bindings para C, Python (PyQt), Java (Qt Jambi), Perl (PerlQt), Gambas (gb.qt), Ruby (QtRuby), PHP (PHP-Qt) y Mono (Qyoto) entre otros.

El API de la biblioteca cuenta con métodos para acceder a bases de datos mediante SQL, así como uso de XML y una multitud de otros para el manejo de ficheros, además de estructuras de datos tradicionales.

1.5.2. GCC 4.2.0



GCC 4.2.0 - Compilador que destaca por admitir multitud de lenguajes. GCC es una colección de compiladores que admite varios lenguajes: C, C++, Objective C, Chill, Fortran y Java así como también las librerías para éstos. Surgió dada la imposibilidad de compilar en un entorno que no fuese PC bajo MS-DOS.

Las siglas GCC significaban GNU C Compiler (Compilador C GNU). En la actualidad al admitir una colección de compiladores las siglas han pasado a significar GNU Compiler Collection (Colección de compiladores GNU).

GCC suministra al usuario muchas herramientas de comprobación de errores, integra una eficiente herramienta de depuración y dispone de muchas opciones de optimización de código, basándose en el microprocesador de destino u optimizaciones sobre la compilación de código inteligente.

Otras características importantes a resaltar son:

- * Soporte del procesador de Intel Itanium
- * Integración del compilador de Java GCJ
- * Eliminación del código muerto utilizando la representación SSA
- * Preprocesador C integrado en los compiladores C, C++ y Objective C
- * Permiso para renombrar el registro
- * Emisión del código Assembler x86 utilizando el estilo de sintaxis Intel
- * Potentes optimizaciones en las llamadas a subrutinas (CALL) optimizando los accesos al STACK (pila del procesador)(15)

1.5.3. Librería Time.h

Time.h es un archivo de cabecera de la librería estándar del lenguaje de programación C que contiene funciones para manipular y formatear la fecha y hora del sistema. Entre sus principales funciones se definen las que a continuación se verán ,en este caso se estudia esta librería específica por la importancia que reviste para el acabado del software.(16)

Nombre	Descripción
--------	-------------



<code>char * asctime(struct tm *)</code>	Recibe una variable de tipo puntero a estructura tm (struct tm*) y devuelve una cadena de caracteres cuyo formato es: "Www Mmm dd hh:mm:ss yyyy" (ej: Tue May 15 19:07.04 2001)
<code>cloc_t clock (void)</code>	Devuelve el número de pulsos de reloj desde que se inició el proceso
<code>char * ctime(time_t *)</code>	Recibe una variable de tipo puntero a time_t (time_t*) y devuelve una cadena con el mismo formato que asctime()
<code>double difftime(time_t, time_t)</code>	Recibe dos variables de tipo time_t, calcula su diferencia y devuelve el resultado (double) expresado en segundos.
<code>struct tm *gmtime(time_t *)</code>	Recibe un puntero a una variable de tiempo (time_t*) y devuelve su conversión como fecha/hora UTC a struct tm a través de un puntero.
<code>struct tm *localtime(time_t *)</code>	Similar funcionalidad a gmtime(), pero devuelve la conversión como fecha/hora LOCAL.
<code>time_t mktime(struct tm *)</code>	Inversamente a gmtime () y localtime (), recibe un puntero a struct tm (struct tm*) y devuelve su conversión al tipo time_t.
<code>time_t time(time_t *)</code>	Devuelve la fecha/hora (time_t) actual o -1 en caso de no ser posible. Si el argumento que se le pasa no es NULL, también asigna la fecha/hora actual a dicho argumento.
<code>size_t strftime(char *,size_t,char *,struct tm *)</code>	Formatea la información pasada mediante la estructura (struct tm*) mediante el formato indicado en una cadena (char*) e imprime el resultado sobre otra cadena (char*) hasta un límite de caracteres (size_t).



1.6. Conclusiones

En este capítulo se pudo constatar que ninguno de estos softwares es factible utilizarlos en nuestra universidad, porque se necesita desarrollar un software que impida que los estudiantes puedan acceder a él, que realice las cosas automáticamente, que genere reportes de cada acción que realice que sirva para Linux, y que se ajuste a una información que nosotros mismos debemos manejar desde una base de datos. Además se ha realizado un estudio sobre las principales herramientas que se utilizaran para desarrollar este software, así como explicado por qué es necesario crear un software adaptable a las condiciones específicas de la Universidad de las Ciencias Informáticas.



CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

2.1. Introducción

Antes de resolver e incluso diseñar el software cuyo propósito es dar solución al problema que nos hayan planteado, debemos modelar una solución inicial del mismo para tener un conocimiento de cómo es aplicado en la actualidad para personalizar los cambios o mejoras que le vamos a realizar. Esta es la idea que se persigue con en este capítulo en el cual se estudiarán los casos de usos y diversos modelos que se van a automatizar.

Frecuentemente los sistemas (conjuntos de procesos y subprocesos integrados en una organización) son difíciles de comprender, amplios, complejos y confusos; con múltiples puntos de contacto entre sí y con un buen número de áreas funcionales, departamentos y puestos implicados. Un modelo puede dar la oportunidad de organizar y documentar la información sobre un sistema.

Cuando el flujo de información es difuso, dudoso, inexacto, con múltiples orígenes o solo contiene eventos o sucesos, cuando existe la imposibilidad de determinar subsistemas por el exceso de interconexiones, cuando hay un solapamiento de responsabilidades y además es difícil establecer reglas de funcionamiento, por citar algunos casos, estamos en presencia de un entorno organizacional el cual deja de funcionar como un negocio.

2.2. Características del modelamiento del negocio (Modelo de Dominio).

En el modelamiento del negocio se pueden definir como objetivos:

Alcanzar una mayor comprensión de las características de la organización a la cual se le va a implementar el sistema, entre ellas se encuentra su estructura y su dinámica.

Definir los principales problemas que presenta la organización en la actualidad y así identificar las mejoras que se le van a realizar.

Garantizar la comprensión por parte de los consumidores, los usuarios finales y los desarrolladores de la organización a la que pertenecen.

Definir los requerimientos necesarios para el futuro sistema que se va a implementar.

De la situación que se presente se definirá si existe un flujo en los procesos que se manejan en el negocio para modelar el mismo o si en las características están tan relacionadas y confusas que no



se logra definir el mismo y hay que utilizar otra alternativa. Esta alternativa será el modelo de dominio, del cual se definirán ahora algunas características.

Un modelo del dominio captura los tipos más importantes de objetos que existen o los eventos que suceden en el entorno donde estará el sistema.

El modelo del dominio se considera en RUP un subconjunto del llamado *modelo de objetos del negocio*. Se puede desarrollar un modelo de objetos del negocio enfocado a la explicación de los productos, entregables o eventos que son importantes en el negocio. Esos modelos, que no incluyen las responsabilidades de las personas que ejecutan las actividades, se refieren a veces como modelo del dominio). (17)

2.2.1. Desarrollo del Dominio

En la Universidad de las Ciencias Informáticas se han entregado computadoras con infinidad de recursos en pos del aprendizaje de los estudiantes, recursos que en su mayoría no son utilizados por la preferencia que representan para los estudiantes otras actividades algo desfavorables de las que hablaremos a continuación.

Juegos en los laboratorios: Los estudiantes (principales usuarios de la red) acostumbran utilizar mucho tiempo de los laboratorios jugando, mientras otros, que en realidad tienen otras necesidades pierden la posibilidad de utilizar el servicio, en la actualidad para contrarrestar este mal habito los técnicos vigilan a los estudiantes y en caso de realizar esta actividad les ordenan abandonar el laboratorio, además de bajar la resolución de video a 0 para evitar que se pueda jugar.

Instalación de Proxy y keylogger: Para contrarrestar esta actividad se han instalado softwares .En caso de los keylogger el kaspersky se encarga de monitorear la instalación de alguno, además de anteriormente la instalación de antikeylogger.Con respecto a los proxys se ha instalado el software GFI,que se encarga de evaluar los puertos hábiles para los usuarios para dilucidar si se ha utilizado otro ,además de la cantidad de conexiones a una PC en un horario determinado, también se utiliza el Netscan para evaluar las carpetas compartidas.

Después de este estudio se determina que lo más factible es realizar un modelo de dominio para lo cual nos apoyaremos en DoMet:

DoMet: Método para la modelación de entornos organizacionales con flujos de información complejos y difusos.DoMet consta de las siguientes actividades:



1. Determinación de la estructura organizacional: consiste en la determinación de la jerarquía por niveles de la organización en estudio, así como las relaciones entre los diferentes niveles y partes de esta estructura, permitiendo determinar partes generadoras y receptoras de información, así como el alcance de conocimiento por las partes de cada una de estas informaciones. El artefacto resultante es el **Organigrama**.
2. Determinación de los eventos principales que ocurren: al no poder delimitarse los procesos por no existir fronteras entre los mismos, se hace necesario listar los eventos principales que ocurren en la organización, y los enlaces con otros eventos de la misma, especificando solapamientos, interconexiones, roles que intervienen, breve descripción de los sucesos y objetivos de estos eventos, así como frecuencia de ocurrencia. El artefacto resultante es la Descripción de Eventos Principales.
3. Determinar informaciones que se manejan: se hace necesario conocer cada una de las informaciones que se manejan por sencillas que puedan parecer y en los formatos que se utilicen (verbal, escrita, digital, entre otras) y la descripción de su contenido o composición. Dado que estas informaciones pueden ser objetos palpables físicos, conceptos que se manejen en la organización o informaciones propiamente dichas, los artefactos resultantes de esta actividad son: el Registro de Informaciones, el Registro de Objetos y el Glosario de Conceptos.
4. Determinar quién o quienes participan y sus roles: al determinar los eventos principales que ocurren, así como el organigrama de la organización, se necesita determinar las personas, entidades, sistemas u otros componentes que interactúan en los eventos principales descritos y una breve descripción de los mismos. El artefacto resultante de realizar esta actividad es Registro de Roles.
5. Definición de clases: contando con las informaciones contenidas entre 1 y 4 descritas anteriormente, se procede a convertir los niveles del organigrama, los eventos principales, las informaciones, los objetos, los conceptos y los roles en clases o entidades de interés para el entendimiento del funcionamiento de la organización construyendo como artefacto resultante, las clases del dominio.
6. Relacionar clases del dominio: como última actividad de DoMet está el relacionar las clases que forman parte del modelo, de acuerdo a lo descrito anteriormente en los pasos previos, quedando como artefacto resultante el modelo de dominio, con su diagrama de clases de dominio tal y como establece el Proceso Unificado de Desarrollo de Software (RUP en sus siglas en inglés).(18)



-Organigrama: Universidad de las Ciencias Informáticas, facultad, laboratorios de docencia, Departamento de Seguridad Informática, Directivos de los laboratorios.

-eventos principales: Juegos en red, Instalación de keylogger, instalación de proxys, búsquedas remotas

Objetos reales: ficheros, computadoras, software.

Quien o quienes: usuarios de la computadora (Estudiante), administrador de redes, técnico de laboratorio, especialista de seguridad informática, Netscan, FGI, Kaspersky.

2.2.2. Representación del modelo de Dominio

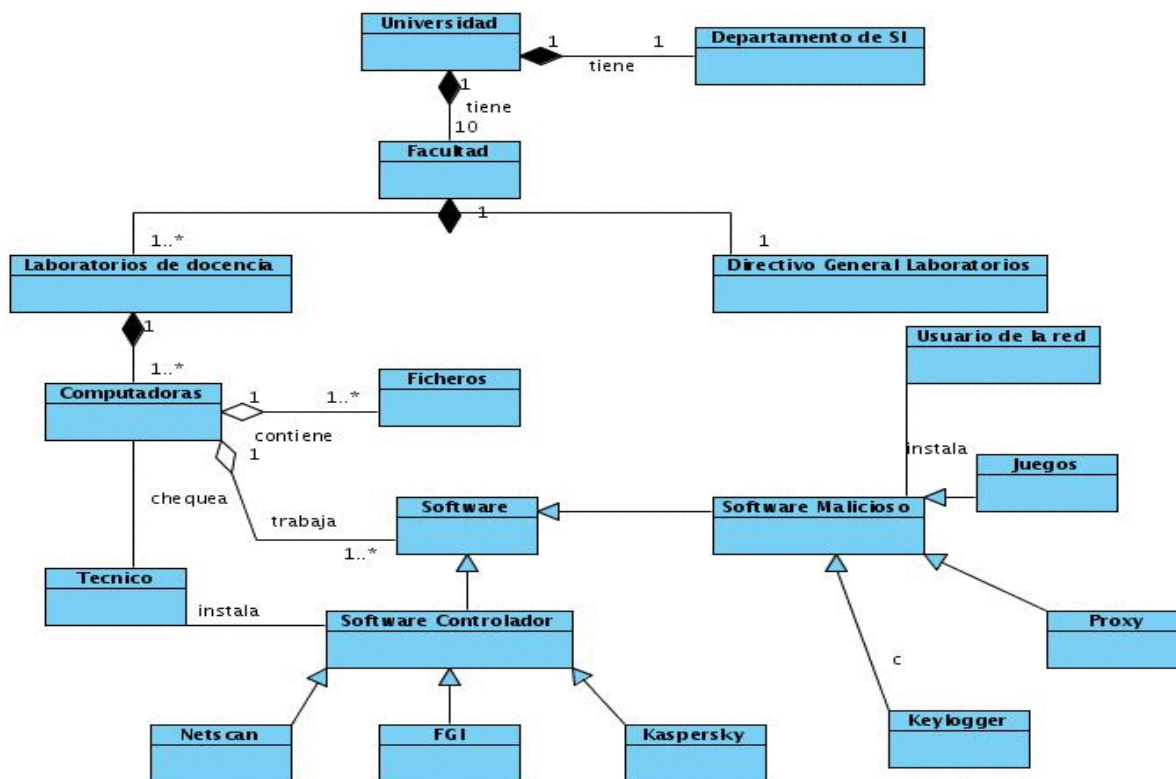


Figura 2.1. Modelo de dominio

2.3. Propuesta del sistema de control de procesos de los ordenadores de la UCI



Este sistema en conjunto va a constar de dos sistemas implementados de forma separada, cada cual cumpliendo características específicas e interactuando con la misma base de datos.

Es bueno aclarar que un sistema funcionara dependiendo del otro debido a que de los procesos que reconozca el principal se generaran los reportes a los cuales se va a acceder mediante el segundo. Ahora se hará una breve reseña de cada uno.

2.3.1. Sistema de control de procesos en los ordenadores:

El sistema debe estar a la escucha de todos los procesos que se están ejecutando en la PC y comparar los mismos con una lista negra de procesos, la cual es obtenida desde una BD para ver si sus características coinciden con estos. En caso de que se detecte un juego o la instalación de un Proxy o keylogger, el proceso deberá ser detenido y avisado al usuario de la red. Este software solo podrá ser accedido y actualizado por los administradores del mismo, los usuarios de la red (estudiantes) no deben interactuar con el.

Permitirá gestionar procesos para mantener actualizada la base de datos de los procesos prohibidos, además de manejar el chequeo de procesos ejecutándose y la detención de los mismos.

Se encargara de actualizar las características de la base de datos con la cual se va a realizar la conexión en caso de ser requerido además de actualizar mediante la interfaz el valor del intervalo del Timer y la hora de acceso a la base de datos para obtener la lista de procesos prohibidos en caso de considerar que necesitan ser cambiados para uno más óptimos.

2.3.2. Sistema de manejo de reportes

El sistema debe permitir a los usuarios que lo accedan mostrar los diversos reportes que se puedan definir con la información que se obtenga de la base de datos. Permitirá manejar los reportes de la base de datos.

2.4. Especificación de los requerimientos del software

Los requerimientos tanto funcionales como no funcionales son una parte fundamental del procesos de formación del software, pues en ellos se define lo que se desea producir tanto desde el punto de vista físico ,es decir ,como se vera el software ,hasta desde el punto de vista funcional ,pues con ellos se define que el software debe llegar a hacer.

2.4.1. Requerimientos no funcionales



Los requerimientos no funcionales son las propiedades que tiene el sistema pero no desde un punto de vista funcional, mas bien representan la definición de la forma en la que el usuario va a ver el producto ya acabado aunque siempre tendrán alguna relación con los funcionales pues de cómo se defina por ejemplo que será el rendimiento, así será la funcionalidad del software.

2.4.1.1. Requerimientos de apariencia o interfaz externa:

El software debe tener una interfaz amigable y comprensible dada las funciones que debe cumplir, para facilitarles a las personas que accedan a él su trabajo.

2.4.1.2. Requerimiento de Usabilidad:

Sólo las personas con permiso de Administración, es decir, acceso por la cuenta de superusuario (root) harán uso del software, los usuarios de la red, pese a ser los factores de la construcción del mismo no podrán acceder a él para manejar ninguna funcionalidad

2.4.1.3. Requerimiento de rendimiento:

La aplicación debe ser rápida a la hora de establecer consultas al servidor de base de datos y realizar de forma óptima sus responsabilidades

2.4.1.4. Requerimiento de Portabilidad:

La aplicación permitirá ser ejecutada en el sistema operativo Linux .Estará instalada en cada PC cliente de los laboratorios y accederá a un Servidor de Base de Datos que corra sobre Linux. El IDE de desarrollo será KDevelop y el Gestor de BD, MySQL.

2.4.1.5. Requerimiento de Soporte:

Para garantizar el soporte de esta aplicación se les explicará a los técnicos del laboratorio, a los trabajadores del departamento de seguridad informática y a quien pueda interesar las funcionalidades del mismo para una mayor comprensión. También se requerirá de una PC en función de servidor para instalar el servidor de base de datos.

2.4.1.6. Requerimiento de Seguridad:

1-Confidencialidad



La aplicación estará protegida de acceso no autorizado a la misma Sólo el administrador de la computadora tendrá permiso para acceder a esta y realizar los cambios pertinentes.

2-Integridad: La información que genere o almacene este documento debe estar protegida de cualquier fallo previsible.No debe permitirse libre accesos ni al software ni a la BD

3-Disponibilidad:

La aplicación debe estar disponible para las personas que tengan derecho a acceder a la misma independientemente de la seguridad que posea.

2.4.1.7. Requerimiento de Software:

Para hacer uso de la aplicación los usuarios necesitan de una computadora que sólo requiere tener instalado el sistema operativo Linux; en el Servidor de Base de Datos se utilizará la Distribución Red Hat Enterprise Linux 4. Para la implementación de la aplicación se utilizará el Gestor de Base de Datos MySQL y como herramienta de desarrollo KDevelop.

2.4.1.8. Requerimiento de Hardware:

Servidor: Pentium 4, CPU 1.00 GHz, 1GB de RAM.

Cliente: Como mínimo una computadora de 128 MB de RAM.

Todas las computadoras deben estar conectadas a una red y tener al menos 100 MB de velocidad.

2.4.2. Requerimientos funcionales

Los requerimientos funcionales son las capacidades o condiciones que el sistema debe cumplir basados en las necesidades del cliente.

En este caso se esbozaran los requerimientos de ambos sistemas.

RF1-Permitir establecer parámetros de conexión de BD así como establecer la misma, en caso de presentarse alguna contingencia con el servidor actual

RF2-Detener procesos en ejecución automáticamente

-**RF2.1**Mostrar un mensaje de notificación al usuario

RF3- Permitir gestionar los procesos de la Base de datos



- **RF3.1** Adicionar procesos a la base de datos

-**RF3.2** Eliminar procesos de la Base de Datos

RF4- Mostrar reportes de la información que se guarda en la Base de Datos

-**RF4.1** Devolver la cantidad de veces que se ha ejecutado un proceso dado

-**RF4.2** devolver dado una dirección IP los procesos que le han sido detenidos

-**RF4.3** Devolver IP de PC utilizadas como Proxy

RF5- Permitir modificar intervalo del timer en caso de considerar que el actual representa algún tipo de desventaja.

RF6-Permitir actualizar la hora de acceso a la base de datos

2.5. Definición de actor del sistema

Un actor del sistema es la persona, empresa, hardware que interactúan con el sistema para beneficiarse de alguna forma del mismo .Este puede tanto intercambiar información como ser un actor pasivo del sistema

Actores del sistema	Descripción
Administrador de la computadora	Persona que se encarga de interactuar con el sistema explotando todas las funcionalidades que permite ,tanto en la gestión de procesos como para realizar las modificaciones
Usuario de la computadora	Persona que levanta los procesos prohibidos que se van a detener.

2.6. Casos de uso del sistema

Un caso de uso del sistema representa a un proceso de sistema, por lo que se corresponde con una secuencia de acciones que producen un resultado observable para los actores del sistema que



interactúan con el mismo. Desde la perspectiva de un actor individual, define un flujo de trabajo completo que produce resultados deseables.

CU-1	CU_Modificar_Conexion_Base_Datos
Actor	Administrador de la computadora
Descripción	El caso de uso inicia cuando el administrador del sistema necesita modificar los atributos de la base datos actual para crear una nueva conexión.El sistema insertara los nuevos datos , enviara un mensaje de confirmación y guardará la información en un fichero
Referencia	RF1

CU-2	CU_Detener_Procesos_Automaticamente
Actor	Usuario de la computadora
Descripción	El caso de cuando un usuario de la red levanta un proceso de los que están comprendidos como prohibidos en la BD .El software ,actualizado por intervalos de tiempo ,debe estar ala escucha de los procesos, detener el proceso en caso de que sea prohibido y enviar un mensaje de notificación al usuario
Referencia	RF2

CU-3	CU_Actualizar_Lista Procesos
Actor	Administrador de la computadora
Descripción	El caso de uso inicia cuando el administrador del sistema, ya autenticado accede a la funcionalidad “Actualizar Procesos ”.El sistema le mostrara una



	ventana auxiliar que le permitirá escoger que desea hacer; adicionar o eliminar un proceso
Referencia	RF3

CU-4	Cu _ Mostrar_ reportes
Actor	Administrador de la computadora
Descripción	El caso de uso inicia cuando el administrador del sistema ya autenticado, accede a la funcionalidad “Mostar Reportes”, la cual le permitirá conocer diversas informaciones
Referencia	RF4

CU-5	CU_Modificar_intervalo
Actor	Administrador de la computadora
Descripción	El caso de uso inicia cuando el administrador del sistema accede a la funcionalidad que permite cambiar el intervalo del timer ,en caso de que la acción se realice satisfactoriamente el sistema enviara un mensaje de confirmación
Referencia	RF5

CU-6	CU_Modificar_Hora_Acceso_Base_Datos
Actor	Administrador de la computadora
Descripción	El caso de uso inicia cuando el administrador del sistema necesita modificar los atributos de la hora a la que se accederá a la base de datos para devolver la

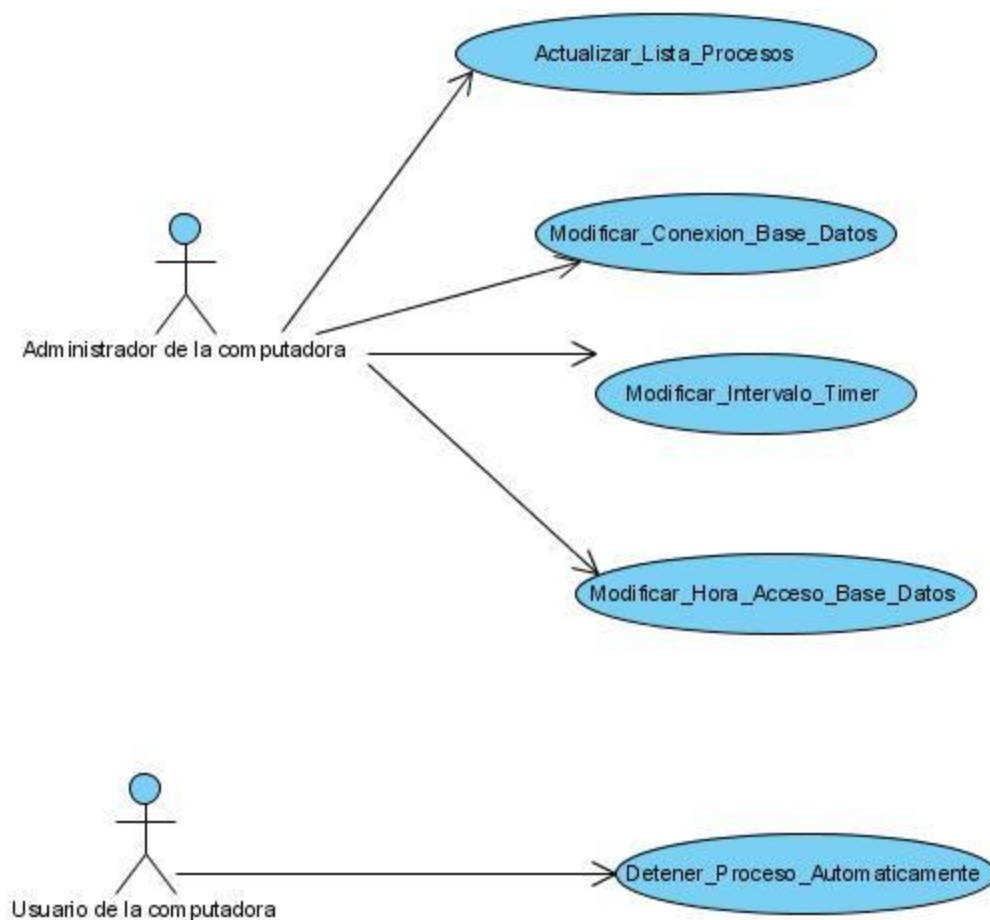


	lista de los procesos prohibidos .El sistema actualizara la información.
Referencia	RF6

2.7. Diagramas de caso de uso del sistema:

Los diagramas de caso del sistema sirven para especificar la funcionalidad y el comportamiento de un sistema mediante sus interacción con los usuarios y/o otros sistemas.Es decir, representa gráficamente los procesos y su interacción con los actores del sistema.En este caso se definirán los diagramas de ambos sistemas.

2.7.1. Diagrama de caso de uso del Sistema de control de procesos





2.7.2. Diagrama de caso de uso del sistema de gestión de reportes



2.8. Casos de uso expandidos

2.8.1. CU_Modificar_Conexion_Base_Datos.

CU_1	CU_Modificar_Conexion_Base_datos	
Propósito	Este Caso de uso tiene como propósito establecer los parámetros de conexión con la BD así como establecer la conexión con la misma.	
Actor: Administrador de la computadora		
Resumen: El caso de uso se inicia cuando el administrador de la computadora accede al sistema para modificar los parámetros de conexión de la base de datos utilizada por el sistema y crear una nueva.		
Referencia	RF1	
Precondiciones		
Acción del Actor		Respuesta del sistema
1-El caso de uso comienza cuando el autor accede a la funcionalidad que permite cambiar la base de datos		2-El Sistema muestra la interfaz con los campos que debe llenar



3-El usuario llena los campos necesarios	
4-El usuario selecciona el botón "Salvar"	5-El sistema actualiza los datos
	6-El sistema guarda la información en un fichero
7-El usuario selecciona el botón " crear Nueva Conexión"	8-El sistema establece una nueva conexión
Flujos Alternos	
Acción del Autor	Respuesta del Sistema
Accion1	1.1 El sistema muestra el mensaje "No se deben dejar campos vacíos."
1.2 El usuario pulsa el botón "OK"	1.3 Se pasa a la acción 2 del caso de uso
Acción 5	5.1 El sistema muestra el mensaje "No se pudo crear nueva conexión"
5.2 El usuario pulsa el botón "OK"	Se pasa a la Accion1

Acción #1 si el usuario pulsa el botón "Aceptar dejando campos vacíos".

Acción #5 Si los parámetros de conexión están incorrectos

2.8.2 .CU_Detener_Procesos.

CU_2	CU_Detener_Procesos.
Propósito	Este caso de uso tiene como propósito reconocer cuando el usuario levanta procesos prohibidos y detenerlos
Actor: Usuario de la red	
Resumen: El caso de uso inicia cuando el usuario de la red levanta un proceso de juego o cualquier	



otro que se considere prohibido que será detenido por el sistema.	
Referencias	RF2
Precondiciones	
Acción del actor	Respuesta del sistema
1-El usuario de la red ejecuta un proceso que se considera prohibido	2-El sistema se mantiene a la escucha de los procesos ejecutándose y los considerados prohibidos en la computadora por intervalos de tiempo
	3-El sistema compara las listas de procesos
	4-El sistema detecta el proceso
	5-El sistema detiene el proceso
	6- Envía un mensaje de aclaración al usuario
	7-El sistema actualiza los reportes en la BD

2.8.3. CU_Actualizar_Lista_Proceso

CU_3	CU _ Actualizar_lista_procesos
Propósito	El propósito de este caso de uso es actualizar los cambios que se realicen en la lista de procesos que se localiza en la base de datos, tanto adicionarlos como eliminarlos.
Actor: Administrador de la computadora	
Resumen: El caso de uso inicia cuando el administrador accede al funcionalidad del sistema “Gestionar Procesos”,la cual le permitirá adicionar o eliminar procesos	
Referencias	RF3
Precondiciones	



Acción del actor	Respuesta del sistema
1-El actor selecciona la opción "Gestionar procesos"	2-El sistema muestra las opciones "Adicionar Procesos" y "Eliminar Procesos".
3-Si el administrador selecciona la opción: a)"Adicionar Procesos".Ver sección:"Adicionar Procesos". b)"Eliminar Procesos ".Ver sección : "Eliminar Procesos".	
Sección " Adicionar Procesos" .	
	4-El Sistema muestra la interfaz que permite adicionar procesos
5-El administrador llena los campos con los datos que del proceso que desea adicionar	
6-El usuario pulsa el botón "Aceptar"	7-El sistema verifica si el proceso ya existe
	8-El sistema adiciona el nuevo proceso
Sección "Eliminar Usuario"	
	9-El sistema muestra la interfaz que permite eliminar Procesos
10-El administrador llena los campos necesarios	
11-El administrador pulsa el botón "Aceptar"	12-El sistema verifica que el proceso exista.
	13-El sistema muestra el mensaje "proceso eliminado satisfactoriamente.



Flujos alternos	Respuesta del sistema
Acción 1F1	1F1.1 El sistema muestra el mensaje "Ha dejado campos vacíos".
1F1.2 El usuario pulsa el botón "OK"	1F1.3 Se pasa a la acción 7 del caso de uso.
Accion2	2.1 El sistema muestra mensaje "El proceso ya existe"
2.2 El usuario pulsa el botón "OK"	2.3 Se pasa a la acción 4 del caso de uso.
Acción 1F2	1F2.1 El sistema muestra el mensaje "Ha dejado campos vacíos"
1F2.2 El usuario pulsa el botón OK	1F2.3 Se pasa a la acción 12 del caso de uso
Accion3	3.1 El sistema muestra el mensaje "Procesos a eliminar incorrecto".
3.2 El usuario pulsa el botón "OK"	3.2 Se pasa a la acción 9 del caso de uso

Accion#1F1 Si se han dejado campos vacíos al adicionar

Acción#1F2 Si se han dejado campos vacíos al eliminar

Acción #2 Si el procesos a adicionar ya existe.

Acción #3 Si el proceso a eliminar no existe.

2.8.4. CU_Mostrar_Reportes

CU_4	CU_Mostrar_Reportes
Propósito	Este caso de uso tiene como objetivo mostrar diversos reportes sobre los procesos
Actor :Administrador de la computadora	



Resumen: el caso de uso inicia cuando el administrador escoge la opción mostrar reportes ,en la cual se le desplegaran los diversos reportes que el sistema contiene	
Referencia	RF4
Acción del actor	Respuesta del sistema
1-El usuario selecciona la opción "Mostrar Reportes"	2-El sistema muestra las opciones "Saber juegos jugados por un usuario ","Saber cantidad de veces que se ha jugado un juego dado" y "Devolver PCs en función de Proxy".
3-Si el actor selecciona la opción: a)" Saber juegos jugados por un usuario ", ver sección "Saber juegos jugados por un usuario". b)" Saber cantidad de veces que se ha jugado un juego dado",ver sección " Saber cantidad de veces que se ha jugado un juego dado " c)"Devolver PCs en función de Proxy " ,ver sección "Devolver PCs en función de Proxy"	
Sección "Saber juegos jugados en una dirección IP "	
	4-El sistema muestra la información de los juegos jugados por el usuario.
Sección "Saber cantidad de veces que se ha jugado un juego dado"	
	5-El sistema muestra la información de la cantidad de veces que se ha jugado determinado juego
Sección "Devolver PCs en función de Proxy"	6-El sistema muestra información de todos los



	IP de PCs que realizan esta función
--	-------------------------------------

2.8.5. CU_Modificar_Intervalo.

CU_5	CU_Modificar_intervalo
Propósito	El objetivo de este caso de uso es brindar la posibilidad de cambiar el intervalo en el que se obtendrá la lista de procesos ejecutándose en la PC
Actor : Administrador de la computadora	
Resumen: El caso de uso inicia cuando el administrador accede a la aplicación para modificar valor que tiene el timer en la base de datos. Inserta los datos necesarios y el sistema envía un mensaje de confirmación	
Referencia	RF5
Acciones del actor	Respuesta del sistema
1-El actor selecciona la opción “Trabajar con timer”	2-El sistema muestra la interfaz para insertar los datos pertinentes para realizar el cambio.
3-El escoge el dato a tener en cuenta.	
4-El administrador pulsa le botón buscar “Aceptar”.	
	5-El sistema actualiza la información
	6-El sistema guarda la información en un fichero
Flujos Alternos	Respuesta del sistema
Acción 4	4.1-El sistema muestra el mensaje “debe llenar todos los campos”
4.2-El usuario pulsa “OK”	4.3-El sistema pasa a la acción 2 del caso de



	USO
--	-----

Acción #4: si el usuario deja los campos vacíos.

2.8.6. CU_Modificar_Hora_Acceso_Base_Datos

CU_6	CU_Modificar_Hora_Acceso_Base_Datos	
Propósito	El objetivo de este caso de uso es brindar la posibilidad de cambiar el valor de la hora en que se obtendrá la lista de procesos	
Actor	: Administrador de la computadora	
Resumen:	El caso de uso inicia cuando el administrador accede a la aplicación para modificar valor que tiene el existe en el fichero que guarda la hora en la que se actualizara la lista de procesos prohibidos.	
Referencia	RF6	
Acciones del actor	Respuesta del sistema	
1-El actor selecciona la opción “Trabajar con timer”	2-El sistema muestra la interfaz para insertar los datos pertinentes para realizar el cambio.	
3-El administrador del sistema escoge el dato a tener en cuenta.		
4-El administrador pulsa le botón busca “Aceptar”.		
	5-El sistema actualiza la información	
	6-El sistema guarda la información en un fichero	
Flujos Alternos	Respuesta del sistema	
Acción 6	4.1-El sistema muestra el mensaje “Debe	



	escoger un nuevo valor”
4.2-El usuario pulsa “OK”	4.3-El sistema pasa a la acción 2 del caso de uso

Acción #6: si el usuario no escoge un valor

2.9Conclusiones:

En este capítulo quedaron reflejadas las principales características relacionadas con el dominio y los principales puntos a tener en cuenta en el sistema, tales como las diversas descripciones de los casos de uso y los principales requerimientos, siempre teniendo en cuenta que este sistema debe estar ejecutando como un servicio más que preste nuestro sistema operativo.



CAPÍTULO 3: ANALISIS Y DISEÑO

3.1. Introducción

En un sistema, el análisis y diseño del mismo consta como uno de los flujos más importantes dentro del proceso unificado del desarrollo de un software. Además manifiestan una cercanía mayor al acceso de datos y a la parte física de la aplicación que se va a desarrollar.

En el modelo de análisis se observa una especificación más exacta de los requisitos, ya que este se encarga de estructurarlos de modo que sea mas fácil su preparación, su comprensión, su modificación y en general su mantenimiento. Creando así una primera aproximación al modelo de diseño. Con el modelo de diseño se va a crear una imagen inicial y probablemente definitiva de lo que va a ser el sistema en si dándole forma a los requisitos ya desde un punto de vista automatizado, que es en si el sistema que vamos a desarrollar usando para esto las herramientas que se definieron en el capitulo anterior .

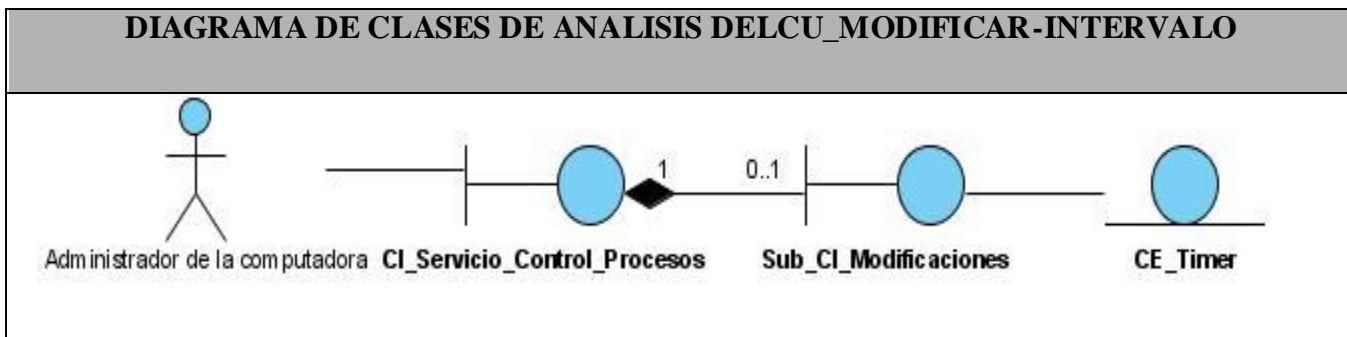
Durante el análisis se ven desde otra perspectiva los requisitos que se describieron en la captura de los mismos, en este flujos estos requisitos son refinados y estructurados buscando una manera de obtener una comprensión y una descripción que sean más precisas y fáciles de mantener, para que la arquitectura del proyecto alcance una estructura sólida.

En el modelo de clases del análisis además, definen diversos artefactos, tal es el caso de las clases de interfaz, entidad y control. Una clase del análisis representa una abstracción de una o varias clases y/o subsistemas del modelo del diseño.



Por otro lado, el modelo de diseño se encarga de intentar mantener la estructura creada por el modelo de análisis. También es el encargado de describir la realización física de los casos de uso, precisando principalmente el impacto que tienen los requerimientos funcionales y no funcionales en el sistema y como influyen en el entorno de implementación del mismo. También puede ser útil para visualizar la implementación y para soportar las técnicas de la programación grafica del sistema.

3.2. Diagramas de clases del análisis del Sistema de control de procesos en los ordenadores



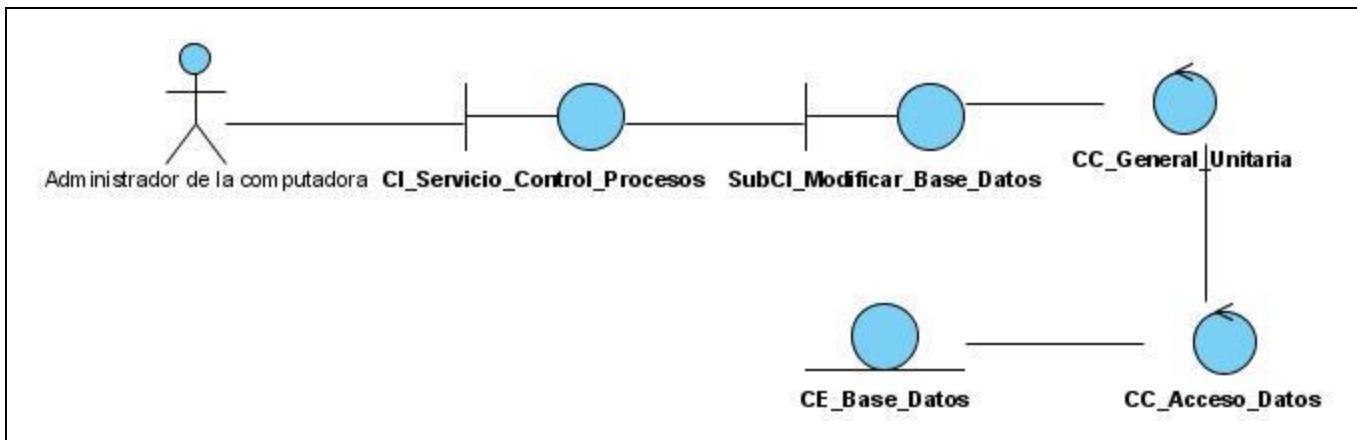


DIAGRAMA DE CLASES DEL ANALISIS DEL CU_MIDIFICAR_HORA_ACCESO_BASE_DATOS

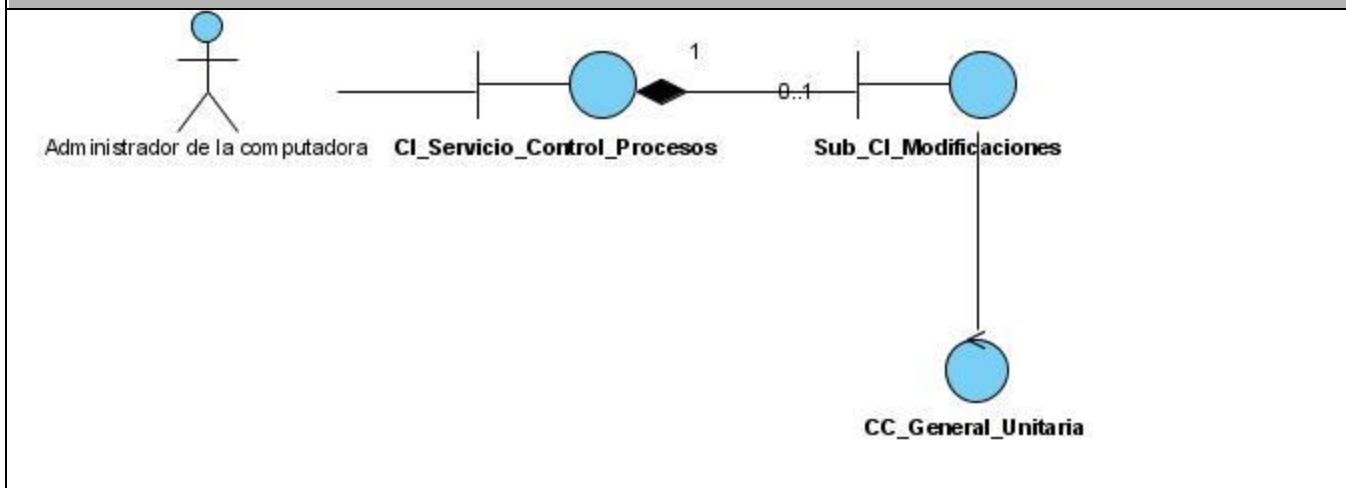




DIAGRAMA DE CLASES DEL ANALISIS DEL CU_DETENER_PROCESOS_ AUTOMATICAMENTE

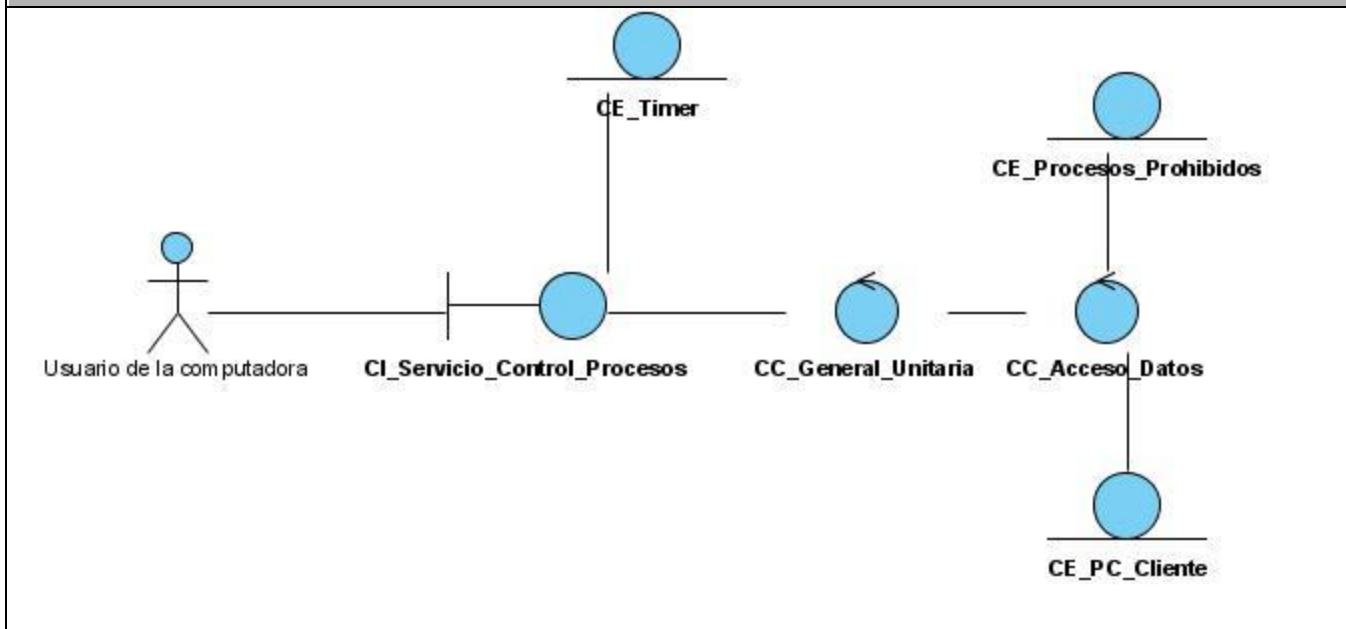
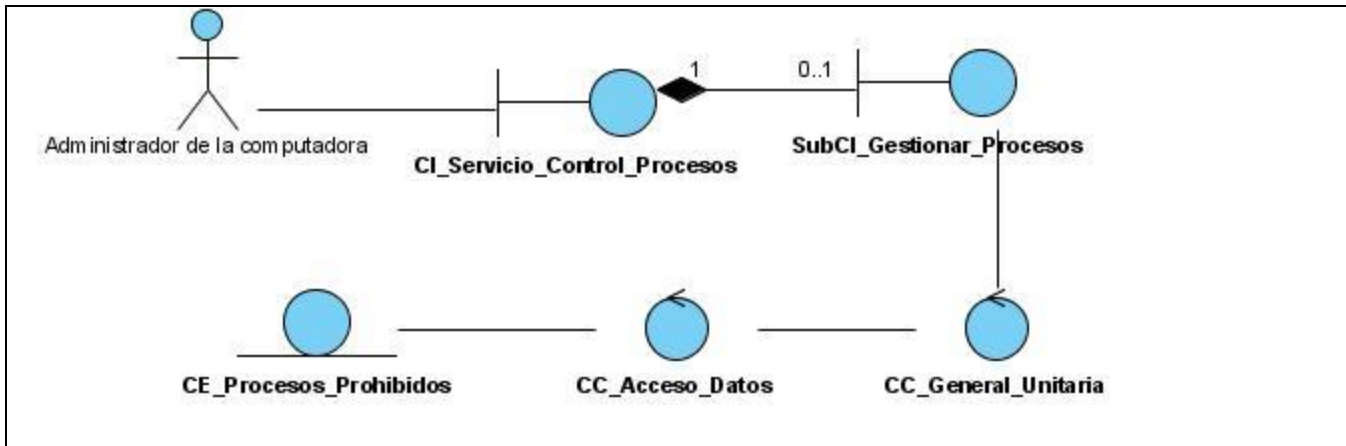
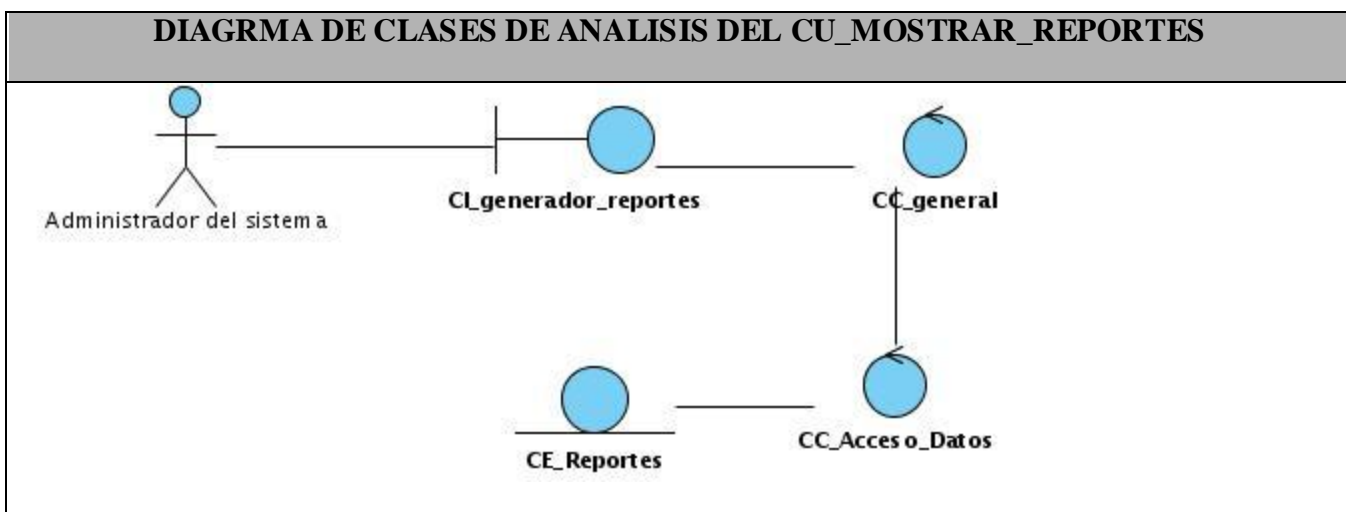


DIAGRAMA DE CLASES DEL ANALISIS DEL CU_ACTUALIZAR_LISTA_PROCESOS



3.3. Diagrama de clases del sistema de manejo de Reportes



3.4. Diagramas de secuencia

Como su nombre lo indica, los diagramas de secuencia representan una secuencia temporal de eventos basados en una interacción ordenada del escenario donde se desarrolle un determinado caso de uso. Se encargan de documentar el flujo lógico de eventos que ocurren en la aplicación, además de observar la vida de los objetos en el sistema.



DIAGRAMA DE SECUENCIA CU _ ACTUALIZAR _LISTA _ PROCESOS

sd Communication Diagram1 CU_Actualizar_Lista_procesos

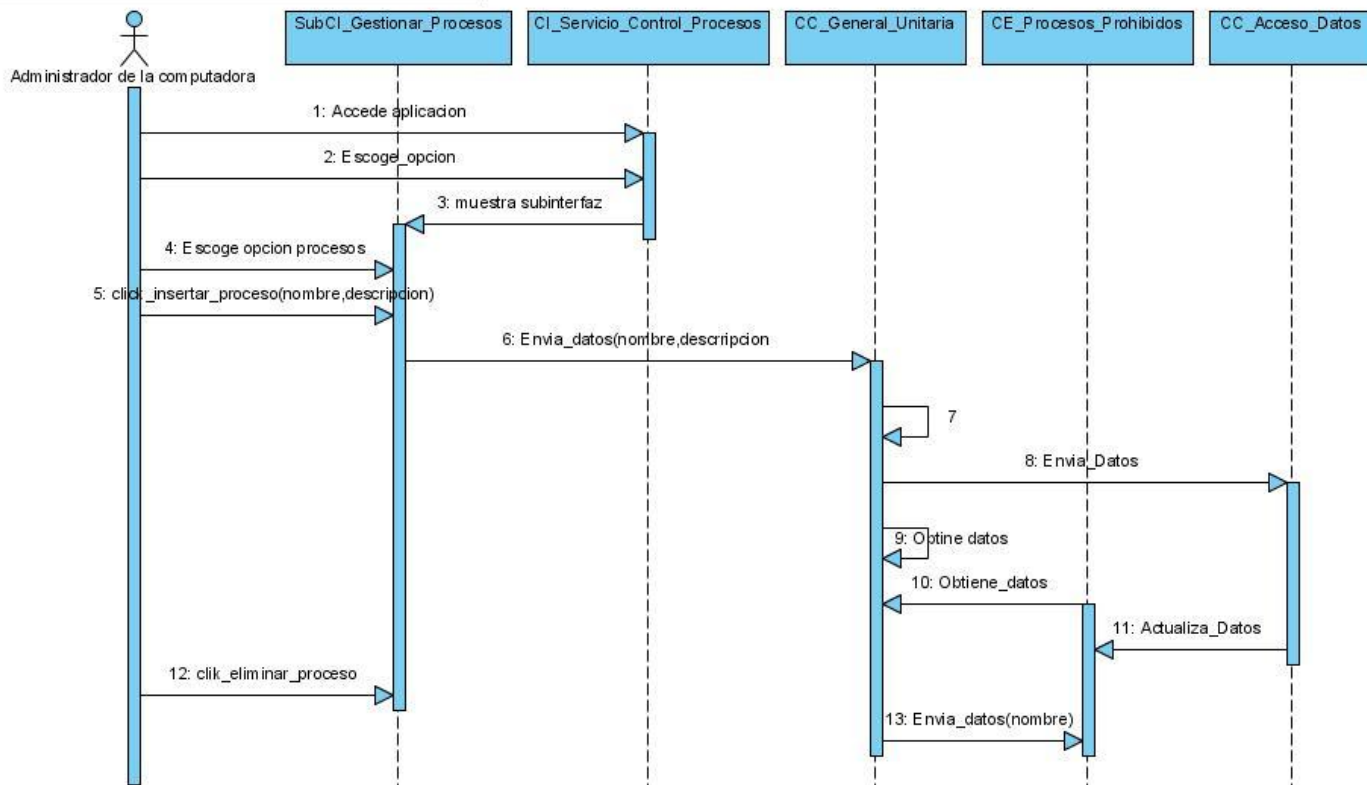




DIAGRAMA DE SECUENCIA CU_MODIFICAR_INTERVALO

sd Communication Diagram2CUModificar_Intervalo

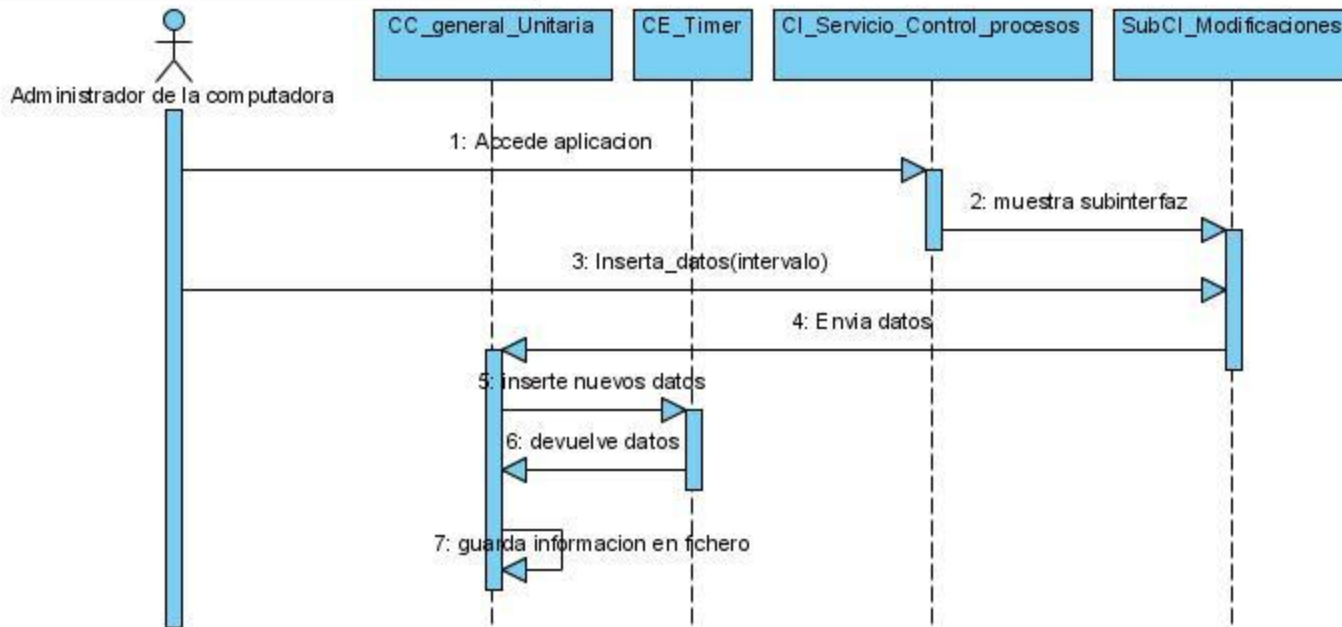




DIAGRAMA DE SECUENCIA CU_MODIFICAR_CONEXION_BASE_DATOS

sd Communication Diagram3_Modificar_Conexion_base_Datos

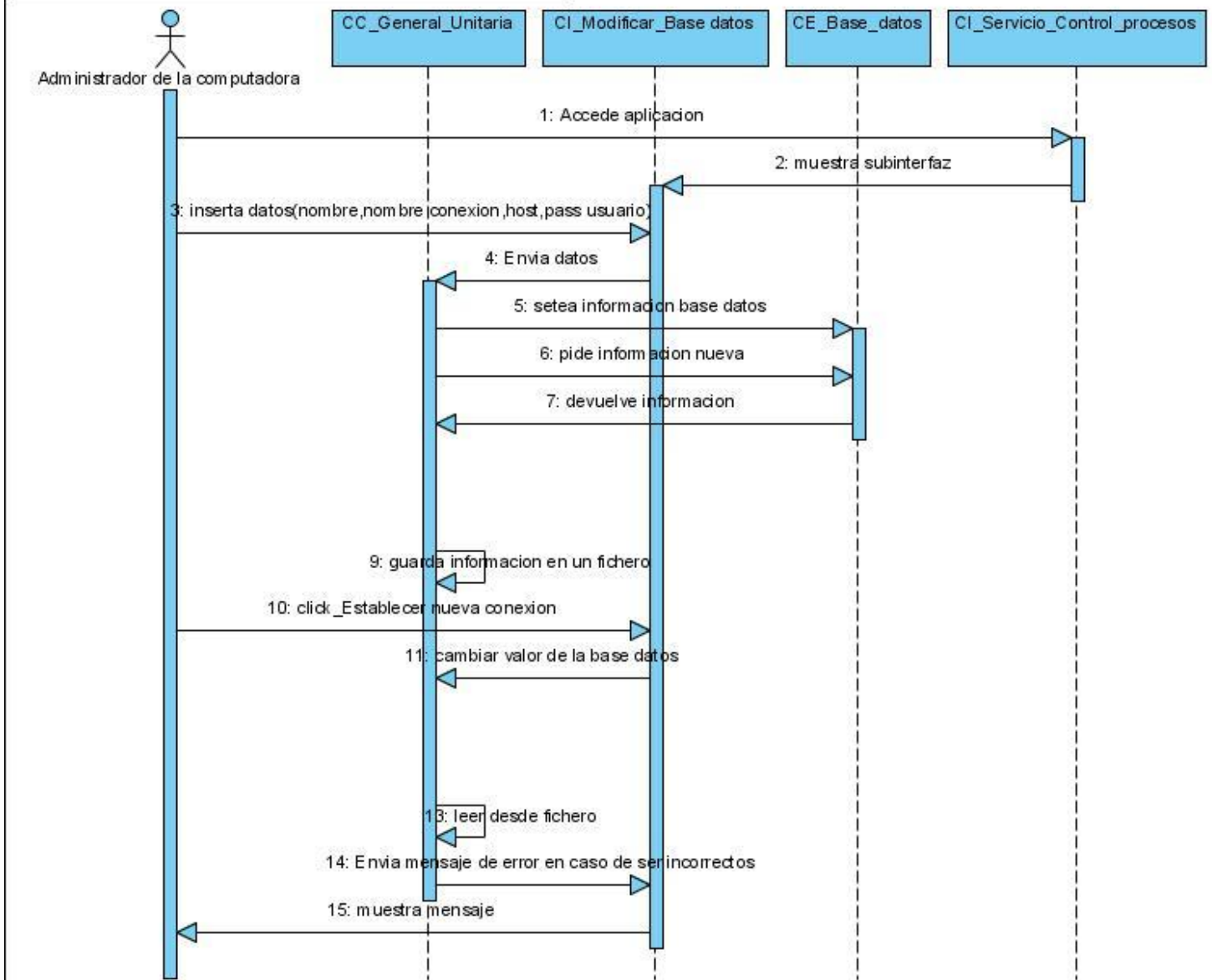




DIAGRAMA DE SECUENCIA CU_MODIFICAR_HORA_ACCESO_BASE_DATOS

sd Communication Diagram4CU_modificar_hora_acceso_base datos

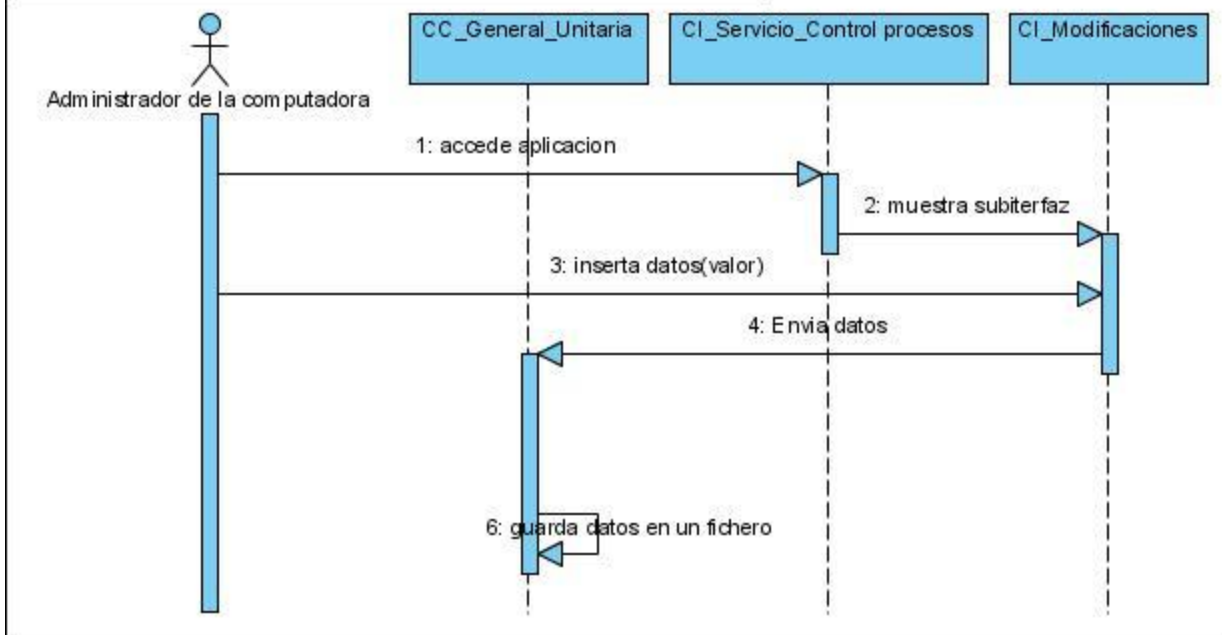


DIAGRAMA DE SECUENCIA CU_DETENER_PROCESOS_AUTOMATICAMENTE

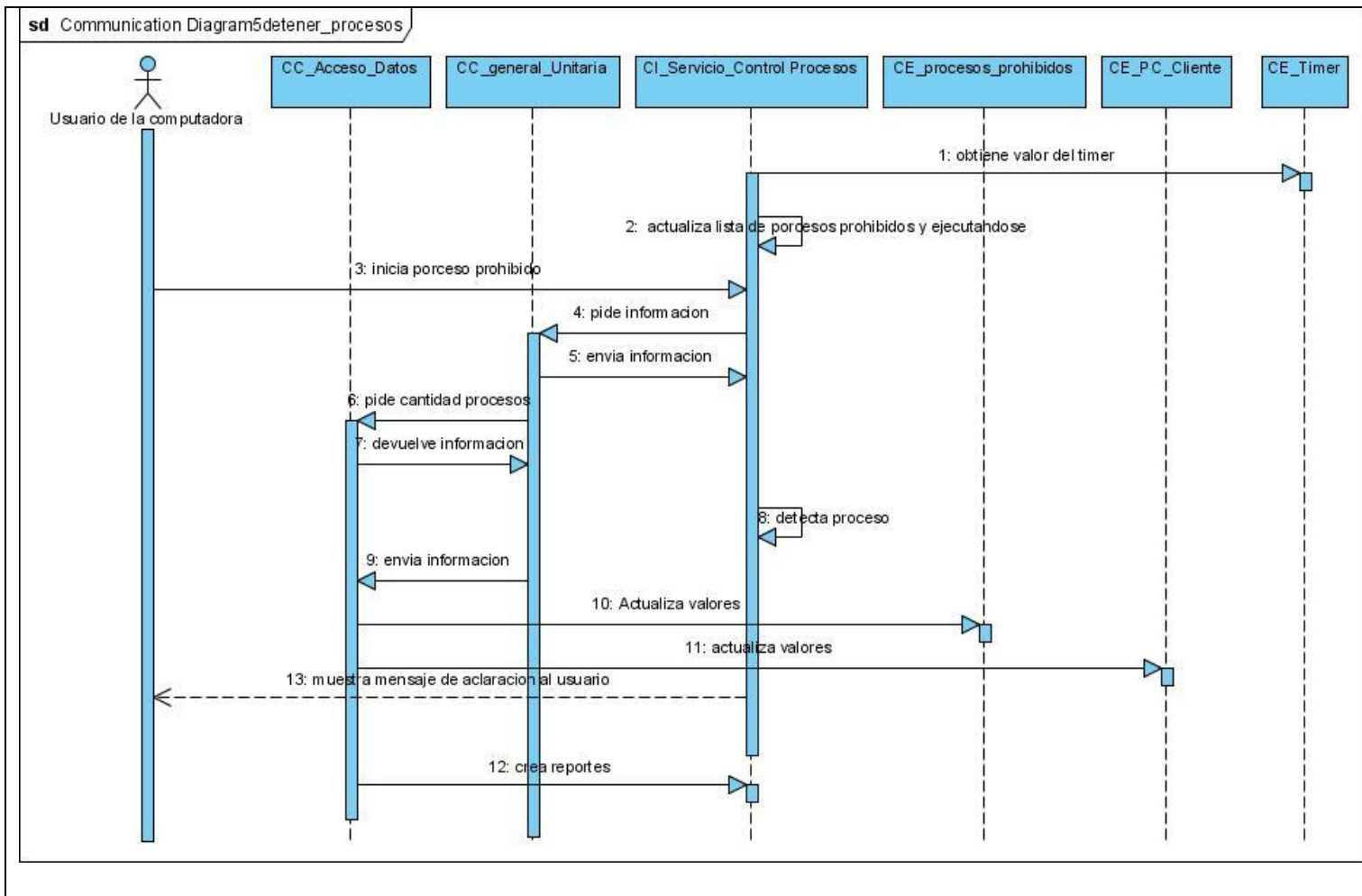
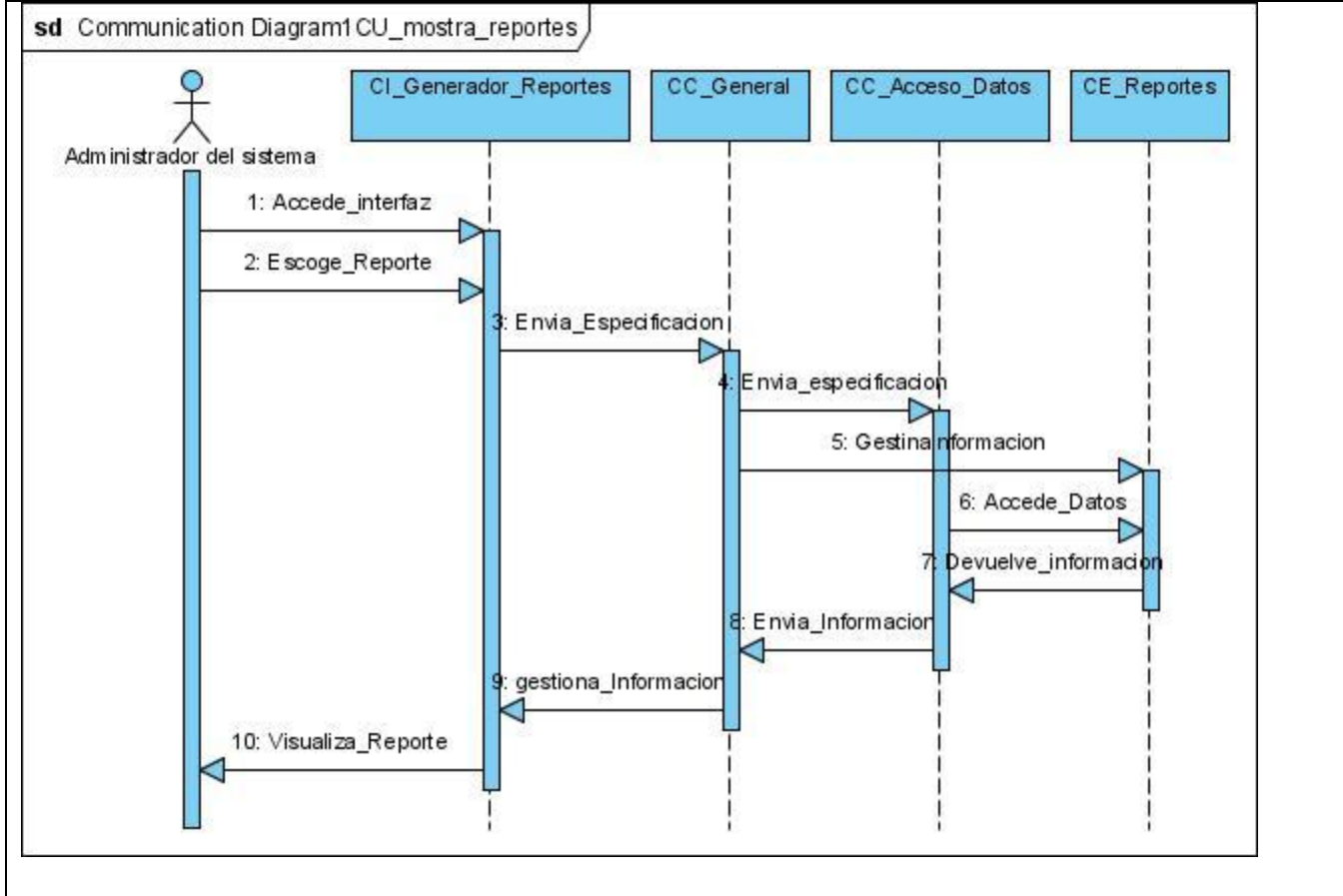




DIAGRAMA DE SECUENCIA CU_MOSTRAR_REPORTES



3.5. Diagramas de clases de Diseño

3.5.1. Estilo arquitectónico

El surgimiento de los estilos arquitectónicos ha venido a evolucionar mucho el diseño de las aplicaciones brindándoles un mejor acabado. En la actualidad son muchos los equipos de programación que construyen sus softwares basándose en cualquiera de estos. En esta oportunidad se hablará sobre el modelo de tres capas ya que pese a que no se pudo utilizar puesto que el IDE no permitía crear carpetas para definir vistas, negocio y datos, si se trató de basarse en el mismo a la hora de definir las clases y sus funcionalidades.



Arquitectura de tres capas: Las arquitecturas en capas son muy utilizadas para el desarrollo de aplicaciones hoy día por las grandes ventajas que proporcionan. El principal objetivo que persigue una arquitectura dividida en n capas es reducir dependencias entre artefactos situándolos en capas lógicas, donde cada capa depende del servicio prestado por la capa inferior y presta un servicio a la capa superior, proporcionando a los desarrolladores ventajas en cuanto al mantenimiento y reutilización de componentes o artefactos. Una arquitectura de tres capas, se puede encuentra dividida de la forma siguiente:

Capa de presentación: Esta es la capa encargada de la interacción con los diferentes tipos de usuarios, se encarga de modelar la forma en que se mostraran y se recogerán los datos entrados por los usuarios, así como de la apariencia visual que tendrá la aplicación. Se comunica con la capa de lógica de negocio a la cual envía todas las solicitudes de los usuarios y recibe las respuesta después de que hallan sido procesadas cada una de estas solicitudes.

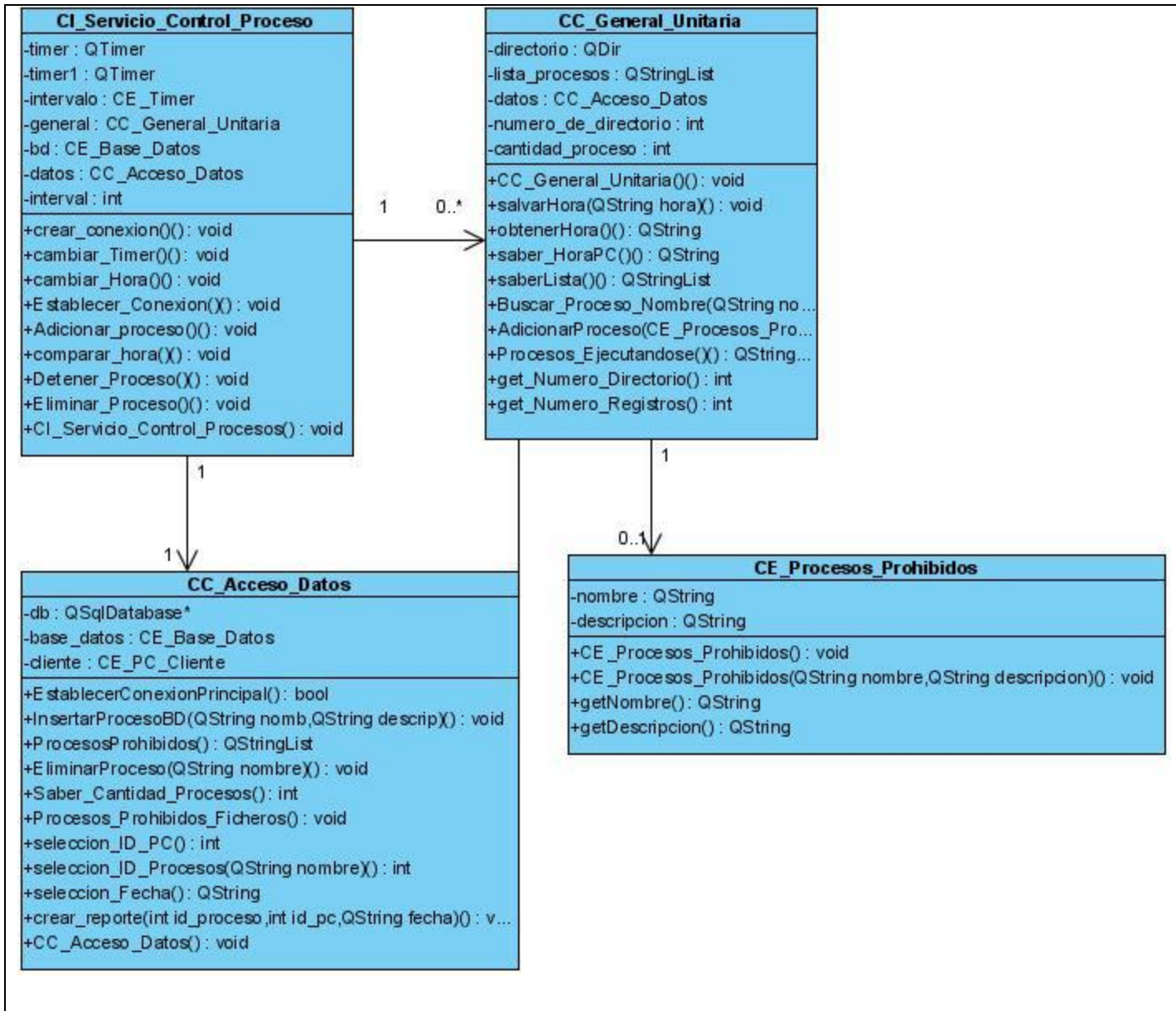
Capa de lógica de negocio: Es la encargada de modelar la lógica de negocio que dará solución a cada uno de los casos de usos de la aplicación, es en esta donde se establecen las reglas o restricciones que debe cumplir la aplicación. Se comunica con la capa de presentación para recibir las solicitudes de los usuarios y enviar las respuestas después del procesamiento.

Capa de acceso a datos: Es la encargada del manejo de datos persistentes, forma parte de la misma el servidor de base de datos como encargado de almacenamiento de datos, se comunica con la capa de lógica de negocio desde donde recibe las solicitudes de almacenar o recuperar información.

Las aplicaciones que se construyen con una arquitectura multicapa tienen entre otras las siguientes características:

3.5.2. Diagrama de clases de diseño del sistema de control de procesos

CU_ACTUALIZAR_LISTA_PROCESO



3.5.2.1. Descripción de las clases de diseño del CU_Actualizar_Lista_Proceso

Nombre: T_CE_Procesos_Prohibidos	
Tipo de clase : Entidad	
Atributo	Tipo



nombre	QString
descripción	QString
Responsabilidad	
Nombre	T_CE_Proceso_Prohibidos():void
Descripción	Constructor sin parámetros
Nombre	CE_Procesos_Prohibidos(QString nombre,QString descrip):void
Descripción	Constructor con parámetros
Nombre	getDescripcion():QString
Descripción	Retorna la descripción del proceso
Nombre	getNombre():QString
Descripción	Retorna el nombre del proceso

Nombre:CC_Accesos_Datos	
Tipo de clase: Controladora	
Atributo	Tipo
db	QSqlDatabase *
pc_cliente	CE_PC_Cliente
Base_datos	CE_Base_Datos
Responsabilidad	
Nombre	CC_Acceso_Datos():void
Descripción	constructor



nombre	EstablecerConexionPrincipal() bool.
Descripción	Crea una conexión con la base de datos
Nombre	InsertarProcesoBD(QString nomb,QString descrip);void
Descripción	Adiciona un nuevo proceso a la base de datos
Nombre	EliminarProceso(QString nomb);
Descripción	Elimina un proceso dado un nombre
Nombre	ProcesosProhibidos():QStringList
Descripción	Devuelve la lista de procesos prohibidos
Nombre	Saber_Cantidad_Procesos (): int.
Descripción	Retorna la cantidad de procesos prohibidos
Nombre	Procesos_Prohibidos_Fichero()
Descripción	Guarda la lista de procesos en un fichero
Nombre	Selección_ID_PCc():int.
Descripción	Devuelve el identificador de la PC
Nombre	Selección_ID_Proceso():int
Descripción	Devuelve el identificador del proceso
Nombre	Selección_Fecha():QString
Descripción	Devuelve la fecha
Nombre	crear_reporte(int id_proceso,int id_pc,QString fecha)
Descripción	Crea un nuevo reporte dados los parametros
Nombre: T_CC_General_Unitaria	



Tipo de clase: Controladora	
Atributo	Tipo
datos	CC_Acceso_Datos
directorio	QDir
lista_procesos	QStringList
numero_de_directorio	int
cantidad_procesos	int
Base_datos	CE_Base_Datos
Responsabilidad:	
Nombre:	T_CC_General_Unitaria():void
Descripcion:	Constructor
Nombre:	InsertarProcesos(T_CE_Procesos_prohibidos x):void
Descripcion:	Adiciona un nuevo proceso
Nombre:	salvarHora(QString hora):void
Descripción:	Guarda la hora en un fichero
Nombre:	obtenerHora():QString
Descripción	Obtiene la hora desde el fichero
Nombre:	Saber_Hora_PC():QString
Descripción	Devuelve la hora de la computadora
Nombre:	Procesos_Ejecutandose():QStringList
Descripción:	Devuelve lista procesos ejecutándose.



Nombre:	SaberLista():QStringList
Descripción	Devuelve la lista de procesos prohibidos guardados en un fichero
Nombre	getNumero_Directorio(): int.
Descripción	devuelve la cantidad de valores que tiene el directorio seleccionado
Nombre	Get_Numero_Registros()int
Descripción	Devuelve la cantidad de procesos prohibidos
Nombre	Buscar_procesos_nombre:(QString nombre)
Descripción	Funcion que verifica sin un procesos existe

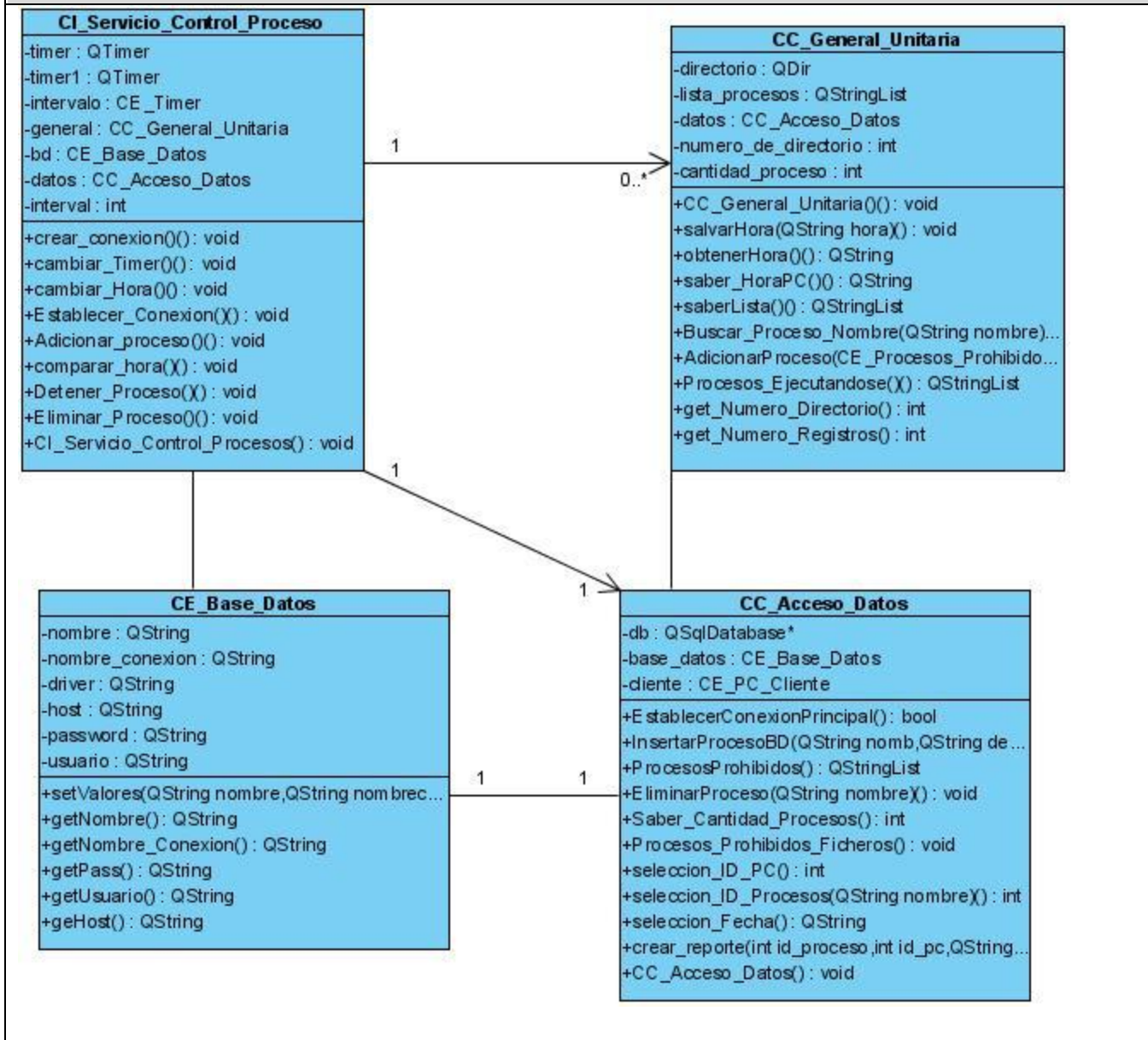
Nombre:Servicio	
Tipo de clase:Interfaz	
Atributo	Tipo
general	CC_General_Unitaria
datos	CC_Acceso_Datos
intervalo	CE_Timer
interval	CE_Base_Datos
bd	QStringList
timer	QTimer
timer1	QTimer
Responsabilidad	
Nombre	Cambiar_Timer():void



Descripción	Cambia el intervalo del timer
Nombre	Cambiar_Hora():void
descripción	Cambia el valor de la hora de acceso a la base de datos
Nombre	Eliminar_Proceso():void
Descripción	Elimina un procesos de la base de datos
Nombre	Detener_Procesos():void
Descripción	Obtiene las listas de procesos ,las compara y en caso de existir algún proceso prohibido ejecutándose ,lo elimina y envía mensaje de aclaración
Nombre	Adicionar_proceso():void
Descripción	Adiciona un nuevo procesos ala base de datos
Nombre	Establecer_conexion():void
Descripción	Crear una nueva conexión.
nombre	Servicio():void
Descripción	constructor
Nombre	Comparar_hora():void
Descripción	Compara la hora de la PC con la hora de accesos a la base de datos



CU_MODIFICAR_CONEXION_BASE_DATOS



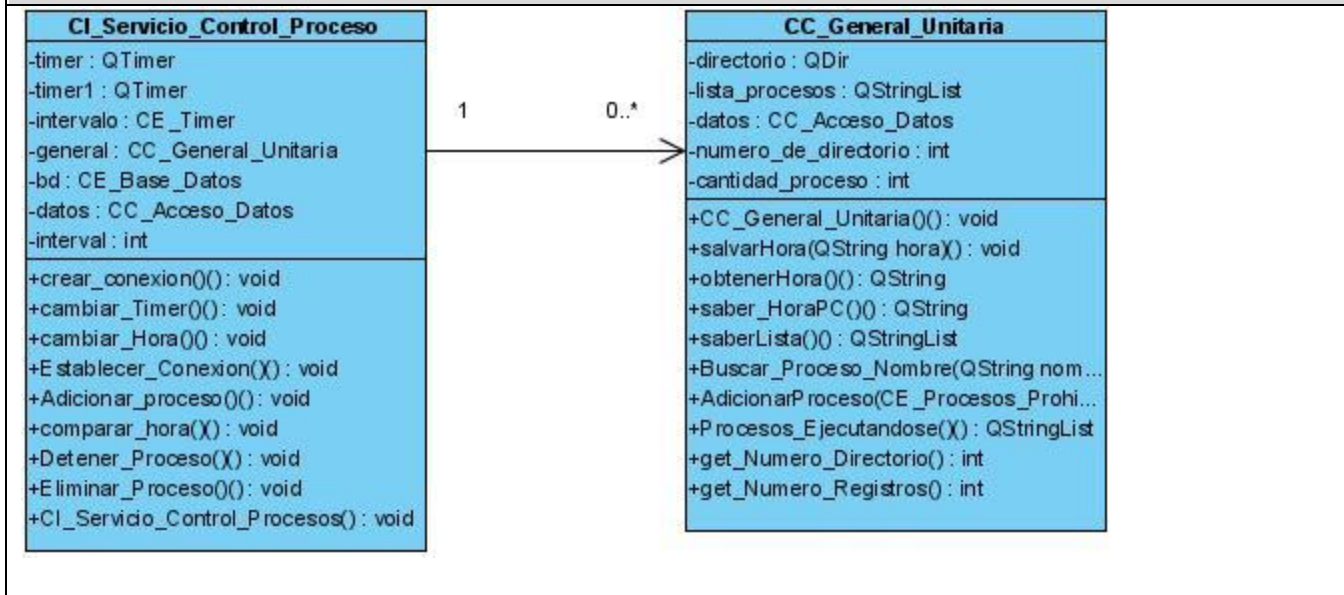
3.5.2.2. Descripción de las clases de diseño del CU_Modificar_Conrxion_Base_Datos



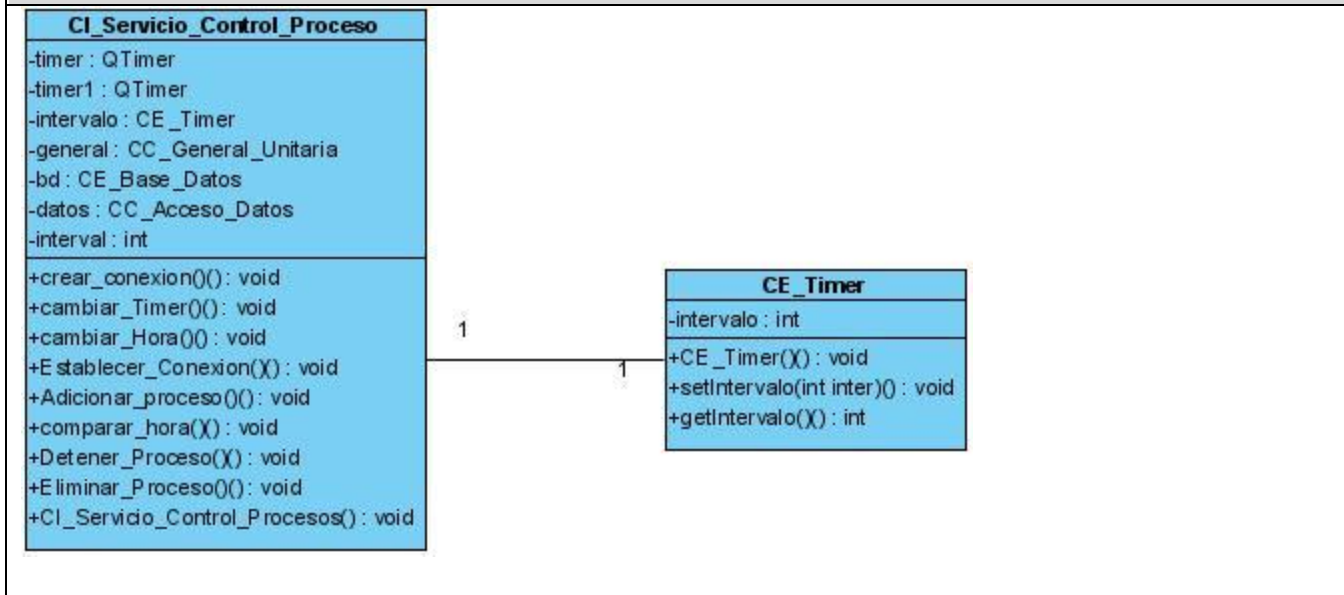
Nombre:CE_Base_Datos	
Tipo de clase:Entidad	
Atributo	Tipo
nombre	string
usuario	string
contraseña	string
hostname	string
driver	string
nombre _ conexión	string
Responsabilidad	
Nombre	CE_Base_Datos(nc, n, u,p,h):void
Descripción	Constructor.
Nombre	get_..... :Tipo de dato de retorno
Descripción	Las propiedades de actualización y retorno de cada atributo. El nombre se corresponderá con el atributo. Tipo de dato de retorno será el tipo de dato del atributo.
Nombre	Set_.....(nd):void
Descripción	Las propiedades de actualización de cada atributo



CU_MODIFICAR_HORA_ACCESO_BASE_DATOS



CU_MODIFICAR_TIMER

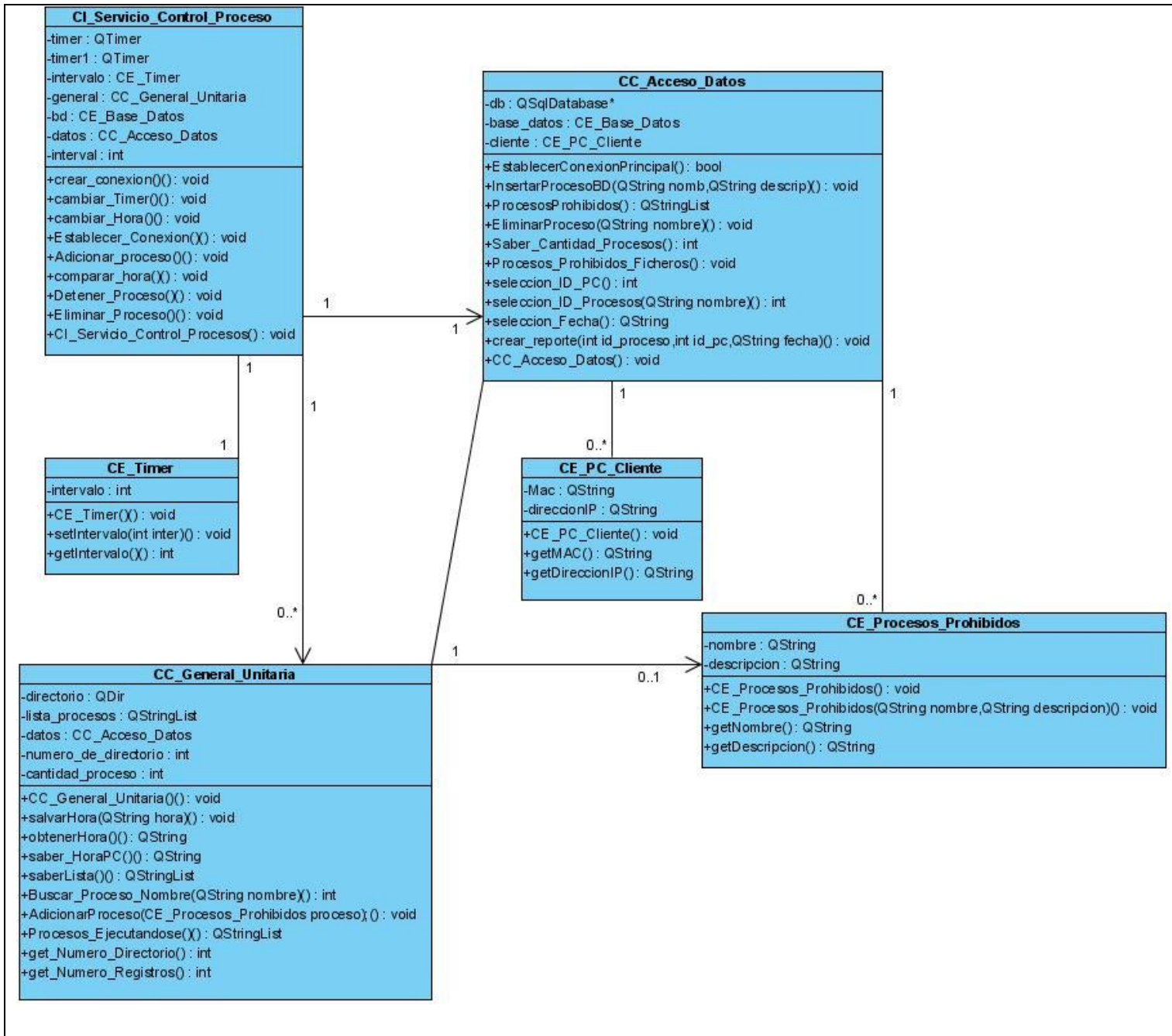


3.5.2.3. Descripción de las clases de diseño del CU_Modificar_Timer



Nombre:CE_Timer	
Tipo de clase :Entidad	
Atributo	Tipo
intervalo	int
Responsabilidad	
Nombre	CE_Timer(intervalo):void
Descripción	constructor
Nombre	Get_intervalo():int
Descripción	Retorna el intervalo
Nombre	Set_intervalo(nd):void
Descripción	Actualiza el nuevo valor del intervalo

CU_DETENER_PROCESOS_AUTOMATICAMENTE



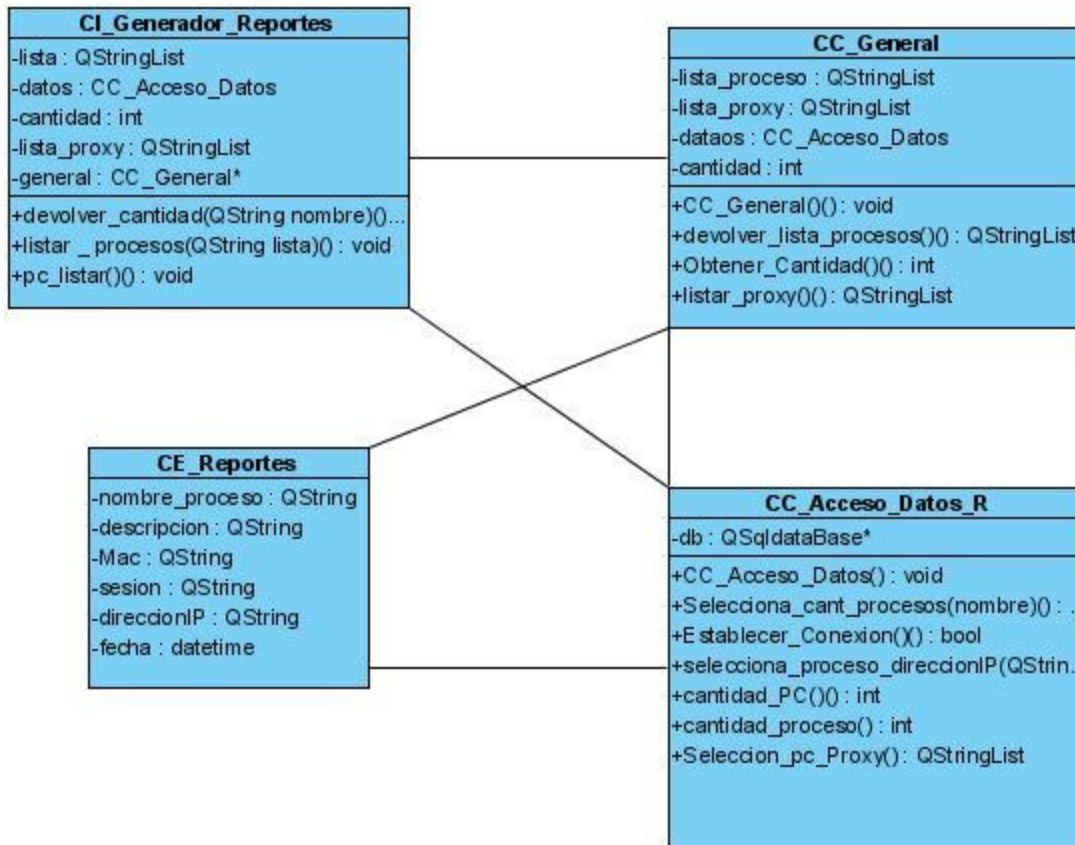
3.5.2.4 Descripción de las clases de diseño del CU_Detener_Procesos_Automaticamente

Nombre: T_CE_PC_Cliente
Tipo de clase: Entidad



Atributo		Tipo
direccionIP		QString
MAC		QString
Responsabilidad		
Nombre	T_CE_PC_Cliente():void	
Descripción	Constructor sin parámetros	
Nombre	T_CE_PC_Cliente (string s,string dip, string Mac):void	
Descripción	Constructor por parámetros.	
Nombre	get_..... :Tipo de dato de retorno	
Descripción	Las propiedades de actualización y retorno de cada atributo. El nombre se corresponderá con el atributo. Tipo de dato de retorno será el tipo de dato del atributo.	

3.6. Diagrama de clases de diseño del sistema de manejo de reportes



3.6.1. Descripción de las clases de diseño del sistema de gestión de reportes.

Nombre:Reportes	
Tipo de clase:Entidad	
Atributo	Tipo
Nombre _ proceso	string
descripcion	string
direccionIP	string
MAC	string
hora	time
fecha	datetime



Responsabilidad	
Nombre	CE_Reportes(np,u,f,h,mac,ip):void
Descripción	Constructor.
Nombre	get_..... :Tipo de dato de retorno
Descripción	Las propiedades de actualización y retorno de cada atributo. El nombre se corresponderá con el atributo. Tipo de dato de retorno será el tipo de dato del atributo.

Nombre:CC_Accesos_Datos	
Tipo de clase: Controladora	
Atributo	Tipo
db	QSqlDatabase *
Responsabilidad	
Nombre	CC_Acceso_Datos():void
Descripción	constructor
nombre	Establecer _Conexión () bool.
Descripción	Crea una conexión con la base de datos
Nombre	Selecciona_cant_procesos (QString proceso): int.
Descripción	Devuelve la cantidad de procesos con el mismo nombre
Nombre	Selección_procesos_direccionIP(QString dir):QStringList
Descripción	Devuelve la lista de procesos detenidos en una dirección IP dada



Nombre	cantidad _ pc ():int.
Descripción	Devuelve la cantidad de veces que esta una dirección IP en la base datos
Nombre	cantidad _ proceso (): int.
Descripción	Devuelve la cantidad de veces que esta un proceso en la base datos
Nombre	Seleccion_pc_proxy ():QStringList
Descripción	Devuelve las direcciones IP de la PC con esta función

Nombre: CC_General	
Tipo de clase: Controladora	
Atributo	Tipo
lista _ proceso	QStringList
lista_proxy	QStringList
cantidad	int.
datos	CC_Acceso_Datos
Responsabilidad	
Nombre	CC_General():void
Descripción	constructor
Nombre	devolver_lista_procesos(QString lista):QStringList
Descripción	Llama al método que se encarga de devolver la lista de procesos en la base de datos
Nombre	Obtener_cantidad(QString nomb):int
Descripción	Devuelve cantidad de procesos



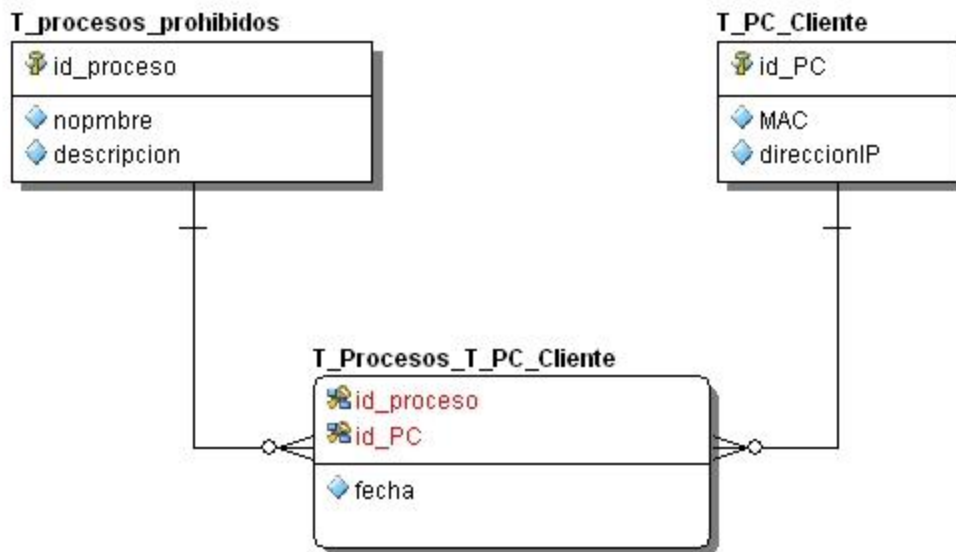
Nombre	lista_proxy():QStringList
Descripción	Devuelve la lista de direcciones IP en función Proxy

Nombre: Reportes	
Tipo de clase: Interfaz	
Atributo	Tipo
Nombre_proceso	QString
descripción_proceso	QString
direccionIP	QString
MAC	QString
sesión	QString
fecha	DateTime
Responsabilidad	
Nombre	CE_Reportes()
Descripción	constructor
Nombre	CE_Reportes(nombre,nac,sesión,ip,fecha,descrip)
Descripción	Muestra la lista de procesos detenidos en una PC
Nombre	Get.....:tipo de dato de retorno
Descripción	Devuelve el valor de todas las funciones que sean atributos

3.7. Diseño de la base de datos



El diagrama entidad relación fue realizado en Edwin Studio 7.0, herramienta que nos permite modelar de forma concisa clara la base de datos que vamos a utilizar ,además de mostrarnos el modelo lógico y físico del mismo y permitirnos generar el mismo para poder utilizar sus ventajas. Este diagrama es útil para ambos sistemas.



3.8.1. Descripción de las tablas de la Base de Datos

Nombre:T_Procesos		
Descripción: Almacena el nombre de todos los procesos prohibidos.		
Atributo	Tipo	Descripción
Id	Integer	Identificador del procesos prohibido
nombre	Varchar(50)	Nombre del proceso
descripcion	Varchar(50)	Descripción física del proceso



Nombre:T_Proceso_T_PC_Cliente		
Descripción:Almacenara los reportes generados tras detener un proceso prohibido		
Atributo	Tipo	Descripcion
Id_proceso	integer.	Identificador del proceso
Id_PC	integer.	Identificador de la PC
fecha	TIMESTAMP/DATE	Fecha en la que se detuvo el proceso

T_PC_Cliente		
Descripción:Almacenara los datos la computadora que levante un reporte		
Atributo	Tipo	Descripción
id_PC	integer	Identificador de la PC
MAC	Varchar(50)	Dirección física de la PC
direccionIP	Varchar(50)	Dirección IP de la PC

3.9. Conclusiones

En este capitulo se mostraron diversos diagramas que se generan en los flujos de análisis y diseño y que son de vital importancia para tener una opinión relativamente cercana de la aplicación que



vamos a desarrollar. También se realizó una representación gráfica de la base de datos con la que se va a trabajar. Siempre en pos de alcanzar con cada diseño un rango de calidad lo mejor posible.



CAPÍTULO 4: IMPLMETACION Y PRUEBAS DEL SISTEMA

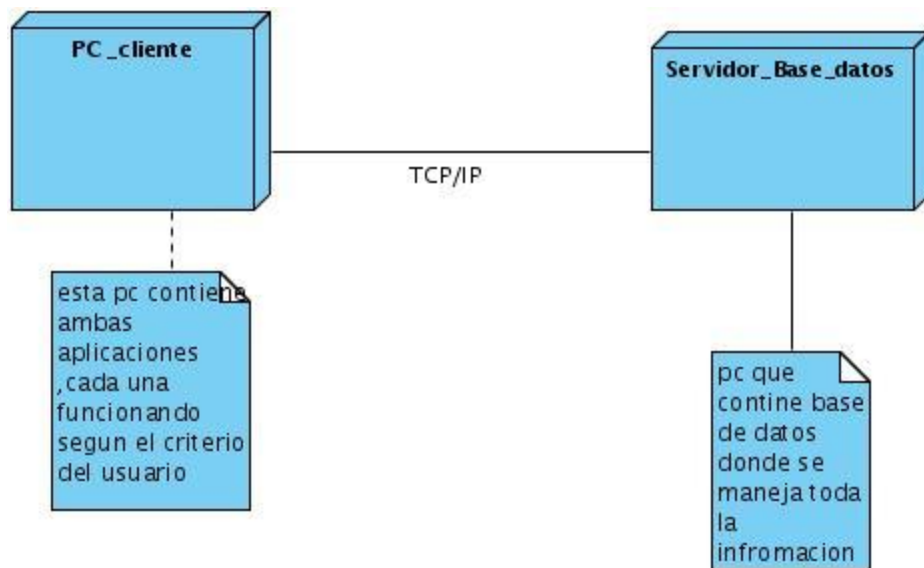
4.1. Introducción.

Luego de realizados un análisis y diseño del software que nos permita tener una idea de lo que se espera y desea del software se llega a este flujo que deviene una parte fundamental para el desarrollo del mismo .En este capitulo se representan los diagramas de despliegue y componentes, además de reflejar el posible cumplimiento de los requisitos mediante las pruebas.

4.2. Implementación

En este flujo se implementa el sistema en términos de componentes, es decir, ficheros de código fuente, scripts, ficheros de código binario, ejecutables y similares., y cómo estos se organizan de acuerdo a los nodos específicos en el modelo de despliegue.

4.2.1. Diagrama de despliegue



4.2.2. Diagrama de componentes

Los diagramas de componentes tienen entre sus principales características la de mostrar los diversos tipos de realización, que incluyen código fuente, binario y ejecutable, además de describir los diversos elementos físicos del sistema así como sus relaciones.

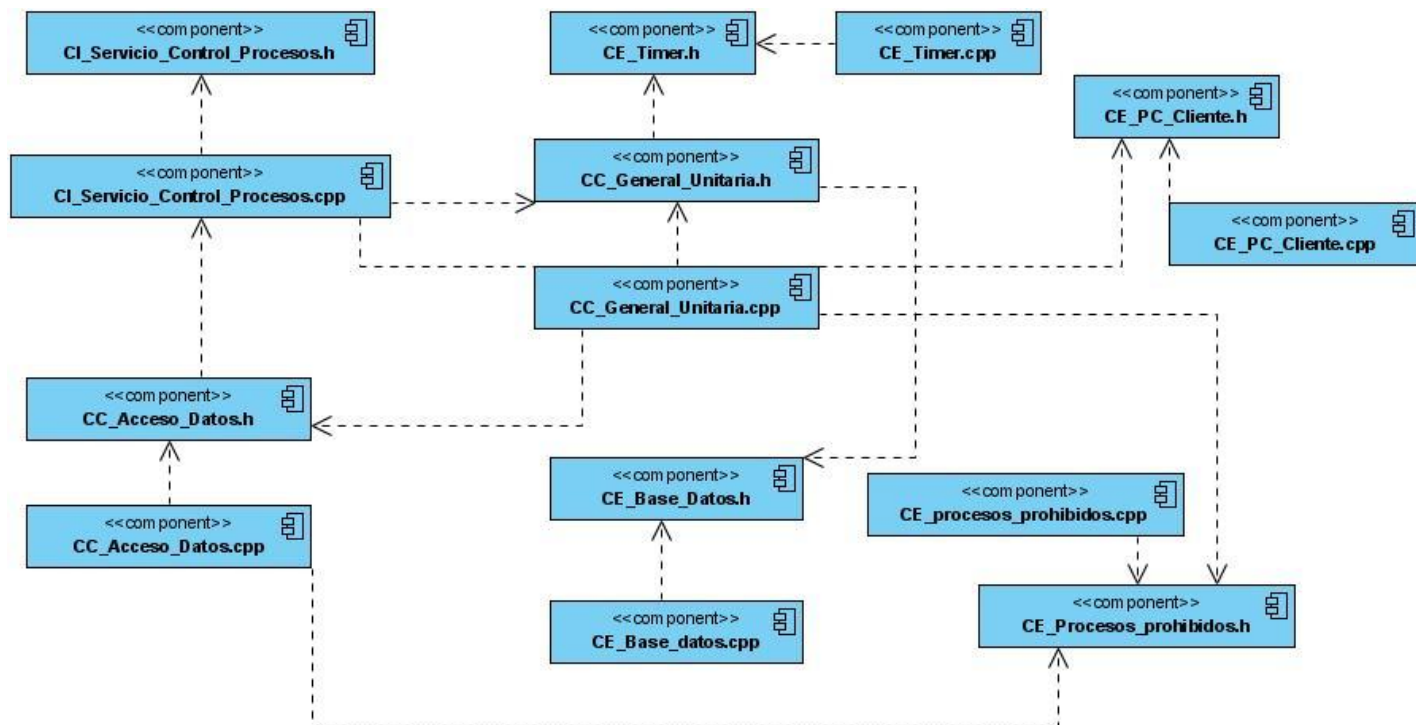


Se denomina componente al empaquetamiento físico de los elementos de un modelo, como son las clases en el modelo de diseño. Algunos estereotipos estándar de componentes son los siguientes:

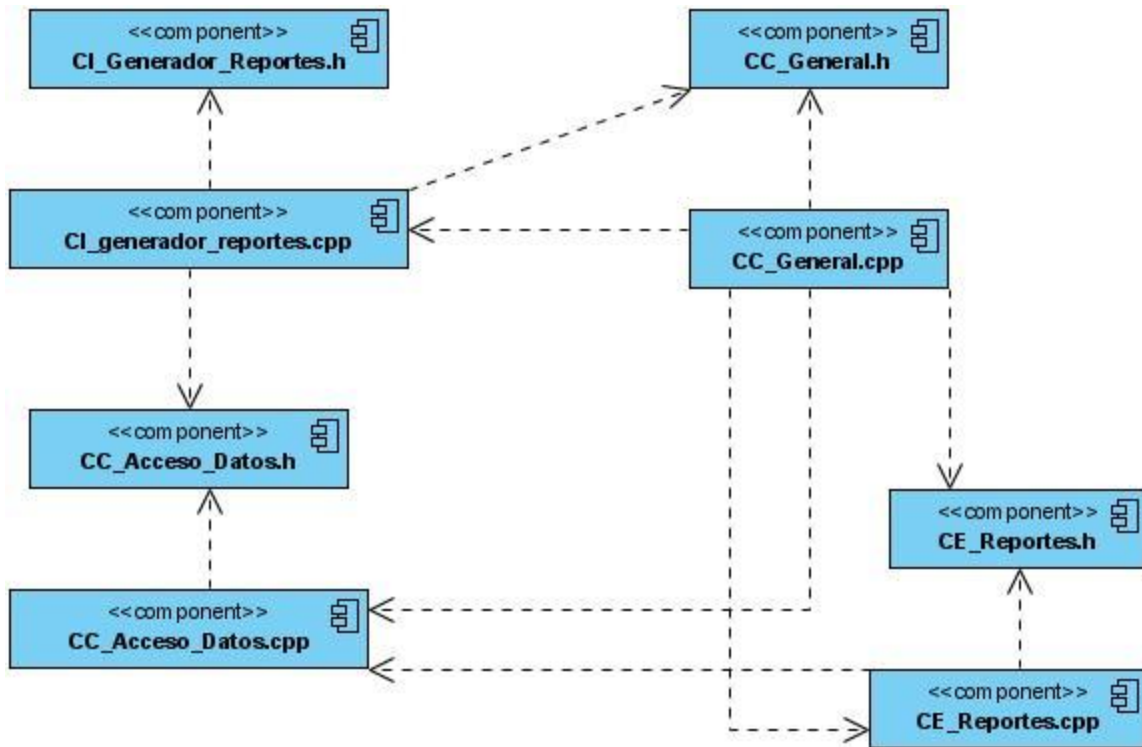
- <<executable>> es un programa que puede ser ejecutado en un nodo.
- <<file>> es un fichero que contiene código fuente o datos.
- <<library>> es una librería estática o dinámica.
- <<table>> es una tabla de base de datos.
- <<document>> es un documento.

Algunos componentes existen en tiempo de compilación, algunos en tiempo de enlace y algunos en tiempo de ejecución, otros en varias de estas. En nuestro caso el entorno de desarrollo no brinda la posibilidad de definir paquetes por lo que se realizara un diagrama de componentes general.

4.2.2.1. Diagrama de componentes del sistema de control de procesos



4.2.2.2. Diagrama de componentes del sistema de gestión de reportes



4.3. Modelo de pruebas

Las pruebas son la parte final del software pues garantizan la calidad del mismo y se encargan de verificar el cumplimiento de los requisitos que se plantearon anteriormente. Las pruebas fundamentales son denominadas pruebas de caja blanca y caja negra.

En este caso utilizaremos el tipo de prueba de caja negra.

4.3.1. Casos de prueba del sistema de Control de procesos

Caso Prueba: CU_Actualizar_Lista_Procesos

Entrada	Resultados	Condiciones
El administrador del sistema inserta el nombre y la descripción del proceso	El Sistema inserta los datos satisfactoriamente	El proceso no ha sido añadido con anterioridad



Entrada	Resultados	Condiciones
El administrador del sistema inserta el nombre y la descripción del proceso	El Sistema muestra el mensaje "El proceso" ya existe	Como el proceso ha sido añadido con anterioridad, el sistema debe mostrar un mensaje de error.

Entrada	Resultados	Condiciones
El administrador del sistema inserta el nombre del proceso que desea eliminar	El sistema actualiza los datos en la base de datos	El proceso se encuentra en la lista de procesos de la base de datos

Entrada	Resultados	Condiciones
El administrador del sistema inserta el nombre del proceso que desea eliminar	El sistema muestra mensaje "El proceso no existe"	El proceso no ha sido añadido con anterioridad y debe lanzar un mensaje de error.

Entrada	Resultados	Condiciones
El administrador pulsa el botón "OK"	El sistema muestra el mensaje "no se deben dejar campos vacíos"	El usuario debe dejar los campos vacíos.

Caso de prueba: Modificar_Conexion_Base_Datos



Entrada	Resultados	Condiciones
El administrador llena los campos nombre, nombre de la conexión, usuario, servidor, contraseña y pulsa el botón "Crear".	El sistema actualiza la información satisfactoriamente	El usuario inserto todos los campos correctamente

Entrada	Resultados	Condiciones
El administrador pulsa el botón "Crear"	El sistema muestra el mensaje "no se deben dejar campos vacíos"	El usuario debe dejar los campos vacíos.

Entrada	Resultados	Condiciones
El administrador pulsa el botón "Crear" después de haber insertado todos los datos excepto el nombre.	El sistema muestra el mensaje "no se deben dejar campos vacíos"	El usuario dejo el campo nombre vacío.

Entrada	Resultados	Condiciones
El administrador pulsa el botón "Crear" después de haber insertado todos los datos excepto el nombre conexión.	El sistema muestra el mensaje "no se deben dejar campos vacíos"	El usuario dejo el campo nombre conexión vacío.



Entrada	Resultados	Condiciones
El administrador pulsa el botón "Crear" después de haber insertado todos los datos excepto el hostname.	El sistema muestra el mensaje "no se deben dejar campos vacíos"	El usuario dejó el campo hostname vacío.

Entrada	Resultados	Condiciones
El administrador pulsa el botón "Crear" después de haber insertado todos los datos excepto el password.	El sistema muestra el mensaje "no se deben dejar campos vacíos"	El usuario dejó el campo password vacío.

Entrada	Resultados	Condiciones
El administrador pulsa el botón "Crear" después de haber insertado todos los datos excepto el usuario.	El sistema muestra el mensaje "no se deben dejar campos vacíos"	El usuario dejó el campo usuario vacío.

Entrada	Resultados	Condiciones
El administrador pulsa el botón "Crear Nueva Conexión".	El sistema muestra el mensaje "No se pudo establecer nueva conexión"	El usuario inserto algún dato erróneo



Entrada	Resultados	Condiciones
El administrador pulsa el botón "Crear Nueva Conexión".	El sistema actualiza los datos satisfactoriamente	El usuario inserto todos los datos para establecer la conexión correctamente.

Caso de prueba CU_Modificar_Timer

Entrada	Resultados	Condiciones
El administrador escoge el nuevo intervalo y pulsa el botón "Aceptar"	El sistema guarda la información	El usuario escogió un intervalo correcto

Entrada	Resultados	Condiciones
El administrador pulsa el botón "Aceptar"	El sistema muestra el mensaje "Por favor ,escoja un valor"	El usuario no escogió un valor.

Caso de prueba CU:Modificar_Hora_Acceso_Base_Datos

Entrada	Resultados	Condiciones
El administrador escoge la nueva hora de actualización y pulsa el botón "Aceptar"	El sistema guarda la información	El usuario escogió una hora correctamente.



Entrada	Resultados	Condiciones
El administrador pulsa el botón "Aceptar"	El sistema muestra el mensaje "Por favor ,escoja un valor"	El usuario no escogió una nueva hora de actualización.

4.3.2.Casos de prueba del sistema de gestion de reportes

Caso de prueba :CU_Mostrar_Reportes

Entrada	Resultados	Condiciones
El administrador pulsa el botón "Devolver cantidad de procesos"	El sistema muestra el mensaje "Por favor ,inserte un proceso"	El administrador no insertó un proceso

Entrada	Resultados	Condiciones
El administrador pulsa el botón "Devolver cantidad de procesos"	El sistema muestra el mensaje "El proceso no existe"	El administrador insertó un proceso que no se encontraba en la base de datos.

Entrada	Resultados	Condiciones
El administrador pulsa el botón "Devolver cantidad de procesos"	El sistema muestra la información pedida	El administrador insertó correctamente los datos

Entrada	Resultados	Condiciones
El administrador pulsa el	El sistema muestra el	El usuario no escogió una dirección



botón “Devolver procesos detenidos de una PC”	mensaje “Por favor ,inserte una dirección”	IP
---	--	----

Entrada	Resultados	Condiciones
El administrador pulsa el botón “Devolver procesos detenidos de una PC”	El sistema muestra el mensaje “La dirección no existe”	El administrador insertó una dirección que no se encontraba en la base de datos.

Entrada	Resultados	Condiciones
El administrador pulsa el botón “Devolver procesos detenidos de una PC”	El sistema muestra la información pedida	El administrador insertó correctamente los datos

4.4.Conclusiones

Con este capitulo se culminan los principales flujos a tener en cuenta para desarrollar un sistema .En el mismo quedaron reflejados los diagramas de despligue y componentes que reflejan como deben funcionar desde un punto de vista fisico y de soporte el software ,ademas demuetra mediante las pruebas los posibles detalles en los que el software puede fallar,que es la principal importancia del flujo de pruebas.



CONCLUSIONES

Con la culminación de este trabajo se ha logrado realizar un sistema que permite a los especialistas del Departamento de Seguridad Informática y a la dirección de los laboratorios de la Universidad de la Ciencias Informáticas, obtener reportes sobre los procesos que se ejecutan y detienen dada una determinada condición en las computadoras de los laboratorios de docencia y a la vez evitar el mal aprovechamiento de los recursos que brindan lo ordenadores, respectivamente.

El Sistema de Control de los Procesos en los Ordenadores, brinda diversas funcionalidades. Con él se pueden detener procesos automáticamente que sean iniciados por los usuarios de la computadora siempre y cuando se consideren prohibidos. Estos procesos se encontrarán en una base de datos a la cual el sistema accederá con el fin de actualizar su lista. También permite actualizar la lista de estos procesos, cambiar la hora de acceso a la base de datos, cambiar el intervalo del timer y los parámetros de conexión a la misma. También generará reportes sobre cada proceso detenido.

La implantación de este sistema debe obtener como resultado una disminución del robo de las contraseñas y en especial un mejor aprovechamiento de los recursos que brinda la computadora. Además de evitar violaciones al código de ética.

Se destacará además que pese a que el sistema está diseñado para ser usado en la Universidad de la Ciencias Informáticas, por sus características puede ser aplicable a cualquier otra organización siempre y cuando se cumplan los requerimientos planteados.



RECOMENDACIONES

Los procesos representan una parte fundamental en el funcionamiento de la PC, pues definen las actividades que se están realizando en la misma a fin de poder iniciarlos, detenerlos, cambiarles la prioridad, etc. Con este sistema se cumplen algunas de estas características de forma automática evitando que el cliente deba acceder al administrador de tareas para realizar estos cambios y permitiendo además que se realicen reportes en pos de saber las estadísticas generadas por el mismo.

Aunque se cumplieron todos los objetivos iniciales propuestos para próximas versiones del mismo se recomienda: que los reportes muestren además el usuario que generó el reporte, que el software permita autenticarse para evitar las posibles violaciones de la contraseña del root, que se muestren los reportes de forma gráfica para un mejor entendimiento del cliente y que cada determinado tiempo se vacíe la base de datos.



BIBLIOGRAFIA

SOM, G. Programación en Linux con KDevelop 1.0. [1999], nº Disponible en: <http://www.elguille.info/linux/KDEV02.HTM>.

JIMENEZ, A. L. El tutorial del desarrollo visual. 2003, nº Disponible en: <http://developer.kde.org/~larrosa/es/visualtutorial/index.html>.

JAIME. Sistema de Archivos Virtual /proc de Linux. 2008, nº Disponible en: <http://www.monografias.com/trabajos50/archivos-virtual-linux/archivos-virtual-linux.shtml>.

DURAN, S. G. MANUAL BÁSICO DE CRON. nº Disponible en: http://www.linuxtotal.com.mx/index.php?cont=info_admon_006.

C. PAREJA, M. O., Á. L. ANDEYRO, C. ROSSI. Desarrollo de algoritmos y técnicas de programación en Pascal 1997, Disponible en: <http://www.mygnet.net/manuales/>. ISBN 84-7897-290-0.

MARK MITCHELL, J. O., ALEX SAMUEL. Advanced Linux Programming. 2001, Disponible en: <http://www.advancedlinuxprogramming.com/about.html>. ISBN 0-7357-1043-0.

MARTINEZ, R. El rincón de Linux 1998-2008, nº Disponible en: <http://www.linux-es.org/>.

RODRÍGUEZ, C. L. Ejemplo de desarrollo software utilizando la metodología RUP. 2003, nº Disponible en: <http://www.dsic.upv.es/asignaturas/facultad/lsi/ejemplorup/>.

SANCHEZ, M. A. M. Metodologías De Desarrollo De Software 2004, nº Disponible en: http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html.

MAHFOUF, A.-M. Qt Designer and KDevelop-3.0 for beginners. 2004, nº Disponible en: <http://women.kde.org/articles/tutorials/kdevelop3/>.

LABRADOR, R. M. G. PROGRAMACIÓN DE SISTEMAS UNIX: COMUNICACIÓN ENTRE PROCESOS. 1996, nº Disponible en: <http://es.tldp.org/Universitarios/seminario-1.html>.



BIBLIOGRAFIA REFERENCIADA

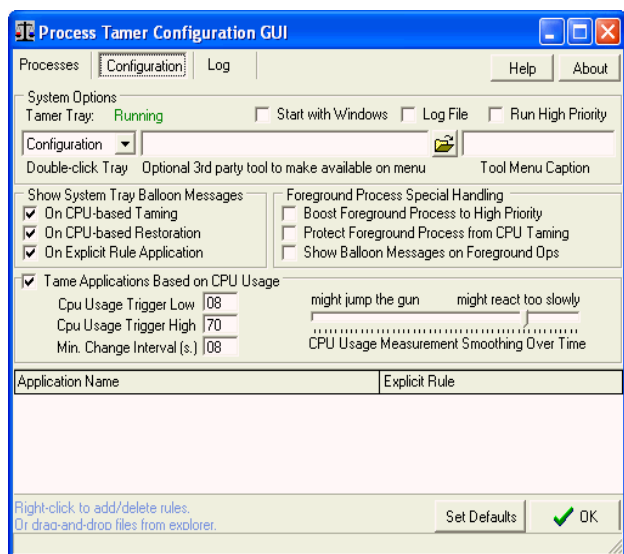
1. VILLANUEVA, P. Windows vs. Linux. [2007], nº Disponible en: <http://www.rinconsolidario.org/linux/win-Lin/win-Lin.html>.
3. RODRÍGUEZ, S. C. *Concepto de proceso*. Disponible en: <http://www.rastersoft.com/OS2/CURSO/PROCESO.HTM>.
5. ---. KDE. [2008], nº Disponible en: <http://es.wikipedia.org/KDE>.
6. ---. GNOME. 2008, nº Disponible en: <http://es.wikipedia.org/GNOME>.
7. WEB. Process Tamer. 2006, nº Disponible en: <http://www.webintenta.com/process-tamer-gestiona-los-procesos-activos.html>.
8. ---. Process Tamer. 2007, nº Disponible en: <http://www.donationcoder.com/Software/Mouser/proctamer/>.
9. ---. RunAlyzer. 2007, nº Disponible en: <http://biboz.net/programas/runalyzer-071>.
10. ---. Faronics-anti-executable-v2.30.000.317. 2008, nº Disponible en: <http://www.x-caleta.com/2008/05/25/faronics-anti-executable-v2.30.000.317.html>.
11. BONET, J. Administrador de tareas con identificador de procesos? [2007], nº Disponible en: <http://kill-process.softonic.com/>.
12. MONOGRAFIAS.COM. Base de datos. 1997, nº Disponible en: <http://www.monografias.com/trabajos11/basda/basda.shtml>.
13. CARBALLEIRA, D. F. G. Sistema gestores de base de datos. nº Disponible en: http://www.error500.net/garbagecollector/archives/categorias/bases_de_datos/sistema_gestor_de_base_de_datos_sgbid.php.
14. LEDERKREMER., M. *Linux 11-Windows Programación visual*. [2005], Disponible en: <http://www.mononeurona.org/archivos/lrx11b.pdf> ISBN 987-526-297-8.
15. DESCONOCIDO. GCC. 2006, nº Disponible en: <http://www.boonic.com/ftopic66000.html>.



16. DAVIDSON, S. R. Librería time ANSI C. 2001, n° Disponible en:
<http://www.conclase.net/c/librerias/libreria.php?lib=time>.
17. APRENDIZAJE, E. V. Conf._2_Modelación _ negocio [1]. [2008], n° Disponible en:
<http://teleformacion.uci.cu/mod/resource/view.php?id=6680>.
18. M.SC. ING. FEBE ÁNGEL CIUDAD RICARDO, I. Y. H. M. DoMet COMO PROPUESTA PARA LA MODELACIÓN DE ENTORNOS ORGANIZACIONALES COMPLEJOS Y DIFUSOS. [2007], n°



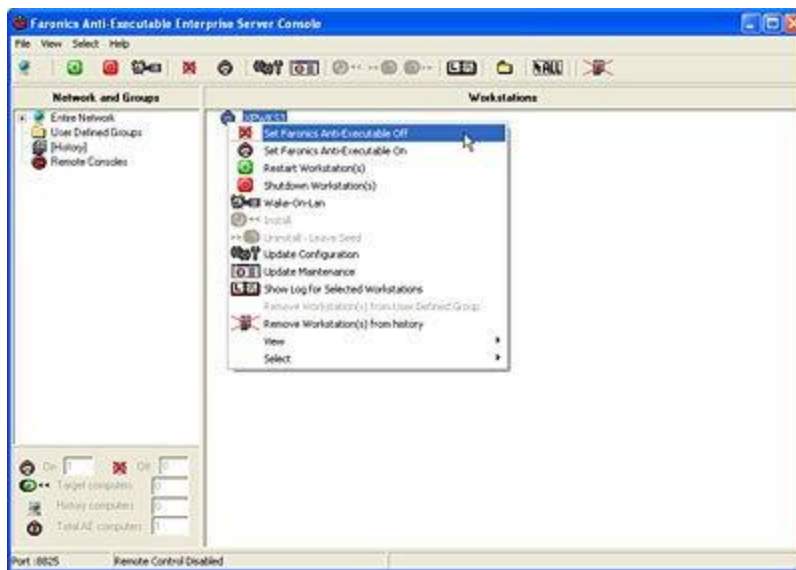
ANEXOS



Anexo1-Imagen del software :Process Tamer



Anexo2-Imagen del software: Process Control Task Manager 2.0



Anexo3- Imagen del software Faronics Anti-Executable



GLOSARIO DE TERMINOS

A:

Automake: GNU Automake es una herramienta de programación que produce programas makefiles portables para el uso de make usado en la compilación de software

B:

Backend: Programa que procesa la información. Rara vez, un programa de tipo backend, interactúa con el usuario. Puede ser usado por el usuario de forma directa, sin embargo, los backend suelen estar complementado con frontend, para facilitar al usuario el manejo del programa.

Binding: En informática, un binding es una "ligadura" o referencia a otro símbolo más largo y complicado, y que se usa frecuentemente. Este otro símbolo puede ser un valor de cualquier tipo, numérico, de cadena, etc., o el nombre de una variable que contiene un valor o un conjunto de valores.

Breakpoint: es una facilidad que tienen los IDEs de desarrollo que permiten detener la ejecución de una aplicación en es punto.

BSD:BSD son las iniciales de Berkeley Software Distribution (en español, Distribución de Software Berkeley) y se utiliza para identificar un sistema operativo derivado del sistema Unix nacido a partir de las aportaciones realizadas a ese sistema por la Universidad de California en Berkeley.

C:

CPU: La unidad central de procesamiento, CPU (por sus siglas del inglés Central Processing Unit), o, simplemente, el procesador, es el componente en una computadora digital que interpreta las instrucciones y procesa los datos contenidos en los programas de computadora. Las CPU proporcionan la característica fundamental de la computadora digital, la programabilidad, y son uno de los componentes necesarios encontrados en las computadoras de cualquier tiempo, junto con el almacenamiento primario y los dispositivos de entrada/salida. Se conoce como microprocesador el CPU que es manufacturado con circuitos integrados. Desde mediados de los años 1970, los microprocesadores de un solo chip han reemplazado casi totalmente todos los tipos de CPU, y hoy en día, el término "CPU" es aplicado usualmente a todos los microprocesadores. Si bien lo correcto



sería decir la CPU (por ser la unidad central de procesamiento), está muy extendido el término en masculino: el CPU.

Código binario: Un **código binario** es cualquier código formado por dos símbolos que pueden ser combinados para codificar información. Por lo general, los dos símbolos empleados son el 0 y el 1. Sin embargo, podría usarse cualquier par de símbolos. A cada uno de los dígitos (dígitos binarios) que forman las combinaciones del código se los denomina bits.

Código fuente: También denominado fuente o texto fuente. Es el texto que contiene las instrucciones del programa, escritas en el lenguaje de programación. Se trata de un archivo de texto legible que se puede copiar, modificar e imprimir sin dificultad. El concepto contrario es el código objeto, que se deriva del código fuente y está diseñado para ser legible sólo por la máquina

CORBA :(Common Object Request Broker Architecture). Especificación de mensajes basada en objetos desarrollada por el Object Management Group (OMG) o grupo de manejo de objetos.

D:

Depurador: Un depurador (en inglés, *debugger*), Programa diseñado para ayudar a programar sin errores una aplicación, ya que permite ver la ejecución instrucción por instrucción. Se pueden visualizar los valores de las variables en cada momento, establecer puntos de ruptura (instrucción que se marca indicando donde detener la ejecución del programa para ver hasta ese momento el resultado de la ejecución) y así subsanar errores lógicos. Los depuradores a nivel de código máquina presentan las instrucciones de máquina reales (desensambladas en lenguaje ensamblador) y permiten al programador observar los registros y ubicaciones en la memoria.

Distribución Linux: Una distribución de Linux es una variante de ese sistema operativo que incorpora determinados paquetes de software para satisfacer las necesidades de un grupo específico de usuarios, dando así origen a ediciones hogareñas, empresariales y para servidores. Pueden ser exclusivamente de software libre, o también incorporar aplicaciones o controladores propietarios. La base de cada distribución incluye el núcleo Linux, con las bibliotecas y herramientas del proyecto GNU y de muchos otros proyectos/grupos de software, como BSD, Xorg, Apache, MySQL, PostgreSQL, Perl, Python, PHP, Gnome y KDE.



Doxygen: es un Generador de documentación para C++, C, Java, Objective-C, Python, IDL (versiones Corba y Microsoft) y en cierta medida para PHP, C# y D. Dado que es fácilmente adaptable, funciona en la mayoría de sistemas Unix así como en Windows y Mac OS X. La mayor parte del código de Doxygen está escrita por Dimitri van Heesch. Doxygen es un acrónimo de dox (document) gen (generator), generador de documentación para código fuente.

F:

Framework: es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, un framework puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Frontend: Entiéndase como la parte (normalmente es un programa) del proceso de trabajo que interactúa con el usuario. El frontend, no procesa la información, si no que le dice al backend, que es lo que quiere el usuario.

G:

GKrellM: es un administrador de tareas basado en el conjunto de bibliotecas GTK+ que crea una pila simple de procesamiento de monitores del sistema. GKrellM puede ser usado para monitorear el estado de la memoria principal, microprocesador(es), discos duros, interfaces de red, volumen del sonido, casillas de correo locales y remotas, entre otros.

GNU: Licencia Publica General. Software desarrollado para distribución sin fines de lucro. El proyecto GNU (GNU es un acrónimo recursivo para "Gnu No es Unix") comenzó en 1984 para desarrollar un sistema operativo tipo Unix completo, que fuera Software Libre. Las variantes del sistema operativo GNU, que utilizan el kernel Linux, son muy utilizadas. La gente a menudo se refiere erróneamente a estos sistemas como "Linux", cuando es mas preciso y concreto llamarlos "GNU/Linux". Hay un kernel GNU en desarrollo, llamado Hurd, que será el llamado a sustituir al kernel Linux cuando este desarrollado.

La Free Software Foundation fue creada por Richard Stallman para financiar el proyecto GNU. Actualmente pone en las manos del usuario de Software Libre múltiples servicios para que la comunidad se desarrolle y sea productiva. Y, sobre todo, LIBRE.



GtK: es una biblioteca del equipo GTK+, la cual contiene los objetos y funciones para crear la interfaz gráfica de usuario. Maneja widgets como ventanas, botones, menús, etiquetas, deslizadores, pestañas, etc.

H:

Hardware: es la parte física de un computador y más ampliamente de cualquier dispositivo electrónico. Conjunto de componentes materiales de un sistema informático. Cada una de las partes físicas que forman un ordenador, incluidos sus periféricos. Maquinaria y equipos (CPU, discos, cintas, modem, cables, etc.). En operación, un computador es tanto hardware como software. Uno es inútil sin el otro. El diseño del hardware especifica los comandos que puede seguir, y las instrucciones le dicen qué hacer. Véase instrucción set.

I:

Intel Itanium:El Itanium, también conocido por su nombre en código Merced, fue el primer microprocesador de la arquitectura Intel Itanium (antes llamada IA64, creada por Hewlett-Packard y desarrollada conjuntamente por HP e Intel) que Intel lanzó al mercado.

K:

Kate: En informática, **Kate** es un editor de textos para KDE. Kate significa **KDE Advanced Text Editor**, es decir Editor de textos avanzado para KDE.

Keylogger: Un keylogger (registrador de teclas) es una herramienta de diagnóstico utilizada en el desarrollo de software que se encarga de registrar las pulsaciones que se realizan sobre el teclado, para memorizarlas en un fichero y/o enviarlas a través de Internet. Un keylogger es un malware del tipo daemon. Es un programa que recoge las pulsaciones que realiza el usuario sobre el teclado para enviárselo a través de Internet, averiguando normalmente las contraseñas que el usuario teclea.

M:

MacOS: Mac OS X (pronunciado Mac-o-ese-diez) es el actual sistema operativo de la familia de ordenadores Macintosh.

Malware: Programa maligno. Son todos aquellos programas diseñados para causar daños al hardware, software, redes,... como los virus, troyanos, gusanos, nukes,... Es un término común que se utiliza al referirse a cualquier programa malicioso.



Mono: es el nombre de un proyecto de código abierto iniciado por Ximian y actualmente impulsado por Novell (tras su adquisición de Ximian) para crear un grupo de herramientas libres, basadas en GNU/Linux y compatibles con .NET según lo especificado por el ECMA.

N:

.NET: es un proyecto de Microsoft para crear una nueva plataforma de desarrollo de software con énfasis en transparencia de redes, con independencia de plataforma de hardware y que permita un rápido desarrollo de aplicaciones. Basado en ella, la empresa intenta desarrollar una estrategia horizontal que integre todos sus productos, desde el sistema operativo hasta las herramientas de mercado.

O:

Opensolaris. Solaris puede considerarse uno de los sistemas operativos más avanzados. Sun denomina así a su sistema operativo.

P:

Plug-in: es una aplicación informática que interactúa con otra aplicación para aportarle una función o utilidad específica, generalmente muy específica, como por ejemplo servir como driver (controlador) en una aplicación, para hacer así funcionar un dispositivo en otro programa. Ésta aplicación adicional es ejecutada por la aplicación principal.

Procesador: Es el microchip encargado de ejecutar las instrucciones y procesar los datos que son necesarios para todas las funciones del computador. Se puede decir que es el cerebro del computador.

Q:

Qmake: QMake es una de las herramientas que se incluyen en el toolkit Qt. Ella nos permite crear un archivo [Makefile](#) que luego será usado para compilar nuestro proyecto. Para ello basta con ejecutar `qmake` dentro del directorio principal del proyecto y él buscará todos los archivos de código fuente existentes en los subdirectorios de manera recursiva. .

R:



Repositorio: Es un término utilizado en el dominio de las herramientas CASE. El repositorio podría definirse como la base de datos fundamental para el diseño; no sólo guarda datos, sino también algoritmos de diseño y, en general, elementos software necesarios para el trabajo de programación.

Root: Usuario de un sistema UNIX encargado de la administración de la máquina. El usuario root opera sin ningún tipo de restricción, por lo tanto, únicamente debe utilizarse para tareas administrativas.

S:

Sinergia:La sinergia es la integración de elementos que da como resultado algo más grande que la simple suma de éstos, es decir, cuando dos o más elementos se unen sinérgicamente crean un resultado que aprovecha y maximiza las cualidades de cada uno de los elementos.

Script: En informática, un script es un guión o conjunto de instrucciones. Permiten la automatización de tareas creando pequeñas utilidades. Es muy utilizado para la administración de sistemas UNIX. Son ejecutados por un intérprete de línea de órdenes y usualmente son archivos de texto. También script Puede considerarse una alteración o acción a una determinada plataforma, Muy parecido a los trucos que se usan para alterar juegos y conseguir cosas extras...

Servidor proxy: Los servidores Proxy son computadoras que aceleran su navegación almacenando copias locales de los sitios Web que Ud. visita frecuentemente.

Shareware: Se denomina shareware a una modalidad de distribución de software, tanto como juegos como programas utilitarios, para que el usuario pueda evaluar de forma gratuita el producto, por un tiempo especificado, aunque también las limitaciones pueden estar en algunas de las formas de uso o las capacidades finales. Las versiones de programas que reciben esta denominación permiten probar sus capacidades sin realizar el desembolso mucho mayor que representaría comprar el programa convencional completo.

SharpDevelop: es un entorno de desarrollo integrado libre para los lenguajes de programación C#, Visual Basic .NET y Boo.Es usado típicamente por aquellos programadores de los citados lenguajes, que no desean o no pueden usar el entorno de desarrollo de Microsoft, el Microsoft Visual Studio. Hay disponible un puerto para Mono/Gtk#, llamado MonoDevelop, el cual funciona en otros sistemas operativos.



SPARC: SPARC (del inglés *Scalable Processor ARCHitecture*) es una arquitectura RISC big-endian. Es decir, una arquitectura con un conjunto reducido de instrucciones.

Solaris: Variante del sistema operativo Unix desarrollada por Sun Microsystems. Solaris incluye OpenWindows, interfaz gráfica de usuario (GUI) basada en X-Windows.

T:

Thread: (Hilo de ejecución) Característica que permite a una aplicación realizar varias tareas simultáneamente. Los distintos hilos de ejecución comparten una serie de recursos tales como el espacio de memoria, los archivos abiertos, situación de autenticación, etc. Esta técnica permite simplificar el diseño de una aplicación que debe llevar a cabo distintas funciones simultáneamente.

U:

Unix: Es una familia de sistemas operativos tanto para ordenadores personales como para mainframes. Soporta gran número de usuarios y posibilita la ejecución de distintas tareas de forma simultánea (multiusuario y multitarea). Su facilidad de adaptación a distintas plataformas y la portabilidad de las aplicaciones (está escrito en lenguaje C) que ofrece hacen que se extienda rápidamente.