UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

Facultad #6

Trabajo de Diploma para Optar por el Título de Ingeniero en Ciencias Informáticas.

Título: Análisis y diseño del componente de acceso a disco mediante buffer para acelerar los procesos de Entrada /Salida.

Autor(es): Yuleimys Becerra Estrada.

Tutor(es): Yoandri Quintana Rondón.

Ciudad de la Habana, <julio, 1 del 2011>.

<Año 53 de la Revolución>

Todos y cada uno de nosotros paga puntualmente su cuota de sacrificio consciente de recibir el premio en la satisfacción del deber cumplido, conscientes de avanzar con todos hacia el Hombre Nuevo que se vislumbra en el horizonte.

Ernesto Che Guevara.

DEDICATORIA.

Dedico este trabajo a las personas más importantes de mi existencia, las cuales me ayudan a no perder mi eje, porque sólo pensarlo me da fuerzas para enfrentarme a cualquier reto que se oponga en mi destino. Porque me aman, me cuidan y me guían, por eso y por todas las cosas que me han dado, a mis queridos abuelos Juana Isidra que ya no está conmigo pero donde este ella siempre va estar orgullosa de mi, Sara, Octavio y Lando muchas gracias por todo, los amos.

A mis padres Grisel y Tomás, por darme la vida. Por brindarme a lo largo de estos cinco años todo su amor, esfuerzo y dedicación. Por ofrecerme su apoyo incondicional y por hacer posible la realización de este gran sueño, los adoro.

A mis hermanitos del alma, más que hermanos, mi amigos, mi todo, para ustedes también va este momento de felicidad los quiero mucho.

A todos mis tíos y tías por quererme tanto y por cuidar todo lo que yo amo y no pude cuidar por estar lejos.

A mis primas y primos que son un orgullo para mí.

Me gustaría dedicarle gran parte del esfuerzo realizado para hacer realidad este trabajo de diploma a Yoandri Quintana Rondón por ser mi amigo, mi tutor y por brindarme todo el apoyo en el transcurso de toda esta trayectoria.

A mis vecinos, que me apoyaron y se preocuparon por mí, los quiero mucho.

A mis amigas de la niñez Yamile, Gloria, Dialys, Yaremis, Yanelis, las quiero mucho mis hermanitas.

A la Revolución, a Fidel y a la UCI por darnos la oportunidad de estudiar y ser mejores personas cada día.

A todos mis amigos, los que siempre estaban cuando más los necesité.

AGRADECIMIENTOS.

Quisiera a través de este trabajo agradecerles a todas las personas que hayan estado involucradas de una manera u otra en mi vida estudiantil.

A mi mamá Grisel, mi papá Tomás por apoyarme, darme ánimos y por creer en mí, los quiero mucho.

A Yalili, Adayli, Diana, Elizabeth, Yamile, y Nixys por quererme y ser como hermanas para mí, por dejarme ser parte de su vida y por estar en la mía, además siempre estuvieron cuando las necesité, por guiarme, pero sobre todo por apoyarme aún cuando estaba tomando una decisión equivocada.

A Iodani Batista, Pavel Mena, Danilo por ayudarme siempre incondicionalmente, por ser mis amigos y sostener mi mano cada vez que pensé desfallecer.

A Yamit, Susana Gómez por compartir tantos momentos inolvidables conmigo.

A mis amigas del apartamento, Yudalis, Leyvis, Yuliet, Yudirenia, Yaneisi por estar siempre que las necesité.

A mis antiguos grupo 9103, 9203, 9307, 9408 y 6506 en general porque con ustedes pasé los mejores años de universidad.

A los profe del proyecto SCCM, que son las personas que me bridaron su mano al hora que lo necesité, gracias por todo.

A los profesores miembros del tribunal de los cortes de tesis: Yanisley Álvarez, Carlo Enrique y Carlo Luis y también a mi oponente por corregirme y por todos los señalamientos realizados.

A todas las personas que me quieren, que me han ayudado aún sin quererlo, a todos los que me han enseñado, aquellos que me han criticado constructivamente y a los que han sabido respetar todas mis decisiones, gracias porque me han reconocido como una persona y me han respetado como tal. A todos ustedes mil gracias, siempre van a estar en mi corazón.

DECLARACIÓN DE AUTORÍA.

Declaro que soy el único autor de este trabajo y autorizo a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los 1 días del mes de julio del año 2011.

_____ _____

Yuleimys Becerra Estrada Ing. Yoandri Quintana Rondón.

DATOS DE CONTACTO.

DATOS DE CONTACTO

Síntesis del Tutor:

Nombre y Apellidos: Yoandri Quintana Rondón.

Especialidad: Ingeniería en Ciencias Informáticas.

Años de experiencia: 2

Teléfono particular:

Dirección electrónica para correspondencia: yqrondon@uci.cu

RESUMEN.

En los últimos 30 años, el crecimiento en velocidad de los procesadores y la memoria principal ha sido superior al crecimiento de acceso a disco. La velocidad del procesador y de la memoria se ha incrementado en dos órdenes de magnitud con respecto al disco. Como resultado de ello se encuentra hoy que los discos son menos, cuatro veces más lentos que la memoria principal. Este avance se espera que continúe en el futuro inmediato. De este modo, el rendimiento de los subsistemas de almacenamiento en disco es de una importancia vital y se han realizado muchas investigaciones sobre maneras de mejorar su rendimiento.

El presente trabajo consiste en la realización del análisis y diseño del componente de acceso a disco mediante buffer, que perfeccionará el rendimiento de los productos del departamento de Señales Digitales. Con el mismo se puede aumentar la velocidad y reducir el tiempo de ejecución de los procesos. Esto permite que el tiempo de acceso y respuesta de los sistemas sea más rápido, propiciando un mejor resultado para los clientes que integran los proyectos.

Para la elaboración del componente se utiliza el lenguaje de programación C++ y RUP como metodología de desarrollo de software. Además se hace el uso de las actuales tecnologías libres que existen en la actualidad a nivel mundial que pueden ser útil en el desarrollo de la propuesta de solución.

PALABRAS CLAVES.

Acceso, Buffer, Componente, Disco, Entrada, Procesos, Salida.

INDICE

ILUSTRACIONES.

TABLAS.

Introducción.

Las tecnologías en el mundo presentan un gran auge dado el desarrollo científico técnico alcanzado hasta hoy. Es por ello que la informatización de la sociedad ha crecido a nivel mundial y ha traído consigo el aumento de capacidad de generación y almacenamiento de la información. Las soluciones informáticas aportan mejoras considerables para el entorno en el cual se aplican, ahorrando tiempo, recursos y generando ganancias. Como consecuencia de la gran competencia existente en el mercado internacional entre las compañías dedicadas al desarrollo de software, surge la necesidad de adaptarse constantemente a los cambios que ocurren en este campo.

El desarrollo tecnológico ha demostrado las infinitas posibilidades que se abren al avanzar en este campo. Se ha podido comprobar que compartir lo que se tiene sobre la tecnología, es la mejor forma de adquirir conocimiento y desarrollo de manera abundante para la sociedad. A causa del mundo consumista en que se vive y la creciente aparición de necesidades, siempre están las puertas abiertas para nuevos productos. La tecnología avanza muy rápido en la actualidad porque cada vez se manejan mayores cantidades de información. En casi todas las partes del mundo existen servidores con altas velocidades y capacidades de almacenamiento. Hoy día los dispositivos de almacenamiento se han vuelto casi tan importantes como el mismo computador.

Para estos tiempos existen varios dispositivos de almacenamientos tales como: CD, DVD, memoria flash, Disco Duro, cintas magnéticas, SAN, memoria interna, etc. En el mundo existen herramientas para acelerar procesos como puede ser el copiado, una de estas puede ser el Talent Copy V pensada para copiar y mover grandes cantidades de archivos. Tiene como desventaja que es una herramienta propietaria.

Cuba, como país en vías de desarrollo, no está ajena a esta situación, por lo que se ha dado a la tarea de informatizar la nación. Con este fin, en el año 1997, se definieron los “Lineamientos estratégicos para la informatización de la sociedad cubana” por el Comité Ejecutivo del Consejo de Ministros de Cuba. A partir de ese momento se realizan una serie de acciones en pos de lograr estos objetivos. El Ministerio de la Informática y las Comunicaciones fue el principal encargado de introducir a Cuba dentro del mundo de la tecnología informática. Como parte de estos cambios las organizaciones del estado se dieron a la tarea de actualizar, tanto la tecnología existente en sus centros, como la manera de recoger y gestionar la

información. A pesar de ser un país bloqueado económicamente, se apuesta por el desarrollo de la Informática para el progreso de la economía y en este sentido se han creado los Politécnicos de Informática, los Joven Club de Computación y Electrónica y se han introducido las carreras universitarias de esta materia y de Computación, dentro de las cuales se encuentra la UCI (Universidad de las ciencias Informáticas) como centro rector de la industria cubana de producción de software.

La UCI es una de las instituciones que desde su creación ha desempeñado un papel importante en la batalla de informatizar todos los campos de la economía del país. Este centro está compuesto por facultades, éstas por centros de desarrollo, los centros por departamentos y los departamentos por proyectos.

En la facultad 6 en el centro GeySed se encuentra el Departamento de Señales Digitales en el cual se desarrollan distintos proyectos tales como: Primicia, Video Web, SCCM, PTRTV, Video Vigilancia que son los encargados del trabajo con medias (audio, video, imágenes). En este departamento es necesario ejecutar procesos de copia, codificación, extracción de fotograma o resumen visual que requieren acceso al disco duro en un tiempo de ejecución. Los proyectos no son los más eficientes porque en varias ocasiones los tiempos de acceso y respuesta no son los esperados.

Se puede decir que el trabajo con media por lo general resulta tedioso ya que la máquina tiene que procesar demasiados datos lo que desespera al usuario, además estos procesos para mejorar su rendimiento necesitan de un hardware específico. Esto no es solo un problema para Cuba sino para el mundo porque las tecnologías que existen para acelerar los proceso de E/S son propietarias y con un alto costo en el mercado.

A partir de la situación descrita anteriormente se identifica como **problema científico:** ¿Cómo lograr agilizar los procesos de E/S sobre altos volúmenes de información almacenados en disco duro que requieren de un rápido tiempo de ejecución?

Una vez identificado el problema científico se define como **objeto de estudio:** Las tecnologías de acceso a disco que permitan acelerar los procesos entradas/salidas (E/S).

Definiendo como **campo de acción:** La tecnología de acceso a disco mediante buffer que permitan agilizar los procesos de entrada/salida en los sistemas del departamento de Señales Digitales.

Objetivo General: Diseñar un componente de acceso a disco mediante buffer que permita acelerar los procesos de entrada/salida para ser aplicado a los proyectos del departamento de Señales Digitales.

Idea a Defender: El análisis y diseño de un componente de acceso a disco mediante buffer que permita acelerar los procesos E/S, garantizará un mejor rendimiento en los productos del departamento de Señales Digitales.

Para dar cumplimiento a los objetivos se plantean las siguientes **tareas de investigación:**

1. Caracterizar las tecnologías de acceso a disco que permitan mejorar los procesos de E/S y hacer una selección de las tecnologías más idónea.
2. Identificar principales conceptos asociados al acceso a disco mediante el buffer.
3. Identificar las herramientas y tecnologías a utilizar para el desarrollo.
4. Realizar el análisis y diseño del componente de acceso a disco mediante buffer que permita acelerar los procesos E/S.
5. Identificar algoritmos de acceso a disco que permitan acelerar los procesos de E/S.
6. Proponer el algoritmo que permita el acceso a disco mediante buffer que permita acelerar los procesos E/S.

La investigación usa como parte de todo trabajo **métodos científicos** que servirán para arribar a los resultados esperados.

Métodos Teóricos:

Histórico-Lógico: Este método se utiliza para el estudio de los temas relacionados con el desarrollo de un componente de acceso a disco mediante buffer, su evolución y para la metodología a utilizar en el desarrollo.

Analítico – Sintético: Este método se utiliza para el análisis de documentos, materiales, y temas relacionados a las mejores prácticas en el desarrollo de un componente de acceso a disco mediante buffer. Permite definir los conceptos fundamentales del tema y se usa para especificar la metodología a utilizar y las dimensiones del acceso a disco.

Método de Modelación: Mediante la modelación se comprenderá mejor cómo debería conformarse el componente, es decir, cómo debería ser implementada posteriormente, evidenciándose así el método.

El presente trabajo está estructurado por **tres capítulos**, distribuidos de la siguiente manera:

Capítulo 1: Fundamento teórico del análisis y diseño del componente de acceso a disco mediante buffer.

Es donde se plantean todos los elementos teóricos que sustentan el problema científico, se definen los principales conceptos asociados al acceso a disco para mejorar los procesos de E/S. Se exponen sus características y técnicas.

Capítulo 2: Tendencias y tecnologías actuales a desarrollar para el análisis y diseño del componente de acceso a disco mediante buffer.

En este capítulo es dónde se especifican las tecnologías a utilizar, la metodología, el lenguaje de programación y se fundamenta el porqué de la elección realizada.

Capítulo 3: Propuesta de la solución del componente de acceso a disco mediante buffer.

En este capítulo se realiza el diseño de la solución, se muestran los diagramas de clases del diseño. Se diseñan los distintos modelos y artefactos que brinda determinada metodología, lo que permitirá un mejor entendimiento del problema. Además se propone el algoritmo del componente.

Capítulo 1: Fundamentación Teórica del análisis y diseño del componente de acceso a disco mediante buffer.

1.1. Introducción:

En el presente capítulo se muestra una descripción de los aspectos más importantes de la investigación, donde se enuncian los conceptos y definiciones que hacen posible una mejor comprensión del tema. Se refleja además el estado del arte sobre las técnicas para el acceso disco y se abordan algunos elementos teóricos acerca de las tecnologías de acceso a disco que permitan mejorar los procesos de E/S.

1.2. Conceptos Asociados al dominio del problema:

Memoria RAM:

RAM son las siglas de (Random Access Memory, Memoria de Acceso Aleatorio). Es la memoria desde donde el procesador recibe las instrucciones y guarda los resultados, además es un tipo de memoria de ordenador a la que se puede acceder aleatoriamente o sea permite el acceso a cualquier byte de memoria sin acceder a los bytes precedentes. La memoria RAM es el tipo de memoria más común en ordenadores y otros dispositivos como impresoras. (Masadelante, 1999)

Existen dos tipos básicos de memoria RAM:

RAM dinámica (DRAM).

RAM estática (SRAM).

La memoria RAM dinámica necesita actualizarse miles de veces por segundo, mientras que la memoria RAM estática no necesita actualizarse, por lo que es más rápida, aunque también más cara. Ambos tipos de memoria RAM son volátiles, es decir, que pierden su contenido cuando se apaga el equipo. La memoria principal o RAM es donde el ordenador guarda los datos que está utilizando en el momento presente. La diferencia entre la RAM y otros tipos de memoria de almacenamiento, como los disquetes o los discos duros, es que la RAM es mucho (mucho) más rápida, y que se borra al apagar el ordenador, no como éstos. (Masadelante, 1999)

CPU.

CPU, abreviatura de Central Processing Unit (unidad de proceso central). Es la parte que constituye el cerebro de cualquier computadora, es el encargado de realizar y dirigir todas las funciones. A veces es referido simplemente como el procesador o procesador central, la CPU es donde se producen la mayoría de los cálculos. En términos de potencia del ordenador, la CPU es el elemento más importante de un sistema informático. (Masadelante, 1999)

Los medios de almacenamiento han evolucionado en forma notable desde las primeras computadoras y este ritmo se ha acelerado con la introducción de las PC. Una de las partes más importantes del sistema de una computadora son los dispositivos que permiten guardar los resultados de los trabajos realizados.

1.3. Medios de almacenamiento más comunes:

1.3.1. Buffer:

Es el área de almacenaje temporal, por lo general en la RAM. El objetivo de la mayor parte del buffer es actuar como un área propia, necesitando el permiso de la CPU antes de la transferencia de datos a un dispositivo. Como la lectura y la escritura de datos a un disco son relativamente lentas, muchos programas almacenan la información de los cambios de datos dentro de un buffer y luego lo copian a un disco. Ejemplo de estos pueden ser los procesadores de texto que emplean un buffer para almacenar los cambios de un archivo. Es un área de memoria principal reservada para contener los datos leídos de un archivo mientras se utilizan. Cuando esta área temporal queda llena, el programa puede empezar a utilizar estos datos. Manejar un buffer implica trabajar con grandes grupos de datos de memoria RAM para que el número de accesos al almacenamiento se reduzca. (Martinez)

1.3.2. Procesos E/S:

Un proceso es un programa en ejecución, el cual son gestionados por el sistema operativo. Las entradas son las señales recibidas por la unidad, mientras que las salidas son las señales enviadas por ésta.

1.3.3. El buffer-cache:

Va incluida en la controladora interna del disco duro, de modo que todos los datos que se leen y escriben en el disco duro se almacenan primeramente en el buffer. Gestiona las memorias intermedias asociadas a los bloques de disco. El contenido de los buffer debe colocarse en páginas de memoria principal. El buffer-caché mantiene copias de bloques de disco individuales. Las entradas del caché están identificadas por el

dispositivo y número de bloque. Cada buffer se refiere a cualquier bloque en el disco y consiste de una cabecera y un área de memoria igual al tamaño del bloque del dispositivo. (Pedroza, 2002)

La caché de buffers es UNIX, es básicamente, una caché de disco. Las operaciones de E/S con el disco se manejan a través de la caché de buffers. La transferencia de datos entre la caché buffers y el espacio de usuario del proceso siempre se produce mediante DMA (Acceso Directo a la Memoria). Como la caché de buffers y la zona de E/S del proceso residen ambas en memoria principal, se usará el DMA para llevar una copia de memoria a memoria. Esta acción no gastará ningún ciclo del procesador, pero consumirá ciclos de bus.

1.3.4. Buffer Sencillo:

La clave de apoyo más simple que el sistema operativo puede ofrecer es el buffer sencillo. Cuando un proceso de usuario realiza una petición de E/S, el sistema operativo le asigna a la operación un buffer en la parte del sistema de la memoria principal

Para los dispositivos de bloques, el esquema del buffer sencillo puede describirse como sigue. Las transferencias de entrada se realizan al buffer del sistema. Cuando se ha completado la transferencia, el proceso mueve el bloque al espacio del usuario y pide otro bloque inmediatamente. Esta técnica se llama lectura por adelantado o entrada anticipada: se realiza esperando que el bloque se necesite más adelante. Para muchos tipos de operaciones, ésta suposición es razonable la mayoría de la veces. La lectura será innecesaria sólo al final de una secuencia de procesamiento.

1.3.5. Buffer doble:

Se puede realizar una mejora del buffer sencillo asignando dos buffers del sistema a cada operación. De esta forma, un proceso puede transferir datos hacia (o desde) un buffer mientras que el sistema operativo vacía (o rellena) el otro. Esta técnica se conoce como buffer doble o intercambio de buffers.
Para las transferencias de bloques, se puede hacer una estimación aproximada del tiempo de transferencia como el máximo de C y T, por tanto, es posible que el dispositivo de bloque funcione a su máxima velocidad si C< T. Por otra lado, si C> T, el buffer doble asegura que el proceso no tendrá que esperar en la E/S. En cualquier caso se consigue una mejora con respecto al buffer sencillo.

1.3.6. Buffer circular:

El esquema del buffer doble debería solucionar el flujo de datos entre un dispositivo de E/S y un proceso. Si preocupa el rendimiento de un proceso determinado, sería deseable que las operaciones de E/S fueran capaces de ir al ritmo del proceso. El buffer doble puede ser inapropiado si el proceso lleva a cabo rápidas ráfagas de E/S. En este caso, el problema puede mitigarse usando más de dos buffers.
Cuando se emplean más de dos, el conjunto de buffers se conoce con el nombre de buffer circular. Cada buffer individual constituye una unidad del buffer circular. Este es, sencillamente, el modelo del productor/consumidor con un buffer limitado.

1.3.7. Memoria Interna:

Es aquella que almacena el estado de las variables que maneja el autómata: entradas, salidas, contadores, relés internos, señales de estado, etc. La memoria se gestiona como un recurso unificado para los programas de usuario y para el caché de disco, de tal forma que toda la memoria libre puede ser usada para caché y ésta puede a su vez ser reducida cuando se ejecuten grandes programas. (Pedroza, 2002)

1.3.8. CD:

Las unidades de CD-ROM son sólo de lectura. Es decir, pueden leer la información en un disco, pero no pueden escribir datos en él. Una regrabadora (CD-RW) puede grabar y regrabar discos compactos. Las características básicas de estas unidades son la velocidad de lectura, de grabación y de regrabación. En discos regrabables es normalmente menor que en los discos grabables una sola vez. (Ortega, 2007)

1.3.9. DVD:

Las unidades de DVD-ROM son aparentemente iguales que las de CD-ROM, pueden leer tanto discos DVD-ROM como CD-ROM. Se diferencia de las unidades lectoras de CD-ROM en que el soporte empleado tiene hasta 17 Gb de capacidad, y en la velocidad de lectura de los datos. Puede leer y grabar imágenes, sonido y datos en discos de varios gigabytes de capacidad. Su mayor capacidad de almacenamiento se debe, entre otras cosas, a que puede utilizar ambas caras del disco y, en algunos casos, hasta dos capas por cada cara, mientras que el CD sólo utiliza una cara y una capa. (Ortega, 2007)

1.3.10. Memoria Flash:

Es un dispositivo de almacenamiento externo que permite funcionar a velocidades muy superiores cuando los sistemas emplean lectura y escritura en diferentes puntos de esta memoria al mismo tiempo. (Ortega, 2007)

1.3.11. Discos flexibles:

A partir de finales de 1970 y hasta principios de 1980 el disco flexible era el principal dispositivo de almacenamiento utilizado por las microcomputadoras. Los programas y la información se guardaban en discos flexibles. Una unidad de disco flexible, es un dispositivo que lee y escribe información de y hacia discos flexibles. Esta flexibilidad es importante, debido a que permite a las cabezas acceder a la información en forma aleatoria en lugar del modo secuencial, es decir, que las cabezas pueden saltar de un punto a otro sin tener que buscar en toda la información almacenada entre las antiguas y nuevas. Ejemplo disquete 3 1/2, disco compacto de memoria. (Hillar, 2000)

1.3.12. Discos rígidos:

Los discos rígidos son los medios de almacenamiento de las PCs. Es donde se tiene el Sistema Operativo instalado y los programas que se utilizan, así como juegos, fotos, música etc. Hay principalmente dos características a tener en cuenta la velocidad y la capacidad de este dispositivo. Es el componente utilizado para almacenar los datos de manera permanente, a diferencia de la memoria RAM, que se borra cada vez que se reinicia el ordenador, motivo por el cual a veces se denomina dispositivo de almacenamiento masivo a los discos rígidos. Ejemplo disco duro, disco rígido externo. (CP-Group, 2007)

1.3.13. Disco Duro:

Es un dispositivo magnético que almacena todos los programas y datos de la computadora. Es la parte del ordenador que contiene la información electrónica y donde se almacenan todos los programas (software). Es uno de los componentes del hardware más importantes dentro de una PC. A pesar de su rápida evolución e incremento de capacidad y velocidad, se hacen insuficientes para las misiones que se requieren. (Ortega, 2007)

Si se habla de disco duro se puede citar los distintos tipos de conexión que poseen los mismos con la placa base, es decir pueden ser SATA, IDE, SCSI o SAS:

- IDE: *Integrated Device Electronics* ("Dispositivo con electrónica integrada") o ATA, controla los dispositivos de almacenamiento masivo de datos, como los discos duros y ATAPI (*Advanced Technology Attachment Packet Interface*) Hasta aproximadamente el 2004, el estándar principal por su versatilidad y asequibilidad. Son planos, anchos y alargados.
- SCSI: Son interfaces preparadas para discos duros de gran capacidad de almacenamiento y velocidad de rotación. Se presentan bajo tres especificaciones:
  - SCSI Estándar (Standard SCSI)
  - SCSI Rápido (Fast SCSI)
  - SCSI Ancho-Rápido (Fast-Wide SCSI)

Su tiempo medio de acceso puede llegar a 7 milisegundos y su velocidad de transmisión secuencial de información puede alcanzar teóricamente los 5 Mbps

- Los discos SCSI Estándares, los 10 Mbps
- Los discos SCSI Rápidos y los 20 Mbps
- Los discos SCSI Anchos-Rápidos (SCSI-2).

Un controlador SCSI puede manejar hasta 7 discos duros SCSI (o 7 periféricos SCSI) con conexión tipo margarita (*daisy-chain*). A diferencia de los discos IDE, pueden trabajar asincrónicamente con relación al microprocesador, lo que posibilita una mayor velocidad de transferencia.

- SATA (Serial ATA): El más novedoso de los estándares de conexión, utiliza un bus serie para la transmisión de datos. Notablemente más rápido y eficiente que IDE. Existen tres versiones.
  - SATA 1 con velocidad de transferencia de hasta 150 MB/s (hoy día descatalogado).
  - SATA 2 de hasta 300 MB/s, el más extendido en la actualidad.
  - SATA 3 de hasta 600 MB/s el cual se está empezando a hacer hueco en el mercado. Físicamente es mucho más pequeño y cómodo que los IDE, además de permitir conexión en caliente.
- SAS (*Serial Attached SCSI*): Interfaz de transferencia de datos en serie. Es sucesor del SCSI paralelo, aunque sigue utilizando comandos SCSI para interaccionar con los dispositivos SAS.

Aumenta la velocidad y permite la conexión y desconexión en caliente. Una de las principales características es que aumenta la velocidad de transferencia al aumentar el número de dispositivos conectados, es decir, puede gestionar una tasa de transferencia constante para cada dispositivo conectado, además de terminar con la limitación de 16 dispositivos existente en SCSI, es por ello que se vaticina que la tecnología SAS irá reemplazando a su predecesora SCSI. Además, el conector es el mismo que en la interfaz SATA y permite utilizar estos discos duros, para aplicaciones con menos necesidad de velocidad, ahorrando costes. Por lo tanto, las unidades SATA pueden ser utilizadas por controladoras SAS pero no a la inversa, una controladora SATA no reconoce discos SAS.

- SAN: (Storage Area Network) Red de área de almacenamiento. Es una red dedicada específicamente a la tarea de transporte de datos para almacenamiento y recuperación. Se conecta a múltiples sistemas de servidores a un conjunto centralizado de almacenamiento en disco. La red SAN permite transferencias de datos entre sistemas de servidores y subsistemas de E / S en la misma velocidad. Totalmente redundante, fiable y veloz por su conexión mediante Canal de Fibra Óptica. (infoviews.S:A)
- NAS: (Network Attached Storage) Almacenamiento Conectado a Red. Define todo sistema que permita compartir almacenamiento de dato en un punto central a través de la red. Este punto central se le conoce como el servidor NAS. El servidor NAS puede incluir uno o más discos duros, y tiene la capacidad de almacenar y compartir data proveniente de diferentes fuentes (computadoras, servidores, servicios en el web, entre otros). El servidor NAS es uno más simplificado y no incluye un sistema operativo completo. Estos también no suelen a incluir accesorios tales como teclados, ratón y monitores.

1.4. Objeto de Estudio.

El objeto de estudio está enmarcado en las tecnologías de acceso a disco que permitan acelerar los procesos de entradas/salidas.

1.4.1. Descripción General:

En el mundo existen varias tecnologías de acceso a disco tales como SCSI, ATA, SATA (Serial ATA), RAID con características, semejanzas y diferencias. Se puede decir que la SCSI es muy cara, por lo tanto

no es muy utilizada. Esta tecnología ha sido la solución para las corporaciones y centros de cálculo con requerimientos de datos de muy alta disponibilidad, a la vez es una interfaz preparada para disco duro de gran capacidad de almacenamiento y velocidad de rotación. De la ATA, se puede decir que controla los dispositivos de almacenamiento masivo de datos, es mucho más simple que SCSI habiendo sido diseñado para sistemas operativos monousuario y monoproceso, y por ello no tiene algunas de las características de alto nivel de la tecnología SCSI. Otras de las tecnologías es la SATA (Serial ATA), mezcla las tecnologías de señal serie con los discos ATA. Los datos se envían en serie, todos por el mismo cable y uno detrás de otro, permitiendo mayores frecuencias. Se puede añadir a los beneficios anteriores que SATA tiene la característica de evitar autobloqueos; en primer lugar, la conexión entre el disco y el controlador es una conexión punto a punto en lugar de una conexión bus. Para cada disco existe un único cable dedicado que lo conecta al controlador, esto va a cambiar la manera de configurar y desarrollar debido a que una topología de conexión punto a punto permite el uso de controladores que pueden extraer mucho más rendimiento a los discos ATA. Por último, la tecnología RAID es un método de combinación de varios discos duros. Ofrece mayor tolerancia a fallos y más altos niveles de rendimientos que en un sólo disco o un grupo de disco duro independientes.

1.4.2. Características de las tecnologías de acceso a disco.

SCSI: (*Small Computer System Interface*) Interface de sistema para computadora.

- Es una tecnología que no es muy usada por su difícil nivel de adquisición. Es típica y casi exclusiva de ordenadores caros, servidores de red. En la década de los 90, esta tecnología ha sido la solución para las corporaciones y centros de cálculo con requerimientos de datos de muy alta disponibilidad. Incorpora un número de alto nivel para poder dar rendimiento y escalabilidad en servidores multiusuario, multi-threading (multi-hilo) para maximizar la transferencia de datos entre la memoria y el controlador y maximizar la eficiencia de las cabezas en su desplazamiento en el disco. Estos discos suelen ser más rápidos a la hora de transmitir datos, a la vez que usan menos al procesador para hacerlo, lo que se traduce en un aumento de prestaciones. El almacenamiento masivo de datos en los grandes servidores y corporaciones, ha sido el dominio de la tecnología SCSI. El cable de interconexión soporta hasta 15 dispositivos por canal SCSI, haciendo de esta manera que la tecnología SCSI sea una plataforma excelente para sistemas de almacenamiento masivo. (Tella, 2003)

ATA (*Advanced Technology Attachment*) Ajunto de tecnología avanzada.

- Es una tecnología que ha sido el dominio de los equipos de sobremesa. Han existido varios intentos de los fabricantes de mover la tecnología ATA a los servidores y a los entornos empresariales, pero esto no ha cuajado debido a falta de características hardware específicas en los discos ATA y sus controladores. El cable ATA es muy corto: sólo 18 pulgadas, y soporta solo dos dispositivos por cable lo que le hace totalmente inadecuado para sistemas extensos de almacenamiento. Fue desarrollada y optimizada para sistemas de escritorio. (Tella, 2003)

SATA (Serial ATA): (*Serial Advanced Technology Attachment*) Tecnología serial avanzada de contacto.

- Esta tecnología mezcla las tecnologías de señal serie con los discos ATA. Esto es importante debido a que soluciona un número importante de problemas que afectan al uso de almacenamiento ATA en sistemas realmente grandes, o cuando las necesidades de almacenamiento son muy altas. El cable es estrecho y flexible por lo que no afecta a los sistemas de ventilación pudiendo llegar hasta el tamaño de 1 metro por lo que los discos pueden ya estar alojados fuera del servidor. Esta tecnología también elimina el requerimiento de tener que usar +5V en las actuales fuentes de alimentación cuyo único sentido era proporcionar este voltaje a los discos. (Tella, 2003)

RAID: (Redundant Array of Independent Disk) Matriz Redundante de Discos Independientes.

- Esta tecnología significa matriz redundante de discos independientes. Empieza a jugar un importante papel en sistemas de almacenamiento. Originalmente inventada para asegurar disponibilidad de datos incluso en el caso de un fallo catastrófico de un disco, se ha vuelto un estándar para hacer subsistemas de almacenamiento en centros de datos en los cuales la disponibilidad 24x7 es un requerimiento imprescindible. Crea un sistema de datos redundantes en el cual la pérdida de un disco completo no impacta en la disponibilidad de los datos. Es un método de combinación de varios discos duros para formar una única unidad lógica en la que se almacenan los datos de forma redundante. En este método, la información se reparte entre varios discos, usando técnicas como el entrelazado de bloques (RAID nivel 0) o la duplicación de discos (RAID nivel 1) para proporcionar redundancia, reducir el tiempo de acceso, y/o obtener mayor ancho de banda para leer y/o escribir, así como la posibilidad de recuperar un sistema tras la

avería de uno de los discos. Un RAID, para el sistema operativo, aparenta ser un sólo disco duro lógico (LUN). Si se produce un fallo, mantiene el servidor activo y en funcionamiento hasta que se sustituya la unidad defectuosa. Ofrece varias opciones, llamadas niveles RAID, cada una de las cuales proporciona un equilibrio distinto entre tolerancia a fallos, rendimiento y coste. Todos los sistemas RAID suponen la pérdida de parte de la capacidad de almacenamiento de los discos, para conseguir la redundancia o almacenar los datos de paridad. Los sistemas RAID profesionales deben incluir los elementos críticos por duplicado: fuentes de alimentación y ventiladores redundantes. (Tella, 2003)

SAN:(Storage Área Network) Red de área de almacenamiento.

- Es una tecnología de red de almacenamiento de altas prestaciones. Su función es centralizar el almacenamiento de los ficheros en una red de alta velocidad y máxima seguridad. Es una solución global donde se comparte todos los recursos de almacenamiento en la compañía, su función es la de conectar de manera rápida, segura y fiable los distintos elementos que la conforman. El tipo de tráfico en una SAN es muy similar al de los discos duros como ATA, SATA y SCSI. La SAN es su compatibilidad con los dispositivos SCSI ya existentes, aprovechando las inversiones ya realizadas y permitiendo el crecimiento a partir del hardware ya existente.

NAS: (Network Attached Storage) Almacenamiento Conectado a Red.

- Es una tecnología de almacenamiento conectada a red. La misma es un servidor de almacenamiento que se puede conectar fácilmente a la red de una compañía para asistir al servidor de archivos y proporcionar espacio de almacenamiento tolerante a fallas. Permite almacenar datos en una sola ubicación que está accesible a cualquier computadora en la misma red. La solución NAS de almacenamiento en la red ampliable y completamente personalizada, tiene una gran fiabilidad de sus componentes, ofrece a los profesionales toda la protección que necesitan.

1.5. Comparación de SAN vs NAS.

Estas tecnologías son las que más importantes se consideran para la parte de acceso y almacenamientos. Se puede decir que la SAN o un NAS permitirán a una empresa gestionar de manera efectiva sus exigencias de almacenamiento de manera independiente a los servidores existentes. Ambas soluciones

parecen casi idénticas y utilizan sistemas RAID que están conectados a una interconexión (red). Las dos gestionan el sistema de archivos, copias de seguridad y espejos se generan en los archivos, no bloques (Esto puede ahorrar ancho de banda y tiempo).

Los productos NAS proveen soluciones de almacenamientos que se conectan mediante una red (TCP/IP) a unos o varios equipos (PC o servidores) con el propósito de establecer un sistema central de almacenamiento que facilite las labores administrativas y respaldo de la información para las micro y las pequeñas empresas. A través de su puerto Ethernet Gigabit incluido en todos sus modelos, podrá acceder a velocidades de transmisión de información de hasta 1 Gbps. La alta disponibilidad de la información es importante para los clientes, por lo que algunos dispositivos NAS incluyen la tecnología RAID, ya que le permite elegir la mejor forma de utilizar su dispositivo para que se ajuste a sus necesidades. Esta potente solución de almacenamiento cuenta con numerosas características personalizables, entre las que se incluye capacidad de control de acceso a más de 500 usuarios. Puede utilizarse para realizar las copias de seguridad o compartir de forma segura archivos de gran importancia. Con este sistema se conseguirá ahorrar una considerable cantidad de espacio en el disco. (infoviews.S:A)

La solución SAN se puede describir como una red de discos de almacenamiento. Simplifica las tareas de administración del sistema. Es la solución perfecta para los responsables de departamentos informáticos que buscan una solución de almacenamiento sofisticada y duradera. Debido a que la información almacenada no reside directamente en ninguno de los servidores de la red o estaciones de trabajo, se optimiza el poder de procesamiento para aplicaciones comerciales como ICM (Sistema de Administración Digital de Documentos) y la capacidad de almacenamiento se puede proporcionar al servidor donde más se requiera. (infoviews.S:A)

Teniendo en cuenta lo explicado, se llega a la conclusión de que la mejor tecnología para el Departamento de Señales Digitales es la NAS. Permite reducir el tiempo de ejecución de los procesos que requieren el acceso a disco. Esta tecnología brindaría algunas ventajas para el departamento como: mayor intercambio de información especialmente en los sistemas operativos, reduce la presión de la red, mejora enormemente la capacidad de almacenamiento. Además se puede trabajar sin afectar el servidor de red para copias de seguridad de archivos mientras se trabaja.

1.6. Características de los Dispositivos de E/S (categorías).

Son todos aquellos dispositivos conectados a un computador y diferentes ha: Unidad Central de procesamiento (UCP), Memoria y Reloj. El código para manejar la E/S de los dispositivos de E/S es amplio y complejo. Resuelven los problemas de: sincronización, detección de interrupciones, llamadas al sistema.

1.7. Dispositivos de entrada y salida.

En la presente investigación se mencionan los distintos tipos de dispositivos que ayudan a interactuar con la computadora y que son muy útiles.

Los dispositivos de Entrada y Salida permiten la comunicación entre la computadora y el usuario. En primer término se habla de los dispositivos de entrada, que como su nombre lo indica, sirven para introducir datos (información) a la computadora para su proceso. Los datos se leen de los dispositivos de entrada y se almacenan en la memoria central o interna. (Miguel, 2000)

A continuación se menciona algunos de los dispositivos de entrada:

Teclado alfanumérico: Se utiliza principalmente como un dispositivo para introducir texto.

Mouse: Es un dispositivo electrónico que permite dar instrucciones a la computadora a través de un cursor que aparece en la pantalla. (Miguel, 2000)

Scanner: Es una unidad de ingreso de información. Permite la introducción de imágenes gráficas al computador mediante un sistema de matrices de puntos, como resultado de un barrido óptico del documento. (Miguel, 2000)

Discos Duros: Son dispositivos de almacenamiento secundario con una superficie circular y plana, que se utilizan para registrar información masiva, programas y datos en computadores personales o microcomputadoras. (Miguel, 2000)

Unidades de Cinta: Debido a que el tamaño de los discos duros está en aumento constante, la práctica de hacer copias de seguridad de los archivos en la computadora usando disquetes se ha vuelto costosa y tediosa. No obstante, la unidad para cinta constituye un dispositivo ideal para hacer copias de seguridad, permitiendo almacenar varios Gigabytes de información en una sola cinta. (Miguel, 2000)

A continuación se mencionan algunos de los dispositivos de salida, los cuales permiten representar los resultados (salida) del proceso de datos. El dispositivo de salida típico es la pantalla o monitor. Otros dispositivos de salida son: impresoras (imprimen resultados en papel), bocinas entre otros. (Miguel, 2000)

Monitor: Dispositivos de salida más comunes de las computadoras con el que los usuarios ven la información en pantalla. (Miguel, 2000)

Impresoras: Como indica su nombre, la impresora es el periférico que el ordenador utiliza para presentar información impresa en papel. (Miguel, 2000)

1.8. Técnicas para la E/S.

- E/S Programada.

El procesador emite la orden de E/S. El proceso espera a que termine la operación antes de continuar.

- E/S dirigida por Interrupciones.

La instrucción de E/S es emitida. El procesador continúa ejecutando instrucciones y el módulo de E/S envía un interrupción cuando está prepara para transmitir.

- Acceso directo a la memoria (DMA).

Es un circuito integrado dedicado que puede enviar y recibir datos más rápido que el microprocesador. Existe un módulo de acceso directo a la memoria que controla el intercambio de datos entre la memoria principal y el Dispositivo de E/S. En esencia un DMA es un microcontrolador especializado optimizado para transferir bloques de datos de un lugar a otro. Permite acceder a la memoria del sistema para leer o escribir independientemente de la CPU principal. Muchos sistemas hardware utilizan DMA, incluyendo controladores de unidades de disco, tarjetas gráficas y tarjetas de sonido.

DMA es una característica esencial en todos los ordenadores modernos, ya que permite a dispositivos de diferentes velocidades comunicarse sin someter a la CPU a una carga masiva de interrupciones. Una transferencia DMA consiste principalmente en copiar un bloque de memoria de un dispositivo a otro. En lugar de que la CPU inicie la transferencia, la misma se lleva a cabo por el controlador DMA. Un ejemplo típico es mover un bloque de memoria desde una memoria externa a una interna más rápida. Tal operación no ocupa el procesador y como resultado puede ser planificado para efectuar otras tareas. Algunos dispositivos de entrada/salida envían datos a la memoria más rápido de lo que el

microprocesador puede manejar. Luego, los dispositivos como discos ópticos y magnéticos utilizan este integrado para acceder a la memoria del sistema.

1.9. Análisis de otras soluciones existentes.

Talent Copy V.

Es una herramienta que se encuentra en su versión de desarrollo. Utiliza parte de la memoria RAM como búfer intermedio que posibilita acelerar el proceso de copia. Utiliza su propio algoritmo para determinar el tamaño de un búfer adecuado, y en el primer paso arroja los archivos a copiar dentro de ese búfer. Acto seguido, el disco duro no necesita leer y escribir frenéticamente, sino que se dedica a tomar los datos directamente desde la RAM, para enfocar a sus cabezales en la operación de escritura. Esto hace que los procesos de copia de archivos dentro de una misma unidad registren una reducción de tiempo muy importante, aunque también se puede apreciar una mayor eficiencia de copia entre diferentes discos duros. Los beneficios de la memoria caché del sistema siguen activos con Talent Copy V, por lo que en caso de repetir la misma operación, el resultado será casi inmediato. No es multiplataforma porque no funciona en cualquier plataforma. Fue desarrollado para Windows Vista, pero ahora tiene soporte para las librerías de Windows 7. Una de sus desventajas es que es propietaria. (Pardo, 2009)

1.10. Conclusiones.

En este capítulo se realizó un estudio de los principales conceptos asociados a las tecnologías de acceso a disco lo que posibilitó un mejor entendimiento del tema a tratar. Se identificó la tecnología NAS como más adecuada porque debido a que es un servidor de almacenamiento separado, que se puede conectar fácilmente a la red de una compañía para asistir al servidor de archivos y proporcionar espacio de almacenamiento tolerante a fallas. Por otra parte se analizaron varios aspectos relacionados con los procesos de E/S, permitiendo conocer los DES, las técnicas para la E/S, facilitando una mejor comprensión del dominio del problema. Se analizó la herramienta de similar propósito Talent Copy V lo que aportó una mejor visión para la realización del componente debido a que esta permite acelerar el proceso de copia haciendo uso de un buffer intermedio.

Capítulo 2: Tendencias y tecnologías actuales a desarrollar para el análisis y diseño del componente de acceso a disco mediante buffer.

2.1. Introducción.

En el presente capítulo se hace un análisis de las tendencias y tecnologías que existen en la actualidad a nivel mundial, que pudieran ser útiles en el desarrollo de la propuesta de solución. Se analizan algunas metodologías de desarrollo de software, lenguaje de modelación y algunos de programación, así como las herramientas de modelado y de desarrollo.

2.2. Metodologías de desarrollo de software.

Una metodología de desarrollo de software es un conjunto de pasos y procedimientos que deben seguirse para desarrollar software.

Durante la elección de una metodología de desarrollo de software para el componente se realizó un análisis sobre dos metodologías bien conocidas *Rational Unified Process* (RUP) y *Extreme Programming (XP).* Se tomaron en cuenta las principales características del componente para seleccionar una guía adecuada que conduzca el proceso de desarrollo hacia los objetivos planteados con la mejor calidad y aprovechamiento de recursos.

2.2.1. RUP.

Es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

Sus características permiten que esta metodología sea adaptable a una gran variedad de sistemas para diferentes áreas de aplicación, diferentes tipos de organización y diferentes tamaños de proyecto; lo que se evidencia en la UCI donde ha sido esta la metodología más extendida en proyectos productivos con diferentes características.

Características de RUP

Es un marco de proceso configurable para satisfacer necesidades específicas, lo que se traduce en una guía adaptable a cada situación que se presente en el desarrollo de software, siendo de gran utilidad en la práctica ingenieril. Además implementa las mejores prácticas de desarrollo de software.

Tiene tres características fundamentales:

- Guiado por los Casos de Uso: A partir de aquí los casos de uso guían el proceso de desarrollo ya que los modelos que se obtienen, como resultado de los diferentes flujos de trabajo, representan la realización de los casos de uso.
- Centrado en la Arquitectura: La arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto, por lo que describe los elementos del modelo que son más importantes para su construcción, los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente. RUP se desarrolla mediante iteraciones, comenzando por los CU relevantes desde el punto de vista de la arquitectura.
- Iterativo e Incremental: Aunque pueda parecer que los flujos de trabajo se desarrollan en cascada, RUP propone que cada fase se desarrolle en iteraciones. Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros. Por ejemplo, una iteración de elaboración centra su atención en el análisis y diseño, aunque refina los requerimientos y obtiene un producto con un determinado nivel, pero que irá creciendo incrementalmente en cada iteración. (Jacobson, 2000)

RUP organiza el ciclo de desarrollo de software en cuatro fases y nueve flujos de trabajo (Ver anexo 1), los que posibilitan con su ejecución ir obteniendo desde una concepción inicial de lo que será el sistema a implementar hasta la definición de características concretas y la implementación del propio sistema. Permite controlar y guiar de manera exitosa el trabajo, pues en todo momento del desarrollo puede conocer quién está haciendo qué, cómo y cuando lo realiza, gracias al claro establecimiento de roles, artefactos, actividades y flujos de trabajo. (KRUCHTEN, 2004)

Ventajas de RUP.

Permite realizar un levantamiento exhaustivo de requerimientos. Establece un Flujo de trabajo dedicado completamente a este objetivo que establece la creación de artefactos donde se describe en lenguaje técnico y formal, entendible a clientes y desarrolladores, las especificaciones que deberá cumplir o

satisfacer el software. Posibilita realizar el Análisis y Diseño, tan completo como sea posible, el cual facilita obtener una mejor visión del sistema en su implementación. Mediante la aplicación de patrones de diseño será posible lograr un producto escalable.

2.2.2. XP

Es una de las metodologías utilizadas en los proyectos a corto plazo y equipos pequeños. Como su nombre lo indica, consiste en una programación rápida o extrema donde el usuario final forma parte del equipo de desarrollo, requisito fundamental para alcanzar el éxito del proyecto.

Características de XP.

Se requiere un grupo pequeño de programadores para trabajar con esta metodología entre 2 – 15 personas y estas irán aumentando conforme sea necesario. Combina las que han demostrado ser las mejores prácticas de desarrollo de software, y las lleva al extremo. El desarrollo de software es riesgoso y difícil de controlar. Se harán pruebas todo el tiempo, no sólo de cada nueva clase (pruebas unitarias) sino que también los clientes comprobarán que el proyecto va satisfaciendo los requisitos (pruebas funcionales). Las iteraciones serán radicalmente más cortas de lo que es usual en otros métodos, esto permite beneficiarse de la retroalimentación tan a menudo como sea posible. (Tobón, 2007)

Ventajas XP.

Intenta reducir la complejidad del software por medio de un trabajo orientado directamente al objetivo, basado en las relaciones interpersonales y la velocidad de reacción. Minimiza el riesgo de fallo del proceso por medio de la disposición permanente de un representante competente del cliente a disposición del equipo de desarrollo. La programación se hace en parejas, pero el código pertenece al equipo completo, no a un programador o pareja, de forma que cada programador puede cambiar cualquier parte del código en cualquier momento si así lo necesita, dejándose en todo caso las mejoras orientadas al rendimiento, para el final. Presenta un diseño evolutivo hace que no se le da apenas importancia al análisis como fase independiente, puesto que se trabaja exclusivamente en función de las necesidades del momento. (Tobón, 2007)

La metodología XP se considera como una metodología ágil, en cambio RUP es clasificada como metodología robusta, lo que significa que esta última enfatiza en cada artefacto generado durante el ciclo de desarrollo del software logrando finalmente obtener una traza del mismo. Estas características son

útiles para grandes equipos de desarrollo que se enfrentan a proyectos de gran envergadura pues permite una fluida comunicación entre las partes que intervienen en el desarrollo y establece un lenguaje claro y común para todos. El componente de acceso a disco mediante buffer para acelerar los procesos de E/S es la respuesta a una necesidad surgida en la mayoría de los proyectos del departamento de Señales Digitales. La construcción del mismo será el resultado de la participación de un equipo compuesto por diversos especialistas por lo que se necesita de una guía como RUP que dirija el proceso de desarrollo, la que además permite que si algún miembro del equipo de desarrollo se retira el que se incorpora por él puede seguir fácilmente porque todo está documentado.

2.3. Lenguaje Unificado de Modelado (UML).

El lenguaje de modelado seleccionado es el Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad.

Es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Es un lenguaje de propósito general para el modelado visual y orientado a objetos, que permite una abstracción del sistema y sus componentes, posibilita establecer una serie de requerimientos y estructuras necesarias para plasmar en un sistema de software previo al proceso intensivo de escribir código. Captura decisiones y conocimiento sobre los sistemas que se deben construir (Mora, 2002)

UML ayuda a los usuarios a entender la realidad desde un punto de vista de la tecnología y posibilita que reflexione antes de invertir y gastar grandes cantidades de dinero en proyectos que no estén seguros en su desarrollo.UML, reduce el costo y el tiempo empleado en la construcción de los módulos que construirán el software.

A partir de la investigación realizada se decide para el desarrollo del componente utilizar este lenguaje de modelado. UML posibilita especificar las características que tendrá el sistema y representar gráficamente los modelos de forma que puedan ser comprendidos por todos los desarrolladores. Permite tanto al cliente

como a los desarrolladores tener una representación real del alcance y la factibilidad que puede o no llegar a tener el componente.

La representación en UML de un software está formada por las 4+1 vistas o modelos parciales separados, relacionados entre sí, estas vistas son:

- Vista de casos de uso.
- Vista lógica.
- Vista de procesos.
- Viste de implementación.
- Vista de despliegue.

Para la modelación utilizando UML, se han desarrollado herramientas CASE (Ingeniería de Software Asistida por Computadora) que sirven de gran ayuda a los desarrolladores de un proyecto.

2.4. Herramientas CASE.

Son distintas aplicaciones informáticas con el objetivo de aumentar la productividad y de reducir los tiempos de trabajo y costos. Estas herramientas ayudan en todos los aspectos del ciclo de vida de desarrollo del software. Como es conocido, los estados en el ciclo de vida de desarrollo de un software son: Modelación del Negocio, Requerimientos, Análisis y Diseño, Construcción, Pruebas y Despliegue. Las herramientas CASE son un elemento muy importante, que le permite al administrador de un proyecto informático llevar adelante un proyecto de forma eficaz y eficiente. También es un hecho que estas mismas herramientas, como toda Tecnología de la Información se encuentren en continua evolución y exista además una gran variedad de proveedores y productos, cada uno de ellos con sus diferentes aplicaciones y especificaciones. Actualmente existen varias herramientas CASE, entre ellas, Rational Rose y Visual Paradigm.

2.4.1. Visual Paradigm.

Es una herramienta colaborativa pues soporta múltiples usuarios trabajando sobre el mismo proyecto y a la vez permite realizar el control de versiones, característica que tiene un impacto positivo en proyectos que cuentan con amplios equipos de desarrollo como los del centro GEySED.

Es una herramienta multiplataforma que facilita la diagramación visual y el diseño de proyectos de un sistema. Soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. También proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML. Dentro de sus principales beneficios se encuentran: la navegación intuitiva entre código y el modelo, superior entorno de modelado visual, soporte completo de notaciones UML y diagramas de diseño automático sofisticado.

2.4.2. Rational Rose.

Es una herramienta de modelado visual para el análisis y diseño de sistemas basados en objetos. Se utiliza para modelar un sistema antes de proceder a construirlo. Cubre todo el ciclo de vida de un proyecto:

- Concepción y formalización del modelo.
- Construcción de los componentes.
- Transición a los usuarios y Certificación de las distintas fases. (Rational Software Corporation, 2003)

Ventajas

Esta herramienta permite, crear, ver y modificar los componentes de un modelo. El mismo habilita asistentes, para crear clases y provee plantillas de código. Admite la integración con otras herramientas de desarrollo. Posee sus propios estereotipos, aunque no diferencia las BD multidimensionales de las relacionales. Es altamente reconocido a nivel mundial. Soporta el desarrollo del ciclo de vida completo del software.

Para el modelado del componente se selecciona la herramienta Visual Paradigm por sus características favorables para el trabajo con tecnologías libres. El uso de ella en la presente investigación implica un ahorro de tiempo en capacitación, debido a su utilización por los equipos de trabajo de los proyectos que integran GEySED. Es una herramienta multiplataforma que utiliza lenguaje de modelado UML y soporta las etapas del ciclo de vida completo de desarrollo de software. Posee la capacidad de ejecutarse sobre diferentes sistemas operativos. Es muy sencillo de usar, fácil de instalar y actualizar. Genera código para varios lenguajes. Posibilita la representación gráfica de los diagramas permitiendo ver el sistema desde diferentes perspectivas, como el de componentes, despliegue, secuencia, casos de uso, clase, actividad, estado, entre otros aspectos que lo pone en ventajas con el Rational Rose.

2.5. Lenguaje de Programación.

2.5.1. C++.

Es un lenguaje de programación, diseñado a mediados de los años 1980, como extensión del lenguaje de programación C. Se puede decir que C++ es un lenguaje que abarca tres paradigmas de la programación: la programación estructurada, la programación genérica y la programación orientada a objetos.

Está considerado como un lenguaje potente, debido a que permite trabajar tanto a alto como a bajo nivel, sin embargo es a su vez uno de los que menos automatismos traen, obliga a hacerlo casi todo manualmente al igual que C lo que dificulta mucho su aprendizaje. (Stroustrup, 1998)

Características.

- Programación orientada a objetos: La posibilidad de orientar la programación a objetos permite al programador diseñar aplicaciones desde un punto de vista más cercano a la vida real. Además permite la reutilización del código de una manera más lógica y productiva.
- Portabilidad: Un código escrito en C++ puede ser compilado en diferentes ordenadores y sistemas operativos sin hacer apenas cambios.

- Brevedad: El código escrito en C++ es muy corto en comparación con otros lenguajes, sobretodo porque en este lenguaje es preferible el uso de caracteres especiales que las "palabras clave".
- Programación modular: Un cuerpo de aplicación en C++ puede estar hecho con varios ficheros de código fuente que son compilados por separado y después unidos. Esta característica permite unir código en C++ con código producido en otros lenguajes.
- Velocidad: El código resultante de una compilación en C++ es muy eficiente, por su capacidad de actuar como lenguaje de alto y bajo nivel.

2.5.2. Java.

Fue diseñado como un lenguaje orientado a objetos desde el principio. Los objetos agrupan en estructuras encapsuladas tanto sus datos como los métodos (o funciones) que manipulan esos datos. La tendencia del futuro, a la que Java se suma, apunta hacia la programación orientada a objetos, especialmente en entornos cada vez más complejos y basados en red. Debido a su semejanza con C y C++, y dado que la mayoría de la gente los conoce aunque sea de forma elemental, resulta muy fácil aprender Java. Los programadores experimentados en C++ pueden migrar muy rápidamente a Java y ser productivos en poco tiempo. (Marañón, 1999)

Características

Es un lenguaje que es compilado, generando ficheros de clases pero estas clases son en realidad interpretadas por la máquina virtual de java, siendo la máquina virtual de java la que mantiene el control sobre las clases que se estén ejecutando. Es multiplataforma, el mismo código java que funciona en un sistema operativo, funcionará en cualquier otro sistema operativo que tenga instalada la máquina virtual java. Además es un lenguaje seguro, ya que la máquina virtual, al ejecutar el código, realiza comprobaciones de seguridad, además el propio lenguaje carece de características inseguras, como por ejemplo los punteros. (Salagar, 2005)

Para la realización de este componente se tuvieron en cuenta algunos aspectos como: tiempos de respuesta, robustez, acceso a hardware y seguridad. A partir de aquí se selecciona como lenguaje de programación C++ pues permite interactuar de forma eficiente con hardware como memoria RAM, disco duro, memorias de video que son importantes para el desarrollo de este componente. Además con esta

selección se cumple con las políticas establecidas en el Departamento Señales Digitales de que las aplicaciones de escritorio deben ser realizadas con C++, lo que no implica problemas para los desarrolladores puesto que trabajan con el mismo.

2.6. Entornos de Desarrollo Integrado (IDE).

2.6.1. QT Creator.

Es un excelente IDE multiplataforma (ya fue probado en Windows 7, Windows XP y Linux) para desarrollar aplicaciones en C++ de manera sencilla y rápida. Cuenta con las siguientes características: Posee un avanzado editor de código C++, es resaltado y auto-completado de código y es un soporte para refactorización de código.

Además presenta el reconocimiento de métodos, la facilidad de creación de formularios y amplia documentación On-line, así como un sin fin de opciones que facilitan el desarrollo de cualquier aplicación.

Como su nombre lo indica, está basado en la librería Qt. que no es más que un framework para el desarrollo de aplicaciones multiplataforma. Incluye clases, librerías y herramientas para la producción de aplicaciones de interfaz gráfica en C++ que pueden operar en varias plataformas. Con el mismo se pueden desarrollar ricas aplicaciones gráficas, incluye soporte de nuevas tecnologías como OpenGL, XML, Bases de Datos, programación para redes, internacionalización y mucho más. Las aplicaciones creadas con Qt son muy elegantes, se ven y se operan mejor que las aplicaciones nativas. (Meyer, 2007)

Este entorno tiene una serie de demos que permiten al usuario auxiliarse de estos para su desarrollo, mostrando no solo la aplicación sino también su código. Por otra parte al estar integrado con QT tiene un fuerte peso para el trabajo con medias, procesos, buffer, ficheros, hilos y redes debido a que tiene clases para todos estos aspectos.

2.6.2. NetBeans

NetBeans consiste en un IDE de código abierto y una plataforma de aplicación la cual permite a los desarrolladores crear rápidamente aplicaciones web, empresariales, de escritorio, y móviles utilizando la plataforma Java, al igual que las plataformas PHP, Java Script y C/C++. Está respaldado por una

comunidad de desarrollo muy activa. Una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito en Java, pero puede servir para cualquier otro lenguaje de programación. Existe además un número importante de módulos para extender el NetBeans IDE. Es un producto libre y gratuito sin restricciones de uso. Es un premiado Entorno de Desarrollo Integrado (IDE) disponible para Windows y Linux entre otros.

Después del análisis sobre los entornos de desarrollo se decide utilizar QT Creator con la librería gráfica QT utilizando como lenguaje de programación C++. Se puede decir que es fundamental para el desarrollo del componente acceso disco mediante buffer. Este IDE está en desventaja con NetBeans porque solo se emplea para el desarrollo de aplicaciones de escritorio pero esto no es problema para el desarrollo del componente y aún así para el trabajo con medias está en ventaja por la utilización de QT. Qt Creator es multiplataforma y está diseñado para hacer que el desarrollo en C++ de la aplicación Qt sea más rápido y fácil.

2.7. Conclusiones

En el capítulo se concluye inicialmente que no existe una herramienta o metodología superior, sino que cada una tiene sus propias características que la convierte en la opción factible de utilizar según las peculiaridades del proyecto que se desee desarrollar. Teniendo en cuenta esto se ha determinado que para el proceso de construcción del componente, se utilizará RUP como metodología de desarrollo. Además, luego del estudio realizado quedaron definidas las principales herramientas tanto de modelado como de desarrollo, para darle solución al sistema que se desea realizar, teniendo presente para esta selección que dichas herramientas debían ser libres, fáciles de emplear para el desarrollo de aplicaciones de escritorio y compatible en varias plataformas para que el componente en futuras versiones pueda ser multiplataforma.

Capítulo 3: Propuesta de la solución del análisis y diseño del componente de acceso a disco mediante buffer.

3.1. Introducción.

En este capítulo se realizará el análisis de la solución propuesta con el objetivo de entender a fondo lo que se quiere desarrollar. Para lograr este objetivo se realiza el modelo del dominio para comprender los conceptos más significativos del tema, así como el levantamiento de requisitos funcionales y no funcionales. También se representa el Diagrama de Casos de Uso del Sistema (DCUS) así como la descripción de los mismos. Además se efectúa el diseño de la propuesta de solución, realizando el diagrama de clases del diseño.

3.2. Modelo del Dominio.

Un modelo de dominio captura los tipos más importantes de objetos en el contexto del sistema. Se crea para documentar los conceptos dominantes y el vocabulario del sistema. Identifica las relaciones entre todas las entidades importantes dentro del sistema e identifica generalmente sus métodos y cualidades importantes. Se puede utilizar con eficacia para verificar y para validar la comprensión del dominio del problema entre los varios poseedores de apuestas del grupo del proyecto. Es especialmente provechoso como una herramienta de la comunicación.

3.3. Modelo del Negocio.

El objetivo del modelo del negocio es describir los procesos, existentes u observados, con el propósito de comprenderlos. Se especifican qué procesos del negocio soportará el sistema. Además de identificar los objetos del negocio implicados, este modelo establece las competencias que se requieren de cada proceso: sus actores, sus trabajadores, sus responsabilidades y las operaciones que llevan a cabo. Consiste en el conjunto de elecciones hechas por la empresa y el conjunto de consecuencias que se derivan de dichas elecciones. Las consecuencias rígidas se asocian generalmente a activos intangibles y tendrán una especial importancia en la evaluación del modelo de negocio.

Para lograr el modelo de negocio se debe comprender la estructura y la dinámica de la organización en la cual se va a implantar un sistema, conjuntamente con los problemas actuales de la organización e identificar las mejoras potenciales, asegurar que los consumidores, usuarios finales y desarrolladores

tengan un entendimiento común de la organización y derivar los requerimientos del sistema que va a soportar la organización.

Como el componente que se va a desarrollar es complejo, resulta difícil encontrar procesos de negocio bien estructurados que permitan realizar un modelado completo de los mismos, por lo que se decidió realizar un modelo de dominio. Este permite, describe y enlaza los principales conceptos del entorno donde funcionará el componente, incluyendo sus principales eventos que detalla los conceptos más trascendentales que intervienen en el problema. Además admite hacer un correcto levantamiento de requisitos. Su objetivo es comprender y describir las clases conceptuales más significativas dentro del dominio del problema, lo cual permite definir los procesos más relevantes. Esto ayuda a los usuarios, desarrolladores y clientes a tener un vocabulario común, contribuyendo con la aceleración del desarrollo del componente.

3.4. Diagrama de Modelo del Dominio.

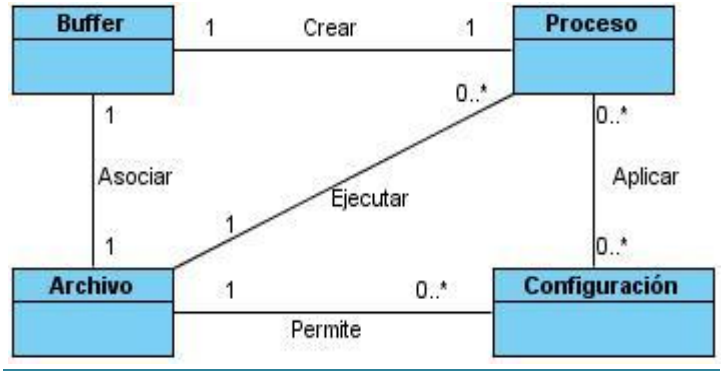


Ilustración 1: Diagrama de modelo del dominio.

3.4.1. Breve descripción del diagrama.

A continuación se muestra una definición de las clases conceptuales del modelo del Domino lo que proporciona un mayor entendimiento de los procesos identificados.

Archivo: Contenedor multimedia que almacena audio, video o imagen.

Buffer: Es el espacio de memoria que se utiliza como regulador y sistema de almacenamiento intermedio.

Proceso: Acción que se ejecuta sobre un archivo.

Configuración: Almacena las configuraciones realizadas sobre cada archivo.

3.5. Especificación de los requisitos de software.

3.5.1. Requisitos Funcionales.

Definen las funciones que el sistema debe ser capaz de realizar. Describen las transformaciones que el sistema realiza sobre las entradas para producir salidas.

RF1- Crear Proceso.

El sistema debe permitir crear procesos para ser aplicados a los archivos.

RF2- Visualizar Proceso.

El sistema debe permitir visualizar todos los procesos en cualquiera de los estados que se encuentre, ya sea iniciado, pausado o no iniciado, además admite iniciar proceso pausado, pausar, cancelar y eliminar proceso.

2.1. Iniciar Proceso Pausado.

El sistema debe permitir iniciar proceso si el estado del mismo es pausado.

2.2. Pausar Proceso.

El sistema debe permitir cancelar proceso si el mismo tiene estado iniciado.

2.3. Eliminar Proceso.

El sistema debe permitir eliminar un proceso una vez que el usuario haya seleccionado el proceso.

RF3- Iniciar Proceso.

El sistema debe permitir iniciar proceso una vez que el usuario haya seleccionado el archivo y el local donde va almacenarlo.

RF4- Cancelar Proceso.

El sistema debe permitir cancelar cualquier proceso que se esté ejecutando.

RF5- Verificar Espacio Libre.

El sistema debe permitir conocer el espacio libre del disco donde se va a realizar el proceso.

3.5.2. Requisitos no Funcionales.

Son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen el producto atractivo, usable, rápido y confiable.

RNF1- Requisito de Usabilidad.

El sistema debe ser sencillo de operar. Todo el proceso debe ser transparente para el usuario, por lo que este no necesariamente debe poseer grandes conocimientos de la informática. Sin embargo debe presentar conocimientos básicos acerca de los procesos que se pueden ejecutar con el componente, dígase: codificación, resumen visual, extracción de fotogramas y otros. Además el sistema deberá estar bajo las normas establecidas de la norma ISO 9241-11, la cual define los parámetros internacionales de usabilidad de cualquier software.

RNF2- Requisito de Disponibilidad.

El componente debe estar disponible las 24 horas del día de los 7 días de la semana.

RNF3- Requisito de Eficiencia.

El tiempo de respuesta estará dado por la cantidad de información a procesar, al igual que la velocidad de procesamiento de la información, la actualización y la recuperación dependerán de la cantidad de información que tenga que procesar la aplicación.

RNF4- Interfaz de hardware.

Se requiere que tengan tarjeta de red (Intel(R) 82566DC Gigabit Network Connection), con memoria RAM de: 512 Mb (mínimo) – 1Gb (recomendado) y un espacio libre en H.D: 750 Mb (mínimo) – 1 Gb (recomendado).

RNF5- Interfaz de software.

El sistema operativo a utilizar es: GNU/Linux aunque según las herramientas utilizadas el componente puede ser usado tanto de Windows como en Linux.

RNF6- Restricciones en el diseño e implementación.

El sistema deberá ser desarrollado como una aplicación de escritorio. Será multiplataforma y libre. Estará implementado en lenguaje C++, utilizando como IDE de desarrollo QT Creator y para la modelación se utiliza UML haciendo uso de la herramienta Visual Paradigm.

3.6. Diagramas de Casos de Uso del Sistema.

A partir de la identificación del actor que interactúa con la aplicación, así como la recopilación del conjunto de funcionalidades escritas en forma de requisitos que a su vez han sido agrupados en casos de uso según sus peculiaridades, se conforma el Modelo de Casos de Uso que se presenta a continuación:

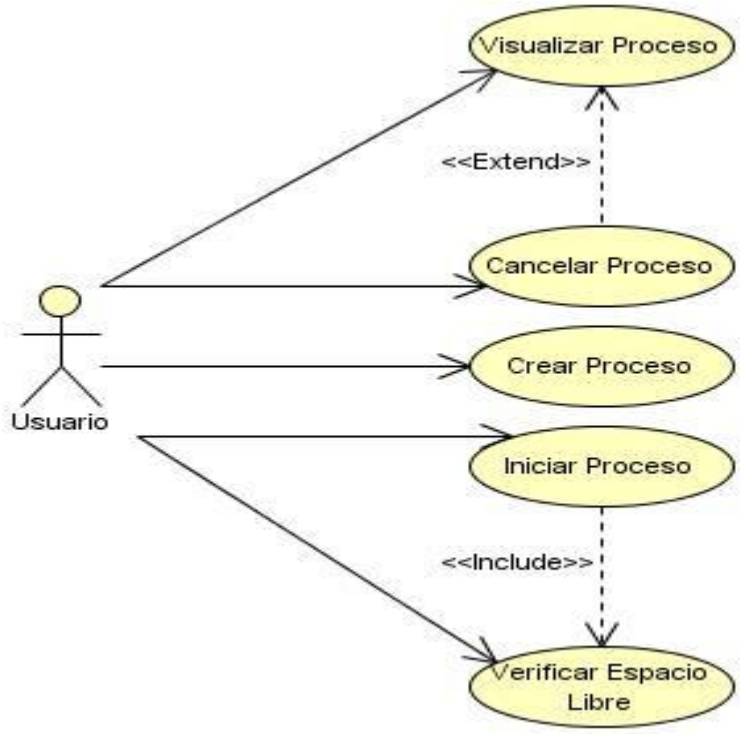


Ilustración 2: Diagrama de CUS.

3.7. Descripción del sistema propuesto. Modelos de Casos de Uso del Sistema.

Luego de haberse realizado la descripción de los requisitos, quienes constituyen las funcionalidades esenciales del sistema, se le da paso a la siguiente etapa de construcción de un software que es la descripción del Modelo de Casos de Uso del Sistema (MCUS). El MCUS describe un sistema en cuanto a su utilización, es un modelo que contiene actores, casos de uso y las relaciones que se establecen entre ellos.

Descripción de los actores.

Actor	Descripción
Usuario	Es el encargado de inicializar los CU, Crear Proceso, Visualizar Proceso, Iniciar Proceso, Cancelar Proceso y Verificar Espacio Libre.

Tabla 1: Descripción de los autores.

3.8. Casos de Uso del Sistema.

Cada uno de los casos de uso identificados presenta características particulares que para su mejor entendimiento se hace necesario describirlas textualmente. Además, para lograr un mejor rendimiento del tiempo en la construcción del software es recomendable identificar los casos de uso arquitectónicamente significativos, que en la descripción textual se diferencian porque tienen la prioridad de crítico.

Caso de Uso:	Crear Proceso.
Actores:	Usuario
Resumen:	El caso de uso se inicia cuando el usuario decide crear procesos.
Precondiciones:	Para crear es necesario que el servicio esté publicado
Referencias	RF1
Prioridad	Crítico

Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Sistema**
1. El usuario selecciona la opción Crear Proceso	1.1. El sistema muestra la interfaz correspondiente a la selección.
2. El usuario introduce los datos necesarios para la creación de un proceso (nombre del proceso y servicio publicado).	2.1. El sistema verifica si se han introducido lo datos correctamente. 2.2. El sistema habilita el botón “Crear”.
3. El usuario presiona el botón “Crear”.	3.1. El sistema muestra un mensaje de notificación de la acción: “El proceso ha sido creado correctamente”.
Flujo Alterno	
Acción del Actor	**Respuesta del Sistema**
Poscondiciones	Se crean los procesos según las condiciones establecidas, es decir si se han introducido lo datos correctamente.
Prototipo de Interfaz	

Tabla 2: Descripción Crear Proceso.

Descripción del Caso de Uso Visualizar Proceso (Ver Anexo 10).

Caso de Uso:	Iniciar Proceso.
Actores:	Usuario
Resumen:	El caso de uso se inicia cuando el usuario decide iniciar proceso una vez que el usuario selecciona el archivo.
Precondiciones:	Debe haber seleccionado el fichero, el origen destino.
Referencias	RF3,RF5
Prioridad	Crítico

Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Sistema**
1. El usuario selecciona la opción Iniciar Proceso	1.1. El sistema muestra la interfaz correspondiente a la selección.
2. El usuario selecciona el archivo al cual se le va aplicar algún proceso	
3. El usuario selecciona el proceso que va a ejecutar	
4. El usuario puede seleccionar si es local o conectar a otro equipo.	
5. En caso de ser local el usuario selecciona la opción local.	5.1. El sistema muestra la interfaz correspondiente a la selección.
6. El usuario selecciona el disco en el cual va a realizar el proceso.	6.1. El sistema habilita el botón Iniciar
7. El usuario presiona el botón Iniciar	7.1. El sistema ejecuta la acción verificar espacio libre, ver CU Verificar Espacio Libre.

	7.2. El sistema crea el buffer a partir del tamaño del archivo seleccionado. 7.3. El sistema calcula el tiempo que demora el proceso en terminar.
	8. El sistema muestra una interfaz del proceso en ejecución, con el tiempo que demora en terminar, y cambia el estado del proceso que estaba en no iniciado a iniciado.
5. En caso de seleccionar conectar a otro equipo el usuario selecciona la opción conectar a otro equipo	5.1 El sistema muestra la interfaz correspondiente a la selección.
6. El usuario escribe la dirección del equipo a conectar	6.1 El sistema habilita el botón Iniciar
7. El usuario presiona el botón Iniciar.	7.1. El sistema ejecuta la acción verificar espacio libre, ver CU Verificar Espacio Libre. 7.2. El sistema crea el buffer a partir del tamaño del archivo seleccionado. 7.3. El sistema calcula el tiempo que demora el proceso en terminar.
	8. El sistema muestra una interfaz del proceso en ejecución, con el tiempo que demora en terminar, y cambia el estado del proceso que estaba en no iniciado a iniciado.
Flujos Alternos	
Acción del Actor	**Respuesta del Sistema**
Poscondiciones	Se inician los procesos según las condiciones establecidas.

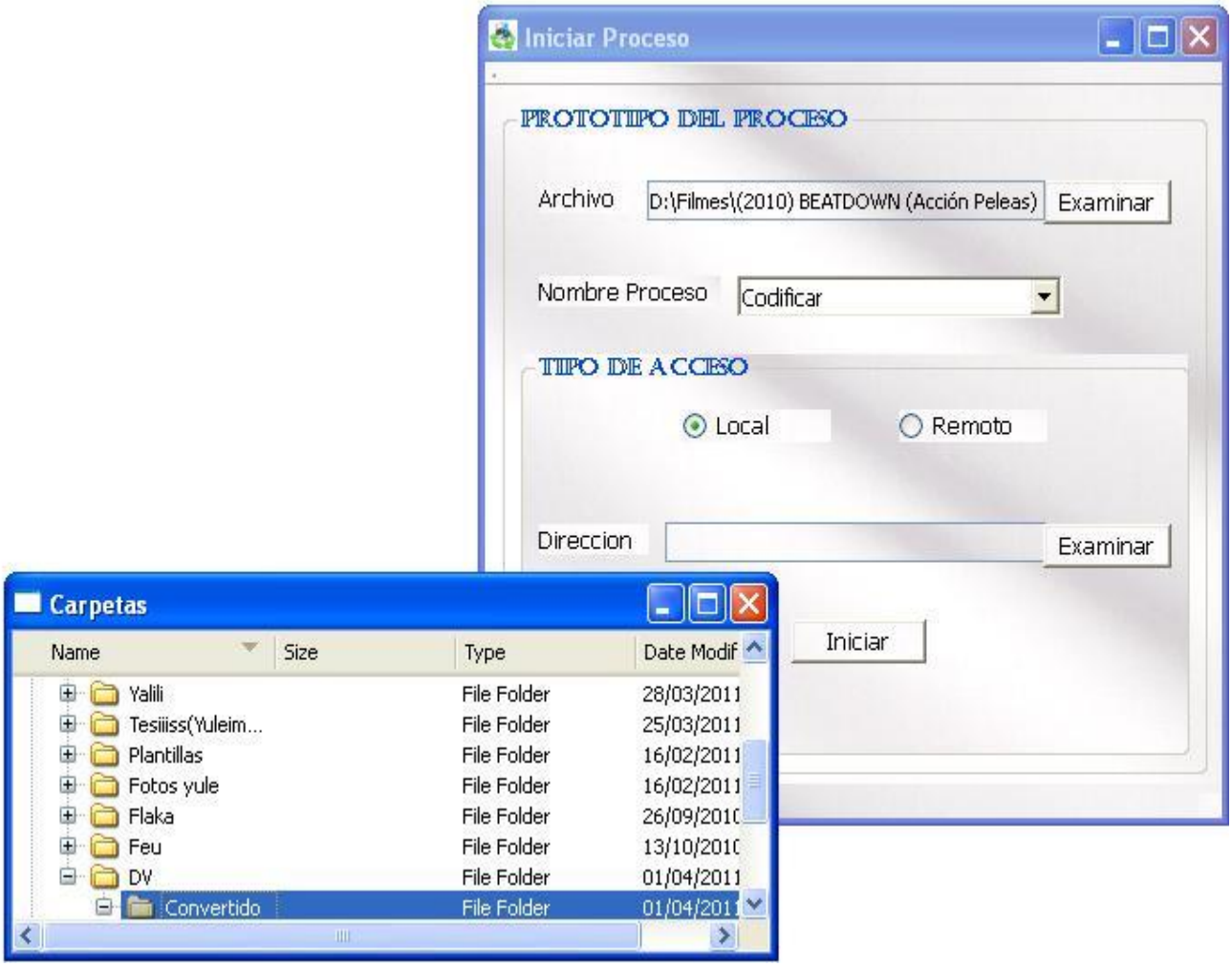
Prototipo de Interfaz.


Tabla 3: Descripción Iniciar Proceso

Descripción de Caso de Uso Cancelar Proceso (Ver Anexo 11).

Caso de Uso:	Verificar Espacio Libre
Actores:	Usuario

Resumen:	El caso de uso se inicia cuando el usuario decide verificar el espacio libre de un disco, antes confirmar si tiene acceso o no.
Precondiciones:	Debe haber seleccionado donde se va a realizar el proceso.
Referencias	RF5
Prioridad	Crítico

Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Sistema**
1. El usuario selecciona la opción Verificar Espacio Libre	1.1. El sistema muestra la interfaz correspondiente a la selección.
2. El usuario selecciona si es local o conectar a otro equipo.	
3. En caso de ser local el usuario selecciona la opción local	3.1. El sistema muestra la interfaz correspondiente a la selección.
4. El usuario selecciona el disco	4.1. El sistema habilita el botón “Aceptar”.
5. El usuario presiona el botón “Aceptar”.	5.1. El sistema verifica el acceso a disco. 5.2. El sistema muestra el siguiente mensaje:”El espacio libre es:”. 5.3. Sistema muestra la interfaz con las propiedades del disco seleccionado.
3. En caso de ser conexión a otro equipo, el usuario selecciona conectar a otro equipo.	3.1. El sistema muestra la interfaz correspondiente a la selección.
4. El usuario introduce los datos del equipo.	4.1. El sistema habilita el botón “Aceptar”.
5. El usuario presiona el botón “Aceptar”.	5.1. El sistema verifica el acceso al disco pidiéndole usuario y contraseña del dominio. 5.2. El sistema muestra el siguiente mensaje:”El espacio libre es:”.

	5.3. Sistema muestra la interfaz con las propiedades del disco seleccionado.
Flujos Alternos	
Acción del Actor	**Respuesta del Sistema**
	5.1. El sistema muestra el siguiente mensaje:" Usted no posee permiso para acceder a este disco". 5.2. El sistema muestra el siguiente mensaje:" No existe espacio libre".
Poscondiciones	Se verifica el espacio libre según las condiciones establecidas.
Prototipo de Interfaz.	

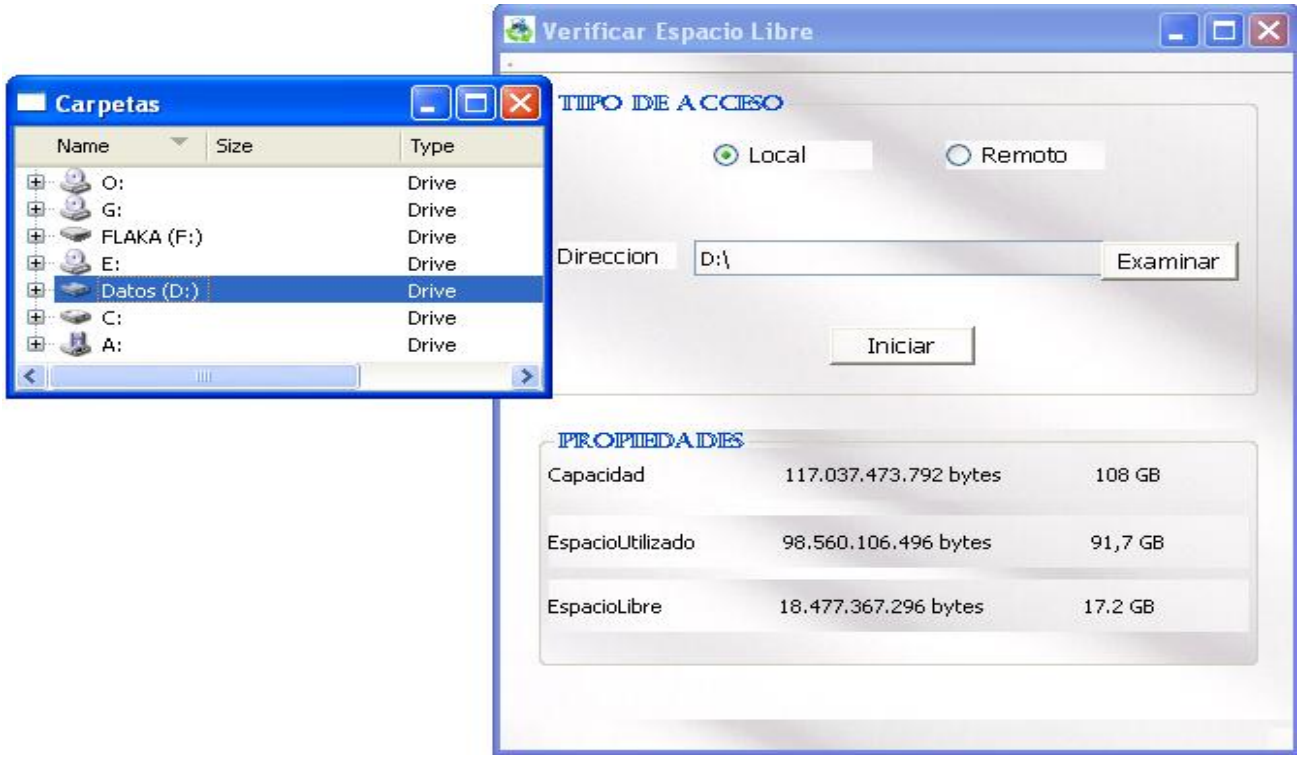


Tabla 6: Descripción Verificar Espacio Libre.

3.9. Técnicas de validación de requisitos

La detección de requisitos suele ser un proceso largo y difícil, para el que se requiere de habilidades psicológicas; una vez detectados todos los requisitos se hace necesario validar los mismos. El objetivo fundamental de la validación de requisitos, es demostrar que realmente estos definen el sistema que se desea desarrollar, garantizando que los resultados obtenidos en la etapa de definición de requisitos son los correctos.

Estas validaciones juegan un papel fundamental, porque si existen errores en el documento que recoge los requisitos, puede traer consigo elevados costos en la producción del software, cuando son descubiertos en el desarrollo o la ejecución del proyecto; ya que algún cambio en un requisito implica cambios o modificaciones en el diseño e implementación. (EVA, 2006)

Existen en el mundo diversas técnicas para la validación de los requisitos como son:

- Prototipos.
- Talleres.
- Reviews o Walk-throughs.
- Auditoria.
- Entrevistas.

3.9.1. Aplicación de las técnicas de validación de requisitos

Durante el período de la validación de los requisitos se tuvieron en cuenta dos técnicas fundamentales, el trabajo en talleres y la creación de prototipo de interfaz de usuario.

El trabajo en talleres favoreció principalmente a aclarar muchas de las funcionalidades del componente y dejar claro el objetivo de cada una de ellas. Para la realización de estos talleres se contó con la presencia del grupo de trabajo SCCM; también se contó con el apoyo de personas con un poco más de experiencia en la parte de los hardware. En cada taller se generaban un grupo de preguntas que favorecían a dar más claridad en las funcionalidades que debía cumplir el mismo, trayendo consigo profundos debates y análisis

detallados de cada uno de los requisitos definidos. Después de cada debate se realizaban las rectificaciones pertinentes y finalmente quedaba aprobado de manera unánime el requerimiento analizado.

La otra técnica empleada fue la realización de prototipos de interfaz de usuario, siendo la técnica que más claridad y solidez le dio a los requisitos identificados. Con la creación de cada prototipo se acercaba más a la realidad del componente y las funcionalidades que realmente se pretendían desarrollar; se comparaba cada requisito detectado con las funcionalidades reflejadas en cada una de las interfaces realizadas y a partir de ahí se realizaba un análisis de la importancia y la robustez del requisito (Ver Anexo 12).

3.10. Técnicas y herramientas de estimación.

Dentro de la ingeniería de Requisitos una de las actividades que requiere vital importancia es la Estimación, pues esta ayuda a reducir los costos y a su vez aumenta los niveles de calidad y servicio. Por lo que las malas estimaciones o no estimaciones, son una de las posibles causas del fracaso de los proyectos en desarrollo.

Teniendo en cuenta que en la construcción de un software hay una forma simple de tener una estimación precisa del esfuerzo requerido, existen diferentes técnicas de estimación basadas en métricas que ayudan a obtener un cálculo aproximado de de la estimación del mismo.

Técnicas de estimación

- COCOMO.
- Estimación basada Líneas de Código.
- Estimación basada Puntos de Función.
- Estimación basada Puntos de Casos de Uso.

Herramientas

Existen varias herramientas de estimación que son implementadas sobre la base de estas técnicas. Entre ellas se encuentra ESTIMAC, que es la herramienta seleccionada para realizar el cálculo del esfuerzo requerido para la realización del proyecto. ESTIMAC utiliza como técnica "Puntos de caso de Uso". La misma es multiplataforma, implementada en la universidad, con una interfaz sencilla y fácil de utilizar tanto para los usuarios inexpertos como para usuarios con conocimiento computacionales. La técnica de

estimación “Puntos de Casos de Uso”, se basa en los Puntos de CU sin Ajustar y en los Puntos de CU Ajustados, donde a través de ambos se calcula el esfuerzo y la estimación final del proyecto.

En los Puntos de CU sin Ajustar se determina la complejidad de los actores del sistema y de los casos de uso en: Simples, Medios y Complejos, tomando cada uno un factor de peso y los Puntos de CU Ajustados se determina un valor de 0-5 el factor de complejidad técnica y el factor ambiente.

Teniendo en cuenta que el esfuerzo total del proyecto es de 2834,46, la jornada laboral del proyecto es de 8 horas-hombres, en la semana se trabajan 40 horas-hombres y un mes tiene como promedio 192 horas laborales. Si trabajan cinco personas 40 horas-hombre como promedio a la semana, el proyecto puede terminar aproximadamente 14,1723 semanas. Esto equivale a 3,54307 meses (de 3 a 4 meses) (Ver Anexo 13).

Diagrama de clase del análisis del CU Crear Proceso.

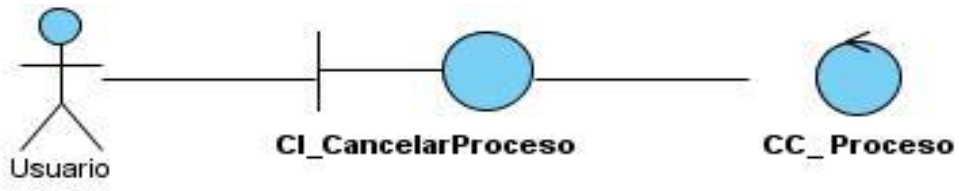


Ilustración 3: Diagrama Crear Proceso.

Diagrama de clase del análisis del CU Visualizar Proceso.

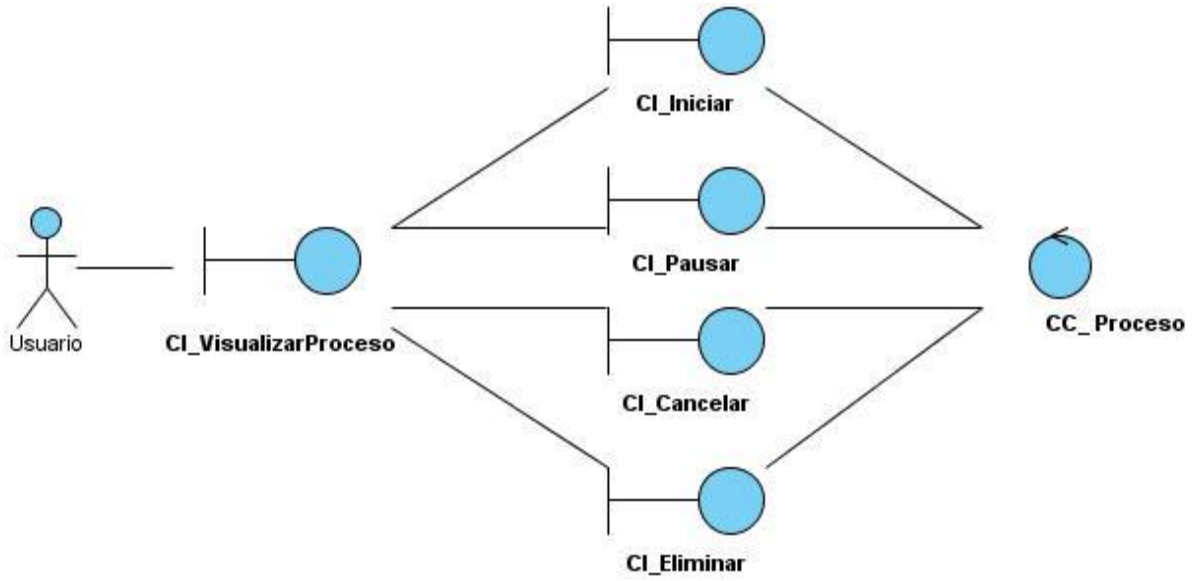


Ilustración 4: Diagrama Visualizar Proceso.

Diagrama de clase del análisis del CU Iniciar Proceso (Ver anexo 2).

Diagrama de clase del análisis del CU Cancelar Proceso (Ver anexo 3).

Diagrama de clase del análisis del CU Verificar Espacio Libre (Ver anexo 4).

Diagrama de colaboración Crear Proceso.

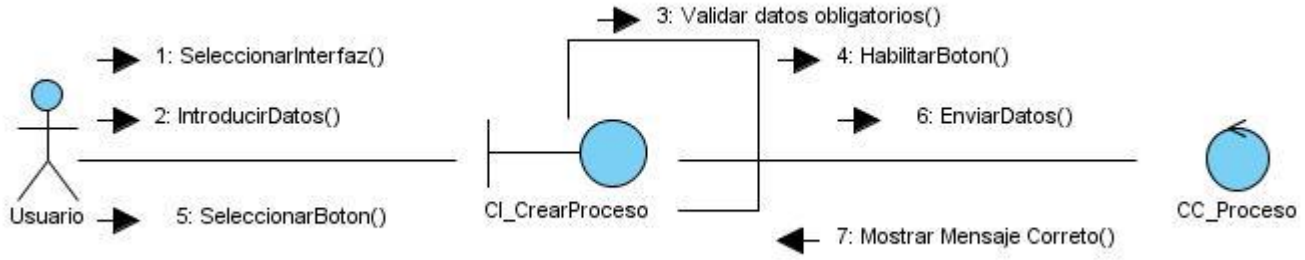


Ilustración 5: Diagrama de colaboración Crear Proceso.

Diagrama de colaboración Visualizar Proceso Iniciar.

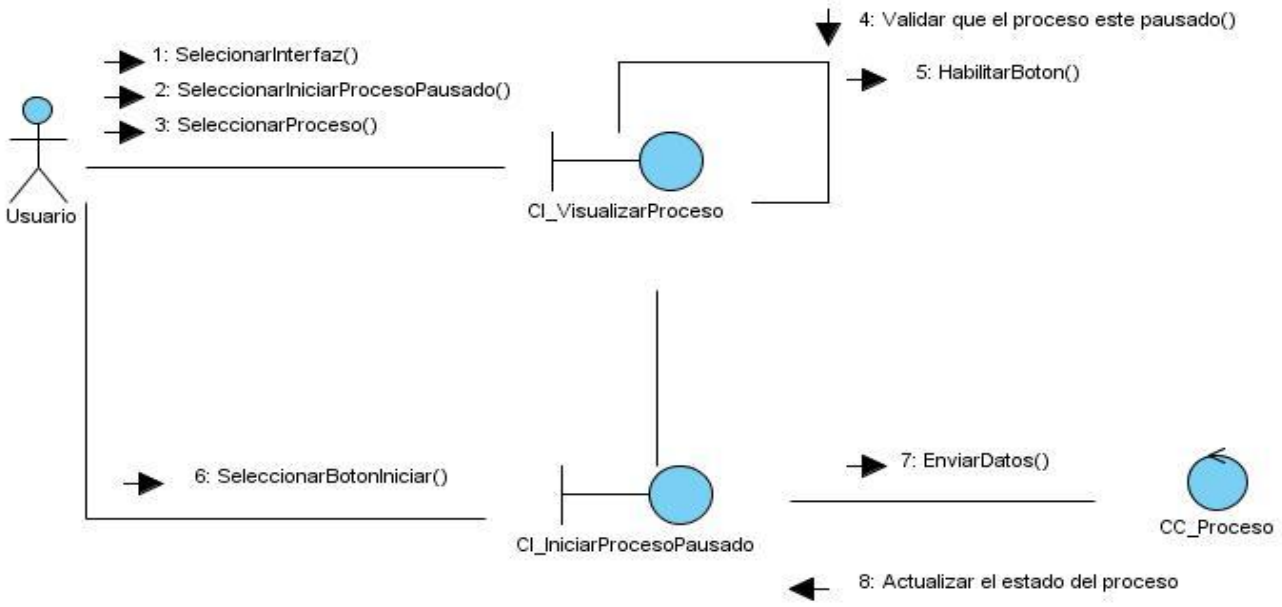


Ilustración 6: Diagrama de colaboración Visualizar Proceso Iniciar.

Diagrama de colaboración Visualizar Proceso Pausar.

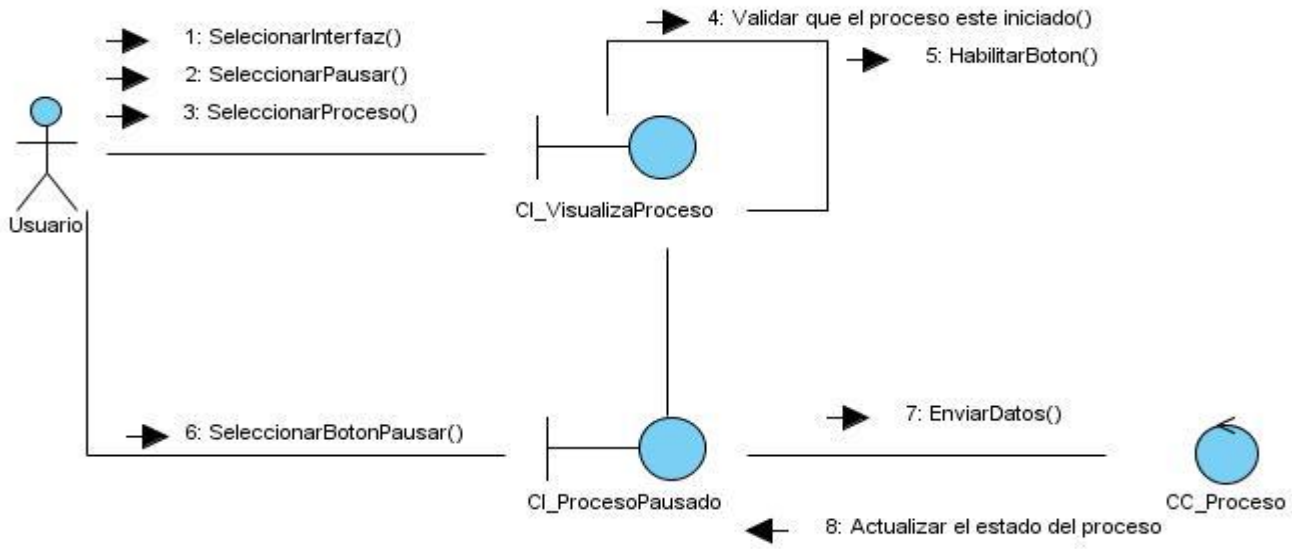


Ilustración 7: Diagrama de colaboración Visualizar Proceso Pausar.

Diagrama de colaboración Visualizar Proceso Cancelar.

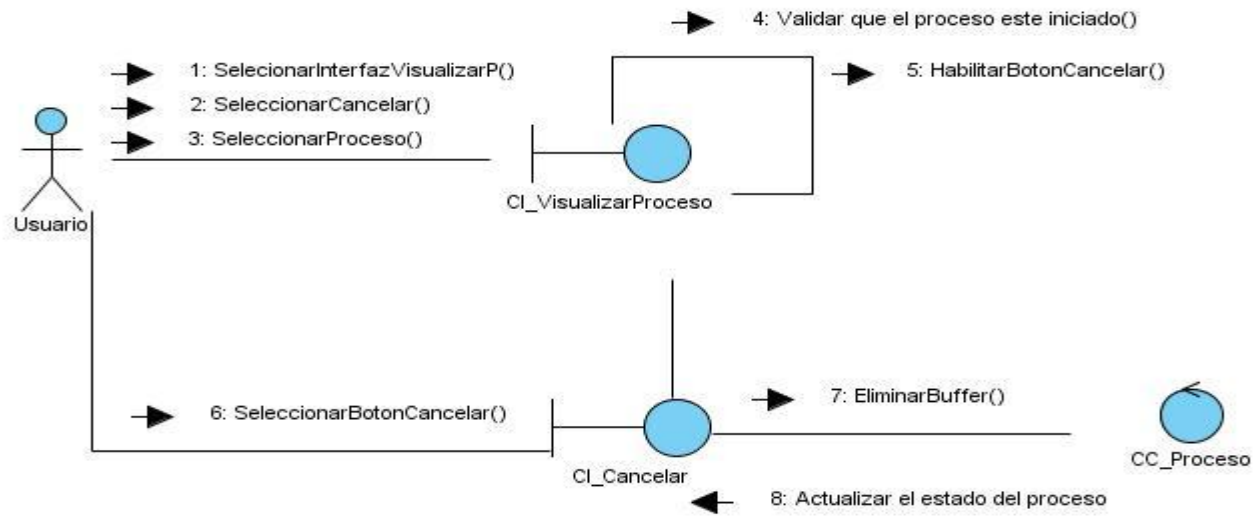


Ilustración 8: Diagrama de colaboración Visualizar Proceso Cancelar.

Diagrama de colaboración Visualizar Proceso Eliminar.

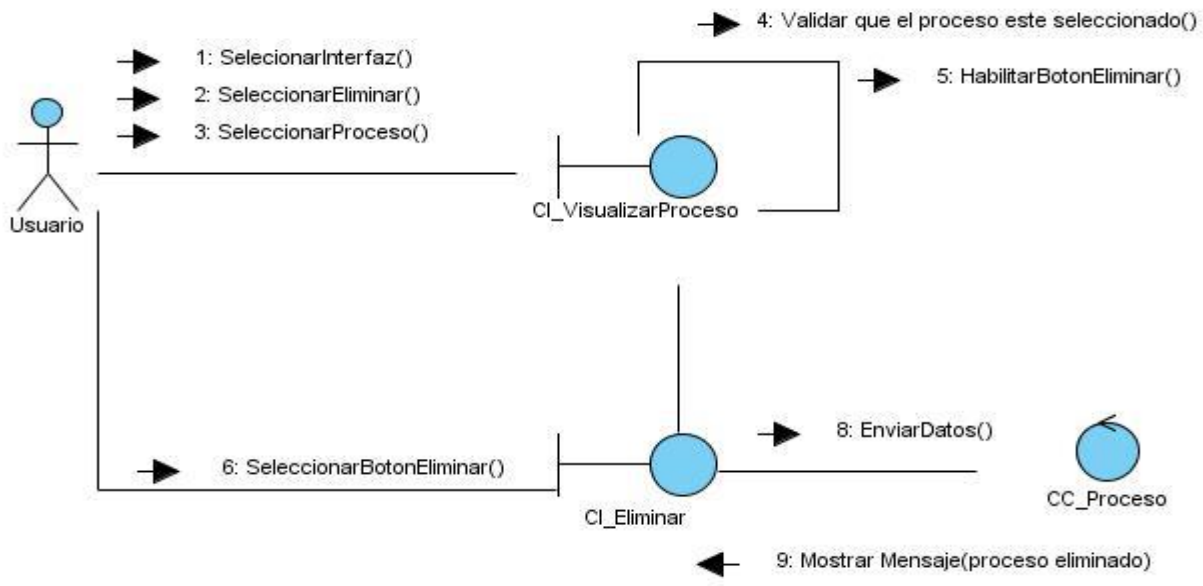


Ilustración 9: Diagrama de colaboración Visualizar Proceso Eliminar.

Diagrama de colaboración Iniciar Proceso Local (Ver anexo 5).

Diagrama de colaboración Iniciar Proceso Remoto (Ver anexo 6).

Diagrama de colaboración Cancelar Proceso (Ver anexo 7).

Diagrama de colaboración Verificar Espacio Libre Local (Ver anexo 8).

Diagrama de colaboración Verificar Espacio Libre Remoto (Ver anexo 9).

A continuación se efectúa el diseño de la propuesta de solución, realizando el diagrama de clase del diseño correspondiente, teniendo en cuenta la arquitectura y los patrones de diseño definidos para la misma.

3.11. Arquitectura

La Arquitectura del Software es el diseño de más alto nivel de la estructura de un sistema. Establece los fundamentos para que analistas, diseñadores, programadores, etc. trabajen en una línea común que permita alcanzar los objetivos del sistema de información, cubriendo todas las necesidades. Es el resultado de ensamblar un número de elementos arquitectónicos de forma adecuada para satisfacer la mayor funcionalidad y requerimientos de desempeño de un sistema, así como requerimientos no funcionales, tales como: confiabilidad, escalabilidad, portabilidad, y disponibilidad. Un diseño correcto de la Arquitectura del sistema es esencial para el éxito o fracaso del proyecto. (Guglielmetti, 2005)

Los patrones son fundamentales para la arquitectura, sus características principales: son la reusabilidad y compatibilidad, en la tecnología basada en componentes también se requiere robustez ya que los componentes han de operar en entornos mucho más heterogéneos y diversos.

Para el desarrollo del componente se decidió utilizar la arquitectura en capas, definiéndose la utilización de dos capas.

Capa de presentación: Es la que ve el usuario, conocida como interfaz gráfica y debe tener la característica de ser amigable, entendible y fácil de usar. Esta capa se comunica únicamente con la capa de negocio. Tiene un fuerte peso el uso del framework QT, pues su principal uso es para el desarrollo de interfaces gráficas. En la misma se utilizaron algunos componentes como son: MainWindows, Layout y Widget.

Capa de negocio: Es donde residen los programas que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se denomina capa de negocio porque es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados. En ella se utilizan clases importantes de Qt. como son: QBuffer, QProcess y QFile.

3.12. Patrones de diseño

Los patrones de diseño han adquirido gran popularidad entre investigadores y diseñadores de software orientado a objeto porque son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y aplicables a distintos problemas típicos de diseño que pueden encontrarse en diferentes contextos. (Larman, 1999)

Para que una solución sea considerada un patrón debe poseer ciertas características, una de ellas es que debe haber comprobado su efectividad resolviendo problemas similares en ocasiones anteriores, otra es que debe ser reusable, lo que significa que es aplicable a diferentes problemas de diseño en distintas circunstancias.

Para un mejor diseño del componente se tuvo en cuenta el estudio de varios patrones como:

3.13. Patrones GRASP (Patrones de Software para la asignación General de Responsabilidad).

Los patrones GRASP describen los principios fundamentales de diseño de objetos para la asignación de responsabilidades. Constituyen un apoyo para la enseñanza que ayuda a entender el diseño de objeto esencial y aplica el razonamiento para el diseño de una forma sistemática, racional y explicable.

Los patrones de asignación de responsabilidades usado para el desarrollo del componente son:

Patrón Experto: Es el principio básico de asignación de responsabilidades. Indica, que la responsabilidad de la creación de un objeto o la implementación de un método, debe recaer sobre la clase que conoce toda la información necesaria para crearlo. De este modo se obtendrá un diseño con mayor cohesión y así la información se mantiene encapsulada (disminución del acoplamiento). (Visconti, 2004)

Patrón Creador: Plantea la necesidad de asignarle a una clase la responsabilidad de crear una instancia de otra clase siempre y cuando agregue los objetos de la clase, los contenga, registre las instancias de estos objetos y los utilice específicamente.

En el desarrollo de componente se puede apreciar que la clase CI_Principal posee la información necesaria para inicializar las clases CI_CrearProceso, CI_VisualizarProceso, CI_IniciarProceso y CI_EspacioLibre, conjuntamente la clase CI_Principal agrega objetos de las clases CI_CrearProceso, CI_VisualizarProceso, CI_IniciarProceso y CI_ EspacioLibre. (Visconti, 2004)

Controlador: Asignar la responsabilidad de controlar el flujo de eventos del sistema, a clases específicas. Esto facilita la centralización de actividades (validaciones, seguridad, etc.). El controlador no realiza estas actividades, las delega en otras clases con las que mantiene un modelo de alta cohesión. Un error muy común es asignarle demasiada responsabilidad y alto nivel de acoplamiento con el resto de los componentes del sistema. (Visconti, 2004)

Bajo acoplamiento: Acoplamiento bajo significa que una clase no depende de muchas clases, ya que uno de los principios para protegerse frente a los cambios es mantener bajo el acoplamiento entre variedades. Resulta evidente que, cuanto menor sea el acoplamiento entre clases, menor influencia tendrán los cambios.

Un cambio en una clase Principal, que depende de otras clases será mucho más complejo que si la misma clase Principal no dependiera de ninguna otra y, por lo tanto, el cambio sólo le afectará a ella. Mantener bajo el acoplamiento entre clases, más que un patrón que se pueda implementar, es un principio que sirve para elegir entre alternativas de diseño: un diseño menos acoplado siempre será mejor que uno más acoplado, porque en el primero será más fácil realizar cambios. Alguno de los beneficios de bajo el acoplamiento: no se afectan por cambios de otros componentes, fáciles de entender por separado, fáciles de reutilizar. (Visconti, 2004)

Alta cohesión: Expresa que se debe asignar una responsabilidad de modo que la cohesión siga siendo alta, la cohesión no es más que la medida de cuán relacionadas y enfocadas están las responsabilidades de una clase.El patrón de Alta Cohesión, más que un diseño directamente implementable en código, se trata de un principio director que nos guiará en el diseño. Una clase estará más cohesionada cuanto más enfocado sea su comportamiento. Es decir, al asignar responsabilidades en nuestro diseño, buscaremos

soluciones que asignen los métodos a las clases de forma coherente, completa y relacionada. De esta forma, obtendremos clases cohesionadas. (Visconti, 2004)

Las ventajas son evidentes. Una clase cohesionada facilita el cambio (objetivo principal de los patrones de diseño). Al realizar un cambio en una clase muy cohesionada, todos los métodos que pueden verse afectados, toda la información que necesitamos controlar, estará a la vista, en el mismo fichero. Este patrón se relaciona con el de Bajo Acoplamiento, porque un diseño cohesionado tendrá un bajo acoplamiento entre clases y, normalmente un diseño con bajo acoplamiento estará formado por clases muy cohesionadas. (Visconti, 2004)

Diagramas de Clases del Diseño

Representa las clases que serán utilizadas dentro del sistema y las relaciones que existen entre ellas. Sirve para visualizar las relaciones entre las clases que involucran el sistema, las cuales pueden ser asociativas, de herencia, de uso y de convencimiento. Un diagrama de clases está compuesto por los siguientes elementos: Clase: atributos, métodos y visibilidad. Relaciones: Herencia, Composición, Agregación, Asociación y Uso.

A partir de los patrones diseño explicados anteriormente se define el siguiente modelo del diseño:

Diagrama de Clase del Diseño: Capa de Presentación

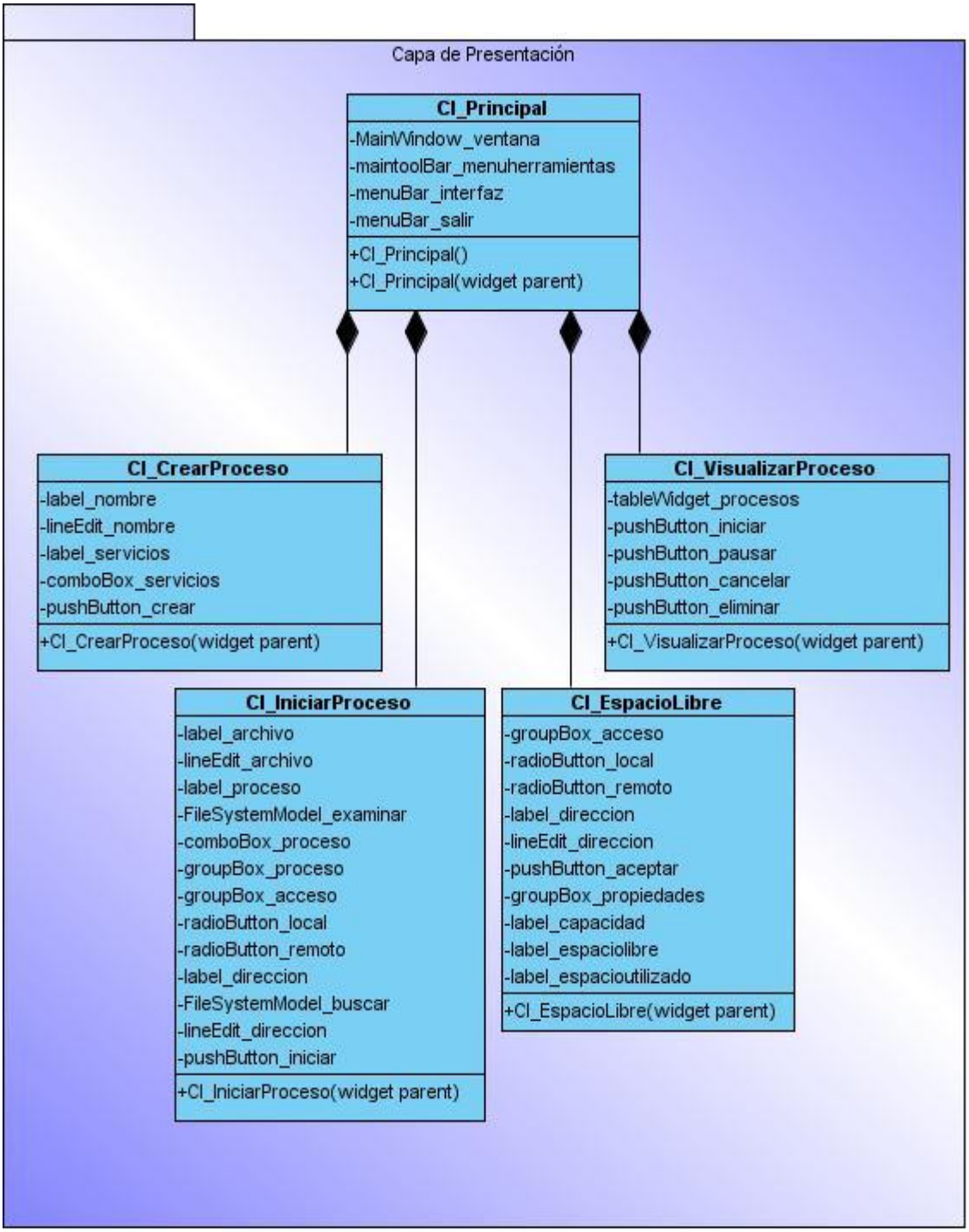


Ilustración 10: Capa de Presentación.

Diagrama de Clase del Diseño: Capa de Negocio

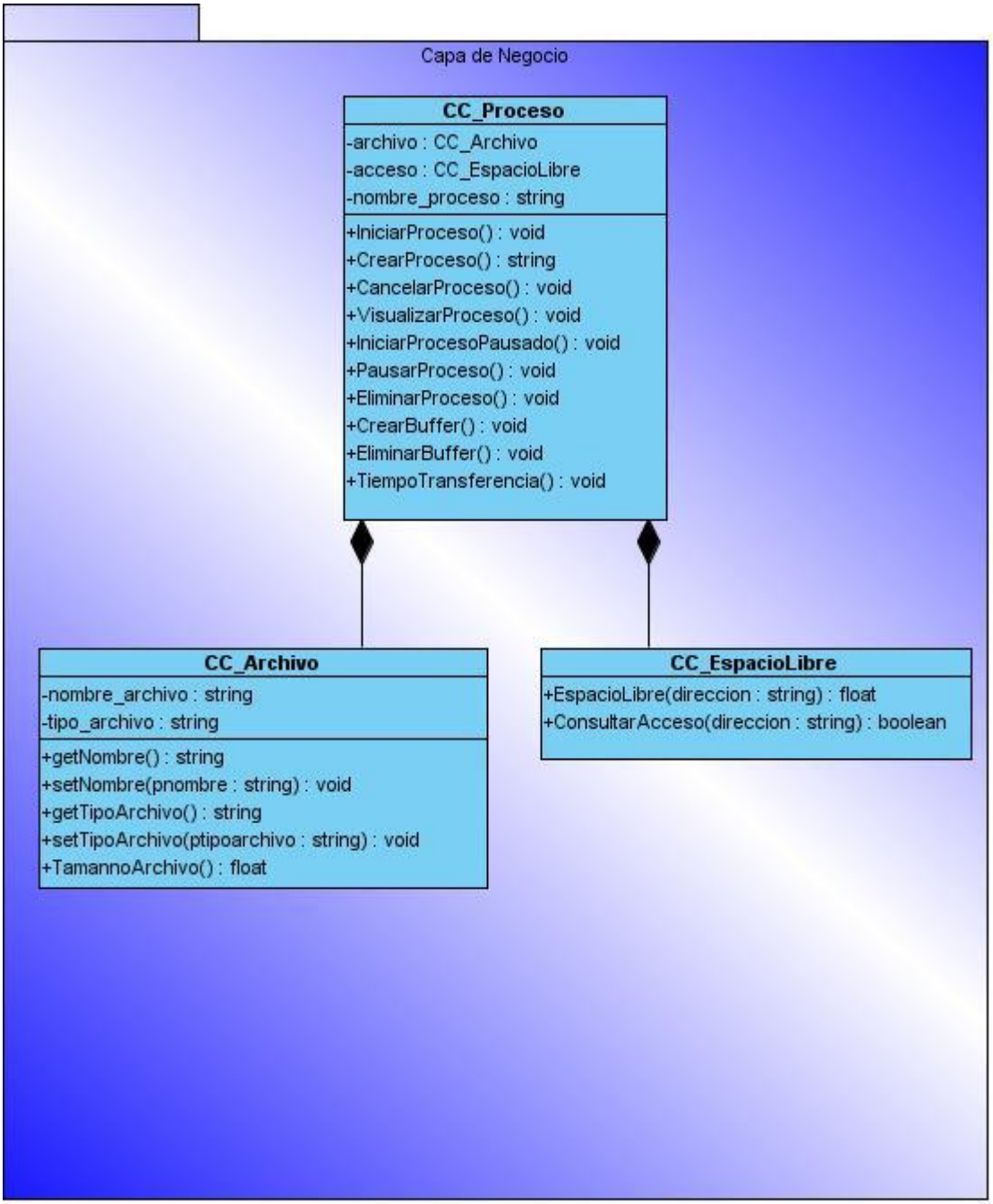


Ilustración 11: Capa de Negocio.

Diagrama de Clase del Diseño General.

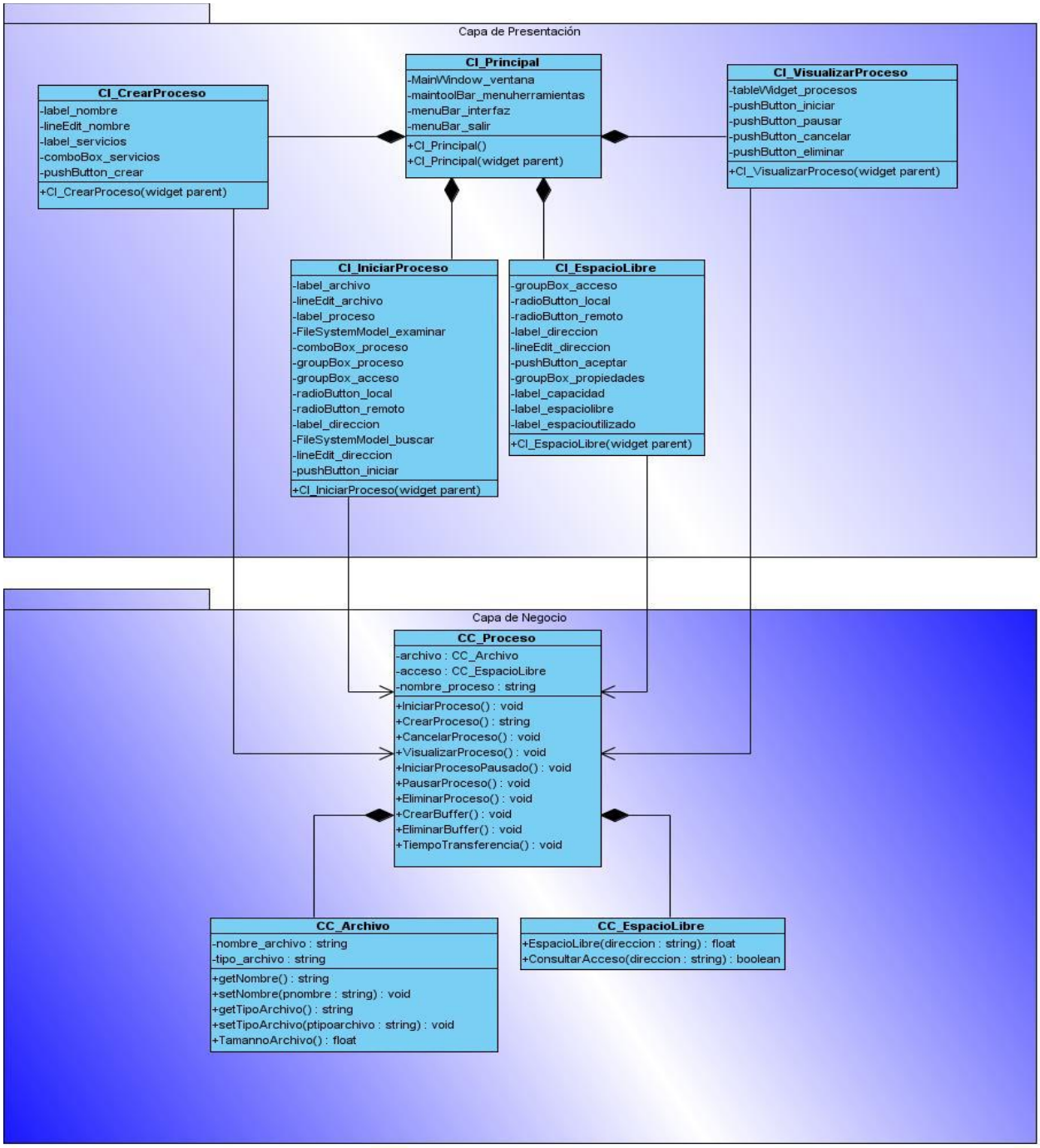


Ilustración 12: Capa General.

Se puede decir que las relaciones de composición y agregación son significativas en el diagrama de clases del diseño, a continuación se hará una breve descripción de las mismas.

Estas relaciones son unidireccionales, implica que una de las clases tendrá una o varias instancias de la otra solamente. La forma de representar la dirección en el diagrama de clases es mediante un rombo hacia la clase contenedora.

En la confección de este diagrama de clase se utilizó la relación de composición y asociación. A continuación se hará una breve explicación de en qué consiste la misma dentro del diseño.

Este diseño está compuesto por dos capaz, de presentación y negocio. Se identificó que la relación presente en la capa de presentación es de composición, es decir que los objetos de las clases CI_CrearProceso, CI_VisualizarProceso, CI_IniciarProceso y CI_ EspacioLibre los cuales son dependientes de la existencia de los objetos de la clase CI_Principal, es decir, se establece una relación fuerte entre ellos, que implica que el tiempo de vida de un objeto de la clase componente está limitado por la existencia del objeto compuesto, además la clase compuesta tiene la responsabilidad de la creación y destrucción de sus objetos componentes.

Esta capa se comunica con la de negocio con la relación de asociación esta recibe las solicitudes y presenta los resultados, además en el diseño de esta capa se utilizó la relación de composición ya que los objetos de las CC_ EspacioLibre y CC_Achivo son dependientes de la existencia de los objetos de la clase CC_Proceso es decir que el tiempo de vida de un objeto de la clases CC_ EspacioLibre y CC_Achivo está limitado por la existencia del objeto de la clase CC_Proceso.

A continuación se detallan los algoritmos de planificación de disco.

3.14. Algoritmos de Planificación de Discos:

Nombre	Descripción	Comentarios
Selección en función del demandante:		
RSS	Planificación Aleatoria	Para análisis y simulación
FIFO	Primero en entrar, primero en salir.	El más justo de todo

PRI	Prioridad del Proceso	El control se lleva aparte de la gestión de la cola del disco.
LIFO	Ultimo en entrar, primero en salir.	Maximiza la utilización de recursos y aprovecha la cercanía.
Selección en función del elemento solicitado.		
SSTF	Primero el más cortó.	Gran aprovechamiento y colas pequeñas.
SCAN	Recorrer el disco de un lado a otro.	Mejor distribución del servicio.
C-SCAN	Recorrer el disco en un solo sentido.	Menor variedad en el servicio.
SCAN de N	SCAN de N registros a la vez.	Garantía de servicio.
FSCAN	SCAN de N pasos, con N=longitud de la cola al comienzo del ciclo del SCAN.	Sensible a la carga.

Tabla 4: Algoritmo de Planificación de discos

FIFO: Primero en entrar, primero en salir, lo que significa que los elementos se procesan de la cola en el orden secuencial. Esta estrategia tiene como ventaja de ser justa por que las peticiones son servidas en el orden que llegaron. (SIERRA, 1990)

PRI: Con sistema de prioridad (PRI), el control de la planificación queda asilado del control del software gestor del disco. Este enfoque no persigue la optimización del uso del disco, sino cumplir con otros objetivos del sistema operativo. Los trabajos por lotes que sean cortos y los trabajos interactivos reciben frecuentemente una prioridad más alta que trabajos mayores que realizan largas operaciones. Esta práctica permite que el sistema haga salir más rápidamente a muchos trabajos cortos y pueda proporcionar un buen tiempo de respuesta interactiva. Sin embargo, los trabajos mayores pueden tener que esperar excesivamente. (SIERRA, 1990)

LIFO: Sorprendentemente, la política de tomar siempre la petición más reciente tiene alguna virtud. En los sistemas de proceso de transacciones, conceder el dispositivo al último usuario acarrea pocos o nulos movimientos del brazo al recorrer un fichero secuencial. El provecho de esta cercanía mejora la productividad y reduce la longitud de las colas. A medida que un trabajo utiliza de forma activa el sistema de archivos, va procesándose tan rápido como es posible. Sin embargo, si el disco está ocupado con una carga de trabajo larga, existe la posibilidad inconfundible de inanición. Una vez que un trabajo ha lanzado una petición de E/S a la cola y haya abandonado la cabeza, no podrá volver a ganar la cabeza a menos que se vayan todos los que estén por delante. (SIERRA, 1990)

SSTF: La política del primero más cortó, es elegir la solicitud de E/S a disco que requieren el menor movimiento posible del brazo del disco desde su posición actual. De este modo, se elige procurando el mismo tiempo de búsqueda no garantiza que sea mínimo el tiempo medio de búsqueda entre una serie de movimientos. Sin embargo esta elección debe ofrecer el rendimiento mejor que el FIFO. Como el brazo debe moverse en ambos sentidos, se puede usar un algoritmo aleatorio de desempate para resolver los cosos de igualdad de distancia. (SIERRA, 1990)

SCAN: El brazo sólo se puede mover en un sentido, resolviendo todas las peticiones pendientes de su ruta, hasta que alcance la ultima pista o hasta que no haya más peticiones en esa dirección. Esta última puntualización se conoce a veces como la política de LOOK. Se cambia entonces la dirección de servicio y el rastreo sigue en sentido opuesto, volviendo a recoger todas las peticiones en orden.

C-SCAN: La política del C-SCAN restringe el rastreo a una sola dirección. Así, cuando se haya visitado la ultima pista en un sentido, el brazo vuelve al extremo opuesto del disco y comienza a recorrerlo de nuevo, lo que reduce el retardo máximo sufrido por las nuevas peticiones. (SIERRA, 1990)

SCAN de N: La política del SCAN de N pasos divide la cola de peticiones del disco en subcolas de longitud N. Las subcolas se procesan una a una mediante un SCAN. Mientras se procesa una cola, se añadirán nuevas peticiones a las otras. Si hay menos de N peticiones disponibles al final del rastreo, entonces todas serán procesadas en el siguiente recorrido. Para valores grandes de N, el rendimiento del SCAN de N pasos se aproxima al del SCAN; con un valor de N = 1, se está adoptando la política FIFO. (SIERRA, 1990)

FSCAN: La política FSCAN emplea dos subcolas. Cuando comienza un rastreo, todas las peticiones están en una de las colas y la otra permanece vacía. Durante el recorrido, todas las peticiones nuevas se

colocan en la cola que inicialmente estaba vacía. De este modo, el servicio de nuevas peticiones se retrasará hasta que se hayan procesado las viejas. (SIERRA, 1990)

El desarrollo del componente de acceso a disco mediante buffer permite acelerar los procesos E/S, garantizando un mejor rendimiento en los productos del departamento de Señales Digitales. Con el mismo se puede aumentar la velocidad y reducir el tiempo de ejecución de los procesos a realizar sobre los materiales, esto permite que el tiempo de respuesta de los productos del departamento sea más rápido propiciando un mejor resultado para los clientes.

A continuación se muestra la propuesta del algoritmo de acceso a disco mediante buffer que permita acelerar los procesos de E/S. Para el diseño del mismo se identifica de forma similar al FIFO, ya que procesa de la cola en el orden secuencial. Esta estrategia tiene como ventaja que no permite la ejecución de varios procesos al mismo tiempo puesto que no da tiempo a liberar la memoria RAM.

1: El algoritmo comienza seleccionando el proceso que desea ejecutar ya sea copiar, codificación, extracción de fotogramas o resumen visual.

2: Se Selecciona el archivo al que se le va a aplicar dicho proceso y la dirección donde se va almacenar.

3: Verificar si hay o no acceso al equipo en el disco especificado.

4: Verificar si hay espacio libre en el disco.

5: Se crear buffer de un tamaño menor a un Megabyte. El tamaño máximo del buffer depende de las características del hardware.

6: El algoritmo termina dando el tiempo que demora el proceso en terminar.

Luego de realizado un estudio no se encontró un pseudocódigo bien definido y estructurado para el desarrollo del componente, ya que debe representar la solución a un algoritmo de la forma más detallada posible, y a su vez lo más parecida al lenguaje que posteriormente se utilizará para la codificación del mismo. Por esta razón se elabora la siguiente propuesta, la cual contiene una serie de condiciones que explican cómo debe ser implementado el componente posteriormente.

1. Inicio.

2. **Si** seleccionó el proceso que desea ejecutar entonces

2.1. Seleccione el archivo al que se le va a aplicar dicho proceso

2.2. Seleccione la dirección donde se va almacenar

si_no

2.3. Bifurcar al paso 5

Fin_si

3. **Si** hay acceso al equipo en el disco especifico entonces

3.1. Verificar si hay espacio libre en el disco.

si_no

3.2. Bifurcar al paso 5

Fin_si

4. **mientras** espacio libre mayor que el tamaño de archivo seleccionado

4.1. Crear buffer de un tamaño menor a un Megabyte y otros similares, basándose en el buffer circular. El tamaño máximo del buffer depende de las características del hardware.

4.2. Calcular el tiempo que demora el proceso en terminar.

fin_mientras

si_no

4.3. Bifurcar al paso 5

5. fin

3.15. Resultados Obtenidos.

Para validar el cumplimiento del objetivo general de la presente investigación, se muestra a continuación una tabla comparativa en la cual se puede observar la diferencia de tiempos que existe al copiar un material audiovisual para un lugar determinado utilizando el copiar del sistema operativo y el del algoritmo propuesto.

Tamaño Buffer	Tamaño Video1	Proceso	Dirección	Tiempo Ejecución antes	Tiempo Ejecución después
2048 Byte	1.7 GB	Copiar	Datos: \	68 segundos	130 segundos
10240 Byte	1.7 GB	Copiar	Datos: \	68 segundos	67 segundos

1048000B	1.7 GB	Copiar	Datos: \	68 segundos	63 segundos
Tamaño Buffer	Tamaño Video2	Proceso	Dirección	Tiempo Ejecución antes	Tiempo Ejecución después
2048 Byte	1.4Gb	Copiar	Datos: \	59 segundos	105 segundos
10240 Byte	1.4Gb	Copiar	Datos: \	59 segundos	55 segundos
1048000 Byte	1.4Gb	Copiar	Datos: \	59 segundos	51 segundos

Tabla 5: Resultados Obtenidos

Después de haber realizado la comparación en cuanto a tiempo de ejecución de los materiales audiovisuales haciendo uso del algoritmo planteado, es posible concluir que la propuesta permite validar el aumento de la velocidad y reducción del tiempo de ejecución de los procesos a realizar sobre los materiales, basado en el uso del buffer. Permite garantizar un mejor acceso al disco duro con la creación del área de almacenamiento temporal en la memoria RAM para mejorar la velocidad a la hora de acceder al dispositivo de almacenamiento. Además se puede decir que entre más grande sea el tamaño del buffer menor será el tiempo de ejecución y la velocidad seria mayor.

3.16. Conclusiones.

En el presente capítulo se hizo un análisis detallado del dominio del problema. Por la complejidad del componente que se va a realizar, porque el flujo de información y los procesos no están bien definidos se decidió realizar modelo del dominio, esto posibilita de una forma más clara definir los requisitos funcionales y no funcionales. Se realizó el diagrama de Casos de Usos del Sistema donde se definieron los CU Crear Proceso, Visualizar Proceso, Iniciar Proceso, Cancelar Proceso, Verificar Espacio Libre, así como la descripción de cada uno de estos. A partir de la descripción de los mismos se confeccionaron los diagramas de clases del análisis y los de colaboración correspondiente a cada uno de ellos. Además se definieron importantes aspectos correspondientes al desarrollo de la solución propuesta, realizando el diagrama de clase del diseño, teniendo en cuenta la arquitectura y los patrones de diseño definidos para la misma, creando de esta forma la entrada para la actividad de implementación futura.

Conclusiones Generales.

Para un mejor entendimiento y compresión de los temas relacionados para el desarrollo de componente se establecieron las bases teóricas y tecnológicas de la solución que se propone. Se plantea como tecnología de acceso para el departamento de Señales Digitales la NAS. Esta potente solución de almacenamiento cuenta con numerosas características entre la que se incluyen capacidad de control de acceso a más 500 usuarios, además puede utilizarse para realizar las copias de seguridad o compartir de forma segura archivos de gran importancia. Se identificaron las herramientas para el modelado y desarrollo del componente, las mismas debían cumplir con una serie de características, como que sean libres, fáciles de emplear para el desarrollo de aplicaciones de escritorio y compatibles en varias plataformas.

Se realizó el modelo de dominio a partir de que los procesos claves no estaban bien definidos, pues es una manera de representar los principales conceptos que se tratan en el entorno del negocio así como las relaciones existentes entre ellos. Se obtuvieron los CUS así como los artefactos que genera la metodología RUP llegando a la propuesta de solución, teniendo en cuenta la arquitectura y los patrones de diseño. Al no existir ningún algoritmo, se hace la propuesta de uno siguiendo el análisis y diseño realizado previamente, al mismo se le hizo una pequeña prueba y los resultados obtenidos fueron satisfactorios, demostrando que cuanto mayor sea el buffer mejor serán los resultados.

Recomendaciones.

Se recomienda realizar la Implementación del componente de acceso a disco mediante buffer.

Continuar el desarrollo de la investigación pero basado en la liberación de la memoria RAM, pues al crear buffer consume mucho recurso.

Realizar la investigación para establecer prioridades en los procesos a ejecutar mediante buffer basándose en la propiedad del framework Qt para establecer prioridades y siguiendo las bases del algoritmo PRI.

Glosario de términos.

E/S (Entrada y Salida): Es la colección de interfaces que usan las distintas unidades funcionales (subsistemas) de un sistema de procesamiento de información para comunicarse unas con otras, o las señales (información) enviadas a través de esas interfaces. Las entradas son las señales recibidas por la unidad, mientras que las salidas son las señales enviadas por ésta.

DES (Dispositivo de entrada y salida): Los dispositivos de entrada y salida son el conjunto de aparatos tecnológicos que usan las distintas unidades de un sistema de procesamiento de información como una computadora para comunicarse unas con otras.

DMA (Acceso directo a la memoria): Permite a cierto tipo de componentes de ordenador acceder a la memoria del sistema para leer o escribir independientemente de la CPU principal. .

DD (Disco Duro): El disco duro es la parte de tu ordenador que contiene la información electrónica y donde se almacenan todos los programas (software). Es uno de los componentes del hardware más importantes dentro de tu PC.

UCP (Unidad Central de procesamiento): Es el componente del computador y otros dispositivos programables, que interpreta las instrucciones contenidas en los programas y procesa los datos.

SAN (Red de área de almacenamiento): Es una tecnología de red de almacenamiento de altas prestaciones. Su función es centralizar el almacenamiento de los ficheros en una red de alta velocidad y máxima seguridad. .

NAS: Almacenamiento Conectado a Red.

RAID: Matriz Redundante de Discos Independientes.

SATA (Serial ATA): Tecnología serial avanzada de contacto.

ATA: Ajunto de tecnología avanzada.

SCSI: Interface de sistema para computadora.

SAS: Interfaz de transferencia de datos en serie.

IDE: Dispositivo con electrónica integrada.

RUP (Proceso Unificado de Rational): Es un proceso para el desarrollo de un proyecto de software que define claramente quien, cómo, cuándo y qué debe hacerse en el proyecto.

XP (Programación Extrema): La programación extrema es una metodología de desarrollo ligera basada en una serie de valores y de prácticas de buenas maneras que persigue el objetivo de aumentar la productividad a la hora de desarrollar programas.

CASE (Ingeniería de Software Asistida por Computadora): representan una forma que permite Modelar los Procesos de Negocios de las empresas y desarrollar los Sistemas de Información Gerenciales.

Bibliografía Referenciada.

BOOCH, GRADY, RUMBAUCH, JAMES y JACOBSON, IVAR. 1997. *The UML specification documents.* Santa Clara : s.n., 1997.

Breidenbach, Walls Craig and. 2005. Eclipse . *Eclipse .* [En línea] 2005. [Citado el: 13 de Enero de 2011.]

CP-Group. 2007. Que son los Discos Rígidos. [En línea] 12 de 11 de 2007. [Citado el: 5 de Mayo de 2011.] http://guia.mercadolibre.com.ar/que-son-discos-rigidos-tamanos-interfaces-todo-necesitas-saber-19752-VGP.

ERP, Equipo de Producción. 2009. *Modelo de Desarrollo orientado a componentes del proyecto ERP - CUBA.* La Habana, Cuba : s.n., 2009.

EVA. 2006. Entorno Virtual de Aprendizaje. [En línea] Universidad de las Ciencias Informáticas, 2006. [Citado el: 13 de 6 de 2011.] http\\eva.uci.cu.

Guglielmetti, Marcos. 2005. Definición de la Arquitectura de Software. [En línea] 10 de febrero de 2005. [Citado el: 20 de marzo de 2011.] http://www.mastermagazine.info/termino/3916.php.

Hillar, Gastón C. 2000. Memorias Auxiliares. [En línea] 2000. [Citado el: 5 de mayo de 2011.] http://www.monografias.com/trabajos16/memorias-auxiliares/memorias-auxiliares.shtml.

infoviews.S:A. NAS. [En línea] [Citado el: 23 de 11 de 2010.] http://www.nas-das.com/NAS/..

—. SAN. [En línea] [Citado el: 23 de 11 de 2010.] http://www.nas-das.com/SAN/..

Jacobson, Booch, Rumbaugh. 2000. *El proceso unificado del desarrollo de software .* s.l. : Addison Wesley, 2000.

James Rumbaugh, Ivar Jacobson, Grady Booch. 2000. *El Proceso Unificado de Desarrollo de Software.* Madrid : s.n., 2000.

KRUCHTEN. 2004. *Metodologías de desarrollo de software.* 2004.

Larman, Craig. 1999. *UML y patrones.* México : s.n., 1999.

Marañón, Gonzalo Álvarez. 1999. Java. [En línea] 1999. [Citado el: 12 de Enero de 2011.] http://www.iec.csic.es/criptonomicon/java/quesjava.html.

Martinez, David Luis. EVA. [En línea] Universidad de las Ciencias Informaticas. [Citado el: 23 de noviembre de 2010.] http://eva.uci.cu/file.php/458/Bibliografia_complementaria/Sistemas_Operativos._David_Luis.pdf..

Masadelante. 1999. Definición de CPU. *Definición de CPU.* [En línea] 1999. [Citado el: 10 de Mayo de 2011.] http://www.masadelante.com/faqs/cpu.

—. 1999. Definición de RAM. [En línea] 1999. [Citado el: 10 de Mayo de 2011.] http://www.masadelante.com/faqs/memoria-ram.

Meyer, Lisandro Damián Nicanor Pérez. 2007. Introducción Qt. *Introducción Qt.* [En línea] Septiembre de 2007. [Citado el: 13 de Enero de 2011.] http://www.cafeconf.org/2007/slides/lisandro_perez_meyer_introduccion_%20a_qt4.pdf.

Miguel, Alcalde Eduardo García. 2000. Dispositivos de entrada y salida. [En línea] 2000. [Citado el: 5 de Mayo de 2011.] http://dispositivosdeentradaysalida.blogspot.com/.

Mora, Francisco. 2002. *UML:Lenguaje Unificado de Modelado.* Alicante : s.n., 2002.

Olsina. 1999. Introducción a la Disciplina de Requisitos de RUP. *Introducción a la Disciplina de Requisitos de RUP.* [En línea] 1999. [Citado el: 15 de Junio de 2011.] http://eva.uci.cu/file.php/102/Curso_2010-2011/Clases/Semana_06/Conf_7/Materiales_complementarios/Introduccion_a_la_Disciplina_de_Requisitos.pdf.

Ortega, Jose Vicente Garrido. 2007. La Unidades de Almacenamiento. . [En línea] 15 de 6 de 2007. [Citado el: 23 de noviembre de 2010.] http://tecnocienciahoy.blogspot.com/ ..

Pardo, Lisandra. 2009. Talent Copy V. [En línea] 23 de 12 de 2009. [Citado el: 5 de 11 de 2010.] http://www.neoteo.com/talent-copy-v-copia-acelerada-de-archivos.neo..

Pedroza, Tatiana Sugey. 2002. Sistema de Archivos y Memoria Cache en Disco Duro . . 2002.

Rational Software Corporation. 2003. Rational Rose. [En línea] 2003. [Citado el: 10 de Enero de 2011.] http://www.slideshare.net/vivi_jocadi/rational-rose.

Salagar, Juan Segura. 2005. Características de lenguaje de programación (Java). [En línea] 2005. [Citado el: 12 de Enero de 2011.] http://tikal.cifn.unam.mx/~jsegura/LCGII/java3.htm.

SIERRA. 1990. *Administración de la Entrada/Salida y planificación de discos.* Boston : s.n., 1990.

Stroustrup, Bjarne. 1998. Lenguaje de programación (C++). [En línea] 1998. [Citado el: 11 de Enero de 2011.] http://www.pergaminovirtual.com/definicion/C.html?PHPSESSID=9a75f4228bde94e4cf0cafade534bc89.

Tella, José Manuel. 2003. Tecnología de acceso a disco. [En línea] 19 de Enero de 2003. [Citado el: 5 de Noviembre de 2010.] http://multingles.net/docs/jmt/ata.htm..

Tobón, Luis Miguel Echeverry. 2007. Caso_práctico_de_la_metodología_XP.pdf. [En línea] 2007. [Citado el: 14 de Enero de 2011.] http://eva.uci.cu/file.php/102/Curso_2010-2011/Clases/Semana_02/Seminario_1/Materiales_complementarios/Caso_práctico_de_la_metodología_XP.pdf.

Visconti, Marcello. 2004. *Fundamentos de Ingeniería de Softaware.* 2004.

Bibliografía.

1. Buffer. [En línea] [Citado el: 23 de 11 de 2010.] http://sopa.dis.ulpgc.es/ii-dso/leclinux/fs/cache/LEC16_CACHE.pdf.

2. La velocidad de copiado de archivos en Windows 7. [En línea] 28/12/2009. [Citado el: 23 de 11 de 2010.] http://windows7noticias.com/como-aumentar-la-velocidad-de-copiado-de-archivos-en-windows-7/.

3. NAS. [En línea] [Citado el: 23 de 11 de 2010.] http://www.dgtallika.com/2010/06/definicin-de-hoy-nas-network-attached-storage/

4. Talent Copy V. [En línea] [Citado el: 5 de 11 de 2010.] http://www.genbeta.com/windows/talent-copy-v-hace-que-la-copia-de-archivos-sea-mas-rapida-en-windows-7vista

5. Talent Copy V. [En línea] [Citado el: 5 de 11 de 2010.] http://www.destrampado.com/tag/download-talent-copy-v/

6. Larman, Craig. *UML y Patrones. Introducción al análisis y diseño orientado a objetos.* México: Prentice Hall, 1999. ISBN: 970-17-0261-1.

7. Capas y niveles. [En línea] [Citado el: 22 de 3 de 2011.] http://es.wikipedia.org/Programación _por capas.