

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS
FACULTAD 3



*Análisis de los módulos Planificación de Disco y Administración
de Memoria de un Laboratorio Virtual de apoyo a la asignatura de
Sistemas Operativos.*

Trabajo de diploma para optar por el título de Ingeniería en Informática.

Autores: Lianyi Ramos León.

Yanelis Pulido Piña.

Tutor: Ing. Dusniel Horta Centeno.

Ciudad de la Habana, Cuba
Junio, 2008

Lo que caracteriza a una inteligencia formada es que puede descansar satisfecha con el grado de precisión que la naturaleza de su asunto permite, y no buscar la exactitud cuando sólo una aproximación de la verdad es posible...

Aristóteles.

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Firma del autor
(Yanelis Pulido Piña)

Firma del autor
(Lianyi Ramos León)

Firma del tutor
(Dusniel Horta Centeno)

OPINIÓN DEL USUARIO DEL TRABAJO DE DIPLOMA

El Trabajo de Diploma, titulado **Análisis de los módulos Planificación de Disco y Administración de Memoria de un Laboratorio Virtual de apoyo a la asignatura de Sistemas Operativos**, fue realizado en La Universidad de las Ciencias Informáticas (UCI). Esta entidad considera que, en correspondencia con los objetivos trazados, el trabajo realizado le satisface

Totalmente

Parcialmente en un ____ %

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

Como resultado de la implantación de este trabajo se reportará un efecto económico considerable. Para que así conste, se firma la presente a los ____ días del mes de _____ del año _____.

Este trabajo no es el producto de la labor individual, sino más bien el resultado del esfuerzo de muchas personas durante los años de formación; para ellos nuestro recuerdo y agradecimiento, en especial:

A la revolución, por permitirme ser parte de ella y darme todo lo que tengo.

A Fidel, por crear esta maravillosa escuela. A Raúl, por continuar forjando una mejor Revolución.

A mi mamita querida, tu eres mi guía y todo lo que se lo aprendí de ti, por estar siempre dispuesta, por ser mi confidente y darme tanto amor, gracias por traerme a la vida.

A mi papito lindo, por preocuparte por mí y apoyarme en todas mis decisiones, gracias por ser mi padre.

A Maye, mi hermanita por llenar mi vida de alegría y hacerla cada día mejor.

A mis abuelos, a los que tengo a mi lado, en especial a Abu por confiar en mi y darme mucho cariño y todo su apoyo, y a los que ya se fueron siempre los recordaré y tendré en mi corazón, estoy segura que estarían orgullosos de mi.

A mis tías y tíos del alma, por darme un buen ejemplo, los quiero mucho.

A mis primas y primos por quererme, soportarme y estar siempre a mi lado.

A Yane, mi amiga y compañera de tesis, por soportarme todos estos años de la carrera, por confiar en mi y por estar a mi lado en los momentos buenos y malos, gracias.

A Handy, por aguantar una y otra vez las mismas historias de mi vida, por ayudarme tanto desde que lo conocí, por ser incondicional conmigo.

A Lili y Yeni, por apoyarme y darme ánimo desde lejos.

A Yuli, por estar siempre dispuesta y poder contar con ella siempre que la necesite.

A mis vecinos, que tuvieron mucha fe en mí y me apoyaron desde que comencé mi carrera.

A mis amigas y amigos por no haberme olvidado en estos años.

A Osain por causarle tantas molestias con mi tesis.

A Iosmel, por ayudarme a salir de la prueba de programación.

A Dusniel, mi tutor, por estar siempre dispuesto, por su comprensión y ayuda.

Y a todos aquellos que de una forma u otra hicieron posible que este gran sueño se realizara e hicieron de mi una mejor persona.

Muchas Gracias

Primero que todo agradecerle a la vida por hoy permitirme estar aquí y dedicarle este trabajo a todas las personas que de una u otra forma, me han ayudado, quiero que sepan que estaré eternamente agradecida, muy en especial:

A mi mamá (mi gordita) por ser la mejor madre del mundo, por todo el amor que me has dado, por confiar incondicionalmente en mi, por sacrificarte tanto por verme graduada, siempre serás la razón de mi vivir.

A mi papá (mi papito) por ser el mejor papá del mundo, por ese amor ciego que me tienes, por tu confianza, por todo esa abnegación por verme graduada, siempre serás el amor de mi vida.

A mis hermanas queridas Yuli y Chae, por enseñarme y guiarme en todos los momentos de mi vida. Las adoro.

A mi sobrino Sergito, para que el día de mañana se sienta orgulloso de su tía.

A toda mi familia en especial a mis abuelos Olga, Marce, Marta y Ovavo por quererme tanto. Mi cariño es para Uds.

A mi otra familia, mi novio, mi suegra, mi suegro, mi tío. Por querer lo mejor para mí, hoy les dedico este trabajo. Saben que los quiero.

A Lia, por ser mí amiga, mí compañera de tesis, mí cómplice estos 5 años de carrera, por estar siempre en las buenas y malas.

A Dayi mi amiga de toda la vida, por ser incondicional conmigo, por tus consejos y tú presencia allí siempre.

A todos mis amigos que son muchos y desafortunadamente no tengo espacio para mencionarlos a todos de forma individual, por quererme, soportarme y ayudarme en todo momento.

A mi tutor, por ese enorme empeño que todo va a salir bien, por su responsabilidad y paciencia conmigo.

A la Revolución y al Comandante por darme la oportunidad de forjarme en la UCI.

A todos los que de una manera u otra han contribuido a mi formación profesional.

A todos, mi amor, mi cariño, mi respeto siempre los voy a llevar en mi corazón.

Muchísimas Gracias.

A Mis Padres hermosos por haberme regalado esta vida llena de amor y cariño, por su comprensión y apoyo.

A mi Hermana por decirme todos los días, "Te quiero" y regalarme una sonrisa.

A mi Abuela querida por enseñarme tanto de la vida y tener fe en mí.

Ustedes que son el centro de mi vida y la razón de mí existir.

Al resto de mi familia, a mis vecinos y amigos

A todos les dedico este sueño hecho realidad muchísimas gracias

Lianyi Ramos León

*A mis dos corazones, Mis Padres, por ser mis amores, mis confidentes, mi mayor tesoro,
por permitirme ser como soy, por ser los más importantes en mi vida.*

A mis Hermanas por acompañarme siempre y quererlas como las quiero.

A mis Abuelitos preciosos que nunca voy a olvidar y siempre los tendré en mi corazón.

A mi otra Familia por ser especial conmigo y siempre estar allí para mí.

A mis amigas de siempre, por querernos como hermanitas.

Para todos es este sueño realizado.

Yanelis Pulido Piña

RESUMEN

Sistemas Operativos es una asignatura básica del plan de estudio UCI en tercer año. Es una asignatura compleja, debido al alto nivel de abstracción que se necesita para impartir el contenido de algunos temas. El claustro de profesores de la facultad 3 tiene la necesidad de crear una herramienta que apoye de forma gráfica tanto las explicaciones del profesor en los temas de *Administración de Memoria* y *Planificación de Disco* como el auto aprendizaje del estudiante en la resolución de ejercicios. El objetivo fundamental del trabajo es realizar el análisis de un laboratorio virtual que favorezca el diseño e implementación de un simulador de los temas correspondientes a *Planificación de Disco* y *Administración de Memoria*. Se realiza un estudio del estado del arte sobre la simulación de Laboratorios Virtuales en el mundo, determinando las ventajas y desventajas que poseen como medios didácticos del Proceso de Enseñanza-Aprendizaje. El estudio de las principales características de las metodologías usadas en el mundo para el Proceso de Desarrollo de Software permitió seleccionar a RUP como la más adecuada para la realización de este trabajo. Además de realizarse una breve reseña de las tareas de un Analista de Sistemas. Como resultado se obtuvo los artefactos correspondientes al modelo del negocio donde se definen, entre otros, las reglas y el diagrama de casos de uso con sus descripciones. Finalmente se desarrolla el modelo del sistema para la futura solución informática determinando los artefactos principales correspondientes incluyendo la propuesta de prototipo no funcional de las interfaces de usuario del sistema y los diagramas de clases del análisis.

Palabras Clave: Simulador, Planificación de Disco, Administración de Memoria, Laboratorio Virtual, Sistemas Operativos.

ÍNDICE

INTRODUCCIÓN	- 1 -
1 FUNDAMENTACIÓN TEÓRICA	- 6 -
1.1 Introducción	- 6 -
1.2 Laboratorios virtuales.....	- 6 -
1.2.1 Ventajas del Laboratorio Virtual.....	- 7 -
1.2.2 Desventajas de los Laboratorios Virtuales.....	- 7 -
1.2.3 Tipos de Laboratorios Virtuales.....	- 8 -
1.2.4 Razones para la creación de un Laboratorio Virtual.....	- 9 -
1.2.5 Funciones en el Proceso de Enseñanza-Aprendizaje.....	- 10 -
1.2.6 Software de simulación en el proceso de aprendizaje.....	- 10 -
1.3 Proceso de Desarrollo de Software.....	- 11 -
1.3.1 Metodologías para el Desarrollo de Software.....	- 12 -
1.3.2 Justificación de la Metodología seleccionada.....	- 20 -
1.3.3 Herramientas.....	- 20 -
1.4 Análisis del Sistema.....	- 23 -
1.4.1 Rol de Analista.....	- 23 -
1.5 Ingeniería de Requerimientos.....	- 24 -
1.6 Conclusiones del capítulo.....	- 27 -
2 MODELACIÓN DEL NEGOCIO	- 28 -
2.1 Introducción.....	- 28 -
2.2 Descripción de los Procesos de Automatización.....	- 28 -
2.3 Modelo de Negocio.....	- 31 -
2.3.1 Reglas del Negocio.....	- 32 -
2.3.2 Actores del Negocio.....	- 32 -
2.3.3 Trabajadores del Negocio.....	- 33 -
2.3.4 Diagrama de Casos de Uso del Negocio.....	- 33 -
2.3.5 Descripción de los Casos de Uso del Negocio.....	- 34 -
2.3.6 Modelo de Objetos del Negocio.....	- 53 -
2.4 Conclusiones del capítulo.....	- 54 -
3 MODELACIÓN DEL SISTEMA	- 55 -
3.1 Introducción.....	- 55 -

3.2	Especificación de los requerimientos de Software.....	- 55 -
3.2.1	Requerimientos Funcionales.....	- 55 -
3.2.2	Requerimientos no Funcionales.....	- 57 -
3.3	Actores del Sistema.....	- 58 -
3.4	Diagrama de Casos de Uso del Sistema.....	- 58 -
3.5	Descripción de los Casos de Uso del Sistema.....	- 59 -
3.6	Diagramas de Clases del Análisis.....	- 81 -
3.7	Análisis de los resultados.....	- 84 -
3.8	Conclusiones del capítulo.....	- 90 -
4	CONCLUSIONES.....	- 91 -
5	RECOMENDACIONES.....	- 92 -
6	BIBLIOGRAFÍA.....	- 93 -
7	GLOSARIO DE TÉRMINOS.....	- 96 -
8	ANEXOS.....	- 102 -

INTRODUCCIÓN

La informática, como herramienta para resolver problemas en la enseñanza constituye un nuevo medio de desarrollo de habilidades teóricas y prácticas y opera como factor que modifica en mayor o menor grado el contenido de cualquier currículum educativo. En este ámbito el uso de entornos virtuales para la realización del proceso de enseñanza y aprendizaje es creciente, en lo que se destaca el uso de software de simulación en diferentes niveles, cuestión que es trascendental en el proceso docente-educativo que se despliega en las universidades en las distintas materias.

A estas tendencias no ha escapado la universidad cubana, siendo destacada en el uso de las Tecnologías de la Informática y las Comunicaciones la Universidad de las Ciencias Informáticas (UCI), lo cual es coherente con el llamado que se realiza desde el V Congreso del Partido Comunista de Cuba cuando acerca del rol de la informática en la sociedad cubana se plantea que:

“El país debe encaminarse resueltamente a la modernización informática mediante un programa integral que involucre a las organizaciones que deben proveer los recursos materiales, financieros e intelectuales y a las entidades económicas, políticas y sociales que deben traducirlos en más y mejores productos y servicios. La industria de los servicios informáticos deberá asegurar la modernidad de su base técnica y organizativa, y la elevación constante del nivel científico-técnico de sus especialistas con vistas a garantizar esos propósitos.” (Resolución Económica del V Congreso del PCC, 1997)

En este marco es destacado el rol que desempeñan los Laboratorios Virtuales (LV), los cuales han tenido particular avance en el desarrollo de la enseñanza y el aprendizaje, constituyendo una poderosa herramienta de apoyo al trabajo tanto del profesor como del estudiante.

Sistemas Operativos (SO) como asignatura técnica fundamental del plan de estudio del 3er año de la carrera de Ingeniería en Ciencias Informáticas, se divide, para su estudio, en cuatro temas fundamentales que abarcan los principales aspectos concernientes a la administración de recursos como función fundamental de un Sistemas Operativos real. La enseñanza de estos es una tarea que se torna profundamente compleja debido al alto nivel de abstracción que requiere por parte de los estudiantes para lograr un buen entendimiento acerca del funcionamiento interno de cada

administrador involucrado, sobre todo, los relacionados con *Planificación de Disco y Administración de Memoria*.

Esto se agudiza aun más si se tiene en cuenta que el claustro de profesores de la asignatura está sustentado en la facultad 3 en un 72% por profesores que, aun siendo egresados de la misma rama, no llegan a acumular más de un año de experiencia laboral y pedagógica, lo cual influye, sin dudas, en la calidad del proceso Docente/Educativo.

Las observaciones realizadas durante la impartición de las clases de SO han arrojado resultados de comprensión muy positivos cuando el profesor, más que limitarse a la mera explicación, hace uso de medios didácticos basados en las TICs, que aunque no tenían toda la calidad necesaria, sí permitieron simular el funcionamiento de los diferentes algoritmos de *Planificación de Disco y Administración de Memoria* de una manera más práctica.

A raíz de los elementos anteriormente expuestos surge la idea, por parte del colectivo de la asignatura, de la implementación y puesta en práctica de una herramienta de calidad que funcione como laboratorio virtual simulador de algoritmos de *Planificación de Disco y Administración de Memoria*, para apoyar las clases del profesor y favorecer el aprendizaje de los estudiantes acerca del funcionamiento interno real de un Sistemas Operativos Moderno.

Sin embargo, se han identificado una serie de situaciones que hacen difícil el comienzo de la implementación de la herramienta y las cuales forman en su conjunto la verdadera **Situación Problémica** que se presenta en el desarrollo de este trabajo:

Primeramente, debe partirse del hecho de que los desarrolladores de la herramienta no son los clientes ni serán los usuarios finales de la misma, por lo que no conocen en profundidad la situación o contexto que se desea tratar.

Desde el punto de vista de la Ingeniería de Software, la falta de una modelación previa del negocio donde se identifiquen todas las necesidades funcionales o no, que debe poseer la herramienta una vez diseñada, implementada y puesta en uso, para favorecer y alcanzar el efecto deseado en el aprendizaje de los estudiantes acerca de la *Planificación de Disco* y la *Administración de Memoria* en las clases de SO, trae como consecuencia la carencia de un diseño eficiente y flexible, de los artefactos indispensables, que les permita a los diseñadores comenzar el diseño del software bajo un entendimiento entre lo deseado por el cliente y lo que se entregará, pero sobre todo con la seguridad de que la *gestión de cambios* posterior será mínima.

La consecuencia central de todo lo antes planteado a criterio de las autoras de este trabajo de diploma se presenta en el siguiente **Problema Científico**: ¿Cómo proporcionar una entrada adecuada al diseño y la implementación de un laboratorio virtual, que simule los algoritmos de *Planificación de Disco y Administración de Memoria* en la asignatura de Sistemas Operativos?. Este problema se enmarca en el **Objeto de Estudio**: La Ingeniería de Software en el desarrollo de laboratorios virtuales cuyo **Campo de Acción** es el Análisis de simuladores de Planificación de Disco y Administración de Memoria en Sistemas Operativos.

En este sentido, el trabajo que aquí se presenta tiene como **Objetivo General**, realizar el Análisis de un laboratorio virtual que favorezca el diseño y la implementación de un simulador, de *Planificación de Disco y Administración de Memoria*, para apoyar el aprendizaje de los estudiantes en las clases de Sistemas Operativos.

La **Hipótesis** planteada en este trabajo es que si se realiza el Modelo de Análisis adecuado de un Laboratorio Virtual que simule la *Planificación de Disco y Administración de Memoria* en las clases de Sistemas Operativos, entonces se favorecerá su diseño e implementación posterior de una manera rápida, eficiente y con una gestión de cambios mínima.

La **Población** que forma parte de la investigación realizada y que da origen al desarrollo de este trabajo enmarca a la matrícula de estudiantes de 3er año en la carrera de Ingeniería en Ciencias informáticas que cursa la asignatura de Sistemas Operativos en la Facultad 3.

Para el cumplimiento exitoso del objetivo general se han definido una serie de **Tareas Investigativas** que logran el desarrollo eficiente de esta investigación.

- Realización de un estudio de las principales experiencias alcanzadas y tendencias existentes en el mundo, referente a los Laboratorios Virtuales de simulación como medios de apoyo a la docencia.
- Realización de un estudio del estado del arte acerca de los principales avances en las tecnologías, las metodologías de Desarrollo de software y sus tendencias actuales.
- Reunión con los profesores de la asignatura de Sistemas Operativos de la Facultad 3 y el Departamento Central para identificar las principales necesidades del sistema en cuestión en esta primera versión.

- Utilización de herramientas CASE para modelar y generar los artefactos correspondientes a la etapa de Análisis de los módulos relacionados con la *Planificación de Disco y Administración de Memoria*.

Para puntualizar el cumplimiento de las tareas planteadas se emplearon los siguientes métodos:

Métodos de trabajo científico.

➤ Métodos Teóricos

- Histórico: Porque es el que me permite estudiar todo el proceso y obtener un conocimiento histórico de su desarrollo y comportamiento.
- Analítico-sintético: En el trabajo se utilizó este método porque permite como método teórico, buscar la esencia del tema, en el análisis de la bibliografía realizando una síntesis de la misma, posibilitando así la extracción de los elementos más importantes.

➤ Métodos lógicos

- Hipotético _ deductivo: porque es el que Permite inferir conclusiones a partir del conocimiento precedente (Hipótesis) y además permite comprobarla.

➤ Métodos particulares

- La entrevista individual no estructurada: prevé el tema pero no lleva un cuestionario rígido. Se aplica a especialistas en el tema, es una forma de obtener criterios de expertos.

La estructura en capítulos de este trabajo queda distribuida de la siguiente manera:

Capítulo 1. FUNDAMENTACIÓN TEÓRICA.

En este capítulo se recogen los principales conceptos sobre los Laboratorios Virtuales, como sus ventajas, desventajas, las razones para su creación y los tipos que existen. Se realiza un estudio general sobre las metodologías, lenguajes y herramientas de modelado de software y se refleja cuál de ellas fue la seleccionada.

Capítulo 2. MODELACIÓN DEL NEGOCIO.

En este capítulo se describe y se modela el negocio de los módulos de *Planificación de Disco* y *Administración de Memoria* para la asignatura de Sistemas Operativos. Se identifican y justifican los actores y trabajadores del negocio, se realiza el diagrama de Casos de Uso del Negocio, los diagramas de actividad correspondiente a cada Caso de Uso, el diagrama de objeto y una descripción de los procesos del negocio generados en este Flujo de trabajo.

Capítulo 3. MODELACIÓN DEL SISTEMA.

Se especifican los requerimientos funcionales y no funcionales del sistema. Se identifican y se justifican los actores del mismo, se muestra el diagrama de Casos de Uso, así como sus descripciones, además se realizan los diagramas de clases de análisis por cada uno de los Casos de Uso del sistema. Finalmente se realiza el análisis de los resultados obtenidos usando para ellos algunas técnicas existentes con este fin.



1 FUNDAMENTACIÓN TEÓRICA.

1.1 Introducción

En correspondencia con el objetivo trazado para la confección de este trabajo, se realiza el siguiente estudio preliminar. En el presente capítulo se abordan los principales conceptos acerca de los Laboratorios Virtuales (LV), observando sus ventajas, desventajas, así como los distintos tipos que existen. Se desarrolla también un análisis sobre las funciones en el proceso de enseñanza-aprendizaje y las razones para su creación. Se reflejan además cuáles son los métodos y tecnologías utilizadas para el desarrollo de este trabajo y los aspectos que influyen en la selección de las mismas.

1.2 Laboratorios virtuales.

Muchos son los autores que han dado una definición de lo que piensan sea un Laboratorio Virtual, a continuación se citan algunos ejemplos:

“Un espacio electrónico de trabajo concebido para la colaboración y la experimentación a distancia con objeto de investigar o realizar otras actividades creativas, y elaborar y difundir resultados mediante tecnologías difundidas de información y comunicación“. (VARY, 2000)

“Es un ambiente heterogéneo, distribuido para solucionar problemas que permite a un grupo de investigadores situados alrededor del mundo trabajar juntos en un conjunto de proyectos. Como en cualquier laboratorio, las herramientas y las técnicas son específicas al dominio de la investigación, pero los requisitos básicos de infraestructura se comparten a través de las disciplinas. Aunque está relacionado con algunas de las aplicaciones de la tele-inmersión, el laboratorio virtual no asume a priori la necesidad de un ambiente inmersivo compartido. (SALAVERRÍA & FERREIRA, 2005)

1.2.1 Ventajas del Laboratorio Virtual.

- Se convierten en una ayuda interactiva para el aprendizaje de contenidos difíciles de demostrar en la realidad.
- El aprendizaje está basado en simulaciones.
- El estudiante aprende por exploración.
- Permiten que el estudiante realice cada laboratorio en cualquier lugar y tiempo, sin necesidad de acudir a las aulas.
- Se incorporan las Tecnologías de la Información y Comunicación en las prácticas educativas y sociales para beneficio de los estudiantes.
- Facilitan la creación de aprendizajes activos.
- Propician ambientes para la resolución de problemas.
- Propician el desarrollo de destrezas de pensamiento y procesos de reflexión.
- Permiten que el estudiante llegue al conocimiento empezando por donde lo desee.
- Promueve un aprendizaje basado en experiencias y repeticiones.
- Permiten que el estudiante busque información por su cuenta.
- Facilitan el aprendizaje interactivo.
- El estudiante se convierte en un sujeto activo que va potenciando su curiosidad intelectual.
- El estudiante ajusta el material a su ritmo e interés“. (ESTRADA & NÁJERA, 2005)
- Nos permite de inicio la interacción entre el medio, el estudiante y el contenido.
- La posibilidad de desarrollar habilidades cognitivas e investigativas a partir del aprendizaje por descubrimiento.
- La capacidad de interacción, simulación y retroalimentación.
- La posibilidad de desarrollar el aprendizaje cooperativo y solidario entre los estudiantes. (GUARDADO & SÁNCHEZ, 2000)

1.2.2 Desventajas de los Laboratorios Virtuales.

Acompañado de las ventajas también se destacan los inconvenientes de este proceso. A continuación mostramos los más destacados:

- Se corre el riesgo de que el estudiante se comporte como un mero espectador. Es importante que las actividades en el LV, vengan acompañadas de un guión que explique el concepto a

estudiar, así como las ecuaciones del modelo utilizado. Es necesario que el estudiante realice una actividad ordenada y progresiva, conducente a alcanzar objetivos básicos concretos.

- El estudiante no utiliza elementos reales en el LV, lo que provoca una pérdida parcial de la visión de la realidad. Además, no siempre se dispone de la simulación adecuada para el tema que el profesor desea trabajar. En Internet existe demasiada información, a veces inútil. Para que sea útil en el proceso de enseñanza-aprendizaje, se ha de seleccionar los contenidos relevantes para los estudiantes. (ROSADO & HERREROS, 2005).
- El uso de los medios tecnológicos en el proceso de enseñanza no garantiza que los estudiantes desarrollen habilidades cognitivas. Es decir, la calidad educativa no depende de los medios tecnológicos sino del uso pedagógico que el docente realice y del contexto en el que se desarrolle“. (GUARDADO & SÁNCHEZ, 2000).
- El estudiante de zonas aisladas geográficamente y que no cuenta con un computador, no puede obtener el conocimiento por este medio.
- Los estudiantes con discapacidad física no pueden usar esta técnica.
- El estudiante podría limitarse únicamente a los contenidos del laboratorio“. (ESTRADA & NÁJERA, 2005).

1.2.3 Tipos de Laboratorios Virtuales.

Otra característica de los LV es su diversidad, en correspondencia con esto existen distintos tipos. Según bibliografía consultada, existen tres clasificaciones, que se presentan a continuación:

- *Laboratorios virtuales software*: son laboratorios desarrollados como un programa de software independiente destinado a ejecutarse en la máquina del usuario, y cuyo servicio no requiere de un servidor Web. Es el caso de programas con instalación propia, que pueden estar destinados a plataformas Unix, Linux, Windows e incluso necesitar que otros componentes de software estén instalados previamente, pero que no necesitan los recursos de un servidor determinado (como bases de datos o módulos de software de servidor) para funcionar. También determinados laboratorios virtuales pensados inicialmente como aplicaciones Java accesibles a través de un servidor Web se pueden considerar de este tipo si funcionan localmente y no necesitan recursos de un servidor en concreto.

- *Laboratorios virtuales Web*: en contraste con el anterior, este tipo de laboratorio se basa en un software que depende de los recursos de un servidor determinado. Esos recursos pueden ser determinadas bases de datos, software que requiere ejecutarse en su servidor o la exigencia de determinado hardware para realizarse. Estos no son programas que un usuario pueda descargar en su computadora para ejecutar localmente de forma independiente.
- *Laboratorios remotos*: se trata de laboratorios que permiten operar remotamente cierto equipamiento, bien sea didáctico como maquetas específicas, o industrial, además de poder ofrecer capacidades de laboratorio virtual. En general, estos laboratorios requieren de equipos servidores específicos que les den acceso a las máquinas a operar de forma remota, y no pueden ofrecer su funcionalidad ejecutándose de forma local. Otro motivo que hace dependientes estos laboratorios de sus servidores es la habitual gestión de usuarios en el servidor. (HERÍAS, 2003)

1.2.4 Razones para la creación de un Laboratorio Virtual.

Las razones están asociadas a las ventajas antes planteadas, para abordar más sobre este aspecto a continuación se citan varios factores que inducen a la creación de un laboratorio virtual.

- Algunos problemas científicos y tecnológicos importantes requieren una inversión cuyo tamaño y escala supera la capacidad de un laboratorio único, incluso de una nación aislada.
- Los recursos humanos y los conocimientos especializados necesarios para alcanzar objetivos científicos y tecnológicos pueden distribuirse entre dos o más instituciones.
- El tema tratado puede exigir la participación de especialistas de distintas regiones debido a la necesidad de datos regionales específicos (nuevos o de archivo), o de pruebas sobre el terreno, así como a los recursos humanos disponibles o a las posibilidades de formación existente.
- La aplicación de los resultados de la investigación para obtener beneficios sociales y económicos puede depender de la participación regional en el proyecto. (VARY, 2000)
- Laboratorios adecuados acordes a la práctica.
- Reducir riesgos al recibir la capacitación (para el equipo y para el estudiante).
- Reducción de costos utilizando laboratorios basados en simulación y tele-operación.
- Incorporación de otros elementos como los tutores inteligentes.

- Oportunidad de mejorar la formación de los estudiantes. (ESTRADA & NÁJERA, 2005).

1.2.5 Funciones en el Proceso de Enseñanza-Aprendizaje.

A continuación se muestran algunas de las funciones que deben cumplir los laboratorios virtuales en el proceso de aprendizaje.

Desde el punto de vista académico:

- Facilitar experiencias concretas y oportunidades para afrontar los errores conceptuales de los estudiantes.
- Lograr hábitos de lectura, de análisis y de síntesis.
- Lograr una adecuada expresión escrita (coherencia en la redacción, ortografía) en la presentación de los resultados.
- Interactuar con diversas fuentes de información incluyendo las Tecnologías de la Información y las Comunicaciones para la actualización del contenido en cuestión, exigiendo la visita a centros de Información Científico Técnico y la interrelación comunicativa entre las fuentes.

Desde el punto de vista investigativo:

- Simular y apreciar el papel del científico en la investigación.
- Desarrollar habilidades de razonamiento lógico e interpretativo.
- Identificar y formular el problema dada una situación problemática.
- Introducir y aplicar métodos de la investigación científica.
- Emplear las Tecnologías de la Información y las Comunicaciones. (MADERA & VIZOSO, 1997)

1.2.6 Software de simulación en el proceso de aprendizaje.

El empleo de la simulación a través de la computadora es de indiscutible utilidad para los estudiantes en el aprendizaje de la asignatura de Sistemas Operativos. Durante los últimos años, ha sido una prioridad para nuestro Ministerio de Educación la introducción de la computadora en el proceso docente-educativo, para lo cual se han invertido cuantiosos recursos y se ha intentado estimular al profesorado hacia la integración de la nueva tecnología a su arsenal didáctico.

Aunque existe una gran diversidad de variantes de aplicación de la computadora como medio de enseñanza, una de ellas, la simulación, ha producido una considerable atracción como herramienta de utilidad para la enseñanza. La simulación resume toda la teoría relacionada con un proceso en el cual se sustituyen las situaciones reales por otras creadas artificialmente, de las cuales el estudiante debe aprender ciertas acciones, habilidades y hábitos que posteriormente debe aplicar en la vida real con igual eficacia.

1.3 Proceso de Desarrollo de Software.

El proceso de desarrollo de software es una descripción de la construcción del software, que contiene actividades organizadas de modo que en conjunto producen código probado. No existe una definición estándar de estas actividades y muchos autores le dan importancia a algunas más que a otras. En el Proceso Unificado de Desarrollo del Software se conocen como flujos de trabajo, y están compuestas por:

- Análisis del Negocio
- Ingeniería de Requerimientos.
- Análisis del Sistema.
- Diseño del Sistema.
- Implementación del Software.
- Prueba o Validación del Software.
- Planificación y Evolución.

En este trabajo se desarrollarán los tres primeros flujos.

En el Análisis del Negocio, el analista necesita un conjunto de técnicas y artefactos que le ayuden a obtener una visión suficientemente buena del entorno, necesita conocer la naturaleza del mismo, entender lo que desean los clientes, y delimitar el alcance del mismo teniendo en cuenta el tiempo disponible, el presupuesto y el personal asignado. Para la captura de los Requerimientos se aprecian diferentes puntos de partidas, en ocasiones se parte de un Modelo de Negocio o de un Modelo de Dominio que se generan en la primera etapa. Luego se procede a la enumeración de los Requisitos Candidatos, se comprende el contexto del Sistema y se capturan los Requisitos Funcionales y No Funcionales. El producto generado de esta etapa es un documento de Especificación de Requerimientos en donde se encuentran definidos todos los servicios requeridos del sistema y las restricciones sobre las que debe operar, una descripción detallada de los Casos de Uso que capturen los requisitos funcionales y no funcionales específicos por cada uno de ellos, y por último se obtiene un

conjunto de esbozos o prototipos interfaz de usuario, que representan el diseño de las interfaces del sistema. Estos resultados son de gran importancia ya que conforman el principal punto de partida para los siguientes flujos de trabajos. En el Análisis se ofrece una especificación más precisa de los Requisitos, refinándolos y estructurados de tal modo que facilite la comprensión y el mantenimiento de los mismos, con el objetivo de conseguir una comprensión más específica de los requisitos y una descripción de los mismos que sea fácil de mantener y que ayude a estructurar el sistema en clases de análisis y paquetes. Se utiliza el lenguaje de desarrollo para razonar aspectos más internos del sistema y facilitar que a partir de este momento el diseñador pueda tener mayor claridad en comprender como debería quedar el sistema en los flujos posteriores. (Jacobson, y otros, 2000).

1.3.1 Metodologías para el Desarrollo de Software.

Con el actual desarrollo acelerado de las tecnologías, los procesos de desarrollo de software se vuelven cada vez más complicados, a razón de esto se hizo necesario establecer una disciplina de desarrollo de software, enfocada a solucionar los diferentes problemas que trae consigo la elaboración de los mismos.

El término de metodología se define como un conjunto de métodos eficientes orientados a conseguir un objetivo propuesto. Son un conjunto de procesos que organizados dan una secuencia de pasos a seguir para obtener los hitos propuestos y finalmente el producto final. Como no existe una metodología de software universal, esta debe guiar a la organización en el desarrollo de sus objetivos. Por ello, la organización, su estructura, canales de información, recursos humanos y físicos, deben ser los adecuados y garantizar el correcto funcionamiento de los procesos y métodos definidos. Las metodologías imponen un proceso disciplinado sobre el desarrollo de software con el fin de hacerlo más predecible y eficiente. Lo hacen desarrollando un proceso detallado con un fuerte énfasis en planificar, inspirados por otras disciplinas de la ingeniería. En dependencia a lo que se quiera construir y de los componentes de que se disponga, se utilizará una metodología u otra. Sin embargo, hay algo que se debe tomar en cuenta: Las características de cada proyecto (equipo de desarrollo o recursos) exigen que la metodología sea configurable y flexible a sus necesidades. (Gómez & Gómez, 2006)

Existen actualmente dos enfoques o clasificaciones de las metodologías, en las que se encuentran:

- Metodologías tradicionales.
- Metodologías ágiles.

1.3.1.1 Metodologías Tradicionales.

Imponen una disciplina de trabajo sobre el proceso de desarrollo del software, con el objetivo de conseguir un software más eficiente y predecible. Para ello, se hace un especial hincapié en la planificación total de todo el trabajo a realizar y una vez que está todo detallado, comienza el ciclo de desarrollo del producto software. Este planteamiento está basado en el resto de disciplinas de ingeniería, a pesar de que el software no pueda considerarse como la construcción de una obra clásica de ingeniería.

Con estas metodologías se lleva trabajando desde hace tiempo y no ha habido en ningún caso ninguna experiencia traumática acerca de su uso. Pero aún así, han recibido diversas críticas, y la más común hace referencia a su carácter excesivamente burocrático. (CACERES & MARCOS)

A continuación se hace mención a una de las metodologías más importante dentro de esta clasificación. Mundialmente es una de las más utilizadas, al igual que en la universidad, debido a que es la metodología que se ajusta regularmente al tipo de desarrollo de los proyectos productivos. Pues de acuerdo a su particularidad, esta metodología es flexible y adaptable a las mismas.

1.3.1.1.1 Proceso Unificado de Desarrollo de Software (RUP).

La metodología RUP, llamada así por sus siglas en inglés Rational Unified Process, divide en 4 fases el desarrollo del software:

- **Concepción**, El Objetivo en esta etapa es determinar la visión del proyecto.
- **Elaboración**, El objetivo es determinar la arquitectura óptima.
- **Construcción**, El objetivo es llevar a obtener la capacidad operacional inicial.
- **Transmisión**, El objetivo es llegar a obtener el reléase del proyecto.

El ciclo de vida RUP es una implementación del Desarrollo en espiral. Fue creado ensamblando los elementos en secuencias semiordenadas. El ciclo de vida organiza las tareas en fases e iteraciones.

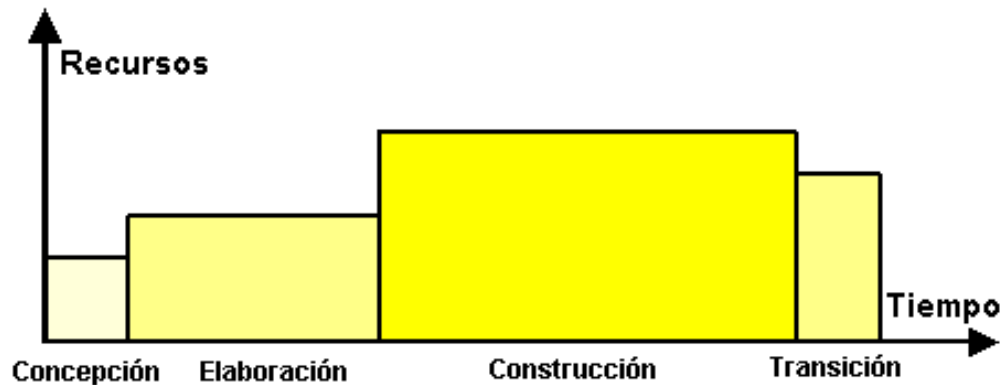


Figura 1. Ciclo de vida de la Metodología RUP.

Cada una de estas etapas es desarrollada mediante el ciclo de iteraciones, la cual consiste en reproducir el ciclo de vida en cascada a menor escala. Los Objetivos de una iteración se establecen en función de la evaluación de las iteraciones precedentes. (SANCHEZ, 2004)

Vale mencionar que el ciclo de vida que se desarrolla por cada iteración, es llevada bajo dos disciplinas:

Disciplina de Desarrollo:

- Ingeniería de Negocios: Entendiendo las necesidades del negocio.
- Requerimientos: Traslado de las necesidades del negocio a un sistema automatizado.
- Análisis y Diseño: Traslado de los requerimientos dentro de la arquitectura de software.
- Implementación: Creando software que se ajuste a la arquitectura y que tenga el comportamiento deseado.
- Pruebas: Asegurándose que el comportamiento requerido es el correcto y que todo lo solicitado está presente. (SANCHEZ, 2004)

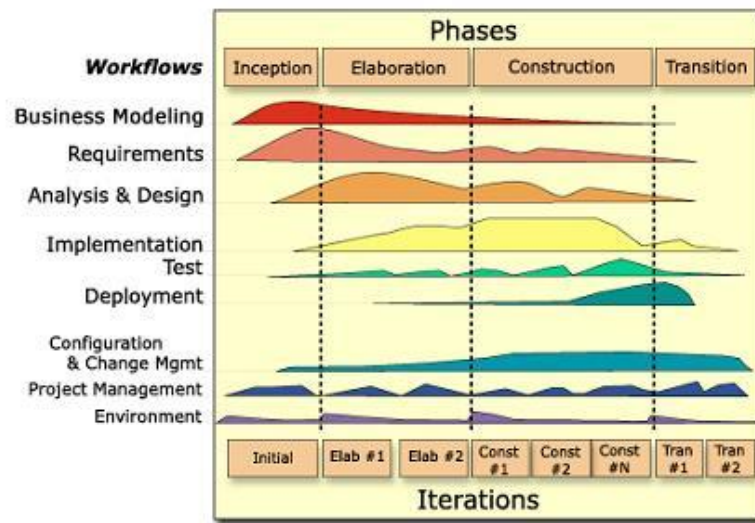


Figura 2. Fases e Iteraciones de la Metodología RUP.

Es recomendable que a cada una de estas iteraciones se les clasifique y ordene según su prioridad, para luego convertirse en un producto entregable al cliente. Esto trae como beneficio la retroalimentación que se tendría en cada entrega o en cada iteración.

Los elementos de RUP son:

- **Actividades**, Son los procesos que se llegan a determinar en cada iteración.
- **Trabajadores**, Vienen hacer las personas o entes involucrados en cada proceso.
- **Artefactos**, Un artefacto puede ser un documento, un modelo, o un elemento de modelo.
- **Flujo de Actividades**, Secuencia de actividades realizadas por trabajadores y que produce un resultado de valor observable.

Características principales de RUP:

- **Guiado por lo casos de uso**: Los casos de uso son el instrumento para validar la arquitectura del software y extraer los casos de prueba.
- **Centrado en la arquitectura**: Los modelos son proyecciones del análisis y el diseño constituye la arquitectura del producto a desarrollar.
- **Iterativo e incremental**: Durante todo el proceso de desarrollo se producen versiones incrementales (que se acercan al producto terminado) del producto en desarrollo.

Una particularidad de esta metodología es que, en cada ciclo de iteración, se hace exigente el uso de artefactos, siendo por este motivo, una de las metodologías más importantes para alcanzar un grado de certificación en el desarrollo del software. (SANCHEZ, 2004)

1.3.1.1.2 Microsoft Solution Framework (MSF).

Esta es una metodología flexible e interrelacionada con una serie de conceptos, modelos y prácticas de uso, que controlan la planificación, el desarrollo y la gestión de proyectos tecnológicos. MSF se centra en los modelos de proceso y de equipo dejando en un segundo plano las elecciones tecnológicas. (SANCHEZ, 2004).



Figura 3. Metodología MSF.

MSF tiene las siguientes características:

- **Adaptable:** es parecido a un compás, usado en cualquier parte como un mapa, del cual su uso es limitado a un específico lugar.
- **Escalable:** puede organizar equipos tan pequeños entre 3 ó 4 personas, así como también, proyectos que requieren 50 personas o más.
- **Flexible:** es utilizada en el ambiente de desarrollo de cualquier cliente.
- **Tecnología Agnóstica:** porque puede ser usada para desarrollar soluciones basadas sobre cualquier tecnología. (SANCHEZ, 2004)

MSF se compone de varios modelos encargados de planificar las diferentes partes implicadas en el desarrollo de un proyecto: Modelo de Arquitectura del Proyecto, Modelo de Equipo, Modelo de Proceso, Modelo de Gestión del Riesgo, Modelo de Diseño de Proceso y finalmente el modelo de Aplicación.

- *Modelo de Arquitectura del Proyecto:* Diseñado para acortar la planificación del ciclo de vida. Este modelo define las pautas para construir proyectos empresariales a través del lanzamiento de versiones.
- *Modelo de Equipo:* Este modelo ha sido diseñado para mejorar el rendimiento del equipo de desarrollo. Proporciona una estructura flexible para organizar los equipos de un proyecto. Puede ser escalado dependiendo del tamaño del proyecto y del equipo de personas disponibles. (MENDOZA, 2004)
- *Modelo de Proceso:* Diseñado para mejorar el control del proyecto, minimizando el riesgo, y aumentar la calidad acortando el tiempo de entrega. Proporciona una estructura de pautas a seguir en el ciclo de vida del proyecto, describiendo las fases, las actividades, la liberación de versiones y explicando su relación con el Modelo de equipo.
- *Modelo de Gestión del Riesgo:* Diseñado para ayudar al equipo a identificar las prioridades, tomar las decisiones estratégicas correctas y controlar las emergencias que puedan surgir. Este modelo proporciona un entorno estructurado para la toma de decisiones y acciones valorando los riesgos que puedan provocar.
- *Modelo de Diseño del Proceso:* Diseñado para distinguir entre los objetivos empresariales y las necesidades del usuario. Proporciona un modelo centrado en el usuario para obtener un diseño eficiente y flexible a través de un enfoque iterativo. Las fases de diseño conceptual, lógico y físico proveen tres perspectivas diferentes para los tres tipos de roles: los usuarios, el equipo y los desarrolladores.
- *Modelo de Aplicación:* Diseñado para mejorar el desarrollo, el mantenimiento y el soporte, proporciona un modelo de tres niveles para diseñar y desarrollar aplicaciones de software. Los servicios utilizados en este modelo son escalables, y pueden ser usados en un solo ordenador o incluso en varios servidores. (SANCHEZ, 2004).

1.3.1.2 Metodologías Ágiles.

Para el Instituto Europeo de Software, en la conferencia dada el 14 de noviembre de 2003: Las Metodologías Ágiles o “ligeras” constituyen un nuevo enfoque en el desarrollo de software, mejor aceptado por los desarrolladores de proyectos que los modelos convencionales (ISO-90002, CMM3) debido a la simplicidad de sus reglas y prácticas, su orientación a equipos de desarrollo de pequeño tamaño, su flexibilidad ante los cambios y su ideología de colaboración.

En contraste a las metodologías mencionadas anteriormente, o sea las clásicas, durante los últimos años han aparecido las llamadas metodologías ágiles. Las cuales aportan nuevas técnicas y métodos de trabajo para el desarrollo de cada etapa de un software. En general esta metodología hace un balance entre los procesos y el esfuerzo, ya que tratan de centrarse en las cuestiones necesarias sin perderse en las burocráticas (GALLO & VERGARA).

1.3.1.2.1 Programación Extrema (XP).

La metodología XP, llamada así por sus siglas en inglés Extreme Programming, es una de las metodologías de desarrollo de software más exitosas en la actualidad utilizada para proyectos de corto plazo. La metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto. (ESCRIBANO, 2002).

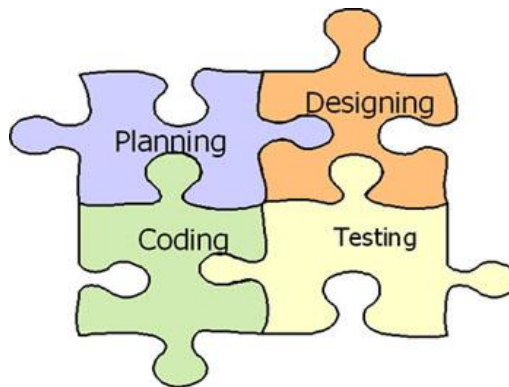


Figura 4. Metodología Extreme Programming.

Características de XP, la metodología se basa en:

- **Pruebas Unitarias:** se basa en las pruebas realizadas a los principales procesos, de tal manera que adelantándonos en algo hacia el futuro, podamos hacer pruebas de las fallas que pudieran ocurrir. Es como si nos adelantáramos a obtener los posibles errores.
- **Re fabricación:** se basa en la reutilización de código, para lo cual se crean patrones o modelos estándares, siendo más flexible al cambio.
- **Programación en pares:** una particularidad de esta metodología es que propone la programación en pares, la cual consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo. Cada miembro lleva a cabo la acción que el otro no está

haciendo en ese momento. Es como el chofer y el copiloto: mientras uno conduce, el otro consulta el mapa.

¿Qué es lo que propone XP?

- Empieza en pequeño y añade funcionalidad con retroalimentación continua.
- El manejo del cambio se convierte en parte sustantiva del proceso.
- El costo del cambio no depende de la fase o etapa.
- No introduce funcionalidades antes que sean necesarias.
- El cliente o el usuario se convierte en miembro del equipo.

Derechos del Cliente

- Decidir que se implementa.
- Saber el estado real y el progreso del proyecto.
- Añadir, cambiar o quitar requerimientos en cualquier momento.
- Obtener lo máximo de cada semana de trabajo.
- Obtener un sistema funcionando cada 3 ó 4 meses.

Derechos del Desarrollador

- Decidir como se implementan los procesos.
- Crear el sistema con la mejor calidad posible.
- Pedir al cliente en cualquier momento aclaraciones de los requerimientos.
- Estimar el esfuerzo para implementar el sistema.
- Cambiar los requerimientos en base a nuevos descubrimientos.

Lo fundamental en este tipo de metodología es:

- La comunicación, entre los usuarios y los desarrolladores.
- La simplicidad, al desarrollar y codificar los módulos del sistema.
- La retroalimentación, concreta y frecuente del equipo de desarrollo, el cliente y los usuarios finales. (SANCHEZ, 2004).

1.3.2 Justificación de la Metodología seleccionada.

Se selecciona al Proceso Unificado de Desarrollo por dos razones fundamentales a partir del estudio previo de otras metodologías:

- Los analistas actuales del sistema no serán los que implementen la herramienta final, lo cual conlleva a que los clientes exijan la mayor claridad en la especificación de los artefactos generados antes de entregar a los desarrolladores.
- RUP es una metodología de desarrollo de software muy bien estructurada, en fases y flujos, que se ajusta perfectamente a lo que exige el cliente pues tiene como base fundamental del desarrollo generar los artefactos completamente documentados, lo cual permite sin dudas mitigar los riesgos en forma temprana y continua, con un progreso demostrable.

1.3.3 Herramientas.

Las herramientas de Ingeniería de Software Asistida por Computadora o CASE (en inglés *Computer Aided Software Engineering*) surgen como solución a ciertos problemas que existían anteriormente como por ejemplo el manejo de la información en proyectos que eran relativamente grandes. Estas herramientas favorecen el apoyo al desarrollo de software, proporcionando un conjunto de programas de asistencia a los analistas para la Ingeniería de Software durante todo el ciclo de vida del desarrollo del sistema.

Las Herramientas CASE son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero. Estas herramientas nos pueden ayudar en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores entre otras.

1.3.3.1 Rational Rose Enterprise Edition.

Rational Rose es la herramienta CASE desarrollada por los creadores de UML, que cubre todo el ciclo de vida de un proyecto, desde la fase de inicio, formalización del modelo, construcción de los componentes, transición a los usuarios y certificación de las distintas fases. Permite establecer una trazabilidad real entre el modelo (análisis y diseño) y el código ejecutable.

Esta herramienta facilita el desarrollo de software en equipo basado en metodología RUP, cada rol tiene su propia vista de arquitectura (vista de Casos de Uso, vista Lógica, vista de Componentes y vista de Despliegue), pero utilizan un lenguaje común para comprender y comunicar la estructura y funcionalidad del sistema en construcción.

Cada analista, desarrollador o diseñador puede usar Rational Rose para definir y comunicar el negocio, el diseño y la arquitectura de la aplicación que se este desarrollando. Es una completa solución para mostrar de forma gráfica el análisis de los procesos del negocio y los requerimientos del sistema. (IBM, 1997)

Rational Rose Enterprise Edition es la herramienta utilizada para la modelación de los Módulos *Planificación de Disco y Administración de Memoria*. Por poseer una integración total con el lenguaje de modelado y la metodología de desarrollo seleccionados, estando estos tres estrechamente vinculados siendo creados por los mismos autores para conformar una metodología de desarrollo robusta.

1.3.3.2 Lenguaje de Modelado Utilizado.

Se define al UML, llamado así por sus siglas en inglés *Unified Modeling Language*, como lenguaje de modelado seleccionado por ser uno de los más conocido y utilizado en la actualidad.

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

Es importante resaltar que UML es un "lenguaje" para especificar y no para describir métodos o procesos. Se utiliza para definir un sistema de software, para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo. Se puede aplicar en una gran variedad de formas para dar soporte a una metodología de desarrollo de software (tal como el Proceso Unificado Racional), pero no especifica en sí mismo qué metodología o proceso usar. (VERA, 2006)

UML cuenta con varios tipos de diagramas:

Diagramas de Estructura enfatizan en los elementos que deben existir en el sistema modelado:

- Diagrama de clases

- Diagrama de componentes
- Diagrama de objetos
- Diagrama de despliegue

Diagramas de Comportamiento enfatizan en lo que debe suceder en el sistema modelado:

- Diagrama de actividades
- Diagrama de casos de uso
- Diagrama de estados

Diagramas de Interacción son un subtipo de diagramas de comportamiento, que enfatiza sobre el flujo de control y de datos entre los elementos del sistema modelado:

- Diagrama de secuencia
- Diagrama de colaboración. (ORALLO, 2005).

1.3.3.3 Visual Studio 2005

Visual Studio .NET es un IDE desarrollado por Microsoft a partir de 2002. Es para el Sistemas Operativos Microsoft Windows y está pensado, principal pero no exclusivamente, para desarrollar plataformas Win32.

Es un conjunto de aplicaciones completo para la creación tanto de aplicaciones de escritorio como de aplicaciones Web de empresa para trabajo en equipo. Aparte de generar aplicaciones de escritorio de alto rendimiento, se pueden utilizar las eficaces herramientas de desarrollo basado en componentes y otras tecnologías de Visual Studio para simplificar el diseño, desarrollo e implementación en equipo de soluciones para empresas.

Constituye un conjunto completo de herramientas de desarrollo para la construcción de aplicaciones Web ASP, servicios Web XML, aplicaciones para escritorio y aplicaciones móviles. Visual Basic .NET, Visual C++ .NET, Visual C# .NET y Visual J# .NET utilizan el mismo entorno de desarrollo integrado (IDE), que les permite compartir herramientas y facilita la creación de soluciones en varios lenguajes. Asimismo, dichos lenguajes aprovechan las funciones de .NET framework, que ofrece acceso a tecnologías clave para simplificar el desarrollo de aplicaciones Web ASP y servicios Web XML.

Visual Studio .NET proporciona diversas plantillas de proyecto que pueden utilizarse para iniciar el desarrollo de aplicaciones distribuidas sin tener que empezar de cero. Las plantillas de empresa definen la estructura inicial de una aplicación distribuida, y proporcionan una guía de arquitectura y tecnología para el diseño de la aplicación. Aparte de las plantillas de empresa predefinidas, se pueden crear plantillas personalizadas que los programadores pueden utilizar en un entorno de equipo. Por

esas razones en la presente investigación se optó por la herramienta Visual Studio 2005 para realizar el diseño de los prototipos de interfaces.

C# (leído en inglés “C Sharp” y en español “C Almohadilla”) es el nuevo lenguaje de propósito general diseñado por Microsoft para su plataforma .NET. Es un lenguaje sencillo, soporta todas las características propias del paradigma de programación orientada a objetos: encapsulación, herencia y polimorfismo, posibilita la seguridad en el manejo de datos, es un sistema de tipos unificados, toma las mejores características de lenguajes preexistentes como Visual Basic, Java o C++ y los combina en uno solo. (Charte Ojeda, 2002)

1.4 Análisis del Sistema.

Antes de comenzar la realización de un proyecto de software, el paso más importante es el análisis del sistema, se considera un camino esencial. Se desarrolla con el propósito de alcanzar una visión más clara sobre lo que el sistema debe hacer, determinando tanto las necesidades del cliente y los límites del sistema, como su estructura y funcionamiento.

Se entiende por análisis al proceso de ingeniería que busca entender el problema que tendrá que resolver el sistema, definir el alcance del sistema, asegurar que el sistema satisfaga las necesidades del usuario, definir los criterios de aceptación y proporcionar una base para el desarrollo de un sistema. Como resultado del análisis, se desarrolla la especificación de requisitos del software. La revisión es esencial para asegurarse que el cliente y el desarrollador tienen el mismo concepto del sistema. Desgraciadamente, incluso con los mejores métodos, la cuestión es que el problema sigue cambiando.

1.4.1 Rol de Analista.

El rol de Analista de Sistema nace de la necesidad de recopilar, desglosar, catalogar y analizar información necesaria de una empresa para poder proponer nuevos métodos mejores, o modificar los actuales y que así aumente el desempeño de los departamentos dentro de la organización. Un Analista se vale de la información de entrada, los procesos modificadores y la información de salida para definir los procesos intermedios y entender con claridad a la organización. Todos estos flujos y procesos son estudiados sistemáticamente para poder determinar si son los adecuados, si se deben mejorar o si deben ser reemplazados por otros. El Analista del Sistema va construyendo una representación del problema del cliente. Para ello no sólo requiere alimentarse del problema y de las perspectivas del

cliente, sino que debe reconocer y escuchar los planteamientos del cliente desde su propia perspectiva tanto personal como de representación desde los paradigmas, metodologías y experiencias que posee. Luego, realizar una reducción del problema y proponer una solución. El Analista de Sistemas es imprescindible en cualquier organización, debido al abanico de destrezas que posee y los beneficios que produce. Su labor no sólo es estudiar la organización y proponer una solución del sistema automatizado, sino también asesorar, supervisar, recomendar y modificar procesos internos; y algunas veces modificar la estructura misma de la empresa.

El proceso de construcción de software se concibe como un conjunto de tareas altamente especializadas donde está claramente definido el papel de cada categoría profesional como es el Analista, el Diseñador y el Programador. El Analista protagoniza numerosos papeles, y en ocasiones, se ve obligado a mantener un equilibrio al asumir simultáneamente más de uno. Los tres papeles principales que el Analista de Sistemas debe desempeñar según bibliografía consultada son: el de consultor, el de especialista de apoyo o soporte y el de agente de cambio. (Kendall, 2005)

Cuando se comienza el Ciclo de Vida de Desarrollo de Sistemas (CVDS), el Analista cumple el papel de consultor, asesorando a la empresa sobre los mejores métodos y sistemas que se pueden emplear para la óptima gestión de información, recomendando sistemas ya sean de tipo manual o de tipo informático, predominando claro, los sistemas informáticos que le dan la vida a ésta profesión. El experto en soporte se identifica con los últimos pasos del CVDS donde el Analista se desempeña en el asesoramiento de hardware y software, basado en el conocimiento y especialmente en la experiencia, sirviendo el Analista muchas veces de escalón para hacer que el sistema desarrollado tenga éxito. Como Agente de Cambio tiene el papel más importante y difícil, la comunicación con los empleados dentro de la fase de recopilación de información; es probable que los empleados piensen que el sistema los va a sustituir, el Analista debe internalizar que el cambio es en pro de la organización y no de un grupo minoritario o sectorial para posteriormente desarrollar sus actividades de manera regular.

1.5 Ingeniería de Requerimientos.

La Ingeniería de Requerimientos también conocida como "Análisis de requerimientos" o "Especificación de requerimientos" cumple un papel primordial en el proceso de producción de software, ya que se enfoca en la definición de lo que se desea producir. Tomando en cuenta las necesidades y condiciones de los inversores. Su principal tarea consiste en la generación de especificaciones correctas que describan con claridad, sin ambigüedades, en forma consistente y compacta, el comportamiento del sistema; de esta manera, se pretende minimizar los problemas relacionados al desarrollo de sistemas.

1.5.1.1 Estrategias para la captura de requisitos.

La captura de requisitos es una actividad sumamente importante mediante la que el equipo de desarrollo de un sistema de software extrae, de cualquier fuente de información disponible, las necesidades que debe cubrir dicho sistema. El proceso de captura de requisitos puede resultar complejo, principalmente si el entorno de trabajo es desconocido para el equipo de analistas, y depende mucho de las personas que participen en él.

A continuación se realiza un breve estudio de diferentes técnicas para la captura de requisitos en el proceso de desarrollo de todo tipo de software.

1.5.1.1.1 Entrevistas.

Le permiten al analista tomar conocimiento del problema y comprender los objetivos de la solución buscada. A través de esta técnica el equipo de trabajo se acerca al problema de una forma natural la estructura de la entrevista abarca tres pasos: identificación de los entrevistados, preparación de la entrevista, realización de la entrevista y documentación de los resultados (protocolo de la entrevista) (DURAN, RUIZ, & TORO, 1999).

A pesar de que las entrevistas son esenciales en el proceso de la captura de requisitos no es una técnica sencilla de aplicar requiere que el entrevistador sea experimentado y tenga capacidad para elegir bien a los entrevistados y obtener de ellos toda la información posible en un período de tiempo siempre limitado. (PAN, ZHU, & JOHNSON, 2001).

1.5.1.1.2 Desarrollo Conjunto de Aplicaciones (JAD).

Esta técnica resulta una alternativa a las entrevistas. Es una práctica de grupo que se desarrolla durante varios días y en la que participan analistas, usuarios, administradores del sistema y clientes. (IBM, 1997)

Está basada en cuatro principios fundamentales: dinámica de grupo, el uso de ayudas visuales para mejorar la comunicación, mantener un proceso organizado y racional y una filosofía de documentación. Esta técnica presenta una serie de ventajas frente a las entrevistas tradicionales, ya que ahorra tiempo al evitar que las opiniones de los clientes se tengan que contrastar por separado, pero requiere un grupo de participantes bien integrados y organizados.

1.5.1.1.3 Tormentas de Ideas.

Es también una técnica de reuniones en grupo cuyo objetivo es que los participantes muestren sus ideas de forma libre. (RAGHAVAN, ZELESNIK, & FORD, 1994)

Consiste en la mera acumulación de ideas y/o información sin evaluar las mismas. Como técnica de captura de requisitos es sencilla de usar y de aplicar, contrariamente al JAD, puesto que no requiere tanto trabajo en grupo como éste. Además suele ofrecer una visión general de las necesidades del sistema, pero normalmente no sirve para obtener detalles concretos del mismo, por lo que suele aplicarse en los primeros encuentros.

1.5.1.1.4 Mapas de Conceptos (CM).

Son grafos en los que los vértices representan conceptos y las aristas representan posibles relaciones entre dichos conceptos (PAN, ZHU, & JOHNSON, 2001)

Son muy usados dentro de la ingeniería de requisitos, pues son fáciles de entender por el usuario, más aún si el equipo de desarrollo hace el esfuerzo de elaborarlo en el lenguaje de éste, deben ser usados con cautela porque en algunos casos pueden llegar a ser ambiguos si no se acompaña de una descripción textual.

1.5.1.1.5 Casos de Uso.

Los casos de uso permiten mostrar el contorno (actores) y el alcance requisitos funcionales expresados como casos de uso) de un sistema. Un caso de uso describe la secuencia de interacciones que se producen entre el sistema y los *actores* del mismo para realizar una determinada función. Los actores son elementos externos (personas, otros sistemas, etc.) que interactúan con el sistema Un actor puede participar en varios casos de uso y un caso de uso puede interactuar con varios actores. La ventaja esencial de los casos de uso es que resultan muy fáciles de entender para el usuario o cliente. (UML, 2001)

1.5.1.1.6 Listas de Chequeo y Cuestionarios.

Requiere que el analista conozca el ámbito del problema en el que está trabajando. Consiste en redactar un documento con preguntas cuyas respuestas sean cortas y concretas Este cuestionario será cumplimentado por el grupo de personas entrevistadas o simplemente para recoger información en forma independiente de una entrevista.

1.5.1.2 Estrategia Seleccionada.

Para la captura de los requisitos de la aplicación se realizaron un conjunto de entrevistas y reuniones de trabajo con los profesores: Ing. Dusniel Horta Centeno e Ing. Carlos Yasmany Hidalgo García. Ambos profesores pertenecen al colectivo de la asignatura Sistemas Operativos de la Facultad 3. De esta manera resultó escogida la Entrevista no estructurada como estrategia seleccionada.

1.6 Conclusiones del capítulo.

Luego de un profundo estudio sobre los principales aspectos relacionados con los Laboratorios Virtuales, esta investigación se concentra en las diferentes metodologías de desarrollo de software que permitió conocer cuál es la que más se ajusta para la modelación de los proyectos de este tipo y se decidió que la metodología a utilizar será el Rational Unified Process (RUP), por sus características. Se define como lenguaje de modelado el UML, como herramienta de modelado se emplea el Rational Rose Enterprise Edition, pues es la que mejor soporta la Metodología y el Lenguaje de modelado seleccionados. A través del estudio de las diferentes estrategias de captura de requisitos se conocieron las más adecuadas a utilizar teniendo en cuenta las características y condiciones del cliente. De esta manera queda definida la metodología y las herramientas a utilizar, con lo que se considera que a partir de este momento se está en condiciones de dar inicio al proceso de Modelado del Negocio.

2 MODELACIÓN DEL NEGOCIO

2.1 Introducción.

La tarea en este capítulo consiste en concretar, desde la perspectiva del analista, las características del sistema que se va a desarrollar. Se describen los procesos involucrados que propiciará en un futuro poder realizar un diseño exitoso. Se presenta una propuesta general de como debe funcionar el sistema, se incluyen además la elaboración del modelo de negocio, los diagramas de actividades para cada uno de los casos de usos seleccionados, y el modelo de objeto.

2.2 Descripción de los Procesos de Automatización.

Con el sistema se automatizarán los siguientes procesos:

- Inicializar Memoria Física.
- Simular Memoria en Mono Programación.
- Simular Memoria en Multiprogramación con Particiones Fijas.
- Simular Memoria en Multiprogramación con Particiones Variables.
- Conformar Memoria Virtual.
- Ejecutar Páginas en Memoria Física.
- Planificar Solicitudes de Acceso al Disco Duro.
- Calcular Parámetros de Disco Duro.

Los contenidos que conforman estos procesos y la descripción de cada uno de ellos se muestra a continuación:

Inicializar Memoria Física: El proceso de particionado de la memoria consiste en dividir la misma en secciones de diferentes tamaños, algunos de los cuales serán reservados exclusivamente para el Sistemas Operativos. Por lo que es parte importante definir cuanto espacio del total de la memoria física se destinará para uso del Sistemas Operativos. En cada partición se alojarán procesos o

segmentos de estas llamadas páginas de procesos, las cuales deben coincidir a lo sumo con el tamaño de la partición.

Simular Memoria en Monoprogramación: El uso de la Monoprogramación como uno de los primeros esquemas de administración de memoria, consiste en dividir la misma en dos partes, una para el programa de usuario y la otra para la porción residente del Sistemas Operativos. Generalmente la zona ocupada por el Sistemas Operativos se almacena en la parte baja de la memoria y el programa usuario en la parte alta. La Monoprogramación implica la existencia de un solo proceso en memoria.

Simular Memoria en Multiprogramación con Particiones Fijas: La multiprogramación presupone la existencia en memoria de varios procesos que comparten el tiempo del procesador. En este caso la memoria es particionada en varias áreas fijas llamadas particiones o regiones. Donde cada programa se almacenará en una de estas particiones. Una situación crítica en la multiprogramación con particiones fijas es la decisión de los tamaños. Si se dan tamaños pequeños, los programas grandes puede que no quepan en ninguna. Por otra parte, si se les dan tamaños grandes, entonces los programas pequeños malgastan memoria, nombrándose a este desperdicio de espacio dentro de las particiones *fragmentación interna*. El número de particiones existentes y la longitud de estas son fijos, porque se definen estáticamente en el momento de la carga del sistema.

Simular Memoria en Multiprogramación con Particiones Variables: En la multiprogramación con particiones variables no existe un número predeterminado de particiones ni las longitudes son fijas, sino que dependen de las necesidades de cada proceso. A cada espacio libre se le llama "hueco". El Sistemas Operativos mantiene una estructura de datos indicando que espacios están libres (los huecos). Si el espacio aparece, entonces se le asigna la cantidad que necesita. Si existiera sobrante, éste se mantiene como un hueco. Al inicio de la operación toda la memoria dedicada a los procesos usuarios constituye un gran hueco. A medida que avanza el tiempo la memoria pudiera quedar fragmentada en varias particiones y varios huecos pequeños que no pueden ser usados por los trabajos que están esperando, ya que sus requerimientos son superiores. A esta situación se le llama *fragmentación externa*.

Conformar Memoria Virtual: Consiste en permitir la ejecución de procesos que puedan no estar completamente en memoria. La forma más común de instrumentar la memoria es por medio del paginado, esta es una técnica que permite dividir la memoria física en particiones de igual tamaño

nombradas *marcos de páginas*. A su vez cada proceso que llega nuevo a memoria se divide en partes de igual tamaño que los marcos de páginas, donde cada una de estas partes recibe el nombre de *página* y se coloca en un marco no ocupado. Esta técnica puede existir incluso sin usar memoria virtual.

En un sistema paginado, toda dirección generada está compuesta por dos partes: un número de página y el desplazamiento dentro de ésta. Es decir, un grupo de bits codificarán la primera información y otros la segunda. Ahora el objetivo es poder colocar cada página en cualquier marco que se encuentre disponible. Para lograrlo se requiere mantener la correspondencia entre las páginas y los bloques donde se encuentran almacenadas. Para cada proceso se mantiene una tabla indexada por el número de página donde cada entrada contendrá la dirección base del marco de página correspondiente. Al Sistemas Operativos le corresponde, en el momento de la carga de un trabajo, localizar la cantidad de marcos de página libres que sean necesarios, construir la tabla de páginas y colocar en un registro base la dirección de ésta, si el trabajo va a ser ejecutado de inmediato.

Ejecutar Páginas en Memoria Física: Para sustituir una página residente en memoria se utilizan diferentes algoritmos de reemplazo como son los siguientes:

- OPT.
- FIFO (First Come First Out).
- LRU (Least Recently Used).
- LFU (Least Frequently Used).
- MFU (Most Frequently Used).
- NRU (Not Recently Used).

Planificar Solicitudes de Acceso al Disco Duro: Existen varios algoritmos para atender las solicitudes que se le van haciendo a un disco estos tienen el propósito de disminuir el tiempo de búsqueda de las pistas como son:

- FCFS (“First Come First Served”).
- SSF o SSTF (“Shortest Seek-Time First”).
- SCAN.
- C-SCAN.
- LOOK.
- C-LOOK.

Calcular Parámetros de Disco Duro: La unidad más pequeña en un disco duro es el sector. Cuando se lee o escribe información al disco, lo que se hace es leer o escribir sectores. Para lograr una buena eficiencia en el uso del disco duro hay que tratar de leer o escribir la mayor cantidad de sectores, en el menor tiempo posible, según sean las solicitudes acumuladas. El tiempo de acceso a un bloque o sector está compuesto de tres partes.

- Se requiere mover la cabeza de lectura/escritura a la pista apropiada.
- Esperar a que el sector deseado pase por debajo del cabezal.
- Por último, se realizará el traspaso de la información (512 bytes) entre el disco.

De los tres factores antes indicados, el único a considerar desde el punto de vista de optimizar el tiempo promedio de servicio a las solicitudes es el tiempo de búsqueda y por ello los algoritmos de planificación tienen como objetivo disminuir este componente.

2.3 Modelo de Negocio.

De acuerdo a RUP, un modelo representa una forma de contemplar el sistema que se modela, según el punto de vista que se elabora. El modelado del negocio es una técnica para comprender los procesos de negocio de la organización. (Jacobson, Booch, & Rumbaugh, 2000)

Modelar el proceso de negocio es una parte esencial de cualquier proceso de desarrollo de software. Permite al analista capturar el esquema general y los procedimientos que gobiernan el negocio. Este modelo provee una descripción de dónde se va a ajustar el sistema de software considerado dentro de la estructura organizacional y de las actividades habituales. También provee la justificación para la construcción del sistema de software al capturar las actividades manuales y los procedimientos automatizados habituales que se incorporarán en nuevo sistema, con costos y beneficios asociados. (Sparks)

Como un modelo preliminar del negocio, permite al analista capturar los eventos, las entradas, los recursos y las salidas más importantes vinculadas con el proceso de negocio. (Sparks)

Un Modelo de Negocio se desarrolla en dos pasos:

1. Confección de un modelo de Casos de Uso del Negocio donde se identifiquen los procesos generales del Negocio, los actores y los Casos de Uso que estos utilizan. Este modelo permite a los modeladores comprender que valor proporciona el Negocio a los actores.
2. Desarrollo de un modelo de Objetos del Negocio, compuesto de trabajadores, entidades del Negocio y unidades de trabajo que realizan los Casos de Uso. A estos objetos se asocian las

Reglas y normas del Negocio. El objetivo es que los trabajadores, entidades y unidades de trabajo definidas realicen los Casos de Uso del Negocio de la manera más eficiente posible (rápido, con precisión y bajos costos). (Jacobson, Booch, & Rumbaugh, 2000).

Los objetivos del modelamiento del negocio son:

- Comprender la estructura y la dinámica de la organización en la cual se va a implantar un sistema.
- Comprender los problemas actuales de la organización e identificar las mejoras potenciales.
- Asegurar que los consumidores, usuarios finales y desarrolladores tengan un entendimiento de la organización.
- Derivar los requerimientos del sistema que va a soportar la organización.

En resumen, el objetivo del modelo del negocio es describir los procesos, existentes u observados, con el propósito de comprenderlos. Se especifican aquí qué procesos del negocio soportará el sistema. Además de identificar los objetos del dominio o del negocio, implicados, este modelo establece las competencias que se requieren de cada proceso: sus trabajadores, sus responsabilidades y las operaciones que llevan a cabo. (Jacobson, Booch, & Rumbaugh, 2000)

2.3.1 Reglas del Negocio.

1. El profesor es el encargado de realizar las solicitudes pedidas por los estudiantes.
2. La unidad de medida a utilizar en la partición de la memoria debe estar dada en (KB).
3. La parte de la memoria utilizada para el Sistemas Operativos debe ser menor que el tamaño total definido para la memoria.
4. Para cumplir con la visualización de un proceso debe de haber llenados los datos requeridos inicialmente.
5. El tamaño de un proceso no debe de exceder el de una partición.
6. Un proceso para ejecutarse tiene que estar en memoria física.
7. En cada marco se almacena una sola página de un proceso.

2.3.2 Actores del Negocio.

Un actor del negocio es cualquier individuo, grupo, organización, máquina o sistema de información externo que interactúa con el negocio. El término actor significa el rol que algo o alguien juega cuando interactúa con el negocio para beneficiarse de sus resultados. (Jacobson, Booch, & Rumbaugh, 2000)

Actores del Negocio	Justificación
Estudiante	Se considera en este rol a cualquier estudiante que inicia el proceso de solicitud al profesor para la realización de un ejercicio de Administración de Memoria o Planificación de Disco.

2.3.3 Trabajadores del Negocio.

Un trabajador define el comportamiento y las responsabilidades de un individuo que actúa en el negocio realizando una o varias actividades, interactuando con otros trabajadores del negocio y manipulando entidades del negocio. (Jacobson, Booch, & Rumbaugh, 2000)

Trabajadores del Negocio	Justificación
Profesor	Se encarga de atender y cumplir las solicitudes realizadas por los estudiantes cuando presentan dudas en la realización de un ejercicio de Administración de Memoria o Planificación de Disco.

2.3.4 Diagrama de Casos de Uso del Negocio.

El diagrama de casos de uso del negocio representa gráficamente los procesos del negocio y su interacción con los actores del negocio. (Boggs & Boggs, 2002).

A continuación se muestra la figura correspondiente al diagrama de casos de uso del negocio.

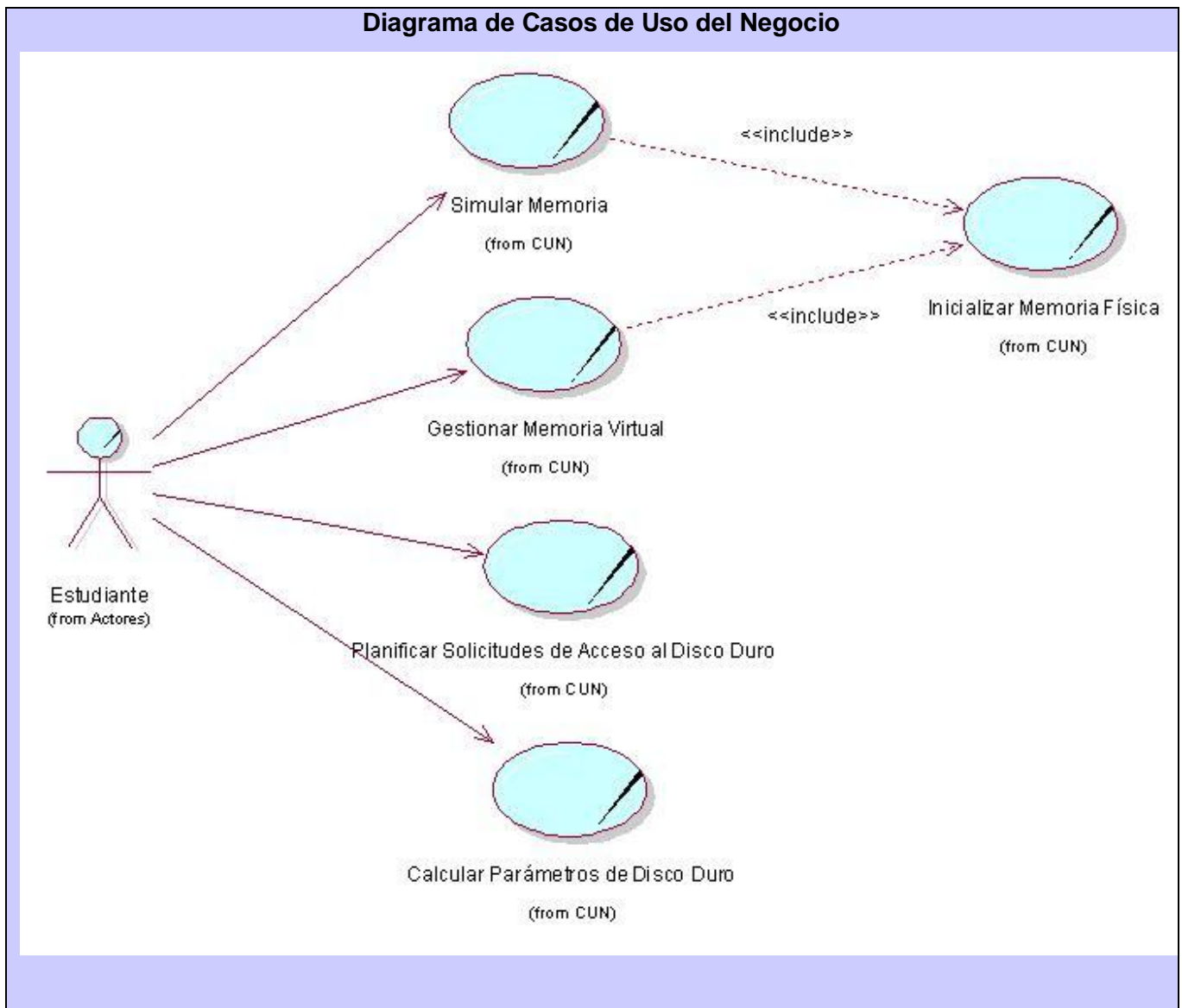


Figura 5 Diagrama de CU del Negocio.

2.3.5 Descripción de los Casos de Uso del Negocio.

La descripción de los Casos de Uso del Negocio es el complemento del Diagrama de Casos de Uso. Permite conocer en detalle como se realizan los procesos, las acciones de los Actores y Trabajadores y los resultados de esto. En ella se especifican, el nombre del Caso de Uso, los Actores y Trabajadores involucrados, las condiciones previas y posteriores a la ejecución de las acciones, el flujo de eventos y

los flujos alternos que se puedan generar, la prioridad y un breve resumen que permita conocer el contenido del Caso de Uso. (Boggs & Boggs, 2002)

Descripción textual del caso de uso del negocio: Inicializar Memoria Física

Caso de Uso:	Inicializar Memoria Física
Código:	CU_1
Actores:	Estudiante
Trabajadores	Profesor
Propósito:	Inicializar la memoria física, constituye la base para el resto de las actividades de trabajo con memoria.
Resumen:	Este caso de uso se inicia cuando un estudiante necesita realizar una actividad de simulación de la memoria por cualquier esquema de administración que se adopte, ya que es necesario conocer cual es la capacidad de memoria con que se cuenta para poder simular así como la reservada para uso del Sistemas Operativos.
Precondiciones:	Debe tenerse la capacidad total de memoria física y la cantidad de esta requerida por el Sistemas Operativos.

Flujo Normal de Eventos

Acción del Actor	Respuesta del Negocio
1. El estudiante se presenta ante el profesor con un ejercicio de simulación para resolver.	2. Solicita los datos necesarios para establecer los parámetros de inicialización de la memoria.
3. Brinda los datos requeridos: <ul style="list-style-type: none"> • Capacidad total de la memoria a establecer. • Unidad de medida a utilizar (KB). • Capacidad de memoria destinada al SO. 	4. Verifica que la capacidad suministrada para reservación del Sistemas Operativos no exceda el total de memoria proporcionada. 5. Lleva a cabo la representación gráfica en papel o pizarra de la memoria física del sistema y su distribución actual acorde a los datos suministrados.
	6. Explica al estudiante el proceso de inicialización de memoria.
7. Recibe la explicación, incluyendo el porcentaje	

que representa el Sistemas Operativos.	
Flujos Alternos	
Acción del Actor	Respuesta del Negocio
	4.1 Informa que los datos solicitados son incorrectos.
Diagrama de Actividades para este Caso de Uso	
Ver Anexo 1	
Poscondiciones:	La inicialización de la memoria física queda lista, mostrándose el espacio destinado a los procesos de usuario.

Descripción textual del caso de uso del negocio: Simular memoria.

Caso de Uso:	Simular Memoria.
Código:	CU_2
Actores:	Estudiante
Trabajadores	Profesor
Propósito:	El presente caso de uso se encarga de la simulación de la memoria física para un entorno monoprogramado o multiprogramado en presencia de particiones fijas y variables.
Resumen:	El caso de uso se inicia cuando el estudiante solicita al profesor ayuda en la realización de un ejercicio de simulación de procesos en memoria y finaliza con la visualización correcta, en papel o pizarra, del estado de la memoria, así como sus particiones y proceso(s) en ella luego de la simulación.
Precondiciones:	Debe estar inicializada la memoria.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Negocio
1. Solicita al profesor ayuda en la simulación de un ejercicio de Administración de Memoria física.	2. Solicita el esquema de administración que se usará: a) <i>Simular Memoria en Monoprogramación.</i> b) <i>Simular Memoria en Multiprogramación.</i>
3. Selecciona el esquema deseado y lo informa.	4. En caso de seleccionar la opción (a) ir a la

	sección “ <i>Simular Memoria en Monoprogramación</i> ”; en caso de seleccionar la opción (b) , ir a la sección “ <i>Simular Memoria en Multiprogramación</i> ”.
Sección “Simular Memoria en Monoprogramación”	
Acción del Actor	Respuesta del Negocio
1. Solicita ayuda al profesor para llevar a cabo la simulación de un proceso en esquema de Monoprogramación de la memoria.	
	2. Inicia el caso de uso Inicializar Memoria Física.
	3. Solicita los datos necesarios para llevar a cabo la simulación de un proceso en esquema de Monoprogramación de la memoria.
4. Brinda los datos solicitados: <ul style="list-style-type: none"> • Identificador del nuevo proceso a entrar en memoria. • Capacidad de memoria requerida por el nuevo proceso. 	5. Chequea que la capacidad de memoria requerida por el proceso, no exceda el total existente de memoria física para procesos de usuario.
	6. Lleva a cabo la representación en papel o pizarra del proceso en la memoria disponible.
	7. Muestra y explica al estudiante los resultados obtenidos y el significado práctico.
8. Pide el cálculo del parámetro correspondiente a fragmentación interna	9. Calcula le fragmentación interna que se produce $TamPartición - TamProceso$, e informa al estudiante.
10. Recibe la explicación correspondiente.	
Flujos Alternos	
Acción del Actor	Respuesta del Negocio
	5.1 Informa que los datos son incorrectos.

Diagrama de Actividades para este Caso de Uso	
Ver Anexo 2	
Sección “Simular Memoria en Multiprogramación”	
Acción del Actor	Respuesta del Negocio
	<p>1. Solicita el modelo de multiprogramación deseado:</p> <p style="margin-left: 40px;">a) <i>Simular Memoria en Multiprogramación con Particiones Fijas.</i></p> <p style="margin-left: 40px;">b) <i>Simular Memoria en Multiprogramación con Particiones Variables.</i></p>
2. El estudiante selecciona la opción deseada.	3. En caso de seleccionar la opción (a) ir a la sección “ <i>Simular Memoria en Multiprogramación con Particiones Fijas</i> ”; en caso de seleccionar la opción (b) , ir a la sección “ <i>Simular Memoria en Multiprogramación con Particiones Variables</i> ”.
Sección “Simular Memoria en Multiprogramación con Particiones Fijas”	
Acción del Actor	Respuesta del Negocio
1. Solicita explicación sobre el trabajo con sistemas multiprogramado.	
	2. Inicia el caso de uso Inicializar Memoria Física.
	3. Solicita los datos necesarios para llevar a cabo la simulación solicitada y su posterior explicación.
4. Brinda los siguientes datos: <ul style="list-style-type: none"> • Cantidad de particiones a crear. • Tamaño de cada partición. 	5. Valida los tamaños suministrados para cada partición respecto a lo que va quedando disponible en la memoria total.
	6. Grafica en papel o pizarra la distribución de

	la memoria en particiones fijas.
	7. Muestra el grado de multiprogramación del nuevo sistema particionado.
8. Recibe la explicación gráfica de la estructura de la memoria particionada.	9. Solicita el tipo de técnica de control de procesos a memoria: <ul style="list-style-type: none"> • Una única cola de entrada. • Múltiples colas de entrada.
10. Informa la técnica a utilizar	11. Solicita la acción a realizar: <ul style="list-style-type: none"> a) Asignar nuevo proceso a una partición. b) Terminar ejecución de un proceso.
12. Informa la opción deseada.	13. En caso de a) ir a la sección “ <i>Asignar nuevo proceso a una partición</i> ”; en caso de b) , ir a la sección “ <i>Terminar ejecución de un proceso</i> ”.
Flujo Alterno	
Acción del Actor	Respuesta del Negocio.
	4.1 Si no es válido el tamaño para cada partición informa al estudiante.
	4.2 Vuelve al paso 3.
Sección “Asignar nuevo proceso a una partición	
Acción del Actor	Respuesta del Negocio
	1. Solicita los datos necesarios.
2. Da la información solicitada, que consta de los siguientes elementos: <ul style="list-style-type: none"> • Identificador del nuevo proceso. • Capacidad de memoria requerida por el nuevo proceso. 	3. Verifica que la cantidad de memoria requerida por el nuevo proceso coincide o es menor que el tamaño de alguna de las particiones libres.
	4. Asigna el proceso a la partición disponible.
	5. Actualiza la distribución de la memoria con

	todos los procesos en sus respectivas particiones.
Flujos Alternos	
Acción del Actor	Respuesta del Negocio
	4.1 Si no existe al menos una partición a la que pueda asignarse el nuevo proceso, este se almacena en la cola de entrada de procesos a ejecutar.
Diagrama de Actividades para este Caso de Uso Ver Anexo 3	
Sección “Terminar Proceso”	
Acción del Actor	Respuesta del Negocio
	1. Verifica que esté reflejado al menos un proceso en la memoria y solicita al estudiante el proceso que desea terminar.
2. Informa el identificador del proceso a terminar.	3. Termina el proceso en cuestión, eliminándolo de la lista de procesos en la memoria y liberando la partición ocupada por este.
	4. Verifica si existe algún proceso en la(s) cola(s) de entrada.
	5. Selecciona el nuevo proceso a ejecutarse siguiendo los siguientes criterios: <ul style="list-style-type: none"> • El proceso debe “caber” en la partición liberada. • Debe tomarse de todos los procesos que cumplan con el primer criterio aquel que lleve más tiempo en espera.
	6. Actualiza gráficamente en la pizarra o el papel la distribución de la memoria de la sección “Simular memoria en

	<p>multiprogramación con particiones fijas” con todos los procesos en sus respectivas particiones, el estado de la fragmentación interna y la cola de procesos en espera.</p>
7. Recibe la explicación gráfica por parte del profesor.	
8. Termina el caso de uso del negocio.	
Flujos Alternos	
Acción del Actor	Respuesta del Negocio
	1.1 Si no existe ningún proceso en ejecución se informa al estudiante.
Diagrama de Actividades para este Caso de Uso	
Ver Anexo 4	
Sección “Simular Memoria en Multiprogramación con Particiones Variables”	
Acción del Actor	Respuesta del Negocio
1. Solicita explicación sobre el trabajo con sistemas multiprogramado.	
	2. Inicia el caso de uso “Inicializar Memoria Física”
	3. Solicita al estudiante la técnica de asignación de memoria a utilizar.
4. Informa la técnica seleccionada que pueden ser:	5. Solicita al estudiante la acción a realizar:
<ul style="list-style-type: none"> • Mapa de bits. • Listas enlazadas. <ul style="list-style-type: none"> ○ Primer Ajuste. ○ Mejor Ajuste. ○ Peor Ajuste. ○ Siguiendo Ajuste. • Sistema camarada. 	<ul style="list-style-type: none"> a) <i>Colocar un nuevo proceso en memoria.</i> b) <i>Eliminar un proceso de la memoria.</i> c) <i>Compactar memoria.</i>

6. Selecciona la opción deseada.	7. En caso de seleccionar la opción (a) ir a la sección “Colocar nuevo proceso en memoria”; en caso de (b) ir a la sección “Eliminar un proceso de la memoria”, en caso de (c) ir a la sección “Compactar Memoria Física”.
Flujos Alternos	
Acción del Actor	Respuesta del Negocio
Sección “Colocar un nuevo proceso en memoria”	
Acción del Actor	Acción del Actor
2. Proporciona los siguientes datos: <ul style="list-style-type: none"> • Identificador del nuevo proceso. • Capacidad de memoria requerida. 	<p>1. Solicita los datos necesarios para colocar un proceso en memoria.</p> <p>3. El profesor verifica que la memoria requerida por el nuevo proceso coincide o es menor que el tamaño total de la memoria física disponible.</p> <p>4. Refleja gráficamente el proceso en la memoria y actualiza la distribución de la misma con todos los procesos en sus respectivas particiones, el estado de la fragmentación externa y la cola de procesos en espera.</p>
Flujos Alternos	
Acción del Actor	Respuesta del Negocio
	3.1 Sino cabe el profesor le indica al estudiante que el proceso a asignar sobrepasa el tamaño de todos los huecos de memoria libres y comienza a desarrollar el flujo “Compactar Memoria Física”.
	3.2 Coloca el proceso en la cola de procesos en espera.
Diagrama de Actividades para este Caso de Uso	
Ver Anexo 5	
Sección “Eliminar un proceso de la memoria”	

Acción del Actor	Respuesta del Negocio
	1. Verifica que en la memoria graficada se esté ejecutando al menos un proceso.
	2. Muestra una lista de los procesos en ejecución.
3. Informa el identificador del proceso a terminar.	4. Elimina el proceso de la lista de procesos en ejecución y libera la partición ocupada por este.
	5. Verifica si existe un hueco adyacente a la partición liberada.
	6. Se unen los huecos adyacentes y forman uno libre de mayor tamaño.
	7. Verifica si existe algún proceso en la cola de entrada.
	8. Verifica si el próximo proceso a entrar en memoria cabe en algún hueco disponible.
	9. Se coloca el proceso en la memoria.
	10. Actualiza gráficamente en papel o pizarra la distribución de la memoria con todos los procesos en sus respectivas particiones, el estado de la fragmentación externa y la cola de procesos en espera.
11. El estudiante recibe la explicación detallada.	
Flujos Alternos	
Acción del Actor	Respuesta del Negocio
	1.1 Si no existe ningún proceso en ejecución el profesor informa que no es posible efectuar la acción.
	5.1 Si no existe hueco adyacente lo informa al estudiante.
	7.1 Si no hay procesos en la cola, informa al estudiante que no hay ninguno esperando para ser atendido.

	8.1 Si no cabe el profesor le indica al estudiante que el proceso a asignar sobrepasa el tamaño de todos los huecos libres y comienza a desarrollar el flujo “Compactar Memoria Física”.
Diagrama de Actividades para este Caso de Uso Ver Anexo 6	
Sección “Compactar Memoria Física”	
Acción del Actor	Respuesta del Negocio
	1. Se reubican los procesos en un extremo de la memoria y se actualiza.
	2. Comprueba si existe algún proceso en la cola de entrada.
	3. Comprueba si el próximo proceso a entrar en memoria cabe en algún hueco disponible.
	4. Actualiza gráficamente en papel o pizarra la distribución de la memoria con todos los procesos en sus respectivas particiones, el estado de la fragmentación externa y la cola de procesos en espera.
Flujos Alternos	
Acción del Actor	Respuesta del Negocio
	2.1 Si no hay procesos en la cola, informa al estudiante que no hay ninguno esperando para ser atendido.
	3.1 Si no cabe el profesor le indica al estudiante que el proceso a asignar sobrepasa el tamaño de todos los huecos libres
Diagrama de Actividades para este Caso de Uso Ver Anexo 7	
Poscondiciones:	La memoria física queda estructurada con todas las particiones y procesos definidos durante todo el proceso de simulación. Representación gráfica del esquema de memoria con particiones variables.

Descripción textual del caso de uso del negocio: Gestionar Memoria Virtual.

Caso de Uso:	Gestionar Memoria Virtual
Código:	CU_3
Actores:	Estudiante
Trabajadores:	Profesor
Propósito:	El presente caso de uso se encarga de la gestión de la memoria virtual en presencia de conformar la misma, reemplazar una página existente en ella, siempre haciendo uso del paginado.
Resumen:	El caso de uso se inicia cuando un estudiante se presenta ante el profesor con un ejercicio de Gestión de memoria virtual que puede implica primeramente la conformación de la memoria virtual para luego dar paso a la entrada de procesos en memoria o el reemplazo de páginas. El caso de uso finaliza con la representación grafica de la memoria virtual y física y los procesos en la misma.
Precondiciones:	Debe tenerse el tamaño total de la memoria virtual y el tamaño de cada marco de página.

Flujo Normal de Eventos

Acción del Actor	Respuesta del Negocio
1. Solicita al profesor gestionar la memoria virtual mediante un ejercicio propuesto.	2. El profesor solicita los datos necesarios para conformar la memoria virtual.
3. Informa los siguientes datos. <ul style="list-style-type: none"> • Tamaño total de la memoria virtual. • Tamaño de cada marco de página. 	4. Valida los datos suministrados.
	5. Con la información recibida calcula la cantidad de marcos tanto de la memoria física como de la memoria virtual.
	6. Lleva a cabo la representación gráfica en papel o pizarra tanto de la memoria virtual como de la memoria física y su distribución actual acorde a los datos suministrados.

7. Recibe la explicación gráfica.	8. Solicita la acción que desea gestionar el estudiante: <i>a. Adicionar proceso a memoria.</i> <i>b. Ejecutar página en Memoria Física.</i>
9. Informa la opción deseada.	10. En caso de seleccionar la opción: (a): ir a la sección “ <i>Adicionar proceso a memoria</i> ”; (b): ir a la sección “ <i>Ejecutar página en Memoria Física</i> ”.
Flujos Alternos	
Acción del Actor	Respuesta del Negocio
	4.1 Informa que los datos son incorrectos.
Sección “Adicionar proceso a memoria”	
Acción del Actor	Respuesta del Negocio
	1. Solicita los datos necesarios.
2. Brinda los siguientes datos: <ul style="list-style-type: none"> • Identificador del proceso. • Cantidad de memoria requerida por el proceso. 	3. Valida los datos recibidos. 4. Procede a dividir el proceso en páginas de igual tamaño que el establecido para los marcos.
	5. Representa cada página del proceso en la memoria virtual.
	6. Explica al estudiante la representación grafica de la memoria virtual.
7. Recibe la explicación del profesor.	
Flujos Alternos	
Acción del Actor	Respuesta del Negocio
	3.1 Informa que los datos son incorrectos.
Diagrama de Actividades para este Caso de Uso Ver Anexo 8	
Sección “Ejecutar página en Memoria Física”	
Acción del Actor	Respuesta del Negocio

1. Desea realizar la ejecución de una página y solicita al profesor la ayuda necesaria.	
	2. Solicita la información necesaria relacionada con el proceso.
<p>3. El estudiante informa los siguientes datos:</p> <ul style="list-style-type: none"> • Identificador del proceso. • Página del proceso en cuestión. • El tipo de remplazo: <ul style="list-style-type: none"> ○ Reemplazo Local. ○ Reemplazo Global. <p>El algoritmo de remplazo:</p> <ul style="list-style-type: none"> • OPT (Algoritmo óptimo). • FIFO (First Come First Out). • LRU (Least Recently Used). • LFU (Least Frequently Used). • MFU (Most Frequently Used). • NRU (Not Recently Used). 	4. Verifica que los datos sean correctos.
	5. Procede a realizar en la pizarra o en papel la ejecución del proceso y con ello el remplazo de páginas.
	6. Actualiza a memoria y muestra gráficamente en papel o pizarra el estado de la memoria.
	7. Explica al estudiante el estado de la memoria e informa el total de remplazos de páginas existentes.
8. El estudiante recibe la explicación	
9. Finaliza el caso de uso.	
Flujos Alternos	
Acción del Actor	Respuesta del Negocio
	4.1 Informa que los datos son incorrectos.

Diagrama de Actividades para este Caso de Uso

Ver Anexo 9

Poscondiciones:	Tanto la memoria virtual como la memoria física deben quedar estructuradas con los marcos de igual tamaño correspondientes y las páginas de procesos existentes en cada uno.
------------------------	--

Descripción textual del caso de uso del negocio: Planificar solicitudes de acceso al Disco Duro.

Caso de Uso:	Planificar solicitudes de acceso al Disco Duro.
Código:	CU_4
Actores:	Estudiante (Inicia).
Trabajadores	Profesor
Propósito:	Servir las solicitudes de acceso a los cilindros de un disco duro, mediante la resolución de ejercicios usando diferentes algoritmos de Planificación de Disco, en la asignatura de Sistemas Operativos.
Resumen:	El caso de uso se inicia cuando el estudiante solicita la ayuda del profesor en la resolución de algún ejercicio de Planificación de Disco duros mediante cualquiera de los algoritmos estudiados en las clases de Sistemas Operativos.
Precondiciones:	Debe tenerse la cantidad de cilindros que componen el disco, las listas de solicitudes a servir y los algoritmos de planificación.

Flujo Normal de Eventos

Acción del Actor	Respuesta del Negocio
1. El caso de uso se inicia cuando el estudiante solicita al profesor que le ayude a resolver un ejercicio de Planificación de Disco mediante uno de los algoritmos estudiados en clases.	2. El profesor solicita al estudiante los datos necesarios para resolver el ejercicio.

<p>3. Informa los siguientes datos:</p> <ul style="list-style-type: none">• Cantidad de cilindros que componen el disco.• Cilindro que se sirve actualmente.• Cilindro servido previamente.• Lista de solicitudes a servir.• Instante de llegada de cada solicitud.• Algoritmo de planificación a utilizar.• Tiempo de posicionamiento promedio entre cilindros contiguos.• Tiempo Promedio de Acceso.	<p>4. Verifica la veracidad de los datos proporcionados.</p>
	<p>5. Actualiza en papel o pizarra la tabla de solicitudes y la llena con la lista de estas a servir.</p>
	<p>6. Haciendo uso del algoritmo dado por el estudiante procede a calcular en papel o pizarra cada una de las solicitudes previstas.</p>
	<p>7. Para cada solicitud actualiza la tabla con los siguientes datos:</p> <ul style="list-style-type: none">• Próximo cilindro a servir.• Desplazamiento entre el cilindro actual y el cilindro a servir.
	<p>8. Una vez satisfechas todas las solicitudes calcula los siguientes datos:</p> <ul style="list-style-type: none">• Movimiento total del cabezal del disco.• Tiempo total de servicio.
	<p>9. Explica al estudiante gráficamente en pizarra o papel el funcionamiento del algoritmo y el resto de los resultados obtenidos.</p>
<p>10. Recibe la explicación.</p>	
<p>11. Termina el caso de uso.</p>	

Flujo alterno	
Acción del Actor	Respuesta del Negocio
	4.1 Informa al estudiante la inexactitud de los datos recibidos.
Diagrama de Actividades para este Caso de Uso Ver Anexo 10	
Poscondiciones:	Deben quedar servidas las solicitudes de acceso a los cilindros de un disco duro, utilizando los diferentes tipos de algoritmos.

Descripción del Caso de Uso: Calcular parámetros de Disco Duro.

Caso de Uso:	Calcular parámetros de Disco Duro.
Código:	CU_5
Actores:	Estudiante (Inicia).
Trabajadores:	Profesor
Propósito:	Calcular los parámetros de discos duros más comunes impartidos en las clases de Sistemas Operativos: tanto la capacidad del disco como la conversión de sectores lógicos a físicos y viceversa.
Resumen:	El caso de uso inicia cuando el estudiante se presenta ante el profesor y le pide ayuda acerca de un ejercicio de cálculo de parámetros de discos duros. El profesor realiza los cálculos correspondientes y le explica al estudiante los resultados y el procedimiento correspondiente para realizar las conversiones necesarias.
Precondiciones:	Debe tenerse la cantidad de platos del disco, la cantidad de sectores por pistas y el tamaño de cada sector.

Flujo Normal de Eventos	
Acción del Actor	Respuesta del Negocio
1. Se presenta con un ejercicio de cálculo de parámetros de discos y solicita ayuda al profesor.	2. Solicita al estudiante que lea el problema propuesto.
3. El estudiante lee el problema.	4. El profesor pide los datos iniciales necesarios para la resolución del problema.

<p>5. Informa los datos extraídos del problema.</p> <ul style="list-style-type: none"> • Cantidad de platos del disco. • Cantidad de cilindros. • Cantidad de sectores por pistas. • Tamaño de cada sector. 	<p>6. Verifica los datos recibidos.</p>
	<p>7. Solicita al estudiante el tipo de parámetro que desea calcular:</p> <ul style="list-style-type: none"> • (a) Ver sección <i>Calcular Capacidad del disco.</i> • (b) Ver sección <i>Calcular Dirección Física.</i> • (c) Ver sección <i>Calcular Dirección Lógica.</i>

Sección Calcular Capacidad del disco.

<p>1. El estudiante solicita al profesor <i>Calcular Capacidad del Disco.</i></p>	<p>2. Solicita la unidad de medida en que desea realizar el cálculo de la capacidad final.</p>
<p>3. El estudiante informa la unidad de medida.</p>	<p>4. El profesor empleando la fórmula siguiente:</p> $\text{Num.Sectores} = \frac{\text{Numcaras} * \text{NumCilindros} * \text{Sectores}}{\text{Pista} * \text{TamSector.}}$ <p>Realiza el cálculo de la capacidad total del disco en papel o pizarra.</p>
	<p>5. Explica al estudiante el proceso de conversión y el resultado final obtenido.</p>
<p>6. Recibe la explicación.</p>	

Diagrama de Actividades para este Caso de Uso

Ver Anexo 11

Sección Calcular Dirección Física.

Acción del Actor	Respuesta del Negocio
<p>1. El estudiante solicita al profesor <i>Calcular Dirección Física.</i></p>	<p>2. Solicita la dirección lógica del sector correspondiente.</p>
<p>3. Informa la dirección lógica correspondiente.</p>	<p>4. Procede a calcular la dirección física siguiendo los siguientes pasos:</p> <ul style="list-style-type: none"> • Dividir la dirección lógica por el número de sectores por pista. • Dividir el número de caras completas

	obtenido en el punto anterior por el número de caras por cilindro que admite el disco.
	5. Obtiene la dirección física en el formato (Cilindro, cara, sector).
	6. Explica al estudiante el resultado y el procedimiento a seguir.
7. Recibe la explicación por parte del profesor.	
Diagrama de Actividades para este Caso de Uso Ver Anexo 12	
Sección Calcular Dirección Lógica	
Acción del Actor	Respuesta del Negocio
1. El estudiante solicita al profesor <i>Calcular Dirección Lógica</i> .	2. Solicita la dirección física del sector correspondiente.
3. Informa la dirección física correspondiente: <ul style="list-style-type: none"> • Cilindro. • Cara. • Sector. 	4. Valida el valor de cada componente informado.
	5. Procede a calcular la dirección lógica siguiendo la siguiente fórmula: $nb = k + \text{sec/pista} * (j + \text{caras/cilindro} * i)$
	6. Obtiene la dirección lógica equivalente.
	7. Explica al estudiante el resultado y el procedimiento a seguir.
8. Recibe la explicación por parte del profesor.	
Flujo alternativo	
Acción del Actor	Respuesta del Negocio
	4.1 El valor de algunos de los componentes es incorrecto. Regresa a 4.
Diagrama de Actividades para este Caso de Uso Ver Anexo 13	

Poscondiciones: Deben quedar calculados los parámetros del disco duro, como su capacidad, dirección lógica y física.

2.3.6 Modelo de Objetos del Negocio.

El Modelo de Objetos del Negocio es la representación de los trabajadores, las entidades y sus relaciones. Da una idea del manejo de las entidades generadas y muestra el diseño o los procesos desde un punto de vista estático.

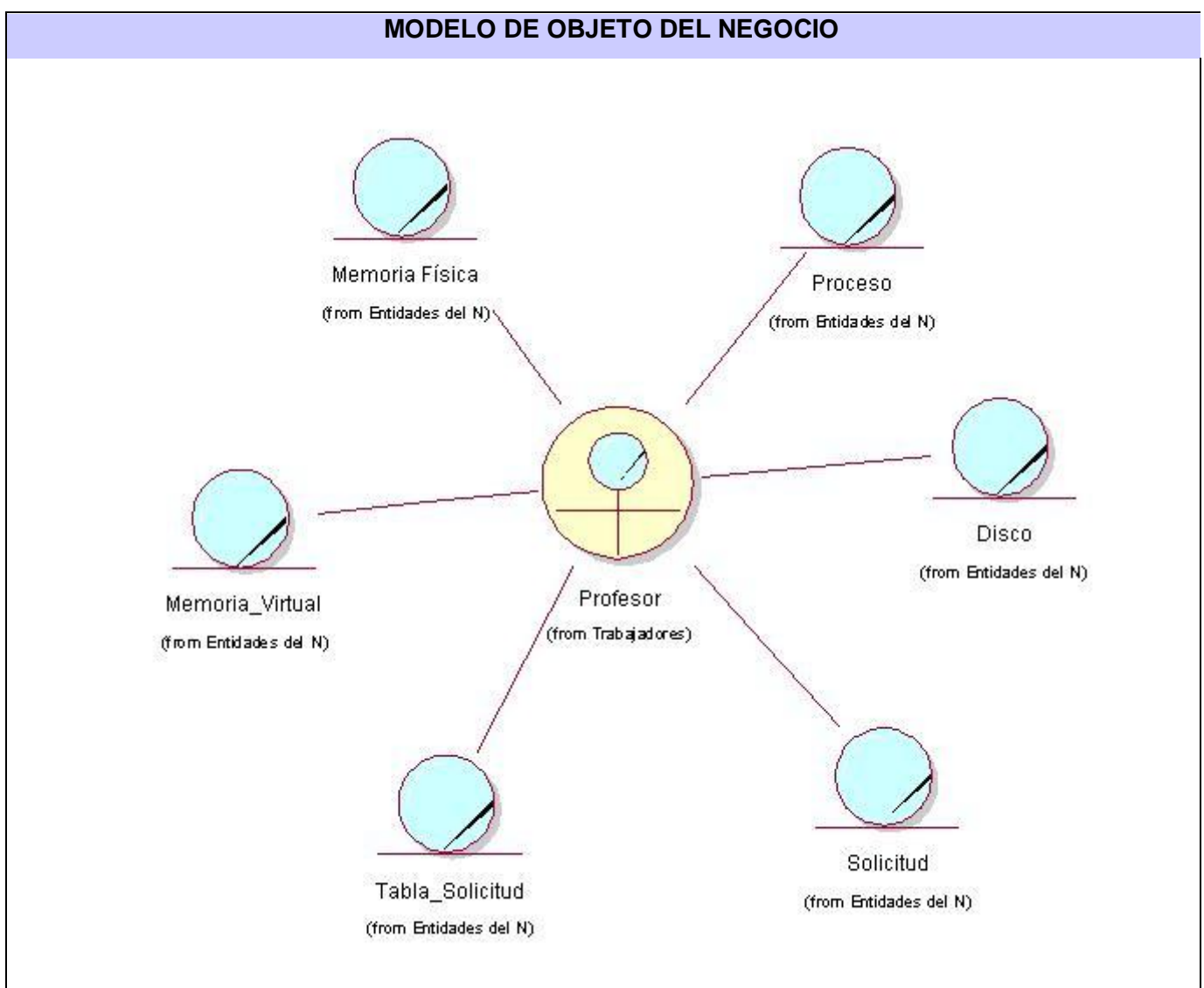


Figura 6 Modelo de Objeto del Negocio

2.4 Conclusiones del capítulo.

Se identificaron los Procesos de automatización, se definieron los Casos de Uso del Negocio, modelando en un diagrama sus relaciones con los Actores, lo que brinda una visión global de los procesos a analizar. Se identificaron los Actores y Trabajadores del Negocio, de modo que se sientan las bases para la definición posterior de los Usuarios que interactuarán con el Sistema. Se definieron las Reglas de Negocio, que se traducen en las condiciones que deberá cumplir el Sistema a modelar. A partir del Análisis hecho, se puede pasar a definir los Requisitos Funcionales y No Funcionales que deben guiar el Modelo de Sistema.

3 MODELACIÓN DEL SISTEMA

3.1 Introducción.

La captura de Requerimientos y el Modelado del Sistema son dos de los pasos más significativos de un proceso de desarrollo de Software. En este capítulo se procederá a la obtención de los artefactos correspondientes al Modelo del Sistema. Se obtienen en un primer momento el conjunto de requisitos funcionales y no funcionales que le darán vida al futuro sistema. A partir de este primer paso se obtendrá el diagrama de casos de uso del sistema como forma de agrupar los requerimientos y por ultimo se realizará el Diagrama de Clases del Análisis.

3.2 Especificación de los requerimientos de Software.

Una especificación de requisitos del software es una descripción completa del comportamiento del sistema a desarrollar. Incluye un conjunto de casos de uso que describen todas las interacciones que se prevén que los usuarios tendrán con el software. Los requisitos se pueden clasificar en: funcionales y no funcionales.

Las estrategias recomendadas para la especificación de los requisitos del software están descritas por IEEE 830-1998. Este estándar describe las estructuras posibles, contenido deseable, y calidades de una especificación de requisitos del software. (PAN, y otros, 2001)

Todas las ideas que los clientes, usuarios y miembros del equipo de proyecto tengan acerca de lo que debe hacer el sistema, deben ser analizadas como candidatas a requisitos. En los siguientes epígrafes se listan los requisitos funcionales y no funcionales identificados a partir de la modelación del negocio realizada.

3.2.1 Requerimientos Funcionales.

En la realización de los casos de uso del negocio, se obtienen las actividades que serán objeto de automatización. Estas actividades no son exactamente los requerimientos funcionales, pero si son el punto de partida para identificar qué debe hacer el sistema. Los requerimientos funcionales definen las

características que debe poseer el sistema. Los requisitos funcionales para este sistema se presentan a continuación:

Tabla de requisitos funcionales del sistema.

REFERENCIA	REQUISITO FUNCIONAL		
RF1	Inicializar memoria física.	ADMINISTRACIÓN DE MEMORIA	
RF2	Visualizar estado de la memoria física.		
RF3	Visualizar los procesos en memoria.		
RF4	Particionar memoria en tamaños fijos.		
RF5	Mostrar grado de multiprogramación.		
RF6	Asignar nuevo proceso a memoria.		
RF7	Visualizar Fragmentación interna.		
RF8	Liberar un proceso de la memoria.		
RF9	Visualizar colas de entrada de procesos.		
RF10	Compactar la memoria física.		
RF11	Administrar huecos libres.		
RF12	Unir huecos libres de memoria consecutivos.		
RF13	Calcular Fragmentación Externa de memoria.		
RF14	Paginar un nuevo proceso.		
RF15	Dividir memoria en marcos de igual longitud.		
RF16	Paginar un proceso en memoria.		
RF17	Generar la tabla de páginas de los procesos.		
RF18	Calcular dirección física de una página lógica.		
RF19	Configurar Memoria Virtual.		
RF20	Establecer nuevo proceso en la memoria virtual.		
RF21	Reemplazar página en memoria física.		
RF22	Visualizar tabla de reemplazo de páginas.		
RF23	Servir solicitud de acceso al disco duro.		
RF24	Calcular movimiento total del cabezal del disco duro.		
RF25	Calcular tiempo total de servicio de una solicitud.		
RF26	Visualizar movimiento del brazo del cabezal del disco.		

RF27	Calcular capacidad del disco duro	PLANIFICACIÓN DE DISCO
RF28	Calcular dirección lógica de un sector físico.	
RF29	Calcular dirección física de un sector lógico.	

3.2.2 Requerimientos no Funcionales.

Los requerimientos no funcionales son características que describen alguna forma o restricción para la realización de algún requerimiento (funcionalidad) o conjunto de ellas e inclusive todos los requerimientos. Se consideran propiedades o cualidades que debe tener el producto y que garantizan el soporte y buen funcionamiento de las acciones del sistema. (Jacobson, y otros, 2000)

En algunos casos los requisitos no funcionales se asocian a casos de uso específicos y en otros, al ser más generales e involucrar varios casos de uso pues quedan agrupados aparte.

A continuación se muestran los requerimientos no funcionales:

Apariencia o interfaz externa:

- **RNF1:** El diseño de la interfaz debe ser sencillo y fácil de usar con reconocimiento visual a través de elementos visibles que identifiquen cada una de sus acciones.
- **RNF2:** La combinación de colores debe ser agradable a la vista del usuario.
- **RNF3:** Debe contar con un vínculo a la ayuda en cada ventana de trabajo.

Usabilidad:

- **RNF4:** El sistema puede ser usado por cualquier estudiante o profesor que posea conocimientos básicos sobre el funcionamiento interno de un Sistemas Operativos.

Rendimiento:

- **RNF5:** La respuesta a solicitudes más complejas de los usuarios del sistema no debe exceder 9 segundos.

Portabilidad:

- **RNF6:** Debe poder ejecutarse tanto en Sistemas Operativos desde Windows XP Profesional Service Pack 1 o Superior como en Sistemas Operativos Linux.

Hardware:

- **RNF7:** Tarjeta de memoria RAM de 128 MB.
- **RNF8:** Procesador Pentium II o superior a 250 MHz como mínimo.

3.3 Actores del Sistema.

Los actores del sistema pueden representar el rol que juega una o varias personas, un equipo o un sistema automatizado, son parte del sistema, y pueden intercambiar información con él o ser recipientes pasivos de información. (Jacobson, y otros, 2000)

En este caso los actores que interactúan con el sistema se definen en la siguiente tabla.

Actores del Sistema	Justificación
Usuario	Actor base que representa cualquier persona que hace uso del sistema para resolver un ejercicio relacionado con <i>Administración de Memoria</i> y <i>Planificación de Disco</i> .
Estudiante	Se considera en este rol a cualquier estudiante que al tener dudas en la resolución de algún ejercicio de <i>Administración de Memoria</i> o <i>Planificación de Disco</i> hace uso del sistema como herramienta de apoyo.
Profesor	Se encarga de usar el sistema como herramienta de apoyo en el aula para facilitar la comprensión por parte de los estudiantes al explicar los temas relacionados con <i>Administración de Memoria</i> y <i>Planificación de Disco</i> .

3.4 Diagrama de Casos de Uso del Sistema.

El modelado de casos de uso es la técnica más efectiva para modelar los requisitos del sistema. Los casos de uso se utilizan para modelar el funcionamiento que desea el cliente que tenga el sistema.

A continuación se muestra el diagrama de casos de uso del sistema.

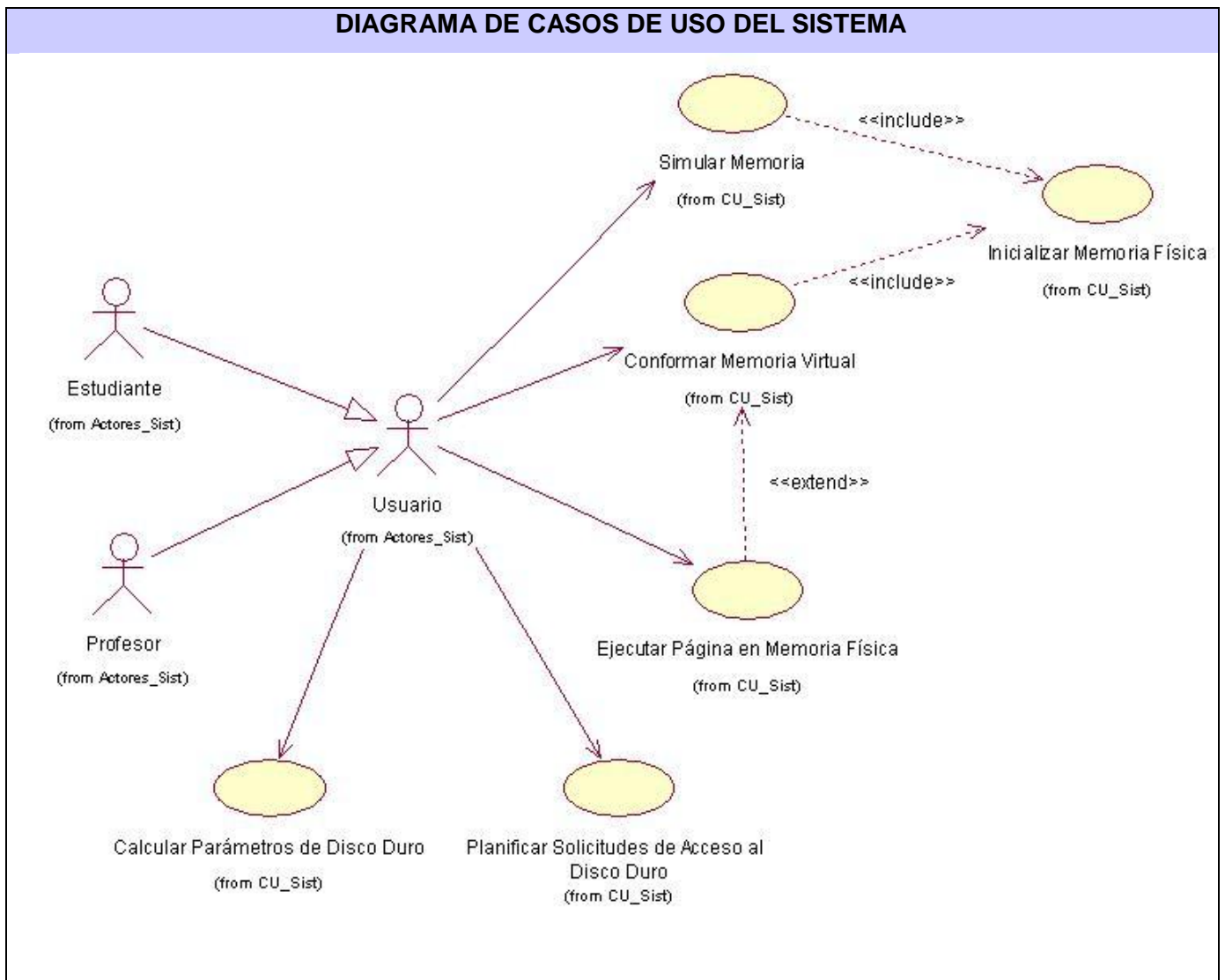


Figura 7 Diagrama de CU del Sistema.

3.5 Descripción de los Casos de Uso del Sistema.

Descripción del Caso de uso: **Inicializar Memoria Física.**

Caso de Uso:	Inicializar Memoria Física (Incluido)
Código	CU_1.
Actor	CU_2, CU_3

Propósito	Definir tamaño total de la memoria física y cantidad reservada para el Sistemas Operativos.
Resumen:	Caso de uso incluido que se inicia cuando es llamado por otro caso de uso para inicializar la memoria como primer paso para simular cualquier operación con la memoria y culmina con la representación grafica de la memoria física y el espacio libre para procesos de usuario.
CU asociados	
Referencias	RF1, RF2, RF3
Precondiciones	Tamaño total de la memoria y cantidad destinada al Sistemas Operativos.
Poscondiciones	La memoria física queda inicializada y el espacio destinado a los procesos de usuario queda definido.
Prioridad	Crítico

Flujo Normal de Eventos

Acción del Actor	Respuesta del Sistema
1. Da clic en el botón <i>Inicializar Memoria</i> de la interfaz <i>Inicializar Memoria Física</i> .	2. El sistema muestra la interfaz con los siguientes datos a introducir: <ul style="list-style-type: none"> • Capacidad total de la memoria a establecer. • Unidad de medida a utilizar (KB). • Capacidad de memoria destinada al Sistemas Operativos.
3. Introduce los datos solicitados y da clic en el botón <i>Inicializar</i> .	4. Comprueba que la capacidad suministrada para el Sistemas Operativos no exceda el total de memoria proporcionada. 5. Conformamente automáticamente la memoria y si distribución inicial.
	6. Visualiza la interfaz con la memoria graficada y su distribución actual acorde a los datos suministrados.

Flujos Alternos

Acción del Actor	Respuesta del Sistema
	4.1 Si no es válido el contenido de los datos, se

muestra nuevamente la interfaz de inicialización con los campos errados marcados para corrección.

Prototipo no Funcional de Interfaz de usuario



Descripción del Caso de uso: **Simular Memoria.**

Caso de Uso	Simular Memoria.
Código	CU_2
Actores	Usuario
Propósito	El presente caso de uso se encargará de la simulación de la memoria física para un entorno monoprogramado o multiprogramado en presencia de particiones fijas y variables.
Resumen:	El caso de uso se inicia cuando el actor del sistema selecciona la opción de simulación de procesos en memoria y finaliza con la visualización correcta del estado de la memoria, así como sus particiones y procesos en ella luego de la simulación.
CU asociados	Inicializar Memoria Física.
Referencias	RF4, RF5, RF6, RF7, RF8, RF9, RF10, RF11, RF12, RF13
Precondiciones	La memoria física debe estar inicializada
Poscondiciones	La memoria física queda estructurada con todas las particiones y procesos

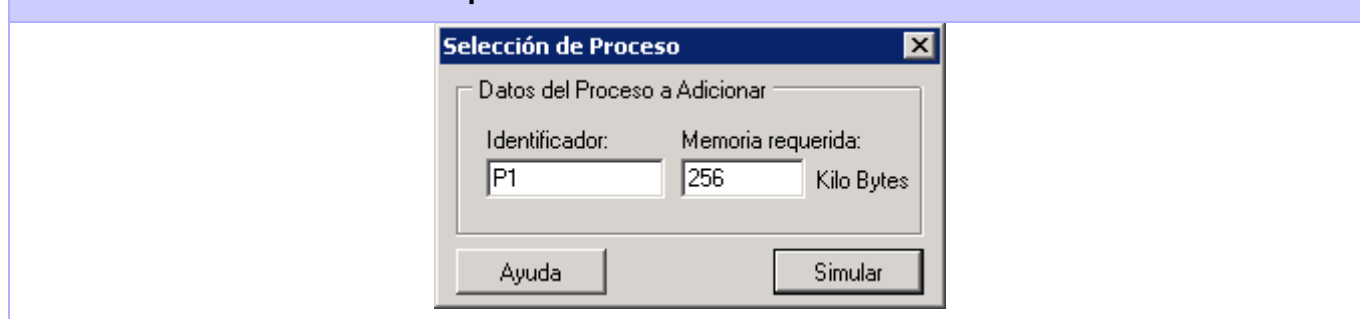
	definidos durante todo el proceso de simulación. Representación gráfica del esquema de memoria con particiones variables.
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. En el menú <i>Simular Memoria</i> selecciona la opción <i>Simular memoria física</i> .	2. Se invoca el caso de uso incluido <i>Inicializar Memoria Física</i> .
	3. El sistema muestra la interfaz <i>Simular Memoria Física</i> , con las siguientes opciones: <ul style="list-style-type: none"> a) Simular Memoria en Monoprogramación. b) Simular Memoria en Multiprogramación con Particiones Fijas. c) Simular Memoria en Multiprogramación con Particiones Variables.
	4. En caso de seleccionar la opción: <ul style="list-style-type: none"> a) Ir a la sección “<i>Simular Memoria en Monoprogramación</i>”; b) Ir a la sección “<i>Simular Memoria en Multiprogramación con Particiones Fijas</i>”. c) Ir a la sección “<i>Simular Memoria en Multiprogramación con Particiones Variables</i>”.
Sección “Simular Memoria en Monoprogramación”	
Acción del Actor	Respuesta del Sistema
1. Selecciona la opción <i>Simular Memoria en Monoprogramación</i>	2. Muestra la interfaz <i>Selección de Proceso</i> que solicita la introducción de los siguientes datos. <ul style="list-style-type: none"> • Identificar el nuevo proceso a entrar en memoria. • Memoria requerida por el nuevo proceso
3. Introduce los datos solicitados.	4. El sistema comprueba que la capacidad de memoria requerida por el proceso, no exceda el total existente de memoria física para procesos

	de usuario.
	5. Crea un nuevo proceso y lo adiciona al área de memoria destinada a procesos de usuario.
	6. Se actualiza gráficamente la memoria en la interfaz “ <i>Memoria Principal</i> ”, reflejándose así la nueva estructura.
	7. Se actualiza en la interfaz “ <i>Memoria Principal</i> ”, el valor correspondiente a la fragmentación interna.

Flujos Alternos

Acción del Actor	Respuesta del Sistema
	4.1 Si los datos no son validos se muestra nuevamente la interfaz <i>Selección de Proceso</i> con los campos errados marcados para ser corregidos.

Prototipo no Funcional de Interfaz de usuario



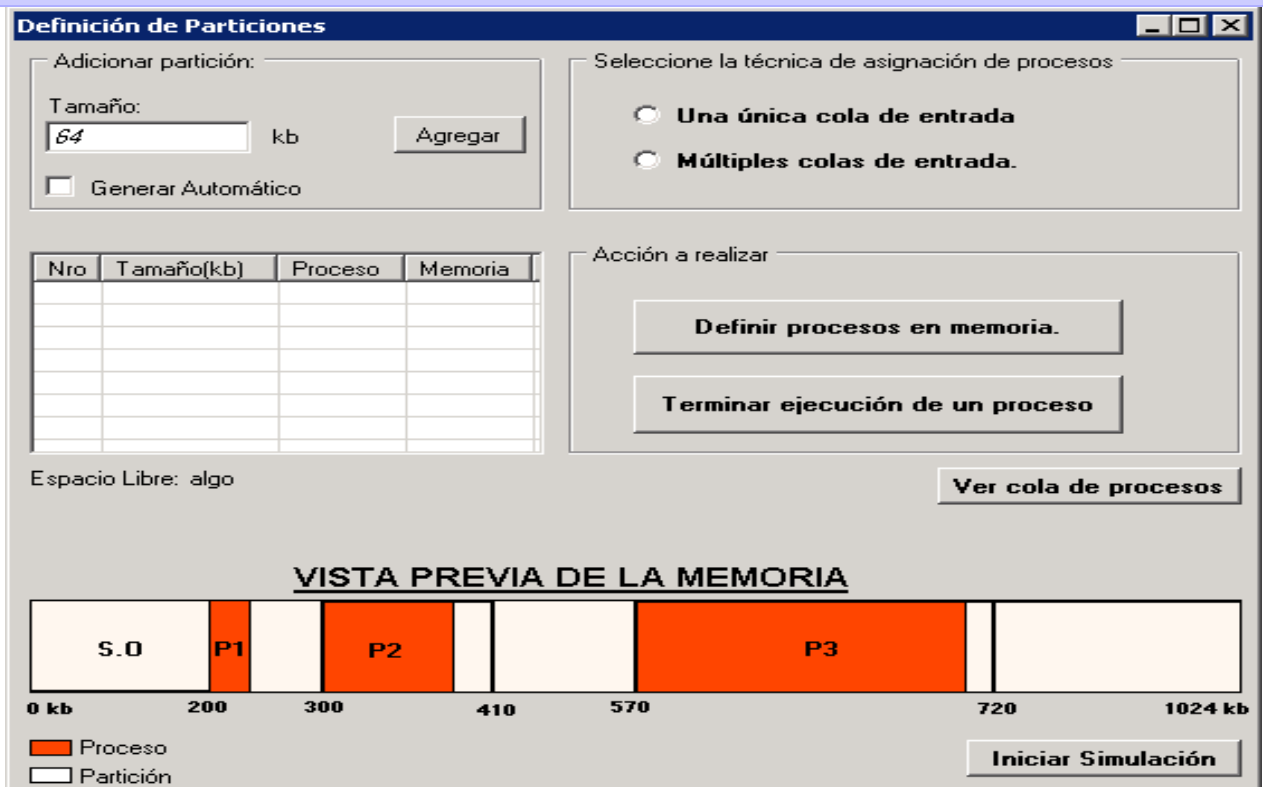
Sección “Simular Memoria en Multiprogramación con Particiones Fijas”

Acción del Actor	Respuesta del Sistema
1. Selecciona la opción “ <i>Simular Memoria en Multiprogramación con Particiones Fijas</i> ”.	2. Muestra la interfaz “ <i>Definición de Particiones</i> ” solicitando la introducción de los siguientes datos: <ul style="list-style-type: none"> • Id de partición. • Tamaño de partición.
3. Introduce los datos solicitados para cada partición deseada y presiona el botón <i>Agregar</i> .	4. Valida los tamaños suministrados para cada partición respecto a la capacidad que va

	quedando disponible en la memoria total.
	5. Actualiza en la interfaz el campo <i>Grado de multiprogramación</i> .
	6. Se habilita en la interfaz la opción " <i>Seleccione la técnica de asignación de procesos</i> " <ul style="list-style-type: none"> • Una única cola de entrada. • Múltiples colas de entrada.
7. Selecciona la técnica a utilizar.	8. Se habilita la opción " <i>Acción a realizar</i> ": <ul style="list-style-type: none"> • Definir procesos en memoria. • Terminar ejecución de un proceso.
9. Selecciona la opción <i>Definir procesos en memoria</i> .	10. Muestra la interfaz "Definición de Proceso" con los siguientes datos solicitados: <ul style="list-style-type: none"> • Identificador del proceso. • Cantidad de memoria requerida.
11. Define los datos necesarios para cada proceso y presiona para cada uno el botón " <i>Agregar</i> ".	12. Valida los datos proporcionados para cada proceso y lo asigna a la cola de procesos correspondiente.
13. Presiona el botón " <i>Iniciar Simulación</i> "	14. El sistema muestra la interfaz " <i>Memoria Principal</i> " y paso a paso va seleccionando procesos y adicionándolos a las particiones creadas según la política definida.
	15. Se actualiza la interfaz " <i>Memoria principal</i> " con todos los procesos con sus respectivas particiones, el estado de la fragmentación interna y la cola de procesos en espera.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	4.1 Si no es válido el contenido de los datos, se vuelve a mostrar la interfaz con los campos errados marcados para corrección.
8.1 Selecciona la opción " <i>Terminar ejecución de un proceso</i> ."	8.2 Se muestra la tabla con todos los procesos existentes en memoria.

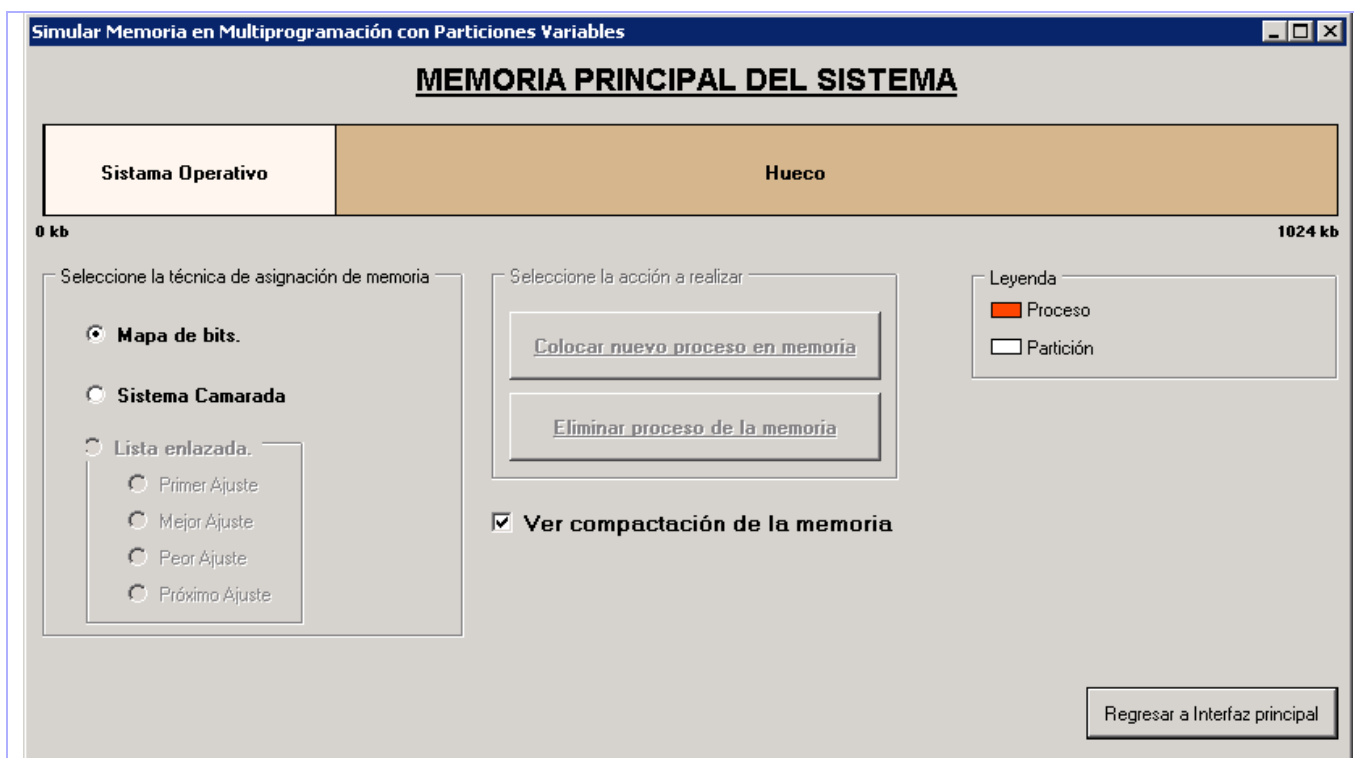
8.3 Selecciona el proceso en la tabla y presiona el botón “ <i>Liberar Proceso</i> ”	8.4 Muestra la interfaz “ <i>Memoria Principal</i> ” y actualiza la memoria mostrando la partición libre.
	8.5 Si existen procesos en la cola de entrada selecciona un nuevo proceso a ejecutarse siguiendo los siguientes criterios: <ul style="list-style-type: none"> • El proceso debe “caber” en la partición liberada. • Debe tomarse de todos los procesos que cumplan con el criterio “aquel que lleve más tiempo en espera”.
	12.1 Si no es válido el contenido de los datos, se vuelve a mostrar la interfaz con los campos errados y marcados para corrección.

Prototipo no Funcional de Interfaz de usuario



Sección “Simular Memoria en Multiprogramación con Particiones Variables”	
Flujo Normal de eventos	
Acción del Actor	Respuesta del Sistema
1. Selecciona la opción “ <i>Simular Memoria en Multiprogramación con Particiones Variables</i> ”.	2. Se muestra la interfaz “ <i>Multiprogramación con Particiones Variables</i> ” con la representación gráfica de la memoria Principal con una partición reservada para el Sistemas Operativos y el resto como un hueco libre.
	3. Se activa la opción “ <i>Seleccione la técnica de asignación de memoria</i> ”: <ul style="list-style-type: none"> • Mapa de bits. • Lista Enlazadas: <ul style="list-style-type: none"> ✓ Primer acceso. ✓ Mejor Acceso. ✓ Próximo acceso. ✓ Peor Acceso. • Sistema Camarada
4. Selecciona una de las técnicas de asignación.	5. Se habilita la opción “ <i>Seleccione la acción a realizar</i> ”: <ul style="list-style-type: none"> a) Colocar nuevo proceso en memoria. b) Eliminar proceso de la memoria.
6. Selecciona la opción deseada.	7. En caso de seleccionar la opción: <ul style="list-style-type: none"> a) Ir a la sección “Colocar nuevo proceso en memoria”; b) Ir a la sección “Eliminar un proceso de la memoria”.
Sección “Colocar nuevo proceso en memoria”	
Acción del Actor	Respuesta del Sistema
1. Selecciona la opción “ <i>Colocar nuevo proceso en memoria</i> ”	2. Muestra la interfaz “ <i>Definición de Proceso</i> ” que solicita la introducción de los siguientes datos: <ul style="list-style-type: none"> • Identificador del nuevo proceso.

	<ul style="list-style-type: none"> • Capacidad de memoria requerida.
3. Introduce los datos solicitados para cada proceso presionando el botón <i>Agregar</i>	4. Verifica que la memoria requerida por el nuevo proceso coincide o es menor que el total de memoria disponible.
	5. Automáticamente según la técnica seleccionada procede a insertar el proceso en la memoria, creando una partición de igual tamaño que el requerido por el proceso.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	4.1 Si los datos no son validos se vuelve a mostrar la interfaz de creación de contenidos con los campos errados marcados para ser arreglados.
	5.1 Sino no existe un hueco lo suficientemente grande para insertar el proceso se procede a <i>compactar la memoria</i> .
	5.2 El sistema mueve todos los procesos hacia un extremo de la memoria hasta que se forma un hueco lo suficientemente grande.
	5.3 Cada paso de la compactación es mostrado al usuario mediante la interfaz "Compactado de Memoria".
Prototipo no Funcional de Interfaz de usuario	



Sección “Eliminar proceso de la memoria”

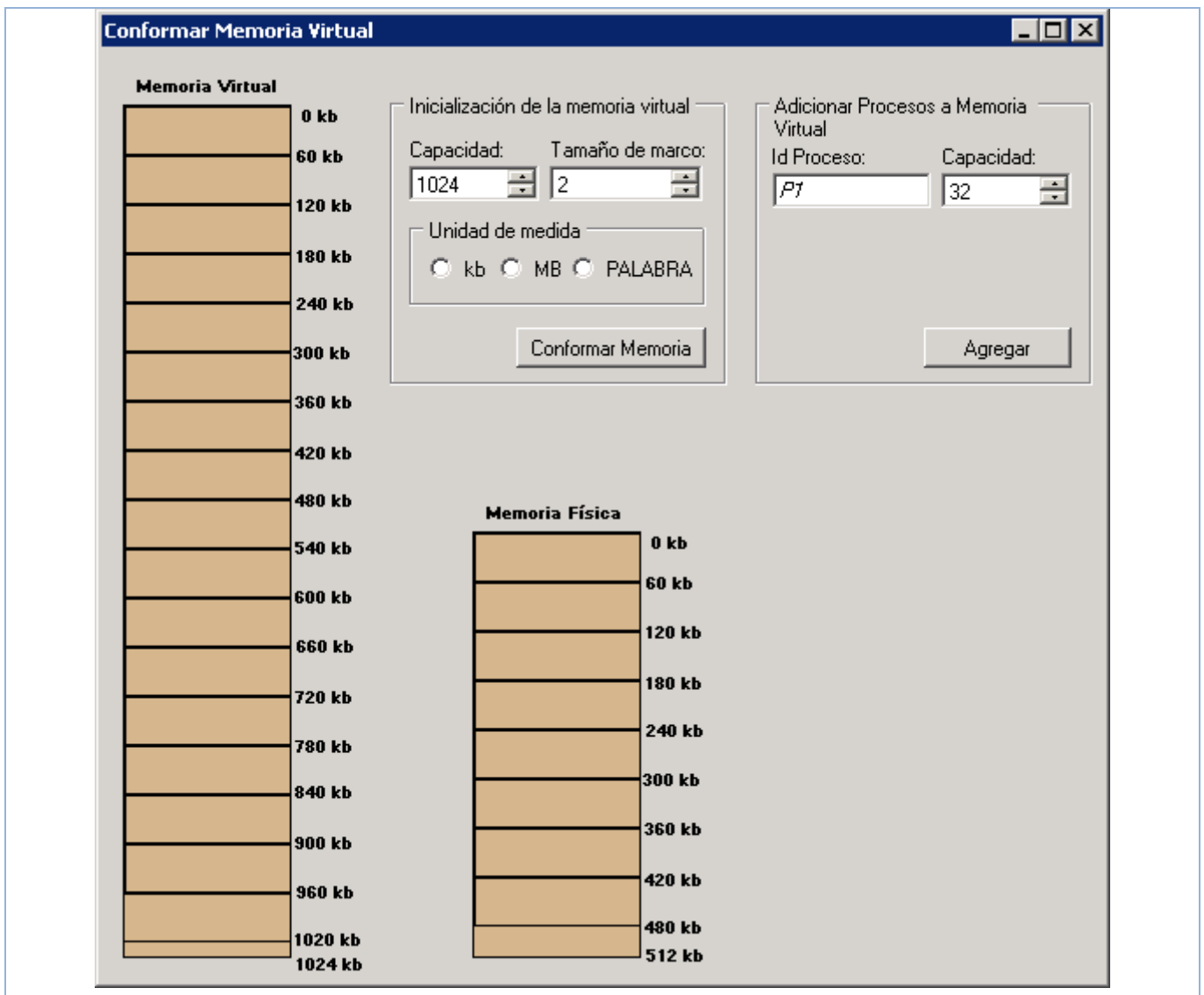
Acción del Actor	Respuesta del Sistema
1. Selecciona la opción “ <i>Eliminar proceso de la memoria</i> ”	2. Se muestra la interfaz “ <i>Lista de Procesos en Ejecución</i> ” con la lista de todos los procesos en ejecución.
3. Selecciona el identificador de un proceso a eliminar.	4. Termina el proceso seleccionado eliminándolo de la lista de procesos en ejecución y liberando la partición ocupada por este.
	5. Verifica si existe un hueco adyacente a la partición liberada. Se unen los huecos adyacentes y forman uno solo de mayor tamaño.
	6. Verifica si existe algún proceso en la cola de entrada.
	7. Adiciona un proceso existente en la cola de entrada a un hueco disponible.
	8. Se actualiza la interfaz “ <i>Memoria Principal</i> ” con todos los procesos con sus respectivas

	particiones, el estado de la fragmentación externa y la cola de procesos en espera.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	2.1 Si no existe ningún proceso en ejecución muestra la interfaz “Mensaje” informando que no se puede terminar.

Descripción del Caso de uso: “Conformar Memoria Virtual”

Caso de Uso	Conformar Memoria Virtual.
Código	CU_5
Actores	Usuario
Propósito	Crear el esquema de memoria virtual del sistema.
Resumen:	Se inicia cuando el estudiante selecciona la opción simular memoria virtual, se introducen los datos solicitados, se divide en páginas o en marcos de páginas y termina con la visualización de un proceso en memoria.
CU asociados	Inicializar Memoria Física (<Include>). Reemplazar Página (<extend>).
Referencias	RF14, RF15, RF16, RF17, RF18, RF19, RF20.
Precondiciones	Cantidad de memoria a utilizar y el tamaño de cada marco de página.
Poscondiciones	Tanto la memoria virtual como la física quedan conformadas con sus distribuciones en marcos virtuales.
Prioridad	Crítico
Curso Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. Selecciona la opción “Conformar Memoria Virtual”.	2. Muestra la interfaz “Conformar Memoria Virtual” y solicita la introducción de los siguientes datos: <ul style="list-style-type: none"> • Capacidad a establecer para la memoria virtual. • Unidad de medida a utilizar (KB). • Tamaño de marco virtual.

3. Introduce los datos solicitados y presiona el botón “Conformar Memoria”.	4. El sistema comprueba la veracidad de los datos, y muestra la interfaz “Conformando Memoria Virtual”, mostrando paso a paso y gráficamente la generación de los marcos tanto físicos como virtuales.
	5. Muestra nuevamente la interfaz “Conformar Memoria” y se activa la opción “Adicionar Procesos a Memoria” solicitando la introducción de los siguientes datos: <ul style="list-style-type: none"> • Identificador del nuevo proceso. • Capacidad de memoria requerida.
6. Introduce los datos solicitados para cada proceso y presiona el botón “Agregar”.	7. El sistema comprueba la veracidad de los datos y procede a dividir cada proceso en páginas de igual tamaño que los marcos.
	8. El sistema muestra gráficamente la colocación de cada página creada en algún marco virtual vacío.
Flujo Alternativo	
Acción del Actor	Respuesta del Sistema
	4.1 Si los datos no son válidos se muestran los campos errados marcados en rojo para ser arreglados.
	7.1 Si los datos no son correctos se muestran los campos errados marcados en rojo para ser arreglados.
Prototipo no Funcional de Interfaz de usuario	



Descripción del Caso de uso: **“Ejecutar Página en Memoria Física”**

Caso de Uso	Ejecutar Página en Memoria Física
Código	CU_6
Actores	Usuario
Propósito	Realizar el reemplazo de una página en memoria física.
Resumen:	Se inicia cuando un proceso que se encuentra en memoria virtual necesita ejecutar alguna de sus páginas en memoria física y esta no posee marcos

	libres, para ello se utilizan diferentes algoritmos de reemplazo y termina cuando la página es reemplazada.
CU asociados	
Referencias	RF21, RF22.
Precondiciones	Existencia de procesos en memoria virtual. La memoria física no debe tener marcos libres.
Poscondiciones	La página correspondiente es reemplazada y la memoria tanto virtual como física es actualizada.
Prioridad	Crítico
Curso Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. Selecciona la opción “ <i>Sacar Páginas a Memoria</i> ” en la interfaz “ <i>Conformar Memoria Virtual</i> ”.	2. Se activan la opciones: <ul style="list-style-type: none"> • “<i>Seleccione el algoritmo de Reemplazo</i>”. <ul style="list-style-type: none"> • OPT (Algoritmo óptimo). • FIFO (First Come First Out). • LRU (Least Recently Used). • LFU (Least Frequently Used). • MFU (Most Frequently Used). • NRU (Not Recently Used). • “<i>Tipo de Reemplazo</i>”: <ul style="list-style-type: none"> • <i>Reemplazo Local</i>. • <i>Reemplazo Global</i>.
3. Selecciona un algoritmo de reemplazo y el tipo correspondiente.	4. Se activa la opción “ <i>Introduzca las páginas a ejecutar</i> ”.
5. Introduce cada página a ejecutar y demás datos según requerimientos del algoritmo de reemplazo seleccionado. Presionando para cada página el botón “ <i>Agregar a la lista</i> ”.	6. Valida los datos de cada página introducida.
7. Presiona el botón “ <i>Simular Salida de Páginas</i> ”.	8. Muestra gráficamente paso por paso el proceso de salida de las páginas hacia la memoria física.
	9. Calcula el total de reemplazos y fallos realizados y

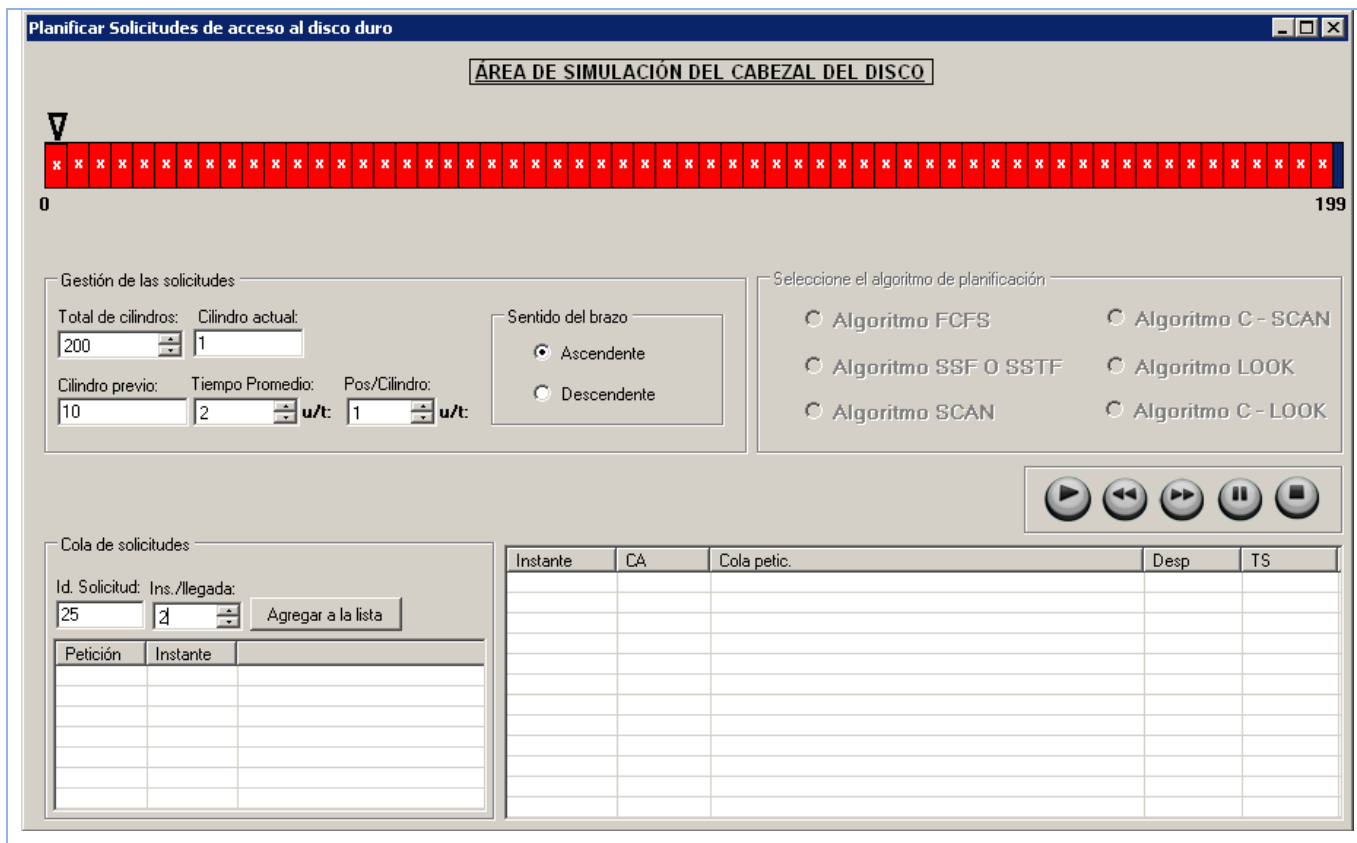
Descripción del Caso de uso: **“Planificar solicitudes de acceso al Disco Duro.”**

Caso de Uso	Planificar solicitudes de acceso al Disco Duro.
Código	CU_7
Actores	Usuario
Propósito	Simular Planificación de las solicitudes de acceso a las pistas que se producen en un disco duro.
Resumen:	Se inicia cuando el usuario realiza un ejercicio de Planificación de Disco y necesita comprobar la respuesta o comprender el funcionamiento de un algoritmo de planificación al servir una lista de solicitudes. El caso de uso finaliza con la lista de solicitudes servidas, la visualización gráfica del proceso y el cálculo del tiempo de servicio y el movimiento total del cabezal.
CU asociados	
Referencias	RF23, RF24, RF25, RF26.
Precondiciones	Lista de solicitudes a servir y algoritmos de planificación
Poscondiciones	Lista de solicitudes servidas, visualización gráfica del proceso, tiempo de servicio y el movimiento total del cabezal.
Prioridad	Crítico

Curso Normal de Eventos

Acción del Actor	Respuesta del Sistema
1. En la interfaz principal <i>“Simulador de Disco y Memoria”</i> , selecciona la opción <i>“Planificar Solicitudes”</i> .	2. Muestra la interfaz <i>“Planificar Solicitudes”</i> que solicita la introducción de los siguientes datos: <ul style="list-style-type: none"> • Cantidad de cilindros del disco. • Cilindro servido actualmente. • Cilindro servido previamente. • Sentido del movimiento del brazo. • Tiempo promedio de acceso (rotación – latencia + transferencia). • Tiempo de posicionamiento entre cilindros contiguos. • Cola de solicitudes. <ul style="list-style-type: none"> ○ Id del próximo cilindro a servir.

	<ul style="list-style-type: none"> ○ Instante de llegada de la solicitud.
3. Introduce los datos necesarios y para cada solicitud presiona el botón “Adicionar a la lista”.	4. El sistema comprueba la veracidad de los datos. Y se activa la opción “ <i>Seleccione el algoritmo de planificación</i> ”. Con las siguientes opciones: <ul style="list-style-type: none"> • FCFS (“First Come First Served”) • SSF o SSTF (“Shortest Seek-Time First”) • SCAN • C-SCAN • LOOK • C-LOOK
5. Selecciona el algoritmo y oprime el botón “Comenzar simulación”.	6. Muestra gráficamente paso a paso el movimiento del cabezal para cada solicitud servida.
	7. Calcula el valor correspondiente a: <ul style="list-style-type: none"> • Desplazamiento total realizado por el cabezal. • Tiempo total de servicio. • Tiempo para cada solicitud.
	8. Muestra en una tabla los valores obtenidos anteriormente.
Flujo Alterno	
Acción del Actor	Respuesta del Sistema
	4.1 Si los datos no son validos se marcan en rojo los campos errados para ser arreglados.
Prototipo no Funcional de Interfaz de usuario	



Descripción del Caso de uso: **“Calcular parámetros de Disco Duro”**

Caso de Uso	Calcular parámetros de Disco Duro
Código	CU_8
Actores	Usuario
Propósito	Calcular capacidad del disco duro, así como convertir sectores físicos a lógicos y viceversa.
Resumen:	Se inicia cuando el usuario necesita comprobar su respuesta en la realización de un ejercicio de cálculo de parámetros de disco y hace uso de la herramienta para chequear automáticamente el resultado. El sistema realiza los cálculos correspondientes y muestra al usuario los resultados, finalizando así el caso de uso.
CU asociados	
Referencias	RF27, RF28, RF29.

Precondiciones	Total de cilindros y sectores del disco duro así como la capacidad del sector.
Poscondiciones	Parámetros de disco calculados.
Prioridad	Crítico
Curso Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. En la interfaz principal “ <i>Simulador de Memoria y Disco</i> ” selecciona el menú “ <i>Calcular parámetros de disco</i> ”.	2. Muestra la interfaz “ <i>Calcular parámetros de disco</i> ” solicitando los siguientes datos: <ul style="list-style-type: none"> • Cantidad de platos del disco. • Cantidad de cilindros. • Cantidad de sectores por pistas. • Tamaño de cada sector.
3. Introduce los datos y presiona el botón “ <i>Inicializar</i> ”	4. Activa la opción “ <i>Seleccione tipo de parámetro a calcular</i> ” con las siguientes opciones: <ul style="list-style-type: none"> (a) Ver sección “<i>Calcular Capacidad del disco</i>”. • (b) Ver sección <i>Calcular Dirección Física</i>. • (c) Ver sección <i>Calcular Dirección Lógica</i>.

Prototipo no Funcional de Interfaz de usuario

Sección “Calcular Capacidad del disco”

Curso normal de los eventos

Acción del Actor	Respuesta del Sistema
1. Selecciona la opción “Calcular capacidad de disco”	2. Muestra la interfaz “Calcular Capacidad de disco” solicitando el siguiente dato: <ul style="list-style-type: none"> • Unidad de medida en que desea realizar el cálculo de la capacidad final. <ul style="list-style-type: none"> ○ Byte. ○ KB. ○ MB ○ GB.
3. Selecciona la unidad de medida y presiona el botón “Realizar Cálculo Final”.	4. El sistema automáticamente haciendo uso de la fórmula: <ul style="list-style-type: none"> • $CD = \text{Numcaras} * \text{NumCilindros} * \text{Sectores} / \text{Pista} * \text{TamSector}$. Calcula la Capacidad del Disco.
	5. Visualiza en pantalla el resultado.

Prototipo no Funcional de Interfaz de usuario



Sección “Calcular Dirección Física”

Curso normal de los eventos

Acción del Actor	Respuesta del Sistema
1. Selecciona la opción “Calcular Dirección Física”:	2. Muestra la interfaz “Calcular Dirección Física” solicitando el siguiente dato: <ul style="list-style-type: none"> • Dirección Lógica del sector.

2. Introduce la dirección lógica correspondiente.	4. Calcula automáticamente la dirección física siguiendo los siguientes pasos: <ul style="list-style-type: none"> • Dividir la dirección lógica por el número de sectores por pista. • Dividir el número de caras completas obtenido en el punto anterior por el número de caras por plato que admite el disco.
	5. Conformar la dirección física en el formato (Cilindro, cara, sector).
	6. Actualiza la interfaz con el valor obtenido.

Flujo Alternativo

Acción del Actor	Respuesta del Sistema
	5.1. Si la dirección física está fuera del rango real del disco se muestra un mensaje de error al usuario.

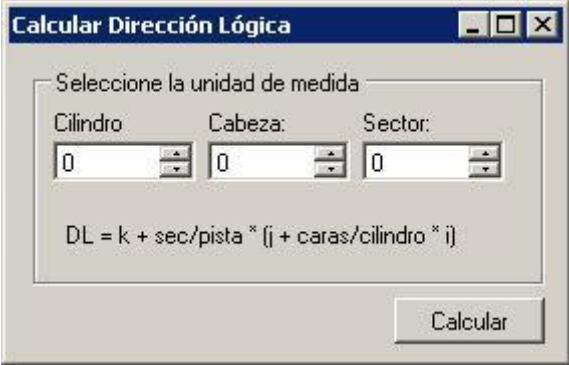
Prototipo no Funcional de Interfaz de usuario



Sección "Calcular Dirección Lógica"

Curso normal de los eventos

Acción del Actor	Respuesta del Sistema
1. Selecciona la opción "Calcular Dirección Lógica":	2. Muestra la interfaz "Calcular Dirección Lógica" solicitando el siguiente dato: <ul style="list-style-type: none"> • Dirección Física del sector en formato (i, j, k). <ul style="list-style-type: none"> ○ Cilindro. ○ Cara. ○ Sector físico.
3. Introduce la dirección física solicitada y presiona el botón "Calcular".	4. Valida la dirección proporcionada.

	5. Calcula automáticamente la dirección lógica con la siguiente fórmula: <ul style="list-style-type: none"> • $DL = k + sec/pista * (j + caras/cilindro * i)$
	6. Actualiza la interfaz con el valor obtenido.
Flujo alterno	
Acción del Actor	Respuesta del Sistema
	4.1 Muestra un mensaje al usuario indicando que la dirección está fuera de rango.
Prototipo no Funcional de Interfaz de usuario	
	

Esquema de interacción del prototipo no funcional.

A partir de la descripción de los casos de uso del sistema y la propuesta no funcional de las interfaces de usuario, se considera necesario realizar un esquema que refleje de manera resumida y explícita las formas de interacción entre las interfaces que conformarán la herramienta una vez implementada.

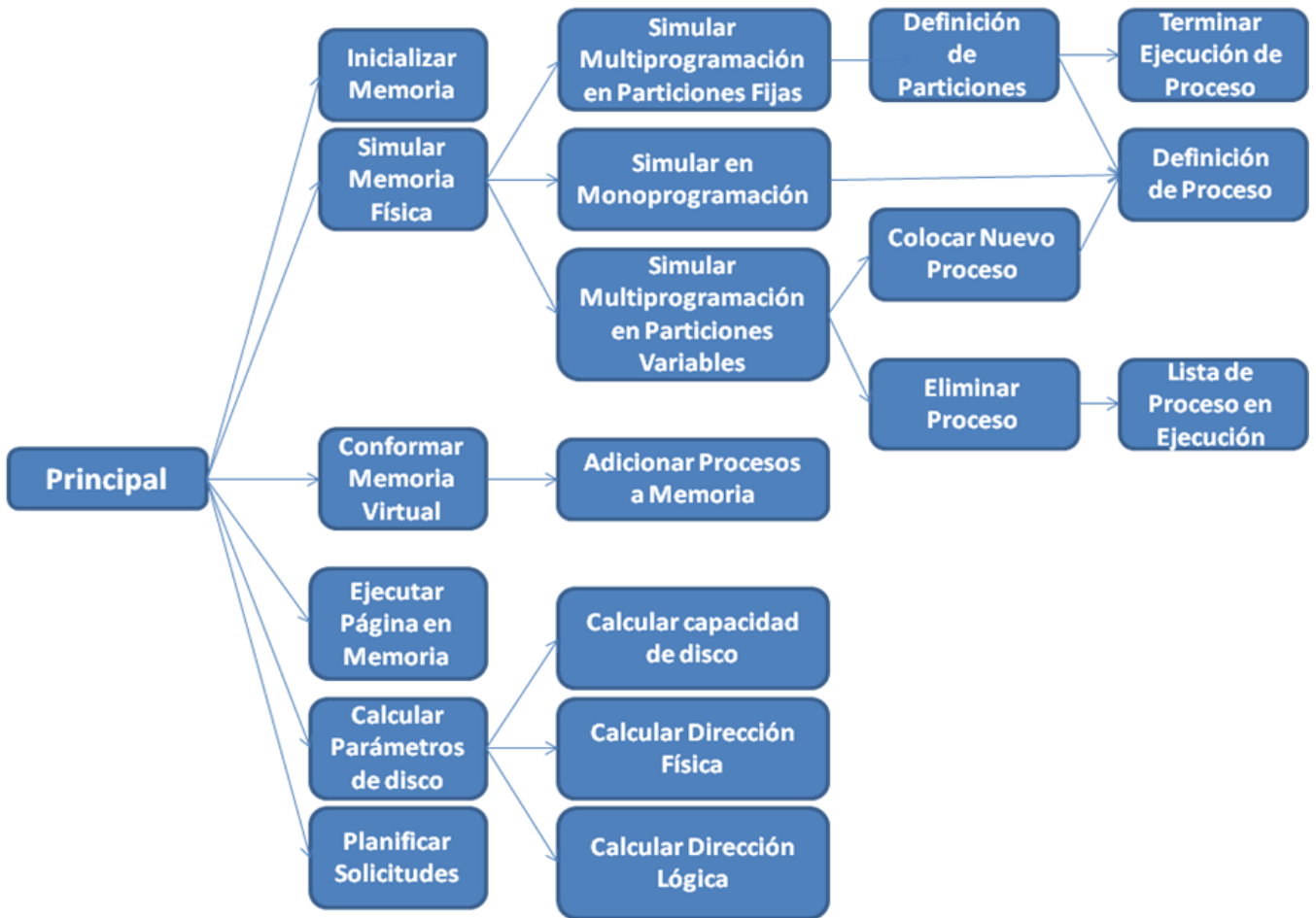


Figura 8 Esquema de Interacción del Prototipo No Funcional.

3.6 Diagramas de Clases del Análisis.

A continuación se muestran los diagramas de clases de análisis de algunos de los casos de usos.

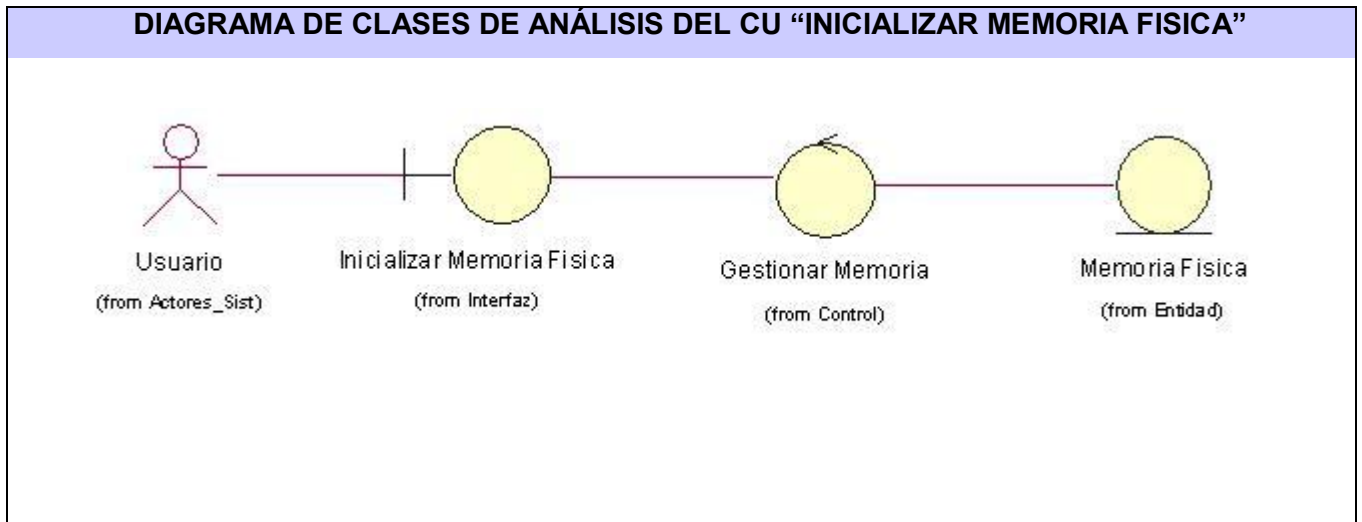


Figura 9 Diagrama de Clases del Análisis del CU Inicializar Memoria Física

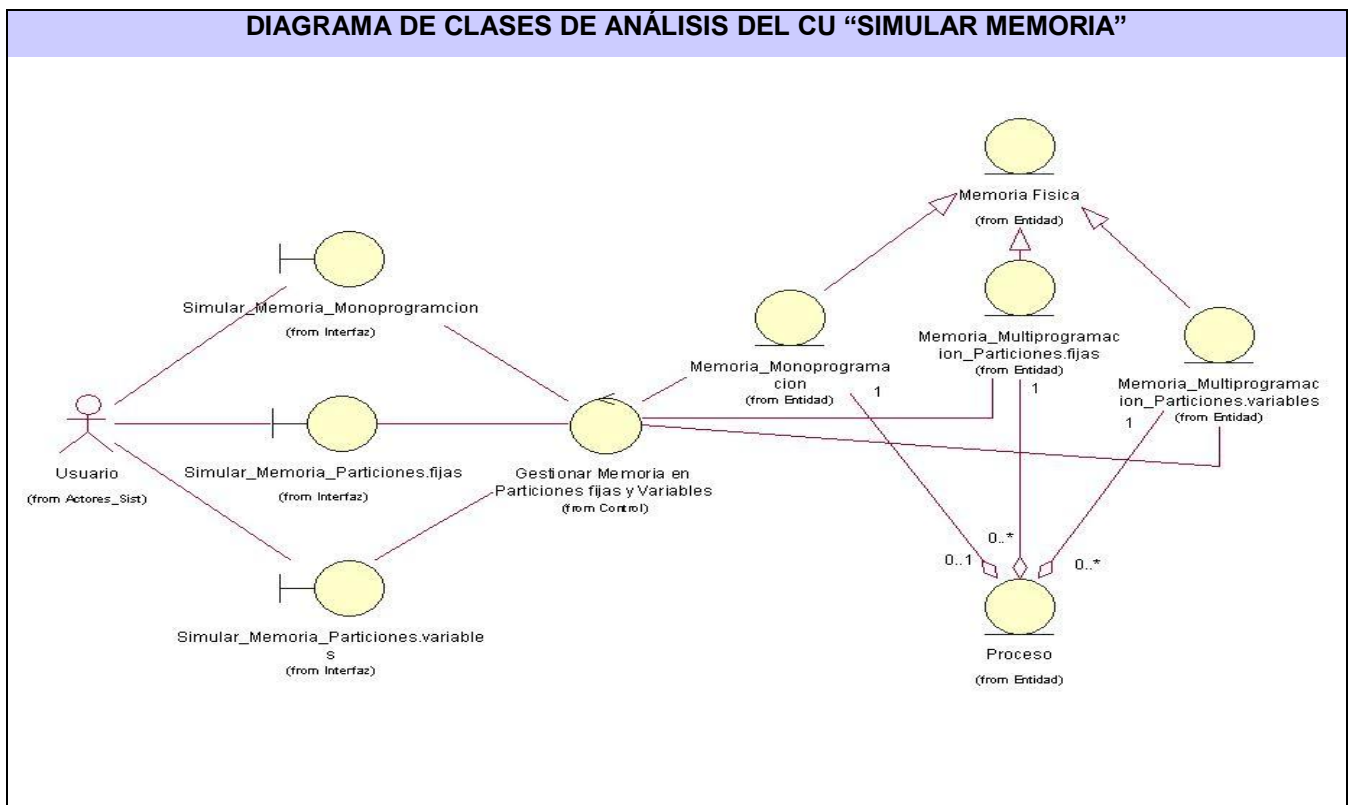


Figura 10 Diagrama de Clases del Análisis del CU Simular Memoria

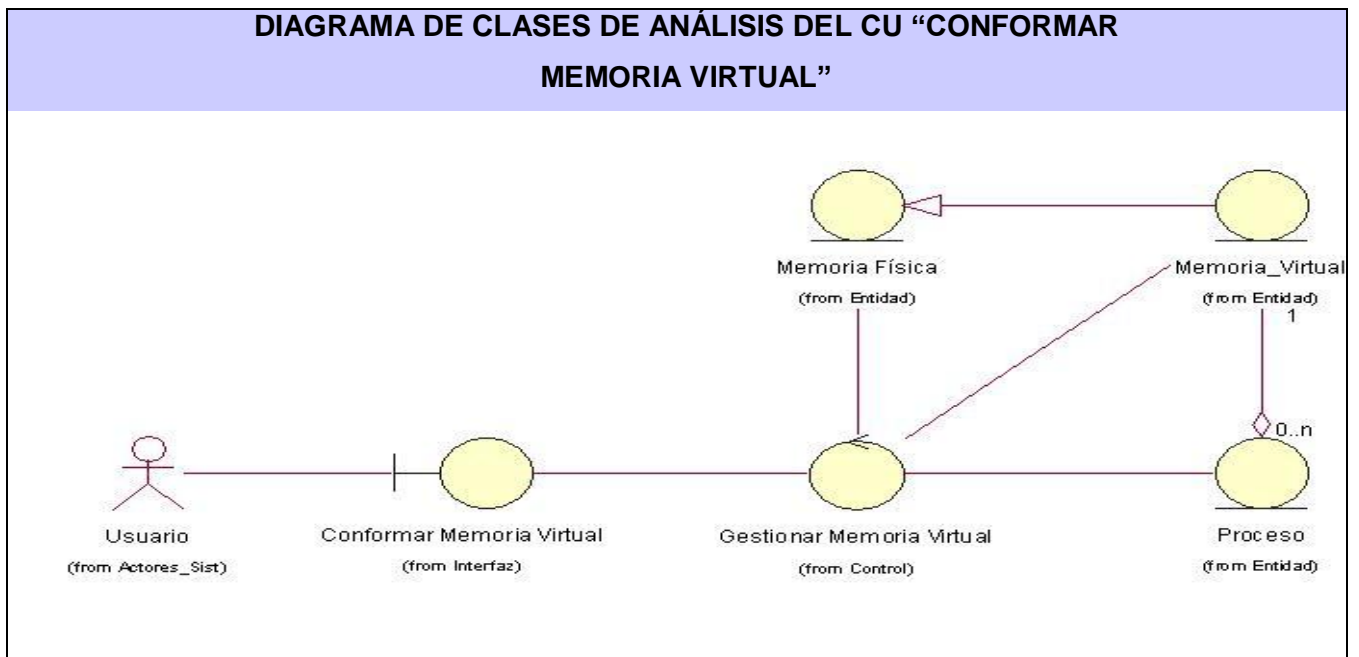


Figura 11 Diagrama de Clases de Análisis del CU Conformar Memoria Virtual

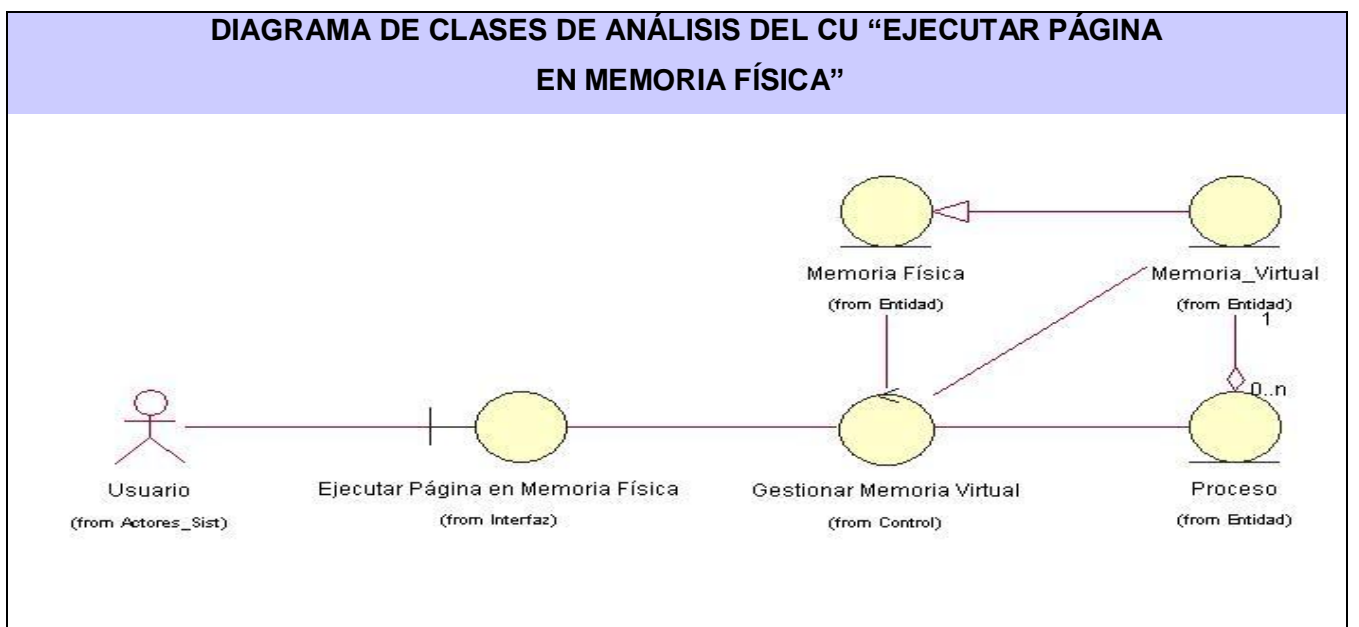


Figura 12 Diagrama de Clases de Análisis del CU Ejecutar Páginas en Memoria Física

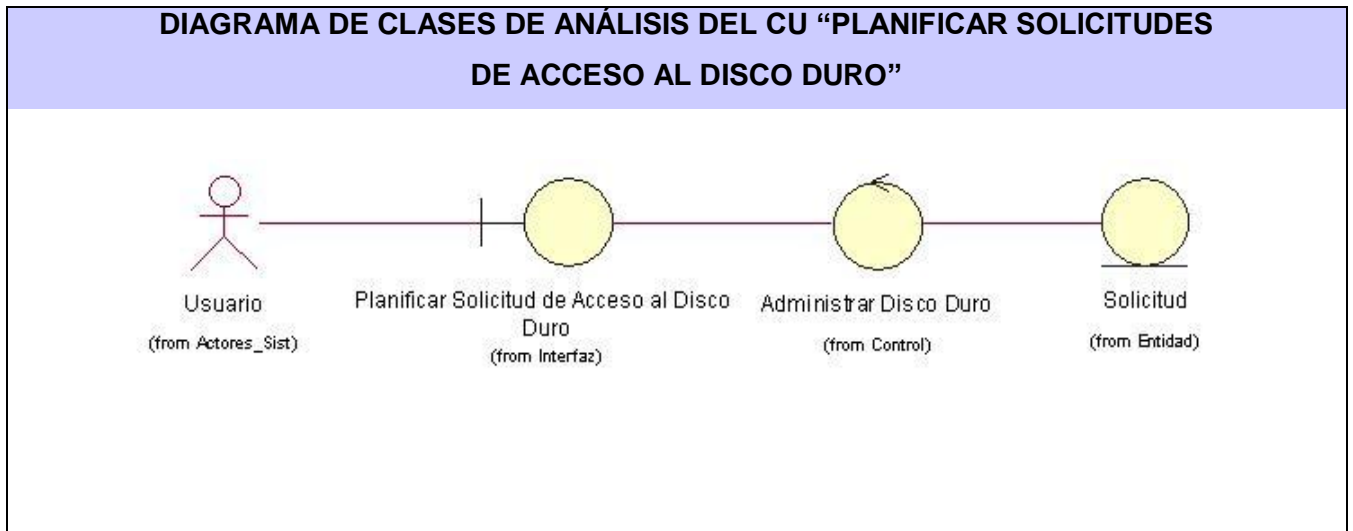


Figura 13 Diagrama de Clases de Análisis del CU Planificar Solicitudes de Acceso al Disco Duro

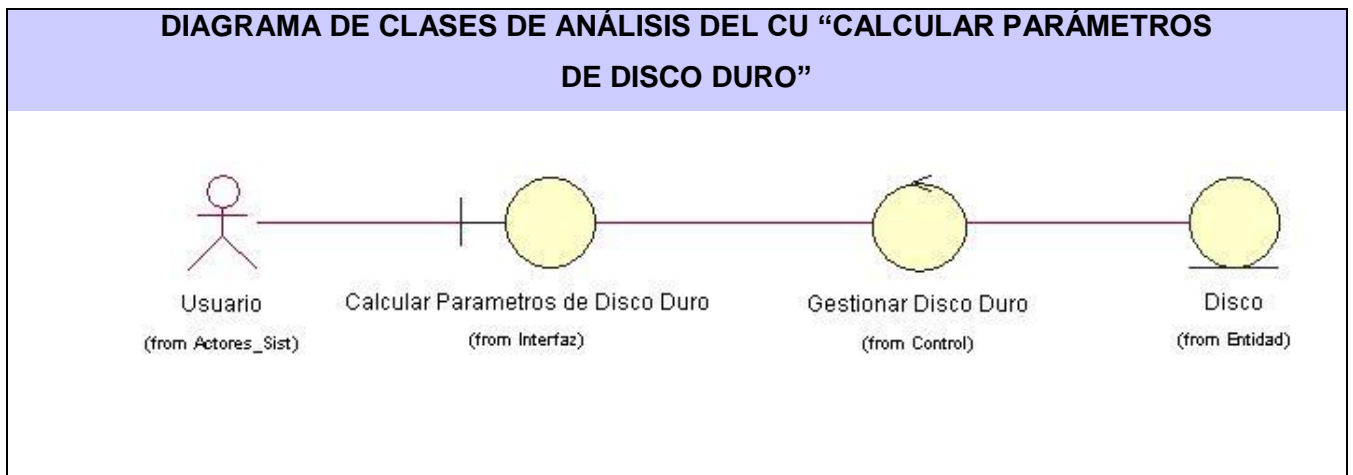


Figura 14 Diagrama de Clases de Análisis del CU Calcular Parámetros del Disco Duro

3.7 Análisis de los resultados.

Es recomendable, antes de comenzar la implementación final del sistema, realizar un análisis de la solución propuesta, que permita valorar y comprobar la efectividad y correspondencia de los artefactos generados, atendiendo a las exigencias del cliente.

Para la realización de este análisis se han tenido en cuenta dos criterios fundamentales, los cuales se exponen a continuación.

Prototipo no Funcional de Interfaz de Usuario.

Los prototipos surgen para validar los requerimientos identificados, estos son simulaciones del posible producto, que luego son utilizados por el usuario final, permitiendo conseguir una importante retroalimentación en cuanto a si el sistema diseñado en base a los requerimientos recolectados le permite al usuario realizar su trabajo de manera eficiente y efectiva. El prototipo no funcional da una idea de cómo quedará el sistema a desarrollar, sobre este se realizan evoluciones y modificaciones hasta determinar claramente los requisitos.

El uso de prototipos no funcionales de interfaces de usuario constituye uno de los mecanismos más usados por los analistas para certificar con el cliente tanto la exactitud de los requisitos como su calidad.

Para la solución propuesta se han generado un total de 20 prototipos no funcionales de interfaces de usuario.

Con el objetivo de obtener una valoración sobre el trabajo realizado se somete el mismo al criterio de varios profesores del colectivo de la asignatura Sistemas Operativos, los cuales poseen experiencia tanto en la impartición de la materia como en el desarrollo de sistemas informáticos. Para obtener la valoración antes mencionada se elaboran una serie de preguntas a tener en cuenta.

- ¿Cumplen con una misma expectativa el diseño de las interfaces elaboradas?
- ¿Es de fácil entendimiento la utilización de las interfaces propuestas?
- ¿Existen interfaces para todas las peticiones que el actor le hace al sistema?
- ¿Se observan todas las posibles interfaces para poder alcanzar el objetivo del caso de uso?

Teniendo como base estos elementos se obtuvieron varios resultados de las valoraciones.

- Se realizaron dos encuentros con el cliente donde se les mostró la propuesta de los prototipos de interfaces y el grado de aceptación de los mismos resultó muy favorable, identificándose también otras funcionalidades que permitieron enriquecer el conjunto de requisitos y con ello la calidad de las respuestas que debe proporcionar el sistema una vez implementado completamente.

- Se presentaron al cliente algunas animaciones en la cual se mostró la propuesta de automatización de los procesos de Planificación de Disco, la cual fue aceptada y se adoptó en la propuesta de interfaz para este caso de uso.

A continuación se transcriben las opiniones generales de los entrevistados:

Ing. Dusniel Horta Centeno

J'Dpto de Sistemas Digitales. Facultad 3.

Profesor de la Asignatura Sistemas Operativos.

Ing. Carlos Yasmany Hidalgo García

Profesor Titular de la Asignatura Sistemas Operativos. Facultad 3.

Ing. Maylin Bacallao Martínez

Profesora de la Asignatura Sistemas Operativos. Facultad 3.

Ing. Dayma Dientao Batista

Profesora de la Asignatura Sistemas Operativos. Facultad 3.

Criterio General:

La modelación del sistema que se propone en el desarrollo de este trabajo evidencia un alto valor práctico que se materializará con la implementación final de la misma. Debe tenerse en cuenta que el proceso de enseñanza de la asignatura Sistemas Operativos es muy complejo en varios de sus temas, debido al alto nivel de abstracción que requiere, tanto para su explicación por parte del profesor como para la comprensión por los estudiantes.

Al realizar el análisis de los artefactos generados y las secuencias de las acciones que se automatizarán, se llega a la conclusión, por parte de los evaluadores, de que estos contemplan y describen claramente la forma de favorecer una mejor comprensión del contenido, teniendo en cuenta el uso de elementos gráficos en la simulación de los algoritmos de Administración de Memoria y Planificación de Disco, algo que se torna muy complejo cuando se efectúa en papel o pizarra.

El uso de interfaces comunes en algunos flujos y actividades de diferentes casos de uso, proporciona homogeneidad en el diseño del sistema. Además la sencillez en la propuesta de las mismas ofrece mayor claridad en la interpretación de los resultados del proceso de simulación.

Patrones de Caso de Uso utilizados.

La técnica de modelación de los casos de usos, es una de las más utilizadas actualmente, lo cual no significa que sea fácil. Para muchos especialistas, le resulta una actividad bastante trabajosa escribir los casos de uso, pues uno de los aspectos que contribuye a esta dificultad es que no existen criterios para comprobar la efectividad y calidad de los mismos. Es por eso que se crean los patrones para los casos de usos.

Estos patrones son diferentes soluciones a problemas reales que se identificaron en el desarrollo de disímiles proyectos. Existen dos categorías o clasificaciones de los patrones para casos de usos: los de procesos y los estructurales.

Para la aprobación de los Casos de Uso del presente trabajo se utilizaron un conjunto de patrones los cuales influyeron sin dudas en la claridad y legibilidad del DCUS propuesto finalmente. Ver Anexo 14.

Aplicación de Métricas

La medición es fundamental para cualquier disciplina de ingeniería, y la ingeniería del software no es una excepción. La medición permite tener una visión más profunda proporcionando un mecanismo para la evaluación objetiva. (Pressman, 2002)

En la actualidad en el proceso de desarrollo de software existen un conjunto de métricas, las cuales se utilizan para la validación de los requisitos identificados en la realización de un software, estas métricas permiten a los analistas validar de una manera correcta que los requisitos identificados durante el proceso de desarrollo, tienen la calidad requerida y cumplen con las normas internacionales. Además existen métricas que permiten validar también el modelo de casos de uso. A continuación se mencionarán algunas de las métricas que fueron aplicadas a los requisitos de software y al modelo de casos de uso del sistema de los módulos *Planificación de Disco (PD)* y *Administración de Memoria (AM)*.

Métricas de la calidad de la especificación

$$R_t = R_f + R_{nf}$$

$$37 = 29 + 8$$

Donde:

Rt: Total de requisitos.

Rf: Cantidad de requisitos funcionales.

Rnf: Cantidad de requisitos no funcionales.

Para determinar la especificidad (ausencia de ambigüedad) de los requisitos. Se explica una métrica basada en la consistencia de la interpretación de los revisores para cada requisito:

Se calcula Q1 para determinar la especificidad de los requisitos.

$$Q1 = R_{ui} / R_t$$

$$0.94 = 35/37$$

Donde:

R_{ui}: Número de requisitos para los que todos los revisores tuvieron interpretaciones idénticas.

Q1: Ausencia de ambigüedad.

Grupo de Revisores que aplicaron esta métrica:

- **Ing. Dusniel Horta Centeno.**
- **Ing. Carlos Yasmany Hidalgo García.**
- **Ing. Maylin Bacallao Martínez.**
- **Ing. Dayma Dientao Batista.**

Cuanto más cerca de 1 esté el valor de Q1, menor será la ambigüedad de la especificación.

El valor de Q1 = 0.94, esto demuestra que los requisitos se encuentran con un alto nivel de especificidad.

Modelo de Métricas para Análisis y Diseño Orientado a Objetos Basado en UML

El modelo tiene por objetivo medir la calidad de los productos intermedios generados en un proyecto de software. El modelo define cuatro atributos genéricos de propiedades de calidad: consistencia, correctitud, completitud y complejidad, que tienen un significado concreto de acuerdo al tipo de artefacto software y al nivel de abstracción que éste describe. Un atributo se analiza en términos de un conjunto de factores cada uno de los cuales tendrá asociada una métrica. Se explican seguidamente de forma resumida los factores que serán evaluados.

Completitud: Grado en que se ha logrado detallar todos los casos de uso relevantes.

Consistencia: Grado en que los casos de uso del sistema describen las interacciones adecuadas entre el usuario y el sistema.

Correctitud: Grado en que las interacciones actor/sistema soportan adecuadamente el proceso del negocio.

Complejidad: Grado de claridad en la presentación de los elementos que describen el contexto y la claridad del sistema.

Para la aplicación de un grupo de métricas al diagrama de caso de uso del sistema. Ver Anexo 15.

Seguidamente se muestra un gráfico donde se evidencia mediante valores numéricos la medición de los factores de calidad que han sido evaluados.

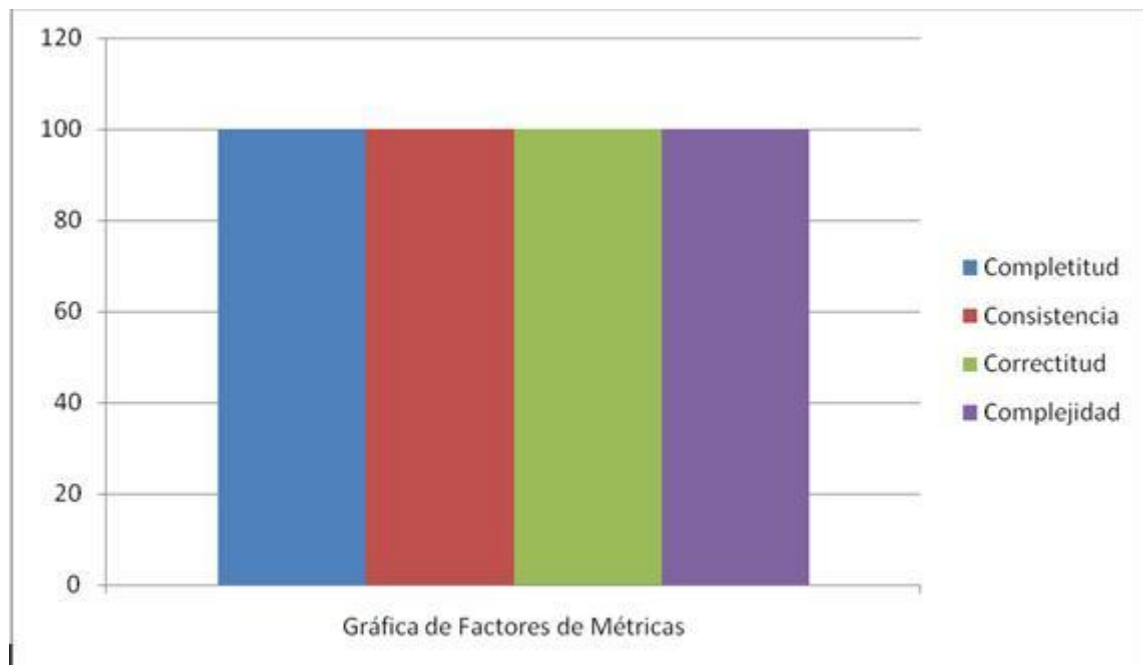


Figura 15 Gráfica de resultados por factores de las métricas aplicadas.

Con la aplicación de estas métricas se pudieron evaluar los factores completitud, consistencia, correctitud y complejidad del diagrama de casos de uso del sistema lo que facilitó la posibilidad de demostrar que los requisitos identificados son implementados en al menos un caso de uso, abarcando de esta forma todas las necesidades expresadas por el cliente, que los casos de uso fueron descritos detalladamente mostrando el flujo alterno a parte del flujo básico lo que da una mayor legibilidad de los mismos, evidenciándose además que estos son iniciados por la interacción del usuario con el sistema o por un evento interno dentro del mismo. Todo esto expuesto anteriormente permitió evidenciar que el

artefacto caso de uso del sistema se encuentra con la calidad requerida para entrar al flujo de trabajo análisis y diseño donde se comenzaría la realización del software.

3.8 Conclusiones del capítulo.

Después del análisis de la disciplina de Requerimientos, se identificaron los Actores que serán los encargados de interactuar con el Sistema. Se ha obtenido, a través del análisis de los procesos modelados en forma de Casos de Uso y cumpliendo con las fases del ciclo de vida de RUP, un Modelo de Sistema que permitirá al equipo que diseñará e implementará el Software, comprender los procesos, facilitando su posterior desarrollo.

Al mismo tiempo se aplicaron algunos Patrones para los Casos de Uso y Métricas tanto para la especificidad de los requisitos como para los Casos de Uso del Sistema. Además de las anteriores técnicas se emplea el Prototipo no Funcional de Interfaz de Usuario como forma de certificar los resultados.

4 CONCLUSIONES

Con el desarrollo de este trabajo, se considera que han sido cumplidos todos los objetivos propuestos:

1. El empleo de RUP como metodología de desarrollo ha sido eficaz y productivo, pues la obtención de los artefactos definidos en cada etapa contribuyen a la clarificación y comprensión de los Modelos elaborados en el presente trabajo. Además, al seguir el orden de dichas Fases y la continuidad de cada artefacto mediante los ciclos de desarrollo, ha logrado una mayor calidad en el resultado final de los mismos.
2. Al estudiar los procesos del Negocio se logró un mejor entendimiento; se identificaron los procesos a automatizar, y se definieron las Reglas del Negocio, que resultan un conjunto de precondiciones a cumplir y alrededor de las cuales se realizó el Modelo de Sistema.
3. La especificación de los Requisitos permitió identificar las funcionalidades y características generales que tendrá el Sistema, para dar cumplimiento a las necesidades y expectativas del cliente, quedando todo plasmado en un Modelo de Sistema; además, la identificación del Actor del Sistema y su papel dentro de los procesos.
4. Esta propuesta de Análisis constituye una base para el trabajo de los diseñadores e implementadores del Sistema, que se encargarán de continuar los próximos flujos definidos por RUP.

5 RECOMENDACIONES

Toda obra humana es perfectible y las que constituyen un resultado intelectual están sujetas al constante cambio producto de la superación de sus creadores. Este trabajo no se encuentra exento de esto y se propone por tanto que se tomen en consideración aquellos elementos que por cuestiones de tiempo o recursos no fue posible incorporar. Se recomienda que en futuros desarrollos de este producto o similares se realice:

1. Realizar los flujos correspondientes a diseño e implementación de la solución propuesta en este trabajo.
2. Realizar el análisis y modelado del tema que abarca la segmentación de memoria como elemento importante de cualquier Sistemas Operativos.
3. Incluir los módulos correspondientes a la Planificación de Proceso y Administración de Recursos.
4. Incluir funcionalidades que permitan guardar en ficheros los ejercicios que se simulan y su posterior recuperación.

6 BIBLIOGRAFÍA

2005. Antecedentes Históricos. *Portal de Nuevas Tecnologías para el Aprendizaje Activo*. [En línea] 8 de Marzo de 2005. [Citado el: 15 de Noviembre de 2007.] http://didas.disca.upv.es:8080/portal_recursos/folder_acciones/folder_accion5/definicion/antecedentes/document_view.

ESTRADA, V. H. M. y NÁJERA, J. M. 2005. La evolución de los laboratorios virtuales durante una experiencia de cuatro años con estudiantes a distancia. *Tropical Biology Online: International Journal of Biology and Conservation*. [En línea] 2005. [Citado el: 20 de Noviembre de 2007.] <http://rbt.biologia.ucr.ac.cr/public/evollab4.doc>.

GUARDADO, J. C. y SÁNCHEZ, M. E. O. 2000. El Laboratorio Virtual: una posibilidad para el desarrollo de habilidades en las Ciencias Naturales. *Red Experimental*. [En línea] 2000. [Citado el: 26 de Noviembre de 2007.] <http://redexperimental.gob.mx/descargar.php?id=465>.

HERÍAS, F. A. C. 2003. Propuesta de Portal de la Red de Laboratorios Virtuales y Remotos de CEA. *Dpto. de Física, Ingeniería de Sistemas y Teoría de la Señal*. [En línea] 2003. [Citado el: 7 de Diciembre de 2007.] <http://www.disc.ua.es/docenweb/Docs/PropuestaDePortal.pdf>.

MADERA, E. J. C. y VIZOSO, T. Á. 1997. Las prácticas de Laboratorio Docentes en la enseñanza de la Física. *Tesis, Documentos, Publicaciones y Recursos Educativos*. [En línea] 1997. [Citado el: 10 de Diciembre de 2007.] <http://www.monografias.com/trabajos29/practicas-laboratorio/practicas-laboratorio.shtml>.

1997. Resolución Económica del V Congreso del PCC. *Partido Comunista de Cuba*. [En línea] 9 de Octubre de 1997. [Citado el: 12 de Noviembre de 2007.] http://www.pcc.cu/pccweb/pcc/congresos/V_congreso/resolucion.php.

ROSADO, L. y HERREROS, J. R. 2005. Nuevas aportaciones didácticas de los laboratorios virtuales y remotos en la enseñanza de la Física. *Formatex Research Center*. [En línea] 2005. [Citado el: 23 de Noviembre de 2007.] <http://www.formatex.org/micte2005/286.pdf>.

SALAVERRÍA, Á y FERREIRA, L. F. 2005. Laboratorio Virtual para el auto aprendizaje de la electrónica aplicada. *E.U.I.T. de Telecomunicaciones*. [En línea] 2005. [Citado el: 12 de Noviembre de 2007.] <http://www.euitt.upm.es/taee06/papers/SD/p90.pdf>.

VARY, J. P. 2000. Informe de la Reunión de Expertos sobre Laboratorios Virtuales. *Sitio Oficial de la UNESCO*. [En línea] 2000. [Citado el: 12 de Noviembre de 2007.] <http://unesdoc.unesco.org/images/0011/001191/119102s.pdf>.

- CACERES, P. y MARCOS, E.** *Procesos Agiles para el Desarrollo de Aplicaciones Web*
- DURAN, A. B,B, RUIZ, A. y TORO, M. 1999.** *A Requirements Elicitation Approach Based in Templates and Patterns*. Buenos Aires, Argentina : Workshop de Engenharia de Requisitos, 1999.
- ESCRIBANO, G. F. 2002.** Introduccion a Extreme Programming. [En línea] 2002. [Citado el: 2 de Febrero de 2008.] <http://www.infoab.uclm.es/asignaturas/42551/trabajosAnteriores/Presentacion-XP.pdf>.
- IBM, OOTC. 1997.** *Developing Object Oriented Software*. Prentice - Hall : s.n., 1997.
- Jacobson, Ivar, Booch, G y Rumbaugh, J. 2000.** *"El Proceso Unificado de Desarrollo de Software"*. Madrid : The Addison- Wesley Object Technology Series, 2000.
- Kendall, K. K. 2005.** *"Análisis y Diseño de Sistemas"*. Mexico : Prentice Hall, 2005.
- MENDOZA, M. A. 2004.** Metodologías de Desarrollo de Software. [En línea] 7 de Junio de 2004. [Citado el: 23 de Enero de 2008.] http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software.html
- ORALLO, E. H. 2005.** El Lenguaje Unificado de Modelado (UML). [En línea] 2005. [Citado el: 2 de Marzo de 2008.] <http://www.disca.upv.es/enheror/pdf/ActaUML.PDF>.
- PAN, D., ZHU, D. y JOHNSON, K. 2001.** *Requirements Engineering Techniques.Internal Report*. University of Calgary. Canada : s.n., 2001.
- RAGHAVAN, S., ZELESNIK y FORD, G. 1994.** *Lectures Notes of Requirements Elicitation CMU/SEI - 94 - EM- 10*. s.l. : Educational Materials, 1994.
- SANCHEZ, M. A. 2004.** *"Tesis Metodologías de Desarrollo de Software Ing. Informático"*. Perú : Team Soft, 2004.
- UML. 2001.** *Unified modeling Language*. s.l. : Version 1.4, 2001.
- VERA, K. L. 2006.** Ingeniería de Software - RUP - UML. [En línea] 2006. [Citado el: 23 de Febrero de 2008.] <http://www.mmug.cl/articulos.php>.
- LARMAN, C.** UML y Patrones. Introducción al análisis y diseño orientado a objeto. 1999. p.

MSF, A. d. (s.f.). M.S.F. Obtenido de "Disciplina de administración del proyecto - M.S.F.":
<http://www.gpicr.com/msf.aspx>

Somerville, I. (2002). "Ingeniería de Software.". Addison Wesley.

Letelier, P., & Penadés, M. d. (s.f.). *Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP).***Beck, K. (2000).** "Una explicación de la programación extrema. Aceptar el cambio". Addison Wesley.

Bernárdez, D., Toro (2004). *UNA PROPUESTA PARA LA VERIFICACIÓN DE REQUISITOS BASADA EN MÉTRICAS.* *Revista de Procesos y Métricas de las Tecnologías de la Información.* VOL. 1, Nº 2,12.

Francisco Charte Ojeda. *Programación Visual C#. NET Ediciones Anaya Multimedia (GRUPO ANAYA, S.A.)*,2002.

Roger S. Pressman (2002). *Ingeniería de Software. Un enfoque práctico.* España.

7 GLOSARIO DE TÉRMINOS

A

Análisis (Flujo de Trabajo)

Flujo de trabajo fundamental cuyo propósito principal es analizar los requisitos descritos en la captura de requisitos, mediante su refinamiento y estructuración. El objetivo de esto es: lograr una comprensión más precisa de los requisitos y obtener una descripción de los requisitos que sea más fácil de mantener y que ayude a dar estructura al sistema en su conjunto.

Artefacto

Pieza de información tangible que es creada, modificada y usada por los trabajadores al realizar actividades; representa un área de responsabilidad, y es candidata a ser tenida en cuenta para el control de la configuración. Un artefacto puede ser un modelo, un elemento de un modelo, o un documento.

Actor

Conjunto coherente de roles que los usuarios de casos de uso desempeñan cuando interactúan con estos casos de uso.

Actores del Negocio

Serán aquellas personas que se beneficiarán directamente con el Negocio existente.

Actores del Sistema

Serán los trabajadores del Negocio, que se beneficiarán directamente con el sistema ya implementado (por ejemplo se beneficiarán con una mayor efectividad y un menor costo y esfuerzo en las operaciones).

Analista de sistema

Conduce y coordina la extracción de requerimientos y la modelación de casos de uso perfilando la funcionalidad del sistema y delimitándolo.

C

CASE

Ingeniería de Software Asistida por Ordenador. Computer Aided Software Engineering

Caso de Uso

Es una descripción de un conjunto de secuencia de acciones, incluyendo variaciones, que un sistema lleva a cabo y que conduce a un resultado observable de interés para un actor determinado.

CMM

(Capability Maturity Model), es un modelo de evaluación de los procesos de una organización.

D

Diagrama

La presentación grafica de un conjunto de elementos, usualmente representado como un grafo conectado de vértices (elementos). Y arcos (relaciones).

Diagramas de Actividades

Diagramas que muestran el flujo de actividad a actividad, tratan la vista dinámica de un sistema. Es un caso especial de diagramas de estado en el cual todos o casi todos los estados son estados de acción y en el cual todas o casi todas las transiciones son disparadas por la terminación de las acciones en los estados origen.

Diagrama de Casos de Uso

Diagramas que muestran un conjunto de Casos de Uso y de Actores y sus relaciones; los diagramas de Casos de Uso muestran los Casos de Uso de un sistema desde un punto de vista estático.

Diagramas de Objetos

Diagrama que muestra un conjunto de objetos y sus relaciones entre si y con los trabajadores del Negocio en un momento determinado; muestran el diseño o los procesos de un sistema desde un punto de vista estático.

Dirección Lógica

Direcciones existentes en los programas.

Dirección Física

Es el resultado obtenido de la transformación de las Direcciones Lógicas.

E

Entidades

Representa un contenedor de información, algo físico que se utilice en el proceso del negocio y que sirva para obtener información o para actualizar información. Generalmente tiene estados, en dependencia de en qué momento aparezca como parte del proceso.

F

Fase

Período de tiempo entre dos hitos principales de un proceso de desarrollo.

Flujo alternativo

Camino que pueden tomar las actividades en un Caso de Uso que se aleja del Flujo Normal de Eventos como una alternativa. Pueden existir varios.

Flujo de eventos

Camino normal que toman las actividades realizadas en un Caso de Uso.

G

Grado de Multiprogramación

Es el número de particiones existentes en la memoria.

H

Hueco

Indica un espacio libre en la memoria.

I

Ingeniería de Software

Disciplina de la Ingeniería que concierne a todos los aspectos de la producción de software. Es una parte de la Ingeniería de Sistemas (concierne a todos los aspectos del desarrollo de sistemas basados en computo, que incluyen hardware, software y el proceso de Ingeniería).

IEEE

Instituto de IEEE de los ingenieros electrónicos eléctricos. Fundado en 1963, IEEE es una organización integrada por ingenieros, científicos, y estudiantes. IEEE es el mejor conocido para los estándares que se convierten para la industria del ordenador y del elemento electrónico.

ISO 9000

Es un conjunto de normas de calidad, establecidas por la Organización Internacional para la Estandarización (ISO), para proporcionar un marco normativo alrededor del cual se basa eficazmente un sistema administrativo de calidad.

M

Marcos de Páginas

Memoria de la máquina que se divide en bloques de igual longitud donde solo se almacena una página.

Metodologías

Se encargan de elaborar estrategias de desarrollo de software que promuevan prácticas adaptativas en vez de predictivas; centradas en las personas o los equipos, orientadas hacia la funcionalidad y la entrega, de comunicación intensiva y que requieren implicación directa del cliente.

Memoria Virtual

Es una técnica que consiste en permitir la ejecución de procesos que puedan no estar completamente en memoria.

Monoprogramación

Presupone la existencia en memoria de un solo proceso.

Multiprogramación

Presupone la existencia en memoria de varios procesos que comparten el tiempo del procesador.

MSF

Metodología. Microsoft Solution Framework.

N

Negocio

Cualquier ambiente o entorno en cual esta enmarcado el problema.

P

Páginas

Programas que se dividen en unidades de tamaño fijo.

Particiones Fijas

El número de particiones definidas estáticamente en el momento de la carga del sistema.

Particiones Variables

Dependen de las necesidades planteadas por cada trabajo, esto ocurre cuando no existe un número predeterminado de particiones ni las longitudes son fijas.

Pos condiciones

Una restricción que ha de ser cierta al completarse una operación.

Precondiciones

Una restricción que ha de ser cierta cuando una operación es invocada.

Proceso

Conjunto de operaciones, acciones, cambios o funciones que se realizan para alcanzar un fin.

Proceso del Negocio

Funciones que se desarrollan en el ambiente o entorno que definimos como negocio.

PEA

Proceso de enseñanza-aprendizaje.

R

Reglas del Negocio

Describen políticas que deben cumplirse o condiciones que deben satisfacerse, por lo que regulan algún aspecto del negocio.

RUP

Proceso Unificado de Software (Rational Unified Process).

Requerimiento Funcional (Requisito Funcional)

Una capacidad que el sistema cumplirá. Es una tarea simple enunciada con un solo verbo. Se corresponde con futuras opciones, acciones ocultas y condiciones extremas a determinar por el software.

Requerimiento No Funcional (Requisito No Funcional)

Propiedades o cualidades que el producto de software debe tener.

Rol

Papel que desempeña una persona en un determinado momento; una misma persona puede desempeñar distintos roles a lo largo del proceso. Comportamiento específico de una entidad que participa en un contexto particular.

S

Software

Es la suma total de los programas de cómputo, procedimientos, reglas, documentación y datos asociados que forman parte de las operaciones de un sistema de cómputo y que “un producto de software es un producto diseñado para un usuario”.

Simulador

Es un aparato que permite la simulación de un sistema, reproduciendo su comportamiento. Los simuladores reproducen sensaciones que en realidad no están sucediendo. Pretende reproducir tanto las sensaciones físicas (velocidad, aceleración, percepción del entorno) como el comportamiento de los equipos de la máquina que se pretende simular.

T

TIC

Tecnología de la Informática y las Comunicaciones.

U

UCI

Universidad de Ciencias Informáticas

UML

El Lenguaje Unificado de Modelado prescribe un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos, y describe la semántica esencial de lo que estos diagramas y símbolos significan. Mientras que ha habido muchas notaciones y métodos usados para el diseño orientado a objetos, ahora los modeladores sólo tienen que aprender una única notación.

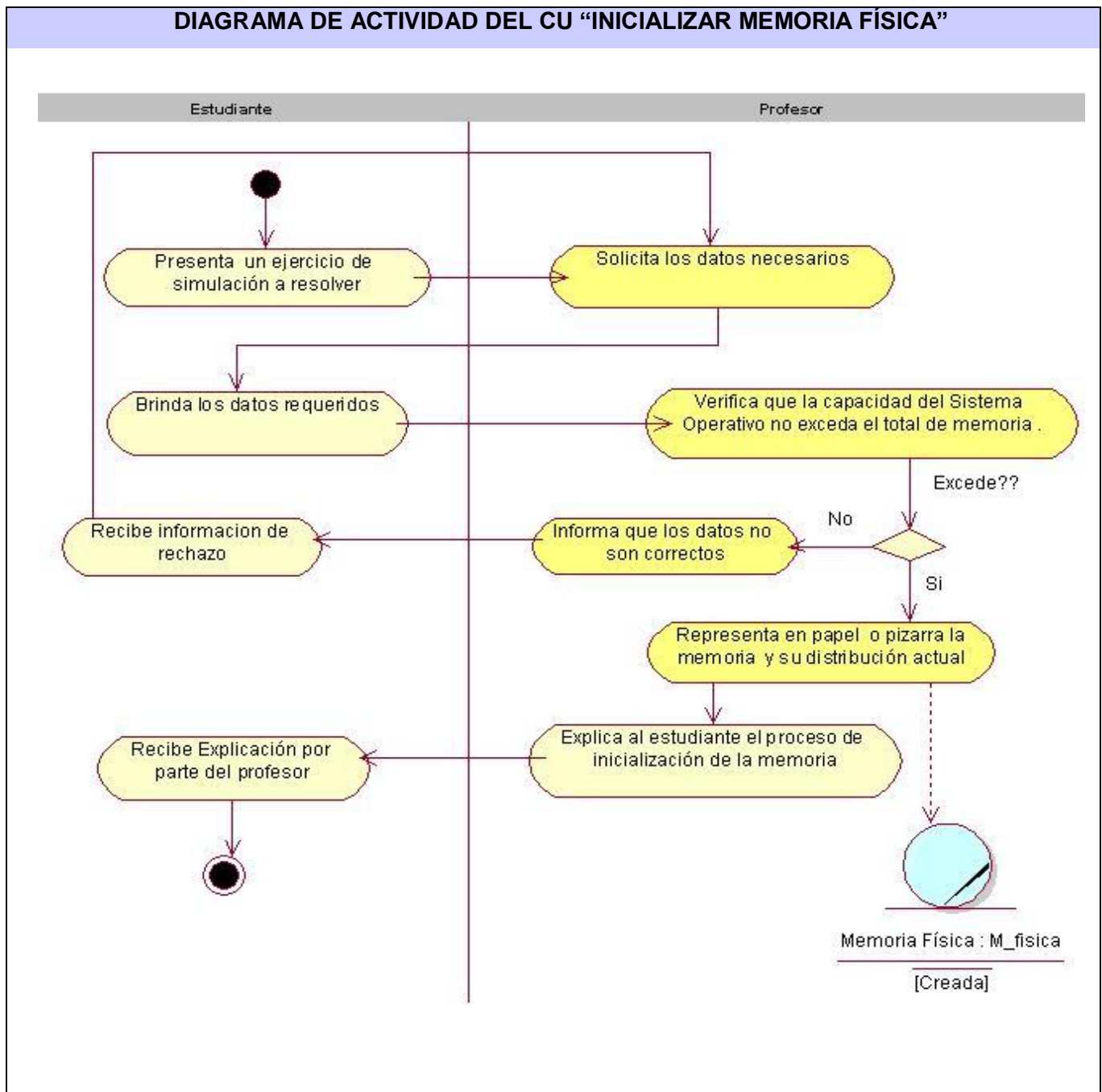
X

XP

Extreme Programming. Es una metodología de desarrollo de software.

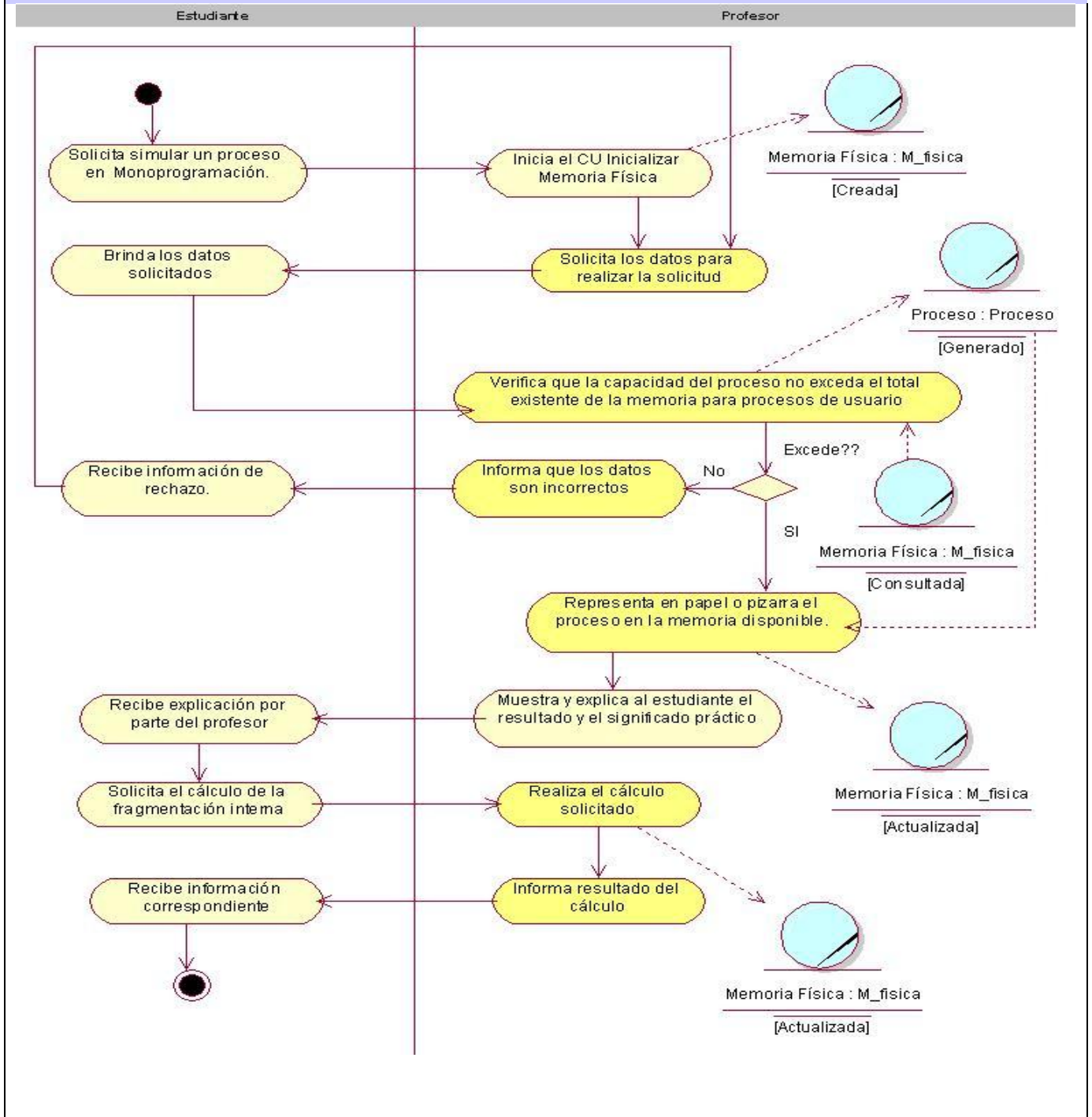
8 ANEXOS

Anexo 1



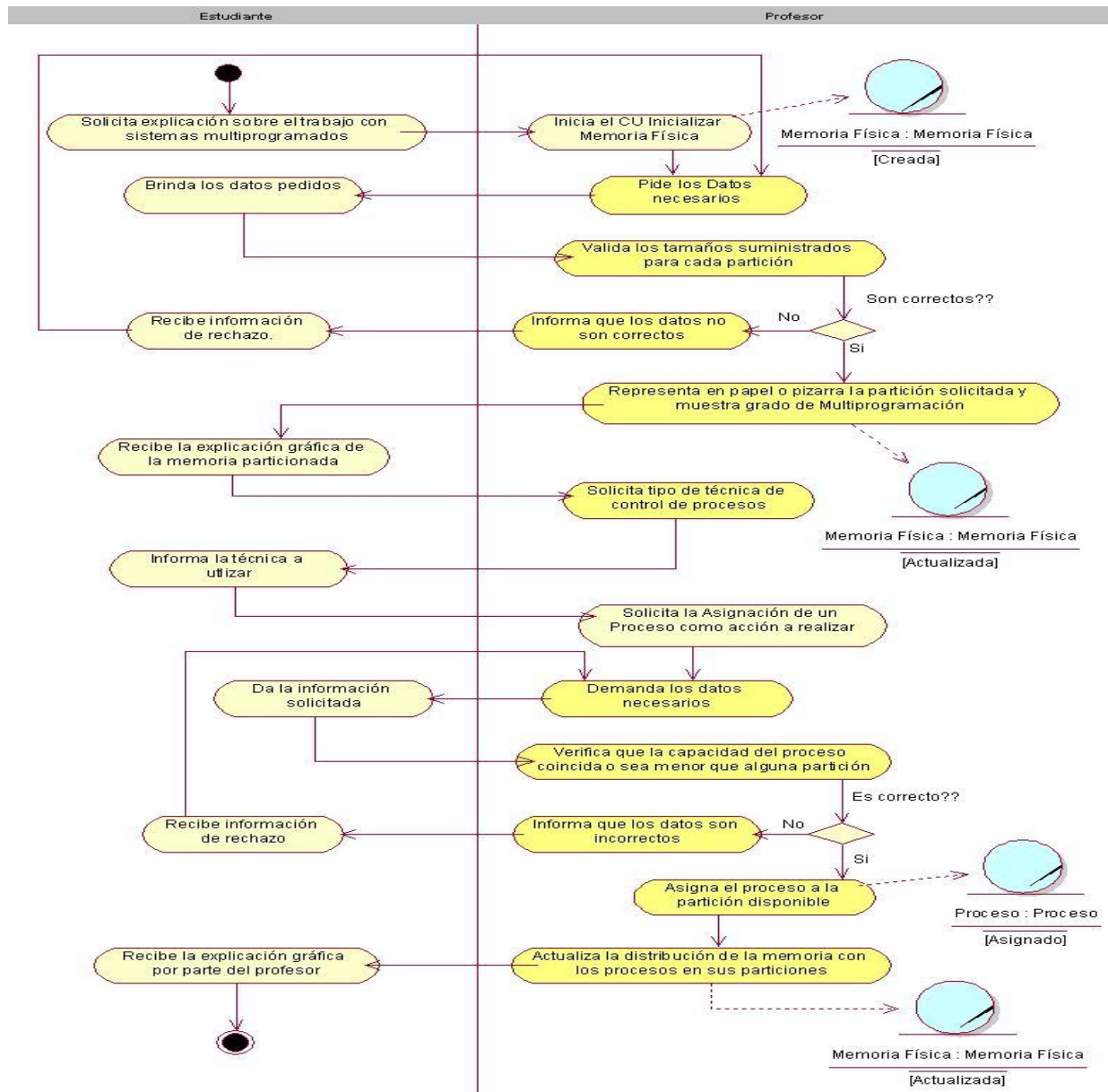
Anexo 2

DIAGRAMA DE ACTIVIDAD DEL CU "SIMULAR MEMORIA EN MONOPROGRAMACIÓN"



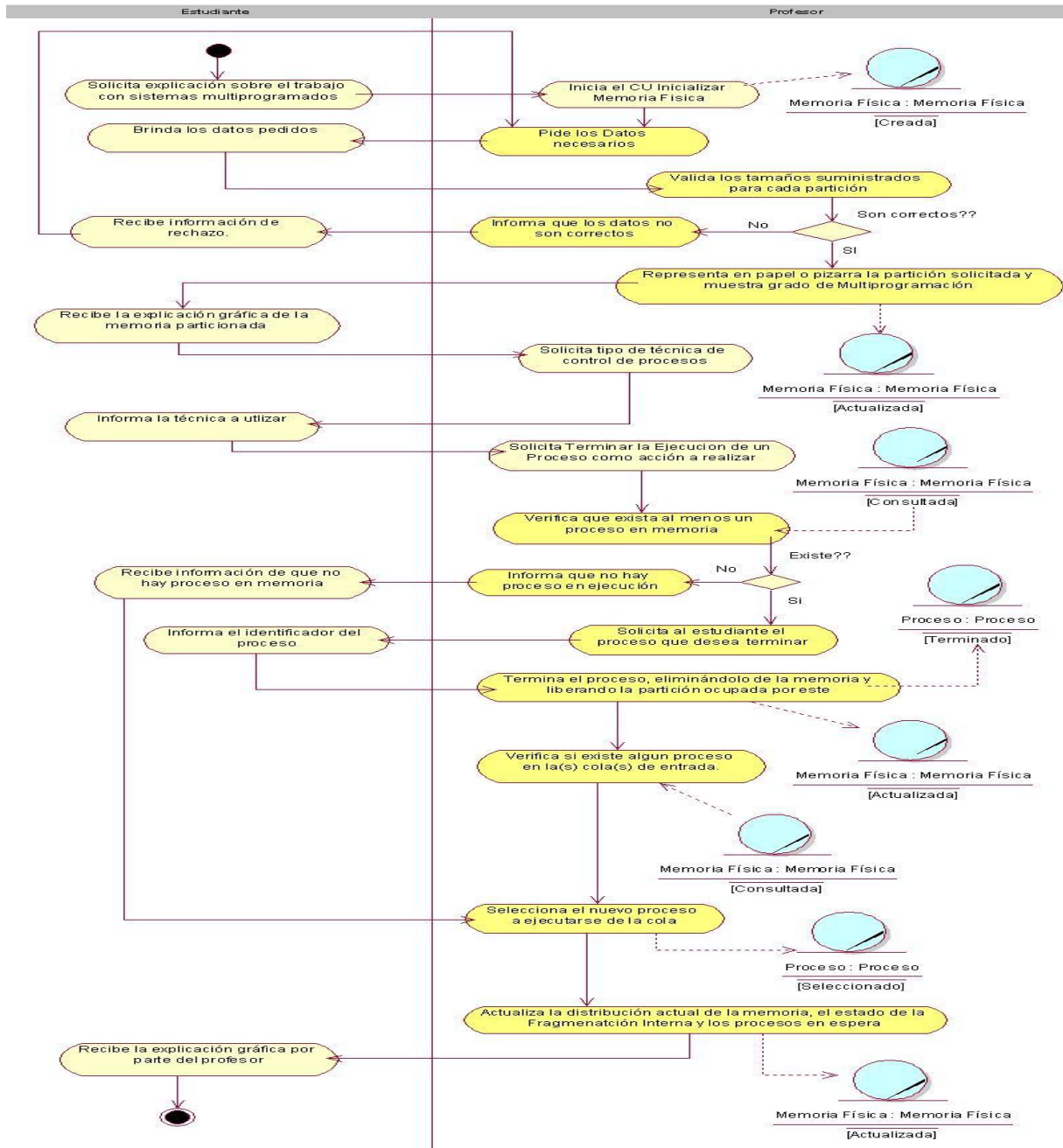
Anexo 3

DIAGRAMA DE ACTIVIDAD DEL CU "SIMULAR MEMORIA EN MULTIPROGRAMACIÓN CON PARTICIONES FIJAS_SECCIÓN_ASIGNAR PROCESO"



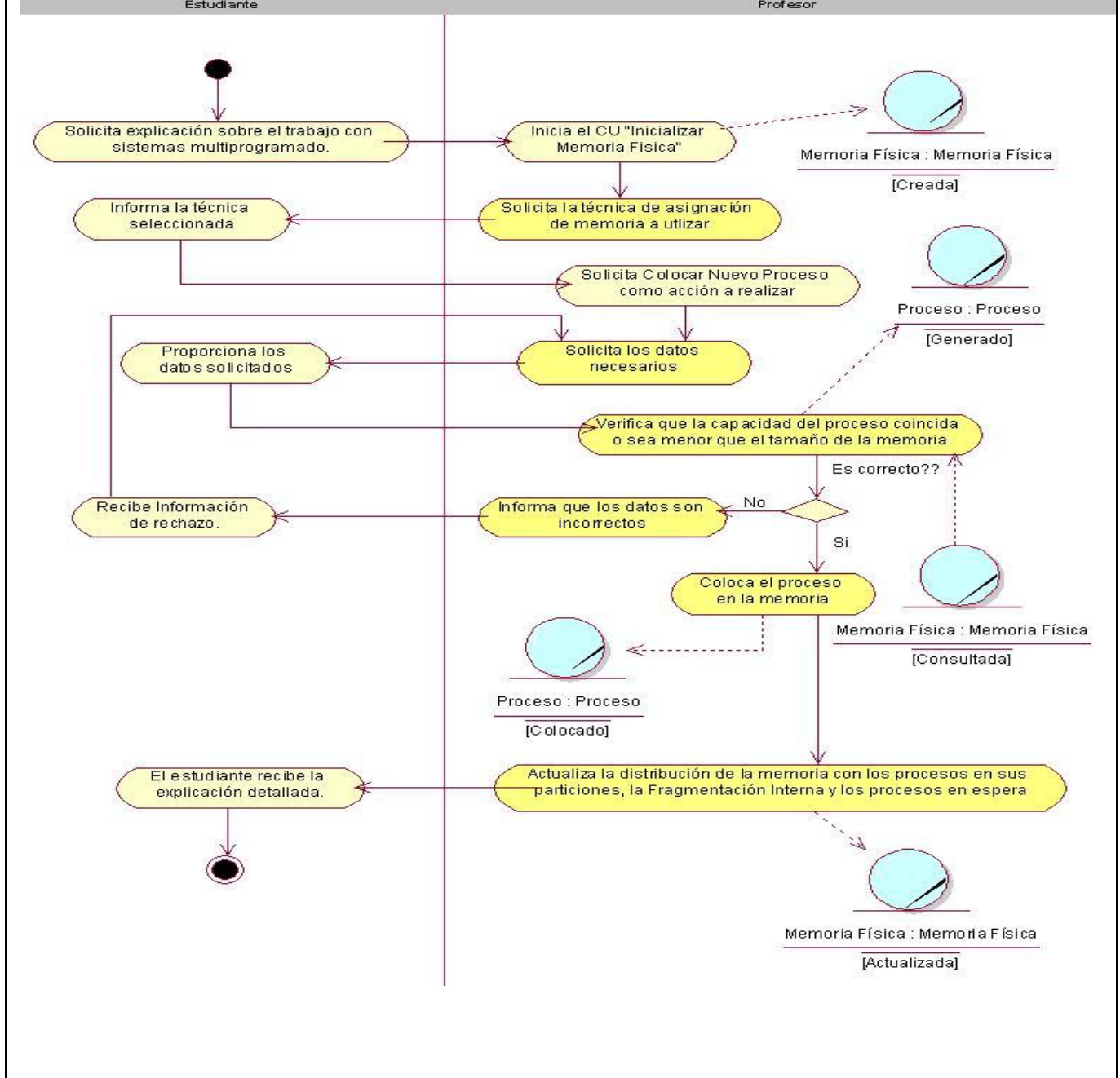
Anexo 4

DIAGRAMA DE ACTIVIDAD DEL CU "SIMULAR MEMORIA EN MULTIPROGRAMACIÓN CON PARTICIONES FIJAS_SECCIÓN_TERMINAR PROCESO"



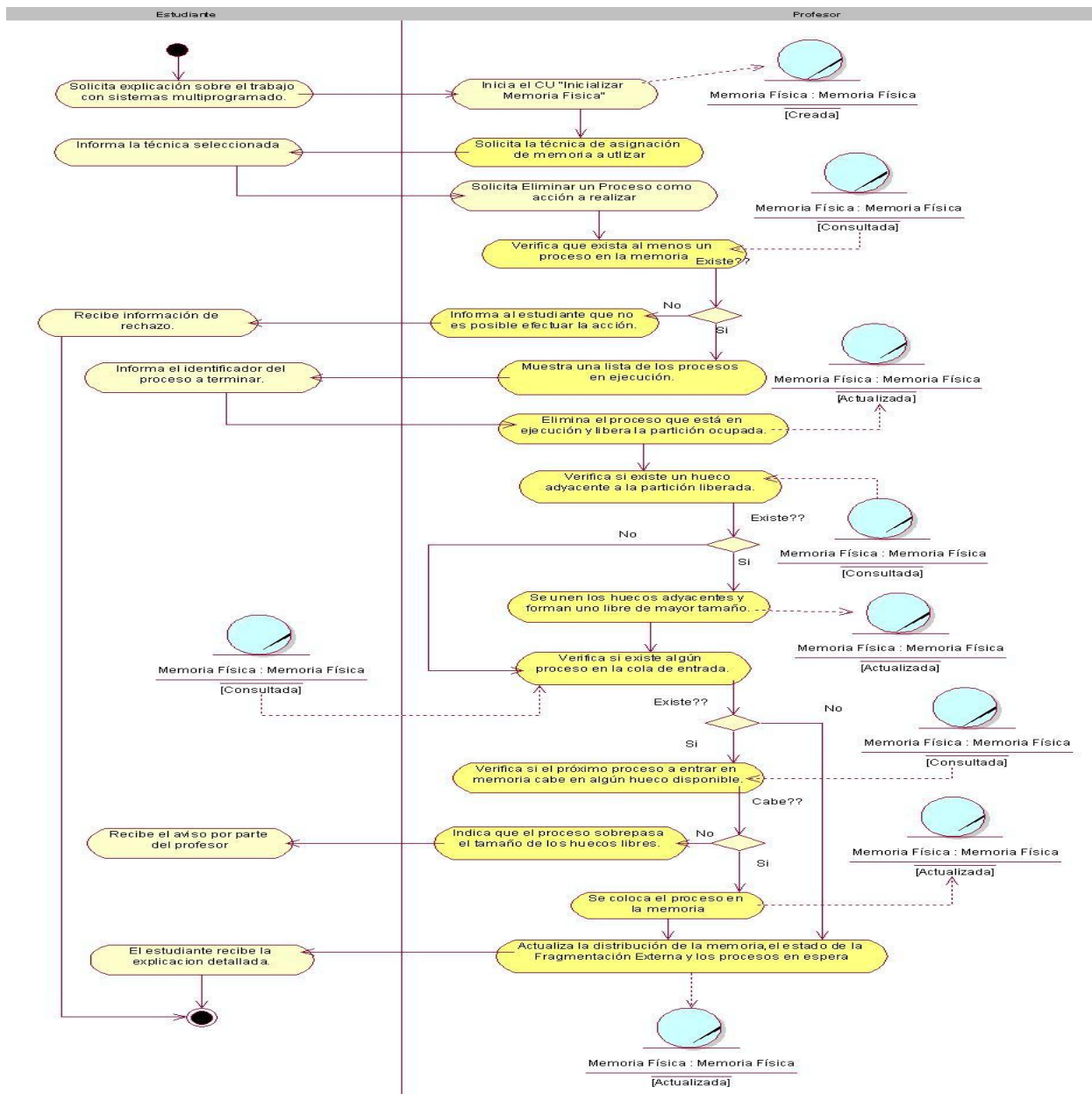
Anexo 5

DIAGRAMA DE ACTIVIDAD DEL CU "SIMULAR MEMORIA EN MULTIPROGRAMACIÓN CON PARTICIONES VARIABLES_SECCIÓN_COLOCAR PROCESO EN MEMORIA"



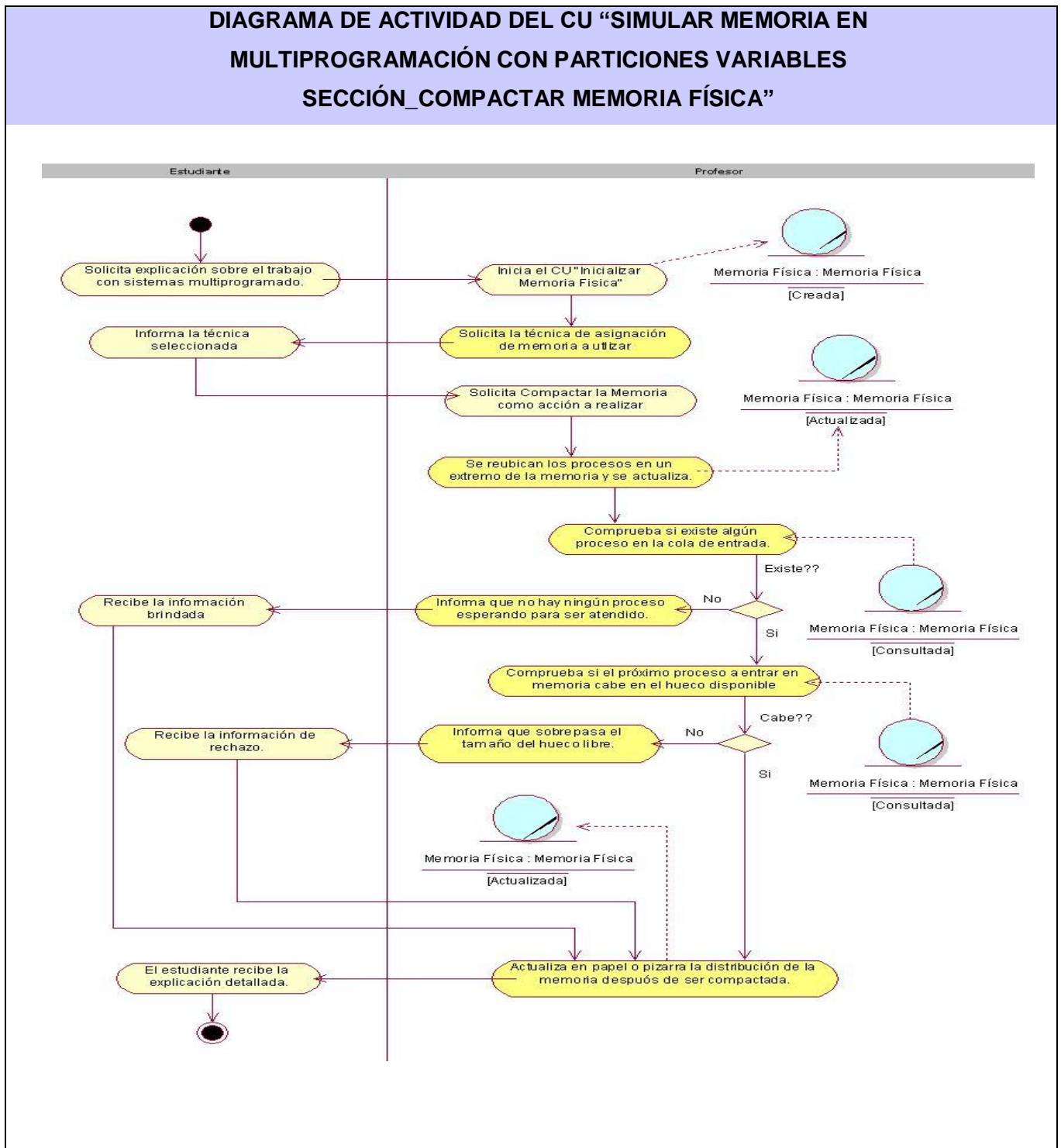
Anexo 6

IAGRAMA DE ACTIVIDAD DEL CU "SIMULAR MEMORIA EN MULTIPROGRAMACIÓN CON PARTICIONES VARIABLES SECCIÓN_ELIMINAR PROCESO"



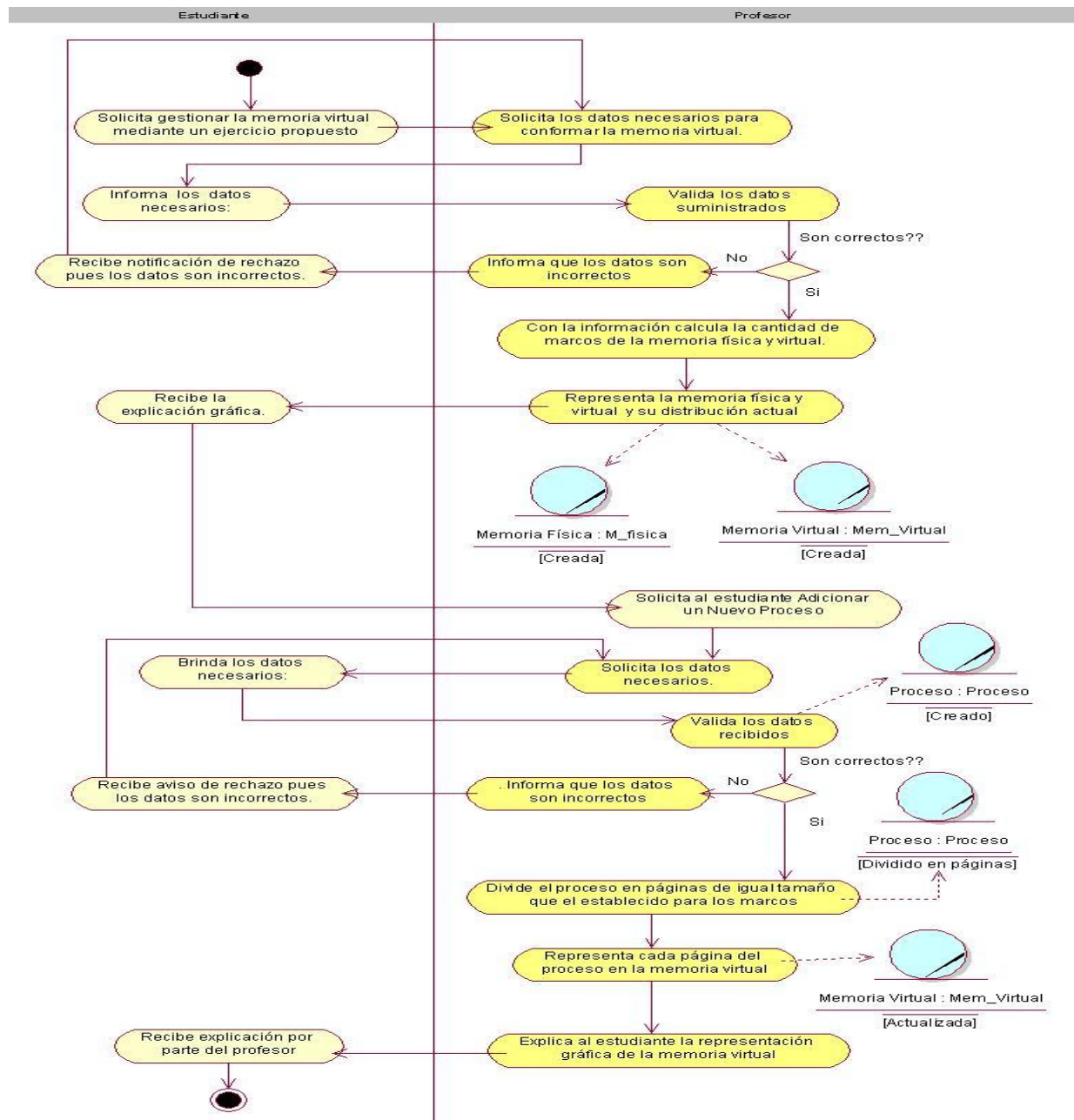
Anexo 7

DIAGRAMA DE ACTIVIDAD DEL CU "SIMULAR MEMORIA EN MULTIPROGRAMACIÓN CON PARTICIONES VARIABLES SECCIÓN_COMPACTAR MEMORIA FÍSICA"

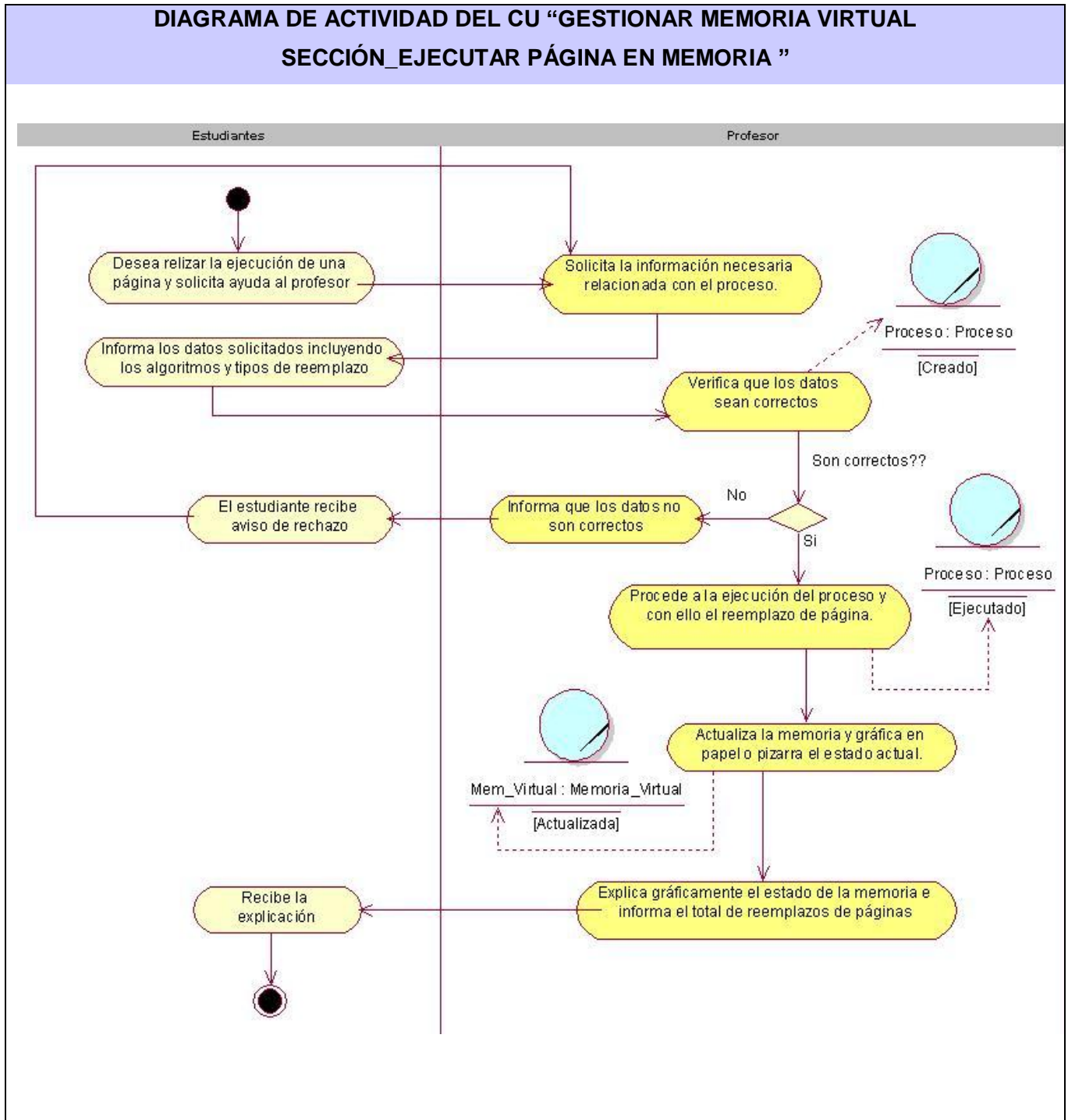


Anexo 8

DIAGRAMA DE ACTIVIDAD DEL CU "GESTIONAR MEMORIA VIRTUAL
SECCIÓN_ADICIONAR PROCESO A MEMORIA "

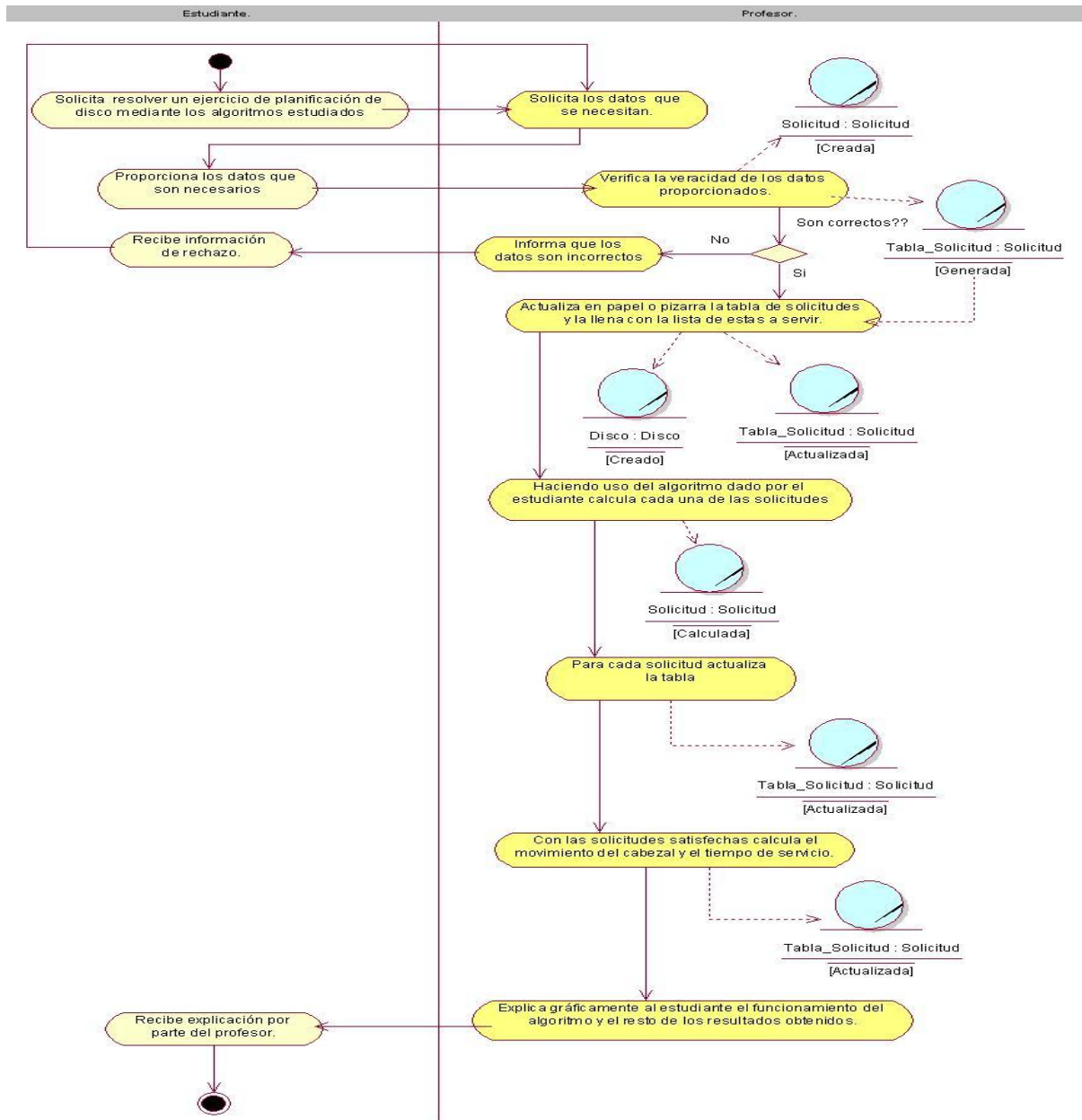


Anexo 9



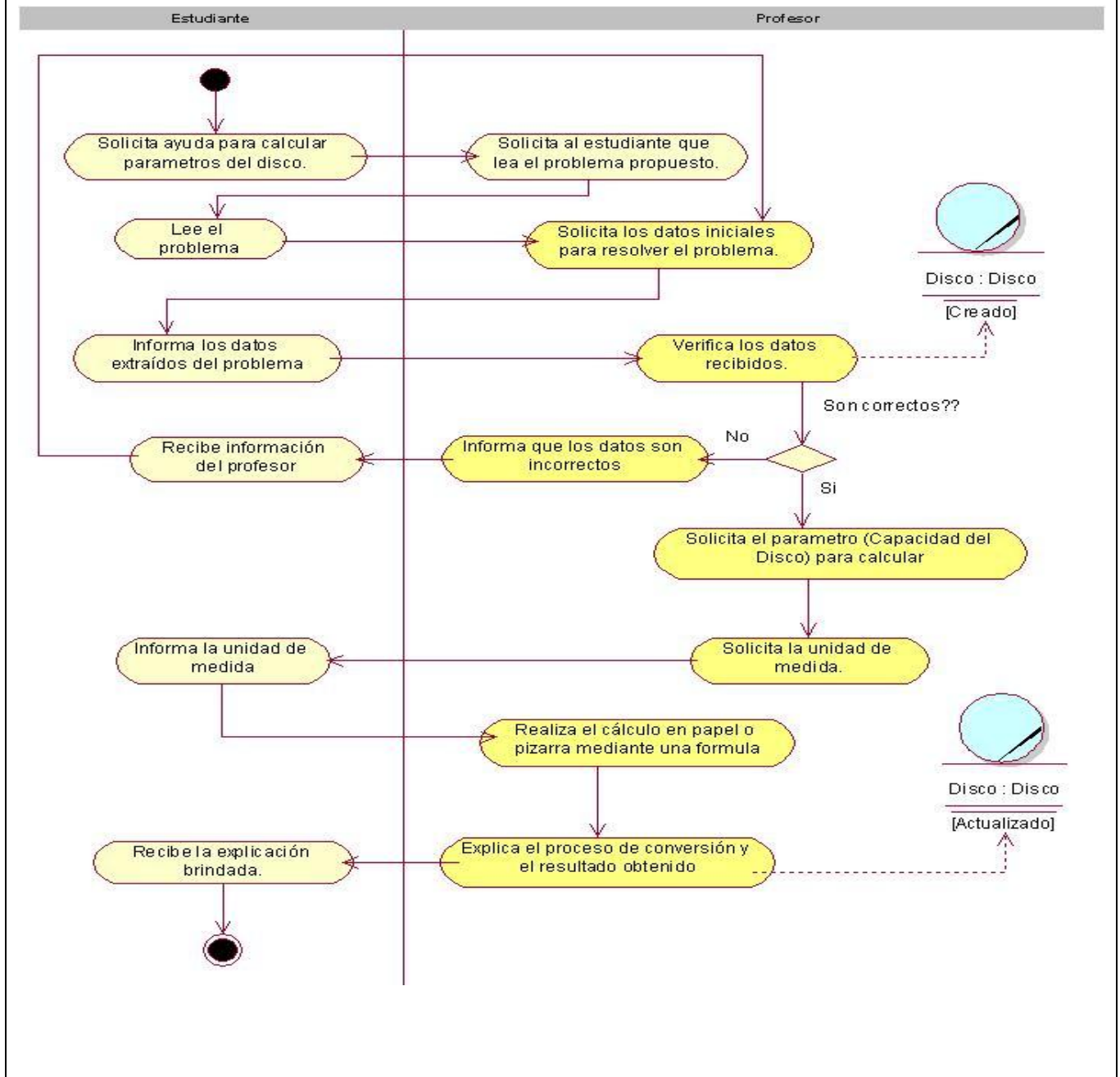
Anexo 10

DIAGRAMA DE ACTIVIDAD DEL CU “PLANIFICAR SOLICITUDES DE ACCESO AL DISCO DURO”



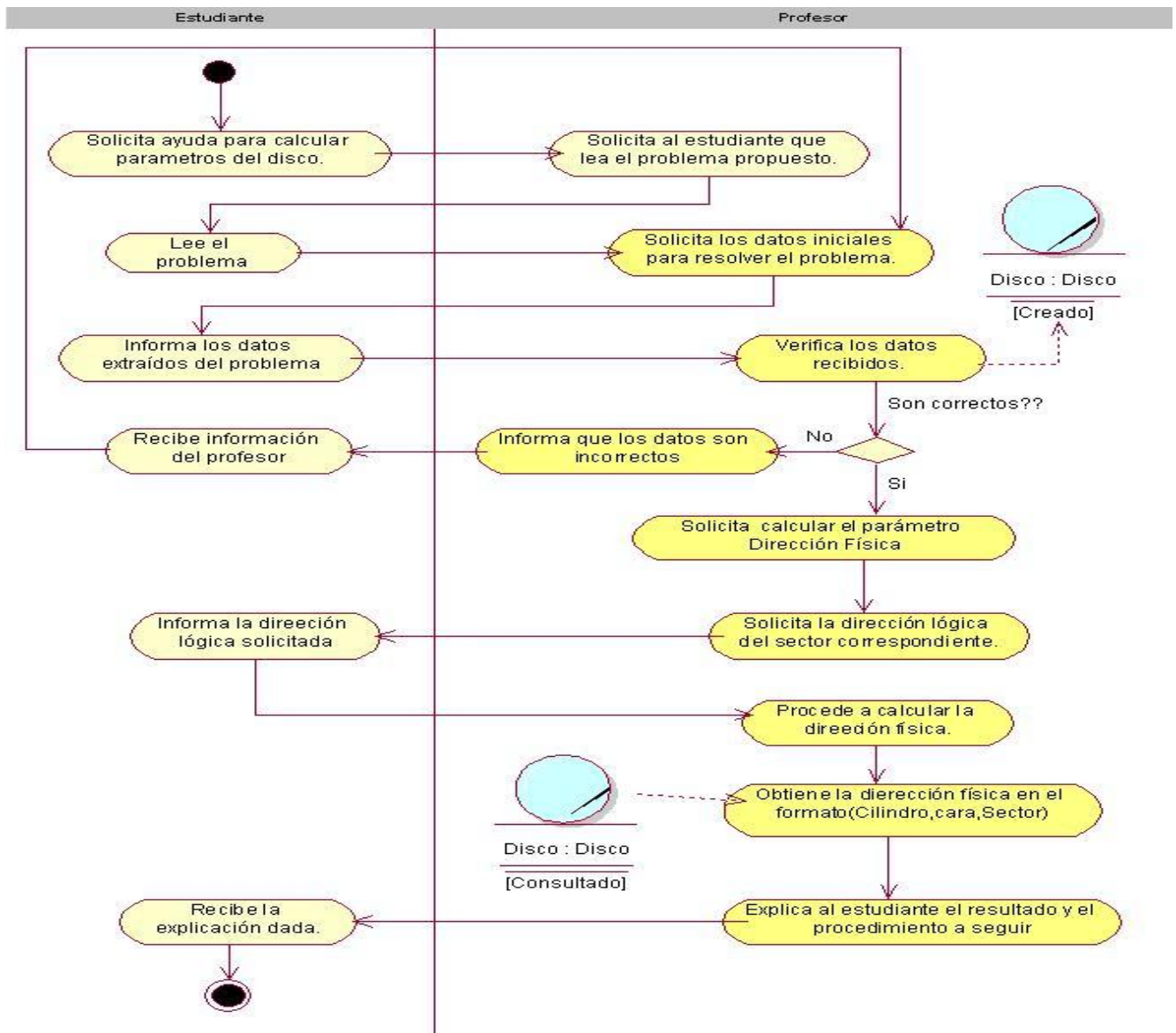
Anexo 11

**DIAGRAMA DE ACTIVIDAD DEL CU “CALCULAR
PARÁMETROS DEL DISCO DURO
SECCIÓN_CALCULAR CAPACIDAD DEL DISCO”**



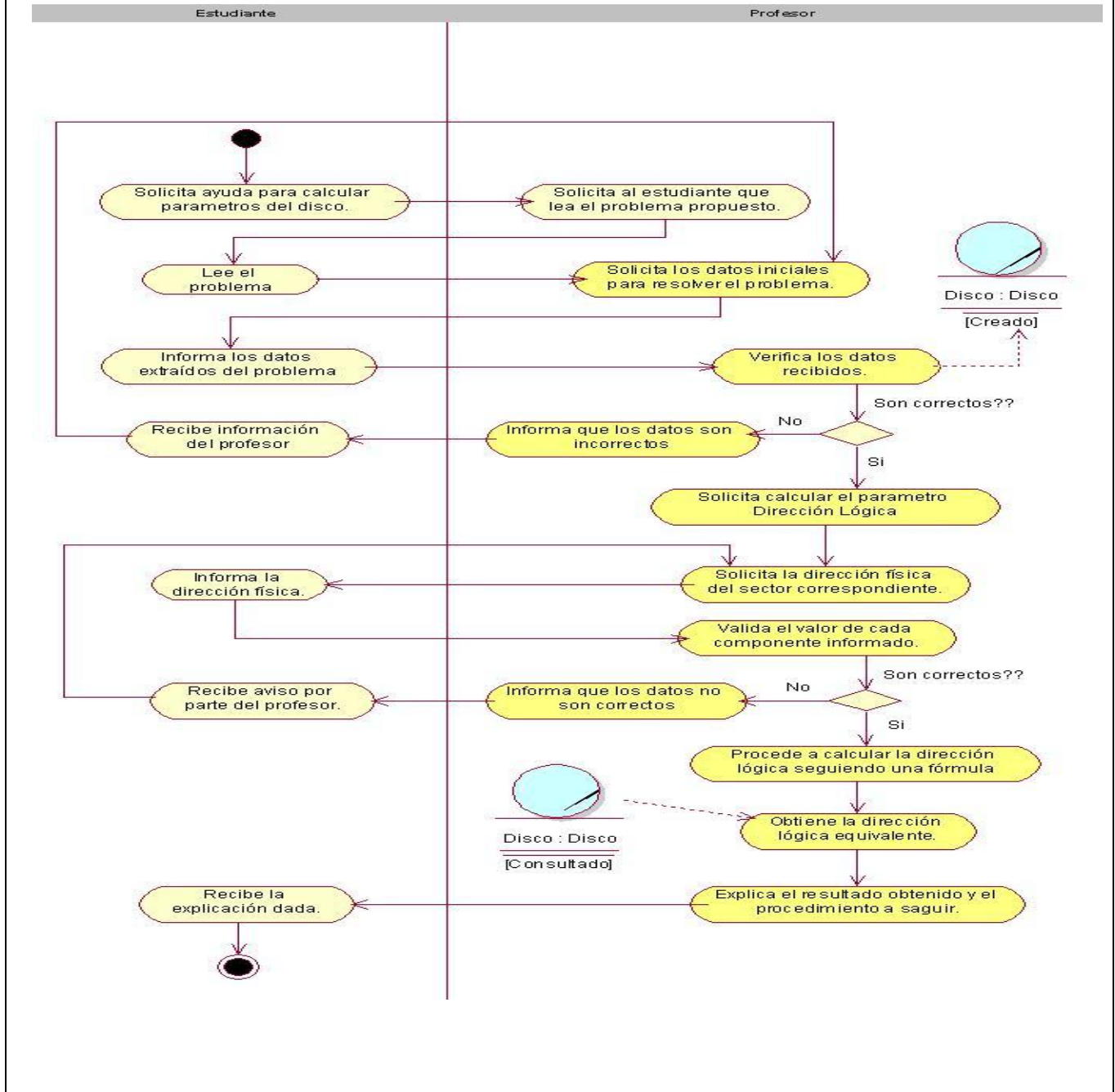
Anexo 12

**DIAGRAMA DE ACTIVIDAD DEL CU "CALCULAR
PARÁMETROS DEL DISCO DURO
SECCIÓN_CALCULAR DIRECCIÓN FÍSICA"**



Anexo 13

DIAGRAMA DE ACTIVIDAD DEL CU "CALCULAR PARÁMETROS DEL DISCO DURO SECCIÓN_CALCULAR DIRECCIÓN LÓGICA"



Anexo 14. Patrones aplicados a los Casos de Uso.

Nombre		Descripción
Procesos	Pequeño Grupo de Escritores	Propone crear un equipo balanceado, usando a las personas que tienen diferentes habilidades y personalidades, para que los equipos no entren en la rodada de mirar los casos de uso de la misma manera. Restringir el número de personas en un equipo de trabajo a solo dos o tres personas. Es más fácil para un equipo pequeño encontrarse, comunicarse para alcanzar un acuerdo general.
Estructurales	Completar una Única Meta	Cada caso de uso debe describir un logro de interés para el actor primario. Describe cada caso de uso dirigiéndose hacia una completa y bien definida meta. Seleccione y nombre la meta del actor primario que usted desea resaltar.
	El nombre revela la intención	Nombra los casos de uso utilizando un verbo activo o frase que represente la meta del actor primario. El nombre debe reflejar la intención del caso de uso y reflejar un único objetivo e intención que el actor está intentando lograr.
	Escenario más Fragmentos.	Escribe los eventos del flujo principal como un escenario simple sin considerar posibles fallos. Debajo ubica los fragmentos del flujo que muestran que condición alternativa puede ocurrir.
	Alternativas Exhaustivas, Integrales.	Captura todos los fallos y alternativas que deben ser manejados en el caso de uso. Una vez que tengas identificados todos los casos de uso y su flujo principal identifica tantas variaciones como puedas en el flujo principal. Captura todas las variaciones que quieras que el sistema maneje, a pesar de eso, se es selectivo.
	Adorno, Decoración.	Detalles no funcionales como reglas de negocio, bocetos de interfaces de usuarios, interfaces de protocolos externos, reglas de validación de datos y algunos asuntos sobresalientes deben ser añadidos al caso de uso en la sección suplementaria. Crea campos dentro de la plantilla del caso de uso que fuera del texto del escenario apoye la información auxiliar que es útil asociar con el caso de uso.
	Preciso y Legible	Escribe cada caso de uso lo suficientemente legible a fin de que los clientes se los lean, los evalúen y precisen lo suficiente a fin de que los implementadores entiendan qué están construyendo.

Anexo 15 Aplicación de Métricas Orientadas a Objetos al Diagrama de Casos de Uso del Sistema

Factores de Completitud	Métricas Asociadas	Módulo de PD y AM
Factor 6. ¿Se presenta una descripción resumida (descripción de alto nivel) de todos los casos de uso del negocio?	Métrica 6: Número de casos de uso que no tiene descripción resumida Umbral: < 10% Acción sugerida: Completar la descripción resumida del caso de uso	Total de Casos de Uso: 8 Número de casos de uso que no tiene descripción resumida: 0 Representa: 0%
Factor 7. ¿Están definidos todos los requisitos que justifican la funcionalidad del caso de uso?	Métrica 7: Número de requisitos omitidos por caso de uso Umbral < 10% Métrica 8: Número de casos de uso que tienen requisitos omitidos Umbral < 10% Acción sugerida: Revisar la lista de requisitos para determinar cuáles serán apoyados por cada caso de uso	Total de Requisitos: 37 Total de Casos de Uso: 8 Número de requisitos omitidos por caso de uso: 0 Representa: 0% Número de casos de uso que tienen requisitos omitidos: 0 Representa: 0%
Factor 8. ¿Existen requisitos que no han sido considerados en algún caso de uso?	Métrica 9: Número de requisitos que no son considerados en ningún caso de uso.	Total de Requisitos: 37 Número de requisitos que no son considerados en ningún caso de uso: 0 Representa: 0%
Factor 10. ¿Se presenta una descripción detallada (descripción extendida esencial) de todos los casos de uso del negocio?	Métrica 11: Número de casos de uso que no poseen una descripción extendida. Umbral < 20% Acción sugerida: Interactuar con el usuario para realizar la definición extendida del caso de uso que sea consistente con la definición a alto nivel.	Total de Casos de Uso: 8 Número de casos de uso que no poseen una descripción extendida: 0 Representa: 0%
Factores de Consistencia	Métricas Asociadas	Módulo de PD y AM

Factor 14. ¿El nombre dado a los casos de uso es una expresión verbal que describe alguna funcionalidad relevante en el contexto del usuario?	Métrica 16: Número de casos de uso que tienen un nombre incorrecto. Umbral < 20% Acción sugerida: Modifique el nombre del caso de uso de tal manera que signifique una acción desde el punto de vista del usuario.	Total de Casos de Uso: 8 Número de casos de uso que tienen un nombre incorrecto: 0 Representa: 0%
Factor 15. ¿Representa el caso de uso una interacción observable por un actor?	Métrica 17: Número de casos de uso que no representan una interacción observable por un actor. Umbral < 5% Acción sugerida: Elimine el caso de uso e incorpore su funcionalidad como una responsabilidad del sistema dentro de otro caso de uso.	Total de Casos de Uso: 8 Número de casos de uso que no representan una interacción observable por un actor: 0 Representa: 0%
Factor 19. ¿La descripción del flujo de eventos se inicia con la descripción de una acción externa originada por un actor o por una condición interna del sistema claramente identificable?	Métrica 23: Número de casos de uso cuya descripción extendida no inicia con una acción externa o con una condición monitoreada por el sistema. Umbral: < 10% Acción sugerida: Complete la definición del caso de uso incluyendo la acción fuera del sistema que da inicio al caso de uso o la condición interna que el sistema tiene controlar para dar inicio al caso de uso.	Total de Casos de Uso: 8 Número de casos de uso cuya descripción extendida no inicia con una acción externa o con una condición monitoreada por el sistema: 0 Representa: 0%
Factor 21. ¿Existe una adecuada separación entre el flujo básico de eventos y los flujos alternos y/o flujos subordinados?	Métrica 25: Número de casos de uso complejos que no tienen separación del flujo básico y de flujos alternos. Umbral: < 20% Acción sugerida: Estructure el caso de uso de manera que separe su	Total de Casos de Uso: 8 Número de casos de uso complejos que no tienen separación del flujo básico y de flujos alternos: 0 Representa: 0%

	funcionalidad básica (caso de uso base) de la funcionalidad repetitiva o alternativa. Si hay pasos repetitivos forme un caso de uso que lo incluya y los pasos alternativos formen un caso de uso que lo extienda.	
Factores de Correctitud	Métricas Asociadas	Módulo de PD y AM
Factor 23. ¿Representa el caso de uso requisitos comprensibles por el usuario?	Métrica 28: Número de casos de uso en que los requisitos representados no son comprensibles por el usuario. Umbral: < 5% Acción sugerida: Discuta con el usuario la interacción que describe el caso de uso y ajuste dicha descripción de manera que sea comprensible por el usuario.	Total de Casos de Uso: 8 Número de casos de uso en que los requisitos representados no son comprensibles por el usuario: 0 Representa: 0%
Factores de Complejidad	Métricas Asociadas	Módulo de PD y AM
Factor 29. ¿Los elementos dentro del diagrama están adecuadamente ubicados de manera que facilitan su interpretación?	Métrica 36: Número de elementos del diagrama que requieren reubicación. Umbral: < 30% Acción sugerida: Modifique la ubicación de los elementos del diagrama de manera que los elementos relacionados se encuentren lo más cercano posible.	Total de Casos de Uso: 8 Número de elementos del diagrama que requieren reubicación: 0 Representa: 0%

Anexo 16. Interfaz Principal.

