



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

*Análisis y Diseño del módulo  
Evaluación y Seguimiento  
del  
Laboratorio Virtual de Sistemas Operativos*

*Autor: Bárbara Giselle Santos Figarola  
Tutor(es): Ing. Daimi Lamorú Marciel  
Ing. Carlos Yasmany Hidalgo García*

*La Habana, Junio 2011  
"Año 53 de la Revolución"*

### *Declaración de Autoría*

Declaro ser autora de este trabajo y autorizo a la Facultad 3 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio. Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

Bárbara Giselle Santos Figarola

---

Firma del Autor

Carlos Yasmany Hidalgo García

---

Firma del Tutor

### *Agradecimientos*

*A:*

*Mis abuelas, que son lo más lindo que tengo en la vida.*

*Mi mamá, que aunque siempre lo ponga en duda, la quiero y la admiro como a nadie, y le agradezco una y cada una de las cosas que ha hecho por mí.*

*Mis tíos, por jugar a ser amigos, tíos y padres.*

*Arizandra, por ser la hermana que no tengo.*

*Walter, por devolverme la parte más importante de mi vida, por darme la posibilidad de ser yo en todos los sentidos.*

*Mercy, Dalía, Tiki y Rosana, por estar siempre, hasta cuando no les tocaba.*

*Las muñecas, por enseñarme, que a veces las cosas son demasiado buenas para ser verdad.*

*Carrasco, Villar y Cristina, por despertar tanta admiración en mí.*

*Todas y cada una de las personas que han estado ahí para mí.*

*Gracias*

### *Dedicatoria*

*A: Juan Claro Figarola Rodríguez*

*Por ser la única persona con la que siento que tengo una deuda enorme y al cual me aterraría defraudar.*

*Por ser además de abuelo, padre; y tratar de que mi infancia fuera lo más feliz y extraordinaria posible.*

*Por enseñarme las cosas que mejor he aprendido en mi vida y que de seguro son las más importantes.*

*Por dejarme sentir que el cariño no es sólo regalos.*

*Por demostrarme que en la vida no hay nada realmente imprescindible.*

*Por darme un motivo para sentir dolor, anhelo e inspiración para ser mejor y lograr lo que me proponga en la vida.*

*Por dejarme un millón de recuerdos que a pesar de resaltar el dolor, me hacen reír.*

*Por todo y más, es para ti y no para alguien más.*

### *Resumen*

Las tecnologías de la información y las comunicaciones juegan un papel fundamental en las distintas esferas de la sociedad y en la educación de la misma, aportando nuevos soportes para la información y el conocimiento además de las formas idóneas para su distribución. En este contexto tienen lugar los laboratorios virtuales, los cuales son herramientas didácticas utilizadas para la consolidación de conocimientos y habilidades, además de la realización de otras actividades formativas. Los laboratorios virtuales le brindan al estudiante la posibilidad de apreciar y entender de una manera más clara y concisa la ocurrencia de algunos procesos de difícil comprensión mediante la simulación de los mismos.

En la Universidad de las Ciencias Informáticas, como parte del programa docente se imparte la signatura Sistemas Operativos, la cual cuenta con contenidos muy complejos y de difícil comprensión, para ello se ha desarrollado un Laboratorio Virtual de Sistemas Operativos, permitiéndole al estudiante la consolidación de los conocimientos adquiridos, además apreciar la simulación los procesos estudiados en la asignatura. Sin embargo se dificulta el control y evaluación de las actividades de ejercitación que realiza el estudiantado pues no existe un mecanismo orientado a esta actividad, impidiéndole al profesor realizar una valoración oportuna del autoaprendizaje, impidiendo que le brinde algún tipo de ayuda en aras de resolver las deficiencias que puedan ser identificadas. Con el objetivo de resolver esta problemática en el Laboratorio Virtual de Sistemas Operativos, surge esta investigación, que tiene como propósito modelar las funcionalidades necesarias para la realización del seguimiento puntual y global del autoaprendizaje en el Laboratorio Virtual de Sistemas Operativos.

En función de alcanzar el objetivo propuesto en la presente investigación, fueron definidos el proceso general y los subprocesos correspondientes. Se hizo uso de algunas técnicas de la ingeniería de requisitos en el proceso de captura y validación de los requisitos, así como de la metodología RUP, los lenguajes UML y BPMN y Visual Paradigm como herramienta para el modelado. Se utilizaron también técnicas y métricas para la validación de los artefactos generados.

## Índice

Introducción .....	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	4
Introducción.....	4
1.1 Aprendizaje Electrónico (E-learning).....	4
1.2 Laboratorio Virtual .....	5
1.2.1 Estructura .....	7
1.2.2 Ventajas y Desventajas de los Laboratorios Virtuales .....	8
1.3 Metodologías de Desarrollo de Software .....	9
1.4 Lenguajes de Modelado .....	10
1.5 Herramientas CASE .....	11
1.6 Técnicas de Validación de Artefactos.....	12
1.6.1 Métricas de Calidad de la Especificación .....	13
1.6.2 Matriz de Trazabilidad.....	13
1.6.3 Prototipos no Funcionales de Interfaz de Usuario .....	14
1.6.4 Métricas de Calidad de Casos de Uso del Sistema .....	14
Conclusiones Parciales .....	15
CAPÍTULO 2: DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA .....	16
Introducción.....	16
2.1 Modelo Conceptual.....	16
2.2 Diagrama de Flujo de Procesos.....	17
2.3 Ingeniería de Requisitos .....	18
2.3.1 Especificación de Requisitos de Software.....	18
2.3.2 Requisitos Funcionales .....	19
2.3.3 Requisitos no Funcionales .....	20
2.3.5 Validación de Requisitos del Sistema.....	21
2.4 Diagrama de Casos de Uso del Sistema .....	23
2.4.1 Actores del sistema.....	24
2.4.2 Diagrama de Casos de Uso del Sistema.....	24
2.4.3 Patrones Utilizados .....	25
2.4.4 Descripción Textual de los Casos de Uso del Sistema.....	25
2.5 Análisis del Sistema .....	37
2.5.1 Diagrama de Clases del Análisis.....	37
2.5.2 Diagramas de Interacción .....	38

2.6 Diseño del Sistema.....	44
2.6.1 Diagrama de Clases del Diseño.....	45
2.6.2 Patrones Utilizados.....	45
Conclusiones Parciales.....	48
CAPÍTULO 3: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA.....	49
Introducción.....	49
3.1 Verificación de Requisitos del Sistema.....	49
3.2 Modelo de Métricas Orientada a Objetos aplicadas al Diagrama de Casos de Uso del Sistema.....	50
3.3 Prototipos no Funcionales.....	53
Conclusiones Parciales.....	54
Conclusiones Generales.....	55
Recomendaciones.....	56
Referencias Bibliográficas.....	57
Bibliografías.....	59

## Índice de Figuras

Figura 1. RUP. Fases e Iteraciones. ....	10
Figura 2. Modelo Conceptual. ....	16
Figura 3. Modelo de Procesos. Evaluación y Seguimiento. ....	17
Figura 4. Diagrama de Casos de Uso del Sistema. ....	24
Figura 5. Diagrama de Secuencia. Asignar Tareas. ....	38
Figura 6. Diagrama de Secuencia. Realizar Atención Individual. ....	39
Figura 7. Diagrama de Secuencia. Evaluar Aprendizaje. ....	40
Figura 8. Diagrama de Secuencia. Consultar Estado. ....	41
Figura 9. Diagrama de Secuencia. Consultar Evaluación. ....	42
Figura 10. Diagrama de Secuencia. Realizar Cortes Intermedios. ....	43
Figura 11. Diagrama de Secuencia. Modificar Evaluación Final. ....	44
Figura 12. Diagrama de Clases del Diseño. ....	45
Figura 13. Matriz de Trazabilidad (Requisitos Funcionales - Casos de Uso). ....	49
Figura 14. Matriz de Trazabilidad (Casos de Uso - Diagrama de Secuencia). ....	50



### Índice de Tablas

Tabla 1. Requisitos Funcionales. ....	19
Tabla 2. Requisitos no Funcionales. ....	21
Tabla 3. Actores del Sistema. ....	24
Tabla 4. Descripción Textual. Consultar Estado.....	27
Tabla 5. Descripción Textual. Consultar Evaluación. ....	28
Tabla 6. Descripción Textual. Realizar Atención Individual. ....	30
Tabla 7. Descripción Textual. Evaluar Aprendizaje. ....	32
Tabla 8. Descripción Textual. Modificar Evaluación Final.....	33
Tabla 9. Descripción Textual. Asignar Tareas.....	35
Tabla 10. Descripción Textual. Realizar Cortes Intermedios. ....	36
Tabla 11. Modelo de Métricas Orientas a Objetos.....	53

### **Introducción**

Las Tecnologías de la Información y la Comunicación también conocidas como TICs, son un conjunto de procesos y productos que permiten el almacenamiento, transformación y transmisión de un elevado volumen de información a gran velocidad.

Las TICs han tenido un significativo impacto en la sociedad y en la educación de la misma, aportando nuevos medios y soportes para la información y el conocimiento. Entre los medios aportados está el Entorno Virtual de Aprendizaje (EVA), que no es más que un medio didáctico provisto de información docente, útil para educadores y educandos, como protagonistas del proceso de enseñanza – aprendizaje, que les permite intercambiar sobre temas de interés y actualidad de manera online.

Cuba no se encuentra al margen de la evolución de la informática, demostrado con la creación de programas destinados a la formación y educación tecnológica, tales como los Joven Club de Computación y la universidad de nuevo tipo, surgida al calor de la batalla de ideas llamada Universidad de las Ciencias Informáticas (UCI). La UCI tiene como objetivos fundamentales la formación de ingenieros en ciencias informáticas e impulsar la industria del software cubano.

En la UCI como parte de la vinculación de las TICs al proceso docente educativo, se trabaja en la creación de medios didácticos que contribuyan a la formación del estudiantado de manera amena, flexible y atractiva. Con este objetivo fue desarrollado el Laboratorio Virtual de Sistemas Operativos, en apoyo como bien lo nombre lo indica, a la asignatura Sistemas Operativos, la cual cuenta con temas amplios y complejos.

Los laboratorios virtuales son herramientas didácticas utilizadas en la consolidación de conocimientos y habilidades. De manera general, están compuestos por los módulos Recursos, Administración, Simulación, Comunicación y Evaluación y Seguimiento. Éste último es de vital importancia para el funcionamiento y cumplimiento del objetivo primario de dicha herramienta, pues le permite al educador o profesor, supervisar la labor de los estudiantes en la aplicación, dándole la posibilidad de evaluar su proceso de autoaprendizaje, determinar si se cumplen o no los objetivos propuestos con cada tema tratado, realizar la orientación oportuna para la solución de determinadas dificultades en el aprendizaje,

además de utilizar la evaluación como motivación por el conocimiento y el esfuerzo por mejores resultados. Sin embargo, la idea de desarrollar un laboratorio virtual no tiene concebido un módulo de Evaluación y Seguimiento, por lo que el Laboratorio Virtual no cumple totalmente con su función, pues no permite al profesor realizar las actividades antes mencionadas, las cuales juegan un papel fundamental en el proceso de aprendizaje, pues dan una medida cuantitativa de la evolución del autoaprendizaje y los conocimientos adquiridos. Para la futura implementación e integración del módulo Evaluación y Seguimiento es preciso contar con los modelos de análisis y diseño que permitan una mejor comprensión del funcionamiento del módulo.

De la situación anteriormente descrita se deriva el **problema de investigación**: ¿Cómo garantizar los modelos de análisis y diseño necesarios para la implementación del módulo Evaluación y Seguimiento del Laboratorio Virtual de Sistemas Operativos? El problema planteado define como **objeto de estudio**: El proceso de desarrollo de software.

Se plantea como **objetivo general**: Modelar las funcionalidades del módulo Evaluación y Seguimiento del Laboratorio Virtual de Sistemas Operativos para su futura implementación; teniendo como **campo de acción**: El flujo de trabajo Análisis y Diseño del módulo Evaluación y Seguimiento del Laboratorio Virtual de Sistemas Operativos.

Del objetivo general anteriormente planteado se derivan los siguientes objetivos específicos:

1. Elaborar el marco teórico de la investigación.
2. Realizar los modelos de análisis y diseño.
3. Validar los artefactos obtenidos.

El Análisis y Diseño del módulo Evaluación y Seguimiento del Laboratorio Virtual de Sistemas Operativos permitirá que el desarrollador cuente con los modelos necesarios que le permitan el entendimiento del funcionamiento del módulo para su posterior implementación.

Con el objetivo de vencer satisfactoriamente los objetivos específicos se hará uso de la combinación de los métodos de investigación científica que a continuación se especifican:

Histórico – Lógico: permitirá hacer un análisis de la información más reciente respecto a los laboratorios virtuales, sus características y las tecnologías a utilizar en su desarrollo.

Analítico – Sintético: para lograr una síntesis de los aspectos más relevantes que distinguen un laboratorio virtual para realizar el diseño que satisfaga las aspiraciones existentes.

Modelación: facilitará el modelado de los procesos de difícil comprensión.

Entrevista: propiciará la obtención de toda la información referente y relevante para la investigación a realizar.

Para una mejor organización y comprensión de la presente investigación, la misma se encuentra estructurada de la siguiente manera:

El **Capítulo 1** hace referencia al estado del arte actual, haciendo énfasis en los principales conceptos relacionados con el tema, incluyendo elementos importantes como las metodologías, lenguajes de modelado, herramientas, además de métricas a utilizar den la futura verificación de la calidad de los artefactos obtenidos.

El **Capítulo 2** contiene el diagrama de flujo de proceso, diagrama de casos de uso, diagrama de secuencia, diagrama de clases del análisis, los diagramas de interacción, el diagrama de clases del diseño y las descripciones necesarias para una mejor comprensión de la aplicación que se desea diseñar.

El **Capítulo 3** muestra los parámetros que permiten la verificación y validación de los modelos de análisis y diseño con la utilización de métricas u otras técnicas.

### **CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA**

#### **Introducción**

En este capítulo se brindan las principales definiciones y conceptos acerca de los laboratorios virtuales y los tipos existentes. Se fundamentará el porqué del uso de la herramienta, metodología y lenguaje a utilizar. Se darán a conocer las técnicas de validación a utilizar en la validación de los artefactos generados.

#### **1.1 Aprendizaje Electrónico (E-learning)**

Con la vinculación de las TICs a la educación, aparecen nuevos soportes para la información y el conocimiento a la par de los medios necesarios para difundirlos; provocando un impacto importante en el aprendizaje y en el papel que históricamente jugó el profesor.

La modalidad de aprendizaje surgida con el uso de la tecnología en el proceso de formación, responde al nombre de e-learning (aprendizaje electrónico), que no es más que “el uso de tecnologías basadas en Internet, para proporcionar un amplio abanico de soluciones que aúnen adquisición de conocimiento y habilidades o capacidades” (Rosenberg, 2001). Según éste autor para determinar la realización de e-learning, es preciso tener en cuenta tres factores: la utilización de la red, que se utilicen estándares tecnológicos de Internet, y que se fundamente en soluciones al aprendizaje que rompan las barreras tradicionales de formación.

Al e-learning lo rodean cinco características esenciales, la separación física entre profesor y alumno, el uso masivo de medios técnicos, el alumno como centro de una formación independiente y flexible, el tutor y la comunicación de doble vía asíncrona. De manera general estas características hacen que el proceso de enseñanza-aprendizaje no dependa del vínculo presencial alumno-profesor; logrando desde cierto punto de vista la independencia del alumnado hacia el catedrático, que tradicionalmente protagonizaba y dirigía el proceso; permitiendo que se convierta en tutor y guía del mismo, logrando que su estudiantado siga el camino correcto hacia el aprendizaje haciendo uso de los recursos disponibles en la red, de manera lógica y medida. El uso amplio y adecuado de internet y otros medios técnicos por parte de los

educandos genera una mayor preparación tecnológica así como la adquisición de habilidades y capacidades en la gestión del aprendizaje.

El aprendizaje electrónico es de singular importancia, no solo en la educación a distancia y la semipresencial, sino que también influye en la formación tradicional pues le permite abrir sus horizontes alcanzando mayor productividad, capacitación flexible y entrega oportuna de los recursos y sistemas educativos.

El e-learning, sumado a otros términos derivados de la vinculación educación - tecnología, definen la modalidad de formación a distancia no presencial o semipresencial, sustentada en las TICs, permitiendo no sólo aprender, sino hacerlo de manera diferente obteniendo una adecuada formación didáctico-tecnológica.

### **1.2 Laboratorio Virtual**

Para la sociedad moderna, luego de haber adoptado el modelo de educación a distancia y la formación semipresencial, era necesario el empleo de medios flexibles, accesibles y adaptables. Un medio con el cual el estudiante ponga en práctica los conocimientos adquiridos y sea capaz de analizar y llegar a conclusiones para desarrollar su creatividad en la solución de ejercicios. Debido a las anteriores necesidades surgen los laboratorios virtuales, término del cual se han dado diferentes conceptos y definiciones. Para James P. Vary, laboratorio virtual es “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”. Por otra parte y tomando como referencia otras experiencias y consideraciones, los profesores de la Universidad Nacional de Educación a Distancia consideran que “un laboratorio virtual es la representación de un lugar dotado de los medios necesarios para realizar investigaciones, experimentos y trabajos de carácter científico o técnico, producido por un sistema informático, que da la sensación de su existencia real”.

Los laboratorios virtuales se encuentran clasificados en función de su utilización y la manera de acceder a sus recursos y funcionalidades, en cuyo caso se encuentran clasificados de la manera siguiente:

**Laboratorios virtuales web:** Este tipo de laboratorios se basa en un software que depende de los recursos de un servidor determinado. Estos recursos pueden ser determinadas bases de datos, software que requiere ejecutarse en su servidor, o la exigencia de determinado hardware para ejecutarse. No son programas que un usuario pueda descargar en su equipo para ejecutar de forma independiente. [1]

**Laboratorios Virtuales de Software:** Son laboratorios virtuales 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, M.S. 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.[1]

**Laboratorios Virtuales Remotos:** Se accede a través de Internet a un sistema físico real para su manipulación directa. El software utilizado para el control remoto puede ser un navegador web o una aplicación que necesita ser descargada del servidor del laboratorio. En algunas ocasiones puede que sea posible su visualización e incluso audición en tiempo real. [2] 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. [1]

De manera general, el tipo de laboratorio utilizado o desarrollado va acorde a la situación y contexto en que será aplicado, de igual forma brindan las suficientes ventajas como para ser adoptados como solución a disímiles problemas. En la educación, están asociados tanto a la enseñanza de la lectura, las matemáticas y los colores en edades tempranas, como en la enseñanza de la automática o carreras de perfil tecnológico, pues permiten simular con un gran nivel de realismo, prácticas que en la vida real serían muy costosas, y aún así son de vital importancia en la formación tecnológica. Su aceptación y mayor utilización está dada por la capacidad que poseen de permitir al usuario en la mayoría de los casos de gestionar sus conocimientos y la información necesaria para ella además de no contar con restricciones de tiempo y espacio, sólo de conocimientos mínimos de informática y de la materia tratada.

En Cuba, en la Universidad de las Ciencias Informáticas, existe un equipo de trabajo perteneciente al proyecto PROLAVI (Proyecto de Laboratorios Virtuales), que se dedica al desarrollo de laboratorios virtuales destinado al apoyo del aprendizaje de los estudiantes, obteniendo resultados satisfactorios aunque aún se trabaja en función de ampliar los horizontes funcionales de los mismos.

En la UCI, el grupo de trabajo antes mencionado no es el único que labora en función de apoyar y cooperar con el proceso de aprendizaje, también se desarrolló por parte de un equipo de la Facultad # 3, un Laboratorio Virtual de Sistemas Operativos, el mismo trabaja en función de propiciar una mejor comprensión de la asignatura, lo que conllevaría a un significativo aumento de los conocimientos y las habilidades, y por tanto de los resultados docentes de los estudiantes.

En resumen, son medios electrónicos de significativa importancia en la actualidad, con aplicación en diferentes esferas. Permiten generar nuevos espacios pedagógicos interactivos, donde se promueve la participación interactiva con los contenidos de cada laboratorio; facilitándose la construcción del conocimiento, así como el almacenamiento, transmisión, recuperación, aplicación y enriquecimiento de los contenidos.

### **1.2.1 Estructura**

Los laboratorios virtuales en general, en su estructura cuentan con módulos o subsistemas como Simulación, Comunicación, Administración, Recursos y Evaluación y Seguimiento. Cada módulo aporta una funcionalidad única al laboratorio; el subsistema Simulación es el encargado de simular todos los procesos definidos en correspondencia con los datos incorporados por el usuario, además de gestionar y garantizar que si hay más de una solicitud por usuarios remotos, no existan inconsistencias en el proceso particular de cada uno. Comunicación, como bien su nombre lo indica media la comunicación entre los usuarios remotos (si existen) y entre todos los módulos de manera general, logrando que la aplicación funcione como un todo único. Por su parte Administración y Recursos, se encargan de la gestión de permisos y de información respectivamente.

Como es obvio, la meta de la utilización de un laboratorio virtual no es otra que la mejora de los conocimientos del alumno sobre la materia en cuestión, y no puede lograrse sin que el estudiante reciba una realimentación sobre la validez del trabajo desarrollado dentro del laboratorio, así como que el



profesor conozca el grado de avance experimentado por el alumno. Es por ello que todo el laboratorio virtual no tendría sentido sin la existencia de un módulo de seguimiento y evaluación del alumno que le permitiese a éste autodirigirse en su proceso de aprendizaje, indicándole los puntos sobre los que necesita un mayor refuerzo en su estudio [3].

Al mismo tiempo, este módulo debe servir al profesor tanto para comprobar que el proceso de aprendizaje que lleva asociado el estudiante es el correcto (seguimiento puntual) como para conocer si se han logrado los objetivos perseguidos tras el estudio completo de la asignatura (seguimiento global). De acuerdo con esto, algunas de las tareas que debe llevar a cabo el módulo de seguimiento y evaluación son: a) generación y corrección de test sobre la materia, b) programación de experimentos para el refuerzo y fijación de determinados conceptos, c) evaluación del resultado de los experimentos y d) seguimiento continuo de las simulaciones y del estado del laboratorio [3].

Por un lado, el objetivo es el de ofrecer a los alumnos información precisa acerca de su nivel de asimilación de conocimientos y ayudarles a alcanzar un nivel adecuado de la materia. Por otro lado, permite a los profesores obtener información de la progresión a lo largo del periodo docente de un alumno o un grupo de alumnos (*autoevaluación continuada*), así como para la adaptación de los temarios y la recomendación de lecturas [4]. El sistema incluye un gestor de la base de datos de preguntas y el tratamiento cualitativo y cuantitativo de los resultados de las evaluaciones que proporcionan el perfil de conocimientos de un alumno o grupo de alumnos sobre una determinada materia.

### **1.2.2 Ventajas y Desventajas de los Laboratorios Virtuales**

Los laboratorios virtuales no son más que aproximación de la realidad en la que el estudiante tiene la posibilidad de observar con mayor claridad eventos de la vida real que pudieran resultar muy abstractos o de gran complejidad a la hora de determinar los procesos que ocurren en el mismo. Facilitan la búsqueda y selección de información al integrar las TICs al proceso educativo. Con su utilización el estudiante aprende a prueba y error, sin temor a un accidente ni a las críticas, puede repetir el ejercicio todas las veces que sean necesarias sin afectar en lo más mínimo ninguna otra actividad; y como factor muy importante no depende de tiempo ni espacio para realizar sus prácticas. Contribuye al aprendizaje colaborativo ya que permite la comunicación entre usuarios, haciendo posible la transmisión de experiencias y conocimientos.

La utilización de los laboratorios virtuales también trae consigo consecuencias no deseadas, pues el estudiante puede convertirse en solo un espectador más, por lo que es importante que las actividades estén diseñadas con el objetivo de que el estudiante realice una práctica progresiva y ordenada, para alcanzar los objetivos definidos. La no utilización de elementos reales puede incidir en la pérdida parcial de la visión de la realidad y por tanto resultados menos interesantes.

### **1.3 Metodologías de Desarrollo de Software**

Las metodologías de desarrollo de software son un compendio de procedimientos, técnicas y ayudas para la documentación del desarrollo de software, lo que permite estructurar, controlar y planificar el proceso y se extiende durante todo el ciclo de vida del mismo. No existe una metodología universal, cada proyecto o equipo de trabajo exige un proceso configurable. La metodología a utilizar en la presente investigación es RUP, partiendo del principio de que RUP (Proceso Unificado de Desarrollo), es un proceso para el desarrollo de un producto de software que define quién, cómo, cuándo y qué debe hacerse en el proyecto. Se caracteriza por estar dirigido por casos de uso, centrado en la arquitectura y ser iterativo e incremental. Su evolución está medida por fases e iteraciones, que comprenden las etapas por las que transita el producto hasta ser concluido.

El Proceso Unificado guía a los equipos de proyecto en cómo administrar el desarrollo iterativo de un modo controlado mientras se balancean los requerimientos del negocio, el tiempo al mercado y los riesgos del proyecto. El proceso describe los diversos pasos involucrados en la captura de los requerimientos y en el establecimiento de una guía arquitectónica lo más pronto posible, para diseñar y probar el sistema hecho de acuerdo a los requerimientos y a la arquitectura. El proceso describe qué entregables producir, cómo desarrollarlos y también provee patrones. El proceso unificado es soportado por herramientas que automatizan entre otras cosas, el modelado visual, la administración de cambios y las pruebas. Se caracteriza básicamente por ser vital la captura de requisitos, se necesita de un buen líder de proyecto para garantizar el trabajo del equipo de desarrollo, se realiza un gran número de artefactos lo que puede provocar retrasos por mala preparación de los analistas, las responsabilidades están divididas y es aplicable a todo tipo de proyecto asumiendo sus extensiones [5].

A diferencia de otras metodologías, RUP hace que el proceso sea práctico con bases de conocimiento y guías para la planificación del proyecto, es de gran ayuda para integrar rápidamente a los miembros del

equipo y poner en acción el proceso personalizado. Aporta un framework de proceso configurable que permite seleccionar e implantar los componentes específicos de proceso necesarios para lograr un proceso consistente y customizado para cada equipo y proyecto.

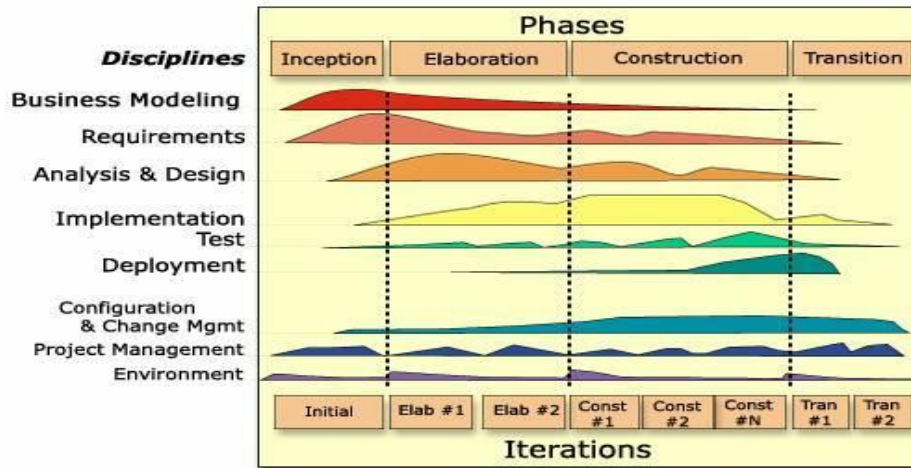


Figura 1. RUP. Fases e Iteraciones.

### 1.4 Lenguajes de Modelado

Para obtener una visión lo más cercana posible a un producto de software los lenguajes de modelado juegan un papel fundamental ya que permiten representar sistemas de todo tipo, de una manera lo más simple y comprensible posible para desarrolladores, analistas y demás integrantes del equipo de desarrollo. Lenguajes de modelado hay varios, entre ellos el Lenguaje Unificado de Modelado y la Notación de Modelado de Procesos del Negocio.

Por las particularidades y propiedades de UML y BPMN ambos lenguajes de modelado serán utilizados en el modelado de la solución propuesta. Por su parte UML, acrónimo de Unified Modeling Language (Lenguaje Unificado de Modelado) ofrece la posibilidad de construir los modelos que define la metodología escogida para desarrollar el software, además del estrecho vínculo en cuanto a objetivos que tienen RUP y UML, lo que permite expresar requisitos y representar todos sus detalles, además hace que se obtenga una documentación que es válida durante todo el ciclo de vida de un proyecto. Aporta ventajas tan significativas como mayor rigor en la especificación, validación y valoración de los modelos y generar el código a partir de los modelos y a la inversa (modelos a partir del código fuente). Por otro lado está BPMN, (Business Process Modeling Notación) es una notación gráfica para el modelado conceptual de

procesos de negocio, muy apropiado para modelar procesos de difícil comprensión que no admiten cambios ni modificaciones, también se modela su flujo de información completo sin omitir ninguno de sus pasos. BPMN fue diseñado para permitir a los modeladores y las herramientas de modelado un poco de flexibilidad a la hora de extender la notación básica y habilitar un contexto apropiado adicional acorde a una situación determinada.

### **1.5 Herramientas CASE**

CASE (Computer Aided Software Engineering / Ingeniería de Software Asistida por Computadoras) está compuesta por diferentes tipos de programas utilizados para ayudar a las actividades del proceso de software, como el análisis de requerimientos, el modelado de sistemas, la depuración y las pruebas. Las herramientas CASE también incluyen un generador de código que automáticamente genera el código fuente a partir del modelo del sistema. Entre las CASE se pueden citar Visual Paradigm, Rational Rose Enterprise Edition y la versión Rational Rose Enterprise Edition Suite 2003.

En este caso la herramienta a utilizar es visual Paradigm para UML, pues además de ajustarse a los estándares establecidos por la metodología RUP y permitir el modelado con los lenguajes seleccionados es una de las herramientas CASE más completas y fáciles de usar. Tiene soporte multiplataforma, además proporciona excelentes facilidades de interoperabilidad pues apoya la importación y exportación a XML de versiones 1.0, 1.2 y 2. Tiene también conexión con Rational Rose en sus archivos de proyecto (.MDL y .CAT) los mismos que además pueden ser cargados por Visual Paradigm para UML. Está apoyado por un conjunto de lenguajes tanto en la generación de código como en la ingeniería inversa, por citar algunos como Java, C++, CORBA IDL, PHP, XML, SChema, Ada y Pyton. Se integra con herramientas Java como Eclipse / IBM, Web Sphen, JBuilder, NetBeans IDE, etc. Fue diseñada para usuarios interesados en sistemas de software de gran escala con el uso del acercamiento orientado a objeto, además apoya los estándares más recientes de las notaciones de Java, UML y BPMN. Posee soporte para trabajo en equipo, lo que permite que varios desarrolladores trabajen a la vez en el mismo diagrama y vean en tiempo real los cambios hechos por sus compañeros. Su diseño se centra en casos de uso y se enfoca al negocio que genera un software de mayor calidad, y se puede contar con la versión libre Visual Paradigm for UML Community Edition que soporta la realización de todos los diagramas

necesarios. Es una de las pocas herramientas CASE que hace análisis textual, siendo esta característica muy útil en la captura de requisitos. Soporta aplicaciones web y es fácil de instalar y actualizar [6].

Además ofrece funcionalidades como:

1. Modelado colaborativo con CVS (Concurrent Versions System) y Subversión.
2. Ingeniería inversa, código a modelo, código a diagrama.
3. Editor de Detalles de Casos de Uso, entorno todo en uno para la especificación de los detalles de los casos de uso, incluyendo la especificación del modelo general y de las descripciones de los casos de uso.
4. Diagramas de flujo de datos.
5. Generación de bases de datos, transformación de diagramas de Entidad-Relación en tablas de base de datos.
6. Ingeniería inversa de bases de datos, desde Sistemas Gestores de Bases de Datos existentes a diagramas de Entidad-Relación.
7. Generador de informes para generación de documentación.
8. Distribución automática de diagramas. Reorganización de las figuras y conectores de los diagramas UML.
9. Modelo para realizar prototipos de interfaz [6].

### **1.6 Técnicas de Validación de Artefactos**

La validación goza de singular importancia en el proceso de desarrollo de software, pues permite que de manera paulatina los desarrolladores tengan una idea de la calidad del trabajo realizado. Por lo general, para analistas y diseñadores es de vital importancia este proceso, ya que de ellos depende que a las siguientes fases de desarrollo se llegue con las funcionalidades específicas que desea el cliente además de un diseño que haga posible que el sistema a implementar sea consistente, óptimo y sencillo.

La ingeniería de software tiene definida varias métricas<sup>1</sup>, que permiten tanto la validación de los modelos del análisis como del diseño, entre ellas es válido destacar las métricas de la calidad de la especificación; definida para el análisis, y las métricas del modelo de diseño, que se desglosan en métricas de diseño de alto nivel, de interfaz, entre otras. El diseño también puede ser validado con la utilización de prototipos no funcionales de interfaz, haciéndole una aproximación al cliente de cómo puede ser la aplicación una vez terminada, esclareciendo que estos pueden tener cambios en la versión final del producto.

### **1.6.1 Métricas de Calidad de la Especificación**

Es una métrica que tiene como objetivo principal medir la calidad del modelo de análisis y su correspondiente especificación de requisitos. Facilita la determinación del grado de especificidad de los requisitos además de saber si están o no bien expresados. Evalúa la relación entre los requisitos igualmente interpretados y los requisitos totales; verifica la consistencia y el nivel de completitud de los requisitos funcionales, y define si hay o no conflictos entre requisitos. Pretende determinar el número de requisitos funcionales que se han especificado, así como el realismo y la verificabilidad de los requisitos.

### **1.6.2 Matriz de Trazabilidad**

En el proceso de desarrollo de software es de vital importancia la verificación de los requisitos funcionales, es decir, es necesario verificar su estado durante todo el ciclo de vida del producto, si están en proceso de ser incluidos, si ya fueron incluidos o en qué momento lo serán. Una de las mejores prácticas para realizar esta verificación o al menos una de las más utilizadas y efectivas, es la matriz de trazabilidad. Para que esta práctica pueda ser aplicada es preciso que los requisitos sean trazables, en otras palabras, que pueda verificarse que parte del producto tiene relación con un requisito determinado.

La matriz de trazabilidad es un formato que relaciona los requisitos con sus orígenes y lo sigue de cerca hasta la culminación del desarrollo. Permite asegurar que cada requisito es importante al relacionarlo con los objetivos de proyecto y provee una estructura para gestionar los cambios en el alcance del producto. Consiste en establecer una relación entre los requisitos y los casos de uso, que permita determinar en qué caso de uso se incluyen.

---

<sup>1</sup>Medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo dado. Incluye el método de medición.

### **1.6.3 Prototipos no Funcionales de Interfaz de Usuario**

El prototipo no-funcional de la interfaz de usuario está compuesto por el diagrama jerárquico de pantallas, en el cual se muestran las diferentes interfaces a través de las cuales el usuario podrá acceder a las funcionalidades que ofrece la aplicación. El diseño de este prototipo sólo muestra las formas y colores que presentarán las pantallas de la interfaz de la aplicación, sin incluir funcionalidad. Este tipo de prototipo se utiliza para validar con el usuario el estilo de las pantallas de interfaz que presentará la aplicación [7].

Los prototipos no funcionales de interfaz de usuario juegan un papel fundamental en el proceso de desarrollo de software, ya que aportan una manera efectiva y sencilla de comprobar si el diseño propuesto de las interfaces para cada una de las funcionalidades del sistema está acorde con lo que desea el cliente o el usuario final, por tales motivos constituyen otra forma de validar tanto los requisitos como el diseño del sistema. Los mismos están integrados a las descripciones textuales de casos de uso aportándole elementos visuales de significativa importancia.

Permiten mostrar las interacciones usuario- sistema, además de la información a mostrar y su ubicación, permitiendo realizar todos los cambios pertinentes antes de la implementación y corregir posibles malas interpretaciones. Con esta técnica los analistas solo proponen el diseño de la solución, y el cliente o usuario final debe ser capaz de entender y comprender la propuesta, y validando de esta forma el trabajo realizado.

### **1.6.4 Métricas de Calidad de Casos de Uso del Sistema**

Las métricas de calidad para casos de uso, le permiten al desarrollador o al equipo de desarrollo tener una medida del grado de Completitud, Correctitud, Consistencia y Complejidad que tienen los casos de uso del sistema.

- Completitud: Grado en el que se detallan los casos de uso relevantes.
- Consistencia: Grado en que los casos de uso describen las interacciones de los usuarios y el sistema.
- Correctitud: Grado en que las interacciones de los usuarios con el sistema soportan apropiadamente el modelo de negocio.

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

La aplicación de las métricas de calidad a los casos de uso del sistema constituye otra manera de verificar que los requisitos funcionales identificados y los casos de uso están debidamente descritos y relacionados, además de percibir tempranamente ciertos y determinados problemas asociados a los factores medibles antes mencionados.

### **Conclusiones Parciales**

En este capítulo se abordaron temas que permitieron realizar una síntesis de las características y aspectos esenciales que resultarán en un importante aporte a solución a proponer dando cumplimiento al primero de objetivos específicos trazados.

- Con el estudio realizado acerca de los laboratorios virtuales y su composición, haciendo énfasis en el módulo Evaluación y Seguimiento, se determinaron los aspectos fundamentales para proceder al modelado de las funcionalidades de dicho módulo.
- La utilización de la metodología RUP, los lenguajes de modelado UML y BPMN y Visual Paradigm for UML en su versión 7.1, permitirán generar los artefactos bajo los estándares establecidos en el desarrollo de software.
- Fueron analizadas las técnicas de validación que serán utilizadas para la validación de los artefactos midiendo de ésta forma la calidad de los mismos.



## CAPÍTULO 2: DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

### Introducción

En el presente capítulo se ofrece una descripción detallada de la solución propuesta con la utilización de los modelos y diagramas que tributen a una comprensión bastante aproximada del funcionamiento de la solución propuesta.

### 2.1 Modelo Conceptual

El modelo conceptual es una representación de conceptos u objetos en el dominio del problema y las relaciones entre ellos, con los cuales se describe cada parte del proceso real. Tiene como objetivo lograr una mejor comprensión del problema tratado. En este caso, el modelo responde al proceso Evaluación y Seguimiento, en cual se detallan las acciones que se realizan con la intervención de los objetos involucrados.

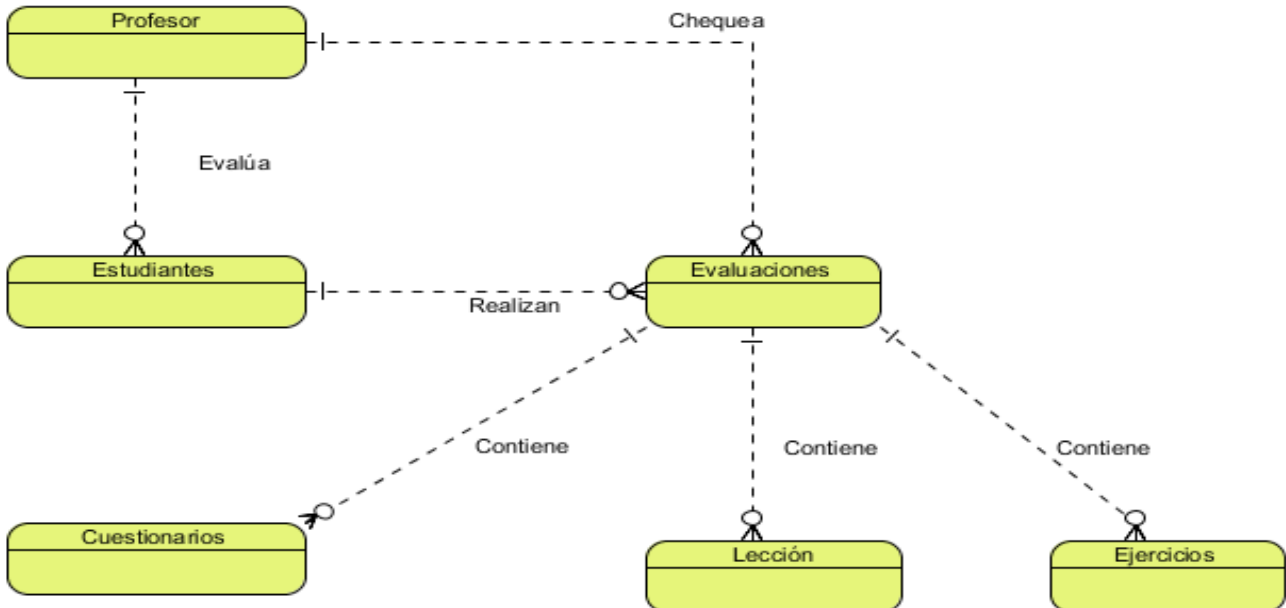


Figura 2. Modelo Conceptual.

### 2.2 Diagrama de Flujo de Procesos

Un proceso no es más que un conjunto de actividades (con sus respectivas entradas y salidas) y sus relaciones que le permiten al equipo de desarrollo lograr comprender y describir el funcionamiento de un determinado sistema o parte del mismo.

Un modelo de proceso de negocio describe cómo funciona el negocio, indicando las actividades involucradas en el mismo y la relación existente entre ellas. El modelado de procesos del negocio es empleado para la captura, y documentación y rediseño de procesos del negocio [8].

El proceso Evaluación y Seguimiento es de vital importancia en el proceso de formación tanto para el estudiante como para el profesor, como se explicó en el capítulo anterior. Para esta solución, el proceso tiene características distintas a las definidas por otros módulos de igual nombre y objetivo, pues separa los tipos de evaluación de la evaluación en sí, pues esta vez solo se dedica a establecer valoraciones cuantitativas de la actividad desarrollada por los estudiantes en la aplicación, además de planificar y orientar actividades en función de la consolidación de uno o varios temas teniendo en cuenta los resultados obtenidos. De manera general y con elementos más representativos el siguiente modelo de procesos describe las acciones y actividades que contempla el proceso Evaluación y Seguimiento.

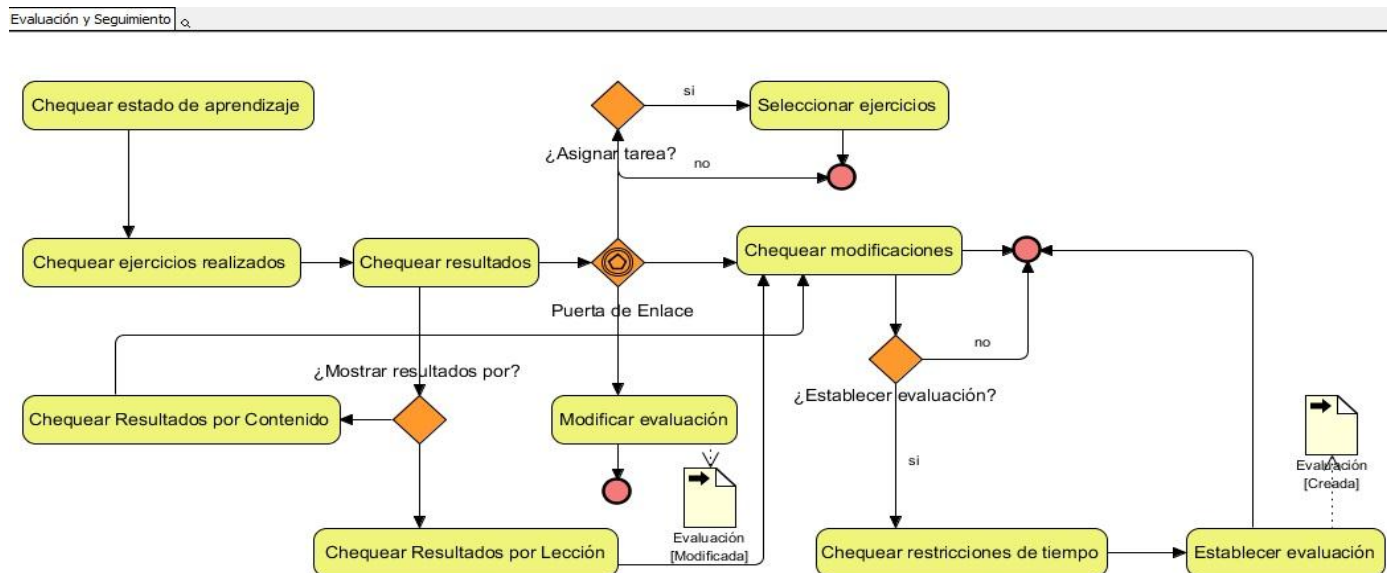


Figura 3. Modelo de Procesos. Evaluación y Seguimiento.

### **2.3 Ingeniería de Requisitos**

Para el desarrollo de un producto de software y la conformidad del cliente con el sistema que se le entrega, es de vital importancia que el sistema sufrague sus expectativas y necesidades. La única garantía que tiene el equipo de desarrolladores de lograr esto, es haciendo una eficiente gestión de requisitos. Es válido también resaltar que los requisitos expresan las necesidades y restricciones atribuibles a un producto de software que contribuye a la solución de algún problema del mundo real [9]. De manera general los requisitos se pueden clasificar en funcionales, que son operaciones o funciones que debe ser capaz de realizar el sistema y los no funcionales, que no son más que las condiciones o propiedades que éste debe tener, son los encargados de imponer las restricciones con las que deben ser implementados los requisitos funcionales.

Ingeniería de Requisitos (IR) o Ingeniería de Requerimientos, se encarga de descubrir, documentar y mantener una serie de requerimientos para un sistema basado en computadora. Según el tamaño del proyecto y el tipo de sistema a desarrollar esta disciplina cambia tanto el número de actividades o procesos como sus nombres. El hecho de ser una “ingeniería” implica que se utilicen varias técnicas sistemáticas y repetibles con el objetivo de corroborar que los requerimientos sean completos, consistentes y relevantes.

Como parte de la IR, en el proceso captura de los mismos, se utilizó la técnica Elicitación, para la cual se establecen varias técnicas, en este caso se desarrolla a partir de una entrevista con el cliente, es el proceso mediante el cual el equipo de desarrollo comienza a dar los primeros pasos con el objetivo de definir que quiere el cliente y cómo lo quiere.

Una vez definidos los requisitos del sistema, los mismos fueron analizados y transformados en requisitos formales, dando paso al proceso de especificación de requisitos del sistema.

#### **2.3.1 Especificación de Requisitos de Software**

La especificación de requisitos es la etapa de la IR que se dedica a la definición de qué debe hacer el sistema y cómo, además de la descripción de cada uno de los requisitos identificados. A continuación se muestran los requisitos funcionales y no funcionales del sistema y sus respectivas descripciones.

### 2.3.2 Requisitos Funcionales

Referencias	Requisitos Funcionales
RF01	Chequear Resultados. Muestra un reporte de todas las evaluaciones de los ejercicios y cuestionarios.
RF02	Chequear Modificaciones. Muestra un reporte con los cambios recientes en los resultados obtenidos por los estudiantes en los ejercicios resueltos.
RF03	Chequear Estado de Aprendizaje. Muestra un reporte del último contenido visitado por el estudiante y su recorrido en el mismo.
RF04	Chequear Evaluaciones por Contenido. Muestra un reporte de todas las evaluaciones del estudiante en cada contenido.
RF05	Chequear Ejercicios Realizados. Muestra un reporte de los ejercicios realizados con la información siguiente: Lección y Contenido al que pertenece, calificación, si subió nota o no y si cumplió el tiempo establecido.
RF06	Chequear Restricciones de Tiempo. Indica si se cumplió el tiempo establecido para realizar la lección.
RF07	Establecer Evaluaciones. Permite asignar una evaluación.
RF08	Chequear Registro de Evaluaciones. Muestra un reporte de las evaluaciones del estudiante de los contenidos, las lecciones y los cortes.
RF09	Establecer Cambio de Evaluaciones. Permite modificar una nueva evaluación.
RF10	Chequear Evaluaciones por lección. Muestra un reporte de las evaluaciones obtenidas en cada lección.
RF11	Enviar Mensaje de Notificación. Envía una notificación al estudiante informándole la nota que obtuvo en el tema o en el corte.

Tabla 1. Requisitos Funcionales.

## Descripción de la Solución Propuesta

### 2.3.3 Requisitos no Funcionales

Clasificación	Requisitos
<b>Apariencia o Interfaz Externa:</b>	
RNF1	La interfaz debe ser agradable al usuario con una estructuración de fácil comprensión para el usuario.
RNF2	La interfaz externa debe ser capaz de guiar al usuario en sus acciones con elementos lo más representativos posible
RNF3	Los colores utilizados en la aplicación deben tener una combinación que resulte agradable a la vista del usuario.
RNF4	El vínculo a la ayuda del sistema debe estar visible y disponible en la ventana principal de la aplicación.
RNF5	La aplicación debe utilizar principalmente el idioma español, exceptuando las palabras técnicas, que al ser traducidas modifiquen su significado.
<b>Usabilidad</b>	
RNF6	Los usuarios que accedan a la aplicación deben tener un mínimo de experiencia en el trabajo con aplicaciones de este tipo.
RNF7	Los usuarios deben tener conocimientos básicos del funcionamiento de un sistema operativo.
<b>Rendimiento</b>	
RNF8	El tiempo de respuesta de la aplicación no debe exceder los 9 segundos a partir de que se invoque alguna funcionalidad.
RNF9	Los usuarios deben tener la posibilidad de acceder a la aplicación las 24 horas del día.
<b>Portabilidad</b>	
RNF10	La aplicación debe poder ejecutarse tanto en Windows como en Linux.
<b>Seguridad</b>	
RNF11	La aplicación debe estar diseñada de tal manera que cada usuario tenga acceso solamente a la información que le corresponde.

## Descripción de la Solución Propuesta

Hardware	
RNF12	Tarjeta de memoria RAM de 128 MB o superior.
RNF13	Procesador Pentium II o superior a 250 MHz como mínimo.
RNF14	Computadora cliente de 40 Gb de disco duro o superior.
Software	
RNF15	Para acceder a la aplicación debe estar instalado al menos un navegador, puede ser Internet Explorer, Mozilla Firefox o Google Chrome.

Tabla 2. Requisitos no Funcionales.

### 2.3.5 Validación de Requisitos del Sistema

Luego de la definición de los requisitos, es imprescindible que estos sean validados, con el objetivo de determinar si reflejan las necesidades del cliente, además de asegurar que el análisis realizado y los resultados están en correspondencia y son correctos.

Las métricas de calidad de especificación de los requisitos permiten determinar el estado de características medibles y circunstanciales en la calidad del análisis, como son la consistencia tanto externa como interna, compleción, comprensión, entre otras. De la aplicación de las métricas y la validación, se ofrecen los detalles a continuación:

Primeramente se determinaron los valores de las variables que participan fundamentalmente en los cálculos a realizar:

$R_f$ : Número de requisitos funcionales.

$R_{nf}$ : Número de requisitos no funcionales.

$R_t$ : Número total de requisitos.

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

$$R_t = 13 + 15$$

$$R_t = 28$$

Los parámetros a medir son Especificidad, Consistencia Interna y Externa y Estabilidad.

## Descripción de la Solución Propuesta

### ➤ Especificidad.

Para determinar la especificidad o ausencia de ambigüedad de los requisitos, se hizo uso de la métrica basada en la consistencia de la interpretación de los revisores para cada requisito:  $Q1 = \frac{R_{ii}}{R_t}$

Donde  $R_{ii}$  es el número de requisitos para los que los entrevistados tuvieron interpretaciones idénticas,  $R_t$  es el número total de requisitos, la relación entre ellos arroja un valor asociado a  $Q1$ , cuyo valor se utiliza para indicar el nivel de ambigüedad de los requisitos, mientras más cerca esté el valor de  $Q1$  de 1 menos ambigüedad existirá en la especificación. Realizando la sustitución de variables por sus valores correspondientes se obtiene que:

$$Q1 = \frac{27}{28} = 0.96$$

### ➤ Consistencia Interna:

Esta medida se determinó con la relación  $Q2 = \frac{(R_e - R_{ce})}{R_e}$  con  $R_c$  como número de requisitos especificados y  $R_{ce}$ , número de requisitos con conflictos en la especificación. El valor de  $Q2$  se encuentra entre 0 y 1, y el resultado más conveniente de esta métrica es el más aproximado a 1, y expresa que no existen subconjuntos de requisitos contradictorios.

Según los revisores:  $R_c = 11$  y  $R_{ce} = 0$ ;

$$Q2 = \frac{11-0}{11} = 1.$$

### ➤ Consistencia Externa:

La consistencia externa se halló con la relación  $Q3 = \frac{R_{cd}}{R_t}$  con  $R_{cd}$  como número de requisitos consistentes en otros documentos y  $R_t$  como total de requisitos. El valor de  $Q3$  siempre está entre 0 y 1, y su valor óptimo es el más cercano a 1, expresando que los requisitos del software no están en contradicción con los requisitos del sistema. Con  $R_{cd} = 11$  y  $R_t = 11$

$$Q3 = \frac{11}{11} = 1.$$

### ➤ Estabilidad

La estabilidad fue definida con la relación  $E = \frac{R_t - R_m}{R_t}$  donde  $R_m$  son los requisitos modificados, y  $R_t$  que es equivalente al número de requisitos. Se considera como valor óptimo para esta métrica el valor más próximo a 1. Luego de la sustitución correspondiente:

## *Descripción de la Solución Propuesta*

---

---

$$E = \frac{11-1}{11} = 0.91.$$

De manera general los valores obtenidos para cada uno de los factores, es recomendable que para que un valor sea clasificado como óptimo éste se encuentre lo más cercano a 1 posible, en este caso particular quedó establecido que cada uno de estos factores se clasificara como óptimo a partir del valor 0.90. En el caso de la estabilidad se estipuló lo siguiente:

Alta: ( $0.90 \leq E \leq 1$ ); Media: ( $0.80 \leq E < 0.90$ ) y Baja: ( $0.7 \leq E < 0.80$ ).

Según los criterios establecidos por las métricas y las consideraciones particulares para determinar la calidad de la especificación de requisitos realizada, se obtuvo como resultado que la misma cumple satisfactoriamente con las características deseables para cualquier especificación de requisitos de un sistema, pues es concisa, entendible, correcta, pues uno y cada uno de los requisitos resuelve un problema real de manera concreta, Es trazable, realizable y sobre todo no cuenta con conflictos por lo que además posee buena consistencia interna.

Luego de analizados los resultados obtenidos y la repercusión que podían tener los mismos en la calidad del sistema que se diseña, se realizaron los siguientes ajustes:

- Se suprimió el requisito funcional Chequear Registro de Evaluaciones.
- Se cambió el requisito Chequear Evaluaciones por Chequear Resultados, pues el primero de los mencionados surgió de una interpretación errónea del diagrama de procesos.

### **2.4 Diagrama de Casos de Uso del Sistema**

El diagrama de casos de uso del sistema es un modelo que hace posible establecer la interacción del sistema con los usuarios, indicando que acciones puede realizar cada uno. Para la realización del mismo se tuvo en cuenta la situación particular a modelar y los patrones que podrían ajustarse a la misma.



### 2.4.1 Actores del sistema

Actor	Descripción
Estudiante	Es quien realiza la acción Consultar Evaluación.
Profesor	Realiza las acciones Evaluar Aprendizaje, Consultar Estado, Realizar Cortes Intermedios, Asignar Tareas, Modificar Evaluación y Realizar Atención Individual.

Tabla 3. Actores del Sistema.

### 2.4.2 Diagrama de Casos de Uso del Sistema

El Diagrama de Casos de uso del Sistema, es un modelo de las funciones deseadas para el sistema y su entorno, y sirve como contrato entre el cliente y los desarrolladores. Se utiliza como entrada esencial para las actividades de análisis y diseño [10].

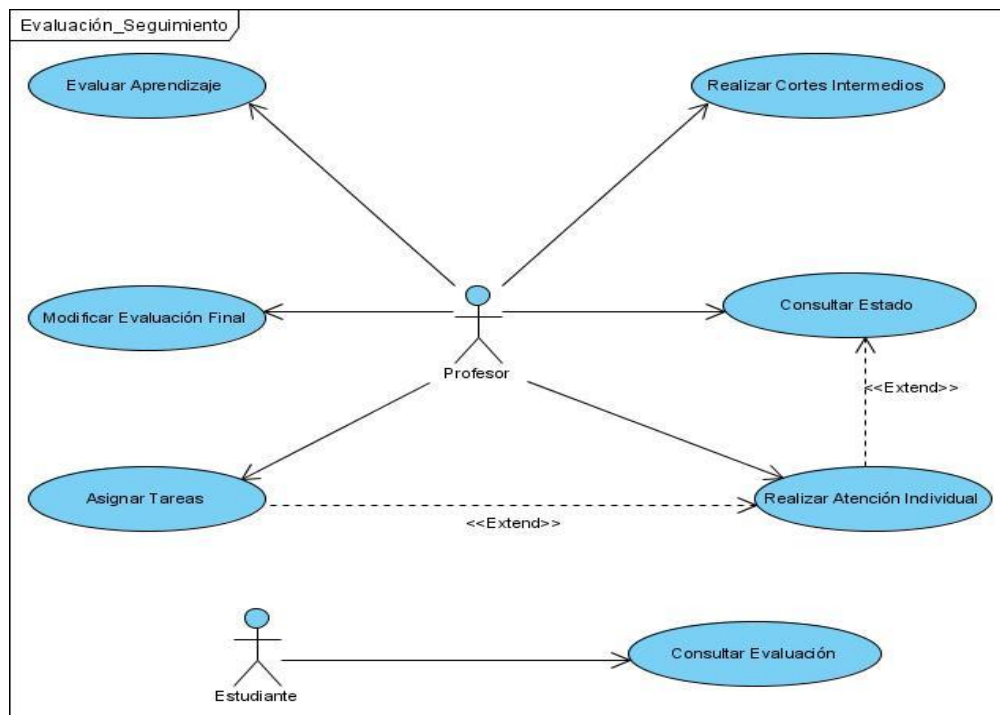


Figura 4. Diagrama de Casos de Uso del Sistema.

### 2.4.3 Patrones Utilizados

En la realización del Diagrama de Casos de Uso se tuvo en cuenta la utilización de patrones de Casos de Uso, los cuales van asociados a la asignación de responsabilidades y a la relación medida entre ellos. En este caso fueron utilizados los patrones Extensión Concreta y El nombre revela la intención.

**Extensión concreta:** Consiste en la relación de un caso uso base y uno extendido, donde el extendido, que debe ser concreto, se puede realizar de manera independiente aunque también forma parte del caso de uso base que puede ser abstracto o concreto.

**El nombre revela la intención:** Es un patrón muy sencillo, aunque complejo en la utilización, pues no es más que un verbo de acción, acompañado de una frase que indique la meta del actor, y de conjunto describan claramente en qué consiste el caso de uso.

### 2.4.4 Descripción Textual de los Casos de Uso del Sistema


Los diagramas de casos de uso son artefactos que se obtienen durante el proceso de desarrollo de software, y la descripción de los mismos sirve para enunciar paso a paso el flujo de eventos de cada uno. Permiten además agrupar los escenarios de usabilidad en un sistema, facilitan la trazabilidad de los requisitos además de artefactos como clases, objetos y código, y distribuir el proceso en piezas reutilizables con el uso de las relaciones que se pueden establecer entre ellos.

Descripción textual de “Consultar Estado”	
<b>Caso de uso:</b>	Consultar Estado
<b>Actores:</b>	Profesor
<b>Resumen:</b>	El caso de uso se inicia cuando el usuario selecciona la opción “Consultar Estado” en el menú principal, luego realiza una secuencia de actividades que le permiten apreciar las evaluaciones de los estudiantes en uno o varios contenidos.
<b>Precondiciones:</b>	
<b>Post condiciones:</b>	
<b>Casos de usos asociados:</b>	Atención Individual y Asignar Tareas.

## Descripción de la Solución Propuesta

<b>Referencia:</b>	RF01, RF02, RF03
<b>Prioridad:</b>	Normal
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El caso de uso comienza cuando el usuario selecciona la opción Consultar Estado.	2. Chequear Estado de Aprendizaje. 3. Chequear Resultados. 4. Chequear Modificaciones. 5. Muestra un resumen de la información, y las opciones: "Aceptar" "Asignar Tareas" "Realizar Atención Individual"
6. Selecciona la opción "Aceptar" y termina el caso de uso.	
<b>Flujo Alternativo # 1 Asignar Tareas.</b> <b>Ver Descripción Textual. Asignar Tareas.</b>	
<b>Flujo Alternativo #2. Realizar Atención Individual</b> <b>Ver Descripción Textual. Realizar Atención Individual.</b>	
<b>Prototipo no Funcional</b>	

## Descripción de la Solución Propuesta



**Profesor**

Página Principal - Evaluaciones - Consultar Estado

**Estudiante:**  
Bárbara Giselle Santos Figarola  
grupo: 5305  
Online: No

Nombre o usuario



**Buscar Estudiante**

Usuario

Herramientas

Chat

Usuarios en Línea

 Salir


Contenido 1

Contenido 2

Contenido 3

✔ Atención Individual

✔ Asignar Tareas

Contenido Actual		
Ejercicios y Cuestionarios	Nota	Cambios de Nota

Tabla 4. Descripción Textual. Consultar Estado.

Descripción textual “Consultar Evaluación”	
<b>Caso de uso:</b>	Consultar Evaluación
<b>Actores:</b>	Estudiante
<b>Resumen:</b>	El caso de uso se inicia cuando el usuario selecciona la opción Consultar Evaluación, el mismo le da la posibilidad de conocer sus evaluaciones en uno o varios contenidos.
<b>Precondiciones:</b>	El usuario tiene que haber accedido a la aplicación.
<b>Post condiciones:</b>	El usuario podrá ver el estado de sus evaluaciones.
<b>Casos de usos asociados:</b>	-
<b>Referencia:</b>	RF01y RF03.

## Descripción de la Solución Propuesta

<b>Prioridad:</b>	Normal
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El caso de uso comienza cuando el usuario selecciona la opción Consultar Evaluación.	2. Chequear Estado de Aprendizaje. 3. Chequear Resultados. 4. Muestra un resumen de las evaluaciones, y las opciones "Aceptar" y "Cancelar"
5. Selecciona la opción Aceptar para terminar la acción.	
<b>Prototipo no Funcional</b>	
<p>The screenshot shows a web page titled 'Página Principal - Evaluaciones - Consultar Evaluación'. On the left, there is a user profile for 'Estudiante' (Barbara Giselle Santos Figarola, grupo: 5305, Online: No) with search and navigation buttons. On the right, there is a table with columns 'Ejercicios y Cuestionarios', 'Nota', and 'Cambios de Nota'. Above the table, there are buttons for 'Contenido 1', 'Contenido 2', and 'Contenido 3'.</p>	

Tabla 5. Descripción Textual. Consultar Evaluación.

## *Descripción de la Solución Propuesta*

<b>Descripción textual “Realizar Atención Individual”</b>	
<b>Caso de uso:</b>	Realizar Atención Individual
<b>Actores:</b>	Profesor
<b>Resumen:</b>	El caso de uso se inicia cuando el usuario selecciona la opción “Realizar Atención Individual” en el menú principal, en el mismo el usuario puede consultar o ver las evaluaciones de los estudiantes, con esta información puede o no indicar una colección de ejercicios de ejercitar algún contenido o erradicar alguna deficiencia detectada.
<b>Precondiciones:</b>	-
<b>Post condiciones:</b>	-
<b>Casos de usos asociados:</b>	- Consultar Estado, Asignar Tareas
<b>Referencia:</b>	RF03, RF11 y RF04
<b>Prioridad:</b>	Normal
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. Selecciona la opción Consultar Estado.	2. Chequear Estado de Aprendizaje. 3. Chequear Resultados de Evaluaciones por Contenido. 4. Muestra un resumen de los resultados obtenidos.
5. El profesor verifica las habilidades y contenidos vencidos Selecciona la opción Asignar Tareas.	6. Cuando se selecciona la opción Asignar Tareas se activa el Subsistema de Ejercicios, tomando este el control de todas las acciones desde este momento. Muestra un formulario con los tipos de ejercicios.
7. Selecciona el tipo de ejercicio para proceder a la selección y acepta.	8. Muestra un formulario con los ejercicios a seleccionar para asignar la tarea.
9. Selecciona los ejercicios en	10. Envía al estudiante una notificación de los ejercicios

## Descripción de la Solución Propuesta




función de la habilidad a tratar y acepta la selección.	asignados.										
<b>Flujo Alternativo 1. Asignar tareas.</b>											
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>										
5. Si el promedio de las notas por contenido es aceptable no se asigna la tarea y termina el caso de uso.											
<b>Prototipo no Funcional</b>											
<div style="display: flex; justify-content: space-between; align-items: flex-start;"> <div style="text-align: center;">  <p><b>Profesor</b></p> </div> <div style="text-align: right;"> <p><b>Página Principal - Evaluaciones - Atención Individual</b></p> <p><b>Atención Individual</b>      <b>Asignar Tareas</b>      <b>Contenidos</b></p> <p><input checked="" type="checkbox"/> Estudiante       <input type="checkbox"/> Grupo</p> <div style="border: 1px solid black; padding: 2px; display: inline-block;">             Contenido 1              Contenido 2              Contenido 3           </div> </div> </div> <div style="margin-top: 10px;"> <p><b>Estudiante:</b>  <b>Bárbara Giselle Santos Figarola</b>  <b>grupo: 5305</b>  <b>Online: No</b></p> <div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">Nombre o usuario</div> <p><b>Buscar Estudiante</b></p> <div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">Usuario</div> <div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">Herramientas</div> <div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">Chat</div> <div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">Usuarios en Línea</div> <div style="display: flex; justify-content: space-between; align-items: center; margin-top: 10px;"> <span></span>  </div> </div> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 10px;"> <thead> <tr> <th style="width: 50%;">Actividad y Nombre</th> <th style="width: 10%;">Tiempo</th> <th style="width: 10%;">Nota</th> <th style="width: 10%;">Intentos</th> <th style="width: 20%;"></th> </tr> </thead> <tbody> <tr> <td style="height: 150px;"></td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>		Actividad y Nombre	Tiempo	Nota	Intentos						
Actividad y Nombre	Tiempo	Nota	Intentos								


Tabla 6. Descripción Textual. Realizar Atención Individual.

## *Descripción de la Solución Propuesta*

<b>Descripción textual “Evaluar Aprendizaje”.</b>	
<b>Caso de uso:</b>	Evaluar Aprendizaje
<b>Actores:</b>	Profesor
<b>Resumen:</b>	El caso de uso se inicia cuando el usuario selecciona la opción “Evaluar Aprendizaje” en el menú principal, visualizando de esta todos los resultados del estudiante en diferentes contenidos, lo que permitirá otorgar una calificación del aprendizaje
<b>Precondiciones:</b>	-
<b>Post condiciones:</b>	El usuario podrá establecer la nota correspondiente.
<b>Casos de usos asociados:</b>	-Asignar Tareas
<b>Referencia:</b>	RF03, RF05,RF04, RF02, RF06, RF07,RF11
<b>Prioridad:</b>	Normal
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El caso de uso comienza cuando el usuario selecciona la opción Evaluar Aprendizaje.	2. Chequear Estado de Aprendizaje. 3. Chequear Ejercicios Realizados. 4. Chequear Evaluaciones por Contenido. 5. Chequear Modificaciones. 6. Chequear Restricciones de Tiempo. 7. Establecer Evaluación. 8. Mostrar un resumen con los resultados obtenidos, y muestra las opciones: “Aceptar” “Cancelar”
9. Establece la evaluación correspondiente y selecciona la opción aceptar.	10. Establece la evaluación y envía una notificación informando que el contenido ha sido vencido y la nota obtenida.
<b>Prototipo no Funcional</b>	




## Descripción de la Solución Propuesta




**Profesor**

Estudiante:  
Bárbara Giselle Santos Figarola  
grupo: 5305  
Online: No

Buscar Estudiante



Contenidos Contenido 1  
Contenido 2  
Contenido 3

 **Asignar Tareas**

<b>Evaluación:</b>					
<b>Contenido Actual</b>					
<b>Resultados por contenido</b>					
<b>Lección:</b>					
Ejercicios	Tema	Nota	Intentos	Subió nota	Tiempo

Tabla 7.Descripción Textual. Evaluar Aprendizaje.

Descripción textual “Modificar evaluación final”	
<b>Caso de uso:</b>	Modificar Evaluación Final
<b>Actores:</b>	Profesor
<b>Resumen:</b>	El caso de uso se inicia cuando el usuario selecciona la opción “Modificar Evaluación”, a partir de este momento el usuario puede ver las evaluaciones obtenidas por el estudiante luego de la última evaluación y teniendo esto en cuenta modifica o no la evaluación.
<b>Precondiciones:</b>	Debe haber alguna evaluación registrada.
<b>Post condiciones:</b>	
<b>Casos de usos asociados:</b>	-
<b>Referencia:</b>	RF08y RF09
<b>Prioridad:</b>	Normal

## Descripción de la Solución Propuesta

Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El caso de uso comienza cuando el usuario selecciona la opción Modificar Evaluación Final, indica el nombre y usuario del estudiante al que desea modificarle la evaluación.	3. Chequear Resultados. 4. Establecer nueva evaluación. 5. Muestra el registro de evaluaciones, el campo para modificar la evaluación las opciones “Aceptar” y “Cancelar”.
6. Ingresa la nueva evaluación y selecciona la opción “Aceptar”.	7. Establece la evaluación nueva.




Prototipo no Funcional																													
 <b>Profesor</b>	<p style="text-align: center;"><b>Página Principal - Evaluaciones - Modificar Evaluación Final</b></p> <div style="display: flex; justify-content: space-between; align-items: flex-start;"> <div style="width: 60%;"> <p><b>Estudiante:</b>  <b>Bárbara Giselle Santos Figarola</b>                      grupo: 5305                      Online: No</p> <p><input type="text" value="Nombre o usuario"/></p> <p>Buscar Estudiante</p> <p><input type="text" value="Usuario"/></p> <p><input type="text" value="Herramientas"/></p> <p><input type="text" value="Chat"/></p> <p><input type="text" value="Usuarios en Línea"/></p> <p><input type="button" value="Salir"/> </p> </div> <div style="width: 35%;"> <p>Contenidos</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td>Contenido 1</td></tr> <tr><td>Contenido 2</td></tr> <tr><td>Contenido 3</td></tr> </table> </div> </div> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 10px;"> <tr><td colspan="5"><b>Evaluación:</b></td></tr> <tr><td colspan="5"><b>Contenido Actual:</b></td></tr> <tr><td colspan="5"><b>Resultados del Contenido:</b></td></tr> <tr> <th style="width: 20%;">Ejercicios</th> <th style="width: 20%;">Tema</th> <th style="width: 15%;">Nota</th> <th style="width: 15%;">Nota/Intento</th> <th style="width: 30%;">Tiempo</th> </tr> <tr> <td style="height: 150px;"></td> <td></td> <td></td> <td></td> <td></td> </tr> </table>	Contenido 1	Contenido 2	Contenido 3	<b>Evaluación:</b>					<b>Contenido Actual:</b>					<b>Resultados del Contenido:</b>					Ejercicios	Tema	Nota	Nota/Intento	Tiempo					
Contenido 1																													
Contenido 2																													
Contenido 3																													
<b>Evaluación:</b>																													
<b>Contenido Actual:</b>																													
<b>Resultados del Contenido:</b>																													
Ejercicios	Tema	Nota	Nota/Intento	Tiempo																									

Tabla 8. Descripción Textual. Modificar Evaluación Final

## *Descripción de la Solución Propuesta*

<b>Descripción Textual “Asignar Tareas”.</b>	
<b>Caso de uso:</b>	Asignar Tareas
<b>Actores:</b>	Profesor
<b>Resumen:</b>	El caso de uso se inicia cuando el profesor selecciona la opción “Asignar Tareas”, lo que le permite al profesor orientar ejercicios, ya sea como estudio independiente de las lecciones o como parte de la consolidación y solución a deficiencias.
<b>Precondiciones:</b>	
<b>Post condiciones:</b>	
<b>Casos de usos asociados:</b>	-
<b>Referencia:</b>	
<b>Prioridad:</b>	Normal
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El caso de uso comienza cuando el profesor selecciona la opción “Asignar Tareas”.	2. Se activa el subsistema de ejercicios, y se realizan las acciones siguientes: Muestra un formulario con los tipos de ejercicios a asignar.
3. El profesor selecciona el tipo de ejercicio que desea asignar.	4. Muestra un formulario con los ejercicios a seleccionar para asignar la tarea.
5. Selecciona los ejercicios en función de la habilidad a tratar y acepta la selección.	6. Envía al estudiante una notificación de los ejercicios asignados.
<b>Prototipo no Funcional</b>	


## Descripción de la Solución Propuesta




**Profesor**

**Estudiante:**  
**Bárbara Giselle Santos Figarola**  
**grupo: 5305**  
**Online: No**

Buscar Estudiante



**Página Principal - Evaluaciones - Asignar Tareas**

 **Notificar**

**Área de Trabajo -**  
**Subsistema de Ejercicios**


Tabla 9. Descripción Textual. Asignar Tareas.

Descripción Textual “Realizar Corte Intermedio”.	
<b>Caso de uso:</b>	Realizar Corte Intermedio
<b>Actores:</b>	Profesor
<b>Resumen:</b>	El caso de uso se inicia cuando el usuario selecciona la opción “Realizar Corte Intermedio” en el menú principal, en cual revisa las evaluaciones alcanzadas por el estudiante, y basado en éstas, establece la nota correspondiente al período que se evalúa.
<b>Precondiciones:</b>	-
<b>Post condiciones:</b>	El profesor podrá establecer la nota correspondiente.
<b>Casos de usos asociados:</b>	-
<b>Referencia:</b>	RF02, RF04, RF03, RF05, RF11
<b>Prioridad:</b>	Normal
<b>Flujo Normal de Eventos</b>	

## Descripción de la Solución Propuesta

Acción del Actor	Respuesta del Sistema
1. El caso de uso comienza cuando el usuario selecciona la opción Evaluar Aprendizaje.	2. Chequear Estado de Aprendizaje. 3. Chequear Resultados de Evaluaciones por Contenido. 4. Chequear Modificaciones. 5. Chequear Ejercicios Realizados. 6. Mostrar un resumen con los resultados obtenidos.
8. Establece la evaluación correspondiente y selecciona la opción aceptar.	9. Establece la evaluación y envía una notificación al estudiante informando la evaluación obtenida en el corte.

### Prototipo no Funcional



**Profesor**

**Estudiante:**  
Bárbara Giselle Santos Figarola  
grupo: 5305  
Online: No

Nombre o usuario



Buscar Estudiante

Usuario




Herramientas

Chat

Usuarios en Línea

 Salir


**Página Principal - Evaluaciones - Modificar Evaluación Final**

 Contenidos
 Ejercicios
 Modificaciones

Evaluación Final:				
Contenido Actual :				
Actividad	Tema	Nota	Tiempo	Intentos

Tabla 10. Descripción Textual. Realizar Cortes Intermedios.

### 2.5 Análisis del Sistema

El modelo de análisis del sistema no es más que una representación de la interacción existente entre las clases del sistema y el actor. Este modelo está hecho en el lenguaje del programador con la intención de lograr un total entendimiento del sistema en el momento de su implementación. Para modelar el diagrama de clases del sistema, intervienen las clases entidad, controladora e interfaz, las cuales cuentan con estereotipos propios y reglas establecidas para su interrelación.

**Clase entidad:** almacena la información persistente del sistema, es decir contiene los datos principales para su funcionamiento.

**Clase controladora:** es la encargada de gestionar toda la información que se solicita por parte del usuario y se muestra por parte de la interfaz. Tiene relación con las dos clases restantes.

**Clase interfaz:** se encarga de mediar la comunicación del usuario con el sistema y mostrar la información solicitada.

#### 2.5.1 Diagrama de Clases del Análisis

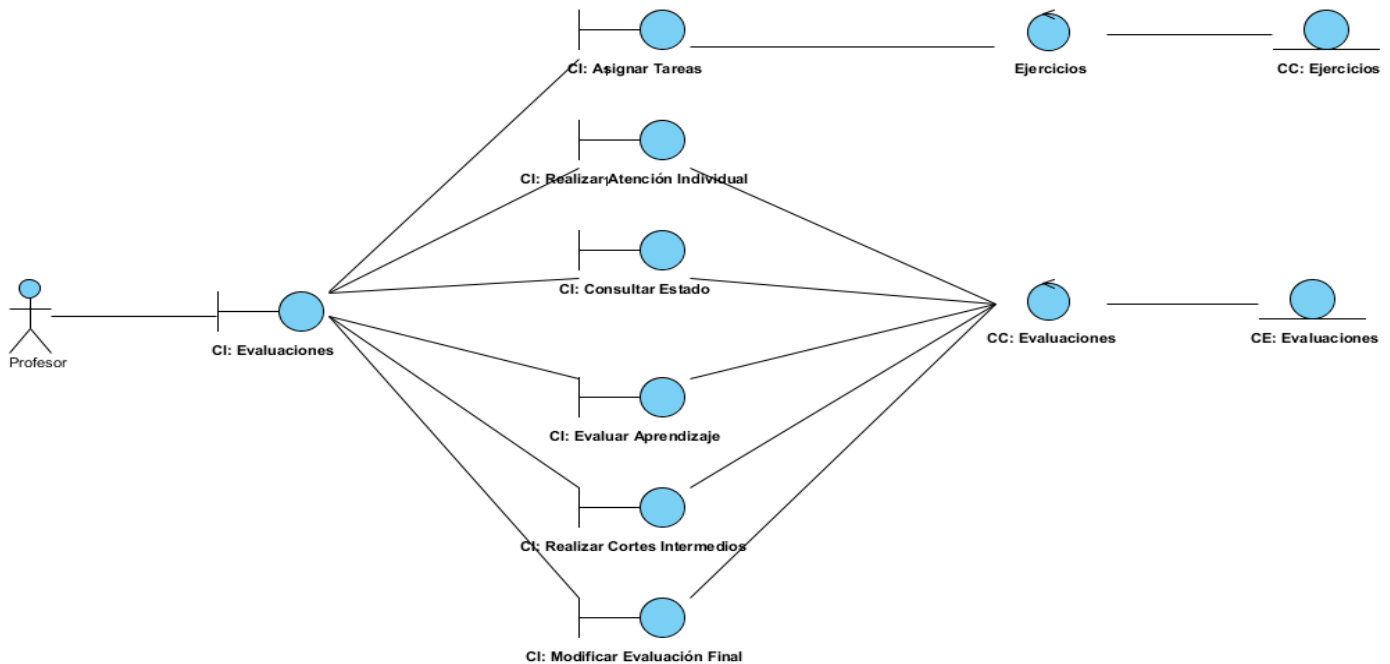


Figura 5. Diagrama de Clases del Análisis.

### 2.5.2 Diagramas de Interacción

Diagramas de interacción, es la clasificación que engloba a los diagramas de secuencia y colaboración, los cuales determinan la comunicación entre objetos y describe las reglas de actividad interna (disparos) del sistema de una forma gráfica. Dichos diagramas se realizan de forma independiente para cada uno de los casos de uso identificados apoyado en la previa realización de la descripción textual de los mismos y sus respectivos diagramas de clases del análisis.

Al igual que los Diagramas de Colaboración los diagramas de secuencia se utilizan para ilustrar la realización de un CU. Son particularmente importantes para los diseñadores pues aclaran los roles jugados por los objetos en un flujo, lo cual le proporciona un gran valor para la determinación de las responsabilidades de las clases. A diferencia del Diagrama de Colaboración este incluye secuencia cronológica de los mensajes y no la relación entre los objetos, por lo que es mejor su utilización cuando el orden en el tiempo de los mensajes es de importancia [11].

#### 1. Diagrama de Secuencia. Asignar Tareas.

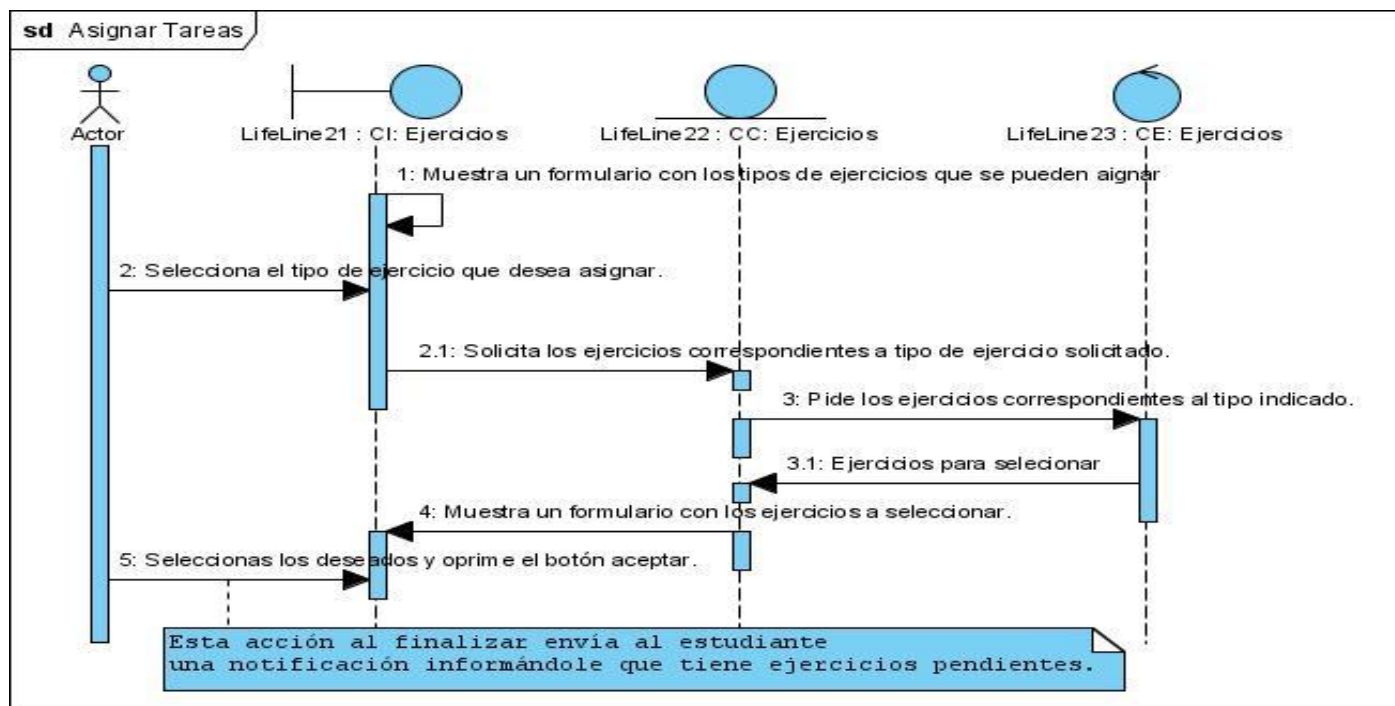


Figura 6. Diagrama de Secuencia. Asignar Tareas.

### 2. Diagrama de Secuencia. Atención Individual.

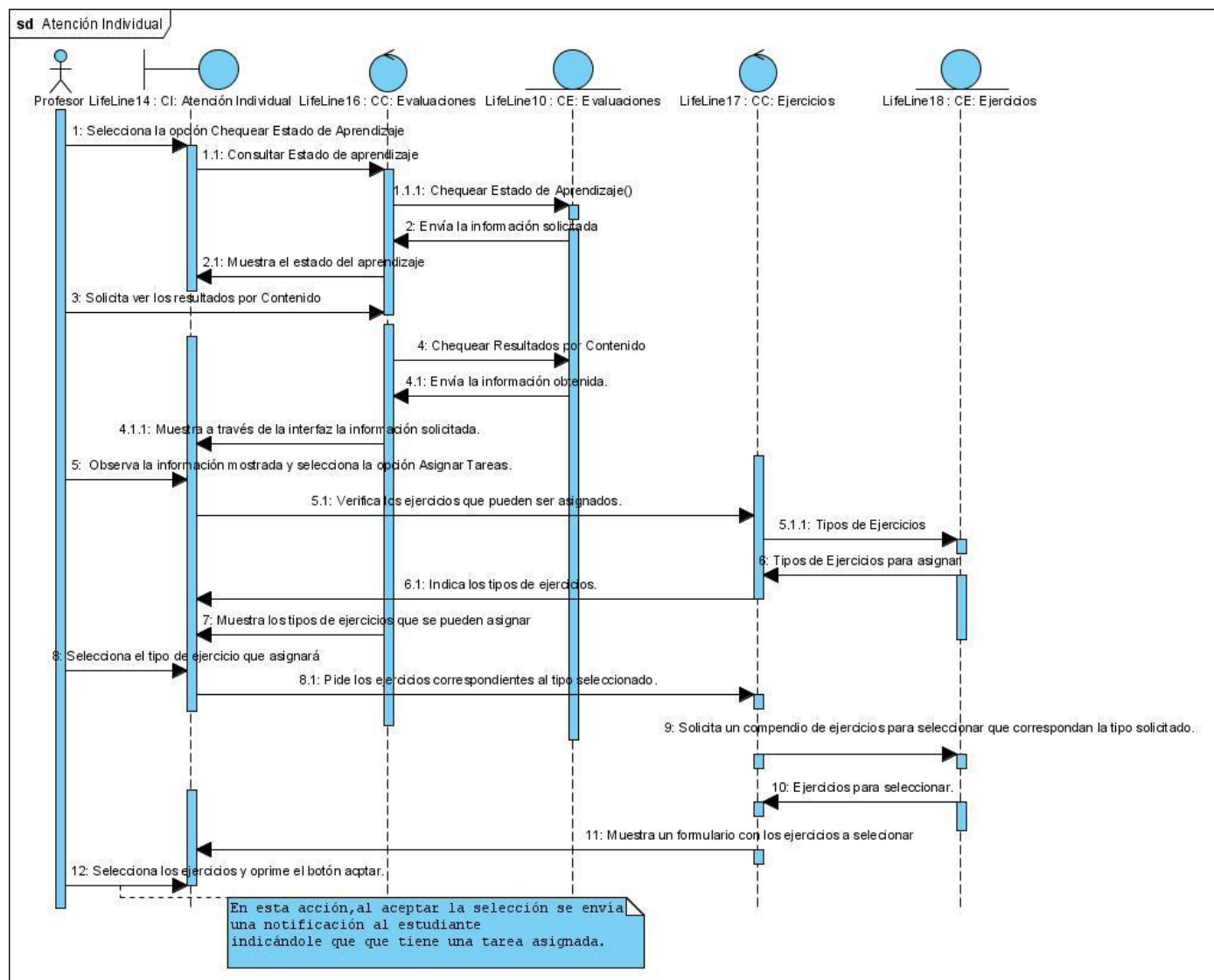


Figura 7. Diagrama de Secuencia. Realizar Atención Individual.



## Descripción de la Solución Propuesta

### 3. Diagrama de Secuencia. Evaluar Aprendizaje.

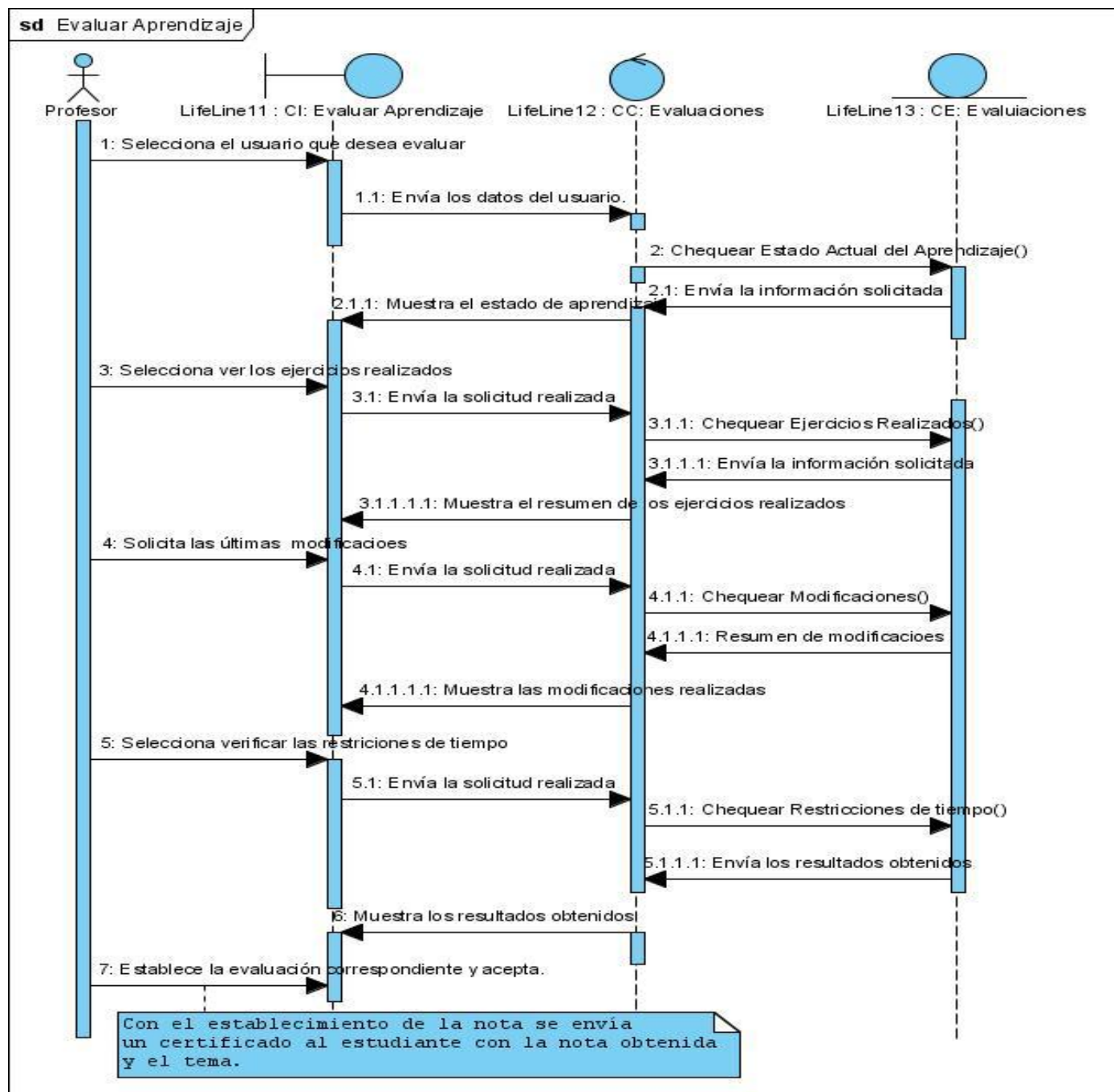


Figura 8. Diagrama de Secuencia. Evaluar Aprendizaje.

### 4. Diagrama de Secuencia. Consultar Estado.

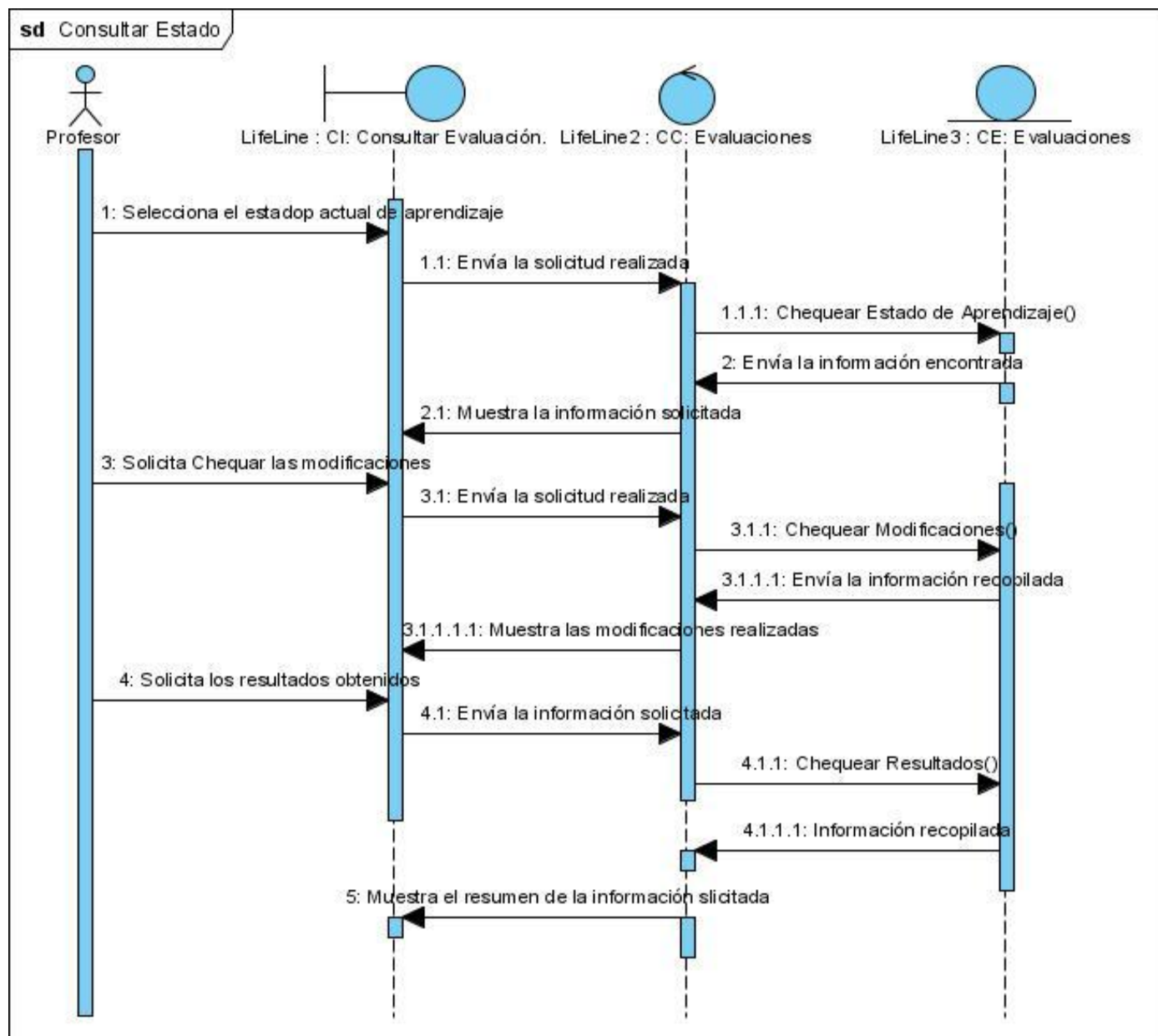


Figura 9. Diagrama de Secuencia. Consultar Estado.

### 5. Diagrama de Secuencia. Consultar Evaluación.

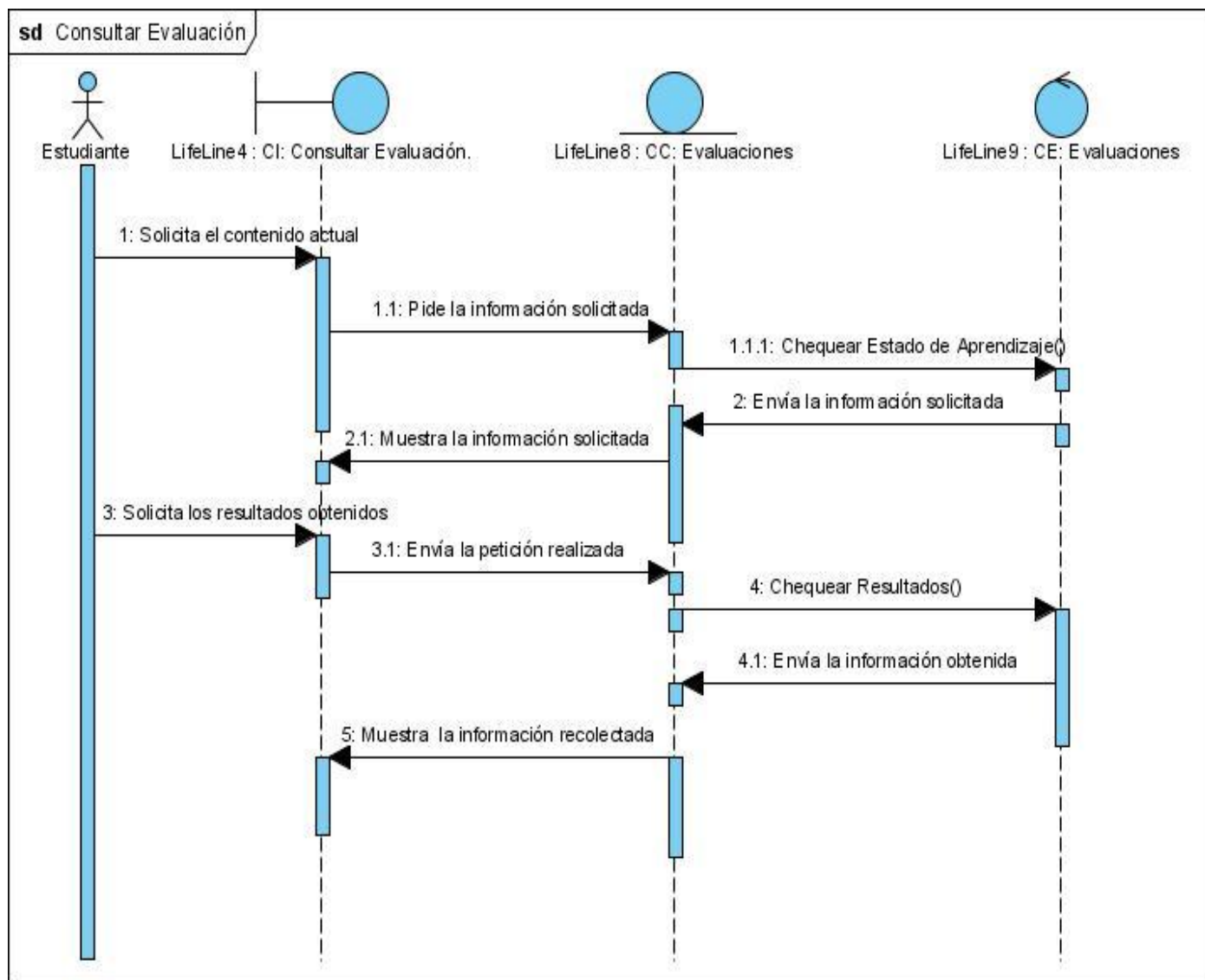


Figura 10. Diagrama de Secuencia. Consultar Evaluación.

### 6. Diagrama de Secuencia. Cortes Intermedios.

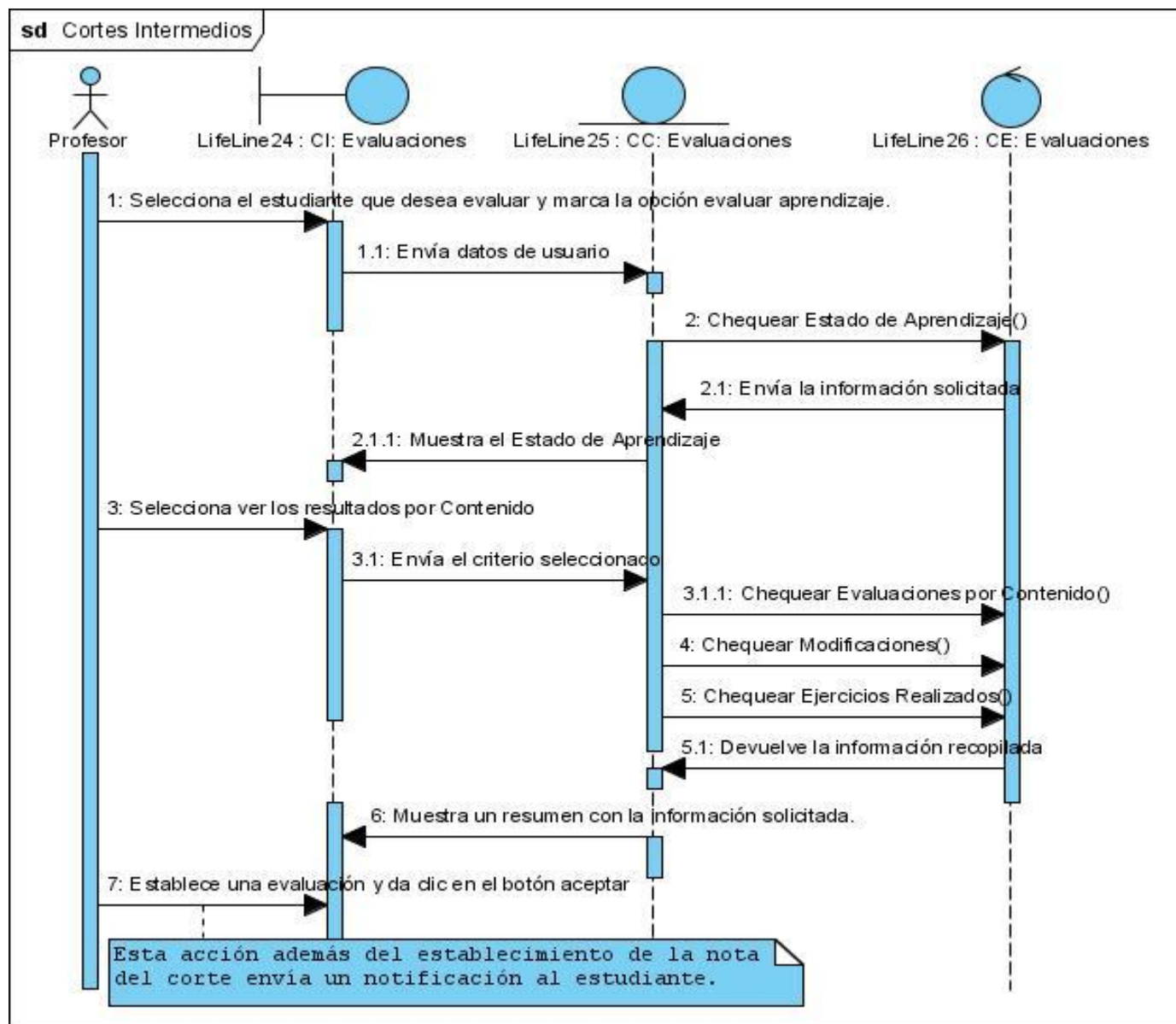


Figura 11. Diagrama de Secuencia. Realizar Cortes Intermedios.

### 7. Diagrama de Secuencia. Modificar Evaluación Final.

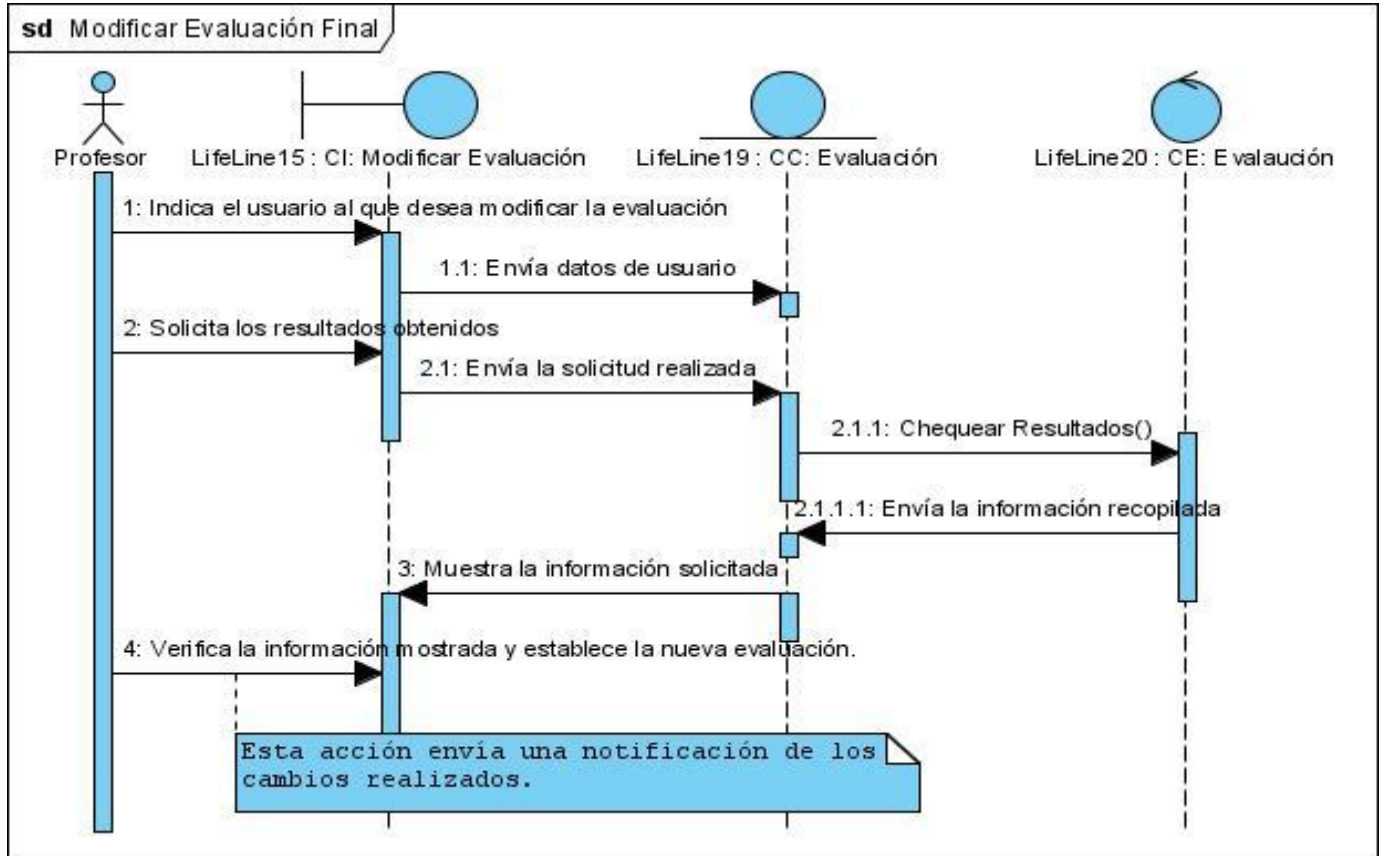


Figura 12. Diagrama de Secuencia. Modificar Evaluación Final.

## 2.6 Diseño del Sistema

El diseño del sistema centra su principal objetivo en la distribución de funcionalidades afines, agrupándolas en subsistemas o componentes del sistema, que con la relación e interacción existente entre ellos, logran abarcar los requisitos del sistema en general. Para la realización de un diseño que a la larga sea satisfactorio y de fácil comprensión, es preciso llevar a cabo actividades como dividir requerimientos, identificar subsistemas, asignar requerimientos a los subsistemas, especificar las funcionalidades de los subsistemas y definir las interfaces de los subsistemas.

### 2.6.1 Diagrama de Clases del Diseño

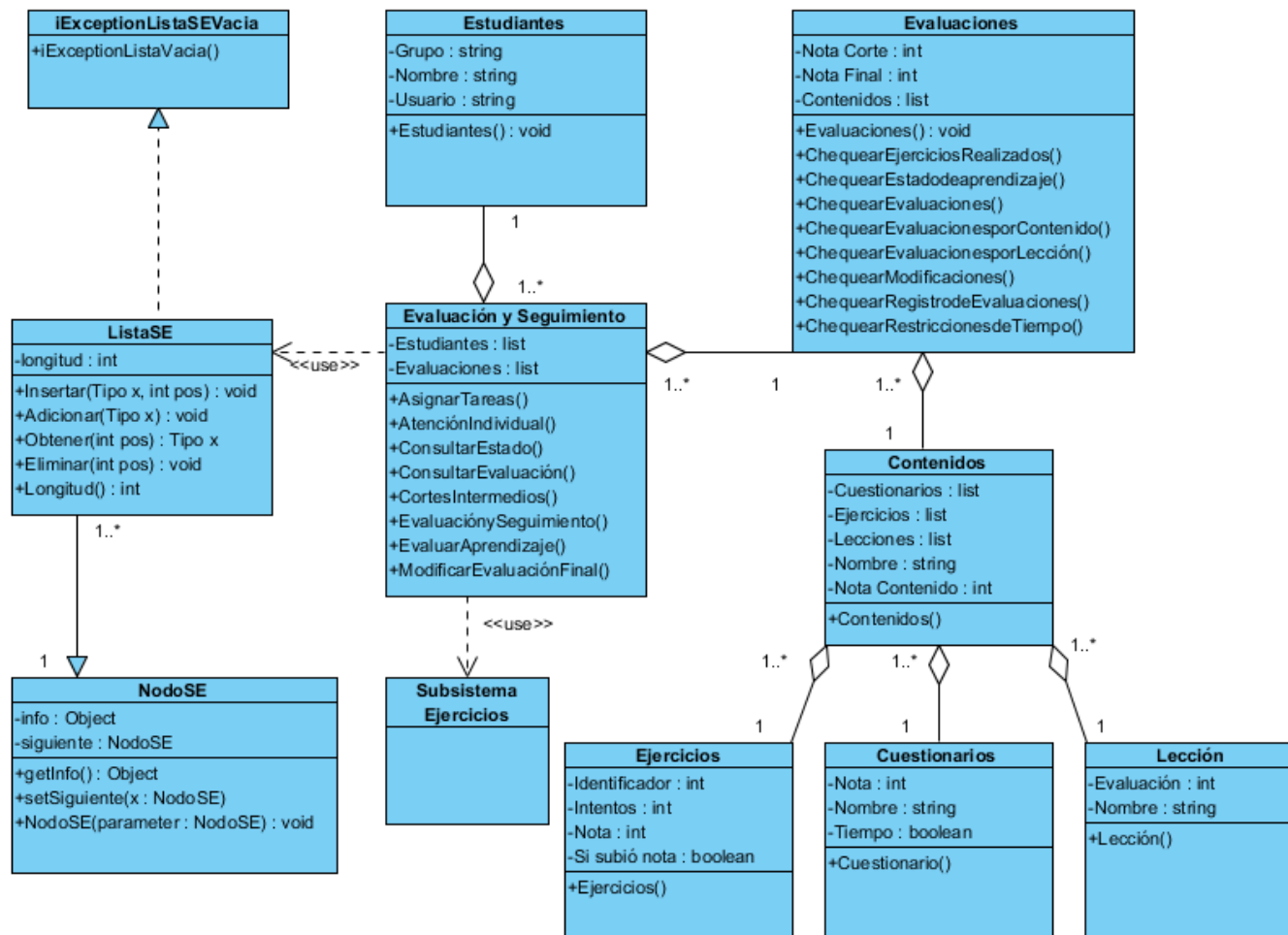


Figura 13. Diagrama de Clases del Diseño.

### 2.6.2 Patrones Utilizados

Para lograr un diseño óptimo que conlleve a una aplicación bien estructurada es preciso la utilización de patrones de diseño, que no son más que “soluciones simples y elegantes a problemas específicos y comunes del diseño orientado a objetos” según Joaquín Gracia, entre los más conocidos figuran los patrones para la asignación de responsabilidades (GRASP) y los patrones GOF, encargados de describir y brindar soluciones a problemas relacionados con el comportamiento y el propósito.

## *Descripción de la Solución Propuesta*

---

Las decisiones poco acertadas dan origen a sistemas y componentes frágiles y difíciles de mantener, entender, reutilizar o extender. Una implementación hábil se funda en los principios cardinales que rigen un buen diseño orientado a objetos. Los patrones GRASP codifican buenos principios y sugerencias relacionados frecuentemente con la asignación de responsabilidades. GRASP es un acrónimo que significa General Responsibility Assignment Software Patterns (patrones generales de software para asignar responsabilidades) [13].

Los patrones GRASP abarcan un gran grupo, de ellos fueron utilizados:

**Patrón Experto:** Se conserva el encapsulamiento, ya que los objetos se valen de su propia información para hacer lo que se les pide. Esto soporta un bajo acoplamiento, lo que favorece al hecho de tener sistemas más robustos y de fácil mantenimiento. El comportamiento se distribuye entre las clases que cuentan con la información requerida (atributos), alentando con ello definiciones de clase “sencillas” y más cohesivas que son más fáciles de comprender y de mantener. Así se brinda soporte a una alta cohesión. Es preciso tener en cuenta aspectos del sistema como lógica del negocio, persistencia a la base de datos e interfaz de usuario [13].

**Patrón Creador:** La creación de objetos es una de las actividades más frecuentes en un sistema orientado a objetos. En consecuencia, conviene contar con un principio general para asignar las responsabilidades concernientes a ella. El diseño, bien asignado, puede soportar un bajo acoplamiento, una mayor claridad, el encapsulamiento y la reusabilidad. El patrón Creador guía la asignación de responsabilidades relacionadas con la creación de objetos; su propósito fundamental es encontrar un creador para conectarlo con el objeto producido en cualquier evento. Para la asignación de responsabilidades utilizando este patrón, el mismo indica que se asigna la responsabilidad de que una clase B cree un Objeto de la clase A solamente cuando:

- B contiene a A.
- B es una agregación (o composición) de A.
- B almacena a A.
- B tiene los datos de inicialización de A (datos que requiere su constructor).
- B usa a A.

## *Descripción de la Solución Propuesta*

---

**Bajo Acoplamiento:** El grado de acoplamiento no puede considerarse aisladamente de otros principios como Experto y Alta Cohesión. Sin embargo, es un factor a considerar cuando se intente mejorar el diseño. Con el bajo acoplamiento no hay afectación por cambios en otros componentes, resultan fáciles de entender por separado y fáciles de reutilizar. Para determinar el nivel de acoplamiento de clases, son recomendables los diagramas de colaboración de UML. Uno de los principales síntomas de un mal diseño y alto acoplamiento es una herencia muy profunda. Siempre hay que considerar las ventajas de la delegación respecto de la herencia [14].

**Alta Cohesión:** La cohesión es una medida de cuán relacionadas y enfocadas están las responsabilidades de una clase. Éste patrón mejora la claridad y facilidad con que se entiende el diseño, se simplifica el mantenimiento y las mejoras de funcionalidad, a menudo se genera un bajo acoplamiento y soporta mayor capacidad de reutilización. Un buen diseño se produce cuando se crean los denominados “paquetes de servicio” o clases agrupadas por funcionalidades que son fácilmente reutilizables (bien por uso directo o por herencia), ofrecen facilidad para su comprensión, conservación y transformaciones [14].

Este patrón se puede manifestar de las formas siguientes:

- Muy baja cohesión: Una clase es la única responsable de muchas cosas en áreas funcionales heterogéneas.
- Baja cohesión: Una clase tiene la responsabilidad exclusiva de una tarea compleja dentro de un área funcional.
- Alta cohesión: Una clase tiene responsabilidades moderadas en un área funcional y colabora con las otras para llevar a cabo las tareas.
- Cohesión moderada: Una clase tiene peso ligero y responsabilidades exclusivas en unas cuantas áreas que están relacionadas lógicamente con el concepto de clase pero no entre ellas.

Los patrones de diseño no son fáciles de entender, pero una vez entendido su funcionamiento, los diseños serán mucho más flexibles, modulares y reutilizables. Han revolucionado el diseño orientado a objetos y todo buen arquitecto de software debería conocerlos. Los patrones GoF están dirigidos al diseño en función de creación, comportamiento y estructura.



## *Descripción de la Solución Propuesta*

---

---

Patrones de creación:

Prototipo. Especifica los tipos de objetos a crear por medio de una instancia prototípica, y crear nuevos objetos copiando este prototipo.

- Singleton. Garantiza que una clase sólo tenga una instancia, y proporciona un punto de acceso global a ella.

Patrones estructurales:

- Composite. Combina objetos en estructuras de árbol para representar jerarquías de parte-todo. Permite que los clientes traten de manera uniforme a los objetos individuales y a los compuestos.
- Facade. Proporciona una interfaz unificada para un conjunto de interfaces de un subsistema. Define una interfaz de alto nivel que hace que el subsistema sea más fácil de usar.

Patrones de comportamiento:

- Memento. Representa y externaliza el estado interno de un objeto sin violar la encapsulación, de forma que éste puede volver a dicho estado más tarde.
- Observer. Define una dependencia de uno-a-muchos entre objetos, de forma que cuando un objeto cambia de estado se notifica y actualizan automáticamente todos los objetos [14].

### **Conclusiones Parciales**

En este capítulo se le da cumplimiento al segundo de los objetivos específicos de esta investigación con la realización de los modelos de análisis y diseño. Dichos modelos facilitan el proceso de comprensión de la solución que se propone, para la cual se utilizaron y tuvieron en cuenta aspectos fundamentales del sistema que se necesita, como los requisitos funcionales y no funcionales descritos en el capítulo. Con los modelos de análisis y diseño, los requisitos funcionales fueron transformados, convirtiéndose en la entrada fundamental para la etapa de implementación.

## CAPÍTULO 3: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

### Introducción

El presente capítulo estará dirigido a la validación de los artefactos generados y los requisitos funcionales y no funcionales obtenidos durante el proceso de elicitación. La validación se hace con el objetivo de demostrar la calidad del trabajo realizado y la utilidad que tendrán los modelos de análisis y diseño en la futura implementación del sistema.

### 3.1 Verificación de Requisitos del Sistema

Para revivificar la trazabilidad de los requisitos funcionales desde su captura y especificación hasta los diagramas de secuencia, se utilizó la matriz de trazabilidad, técnica que permite verificar que los casos de uso engloben o abarquen todos los requisitos funcionales del sistema, además de verificar en qué casos de usos está incluido cada requisito así como en los Diagramas de Secuencia.

Relationships: - direct only	CU1: Evaluar Aprendizaje	CU2: Asignar Tareas	CU3: Realizar Atención Individual	CU4: Realizar Cortes Intermedios	CU5: Consultar Estado	CU6: Consultar Evaluación	CU7: Modificar Evaluación Final
RF1: Chequear Estado de Aprendizaje	↗		↗	↗	↗		
RF2: Chequear Restricciones de Tiempo	↗						
RF4: Chequear Evaluaciones por Contenido	↗		↗	↗			
RF5: Chequear Evaluaciones por Lección			↗	↗			
RF6: Chequear Resultados					↗	↗	↗
RF7: Chequear Modificaciones	↗				↗	↗	↗
RF8: Chequear Ejercicios Realizados	↗						
RF9: Establecer Evaluaciones.	↗		↗				
RF10: Establecer Cambio de Evaluaciones.							↗
RF11: Enviar Mensaje de Notificación.	↗						

Figura 14. Matriz de Trazabilidad (Requisitos Funcionales - Casos de Uso)

Relationships: - direct only		CU1: Evaluar Aprendizaje	CU2: Asignar Tareas	CU3: Realizar Atención Individual	CU4: Realizar Cortes Intermedios	CU5: Consultar Estado	CU6: Consultar Evaluación	CU7: Modificar Evaluación Final
RF1: Chequear Estado de Aprendizaje		↗	↗	↗	↗	↗		
RF2: Chequear Restricciones de Tiempo		↗						
RF4: Chequear Evaluaciones por Contenido		↗		↗	↗			
RF5: Chequear Evaluaciones por Lección				↗	↗			
RF6: Chequear Resultados						↗	↗	↗
RF7: Chequear Modificaciones		↗				↗	↗	↗
RF8: Chequear Ejercicios Realizados		↗						
RF9: Establecer Evaluaciones.		↗		↗				
RF10: Establecer Cambio de Evaluaciones.								↗
RF11: Enviar Mensaje de Notificación.		↗						

Figura 15. Matriz de Trazabilidad (Casos de Uso - Diagrama de Secuencia)

Luego de realizada la matriz de trazabilidad, se concluye que cada requisito se intercepta al menos con un caso de uso, demostrando que existe concordancia entre unos y otros, así como los casos de uso coinciden con los diagramas de secuencia. La trazabilidad de los requisitos es una forma eficaz de comprobar el cumplimiento de los objetivos del módulo Evaluación y Seguimiento a través de la realización de los requisitos funcionales. Estas matrices expresan que los requisitos descritos cubren el 100% de los objetivos planteados.

### 3.2 Modelo de Métricas Orientada a Objetos aplicadas al Diagrama de Casos de Uso del Sistema

Para la aplicación de las métricas de calidad al diagrama de casos de uso del sistema, se tuvieron en cuenta la cantidad de requisitos funcionales existentes, las descripciones textuales de casos de uso, además de aspectos específicos requeridos por cada métrica utilizada y los criterios establecidos por las mismas para la interpretación de los resultados.

## Validación de la Solución Propuesta

Factores de Completitud	Métricas Asociadas	Evaluación del módulo Seguimiento y Evaluación
<b>Factor1:</b> ¿Han sido definidos todos los roles relevantes de usuario encargado de generar, modificar o consultar información?	<b>Métrica 1:</b> Número de roles relevantes omitidos	Total de roles relevantes: 1 Número de roles omitidos: 0 Representa: 100%
<b>Factor2:</b> ¿Están definidos todos los requisitos que justifican la funcionalidad del caso de uso?	<b>Métrica 2:</b> Número de requisitos omitidos por caso de uso.  <b>Métrica3:</b> Número de casos de uso que tienen requisitos omitidos.	<b>Total de requisitos:</b> 11 Número de requisitos omitidos por casos de uso: 0 Representa: 100% <b>Total de casos de uso:</b> 7 Número de casos de uso que tienen requisitos omitidos: 0 Representa: 100%
<b>Factor 3</b> ¿Se describen las condiciones de excepción relevantes que debe contemplar cada flujo de eventos?	<b>Métrica 4:</b> Número de casos de uso que no describen condiciones de excepción relevante.	<b>Total de casos de uso:</b> 7 Número de casos de uso que no describen condiciones de excepción relevantes: 5 Representa: 71,42 %
Factores de Consistencia	Métricas asociadas	Evaluación del módulo Seguimiento y Evaluación
<b>Factor 4</b> ¿El nombre dado a los casos de uso es una expresión verbal que	<b>Métrica 5:</b> Número de casos de uso que tienen nombre incorrecto.	<b>Total de casos de uso:</b> 7 Número de casos de uso que tiene nombre incorrecto: 0

## *Validación de la Solución Propuesta*

describe alguna funcionalidad relevante en el contexto del usuario?		Representa: 0%
<b>Factor 5</b> ¿Representa el caso de uso una interacción observable por el actor?	<b>Métrica 6:</b> Número de casos de uso que no representan una interacción observable por un actor.	<b>Total de casos de uso:7</b> Número de casos de uso que no representan una interacción observable por un actor: 0 Representa: 0%
<b>Factor 6</b> ¿La descripción del flujo de eventos se inicia con la descripción de una acción externa originada por un actor o una condición interna del sistema claramente identificable?	<b>Métrica 7:</b> Número de casos de uso cuya descripción extendida no inicia con una acción externa o con una condición monitoreada por el sistema.	<b>Total de casos de Uso:7</b> Número de casos de uso cuya descripción extendida no inicia con una acción externa o con una condición monitoreada por el sistema: 0 Representa: 100%
<b>Factor 7:</b> Si en el caso de uso interviene más de un actor, ¿existe claridad en cuál de ellos es el actor iniciador?	<b>Métrica 8:</b> Número de casos de uso con más de un actor, que no describe cuál es el actor iniciador.	<b>Total de casos de Uso: 7</b> Número de casos de uso con más de un actor, que no describe cuál es el actor iniciador: 0 Representa: 100%
<b>Factor de Correctitud</b>	<b>Métricas Asociadas</b>	<b>Evaluación del módulo Seguimiento y Evaluación</b>
<b>Factor 8</b> ¿Representa el caso de uso requisitos comprensibles por el usuario?	<b>Métrica 9:</b> Grado en que los requisitos representados por el caso de uso son comprensibles por el usuario.	<b>Total de requisitos: 11</b> Cantidad de requisitos que no son comprensibles por el usuario: Representa: 0%

	<b>Métrica 10:</b> Número de casos de uso en que los requisitos representados no son comprensibles por el usuario.	<b>Total de casos de Uso: 7</b> Número de casos de uso en que los requisitos representados no son comprensibles por el usuario: 0 Representa: 0%
<b>Factores de Complejidad</b>	<b>Métricas Asociadas</b>	<b>Evaluación del módulo Seguimiento y Evaluación</b>
<b>Factor 9</b> ¿Los elementos dentro del diagrama están Adecuadamente ubicados de manera que facilitan su implementación?	<b>Métrica 11:</b> Número de elementos del diagrama que requieren reubicación.	<b>Total de casos de Uso:7</b> Número de elementos del diagrama que requieren reubicación: 0 Representa: 100%

**Tabla 11. Modelo de Métricas Orientas a Objetos.**

Una vez aplicado el modelo de Métricas Orientadas a Objetos al Diagrama de Casos Uso del Sistema, queda demostrado que cuenta con la calidad requerida para realizar diseño del sistema, al encontrarse acorde con todos los requisitos identificados mediante los casos de uso, los cuales presentan una descripción detallada con todas las acciones del flujo de eventos redactado en función del iniciador y responsable de cada actividad. En la evaluación efectuada, el DCUS obtuvo una calificación de 98,47% de funcionalidad, la contribución de cada atributo a la calidad total fue: Completitud 93,86%, Consistencia 100%, Correctitud 100% y Complejidad 100%.

### **3.3 Prototipos no Funcionales**

Dentro de los prototipos no funcionales pueden identificarse dos grupos, los prototipos desechables, empleados en la validación de requisitos con una existencia corta en el ciclo de vida del proyecto; y los prototipos evolutivos, los cuales son mostrados al cliente y son los encargados de validar los requisitos contribuyendo también a su entendimiento. En la presente investigación se hizo uso de los prototipos desechables.

## *Validación de la Solución Propuesta*

---

---

Para llevar a cabo el proceso de validación de los requisitos y el diseño de la aplicación, los prototipos fueron mostrados a los usuarios finales, integrados a las descripciones textuales a las pertenecen, con el objetivo de situar al usuario en el contexto de cada funcionalidad y lo que ocurre en ellas. De esta forma pudieron identificar claramente las inconsistencias de los prototipos con lo que desean de la aplicación, además de hacer todas las sugerencias pertinentes antes de la implementación.

Una vez aplicada esta técnica, los prototipos resultaron tener buena aceptación por parte de los clientes, partiendo de que los requisitos identificados satisfacen sus necesidades y se comprobó además la ausencia de ambigüedades e inconsistencias.

### **Conclusiones Parciales**

En el capítulo que concluye se evaluaron y analizaron los resultados obtenidos. Los modelos mostrados en el capítulo destinado a la descripción propuesta, fueron sometidos a un proceso de evaluación con el objetivo de medir y validar la calidad de los mismos, por lo que se concluye lo siguiente:

- La realización de la matriz de trazabilidad permitió definir relación de los requisitos con los casos de uso, así como el cumplimiento de las funcionalidades del módulo Evaluación y Seguimiento a través de los casos de uso.
- Se validó el diseño de la aplicación a través de los prototipos no funcionales, los cuales sirvieron de técnica de validación arrojando como resultado la calidad de los mismos y el elevado nivel de aceptación alcanzado, al no detectarse ambigüedades.
- Se analizó y validó el diagrama de casos de uso mediante las métricas orientada a objetos, lo que arrojó como resultado que el diagrama de casos de uso cuenta con 100% de funcionalidad.

### **Conclusiones Generales**

Con la realización de esta investigación y teniendo en cuenta los objetivos específicos trazados para la misma se concluye que:

- El estudio realizado sobre los laboratorios virtuales y en especial del módulo Evaluación y Seguimiento, permitió valorar la importancia del mismo en el apoyo a la formación y consolidación de conocimientos y habilidades.
- Con la utilización de las técnicas de ingeniería de requisitos se capturaron, especificaron y validaron los requisitos del sistema, obteniendo la calidad necesaria en los mismos y un elevado nivel de aceptación.
- Los modelos de análisis y diseño generados, permiten una mejor comprensión de la solución propuesta ya que fueron generados bajo los estándares de calidad establecidos, demostrado con la validación realizada, por lo que le permitirán desarrollador entender el funcionamiento del módulo Evaluación y Seguimiento y proceder a su implementación.



### **Recomendaciones**

Luego de haberse cumplido los objetivos propuestos para la presente investigación se recomienda:

- Realizar una segunda iteración del proceso de refinamiento de los requisitos, con el objetivo de verificar si los mostrados en esta investigación aún son suficientes o se le deberían realizar cambios.
- Realizar la implementación del módulo Evaluación y Seguimiento del Laboratorio Virtual de Sistemas Operativos.
- Hacer un estudio de las posibles funcionalidades que pudieran adicionarse a la presente propuesta con el objetivo de hacerlo más completo y eficiente.

### Referencias Bibliográficas

1. Vary, James P. Informe de la reunión de expertos sobre laboratorios virtuales. [Online] 2000. [Cited: Marzo 18, 2011.] <http://unesdoc.unesco.org/images/0011/001191/119102s.pdf>.
2. Recursos didácticos basados en Internet para el apoyo a la enseñanza de materias del área de la Ingeniería de Sistemas y Automática. Candelas Herias, Francisco A. and Sánchez Moreno, José. 2, Madrid, España : s.n., Abril 2005, Revista Iberoamericana de Automática e Informática Industrial, Vol. 2, pp. 93-101. ISSN 1697-7912. [Online] <http://rua.ua.es/dspace/handle/10045/4608>
3. Dormido, S., Sánchez, J. and Morilla, F. Laboratorios virtuales y remotos para la práctica a distancia de la automática. [Online] [Cited: Marzo 20, 2011.] [http://e\\_spacio.uned.es](http://e_spacio.uned.es).
4. García-Beltrán , Ángel and Martínez, Raquel. AulaWeb: un sistema para la gestión, evaluación y seguimiento de asignaturas. [Online] [Cited: Marzo 18, 2011.] <http://www.dii.etsii.upm.es>.
5. Introducción al desarrollo del software. EVA. [Online] 2008. [Cited: Marzo 28, 2011.] <http://eva.uci.cu/mod/resource/view.php?id=11361>.
6. Documento Prototipo. [Online] [Cited: Marzo 29, 2011.] <http://siv.cenditel.gob.ve/miv/wiki/DocumentoPrototipo>
7. Fernández, P., Salaverria, A. and Valdés, A. Laboratorio Virtual como conjunto de objetos de aprendizaje de los dispositivos electrónicos. Utilización como herramienta de autoevaluación.[Online] [Cited: Marzo 25, 2011.] <http://taee.euitt.upm.es/Congresosv2/2008/papers/2008S2D03.pdf>.
8. Andriano, Natalia Valeria . Comparación del Proceso de Elicitación de Requerimientos en el desarrollo de Software a Medida y Empaquetado. Propuesta de métricas para la elicitación. [Online] Agosto 2006. [Cited: Marzo 28, 2011.] <http://postgrado.info.unlp.edu.ar/Carrera/Magister/IngenieriadeSoftware/Tesis/Andriano.pdf> .
9. Visual Paradigm para UML. Visual-Paradigm.com [Online][Cited: Marzo 28,2011.] 2001 <http://www.visual-paradigm.com/product>
10. Fase de Inicio. Flujo de trabajo de requerimientos. Ingeniería de Software 1. Conferencia #4. Curso 2008-2009. 27.[Online] [Cited: Marzo 28,2011.] <http://eva.uci.cu>
11. Objetivos del Análisis y el Diseño. [Online][Cited: Marzo 28,2011.] [http://www.ecured.cu/index.php/Flujo\\_de\\_Trabajo\\_Análisis\\_y\\_Diseño](http://www.ecured.cu/index.php/Flujo_de_Trabajo_Análisis_y_Diseño)

## *Referencias Bibliográficas*

---

---

12. Ingeniería de Requisitos en Aplicaciones para la Web: Un estudio comparativo. Universidad de Sevilla. España y Universidad de Munich y F.A.S.T. Munich, Alemania., GmbH : s.n
13. Ingeniería de Requisitos en Aplicaciones para la Web: Un estudio comparativo. Universidad de Sevilla. España y Universidad de Munich y F.A.S.T. Munich, Alemania., GmbH : s.n.
14. García, Joaquín. Patrones de diseño. Diseño de Software Orientado a Objetos. <http://www.ingenierosoftware.com>. [Online] Mayo 27, 2005. [Cited: Marzo 29, 2011.] <http://www.ingenierosoftware.com/analisisydiseno/patrones-diseno.php>
15. Fundamentos de Ingeniería de Software. <http://www.ingenierosoftware.com>. [Online] [Cited: Marzo 29, 2011.] <http://www.ingenierosoftware.com/analisisydiseno/patrones-diseno.php>
16. Sicilia, Miguel-Ángel. Métrica para calcular la Calidad del Diseño Orientado a Objetos del Software. [Online] Enero 8, 2009. [Cited: Marzo 30, 2011.] <http://cnx.org/content/m17463/latest/>

### Bibliografías

1. Rosario, Jimmy, 2006, "TIC: Su uso como Herramienta para el Fortalecimiento y el Desarrollo de la Educación Virtual". Disponible en el ARCHIVO del Observatorio para la CiberSociedad en <http://www.cibersociedad.net/archivo/articulo.php?art=221>
2. Rosario, Jimmy, 2007, "Las aulas virtuales como modelo de gestión del conocimiento". Disponible en el ARCHIVO del Observatorio para la CiberSociedad en <http://www.cibersociedad.net/archivo/articulo.php?art=231>
3. Álvarez, B.; González, C. y García, N. (2007, Mayo). La motivación y los métodos de evaluación como variables fundamentales para estimular el aprendizaje autónomo. *Red U. Revista de Docencia Universitaria*, Número 2. Consultado (día/mes/año) en [http://www.redu.um.es/Red\\_U/2](http://www.redu.um.es/Red_U/2)
4. Vary, James P., 2000, "Informe de la reunión de expertos sobre laboratorios virtuales" [En Línea] <http://unesdoc.unesco.org/images/0011/001191/119102s.pdf>.
5. Márquez, Y.M., Propuesta de sistema de evaluación del aprendizaje autónomo del idioma Inglés en un entorno virtual de aprendizaje en la Universidad de las Ciencias Informáticas. 2007: p. 150.
6. Metodologías Ágiles en el Desarrollo de Software. *Todo Ágil [Online]* <http://www.willydev.net/descargas/prev/ToDoAgil.pdf>
7. Metodologías de desarrollo de software. Introducción al desarrollo del software. [Online] <http://eva.uci.cu/mod/resource/view.php?id=11361>
8. Garlisco, Jordana, 2008, "Introducción an Open UP - Mug Bs As - Oct 08" [Online] [http://grupos.emagister.com/documento/metodologias\\_agiles\\_\\_introduccion\\_al\\_open\\_up/1046-216995](http://grupos.emagister.com/documento/metodologias_agiles__introduccion_al_open_up/1046-216995)
9. Calero, Manuel Solís., 2003, "Una explicación de la programación extrema (XP)" [Online] <http://www.willydev.net/descargas/prev/explicaxp.pdf>
10. Orallo, Enrique Hernández, "El lenguaje unificado de modelado (UML)", [Online] <http://www.disca.upv.es/enheror/pdf/ActaUML.PDF>
11. De la Vara, José Luis González Sánchez, Juan Díaz, "Derivación de modelos de tareas a partir de modelos BPMN ", [Online] <http://alarcos.inf-cr.uclm.es/pnis/articulos/pnis-07-SanchezJ-DMTBPMN.pdf>
12. Mora, Beatriz FR, García, Félix, Piattini, Mario. Experiencia en transformación de modelos de procesos de negocios desde BPMN a XPDL.: 14.[Online]

## Referencias Bibliográficas

---

<http://kuainasi.ciens.ucv.ve/ideas07/documentos/articulos.../Articulo35.pdf>

13. Visual Paradigm para UML. Visual-Paradigm.com [Online] 2001

<http://www.viseual-paradigm.com/product>

14. Rational Rose Enterprise Edition. [Online]

[http://www.ecured.cu/index.php/Rational\\_Rose\\_Enterprise\\_Edition](http://www.ecured.cu/index.php/Rational_Rose_Enterprise_Edition)

15. Herreros, L. Rosado y J. R. 2005. Nuevas aportaciones didácticas de los laboratorios virtuales y remotos en la enseñanza de la Física. Nuevas aportaciones didácticas de los laboratorios virtuales y remotos en la enseñanza de la Física. [En línea] 2005. <http://www.formatex.org/micte2005/286.pdf>.

16. TORO, A. D.; JIMÉNEZ, B. B., et al. A Requirements Elicitation Approach Based in Templates and Patterns. [Artículo Académico]. Sevilla, España: Disponible en:

<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.92.4047&rep=rep1&type=pdf>.

17. ESCALONA, M. J. y KOCH, N. Requirements Engineering for Web Applications – A Comparative Study. Journal of Web Engineering, vol. 2, nº p. 193-212. Disponible en:

<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.72.911&rep=rep1&type=pdf>.

18. Ingeniería de Requisitos en Aplicaciones para la Web: Un estudio comparativo. Universidad de Sevilla. España y Universidad de Munich y F.A.S.T. GmbH, Munich, Alemania.

19. CORPORATION, I. Ayuda del Rational Unified Process en español. 2006.

20. ÖVERGAARD, G. y PALMKVIST, K. *Use Cases Patterns and Blueprints*. Addison Wesley Professional, 2004, ISBN 0-13-145134-0.

21. Procesos de negocio y ventajas competitivas. 2007 Cited[31/03/11] Disponible en <http://eva.uci.cu>

22. Meléndez, Felipe de la Cruz. La Nueva versión del PMBOK-4ta Edición. Disponible en <http://www.pmi.org.pe>

23. Documento Prototipo Disponible en <http://siv.cenditel.gob.ve/miv/wiki/DocumentoPrototipo>

24. UNED: Universidad Nacional de Educación a Distancia, Departamento de Informática y Automática. Laboratorio virtual. 2010, Enero. [En línea]. Disponible en: <http://lab.dia.uned.es/rllab/contenido/labvirtual.html>