

Universidad de Ciencias Informáticas

Facultad 7

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

**Título: Diseño de una herramienta informática
para la realización de Evaluaciones Económicas
de Inversiones.**

**Autores: Nadier Carlos Luna García
Rubén Matos González**

**Tutores: Ing. Daneysi Granados Rodríguez
Lic. Wilber Chang Ramírez**

Ciudad de la Habana, Junio 2010

“Año 52 de la Revolución”

Declaración de autoría

Declaramos ser los únicos autores de este trabajo y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales del mismo, con carácter exclusivo.

Para que así conste firmo la presente a los 22 días del mes de junio del año 2010.

Firma del Autor

Rubén Matos González

Firma del Autor

Nadier Carlos Luna García

Firma del Tutor

Ing. Daneysi Granados Rodríguez

Firma del Tutor

Lic. Wilber Chang Ramírez

Datos de contacto

Ing. Daneysi Granados Rodríguez:

Ingeniera en Ciencias Informáticas, graduada en la UCI, curso 2006-2007. Profesora de la dirección de producción 2 (IP2), actualmente como parte de su trabajo en esta dirección presta servicios en las oficinas adjuntas al Consejo de Estado en 1ª y B, Vedado. Además Imparte Gestión de Software en la fac 7, en la cual impartió también BD, ISWI y II. Correo electrónico: dgranado@uci.cu.

Lic. Wilber Chang Ramírez:

Licenciado en Economía graduado en la Universidad de Holguín “Oscar Lucero Moya” en el curso 2005-2006. Actualmente trabaja en la empresa Albet Ingeniería y Sistemas S.A. como Especialista Comercial, en la misma empresa laboró como Especialista en Contabilidad durante un año. Es profesor Instructor e impartió las asignaturas de Comercio Electrónico y Administración de Empresas desde sus inicios en la UCI en el 2006 y lo continuó haciendo luego de cambiar de trabajo a la empresa antes mencionada. Correo electrónico: wchang@uci.cu y wchang@albet.cu.

ÍNDICE

INTRODUCCIÓN	1
CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA	5
1.1 ¿QUÉ ES UNA EVALUACIÓN ECONÓMICA DE INVERSIONES PARA PROYECTOS DE INVERSIÓN?	5
1.2 SISTEMA AUTOMATIZADO EXISTENTE VINCULADO AL CAMPO DE ACCIÓN.....	5
1.3 TENDENCIAS ACTUALES	6
1.3.1 METODOLOGÍA DE DESARROLLO DE SOFTWARE.....	6
1.3.2 LENGUAJE DE MODELADO.....	7
1.3.3 HERRAMIENTAS CASE.....	7
1.3.4 SISTEMAS GESTORES DE BASE DE DATOS (SGBD)	8
1.4.5 FUNDAMENTACIÓN DEL LENGUAJE DE PROGRAMACIÓN A UTILIZAR.....	9
1.4.6 ENTORNO DE DESARROLLO INTEGRADO	10
1.4.7 FRAMEWORK UTILIZADO EN LA SOLUCIÓN	11
1.5 CONCLUSIONES	13
CAPÍTULO 2. DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA.	13
2.1 ¿QUÉ ES LA UPI “PROYECTO FUTURO”?.....	13
2.1.1 OBJETIVOS DE LA UPI “PROYECTO FUTURO”	14
2.1.2 PROCESOS CLAVES EN LA ACTIVIDAD INVERSIONISTA	14
2.2 OBJETO DE AUTOMATIZACIÓN	18
2.3 INFORMACIÓN QUE SE MANEJA	19
2.4 PROPUESTA DE SISTEMA	19
2.5 MODELO DE NEGOCIO.....	20
2.5.1 REGLAS DEL NEGOCIO	20
2.5.2 PROCESOS DEL NEGOCIO.....	20
2.5.3 ACTORES DEL NEGOCIO	21
2.5.3.1 ACTORES DE LA EVALUACIÓN ECONÓMICA A PARTIR DE PRESUPUESTO DE IDEAS CONCEPTUALES	22
2.5.3.2 ACTORES DE LA EVALUACIÓN ECONÓMICA A PARTIR DE PRESUPUESTO EJECUTIVO.....	22
2.5.3.3 ACTORES DE LA EVALUACIÓN ECONÓMICA DE UN PLAN DE NEGOCIOS (SE ENTIENDE COMO LA EVALUACIÓN DEL SISTEMA UCI)	22

2.5.4 TRABAJADORES DEL NEGOCIO	22
2.5.5 DIAGRAMA DE CASOS DE USO DEL NEGOCIO	23
2.5.6 REALIZACIÓN DE LOS CASOS DE USOS DEL NEGOCIO.....	23
2.5.6.1 CASO DE USO “EVALUAR ECONÓMICAMENTE UNA OBRA A PARTIR DE PRESUPUESTO DE IDEAS CONCEPTUALES” .	23
2.5.6.2 CASO DE USO “EVALUAR ECONÓMICAMENTE UNA OBRA A PARTIR DE PRESUPUESTO EJECUTIVO”	24
2.5.6.3 CASO DE USO “EVALUAR ECONÓMICAMENTE UN PLAN DE NEGOCIOS”	25
2.6 ESPECIFICACIÓN DE LOS REQUISITOS DE SOFTWARE.....	25
2.6.1 REQUISITOS FUNCIONALES	26
2.6.2 REQUISITOS NO FUNCIONALES	31
2.7 DEFINICIÓN DE LOS CASOS DE USO.....	32
2.7.1 ACTORES DEL SISTEMA.....	32
2.7.2 LISTADO DE CASOS DE USO	33
2.7.3 DIAGRAMA DE CASOS DE USO DEL SISTEMA	36
2.7.4 PATRONES DE CASOS DE USO	37
2.7.5 CASOS DE USOS A DESARROLLAR POR CICLO	38
2.8 CASOS DE USO EXPANDIDOS.....	39
2.9 CONCLUSIONES	50
<u>CAPÍTULO 3. ANÁLISIS Y DISEÑO.....</u>	<u>51</u>
3.1 MODELO DE ANÁLISIS	51
3.1.1 MODELO DE CLASES DE ANÁLISIS	51
3.1.2 ARQUITECTURA DEL SISTEMA	52
3.1.3 PATRONES DE DISEÑO	54
3.2 MODELO DE DISEÑO.....	55
3.2.2 DIAGRAMAS DE INTERACCIÓN	57
3.2.3 DISEÑO DE LA BASE DE DATOS	58
3.2.4 INTERFAZ.....	62
3.2.5 TRATAMIENTO DE ERRORES	62
3.2.6 CONCEPCIÓN DE LA AYUDA	62
3.2.7 SEGURIDAD.....	63
3.3 CONCLUSIONES	63
<u>CONCLUSIONES.....</u>	<u>65</u>
<u>RECOMENDACIONES</u>	<u>66</u>
<u>REFERENCIAS BIBLIOGRÁFICAS</u>	<u>67</u>

BIBLIOGRAFÍA.....68

GLOSARIO DE TÉRMINOS71

INTRODUCCIÓN

Las evaluaciones económicas han sido utilizadas en todo el mundo desde que el hombre sintió la necesidad de hacer canjes y saber si estos eran viables o no, incluso antes de surgir el dinero como medio de cambio para darle valor a las mercancías.

Los recursos financieros son limitados tanto para las empresas como para las personas, por lo que el uso racional y acertado de los mismos es lo que marca la diferencia en cuanto a la supervivencia en el mercado. Se parte entonces de la premisa de que no se invierte dinero sin la seguridad de que se obtendrá un retorno del mismo o este de forma incrementada, igualmente sucede cuando se necesita financiamiento para ejecutar una inversión, pues el que financia necesita una garantía de que su dinero retornará con intereses. En ambos casos, la confirmación está en una evaluación económica.

Las evaluaciones económicas se realizan con diferentes fines, ya sean sobre proyectos de inversión para la construcción de un puente, de un hotel, de un hospital, de una escuela, de una fábrica; o de un producto, de una nueva línea de producción en una fábrica, entre otros.

La evaluación de un proyecto de inversión, tiene por objeto conocer su rentabilidad económica, financiera y social, de manera que resuelva una necesidad humana en forma eficiente, segura y rentable, asignando los recursos económicos con que se cuenta, a la mejor alternativa. En la actualidad una inversión inteligente requiere de un proyecto bien estructurado y evaluado. Este debe indicar la pauta a seguir como la correcta asignación de recursos, igualar el valor adquisitivo de la moneda presente en la moneda futura y estar seguros de que la inversión será realmente rentable. Además de decidir el ordenamiento de varios proyectos en función de su rentabilidad y tomar una decisión de aceptación o rechazo.

La evaluación de proyectos, se ha transformado en un instrumento prioritario, entre los agentes económicos que participan en la asignación de recursos, para implementar iniciativas de inversión; esta técnica, debe ser tomada como una posibilidad de proporcionar más información a quien debe decidir, así será posible rechazar un proyecto no rentable y aceptar uno rentable. La realización de proyectos de inversión es importante para el trabajo multidisciplinario de administradores, contadores, economistas, ingenieros, psicólogos, entre otros, con el objetivo de introducir una nueva iniciativa de inversión, y elevar las posibilidades del éxito.

El planteamiento sistemático, metodológico y científico de proyectos, es de gran importancia en los proyectos de inversión, ya que complementan la visión empírica y la acción empresarial. A nivel de empresa la importancia es tal que el éxito de las operaciones normales se apoya principalmente, en las utilidades que el proyecto genera.

La inversión que se toma en cada empresa sobre la base de la influencia de las decisiones de inversión, puede minimizar costos, tener precios más accesibles, nuevas fuentes de trabajo, entre otras.

En la actualidad, en Cuba algunas empresas llevan a cabo las evaluaciones de las inversiones que realizan en hojas de cálculo Excel, pero este utiliza más de 1450 celdas para ubicar la información de un Estudio de Factibilidad Técnica Económica determinado, lo que implica que el proceso de consulta y llenado de los datos resulte engorroso y propenso a errores.

El documento hecho en dicha hoja de cálculo no permite mantener un historial de evaluaciones realizadas. Si se quisiera obtener un listado de todas las evaluaciones que se encuentran en ejecución, no podría hacerse con el Excel, pues no lo permite. En cuanto a la seguridad, en la hoja de cálculo no cuenta con un sistema que permita el control de acceso a los datos, los cuales pueden ser consultados por cualquier persona ajena o no a esta información. Por otro lado, el documento no cuenta con una funcionalidad que permita generar informes dada la solicitud de una información determinada, además de que la información está propensa a perderse o dañarse.

La Universidad de las Ciencias Informáticas (UCI) es una entidad presupuestada con características muy peculiares. Surgió como una obra más de la Batalla de Ideas. Para la creación de la misma se han tenido que desarrollar hasta la fecha diferentes tipos de inversiones, tanto en el plano constructivo como en el aseguramiento de los recursos humanos, la compra de equipamiento y los servicios que se prestan en la ciudad universitaria.

La Universidad de las Ciencias Informáticas de Cuba (UCI) es la primera universidad de la Batalla de Ideas del pueblo cubano, con el fin de fortalecer el desarrollo del sector de la tecnología de informática y las comunicaciones en Cuba. Tiene como misión formar profesionales comprometidos con su Patria, calificados en la rama de la Informática, a partir de un modelo pedagógico flexible, que vincula dinámicamente y coherentemente el estudio, la producción y la investigación, acorde con las necesidades sociales del país y de otros pueblos hermanos.

Albet S. A. Ingeniería y Sistemas es una Empresa Comercializadora creada en 2005 para dar respuesta a proyectos que se gestaron fundamentalmente en el marco del Convenio Integral de Cooperación con Venezuela. Es la “cara comercial” para todos los proyectos de exportación coordinados a través de la Infraestructura Productiva y en el marco del Programa Nacional para la Producción de Software. Se ejecutan varios proyectos de exportación fundamentalmente en el marco del ALBA. Estos programas promueven la integración y desarrollo los pueblos, los trabajos están orientado hacia sectores donde la UCI exhibe un resultado. Los proyectos de exportación representan anualmente desde el 2004 un importante aporte a la economía del país, demostrando la validez de la idea de que el sustento fundamental de Cuba serán las producciones intelectuales y dentro de ellas la informática. (Portal UCI, 2009)

La UPI “Proyecto Futuro” es una entidad jurídicamente independiente creada el 16 de marzo de 2007, con efectos legales a partir del 9 de marzo de 2007, con posibilidades de manejar cuentas en Moneda Nacional (MN) y Divisa (CUC), capaz de dirigir el proceso inversionista de la UCI y sus Facultades Regionales, con todas las obras que se indiquen desde el Organismo

Rector, según el financiamiento recibido, y con facultades para desempeñar correctamente las funciones que se le asignen.

En la UPI actualmente no se realizan Evaluaciones Económicas de Inversión, lo cual provoca que exista un mal empleo o utilización de los recursos económicos del país. Las inversiones se realizan a ciegas, lo que puede traer consigo pérdidas millonarias, al realizar una inversión que no sea factible económicamente y se puede perder la oportunidad de utilizar los recursos económicos que fueron destinados a esta inversión en otra que pueda ser más rentable.

Este elemento está unido al proceso de reordenamiento del país, donde se establece la no aprobación de inversiones que no estén respaldadas con su respectiva evaluación económica. “El plan 2010 tiene como una de sus premisas que no se admitirán inversiones sin un estudio de factibilidad previamente aprobado”. (Intervención en la sesión de la Asamblea Nacional, 2009)

Debido a la crisis financiera que atraviesa el mundo actualmente, a la necesidad de ahorrar los recursos que posee el país y de invertir con seguridad, se plantea el siguiente **problema científico**: ¿Cómo viabilizar el proceso de Evaluación Económica de Inversiones en la Unidad Presupuestada Inversionista (UPI) “Proyecto Futuro”?

El **objeto de estudio** está compuesto por: los procesos económicos llevados a cabo en la UPI de la UCI. El **campo de acción** se define como el proceso económico para la realización de la Evaluación Económica de Inversiones para los Proyectos de Inversión realizados en la Universidad de las Ciencias Informáticas.

Para resolver el problema identificado se propone el siguiente **objetivo general**: Diseñar una aplicación de escritorio que permita realizar Evaluación Económica de Inversiones para los Proyectos de Inversión en la Universidad de las Ciencias Informáticas.

Para lograr el cumplimiento del objetivo general se proponen las siguientes **tareas de la investigación**:

1. Sintetizar el estudio del estado de arte de las herramientas que realizan Evaluación Económica de Inversiones a nivel nacional e internacional.
2. Analizar el proceso de Evaluación Económica de Inversiones en la Unidad de Proyectos de Inversión (UPI) de la Universidad de las Ciencias Informáticas (UCI).
3. Identificar en el Ministerio de Economía y Planificación los elementos de gastos vigentes para las empresas cubanas.
4. Identificar los elementos de ingresos y gastos en la UPI de la UCI.
5. Asimilar la arquitectura definida por el módulo de Control de Inversiones (ConInver).
6. Realizar el modelo del Negocio.
7. Realizar el levantamiento de requisitos del sistema a desarrollar.
8. Realizar el modelado de casos de uso del sistema.
9. Realizar los artefactos del análisis y diseño del sistema (modelo de análisis y modelo de diseño)

10. Construir el prototipo no funcional del sistema.
11. Documentar el prototipo no funcional del sistema.

En el presente trabajo de diploma se realizará el análisis y diseño de una herramienta para hacer Estudios de Factibilidad Técnica Económica para Proyectos de Inversión, con una base de datos que permita guardar estudios de factibilidad para consultar posteriormente si fuera necesario, además de la organización de la información.

El presente trabajo de diploma está estructurado en tres capítulos, a continuación se describe el contenido de estos:

❖ Capítulo 1: Fundamentación teórica. Incluye un estado del arte de los sistemas u organizaciones que se dedican a realizar estudios de factibilidad a nivel internacional, nacional y de la Universidad, así como tendencias, conceptos fundamentales, tecnologías para realizar estudios de factibilidad, metodologías y aplicaciones en los que se apoya para la solución del problema.

❖ Capítulo 2: Características del sistema. Contiene las principales características del sistema, las causas que originan la situación problemática, y sus consecuencias, los procesos que serán objeto de automatización, la propuesta del sistema y el modelo del negocio a desarrollar, así como sus requerimientos.

❖ Capítulo 3: Análisis y Diseño. Se expone el análisis y diseño de la aplicación informática.

Capítulo 1. Fundamentación Teórica

En el presente capítulo se pretende lograr un basamento teórico sólido acerca de la situación actual que genera la necesidad de realizar este trabajo, del marco en que se desenvuelve y las tendencias actuales en el mundo para resolver problemas como éste. Se hace un estudio minucioso de las tecnologías, métodos y herramientas que serán utilizadas en el desarrollo de este proyecto. Para desarrollar la parte práctica con la calidad adecuada y tener una gama de conocimientos que permitan realizar estudios de factibilidad técnica económica para proyectos de inversión, es preciso dedicar un espacio a un estudio, como el que a continuación se refleja.

1.1 ¿Qué es una Evaluación Económica de Inversiones para Proyectos de Inversión?

La evaluación económica de proyectos de inversión es un proceso sistemático que permite identificar, medir y valorar los costos y beneficios relevantes asociados a una decisión de inversión, para emitir un juicio objetivo sobre la conveniencia de su ejecución desde distintos puntos de vista: económico, privado o social.

Los proyectos de inversión surgen como respuesta a determinadas necesidades humanas y su éxito depende de la importancia de la necesidad a satisfacer. Esto, a su vez, en función del número de personas afectadas y del valor que éstas le asignan al efecto percibido. Desde el punto de vista empresarial, lo que interesa es la posibilidad de que dicho valor corresponda a una determinada capacidad y disposición a pagar; no importando mayormente si el que pagó es un agente privado o público. (Aguilera Vidal & Sepúlveda Palacios, 2005)

1.2 Sistema automatizado existente vinculado al campo de acción

A nivel internacional han sido elaboradas y diseñadas varias soluciones informáticas, con la finalidad de gestionar la información en las empresas, pero en cuanto a la realización de Evaluaciones Económicas para Proyectos de Inversión, sólo EvalAs se acerca a dicho objetivo. A continuación se muestran algunas características de este sistema:

- Permite el ingreso de información de Proyectos de Producción. Datos de: Productos y Producción, Inversiones, Costos Fijos, Costos Variables, Impuestos, entre otros.
- A partir de los datos ingresados, se calcula la Matriz de Flujos de Caja y luego los principales indicadores financieros: VAN (VPN): Valor Actual Neto, Período de Prepago. TIR: Tasa Interna de Retorno, Período de Repago con Descuento, TIR Modificada.
- Los períodos pueden indicarse en años y (desde la versión 1.3) también en meses.
- Manejo de Incertidumbre: se puede realizar Análisis de Sensibilidad y de Escenarios.
- Permite la comparación con otro Proyecto para determinar cuál alternativa es la más rentable.
- Reporte en formato HTML: (incorporado en versión 1.2)

- Puede utilizarse para determinar rentabilidad de proyectos de producción industriales, forestales y agropecuarios. Entre estos últimos pueden citarse emprendimientos de helicultura y cunicultura (se incluyen ejemplos en el software), apicultura, acuicultura, lombricultura, arándanos, hierbas aromáticas y medicinales, huertas, hongos comestibles, entre otros. **(Software Evaluación de Proyectos, s.f.)**

Independientemente de que ese software tiene todas esas funcionalidades, es propietario (contrario a la línea trazada por Cuba en cuanto al tipo de software que debe utilizar); está orientado más bien a sectores netamente productivos y no sigue la metodología de cálculo de las Evaluaciones de Inversiones y Negocio definida por el Ministerio de Economía y Planificación de Cuba.

1.3 Tendencias actuales

Para el desarrollo del sistema de Evaluaciones Económicas en la Unidad Presupuestada Inversionista (UPI) se asimiló la metodología seguida por el módulo de Control de Inversiones (ConInver), además de las herramientas y tecnologías utilizadas, por ser esta investigación parte de uno de los sub-módulos que están en proyecto para ser realizados para beneficio de la UPI.

1.3.1 Metodología de desarrollo de software

El objetivo de un proceso de desarrollo de software es elevar la calidad de este, a través de la transparencia y control sobre su desarrollo y lograr que dichas pautas sean reproducibles en cada proyecto, independientemente de su magnitud. Actualmente, existe una gran cantidad de procesos de desarrollo, agrupados en dos tendencias: los métodos ágiles y los métodos pesados.

1.3.1.1 Rational Unified Process (RUP)

RUP (Proceso Unificado de Desarrollo) es uno de los procesos más generales que existen actualmente, su finalidad no está restringida a guiar desarrollo de software, sino cualquier tipo de proyecto. La estrategia de este proceso es conseguir su objetivo por medio de orden y documentación, lo que lo convierte en el más fiel exponente de los métodos pesados. RUP define cuatro fases (inicio, elaboración, construcción y transición) y dentro de cada una de ellas el equipo de trabajo pasa por todos los flujos que son transversales a las fases, inclusive en varias iteraciones.

Figura 1.1: RUP en dos dimensiones.

El ciclo de vida de RUP se caracteriza por estar:

- a. Dirigido por casos de uso**
- b. Centrado en la arquitectura.**
- c. Iterativo e incremental.**

1.3.2 Lenguaje de modelado

Para el desarrollo de la aplicación se utilizará UML, como el lenguaje con que se modelarán los artefactos que se creen en el proceso de desarrollo del software.

UML es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. UML ofrece un estándar para describir un plano del sistema, incluyendo aspectos conceptuales, tales como procesos de negocios y funciones del sistema, y aspectos concretos, como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables. En fin, UML es el enlace entre quien tiene la idea y el desarrollador, la comunicación es su principal objetivo. **(Popkin Software and Systems)**

Se puede aplicar en una gran variedad de formas para dar soporte a una metodología de desarrollo de software (tal como RUP), pero no especifica en sí mismo qué metodología o proceso usar. El UML está compuesto por diversos elementos gráficos que se combinan para conformar diagramas. Debido a que el UML es un lenguaje, cuenta con reglas para combinar tales elementos.

UML visualiza un sistema desde diferentes perspectivas:

Los **Diagramas de Estructura** enfatizan en los elementos que deben existir en el sistema modelado; incluye, entre otros, Diagrama de clases, Diagrama de componentes, Diagrama de objetos y Diagrama de paquetes.

Los **Diagramas de Comportamiento o Dinámicos** enfatizan en lo que debe suceder en el sistema modelado; ejemplo son el Diagrama de actividades y Diagrama de casos de uso.

Los **Diagramas de Interacción** son un subtipo de diagramas de comportamiento, que enfatiza sobre el flujo de control y de datos entre los elementos del sistema modelado; aquí se puede ver el Diagrama de secuencia y de Diagrama de colaboración. El uso de UML y el trabajo continuo que se ha venido realizando sobre él, lo convierten en una práctica que asegura la especificación de los procesos en el desarrollo del software y es muy favorable su uso como vía de comunicación y documentación. **(Rumbaugh, et al., 2007)**

1.3.3 Herramientas CASE

Las Herramientas CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Computadora) son aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software, reduciendo su coste en términos de tiempo y dinero. Estas herramientas pueden ayudar en muchos de los aspectos del ciclo de vida de desarrollo del software, en tareas como la realización del diseño del proyecto, cálculo de costes, implementación de una parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores, entre otras. **(Visual Paradigm, 2008)**

Entre los principales beneficios que ofrece la utilización de una Herramienta Case se encuentran:

- Mejorar la comunicación entre usuario y especialista, ya que al tener incorporada la visualización de diagramas y de otras herramientas del análisis estructurado, actúa como un elemento que acelera la relación usuario/especialista.
- Permitir la facilidad de revisión de aplicaciones instaladas.
- Ser capaces de generar automáticamente las instrucciones del programa fuente. Esto permite acelerar el tiempo dedicado a la elaboración de programas y asegura además una estructura estándar y consistente para el programa.
- Dar la posibilidad de generar documentación técnica.

Dentro de la gama de herramientas de modelado se encuentran Rational Rose y Visual Paradigm, que son las más usadas en la actualidad. **(Visual Paradigm, 2008)**

1.3.3.1 Visual Paradigm (VP)

Visual Paradigm para UML es un galardonado producto que facilita a las organizaciones el diseño visual de los distintos diagramas a integrar y desplegar sus aplicaciones. Esta herramienta de desarrollo de software ayuda a los equipos de desarrollo en la confección de los distintos modelos que van desde la construcción hasta el despliegue, aumentando al máximo la productividad.

Es una herramienta diseñada para desarrollar software con programación orientada a objetos. Busca reducir la duración del ciclo de desarrollo, brindando ayuda a arquitectos, analistas, diseñadores y desarrolladores; permite el uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación y tiene capacidades de ingeniería directa e inversa.

Presenta un innovador analizador textual, donde se introduce texto extraído de conversaciones con el cliente, se definen actores, entidades, casos de uso disponibles para la generación de artefactos posteriores. Asimismo, posee una herramienta de generación de reportes en formato PDF o HTML configurable y selectiva, se integra con entornos como Eclipse, Hibernate¹ y Subversion², e importa o exporta formatos estándares de otras herramientas CASE como el Rational Rose. **(Visual Paradigm, 2008)**

1.3.4 Sistemas Gestores de Base de Datos (SGBD)

Un SGBD (en inglés: DataBase Management System) puede definirse como un paquete generalizado de software, que se ejecuta en un sistema computacional anfitrión, centralizando el acceso a los datos y actuando de interfaz entre los datos físicos y el usuario. Las principales funciones que debe cumplir un SGBD se relacionan con la creación y mantenimiento de la base de datos, el control de accesos, la manipulación de datos de acuerdo con las necesidades del usuario, el cumplimiento de las normas de tratamiento de datos, evitando redundancias e inconsistencias y manteniendo la integridad. Los SGBD permiten al programador convencional

¹ Hibernate: Herramienta de Mapeo objeto-relacional para la plataforma Java.

² Subversion: Software de sistema de control de versiones.

ahorrarse horas de trabajo dedicadas a la seguridad, gestión de los datos, chequeo de errores, entre otros.

Entre los SGBD comúnmente utilizados en el mundo tenemos MySQL³, Oracle⁴, PostgreSQL⁵, Microsoft SQL Server⁶, Interbase⁷, entre otros. Todos estos presentan un enfoque relacional con un buen basamento matemático centrado en el Álgebra Relacional. **(Soto López & Saborit Ramírez, 2004)**

1.4.4.1 PostgreSQL

Es un magnífico gestor de bases de datos relacional orientado a objetos, es de software libre, se encuentra publicado bajo la licencia BSD⁸ y es multiplataforma. Permite una fácil gestión de los usuarios y de las bases de datos que contenga el sistema. Sirve de soporte a los lenguajes más populares como PHP, C, C++, Java, Python, Ruby⁹, entre otros, y al protocolo de comunicación encriptado por SSL¹⁰. El número de base de datos que puede contener es ilimitado. Se rige por el estándar ANSI-SQL 92/99. (*PostgreSQL*)

Implementa el uso de rollback¹¹, subconsultas y transacciones, haciendo su funcionamiento mucho más eficaz. Tiene la capacidad de comprobar la integridad referencial, así como también la de almacenar procedimientos en la propia base de datos. Además, PostgreSQL, permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos. **(Ventajas de PostgreSQL, 2005)**

1.4.5 Fundamentación del lenguaje de programación a utilizar

La plataforma Java pertenecía a la empresa Sun Microsystem Inc., que en fecha 20 de abril de 2009 fue adquirida por Oracle Corporation, ambas empresas tienen nacionalidad norteamericana, por lo que están sometidas a las normas de exportación y reexportación de los Estados Unidos de América. Estas normas impiden que algunos países, entre ellos Cuba, puedan adquirir productos informáticos.

Java es un lenguaje simple, orientado a objetos, distribuido, intérprete, robusto, seguro, de arquitectura neutra, portátil, de alto desempeño, de hilos múltiples y dinámico..

³ MySQL: SGBD relacional, multihilo y multiusuario con más de seis millones de instalaciones.

⁴ Oracle: Es una herramienta cliente/servidor que se desarrolla para la gestión de Bases de Datos.

⁵ PostgreSQL: SGBD relacional orientada a objetos de software libre, publicado bajo la licencia BSD.

⁶ Microsoft SQL Server: SGBD basado en el lenguaje Transact-SQL.

⁷ Interbase: Sistema de Administración de Base de Datos Relacionales (RDBMS) desarrollada y comercializada por la compañía Borland Software Corporation.

⁸ BSD: Licencia de software otorgada principalmente para los sistemas BSD (Berkeley Software Distribution, en español, Distribución de Software Berkeley). Pertenece al grupo de licencias de software Libre.

⁹ C, C++, Java, Python, Ruby: Lenguajes de programación.

¹⁰ SSL: Sigla en inglés de Secure Sockets Layer, Protocolo de Capa de Conexión Segura. Proporciona cifrado de datos, autenticación de servidores e integridad de mensajes.

¹¹ Rollback: Es una operación que devuelve a la base de datos a algún estado previo.

Análisis de sus características:

Orientado a objetos: Para el programador, esto significa que debe poner atención especial a los datos de la aplicación y a los métodos que manipulan los datos y que no debe pensar estrictamente en términos de procedimientos.

Dinámico y distribuido: En todo momento, una clase de Java se puede cargar en un intérprete de Java en ejecución. Dinámicamente se pueden crear casos de estas clases cargadas. Las bibliotecas de código nativo también se pueden cargar de manera dinámica. Las clases en Java se representan mediante la clase Class; usted puede obtener información dinámicamente acerca de una clase al momento de la ejecución. A Java también se le denomina lenguaje distribuido. Esto significa, simplemente, que proporciona un soporte de alto nivel para redes.

Simple: Los diseñadores de Java intentaron crear un lenguaje que un programador pudiera aprender con rapidez, por lo que el número de constructores del lenguaje se ha mantenido relativamente bajo. Otra meta del diseño fue lograr que el lenguaje fuese familiar a la mayoría de los programadores, para facilitar la migración.

Seguro: Uno de los aspectos más populares de Java es que se trata de un lenguaje seguro. Esto es especialmente importante, dada la naturaleza distribuida de Java. Sin una garantía real de la seguridad, usted no querría descargar código de un sitio al azar en la Internet, ni lo dejaría ejecutar en su computadora. Java se diseñó con la seguridad en mente y proporciona varias capas de controles de seguridad que protegen contra código malicioso; estas capas permiten a los usuarios ejecutar con comodidad programas desconocidos.

Para llevar a cabo el desarrollo de la aplicación, que es objeto de esta investigación, se decidió el uso del lenguaje de programación Java por las ventajas y potencialidades que tiene frente a otros lenguajes, al poder implementarse con independencia de la plataforma y como software libre, permitiendo obtener productos de excelente calidad, en menor tiempo y, por consiguiente, con menores costos. **(Flanagan, 1999)**

1.4.6 Entorno de desarrollo integrado

Un entorno de desarrollo integrado o IDE (acrónimo en inglés de integrated development environment), es un programa informático compuesto por un conjunto de herramientas de programación. Puede dedicarse en exclusiva a un sólo lenguaje de programación o bien, poder utilizarse para varios.

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

Es posible que un mismo IDE pueda funcionar con varios lenguajes de programación. Este es el caso de Eclipse, al que mediante plugins se le puede añadir soporte de lenguajes adicionales. **(Entornos De Desarrollo Integrados, s.f.)**

1.4.6.1 Netbeans

NetBeans es una herramienta de desarrollo para aplicaciones, escrita puramente sobre la base de la tecnología Java, de modo que puede ejecutarse en cualquier ambiente que ejecute Java. Es un producto libre y gratuito, sin restricciones de uso y de código abierto.

Soporta el desarrollo de todos los tipos de aplicaciones Java (J2SE¹², Web, EJB¹³ y aplicaciones móviles). Existe un número importante de módulos para extenderlo. Entre sus características se encuentra un sistema de proyectos basado en control de versiones y refactorización. **(NetBeans)**

Todas las funciones del IDE son provistas por plug-ins, al igual que Eclipse. La versión 6.5 incluye numerosas mejoras y soportes a un nuevo lenguaje. Algunas de las novedades a señalar son:

- Mejoras en la gestión de bases de datos.
- Permite desarrollar fácilmente todo tipo de aplicaciones de escritorio, empresariales, móviles y WEB, con los lenguajes C/C++, PHP, JavaScript¹⁴, Java, Groovy y Ruby, en las plataformas Windows, Linux, Mac OS X y Solaris¹⁵. *(Marioko, 2008)*

Entre sus nuevas características se encuentran: Mejoras al soporte de PHP, JavaScript, Ajax¹⁶, soporte mejorado para Hibernate¹⁷, JPA¹⁸, e integración con Glassfish v3 y MySQL. *(NetBeans.org, 2008)*.

1.4.7 Framework utilizado en la solución

El concepto comúnmente aceptado de un componente es llamado “componente binario”, el cual es un artefacto de software compilado que es integrado dentro de la aplicación en tiempo de ejecución (runtime). El código fuente no es usualmente disponible, por lo tanto, los componentes no pueden ser modificados. Si los componentes están implementados en un lenguaje de programación orientado a objeto, ellos pueden ser extendidos a través de la herencia o delegación. *(Kaiser, 2005)*

¹² J2SE: Siglas en inglés de Java 2 Standard Edition. Edición Estándar. La versión estándar es la más común y cuenta con todo lo necesario para desarrollos de software y acceso a aplicaciones Java.

¹³ EJB: Los Enterprise JavaBeans son una de las API que forman parte del estándar de construcción de aplicaciones empresariales J2EE de Sun Microsystems.

¹⁴ JavaScript: Lenguaje de programación interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas web.

¹⁵ Solaris: Sistema operativo de tipo Unix desarrollado por Sun Microsystems desde 1992 como sucesor de SunOS.

¹⁶ Ajax: Acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML),

¹⁷ Hibernate: Herramienta de Mapeo objeto-relacional para la plataforma Java.

¹⁸ JPA: Es la API de persistencia desarrollada para la plataforma Java EE.

No hay una definición común para los frameworks, pero la mayoría tiene un tema común: la reutilización. Una definición ampliamente aceptada es dada por R. E. Johnson, and B. Foote en 1988 en su publicación “Designing Reusable Classes”:

“Un framework es un conjunto de clases que personifican un diseño abstracto para soluciones de una familia de problemas relacionados...”

Otro punto de vista, según R. E. Johnson y V. F. Russo en “Reusing Object-Oriented Designs” incluye:

“Una clase abstracta es un diseño para un objeto simple. Un framework es el diseño de un conjunto de objetos que colaboran para llevar a cabo un conjunto de responsabilidades. De esta manera los frameworks son diseños a escalas más grandes que las clases abstractas. Los frameworks son una forma de reutilizar el diseño a un nivel superior.” (Johnson & Russo, Reusing Object-Oriented Designs, 1991)

1.4.7.1 Hibernate

Hibernate es un potente framework de mapeo objeto/relacional y servicio de consultas para Java. Es la solución ORM (Object-Relational Mapping) más popular en el mundo Java. Permite diseñar objetos persistentes que podrán incluir polimorfismo, relaciones, colecciones y un gran número de tipos de datos. De una manera muy rápida y optimizada se podrán generar bases de datos en cualquiera de los entornos soportados: PostgreSQL, Oracle, MySQL, entre otros

Sus principales características son:

- Permite expresar consultas utilizando SQL nativo o consultas basadas en criterios.
- Soporta todos los sistemas gestores de bases de datos SQL y se integra de manera elegante y sin restricciones con los más populares servidores de aplicaciones J2EE¹⁹ y contenedores web y por supuesto también puede utilizarse en aplicaciones de escritorio.
- Puede operar proporcionando persistencia de una manera transparente para el desarrollador.
- Soporta el paradigma de orientación a objetos de una manera natural: herencia, polimorfismo, composición y el framework de colecciones de Java.

Se decidió el uso de Hibernate para la implementación de la solución, argumentando que es un poderoso medio para mapear clases de Java a tablas de la base de datos, desacopla la base de datos de las clases de la aplicación. Hibernate también provee con HQL una poderosa vía de comunicación entre el programador y la base de datos, al dotarlo de un lenguaje invariante con respecto a esta y además sintácticamente muy parecido al SQL. Además, es de

¹⁹ J2EE: (Java 2 Enterprise Edition). Define un estándar para el desarrollo de aplicaciones empresariales multicapa diseñado por Sun Microsystems. J2EE simplifica las aplicaciones empresariales basándolas en componentes modulares y estandarizados, proveyendo un completo conjunto de servicios a estos componentes, y manejando muchos de las funciones de la aplicación de forma automática, sin necesidad de una programación compleja.

código abierto y la licencia del producto está eximida de costo. **(Carreras Leyva, y otros, 2009)**

1.5 Conclusiones

Después de un minucioso estudio, no se encontró una aplicación informática que gestionara las evaluaciones económicas de inversiones, según las características que se pedían.

Para efectuar el diseño del sistema propuesto, se hace uso de RUP modelando con UML 2.0 y utilizando como herramienta CASE a Visual Paradigm 6.0, que en su conjunto conforman en la actualidad una de las metodologías más utilizadas en el desarrollo de grandes proyectos.

De las tecnologías que han sido objeto de estudio en este capítulo se seleccionó un grupo de ellas para conformar la propuesta tecnológica y así desarrollar un sistema robusto, multiplataforma y extensible, que permita a los clientes finales un entorno de trabajo amigable y flexible.

- Sistema gestor de base de datos: PostgreSQL.
- Lenguaje de programación: Java.
- Framework: Hibernate.
- Entorno de desarrollo integrado: NetBeans (Versión 6.5).

Capítulo 2. Descripción y análisis de la solución propuesta.

En el presente capítulo se hace un estudio de la Unidad de Proyectos de Inversión (UPI) “Proyecto Futuro”, se enuncian y describen los procesos de negocio, se detallan los que serán objetos de automatización. Así como la información que se maneja, además de describirse los actores y trabajadores que intervienen en el mismo, además de la realización de los diagramas de casos de uso del negocio, diagramas de actividades y el modelo de objeto. También se identifican cuáles son los requisitos funcionales y no funcionales que debe cumplir la aplicación. Se muestra además una descripción detallada de los casos de uso del sistema. De forma general se abordan las características del sistema dejando muy clara la propuesta final que se pretende.

2.1 ¿Qué es la UPI “Proyecto Futuro”?

Conceptualmente, la demanda financiera de la UPI “Proyecto Futuro”, es sufragada con fondos centralizados del Estado Cubano, para el Programa de la Batalla de Ideas, siendo aplicable el mecanismo de solicitud, aprobación y autorización para su otorgamiento, definido por el Ministerio de Economía y Planificación.

La UPI “Proyecto Futuro” es la responsable de que se cumpla la Resolución No 91, del 16/03/06 del MEP para las inversiones que atiende y es la entidad responsable de la inversión desde su concertación inicial hasta su puesta en explotación y terminación total, supervisando y controlando, la marcha adecuada de todo el proceso en sus diferentes etapas.

Tiene como objeto social:

- Normar, planificar, preparar, y controlar el proceso inversionista del Proyecto Futuro, que incluye la Universidad de las Ciencias Informáticas, sus Facultades Regionales y cualquier otro caso que así lo requiera y autorice el Organismo Rector.
- Brindar servicios de dirección integrada y/o administración de proyectos de inversiones en pesos cubanos.
- Prestar servicios de supervisión técnica-ingenieril, dirección facultativa de obras y procuración, evaluación, gestión y provisión de ofertas de suministros en pesos cubanos.
- Prestar servicios de consultoría y asistencia técnica en estimaciones y presupuestos económicos, estudios técnicos económicos de preinversión y postinversión, localización y mercado, en pesos cubanos.

2.1.1 Objetivos de la UPI “Proyecto Futuro”

Tiene como objetivos principales:

- Dirigir y organizar el proceso inversionista de la UCI, desde la perspectiva de áreas funcionales capaces de gestionar y controlar todos los procesos de la inversión desde las ideas conceptuales, proyectos, contrataciones, planeación y ejecución, hasta la creación de un nuevo activo tangible funcional y estético, con la completa observancia de la legalidad.
- Preservar, ahorrar, y utilizar con la mayor eficiencia, calidad y rentabilidad los recursos materiales y financieros puestos a disposición de la actividad.
- Preparar, elaborar y controlar la ejecución del Plan de Inversiones de la UCI, tanto en sus aspectos financieros como del avance físico de su materialización.
- Garantizar la emisión de informaciones periódicas a la dirección de la UCI, al Organismo Rector y a los Órganos correspondientes de la Administración Central del Estado.

2.1.2 Procesos claves en la actividad inversionista

- **Programación:** elaboración y aprobación de ideas conceptuales.
- **Diseño:** elaboración de los proyectos técnico ejecutivos.
- **Preparación técnica:** elaboración de presupuestos, cronogramas de ejecución y balances de equipos y fuerza de trabajo. Garantiza el trabajo de solicitud de financiamiento que es responsabilidad del Inversionista Principal.
- **Control de la ejecución:** incluye el seguimiento detallado de los cronogramas de ejecución, el control de los presupuestos, la certificación para proceder a los pagos y el control de la calidad.
- **Entrega:** garantiza la entrega adecuada del objeto de obra terminado a la propiedad y al sistema de mantenimiento de la institución.

Actualmente es muy difícil llevar de manera efectiva y con la rapidez de respuesta que se requiere todo el control de la información que de forma eficiente requiere el proceso inversionista de una entidad presupuestada. La misma trabaja con el dinero del Estado y no

puede, bajo ningún concepto, excederse de los montos y alcances que le fueron aprobados para el período. **(Carreras Leyva, y otros, 2009)**

2.1.3 Evaluación de Inversiones

Las inversiones a nivel mundial, se evalúan económica y financieramente, existiendo varios métodos para poder realizar su evaluación, permitiendo la toma de decisiones para acometer o no, un Proyecto de Inversión, a partir del aporte de capital que sea capaz de generar la inversión y el rendimiento por peso invertido.

- Evaluación Económica - Se realiza para conocer el valor que aporta el proyecto de inversión, considerando que el 100% del capital invertido es propio, considerándose el presupuesto de inversión, la variación del capital de trabajo necesario para acometer el proyecto, los ingresos que generará dicha inversión al ponerse en explotación, los costos fijos y variables en los que se incurran durante el período a evaluar. Se establece el flujo de caja del proyecto, y se calculan los indicadores económicos (Valor Actual Neto, Tasa Interna de Rentabilidad y Período de Recuperación de la Inversión), que permitirán tomar la decisión de si es rentable invertir o no en ese proyecto.

Debido a la realidad de Cuba, que no tiene acceso al mercado de capitales, es necesario incluir otros dos modelos de evaluación:

- Efecto Balanza de Pago – Donde se realiza un análisis del financiamiento en divisas (Nota: El CUC no es una divisa, es otra moneda nacional. Divisa son aquellas monedas que tienen respaldo en el Mercado de Capitales. Ejemplo: Euro, USD, otras monedas extranjeras).

Analiza cómo la corriente de divisas que genera el proyecto, afecta la Balanza de Pagos del País.

- Modelo Rentabilidad País – Mide la rentabilidad del capital cubano en su conjunto. Solo es aplicable para inversiones con capital extranjero.

No se aplica a empresas exportadoras, con capital 100% cubano.

Por las características del Sistema UCI, solo es aplicable la Evaluación Económica y el Efecto Balanza de Pagos.

En Cuba, existe una Metodología del Ministerio de Economía y Planificación (MEP) para la Elaboración de Estudios de Factibilidad de Inversiones, la cual pudiera convertirse verdaderamente en un instrumento útil, necesario, y constituir una herramienta de trabajo para los inversionistas; pero, debido a la extensión de su contenido y la omisión de temas de interés, es necesaria su simplificación y perfeccionamiento, para que pueda ser utilizada realmente como criterio de selección, en la conformación del Plan de Inversiones del país.

En esto está trabajando la Facultad de Economía de la Universidad de la Habana, en el desarrollo de una nueva metodología, práctica, que se ajuste a las condiciones de la realidad cubana.

2.1.4 Principales Indicadores de rentabilidad

Valor Actualizado Neto (VAN): Se define como el valor actualizado del flujo de ingresos netos obtenidos durante la vida útil económica del proyecto a partir de la determinación por año de las entradas y salidas de divisas en efectivo, desde que se incurre en el primer gasto de inversión durante el proceso inversionista hasta que concluyen los años de operación o funcionamiento de la inversión.

Valor obtenido actualizado a una tasa de interés constante para cada año y mide en dinero corriente el grado mayor de riqueza que generará la inversión.

Una definición más explícita es que el VAN es el valor actualizado del saldo entre el flujo de ingresos y egresos en efectivo generados en un proyecto durante su vida útil. De forma general se puede expresar como:

$$\text{VAN} = -I + \sum_{j=1}^n \frac{FC_j}{(1+k)^j}$$

El proyecto de inversión es viable si el $\text{VAN} > 0$

El VAN se corresponde con el objetivo financiero de la empresa, pues el mismo es el resultado de deducirle al valor actual de los ingresos netos en efectivo generados durante la vida útil de un activo en particular, el valor actualizado del costo de inversión, es decir, que si al VAN se le suma el costo de inversión se obtendría el Valor Actual del Activo (VA).

¿Qué indica que el $\text{VAN} > 0$?

Indica que la inversión produce excedentes superiores, a los que podrían obtenerse invirtiendo la misma cantidad a una tasa de inversión k . En otras palabras, todo proyecto con un VAN positivo permitirá:

1. Remunerar el capital invertido a la tasa de descuento que se establezca y que constituye la eficiencia mínima exigida al proyecto.
2. Recuperar el valor de esta inversión.
3. Tener un superbeneficio o excedente económico por encima de la tasa actualizada y que constituye el VAN del proyecto.

Aspectos que requieren consideración en el cálculo del VAN

- **Valor residual del proyecto:** El período en que se evalúa el proyecto está en correspondencia con su vida útil económica, existen determinados componentes del costo de inversión que mantienen su valor o parte de su valor al final del proyecto. Ejemplo: terrenos, capital de trabajo, valor remanente de los activos fijos, etc.
- **Tratamiento de la Depreciación:** La depreciación es una de las partidas de gastos por lo que no se debe considerar en el cálculo del VAN, pues es un procedimiento contable que no refleja ningún movimiento de efectivo. Esta se considera en el valor de Costo Inicial de la Inversión.
- **Sustitución de equipos:** Al determinar los Flujos de Caja del proyecto, debe considerarse la sustitución de aquellos equipos que tienen una vida útil inferior a la vida útil económica estimada para el conjunto del proyecto. Ejemplo: equipos de computación, de transporte, etc.
- **Actualización de los costos de inversión:** Cuando el período de ejecución de la inversión es superior a un año se requerirá actualizar también el costo de la inversión. En este caso se puede considerar como año 0 el momento en que se inicia el proyecto.

Tasa Interna de Retorno (TIR): Mide hasta qué punto puede subir la tasa de descuento y aún el proyecto sigue siendo rentable financieramente.

En otros términos se dice que la TIR corresponde a la tasa de interés que torna cero el VAN de un proyecto, anulándose la rentabilidad del mismo.

Deberá ser mayor o igual a la tasa de actualización: tasa de interés + riesgo + inflación.

Se define como aquella tasa de actualización o descuento, que hace cero la rentabilidad absoluta neta de la inversión. Es decir, aquella tasa de descuento que iguala el valor actual de la corriente de cobros con el valor actual de la corriente de pagos.

El procedimiento general consiste en utilizar aquellas tasas de actualización que aproximen lo más posible el VAN a cero hasta llegar a que el VAN sea negativo. La TIR, se encontrará entre esas dos tasas.

$$TIR = k_1 + \frac{VAN_p (k_2 - k_1)}{VAN_p + |VAN_n|}$$

La TIR representa la rentabilidad general del proyecto y el criterio de selección corresponderá al proyecto que tenga la mayor TIR y además sea mayor a la tasa de actualización del VAN.

- La diferencia entre el Flujo de Caja anual y la TIR en cada período representa la Recuperación Económica de la Inversión.

➤ La TIR representa una medida de la Rentabilidad Relativa Bruta Anual por unidad monetaria comprometida en el proyecto. Relativa porque se define en tanto por ciento y bruta porque a la misma le falta por deducir el costo de financiación de los capitales invertidos en el proyecto (k).

Período de Recuperación (PR): Mide el tiempo que transcurrirá desde la puesta en explotación de la inversión, hasta que se recupere el capital invertido en el proyecto.

Este indicador mide el número de años que transcurrirán desde la puesta en explotación de la inversión, para recuperar el capital invertido en el proyecto mediante las utilidades netas del mismo.

En otros términos se dice que es el período que media entre el inicio de la explotación hasta que se obtiene el primer saldo positivo o período de tiempo de recuperación de una inversión.

Para su cálculo se partirá de los siguientes criterios:

- Se establecerá el período de recuperación a partir del momento en que el proyecto entra en operación.
- Se partirá de las utilidades netas generadas por el proyecto.
- A las utilidades netas se le adicionará la depreciación
- De utilizarse financiamiento ajeno se excluirán de los costos de los gastos financieros.

$$PR = t_n + \frac{|SA_1|}{|SA_1| + SA_2} - m$$

Estos indicadores sólo han sido mencionados para que se conozcan, aunque serán utilizados en la implementación de esta investigación. (Ministerio de Economía y Planificación, 2001)

2.2 Objeto de automatización

Este sistema tiene como objetivo automatizar la metodología de cálculo que se seguirá de forma manual para realizar la Evaluación Económica de Inversiones a los Proyectos de Inversión en la UPI.

La Evaluación Económica de Inversiones se le realizan a dos tipos de inversiones: obras y planes de obras.

El proceso comienza cuando el inversionista recibe del Departamento de Preparación el Presupuesto de Inversión, el cual se divide en 3 componentes en función de una moneda base y otra complementaria en un período determinado: Construcción y montaje, Equipos y Otros. Posteriormente, el Departamento de Preparación define los ingresos (proporcionado por

ALBET) del sistema. Teniendo en cuenta los tipos de proyectos, se describen de la siguiente manera.

La UCI define el destino final de la obra y ALBET define los datos de comercialización de dicha obra. La UCI define los gastos de explotación (partida de gastos de la UCI). Con las partidas de ingresos y gastos, el inversionista se encarga de confeccionar la tabla del estado de resultados y luego procede a efectuar el flujo de caja. Luego se calculan los indicadores económicos VAN, TIR y PAYBACK, los cuales indican si el proyecto económicamente es factible o no. Si no es factible, se modifica el Presupuesto de inversión y se actualizan los cálculos. (UPI “Proyecto Futuro”)

Este proceso puede desarrollarse para obras específicas o para planes de inversión, los cuales incluirían programas que estarían conformados por obras.

2.3 Información que se maneja

- Presupuesto de inversión.
- Partida de gastos.
- Partida de ingresos.
- Plan de inversiones.

2.4 Propuesta de sistema

Teniendo en cuenta que la información que se genera es manipulada de forma manual y no se dispone de una herramienta que permita hacer la Evaluación Económica de Inversiones a los Proyectos de Inversión de una forma más sencilla, se propone el desarrollo de un sistema informático que optimice la elaboración y análisis de la Evaluación Económica de Inversiones a los proyectos de inversión de la UPI.

Esta aplicación incluirá funciones que le permitirán al inversionista:

- Mostrar flujos de caja.
- Calcular indicadores de factibilidad (VAN, TIR, PAYBACK).
- Mostrar estado financiero.
- Mostrar análisis por elementos y subelementos de gastos y por partidas de ingreso.
- Realizar análisis en diferentes monedas.

A partir de los resultados obtenidos de los diferentes análisis de los indicadores económicos, se podrá determinar si una inversión es económicamente factible o no.

Esta aplicación tendrá por nombre “Evaine”, asociado a la unión de las palabras Evaluación de Inversiones y Negocios.

2.5 Modelo de Negocio

El modelo del negocio es una técnica para comprender los procesos del negocio de la organización. El objetivo fundamental del mismo es identificar los casos de uso del software y las entidades relevantes que este debe soportar de forma que se podría modelar solo lo necesario para comprender el contexto.

2.5.1 Reglas del negocio

Las reglas de negocio describen políticas que deben cumplirse o condiciones que deben satisfacerse, por lo que regulan algún aspecto relacionado con el mismo. El proceso de especificación implicó identificarlas dentro del negocio, evaluar si son relevantes dentro del campo de acción que se está modelando e implementarlas en la propuesta de solución. En este caso se identificaron las siguientes reglas que debe seguir la aplicación que se desarrolle, a fin de respetar y garantizar las restricciones que existen en el negocio:

- El proceso comienza cuando el inversionista recibe del Departamento de Preparación el Presupuesto de Inversión.
- El Director sólo tendrá acceso a generar los reportes.
- El Departamento de Control tiene acceso a todas las funcionalidades del sistema.
- Una vez que el estado de una evaluación es terminado, sólo el Departamento de Control puede cambiarlo.
- Si el usuario escoge trabajar sólo la moneda base, el sistema debe bloquear la columna de la moneda complementaria.

2.5.2 Procesos del negocio.

Los procesos identificados en el negocio fueron:

- ❖ Evaluar económicamente una obra a partir de presupuesto de ideas conceptuales.
- ❖ Evaluar económicamente una obra a partir de presupuesto ejecutivo.
- ❖ Evaluar económicamente un plan de negocios.

La UCI solicita la creación de una obra o de un plan de obras y la UPI se encarga de hacerle la evaluación. Para hacer dicha evaluación, tanto a la obra como en un plan de obra, se siguen pasos similares, pues un conjunto de obras es considerado un plan de obras.

Pasos para evaluar una obra:

Evaluación económica a partir de presupuesto de ideas conceptuales: comienza cuando el Departamento de Preparación de la Unidad Presupuestada Inversionista (UPI) realiza la Tarea Técnica de la Obra, donde se debe definir con la UCI el destino que tendrá dicha

edificación y definir con ALBET, la proyección de los ingresos que generará dicha inversión, al ponerse en explotación. Posteriormente la Subdirección Económica (SDE) de la Unidad Presupuestada Inversionista le entrega al Departamento de Preparación el presupuesto de gastos de la entidad, este es recibido por el jefe del departamento. Luego de recibido el presupuesto inicial, los especialistas entregan al Especialista Superior del departamento, el presupuesto de ideas conceptuales, a partir de los requerimientos dados por la universidad para la obra en cuestión y este realiza la evaluación económica correspondiente para la obra.

Si la evaluación da negativa se procede a revisar con la UCI las ideas conceptuales del proyecto y con ALBET, la proyección de los ingresos que generará dicha inversión. Se devuelve el presupuesto de gastos a la Subdirección Económica (SDE) de la Unidad Presupuestada Inversionista para revisar la posibilidad de reducir gastos. Si la evolución da positiva se decide pasar a proyecto ejecutivo y se le entrega la obra a la Dirección Integrada de Proyecto que corresponda.

Evaluación económica a partir de presupuesto ejecutivo: comienza cuando el Departamento de Preparación entrega a la DIP el proyecto ejecutivo y el presupuesto de gastos estimados en el análisis preliminar y además de la proyección de los ingresos que generará dicha inversión y la evaluación económica a partir del presupuesto de ideas conceptuales. Los especialistas de la Dirección Integrada de Proyecto elaboran el presupuesto correspondiente para la obra, según las especialidades y se lo entregan al Especialista Superior de la Dirección Integrada de Proyecto. Este realiza la evaluación económica para la obra.

Si la evaluación da negativa se procede a revisar diferentes aspectos que permitan disminuir los costos de la inversión además de los gastos de la UCI, UPI y ALBET, para esa obra. Si la evaluación da positiva se propone la obra al Departamento de Control para su inclusión en el plan de inversiones de la entidad y su evaluación en conjunto.

Pasos para evaluar un plan de obras:

Evaluación económica de un plan de negocios (se entiende como la evaluación del sistema UCI): comienza cuando todas las Direcciones Integradas de Proyecto proponen las obras que su evaluación fue positiva al Departamento de Control quien confecciona el plan de inversiones para el período a analizar. Se le solicita a las tres entidades que conforman el sistema UCI (UCI, UPI, ALBET) el presupuesto de gastos para el período. Posteriormente ALBET entrega al Dpto. de Control su proyección de ventas para el período. El departamento de control evalúa la factibilidad del plan de negocios.

Si la evaluación da negativa se procede a revisar las obras que componen el plan, las ventas y los gastos del sistema. Si la evaluación da positiva se elabora el informe y presenta el plan de inversiones para el año al MEP para su aprobación.

2.5.3 Actores del Negocio

Un actor del negocio es un individuo, grupo, entidad, organización, máquina o sistema de información externo que interactúa con el negocio y desempeña un rol determinado, no representa un usuario físico, ya que varios usuarios físicos pueden realizar el mismo papel en el negocio, por otro lado, un mismo usuario puede actuar como diferentes actores. Los involucrados con lo descrito son:

2.5.3.1 Actores de la Evaluación económica a partir de presupuesto de ideas conceptuales

Actores del Negocio	Justificación
UCI	Define el destino que tendrá dicha edificación y los gastos de operaciones en los que incurrirá, para la inversión a evaluar.

2.5.3.2 Actores de la Evaluación económica a partir de presupuesto ejecutivo

Actores del Negocio	Justificación
UCI	Revisión de los gastos de operaciones, en caso de ser necesario.

2.5.3.3 Actores de la Evaluación económica de un plan de negocios (se entiende como la evaluación del sistema UCI)

Actores del Negocio	Justificación
UCI	Define los gastos de operaciones en los que incurrirá, así como la proyección de ventas, para el período.

2.5.4 Trabajadores del Negocio

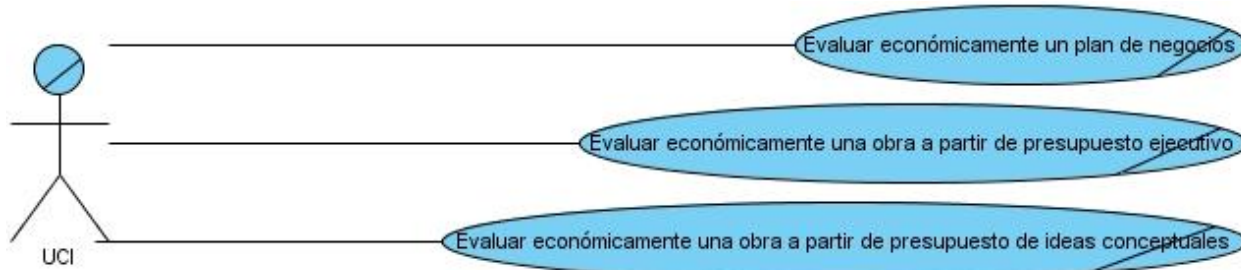
Un trabajador del negocio representa a personas o sistema dentro del negocio que son los que realizan las actividades que están comprendidas dentro de un caso de uso. Estos trabajadores están dentro de la frontera del negocio, son los que por lo general en un futuro se convertirán en usuarios del sistema que se quiere construir. Los involucrados con lo descrito son:

Trabajadores del Negocio	Justificación
Especialistas DIP	Realiza la evaluación de la obra y estima el presupuesto de inversión de las obras de un plan, a partir del presupuesto ejecutivo.

Director del DIP	Revisa, aprueba y presenta los resultados del estudio, al Departamento de Control.
Departamento de Control	Evalúa el plan de inversiones y prepara toda la información de salida que se requiere.
Subdirección económica	Estima el presupuesto de gastos de la UPI y se lo informa la DIP, al Dpto. de Control y al Dpto. de Preparación.
Departamento de Preparación de la UPI	Obtiene los datos necesarios para la realización de la evaluación económica; realiza el presupuesto de ideas conceptuales y realiza la evaluación económica de la obra a partir del presupuesto de ideas conceptuales.
Director General de la UPI	Presenta la Evaluación Económica de la Inversión a la UCI para su aprobación.

2.5.5 Diagrama de Casos de Uso del Negocio

Se representan gráficamente los casos de uso del negocio identificados y su interacción con los actores a través del siguiente diagrama:



2.5.6 Realización de los casos de usos del negocio

La realización de un caso de uso de negocio muestra cómo colaboran los trabajadores y entidades de negocio para ejecutar el proceso.

2.5.6.1 Caso de Uso “Evaluar económicamente una obra a partir de presupuesto de ideas conceptuales”.

Caso de Uso:	“Evaluar económicamente una obra a partir de presupuesto de ideas conceptuales”
Actores:	UCI, ALBET

Trabajadores:	Departamento de Preparación de la UPI, Subdirección económica, Especialista Superior DIP.
Resumen:	Comienza cuando el Departamento de Preparación de la UPI realiza la Tarea Técnica de la Obra. La subdirección económica de la UPI le entrega al departamento de preparación el presupuesto de gastos de la entidad, este es recibido por el jefe del departamento. Los especialistas entregan al Especialista Superior del departamento el presupuesto de ideas conceptuales, a partir de los requerimientos dados por la universidad para la obra en cuestión y este realiza la evaluación económica correspondiente para la obra.

Diagrama de Actividades

Los casos de uso del negocio consisten en la descripción de la de secuencias de actividades que, en conjunto, producen algo para el actor del negocio. El proceso consiste en un flujo básico de una o más alternativas. La estructura del mismo se describe gráficamente con la ayuda de un diagrama de actividad.

Diagrama de Actividades – Caso de Uso “Evaluación económica a partir de presupuesto de ideas conceptuales”.

Diagrama de clases del modelo de objetos

El modelo de objetos del negocio es un modelo interno de un negocio, consiste en describir cómo cada caso de uso del negocio es llevado a cabo por parte de un conjunto de trabajadores que utilizan un conjunto de entidades del negocio y unidades de trabajo.

Diagrama de Clases del Modelo de Objetos – Caso de Uso “Evaluación económica a partir de presupuesto de ideas conceptuales”.

2.5.6.2 Caso de Uso “Evaluar económicamente una obra a partir de presupuesto ejecutivo”

Caso de Uso:	“Evaluar económicamente una obra a partir de presupuesto ejecutivo”
Actores:	UCI, ALBET
Trabajadores:	Departamento de Preparación de la UPI, Especialistas DIP, Especialista Superior DIP.
Resumen:	Comienza cuando el departamento de preparación entrega a la DIP el proyecto ejecutivo, el presupuesto de gastos estimados en el análisis preliminar, la proyección de los ingresos que generará dicha inversión y la

	evaluación económica a partir del presupuesto de ideas conceptuales. Los especialistas de la DIP elaboran el presupuesto correspondiente para la obra, según las especialidades y se lo entregan al Especialista Superior de la DIP.
--	--

Diagrama de Actividades – "Caso de Uso Evaluación económica a partir de presupuesto ejecutivo".

Diagrama de Clases del Modelo de Objetos – Caso de Uso "Caso de Uso Evaluación económica a partir de presupuesto ejecutivo".

2.5.6.3 Caso de Uso "Evaluar económicamente un plan de negocios"

Caso de Uso:	"Evaluar económicamente un plan de negocios"
Actores:	UCI, ALBET
Trabajadores:	Departamento de Control, DIP.
Resumen:	Comienza cuando todas las DIP proponen las obras que su evaluación fue positiva al Departamento de Control quien confecciona el plan de inversiones. Se le solicita a las tres entidades que conforman el sistema UCI (UCI, UPI, ALBET) el presupuesto de gastos para el período. ALBET entrega al Dpto. de Control su proyección de ventas para el período. El departamento de control evalúa la factibilidad del plan de negocios.

Diagrama de Actividades – "Evaluación económica de un plan de negocios".

Diagrama de Clases del Modelo de Objetos – Caso de Uso "Evaluación económica de un plan de negocios".

2.6 Especificación de los requisitos de software

La fase de especificación de requisitos es en la cual debe conseguirse el catálogo de requisitos del sistema que englobe: la definición de los objetivos del sistema, los requisitos de almacenamiento de información, la descripción de los actores del sistema, los requisitos funcionales, descritos a través de los casos de uso, los requisitos de interacción, en lo que se recogerá el sistema de navegación de la aplicación, la interacción con el usuario y los requisitos no funcionales.

El proceso de definir y especificar los requerimientos del software consiste en establecer los servicios que un producto de software debe cumplir; proceso que lleva a la construcción de un documento conocido como Especificación de Requerimientos del Software, esencial para lograr un producto de alta calidad.

Obtener los Requerimientos del Software es paso esencial para la construcción de un producto de software para el posterior desarrollo de las etapas. Un error en este proceso, traería graves consecuencias en la obtención de un producto, pues podría no cumplir las expectativas del cliente.

2.6.1 Requisitos Funcionales

Los Requerimientos Funcionales son capacidades o condiciones con las que debe cumplir el producto a elaborar y están fuertemente ligados a las opciones del programa, no alteran la funcionalidad del software, se mantienen invariables sin importar con que cualidades o propiedades se relacionen.

A partir del estudio y las investigaciones que se realizaron de los procesos del negocio, se obtuvieron una serie de requerimientos funcionales que ha de cumplir el software:

R1. Gestionar Usuario

El sistema debe permitirle al Dpto. de Control gestionar sus usuarios. (Dtor., Especialista Superior e Inversionista.)

R1.1 Registrar Usuario

El sistema debe permitirle al Dpto. de Control introducir la información del usuario nuevo:

1. Nombre.
2. Apellidos.
3. Usuario.
4. Contraseña.
5. Cargo.
6. DIP.

Al registrar un usuario su estado será activo, pero esto no debe ser especificado por el Dpto. de Control.

R1.2 Modificar Usuario

El sistema debe permitirle al Dpto. de Control modificar la información de sus usuarios:

1. Contraseña.
2. Nombre.
3. Apellidos.

4. Cargo.
5. DIP.
6. Estado.

R2. Autenticar Usuario

El sistema debe permitir autenticar a los usuarios, para que estos tengan acceso a las funcionalidades asignadas a este usuario.

R3. Activar/Desactivar Usuario

El sistema debe permitirle al Dpto. de Control activar o desactivar los usuarios que están registrados en el sistema. Los usuarios desactivados no podrán acceder al sistema, sólo los que estén activados lo podrán hacer, por lo que, el Dpto. de Control es el único que permite el acceso al sistema:

R4. Listar Usuarios

El sistema debe permitirle al Dpto. de Control verificar la información de los usuarios y listarlos:

1. Nombre.
2. Apellidos.
3. Contraseña.
4. Usuario.
5. Cargo.
6. DIP.
7. Estado.

R5. Gestionar Evaluación de Inversiones

El sistema debe permitirle al Inversionista gestionar la información requerida de las Evaluaciones Económicas.

R5.1 Crear Evaluación de inversiones

El sistema debe permitirle al Inversionista crear los estudios de factibilidad, asignarlo a la DIP que le corresponda.

1. Nombre.

2. Código de la Evaluación de Inversiones.
3. Fecha de comienzo.
4. Estado de la Evaluación de Inversiones (Activo, Terminado, Anulado).
5. DIP.
6. Tipo de inversión según su naturaleza (Nueva, Ampliación, Remodelación).
7. Tipo de inversión según su destino de explotación (Tecnológica, Turística, Educación, Salud, Vivienda, Industrial, Logística, Cultural, Otros).
8. Categoría nominal (Nominal, No Nominal).
9. Tasa de impuesto.
10. Costo de oportunidad de capital.
11. Elaborado por.
12. Mes inicio. (si es evaluado por meses)
13. Año de inicio

R5.2 Modificar Evaluación de inversiones

El sistema debe permitirle al Inversionista actualizar los datos de la Evaluación.

1. Nombre.
2. Estado de la Evaluación de Inversiones (Activo, Terminado, Anulado)
3. Fecha de terminación.
4. Descripción.

Nota: Una vez que el estado de la evaluación sea terminado, no se puede cambiar la fecha de terminación.

R6. Listar Evaluación de inversiones

El sistema permite listar todas las evaluaciones que han sido creadas en el mismo.

R7. Buscar Evaluación de inversiones

El sistema debe permitirle al inversionista realizar una búsqueda antes de actualizar la evaluación. El criterio que tendrá esta búsqueda será por estado, por fecha, por nombre y por código. El sistema debe incluir la posibilidad de generar un reporte de esa evaluación (todos los criterios de búsquedas mencionados, más indicadores de factibilidad).

R8. Generar Reporte de Estado Financiero

El sistema debe mostrar una ventana donde se debe seleccionar el tipo de análisis que se desea hacer (por gastos, por ingresos, estado de resultados).

R9. Generar Reporte Indicadores de Factibilidad

El sistema debe mostrar una ventana donde se muestre los indicadores de factibilidad (VAN, TIR y Payback) de la inversión que esté abierta.

R10. Generar Reporte de Evaluaciones de inversiones

El sistema debe brindar la posibilidad de escoger las Evaluaciones según los estados y de ellas mostrar el estado de los tres indicadores hasta el momento, más un comentario que el inversionista desee agregar.

R11. Gestionar monedas a utilizar

El sistema debe permitir gestionar las monedas a utilizar en el sistema. Las monedas se especificarán en moneda base y en moneda complementaria.

R11.1. Adicionar Moneda

El sistema debe permitir añadir cuantas monedas desee el usuario.

R11.2. Listar catálogo de Moneda de Cambio

El sistema debe dar la posibilidad de listar los tipos de monedas con sus respectivas tasas de cambio.

R11.3. Tipo de Cambio

El sistema debe dar la posibilidad de establecer las respectivas tasas de cambio.

R11.4. Selección de Moneda Base/Complementaria

El sistema debe permitir escoger cuál será la moneda base y cuál será la complementaria (en caso de que se necesite).

R12. Gestionar Capital de trabajo

- Variación del capital de trabajo.

R13. Gestionar ingresos

El sistema debe permitir introducir las partidas de ingresos que se van a utilizar.

R14. Buscar por DIP

El sistema permite al Especialista realizar una búsqueda mediante la DIP a la que pertenece la Evaluación Económica para buscar la misma y realizarle alguna actualización.

R15. Gestionar costos y gastos

El sistema debe permitir introducir las partidas de gastos que se van a utilizar.

1. Nombre del gasto.
2. Código del gasto.
3. Importe en Moneda Base.
4. Importe en Moneda Complementaria.
5. Clasificación:
 - Fijo o Variable.
 - Directo o Indirecto
 - Administración, Operación o Distribución y Ventas.
 - Misión, Albet, UCI o UPI.
6. Total del gasto.

R16. Gestionar inversiones

1. Construcción y Montaje
2. Equipos
3. Otros.

R17. Gestionar Depreciación

1. Construcción y Montaje
2. Equipos.

Los siguientes parámetros son comunes para ambos:

- Costo.
- Tasa de depreciación.
- Depreciación anual.
- Vida útil.

- Horizonte temporal.
- Depreciación acumulada.
- Valor residual.

2.6.2 Requisitos no Funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido y confiable. A continuación se muestran los requisitos no funcionales.

Apariencia o interfaz externa:

- Diseño sencillo, con pocas entradas, permitiendo que no sea necesario mucho entrenamiento para utilizar el sistema.
- El producto debe ser legible y con colores adecuados, agradables y poco llamativos.
- Diseño perfectamente encuadrado para resoluciones de 1024x768, pero preparado para verse en otras resoluciones.

Usabilidad:

- El sistema podrá ser usado por cualquier persona que posea conocimientos básicos en el manejo de la computadora y de aplicaciones de escritorio en sentido general.

Rendimiento:

- Tiempos de respuestas rápidos, al igual que la velocidad de procesamiento de la información.

Soporte:

Se requiere un servidor de bases de datos con las siguientes características:

- Soporte para grandes volúmenes de datos y velocidad de procesamiento.
- Tiempo de respuesta rápido en accesos concurrentes.
- Tecnología libre.
- Por parte del cliente se requiere soporte Java instalado, Versión 6.0 o superior.

Portabilidad:

- Necesidad de que el sistema sea multiplataforma.

Seguridad:

- Identificar al usuario antes de que pueda realizar cualquier acción sobre el sistema.
- Garantizar que la información sea vista únicamente por quien tiene derecho a verla.
- Garantizar que las funcionalidades del sistema se muestren de acuerdo al nivel de usuario que esté activo.

- Protección contra acciones no autorizadas o que puedan afectar la integridad de los datos.
- Verificación sobre acciones irreversibles (eliminación de datos).

Legales:

- El empleo de este producto no debe violar ninguna ley o licencia por lo que la plataforma escogida para el desarrollo de la aplicación, está basada en la licencia GNU/GPL.
- Este documento y la aplicación pertenecen a la UPI “Proyecto Futuro”.

Confiabilidad:

- La herramienta de implementación a utilizar tiene soporte para recuperación ante fallos y errores.

Funcionalidad:

- Mínima cantidad de vistas para ejecutar todas las funciones posibles (preferentemente que estén relacionadas).
- Operaciones realizadas mediante el teclado.

Software:

- Java.
- PostgreSQL.

Funcionalidad:

- Requerimientos para una estación de trabajo: 256Mb RAM (Recomendado 512Mb), 1GHz, 10Gb HDD.
- Requerimientos para un servidor: 512Mb RAM (Recomendado 1Gb RAM o superior), 1GHz o superior, 60Gb HDD.

2.7 Definición de los casos de uso

2.7.1 Actores del Sistema

Los actores de sistema pueden representar el rol que juega una o varias personas, un equipo o un sistema automatizado y pueden intercambiar información con él o ser recipientes pasivos de información. Los actores suelen corresponderse con los trabajadores o actores del negocio. Teniendo en cuenta todos los requerimientos planteados, se definen varios roles para diferenciar el nivel de acceso al sistema: Dpto. de Control, Director, Especialista.

Actores del Sistema	Justificación
Director	Persona que accede al sistema para obtener de él, todos los tipos de reportes.

Especialista	Es el encargado de generar los diferentes tipos de evaluaciones económicas. Tiene acceso a todos los tipos de reportes.
Departamento de Control	Es el administrador del sistema y accede a todas las funcionalidades del mismo.

2.7.2 Listado de casos de uso

Tabla 2.8: Breve descripción del caso de uso del sistema “Gestionar Usuarios”

CU_1	“Gestionar Usuarios”
Actor	Dpto. de Control.
Descripción	Permite al Dpto. de Control registrar un nuevo usuario y actualizar alguno ya registrado.
Referencia	R1, R1.1, R1.2, R3.

Tabla 2.9: Breve descripción del caso de uso del sistema “Autenticar Usuario”

CU_2	“Autenticar Usuario”
Actor	Usuario (Especialista, Director, Dpto. de Control).
Descripción	Permite a los usuarios ya registrados en el sistema por el Dpto. de Control autenticarse en la aplicación con el rol definido, permitiendo que solo tengan acceso a los recursos que le son permitidos según su rol.
Referencia	R2.

Tabla 2.10: Breve descripción del caso de uso del sistema “Listar Usuarios”

CU_3	“Listar Usuarios”
Actor	Dpto. de Control.
Descripción	Permitirá visualizar todos los usuarios registrados en el sistema, mostrando los datos de estos. Permite que este listado sea guardado en formato Excel o PDF para una dirección especificada, además de poder imprimirlo en uno de estos dos formatos.
Referencia	R4.

Tabla 2.11: Breve descripción del caso de uso del sistema “Gestionar Evaluación de Inversiones”

CU_4	“Gestionar Evaluación de Inversiones”
Actor	Dpto. de Control.
Descripción	Permite gestionar la información de las Evaluaciones de Inversión, tales como, crear una

	evaluación y actualizar los datos introducidos por Dpto. de Control.
Referencia	R5, R5.1, R5.2, R7, “Buscar Evaluación de inversiones” <<extend>>.

Tabla 2.12: Breve descripción del caso de uso del sistema “Listar Evaluación de Inversiones”

CU_5	“Listar Evaluación de Inversiones”
Actor	Dpto. de Control.
Descripción	Permite al Dpto. de Control acceder al listado de todas las Evaluaciones de Inversión que tiene.
Referencia	R6.

Tabla 2.13: Breve descripción del caso de uso del sistema “Generar Reporte de Estado Financiero”

CU_6	“Generar Reporte de Estado Financiero”
Actor	Dpto. de Control, Director, Especialista.
Descripción	Permite obtener un reporte para hacer el tipo de análisis por gastos, por ingresos, estado de resultados
Referencia	R8.

Tabla 2.14: Breve descripción del caso de uso del sistema “Generar Reporte Indicadores de Factibilidad”

CU_7	“Generar Reporte Indicadores de Factibilidad”
Actor	Dpto. de Control, Director, Especialista.
Descripción	Permite obtener un reporte donde se muestren los indicadores de factibilidad por cada Evaluación Económica.
Referencia	R9.

Tabla 2.15: Breve descripción del caso de uso del sistema “Generar Reporte de Evaluaciones de inversiones”

CU_8	“Generar Reporte de Evaluaciones de inversiones”
Actor	Dpto. de Control, Director, Especialista.
Descripción	Permite escoger las Evaluaciones según los estados y de ellas mostrar el estado de los tres indicadores hasta el momento, más un comentario que el inversionista desee agregar.
Referencia	R10.

Tabla 2.16: Breve descripción del caso de uso del sistema “Gestionar monedas a utilizar”

CU_9	“Gestionar monedas a utilizar”
Actor	Dpto. de Control, Especialista.
Descripción	Permite gestionar las monedas a utilizar en el sistema. Las monedas se especificarán en moneda base y en moneda complementaria.
Referencia	R11.

Tabla 2.17: Breve descripción del caso de uso del sistema “Gestionar catálogo de Moneda de Cambio”

CU_10	“Gestionar catálogo de Moneda de Cambio”
Actor	Dpto. de Control, Especialista.
Descripción	Permite añadir cuantos tipos de monedas se necesiten utilizar con sus respectivas tasas de cambio.
Referencia	R12.

Tabla 2.18: Breve descripción del caso de uso del sistema “Gestionar ingresos”

CU_11	“Gestionar ingresos”
Actor	Dpto. de Control, Especialista.
Descripción	Permite introducir las partidas de ingresos que se van a utilizar.
Referencia	R13.

Tabla 2.19: Breve descripción del caso de uso del sistema “Buscar por DIP”

CU_12	“Buscar por DIP”
Actor	Dpto. de Control, Especialista.
Descripción	Permite al Especialista realizar una búsqueda mediante la DIP a la que pertenece la Evaluación Económica para buscar la misma y realizarle alguna actualización.
Referencia	R14.

Tabla 2.20: Breve descripción del caso de uso del sistema “Buscar Evaluación de inversiones”

CU_13	“Buscar Evaluación de inversiones”
Actor	Dpto. de Control , Especialista
Descripción	Permite al especialista buscar alguna Evaluación a la cual se le hará alguna modificación.
Referencia	R7. Extendido de “Gestionar Evaluación de Inversiones”

Tabla 2.21: Breve descripción del caso de uso del sistema “Gestionar gastos”

CU_14	“Gestionar gastos”
Actor	Dpto. de Control, Especialista.
Descripción	Permite introducir las partidas de gastos que se van a utilizar.
Referencia	R15.

Tabla 2.22: Breve descripción del caso de uso del sistema “Gestionar inversiones”

CU_15	“Gestionar inversiones”
Actor	Dpto. de Control, Especialista.
Descripción	Permite introducir el valor de las inversiones.
Referencia	R16.

Tabla 2.23: Breve descripción del caso de uso del sistema “Gestionar depreciación”

CU_16	“Gestionar depreciación”
Actor	Dpto. de Control, Especialista.
Descripción	Permite introducir calcular la depreciación de las inversiones.
Referencia	R17.

Tabla 2.24: Breve descripción del caso de uso del sistema “Gestionar capital de trabajo”

CU_12	“Gestionar capital de trabajo”
Actor	Dpto. de Control, Especialista.
Descripción	Permite calcular la variación del capital de trabajo.
Referencia	R18.

2.7.3 Diagrama de Casos de Uso del Sistema

El Diagrama de Casos de Uso del Sistema es un artefacto de Ingeniería de Software que describe, bajo la forma de acciones y reacciones, el comportamiento del sistema desde el punto de vista del usuario, permitiendo de esta forma el establecimiento de un acuerdo entre clientes y desarrolladores sobre las condiciones y requerimientos que debe cumplir el sistema. Este modelo está formado por actores, casos de uso y las relaciones que se establecen entre estos, es decir, representa gráficamente a los procesos y su interacción con los actores y constituye una entrada de gran valor para las siguientes fases de construcción de un software.

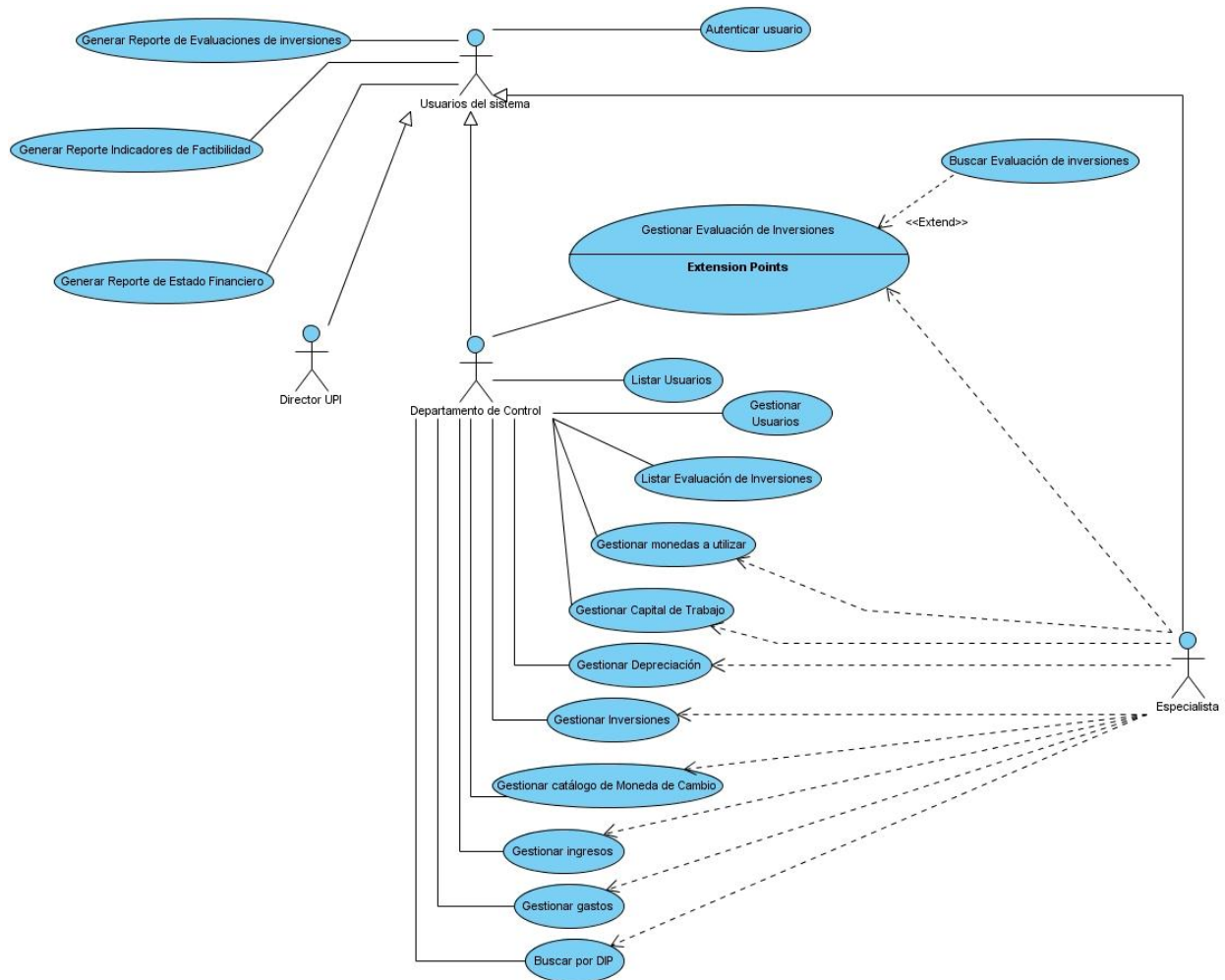


Figura 2.1: Diagrama de Casos de Uso del Sistema.

2.7.4 Patrones de casos de uso

Tras el proceso de análisis de los requisitos y la definición de los casos de uso que se tendrán en el sistema, que responden y abarcan esos requisitos, se representan los actores y casos de uso mediante el Diagrama de Casos de Uso del Sistema (Figura 2.1). En este diagrama se utilizaron varios patrones de casos de uso, estos son:

CRUD parcial: Ya que se representa en los casos de uso Gestionar Usuarios, Gestionar Ingresos, Gestionar monedas a utilizar y Gestionar catálogo de Moneda de Cambio, procesos como crear y modificar.

Múltiples actores, específicamente el patrón Rol común: Pues se destacan actividades comunes para varios actores, por lo que se define un actor del cual heredan los actores con estas funcionalidades comunes.

2.7.5 Casos de Usos a desarrollar por ciclo

Al culminar un ciclo de un proyecto de software, se obtiene una versión del producto. Por eso se debe determinar con anterioridad los casos de usos que se van a desarrollar en cada ciclo de vida del proyecto para poder tener una guía por la cual trabajar, así como trazar las metas.

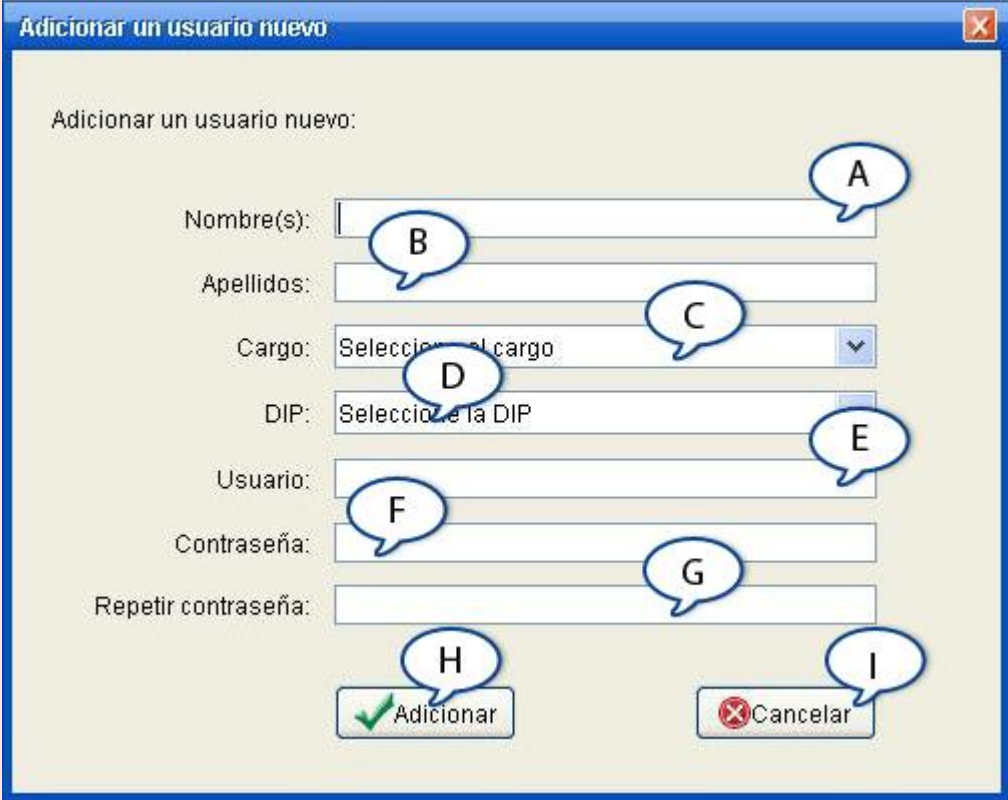
Tabla 2.21: Distribución de Casos de Uso por ciclos de desarrollo.

Ciclo	Código	Nombre del caso de uso	Justificación de la selección
1	CU_1	Gestionar Usuarios.	Es importante que el sistema lleve un estricto control de los usuarios, ya que de estos dependerán los restantes datos. Además de las Evaluaciones de Inversión que a partir de aquí es donde se comienzan a insertar todos los datos. De no haber Evaluaciones no se puede registrar ningún otro dato.
	CU_2	Autenticar Usuario.	
	CU_3	Listar Usuarios.	
	CU_4	Gestionar Evaluación de Inversiones.	
	CU_9	Gestionar monedas a utilizar	
	CU_10	Gestionar catálogo de Moneda de Cambio	
	CU_11	Gestionar ingresos	
	CU_14	Gestionar gastos	
	CU_15	Gestionar inversiones	
	CU_16	Gestionar depreciación	
2	CU_5	Listar Evaluación de Inversiones.	Este sería el 2do y último ciclo de desarrollo de esta aplicación. Luego de tener los usuarios registrados en el sistema y las Evaluaciones, ya podremos comenzar a manejar los datos necesarios dentro del sistema para obtener los reportes necesarios para la UPI, permitiendo con esto que se tenga la aplicación completa y que cubra todas las necesidades existentes.
	CU_13	Buscar Evaluación de inversiones	
	CU_8	Generar Reporte de Evaluaciones de inversiones	
	CU_12	Buscar por DIP	
	CU_6	Generar Reporte de Estado Financiero	
	CU_7	Generar Reporte Indicadores de Factibilidad	

2.8 Casos de Uso expandidos.

Los principales casos de usos arquitectónicamente significativos son Gestionar usuarios y Gestionar Evaluación

Caso de Uso “Gestionar Usuarios”	
CU_1	“Gestionar Usuarios”
Propósito	Permitir al Dpto. de Control gestionar los usuarios en el sistema.
Actor	Dpto. de Control (inicia).
Resumen	El caso de uso inicia cuando el Dpto. de Control decide registrar un nuevo usuario o modificar algún dato de uno ya existente. Realiza la operación deseada y finaliza así el caso de uso.
Referencia	R1, R1.1, R1.2, R3.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El Dpto. de Control necesita registrar o modificar los datos de un usuario.	1.1 El sistema ejecuta alguna de las siguientes acciones: a) Si decide registrar un nuevo usuario, ir a la sección “Agregar un usuario nuevo”. b) Si decide modificar los datos de un usuario, ir a la sección “Modificar datos del usuario”.
Sección “Agregar un usuario nuevo”	
Acción del Actor	Respuesta del Sistema
2. El Dpto. de Control entra los datos necesarios para registrar un usuario al sistema (Nombres (A), Apellidos (B), Cargo (C), DIP a la que pertenece (D), Usuario (E), Contraseña (F), Repetir contraseña (G)), Aceptar (H), Cancelar (I).	2.1 El sistema verifica que todos los campos estén llenos. 2.2 El sistema verifica que este nombre de usuario no exista. 2.3 El sistema verifica que el campo Contraseña (F) y Repetir contraseña (G) sean iguales. 2.4 El usuario se almacena en el sistema, finalizando así el caso de uso.
Prototipo de Interfaz (Interfaz I)	



A. Control para introducir el/los nombre/s del usuario nuevo.

Nombre: Nombre(s) Tipo: JTextField

B. Control para introducir los apellidos del nuevo usuario.

Nombre: Apellidos Tipo: JTextField

C. Control para seleccionar el cargo que tiene este nuevo usuario. Los cargos pueden ser: Administrador, Directivo y Especialista. Este cargo es quien les da a los usuarios el rol que tendrán en el sistema y el nivel de acceso que tendrá cada uno de ellos.

Nombre: Cargo Tipo: JComboBox

D. Control para seleccionar la DIP a la que pertenece el usuario. Estas pueden ser: Preparación de Obras, Docencia, Infra-Estructura Productiva, Logística, Residencia y Facultades Regionales.

Nombre: DIP Tipo: JComboBox

E. Control para introducir el nombre de usuario que va a tener el nuevo usuario del sistema.

Nombre: Usuario Tipo: JTextField

F. Control para registrar la contraseña que tendrá este usuario.

Nombre: Contraseña Tipo: JPasswordField

G. Control repetir la contraseña, lo que hace es que verifica si concuerda con la puesta en el campo anterior.

Nombre: RepetirContraseña Tipo: JPasswordField

H. Control de aceptar una acción (Añadir el nuevo usuario).

Nombre: Aceptar Tipo: JButton

I. Control de cancelar una acción.

Nombre: Cancelar Tipo: JButton	
Flujo Alternativo de Eventos	
Acción del Actor	Respuesta del Sistema
	<p>2.1 Se emite un mensaje para que llene todos los campos “Todos los campos son obligatorios, por favor rectifique”.</p> <p>2.2 Si las contraseñas no coinciden se emite un mensaje “Las contraseñas deben coincidir.”</p> <p>2.3 Si el usuario existe se emite un mensaje informando la existencia del mismo y se finaliza así el caso de uso “Ya se encuentra registrado otra persona con ese identificador, seleccione otro”.</p>
Sección “Modificar datos del usuario”	
Acción del Actor	Respuesta del Sistema
2. El Dpto. de Control selecciona el usuario a modificar.	2.1 El sistema brinda la posibilidad de modificar los datos, menos el nombre de usuario.
3. El Dpto. de Control realiza las actualizaciones deseadas en los campos (Nombres (A), Apellidos (B), Cargo (C), DIP a la que pertenece (D), Contraseña (F), Repetir contraseña (G)), Estado (H) y presiona el botón Aceptar (I) si desea guardar los cambios realizados o si desea Cancelar (J)	<p>3.1 El sistema verifica que todos los campos estén llenos.</p> <p>3.2 El sistema verifica que las contraseñas de los campos Contraseña (F) y Repetir contraseña (G) coinciden.</p> <p>3.3 Se actualiza la información incorporada al usuario, finalizando así el caso de uso.</p>
Prototipo de Interfaz (Interfaz II)	

A. Control donde sale el/los nombre/s actuales del usuario que puede ser modificado.

Nombre: Nombre(s) Tipo: JTextField

B. Control aparecen los apellidos del usuario que puede ser modificado.

Nombre: Apellidos Tipo: JTextField

C. Control que define el cargo que tiene este usuario. Los cargos pueden ser: Administrador, Directivo, Especialista y Jurídico. Este cargo es quien les da a los usuarios el rol que tendrán en el sistema y el nivel de acceso que tendrá cada uno de ellos.

Nombre: Cargo Tipo: JComboBox

D. Control donde aparece la DIP a la que pertenece el usuario. Estas pueden ser: Preparación de Obras, Docencia, Infra-Estructura Productiva, Logística, Residencia y Facultades Regionales.

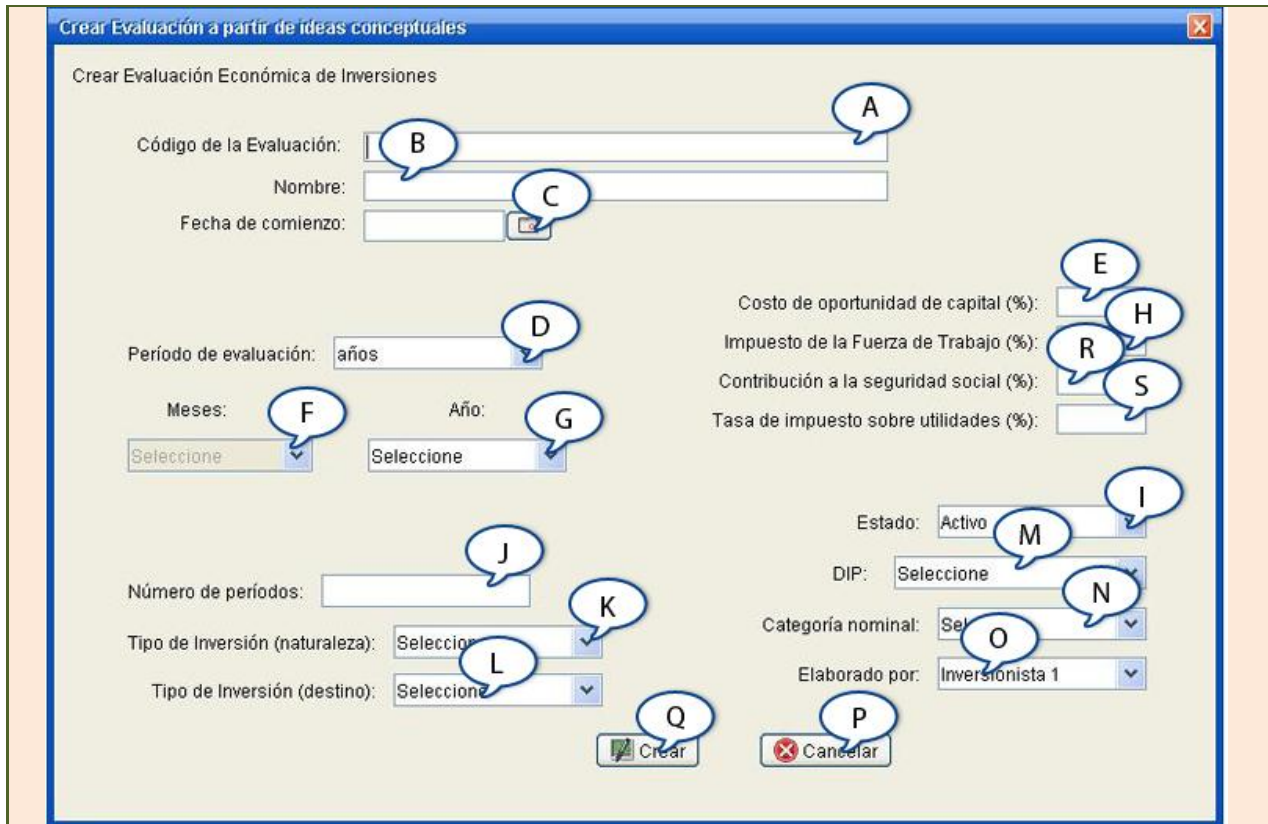
Nombre: DIP Tipo: JComboBox

E. Control donde aparece el nombre de usuario de usuario a modificar, este control aparece desactivado, o sea, es lo único que no podrá ser cambiado.

<p>Nombre: Usuario Tipo: JTextField F. Control para mostrar la contraseña del usuario. Nombre: Contraseña Tipo: JPasswordField G. Control para mostrar la contraseña, lo que hace es que verifica si concuerda con la puesta en el campo anterior. Nombre: RepetirContraseña Tipo: JPasswordField H. Control para Activar/Desactivar Usuario. Una vez el usuario este desactivado, no podrá acceder al sistema hasta que su estado pase a ser nuevamente activado, terminando así el caso de uso. Nombre: Estado Tipo: JComboBox I. Control de aceptar una acción (Modificar usuario). Nombre: Aceptar Tipo: JButton J. Control de cancelar una acción. Nombre: Cancelar Tipo: JButton</p>	
Flujo Alternativo de Eventos	
Acción del Actor	Respuesta del Sistema
	<p>3.1 Se emite un mensaje informando que todos los campos deben ser llenados “Todos los campos son obligatorios, por favor rectifique”.</p> <p>3.2 Se emite un mensaje de que los campos Contraseña (F) y Repetir contraseña (G) deben ser iguales “Las contraseñas deben coincidir”.</p>
Prioridad	Crítico

Caso de Uso “Gestionar Evaluación de Inversiones”	
CU_4	“Gestionar Evaluación de Inversiones”
Propósito	Permitir al Dpto. de Control gestionar los usuarios en el sistema.
Actor	Especialista (inicia).
Resumen	El caso de uso inicia cuando el especialista decide gestionar la información de las Evaluaciones de Inversión, tales como, crear una evaluación y actualizar los datos introducidos por Dpto. de Control o por él mismo.
Referencia	R5, R5.1, R5.2, R7
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El especialista necesita registrar o modificar los datos de una Evaluación, o crear un plan de obras.	<p>1.1 El sistema ejecuta alguna de las siguientes acciones:</p> <p>a) Si decide registrar una evaluación a partir de ideas conceptuales, ir a la sección</p>

	<p>“Crear Evaluación de una inversión a partir de ideas conceptuales”.</p> <p>b) Si decide registrar una evaluación a partir del presupuesto ejecutivo, ir a la sección “Crear Evaluación de una inversión a partir del presupuesto ejecutivo”.</p> <p>c) Si decide modificar los datos de una Evaluación de una inversión, ir a la sección “Modificar Evaluación de una inversión”.</p> <p>d) Si decide crear un plan de obras, ir a la sección “Crear plan de obras”.</p>
Sección “Crear Evaluación de una inversión a partir de ideas conceptuales”	
Acción del Actor	Respuesta del Sistema
<p>2. El Especialista entra los datos necesarios para registrar una Evaluación de Inversión al sistema: código de la Evaluación de Inversiones (A), nombre (B), fecha de comienzo (C), Período de evaluación (D), Costo de oportunidad del capital (E), Períodos en meses (F), Períodos en años (G), Impuesto de la fuerza de trabajo (H), Estado de la Evaluación de Inversiones -Activo, Terminado, Anulado- (I), Número de períodos (J), Tipo de inversión según su naturaleza (Nueva, Ampliación, Remodelación) (K), Tipo de inversión según su destino de explotación (Tecnológica, Turística, Educación, Salud, Vivienda, Industrial, Logística, Cultural, Otros) (L), DIP a la que pertenece (M), Categoría nominal (Nominal, No Nominal) (N), Elaborado por (O), Contribución a la seguridad social (R), Tasa de impuesto sobre utilidades (S).</p>	<p>2.1 El sistema verifica que todos los campos estén llenos.</p> <p>2.2 El sistema verifica que este código de de la evaluación no exista.</p> <p>2.3 La evaluación se almacena en el sistema, finalizando así el caso de uso.</p>
Prototipo de Interfaz (Interfaz V)	



Crear Evaluación Económica de Inversiones

Código de la Evaluación: A

Nombre: B

Fecha de comienzo: C

Período de evaluación: años D

Meses: F Año: G

Seleccione Seleccione

Número de períodos: J

Tipo de Inversión (naturaleza): K

Tipo de Inversión (destino): L

Costo de oportunidad de capital (%): E

Impuesto de la Fuerza de Trabajo (%): H

Contribución a la seguridad social (%): R

Tasa de impuesto sobre utilidades (%): S

Estado: Activo I

DIP: M

Categoría nominal: N

Elaborado por: O

P

A. Control para introducir el código de la Evaluación de la Inversión.

Nombre: Código Tipo: JCalendar

B. Control para introducir el nombre de la Evaluación de la Inversión.

Nombre: Nombre Tipo: JTextField

C. Control para introducir la fecha de comienzo de la Evaluación de la Inversión.

Nombre: Fecha de comienzo Tipo: JTextField

D. Control para escoger el período de la Evaluación. Estos períodos pueden ser: Por años o por meses.

Nombre: Período Tipo: JComboBox

E. Control para introducir el costo de oportunidad de capital de la Evaluación de la inversión.

Nombre: Costo de oportunidad de capital (%) Tipo: JTextField

F. Control para escoger el mes por el que comenzará a realizarse la Evaluación de la inversión.

Nombre: Meses Tipo: JComboBox

G. Control para introducir el año en que comienza a realizarse la Evaluación de la inversión.

Nombre: Año Tipo: JTextField

H. Control para introducir el impuesto de la fuerza de trabajo prevista para la Evaluación de la inversión.

Nombre: Impuesto de la fuerza de Trabajo Tipo: JTextField

I. Control para escoger el estado de la Evaluación. Estos estados pueden ser: Activo, Terminado, Anulado..

Nombre: Estado Tipo: JComboBox

J. Control para introducir el número de períodos que se van a utilizar para realizar la Evaluación de la inversión.

Nombre: Número de períodos Tipo: JTextField

K. Control para escoger el tipo de inversión según su naturaleza (Nueva, Ampliación, Remodelación).

Nombre: Tipo de Inversión (naturaleza) Tipo: JComboBox

L. Control para escoger el tipo de inversión según su destino de explotación (Tecnológica, Turística, Educación, Salud, Vivienda, Industrial, Logística, Cultural, Otros).

Nombre: Tipo de Inversión (destino) Tipo: JComboBox

M. Control para escoger la DIP a la que está asociada la Evaluación. Estas pueden ser: Preparación de Obras, Docencia, Infra-Estructura Productiva, Logística, Residencia y Facultades Regionales.

Nombre: DIP Tipo: JComboBox

N. Control para escoger la Categoría nominal. Estas pueden ser: Nominal, No Nominal.

Nombre: Categoría nominal Tipo: JComboBox

O. Control para seleccionar el nombre del usuario que realizara la Evaluación de la Inversión

Nombre: Elaborado por Tipo: JComboBox

P. Control que permite salir, Cancelar y volver a la pantalla principal.

Nombre: Cancelar Tipo: Jbutton

Q. Control que permite crear una nueva Evaluación.

Nombre: Crear Tipo: Jbutton

R. Control que permite introducir el porcentaje de contribución a la seguridad social de una Evaluación Económica.

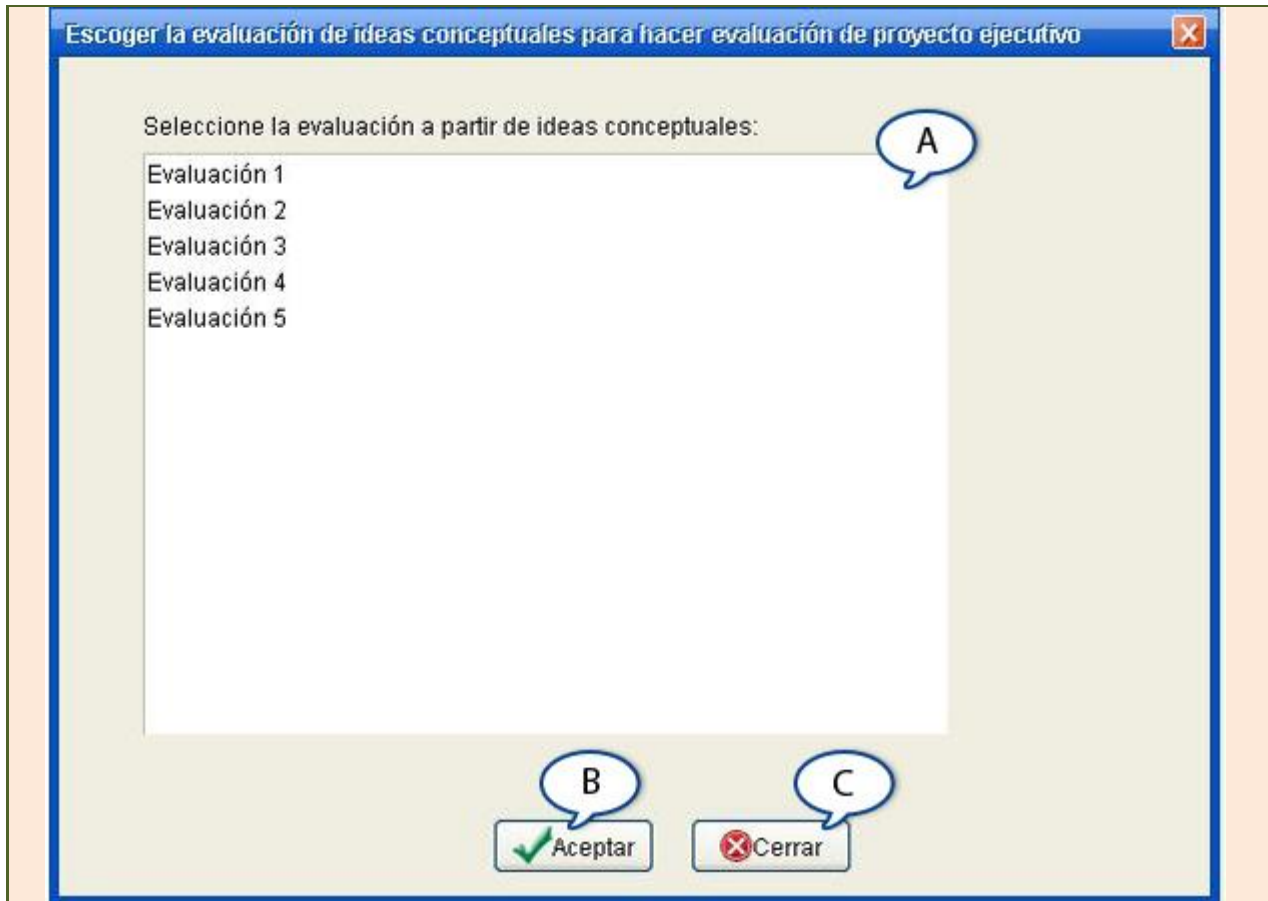
Nombre: Contribución a la Seguridad Social Tipo: JTextField

S. Control para introducir la tasa de impuesto prevista para la Evaluación de la inversión.

Nombre: Tasa de Impuesto sobre Utilidades Tipo: JTextField

Flujo Alternativo de Eventos

Acción del Actor	Respuesta del Sistema
	2.1 Se emite un mensaje para que llene todos los campos "Todos los campos son obligatorios". 2.2 Si el código de la Evaluación existe se emite un mensaje informando la existencia del mismo y se finaliza así el caso de uso "Existe otra Evaluación con ese código. Por favor, rectifique".
Sección "Crear Evaluación de una inversión a partir del presupuesto ejecutivo"	
Acción del Actor	Respuesta del Sistema
2. El usuario escoge la evaluación a partir de las ideas conceptuales que desea pasar a proyecto ejecutivo.	2.1 El sistema muestra una interfaz similar a la de presupuesto de ideas conceptuales para que llenen los datos.
Prototipo de Interfaz (Interfaz VI)	



A. Control que permite seleccionar una Evaluación creada a partir de ideas conceptuales.

Nombre: Crear Tipo: JList.

B. Control que permite crear una Evaluación de proyecto ejecutivo a partir de la evaluación seleccionada.

Nombre: Aceptar Tipo: JButton

C. Control que permite salir y volver a la pantalla principal.

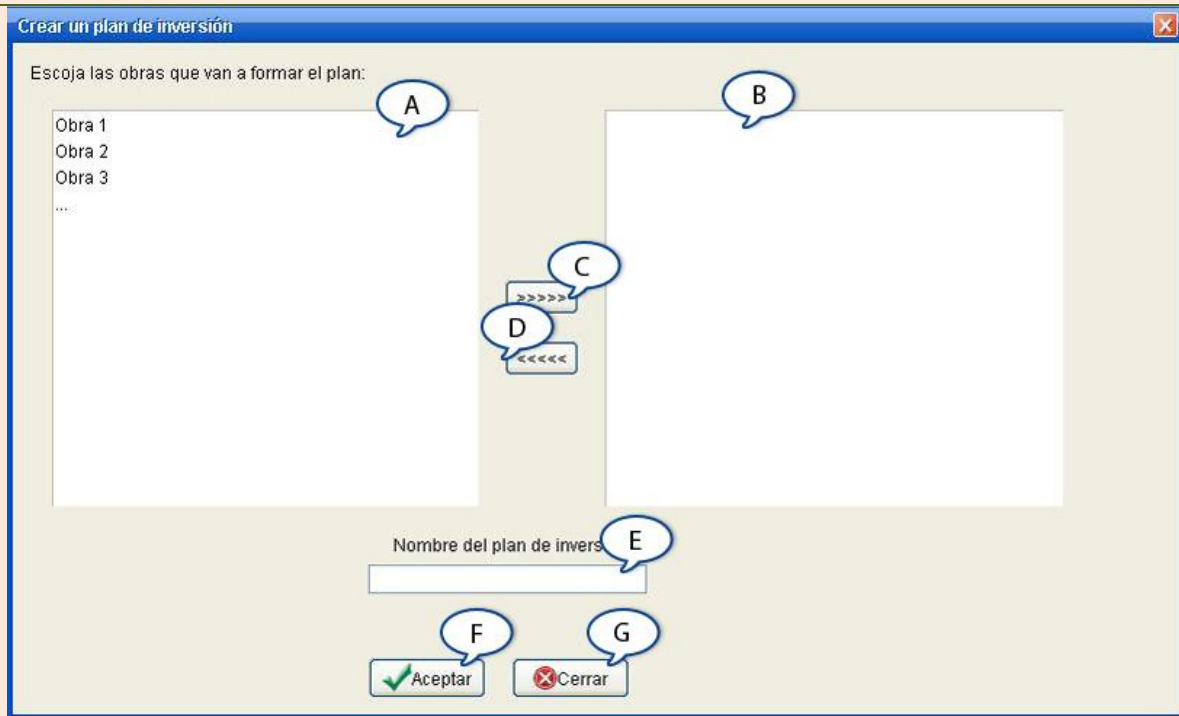
Nombre: Cerrar Tipo: JButton

Flujo Alternativo de Eventos

Acción del Actor	Respuesta del Sistema
	<p>2.1 Si no selecciona ninguna evaluación, el sistema debe mostrar el mensaje: "Debe seleccionar una evaluación para pasar esta a presupuesto ejecutivo."</p> <p>2.2 Si la Evaluación ya ha sido seleccionada, se debe mostrar el mensaje:</p>

	“Ya esa evaluación pasó a presupuesto ejecutivo.”
Sección “Crear Evaluación de un plan de obras”	
Acción del Actor	Respuesta del Sistema
2. El actor escoge las obras que van a formar parte del plan de obras y le pone un nombre a dicho plan de inversión.	2.1 El sistema muestra una interfaz similar a la de crear una inversión a partir de ideas conceptuales.

Prototipo de Interfaz (Interfaz VII)



- A. Control que permite seleccionar las obras que formaran parte de un plan de obras
Nombre: Obras Tipo: JList
- B. Control que muestra las obras que forman un plan de obras.
Nombre: Plan de Obras Tipo: JList
- C. Control que permite adicionar las obras que formaran parte de un plan de obras.
Nombre: >>>>> Tipo: JButton
- D. Control que permite quitar una obra la cual no va a formar parte del plan de obras.
Nombre: <<<<< Tipo: JButton
- E. Control que permite introducir el nombre del plan de obras.

Nombre: Nombre del plan de inversiones Tipo: JTextField	
F. Control que permite crear el plan de obras y volver a la pantalla principal	
Nombre: Aceptar Tipo: JButton	
G. Control que permite salir y volver a la pantalla principal.	
Nombre: Cerrar Tipo: JButton	
Flujo Alternativo de Eventos	
Acción del Actor	Respuesta del Sistema
	<p>2.1 Si no selecciona ninguna evaluación, el sistema debe mostrar el mensaje: "Debe seleccionar una evaluación para pasar esta a presupuesto ejecutivo."</p> <p>2.2 Si deja en blanco el cuadro de texto para el nombre, se emite un mensaje para que lo llene: "No le ha puesto nombre al plan de obras."</p>
Prioridad	Crítico.

2.9 Conclusiones

Con la culminación de este capítulo se han desarrollado y obtenido hasta el momento los artefactos y documentación resultante de los flujos de trabajo modelado del negocio y requerimientos. Se describieron cada uno de los casos de uso y se le da prioridad a aquellos que serán implementados en el primer ciclo de desarrollo, se especificaron los requisitos funcionales y no funcionales obteniendo una total descripción del sistema.

Capítulo 3. Análisis y Diseño.

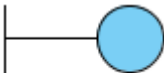


En este capítulo se expone el análisis y diseño del sistema, que aportará una nueva visión del sistema propuesto sobre los requisitos funcionales identificados. Primeramente se realiza el análisis, que representa una aproximación al diseño, con los diagramas de clases correspondientes. A continuación se muestra el diseño del sistema, para ello se realizan los diagramas de clases, así como los diagramas de secuencias. Además, en este capítulo se presenta el diagrama de clases persistentes y el diagrama entidad relación, que constituyen la base para la construcción de la base de datos.

3.1 Modelo de Análisis

El Modelo de análisis se realiza para obtener una visión del sistema sobre los requisitos funcionales expresados ya en un lenguaje técnico, constituye una primera aproximación al modelo de diseño. El mismo ofrece ventajas tales como: suavizar la transición al diseño, apoyar el cambio a otra plataforma de programación, servir para obtener una visión general de la propuesta del sistema así como para planificar y dividir el diseño e implementación en pequeños módulos, apoya la aplicación de reingeniería a aplicaciones existentes (al ser en un lenguaje menos técnico ayuda a entender mejor la propuesta de solución). Está orientado a analizar cómo el sistema va a cumplir sus funcionalidades. El análisis debe capturar los requisitos de usuario sin adoptar prematuramente decisiones de implementación, es decir, omitiendo detalles dependientes de la tecnología, y utilizando conceptos extraídos únicamente del dominio del problema.

3.1.1 Modelo de clases de análisis

Los diagramas de clases del análisis muestran las clases participantes en el caso de uso y las relaciones que existen entre estas

Nombre	Características	Representación
Interfaz	Modelan la interacción entre el sistema y sus actores.	 clase_interfaz
Entidad	Modelan información que posee larga vida y que es a menudo persistente.	 clase_entidad
Control	Coordinan la realización de uno o unos pocos casos de uso coordinando las actividades de los objetos que implementan la funcionalidad del caso de uso.	 clase_control

A continuación se muestran los diagramas de clases del análisis de los 2 casos de uso arquitectónicamente significativos principales:

Diagrama de clase del análisis “Gestionar usuarios”.

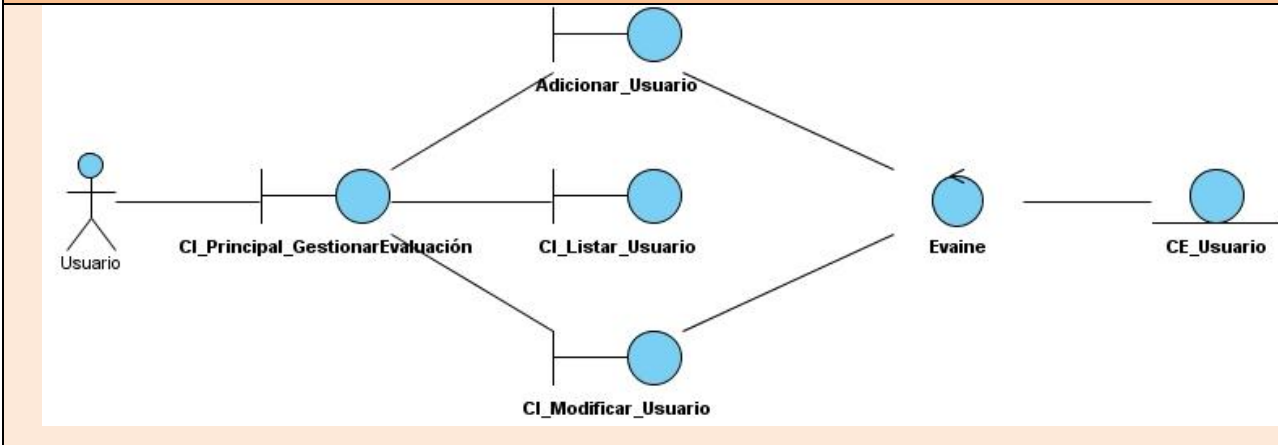
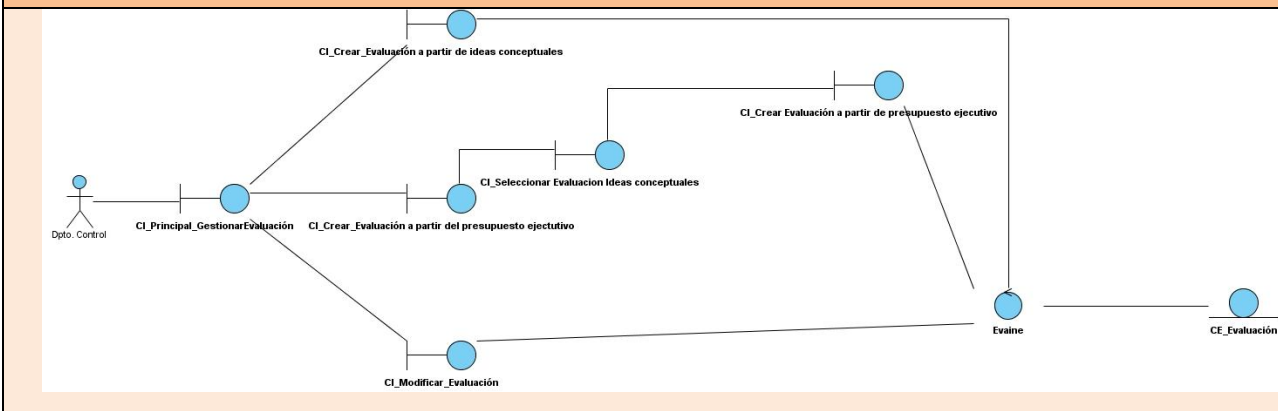


Diagrama de clase del análisis “Gestionar Evaluación”.



3.1.2 Arquitectura del sistema

Con el objetivo de comprender mejor el sistema y organizar el desarrollo se hace necesario describir la arquitectura definida por para el sistema, esta se mantendrá durante el diseño y la implementación. La arquitectura que se empleará para el desarrollo de esta aplicación responde a la Arquitectura en 3 capas, la cual proporciona una buena organización y estructuración entre los distintos niveles de abstracción, donde, un cambio en uno de estos niveles no debe proporcionar cambios en los restantes. El diseño de tres capas se define de la siguiente manera:

Capa de presentación: es la capa mediante la cual el sistema le brinda la posibilidad al usuario de interactuar con él. Se encarga de mostrar la información y captura la misma con el objetivo de procesarla. Además, en ella se realiza una primera validación de la información con el objetivo de evitar posibles errores. Esta capa solo se comunica con la capa lógica del negocio.

Capa lógica de negocio: es una de las capas más importantes. Se encarga de recibir las peticiones de los usuarios y enviar las respuestas. Aquí es donde se establecen todas las reglas que deben cumplirse y se manipula toda la información. Esta capa se comunica con la capa de presentación para recibir las solicitudes y mostrar los resultados, y con la capa de acceso a datos, para solicitar al gestor de base de datos almacenar o recuperar datos de él. Para el acceso a datos esta capa se relaciona con una clase que implementa la interfaz del modelo de persistencia que responde a un patrón llamado Factory, la cual es una puerta de enlace entre la capa de Acceso a Datos y la capa de Lógica de Negocio.

Capa de acceso a datos: Es la capa encargada de manejar todo el flujo de información que entra y sale de la fuente de datos, así como la conexión a la misma. Como se explicaba en la capa anterior, aquí se implementa una clase interfaz que se encarga de establecer un enlace con la capa lógica del negocio, ya que es la capa encargada de establecer la conexión con la base de datos. Ella implementa el patrón de diseño Singleton cuyo principal objetivo, y para el cual está diseñado, es restringir la creación de objetos pertenecientes a una clase, logrando que una clase sólo tenga una instancia.

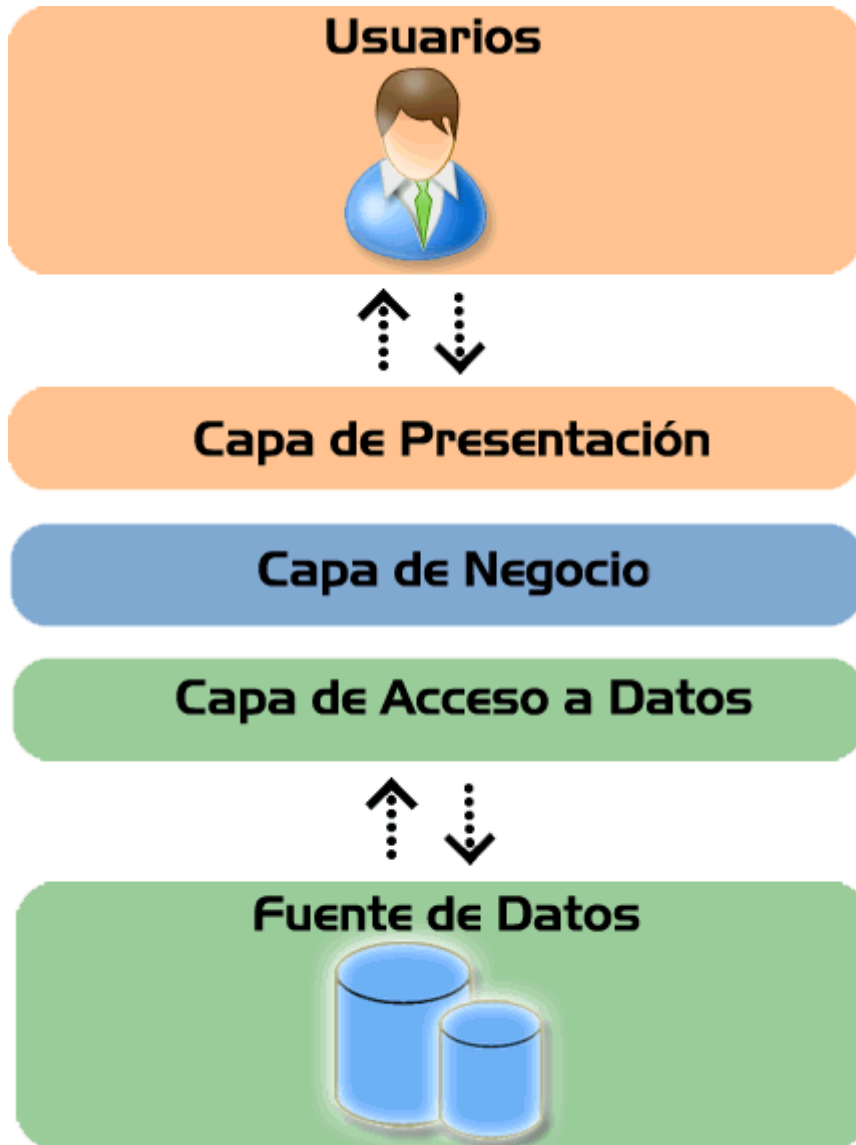


Figura 3.1: Arquitectura de tres capas.

3.1.3 Patrones de Diseño

“Una arquitectura orientada a objetos bien estructurada está llena de patrones. La calidad de un sistema orientado a objetos se mide por la atención que los diseñadores han prestado a las colaboraciones entre sus objetos. Los patrones conducen a arquitecturas más pequeñas, más simples y más comprensibles”. (Booch, 1996)

La capa de acceso a datos de Evaine utiliza el patrón DAO (Data Access Object). El problema que viene a resolver este patrón es el de contar con diversas fuentes de datos de tal forma que se encapsula la forma de acceder a la fuente de datos. Este patrón surge históricamente de la necesidad de gestionar una diversidad de fuentes de datos, aunque su uso se extiende al problema de encapsular no sólo la fuente de datos, sino además ocultar la forma

de acceder a los datos. Se trata de que el software cliente se centre en los datos que necesita y se olvide de cómo se realiza el acceso a los datos o de cuál es la fuente de almacenamiento; así un cambio en el origen de los datos o en la manera de recuperarlos no afecta la capa superior siempre que se implemente la interfaz correspondiente.

Algunas características:

- No es imprescindible, pero en proyectos de cierta complejidad resulta útil que el DAO implemente una interfaz. De esta forma, los objetos cliente tienen una forma unificada de acceder a los DAO.
- El DAO accede a la fuente de datos y la encapsula para los objetos clientes, que oculta tanto la fuente como el modo (JDBC²⁰) de acceder a ella.

3.2 Modelo de Diseño

El modelo de diseño es un refinamiento del análisis. Un modelo de objetos que describe la realización física de los casos de uso. Se centra en los impactos que producen en el sistema a desarrollar los requerimientos funcionales y no funcionales. Se determina la arquitectura general del sistema y su comportamiento dinámico, adaptando la especificación realizada en la etapa anterior. En esta fase se establece el comportamiento dinámico del sistema, es decir, como debe reaccionar ante los acontecimientos.

El resultado obtenido de la etapa de Diseño facilita enormemente la implementación posterior del sistema, pues proporciona la estructura básica del sistema y como los diferentes componentes actúan y se relacionan entre ellos.

3.2.1 Diagrama de clases del diseño

A continuación se muestra el diagrama del diseño donde se describe la estructura del sistema mostrando sus clases, atributos y las relaciones entre ellos. Este diagrama es muy utilizado durante el diseño de los sistemas, y a partir de él se crea el diseño conceptual de la información que se manejará en el sistema, y los componentes que se encargarán del funcionamiento y la relación entre uno y otro.

A continuación se muestran los diagramas de clases del diseño de los 2 casos de uso arquitectónicamente significativos principales:

Diagrama de clase del diseño “Gestionar evaluación”.

²⁰ Es el acrónimo de Java Database Connectivity, un API que permite la ejecución de operaciones sobre bases de datos desde el lenguaje de programación Java.

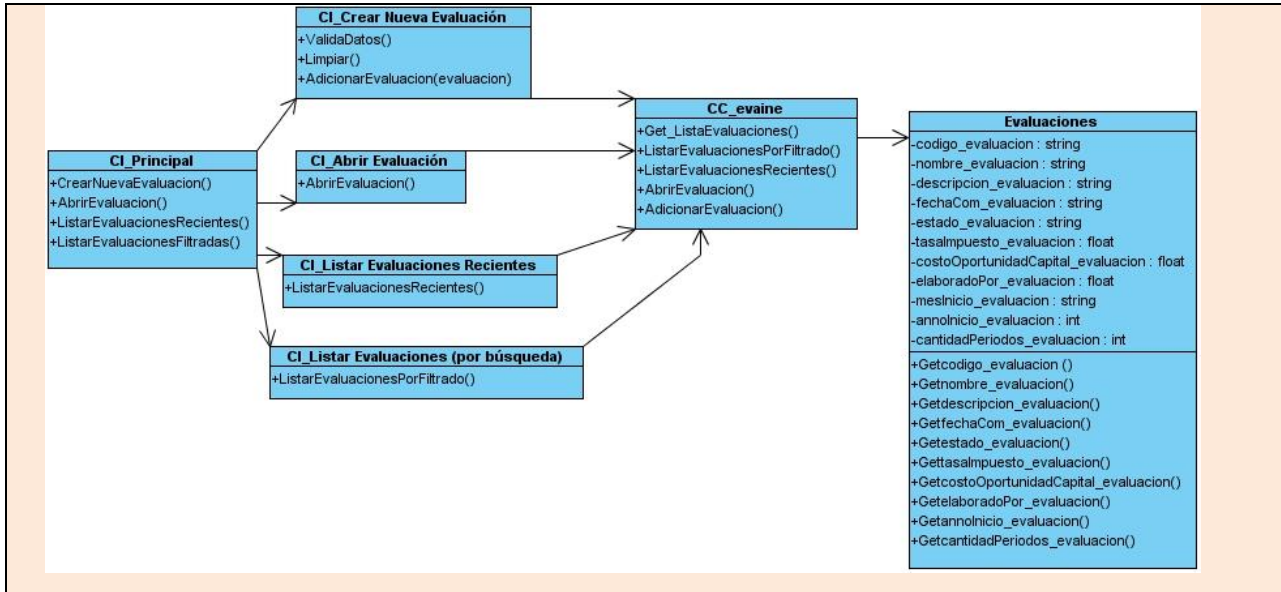
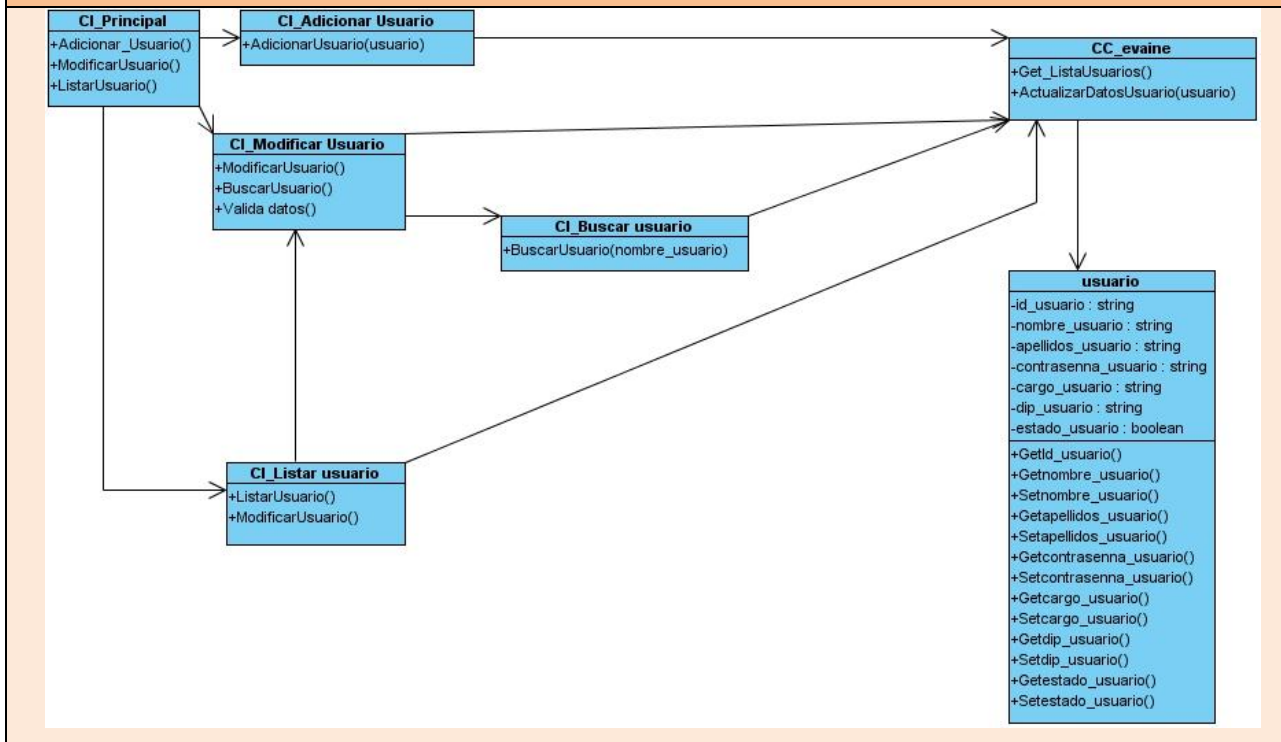


Diagrama de clase del diseño "Gestionar usuario".



3.2.2 Diagramas de Interacción

Los diagramas de secuencia y los diagramas de colaboración (ambos llamados diagramas de interacción) son dos de los cinco tipos de diagramas de UML que se utilizan para modelar los aspectos dinámicos de los sistemas. Un diagrama de interacción muestra una interacción, que consiste en un conjunto de objetos y sus relaciones, incluyendo los mensajes que se pueden enviar entre ellos.

Un diagrama de secuencia es un diagrama de interacción que destaca la ordenación temporal de los mensajes; un diagrama de colaboración es un diagrama de interacción que destaca la organización estructural de los objetos que envían y reciben mensajes.

Los diagramas de interacción se utilizan para modelar los aspectos dinámicos de un sistema. La mayoría de las veces, esto implica modelar instancias concretas o prototípicas de clases, interfaces, componentes y nodos, junto con los mensajes enviados entre ellos, todo en el contexto de un escenario que ilustra un comportamiento. Los diagramas de interacción pueden utilizarse para visualizar, especificar, construir y documentar la dinámica de una sociedad particular de objetos, o se pueden utilizar para modelar un flujo de control particular de un caso de uso.

3.2.2.1 Diagrama de Secuencia

El diagrama de secuencia contiene detalles de implementación del escenario, incluyendo los objetos y clases de secuencia que se usan para implementar este, y mensajes pasados entre los objetos. Para realizar los diagramas que a continuación se muestran se examinó la descripción de los casos de uso para determinar qué objetos eran necesarios para la implementación, convirtiendo los pasos de este en un flujo de acciones sobre el cual caminar.

Los diagramas de secuencia son de gran importancia en el diseño de un sistema debido a que permiten observar las interacciones que ocurren entre los distintos objetos que participan en un escenario determinado.

Un escenario de un caso de uso es un "camino" completo a través del caso de uso. Los usuarios finales del sistema pueden seguir muchos caminos cuando ejecutan la funcionalidad especificada en el caso de uso. Siguiendo el flujo básico serían un escenario.

A continuación se muestran los diagramas de secuencia de los 2 casos de uso arquitectónicamente significativos principales:

Diagrama de secuencia – Caso de Uso “Gestionar Evaluación” – Escenario “Crear evaluación”.

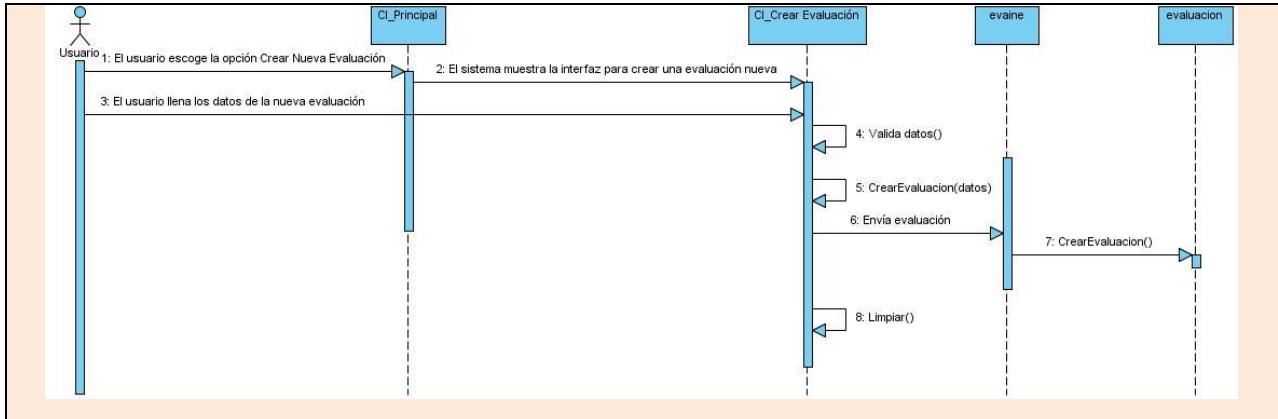
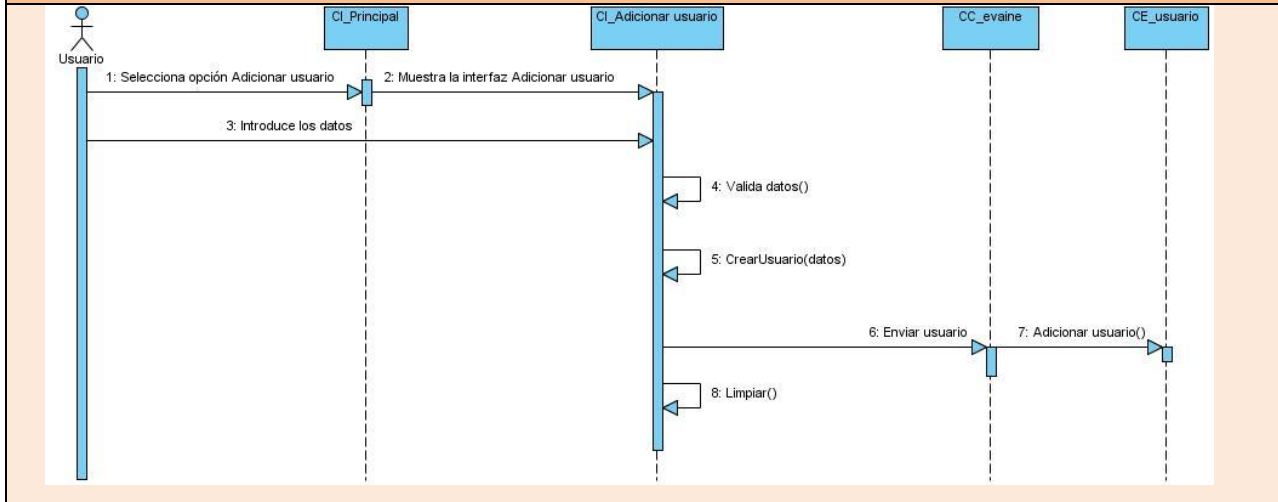


Diagrama de secuencia – Caso de Uso “Gestionar Usuario” – Escenario “Adicionar usuario”.



3.2.3 Diseño de la Base de Datos

Uno de los pasos cruciales en la construcción de una aplicación que maneje una base de datos es, sin dudas, el diseño de la base de datos. En la base de datos que se utiliza en la aplicación que se está diseñando como propuesta de solución para lograr el acceso eficiente a la información con redundancia mínima se toman varias consideraciones, entre las que se encuentran: la velocidad y facilidad de acceso a la información para extraerla.

En el capítulo 2 se describen sólo los casos de usos que serán implementados en el primer ciclo de desarrollo de la aplicación, pero teniendo en cuenta la importancia de la base de datos y para evitar una modificación posterior se decide realizar el modelado de la base de datos en su totalidad, incluyendo todas las entidades persistentes que están relacionadas.

3.2.3.1 Clases persistentes

Todas las clases identificadas en el dominio del análisis no son persistentes. La persistencia es la capacidad de un objeto de mantener su valor en el espacio y en el tiempo. Lo contrario son las clases temporales que son manejadas y almacenadas por el sistema en tiempo de ejecución, por lo que dejan de existir cuando termina el programa.

Una clase del diseño es una construcción similar a la implementación del sistema:

- Las relaciones entre clases de diseño se traducen de manera directa al lenguaje:
 - ✓ Generalización: herencia.
 - ✓ Asociaciones, agregaciones: atributos.
- Los métodos de una clase del diseño tienen correspondencia directa con el correspondiente método en la implementación de las clases.
- Una clase de diseño puede proporcionar interfaces si tiene sentido hacerlo en el lenguaje de programación.

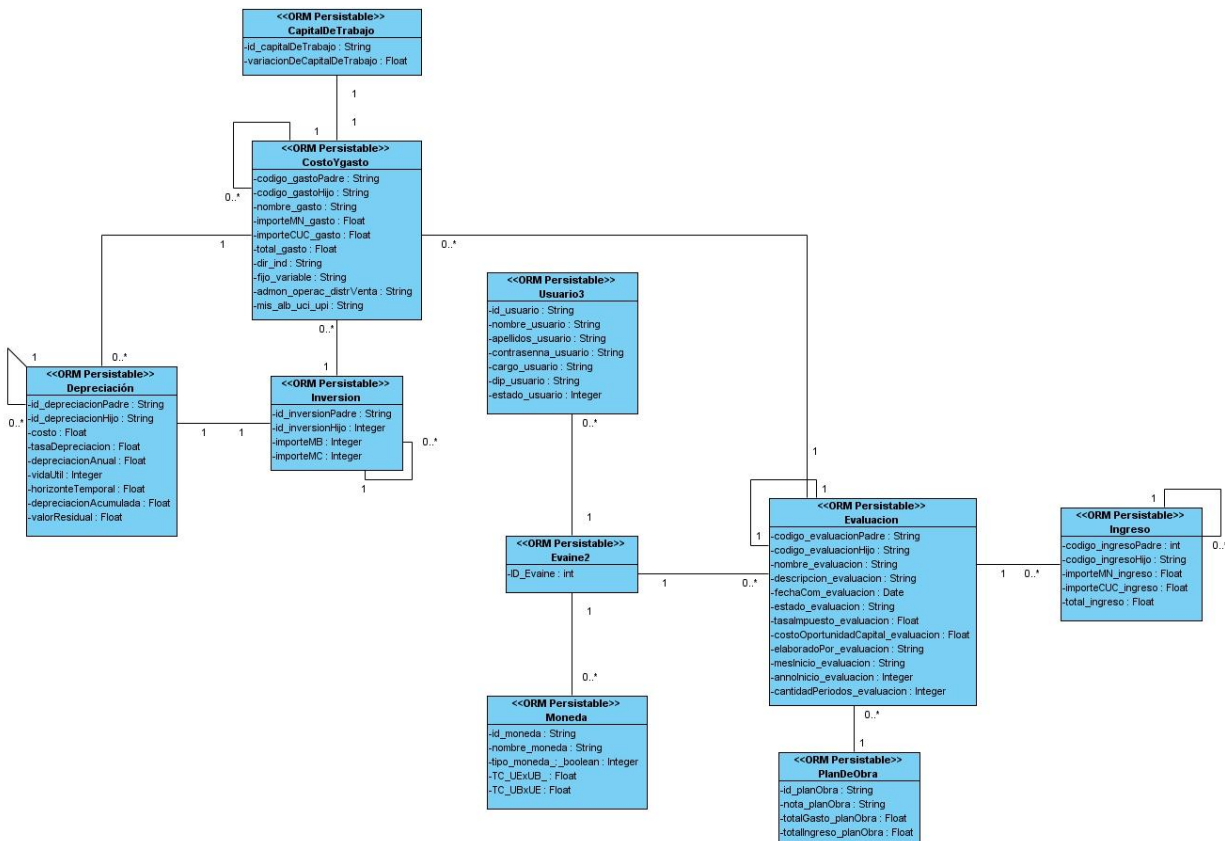


Figura 3.2 Diagrama de clases persistentes.

3.2.3.2 Modelo de datos

Una base de datos es una serie de datos organizados y relacionados entre sí, los cuales pueden ser recolectados y explotados por sistemas de gestión de información. Con el objetivo de lograr la persistencia de los datos y que estos puedan ser utilizados en cualquier momento por el usuario del sistema encargado de gestionar la información se utiliza una base de datos relacional. El uso de esta técnica de almacenamiento está muy difundido actualmente debido a las garantías que ofrece en cuanto a la durabilidad de los datos se refiere.

El modelo de los datos es el encargado de describir de una forma abstracta la representación lógica y física de los datos persistentes en el sistema. Básicamente consiste en una colección de conceptos que se emplean para describir la estructura de la base de datos y que está constituida por entidades, atributos y relaciones.

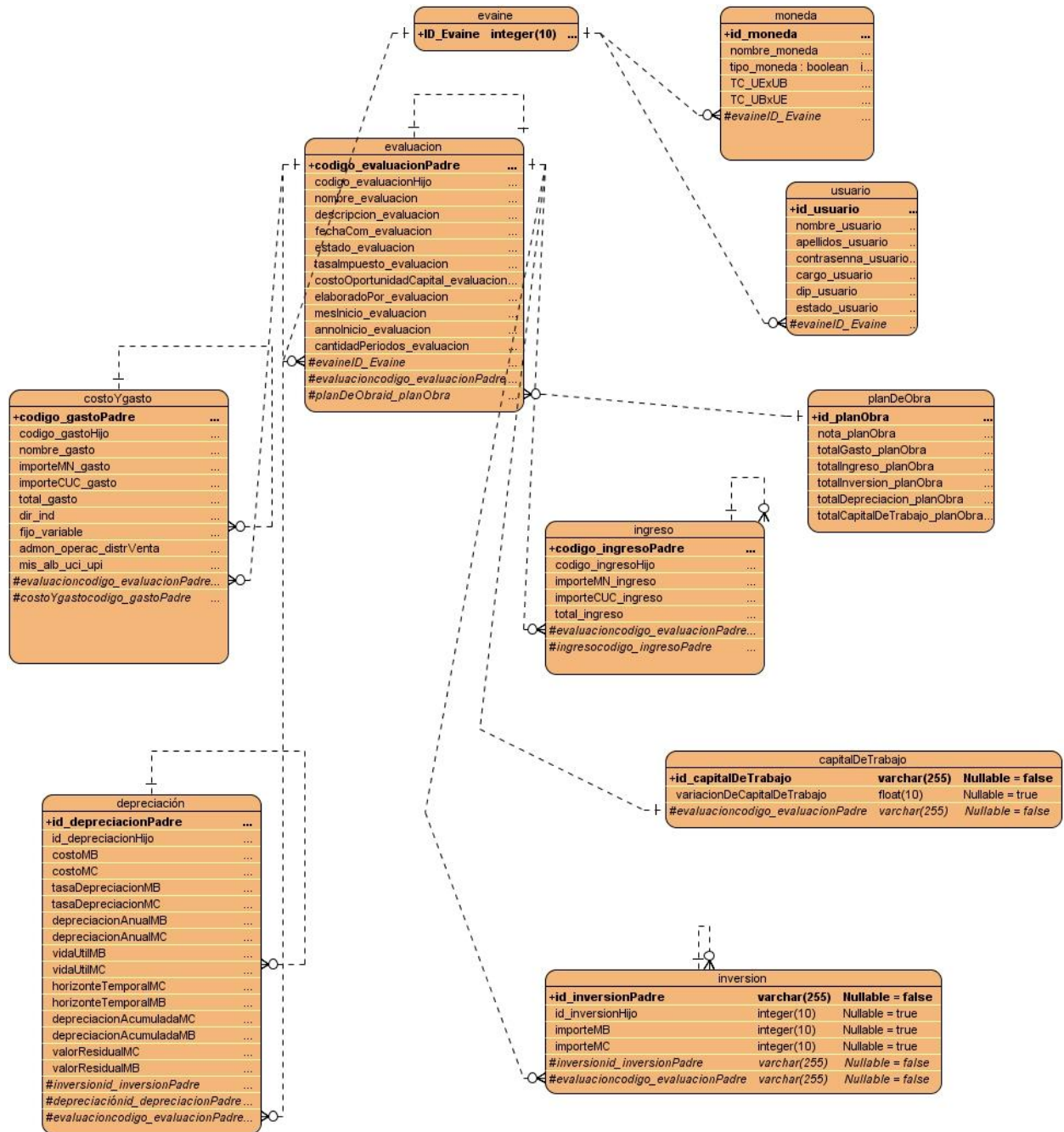


Figura 3.3: Modelo de datos.

3.2.3.2.1 Descripción de las tablas de la Base de Datos

Se describen las tablas en las que se almacena toda la información necesaria para desarrollar el sistema, permitiendo conocer el atributo, su tipo y descripción.

3.2.4 Interfaz

La interfaz es la ventana del software por lo que debe ser: amigable, sencilla, de fácil entendimiento y manejo para el usuario. Este es el elemento más importante para el usuario, ya que es la parte del sistema con la que interactúa y que le facilita además el acceso a los recursos. El diseño de esta interfaz comenzó con la identificación de los requisitos del usuario, de las tareas y del entorno. Debe reflejar equilibrio en la organización de la información, mostrando en todas las interfaces la información en el mismo orden, optimizar la cantidad de elementos en la pantalla, ayudando a una mejor comprensión de la información mostrada, además, cada elemento de la pantalla seguirá el mismo patrón de tamaño, color y forma.

3.2.5 Tratamiento de errores

Para una mayor seguridad y confiabilidad de los usuarios con el sistema se lleva a cabo el tratamiento de los errores que se puedan generar durante el trabajo con la aplicación. Se ve de manifiesto cuando utilizamos expresiones regulares brindadas por la tecnología de programación Java, que proporciona una fortaleza, garantizando rapidez de ejecución del sistema, recibiendo el servidor solo la información correcta, de forma tal que las operaciones de inserción o modificación de registros en la base de datos se realicen correctamente.

Esto también se garantiza en las clases interfaz de usuario donde se implementan funciones que validan toda la entrada de datos permitiendo esto que no se introduzcan datos erróneos, en caso que el usuario desee introducir algún dato en un formato diferente del requerido se le notificará del error, también lo hará cuando deje algún campo obligatorio en blanco. Cada vez que es lanzada una excepción u ocurre un error en el sistema el usuario inmediatamente es notificado de dicho error y de las posibles causas que generaron el mismo. Estos son descritos en un lenguaje que el usuario entiende y los mensajes proporcionan consejos constructivos para recuperarse del error.

Las entradas de datos a las descripciones de cada documento son validadas para evitar que se produzcan errores en el futuro. Utilizando mensajes de confirmación para acciones tales como la creación y modificación de la información, estos se manejan de forma que el usuario sepa en cada momento como se realizan las acciones que él lleva a cabo.

La notificación se realiza con el objetivo de que el usuario sepa en todo momento la acción que va a realizar, ya que la misma puede traer problemas posteriores.

La notificación de los errores se realiza mediante mensajes de error que genera el sistema, de tal modo que el usuario entienda la naturaleza del error y pueda corregirlo.

3.2.6 Concepción de la Ayuda

Se realizará con JavaHelp, ya que se pueden crear las ventanas típicas de ayuda de las aplicaciones informáticas, en las que sale en el lado izquierdo un panel con varias pestañas: índice de contenidos, búsqueda, temas favoritos, índice alfabético, entre otros. Se incluirá una explicación detallada de cómo acceder y trabajar con el sistema, las restricciones que tienen como usuario y cada una de las funcionalidades a las que tienen acceso. Esto le facilitará el

trabajo con la aplicación, permitiéndole a cada momento la posibilidad de saber cómo hacer algo o cómo resolver cierto problema que se le presentase, asegurando con esto que el uso del producto sea lo más eficiente posible.

3.2.7 Seguridad

Debido a la importancia y sensibilidad de los datos que se manejan en la aplicación se hace necesario garantizar la protección y seguridad de la información para lograr el óptimo funcionamiento del sistema y preservar la integridad, autenticidad y confiabilidad de la misma. La seguridad es un elemento esencial cuando se va a desarrollar un sistema informático por tanto, se hace necesario implementar algunas tareas de estricto cumplimiento que permitan asegurar la integridad de los datos. La base de datos permanecerá en lugar restringido y asegurado, además de realizarse salvadas de la información acumulada.

Hay que tener en cuenta que el acceso a la aplicación es de forma controlada, cada usuario tendrá acceso a la información que le concierne. De forma general, en el sistema ninguna información registrada podrá ser eliminada. Esto garantiza que no sea eliminado algún elemento por error, uno de los aspectos más importantes es que tampoco los usuarios registrados podrán ser eliminados, ya que, en cada acción realizada en el sistema se queda guardado el nombre de usuario que realizó la misma, permitiendo saber en todo momento quien realizó algún cambio en el sistema, lo que garantiza la seguridad de la información almacenada. Estarán validados todos los campos de entrada de datos, garantizando que no se introduzca información que posteriormente pueda afectar el buen funcionamiento de la aplicación.

Para lograr una mayor seguridad en los datos se usará la librería Jasypt 1.5. Jasypt (Java Simplified Encryption) es una librería orientada a permitir a los desarrolladores añadir capacidades de cifrado a sus proyectos de manera sencilla, incluyendo: password digesting, cifrado de textos o binarios, integración con Hibernate para cifrado transparente de los datos persistentes e integración con Spring Security²¹ para mejorar la seguridad de los passwords manejados por este framework. También permite el cifrado de valores en archivos .properties y su lectura de manera transparente. Además de usar como se explicó anteriormente el protocolo de encriptación SSL para la conexión segura con el servidor de base de datos.

3.3 Conclusiones

En este capítulo se obtuvo el modelo de análisis y el modelo del diseño para el sistema propuesto. Se obtuvo diferentes artefactos de este flujo de trabajo como son los diagramas de clases del análisis y del diseño, mediante la cual se explicó de forma más detallada las responsabilidades de cada clase, lográndose de esta forma una mayor visión y mejor comprensión de lo que se quiere obtener en el sistema. Se plantearon los principios del diseño incluyendo en él aquellos patrones utilizados para modelar en diseño teniendo en cuenta la

²¹ Spring Security: Antes conocido como Acegi es una API código abierto que provee de servicios de autenticación y autorización a las aplicaciones basadas en Spring.

arquitectura propuesta. Además, se describieron la interfaz, el tratamiento a los errores, la concepción de ayuda y la seguridad del sistema.

CONCLUSIONES

- Con la realización de este trabajo se concluyó el diseño de una herramienta informática que permita realizar evaluaciones económicas a proyectos de inversión.
- Luego de un estudio de los sistemas contables existentes, se demostró que estos no cumplen con los requerimientos exigidos por la UPI “Proyecto Futuro” para llevar a cabo la evaluación de económica de inversiones.
- Se modelaron los flujos de trabajo propuestos por el Proceso Unificado de Desarrollo: Modelado del Negocio, Gestión de Requerimientos, Análisis y Diseño, obteniéndose los artefactos que se generan en cada uno de estos flujos y utilizándose como Metodología de Desarrollo de Software el Proceso Unificado de Desarrollo (RUP), el Lenguaje Unificado de Modelado (UML) y Visual Paradigm for UML 6.4 como herramienta CASE.
- Al realizar un análisis detallado del Modelo de Negocio y el Sistema se obtuvieron 18 requisitos funcionales como resultado del levantamiento de requisitos de software.
- Las herramientas, tecnologías y lenguajes utilizados están de acuerdo con las premisas de soberanía tecnológica por las que aboga la Universidad.
- Las evaluaciones económicas de inversiones se harán con mayor rapidez, habrá mayor seguridad en el manejo de la información y los reportes que se generen ayudarán a tomar decisiones económicas.

RECOMENDACIONES

Los autores recomiendan:

- Realizar la implementación y prueba del Módulo de Evaluación Económica para Proyectos de Inversión.
- Realizar modificaciones y mejoras del Módulo de Evaluación Económica para Proyectos de Inversión, a partir del despliegue del mismo.
- Desarrollar otros sub-módulos de Evaluación Financiera, Balanza de Pagos y Modelo de Rentabilidad del país.

REFERENCIAS BIBLIOGRÁFICAS

1. **Carreras Leyva, Alejandro Manuel and Flores Rodríguez, Yaniris. 2009.** *Sistema de Control de Inversiones "ConInver"*. Ciudad de La Habana : Universidad de las Ciencias Informáticas, 2009.
2. **UPI "Proyecto Futuro".** *Manual de Procedimientos Proceso Inversionista*.
3. **Ministerio de Economía y Planificación. 2001.** *Bases metodológicas para la elaboración de los estudios de factibilidad de las inversiones industriales*. La Habana : s.n., 2001.
4. **B. Foote and R. Johnson. 1988.** *Designing Reusable Classes*. 1988.
5. **Booch, G. 1996.** *Análisis y diseño orientado a objetos con aplicaciones. (2 ed.)*. s.l. : (J. M. Cerva Lovelle, Trad.) Addison-Wesley, 1996.
6. **Flanagan, D. 1999.** *Java en pocas palabras (2da ed., Vol. 1)*. México : (McGRAW-HILL, Ed.), 1999.
7. **Kruchten, P. 2003.** *The Rational Unified Process An Introduction. Third Edition*. 2003.
8. **Marioko. 2008.** *javaHispano*. [Online] Noviembre 19, 2008. [Cited: Diciembre 15, 2008.] http://www.javahispano.org/contenidos/es/netbeans_6_5_final/?menuId=NEWS .
9. **NetBeans (s. f.).** NetBeans. [Online] [Cited: diciembre 3, 2009.] http://www.netbeans.org/index_es.html.
10. **Popkin Software and Systems.** La Web del Programador. [Online] [Cited: enero 26, 2010.] <http://es.tldp.org/Tutoriales/doc-modelado-sistemas-UML/doc-modelado-sistemas-uml.pdf>.
11. **PostgreSQL. (s.f.).** PostgreSQL. [Online] [Cited: Diciembre 5, 2009.] <http://www.postgresql.org/about/>.
12. **S. H. Kaiser. 2005.** *Software Paradigms*. 2005.
13. **V. Russo and R. Johnson. 1991.** *Reusing Object-Oriented Designs*. 1991.
14. **Y. Saborit Ramírez and N. M. Soto López. 2004.** *Sistema de catalogación y recuperación de recursos de información Hubble*. Ciudad de la Habana : s.n., 2004.
15. **TiendaLinux.com. (26 de Diciembre de 2005).** Obtenido de TiendaLinux.com: http://soporte.tiendalinux.com/portal/Portfolio/postgresql_ventajas_html

BIBLIOGRAFÍA

1. Anónimo. "Ingeniería de Software con UML Unified Modeling Language - Lenguaje Unificado de Modelado." <http://www.eduardoleyton.com/apuntes/Uml.pdf>.
2. B. Foote, and R. Johnson. *Designing Reusable Classes*. 1988.
3. Booch, G. *Análisis y diseño orientado a objetos con aplicaciones*. (2 ed.). (J. M. Cerva Lovelle, Trad.) Addison-Wesley, 1996.
4. Carreras Leyva, Alejandro Manuel, and Yaniris Flores Rodríguez. *Sistema de Control de Inversiones "ConInver"*. Ciudad de La Habana: Universidad de las Ciencias Informáticas, 2009.
5. Coloma, F. *Evaluación social de proyectos de inversión*. La Paz. Bolivia: Asociación Internacional de Fomento- Bco, 1991.
6. "Ejemplo de desarrollo software utilizando la metodología XP." <http://www.dsic.upv.es/asignaturas/facultad/lsi/ejemploxp/>. Enero 9, 2004.
7. "El estudio de factibilidad dentro de las etapas de Análisis de Sistemas Administrativos." www.monografias.com.
8. *Entornos De Desarrollo Integrados*. <http://www.slideshare.net/GhaBiithahh/entornos-de-desarrollo-integrados> (accessed enero 18, 2010).
9. "Extreme Programming: A gentle introduction." <http://www.extremeprogramming.org/>. Febrero 17, 2006.
10. Fernández, G. *Introduccion a Extreme Programming. Ingenieria de software II*. 2002.
11. Flanagan, D. *Java en pocas palabras (2da ed., Vol. 1)*. México: (McGRAW-HILL, Ed.), 1999.
12. Fontaine, E. "Evaluación social de proyectos." Ediciones Universidad Católica de Chile. 1997.
13. Issi, G. "Metodologías Ágiles en el Desarrollo de Software." <http://issi.dsic.upv.es/archives/f-1069167248521/actas.pdf#page=9>.
14. Jacobson, I. *El proceso Unificado de Desarrollo de Software*. 1999.
15. Kruchten, P. *A Rational Development Process*. 1996.
16. —. *The Rational Unified Process An Introduction. Third Edition*. 2003.
17. Larman, Craig. *UML y Patrones. Introducción al análisis y diseño orientado a Objetos Volumen I*. Prentice Hall Hispanoamericana, S.A., 1999.

18. —. *UML y Patrones. Introducción al análisis y diseño orientado a Objetos. Volumen II*. Prentice Hall Hispanoamericana, S.A., 1999.
19. Letelier, P. "Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP)." <http://www.cyta.com.ar/ta0502/v5n2a1.htm>.
20. Marioko. Noviembre 19, 2008.
http://www.javahispano.org/contenidos/es/netbeans_6_5_final/?menuId=NEWS (accessed Diciembre 15, 2008).
21. Mideplan. *Manual de Preparación y Presentación de proyectos de Inversión*. Santiago de Chile, 1998.
22. Ministerio de Economía y Planificación. *Bases metodológicas para la elaboración de los estudios de factibilidad de las inversiones industriales*. La Habana, 2001.
23. NetBeans (s. f.). *NetBeans*. http://www.netbeans.org/index_es.html (accessed diciembre 3, 2009).
24. Padrón, J. "Qué es un problema de Investigación." 1996.
http://www.geocities.com/josepadron.geo/Que_es_un_problema.htm.
25. Pavón, Eduardo León. "Visual Paradigm, una herramienta de lo más útil." abril 2, 2007.
<http://slion2000.blogspot.com/2007/04/visual-paradigm-una-herramienta-de-lo.html>.
26. Popkin Software and Systems. "La Web del Programador." <http://es.tldp.org/Tutoriales/doc-modelado-sistemas-UML/doc-modelado-sistemas-uml.pdf> (accessed enero 26, 2010).
27. PostgreSQL. (s.f.). *PostgreSQL*. <http://www.postgresql.org/about/> (accessed Diciembre 5, 2009).
28. Presidente de los Consejos de Estado y de Ministros de Cuba, General de Ejército Raúl Castro Ruz . "Intervención en la sesión de la Asamblea Nacional ." Ciudad de La Habana, Diciembre 2009.
29. Pressman, Roger. *Ingeniería de software. Un enfoque práctico. Parte 1*. Interamericana de España, S.A., 2002.
30. —. *Ingeniería de software. Un enfoque práctico. Parte 2*. Interamericana de España, S.A., 2002.
31. "Proyecto de inversión." www.monografias.com.
32. S. H. Kaiser. *Software Paradigms*. 2005.
33. Sepúlveda, F. *El Proceso de Evaluación de Proyectos. Informe Económico Regional*. Facultad de Ciencias Económicas y Administrativas de la Universidad de Concepción, 1995.

34. *Software Evaluación de Proyectos*. http://www.elsitioagricola.com/Soft/evalas/evalAs1_0.asp# (accessed diciembre 14, 2009).
35. *TiendaLinux.com*. Diciembre 26, 2005.
http://soporte.tiendalinux.com/portal/Portfolio/postgresql_ventajas_html.
36. UPI “Proyecto Futuro”. *Manual de Procedimientos Proceso Inversionista*.
37. V. Russo, and R. Johnson. *Reusing Object-Oriented Designs*. 1991.
38. Y. Saborit Ramírez, and N. M. Soto López. *Sistema de catalogación y recuperación de recursos de información Hubble*. Ciudad de la Habana, 2004.

GLOSARIO DE TÉRMINOS

Eclipse: Entorno de desarrollo integrado de código abierto multiplataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido".

Excel: Programa de Microsoft, el cual consiste en una hoja de cálculo, utilizada para realizar fórmulas matemáticas y cálculos aritméticos exhaustivos.

Factibilidad económica: Se refiere a que se dispone del capital en efectivo o de los créditos de financiamiento necesario para invertir en el desarrollo del proyecto, mismo que deberá haber probado que sus beneficios a obtener son superiores a sus costos en que incurrirá al desarrollar e implementar el proyecto o sistema.

Flujo de caja: En finanzas y en economía se entiende por flujo de caja o flujo de fondos (en inglés cash flow) los flujos de entradas y salidas de caja o efectivo, en un período dado.

Framework: es una estructura conceptual y tecnológica de soporte definida, normalmente con artefactos o módulos de software concretos, con base en la cual otro proyecto de software puede ser organizado y desarrollado

Groovy: Lenguaje de programación orientado a objetos implementado sobre la plataforma Java.

Herramienta CASE: Aplicación informática destinada a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero, se utiliza para la modelación del sistema.

Hibernate: Potente framework de mapeo objeto/relacional y servicio de consultas para Java. Es la solución ORM (Object-Relational Mapping) más popular en el mundo Java.

HTML: Sigla en inglés de HyperText Markup Language (Lenguaje de Marcas de Hipertexto), es el lenguaje de marcado predominante para la construcción de páginas web.

IDE: Un entorno de desarrollo integrado es un programa informático compuesto por un conjunto de herramientas de programación.

Java: Lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 90.

.NET: Plataforma de desarrollo de software con énfasis en transparencia de redes, con independencia de plataforma de hardware y que permita un rápido desarrollo de aplicaciones.

NetBeans: es una herramienta de desarrollo para aplicaciones, escrita puramente sobre la base de la tecnología Java.

PAYBACK: El payback o "plazo de recuperación" es un criterio de valoración de inversiones que permite seleccionar un determinado proyecto en base a cuánto tiempo se tardará en recuperar la inversión inicial mediante los flujos de caja

PDF: Sigla en inglés de Portable Document Format (Formato de Documento Portátil), es un formato de almacenamiento de documentos.

PostgreSQL: SGBD relacional orientada a objetos de software libre, publicado bajo la licencia BSD.

RUP: El Proceso Unificado Racional (Rational Unified Process en inglés, habitualmente resumido como RUP) es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

SGBD: Los sistemas de gestión de bases de datos son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan.

Software: Es el conjunto de programas e instrucciones asociados a una computadora. La parte intangible que hace funcionar un sistema informático y que puede ser modificada con facilidad.

TIR: La tasa interna de retorno o tasa interna de rentabilidad (TIR) de una inversión, está definida como la tasa de interés con la cual el valor actual neto o valor presente neto (VAN o VPN) es igual a cero

UML: Lenguaje Unificado de Modelado (LUM) o (UML, por sus siglas en inglés, Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema.

VAN: Valor actual neto procede de la expresión inglesa Net present value. El acrónimo es NPV en inglés y VAN en español. Es un procedimiento que permite calcular el valor presente de un determinado número de flujos de caja futuros, originados por una inversión

4+1 vistas de la arquitectura: Vista de casos de uso, vista lógica, vista de procesos, vista de componentes y vista de despliegue.

