INSTITUTO SUPERIOR MINERO METALURGICO

“Dr. Antonio Núñez Jiménez”.

Facultad de Metalurgia - Electromecánica

Moa, Holguín

“SISTEMA AUTOMATIZADO PARA EL DIAGNOSTICO DEL CONTROL INTERNO”

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

Autor: Yohander Calzadilla Pérez

Tutor (es): Msc. Gustavo Rodríguez Barcenas
Lic. Andrés Fonseca Hernández

Moa, Cuba
Junio, 2009

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo al Centro INSTITUTO SUPERIOR MINERO METALURGICO “Dr. Antonio Núñez Jiménez” para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmo la presente a los _____ días del mes de _____ del 2009.

Yohander Calzadilla Pérez
Nombre completo del primer autor

Msc. Gustavo Rodríguez Barcenas
Nombre completo del primer tutor

Lic. Andrés Fonseca Hernández
Nombre completo del segundo tutor

OPINIÓN DEL USUARIO DEL TRABAJO DE DIPLOMA

El Trabajo de Diploma, titulado “Sistema Automatizado para el Diagnóstico del Sistema de Control Interno”, fue realizado en nuestra entidad Instituto Superior Minero Metalúrgico de Moa Dr. Antonio Núñez Jiménez. Se considera que, en correspondencia con los objetivos trazados, el trabajo realizado le satisface:

- ☐ Totalmente
- ☐ Parcialmente en un ____ %

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

_____

_____

_____

_____

_____

_____

Como resultado de la implantación de este trabajo se reporta un efecto económico que asciende a _____ MN y/o _____ CUC.

Y para que así conste, se firma la presente a los ____ días del mes de _____ del año 2009

_____ _____

Nombre del representante de la entidad Cargo

_____

Firma Cuño

OPINIÓN DEL TUTOR DEL TRABAJO DE DIPLOMA

Título: Sistema Automatizado para el Diagnóstico del Sistema de Control Interno

Autor: Yohander Calzadilla Pérez.

El tutor del presente Trabajo de Diploma considera que durante su ejecución el estudiante mostró las cualidades que a continuación se detallan.

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

Por todo lo anteriormente expresado considero que el estudiante está apto para ejercer como Ingeniero Informático; y propongo que se le otorgue al Trabajo de Diploma la calificación de:

_____ _____

Nombre completo del primer tutor Nombre completo del segundo tutor

Fecha: _____

Agradecimientos

A Dios por protegerme y permitir hacer realidad mi sueño…

A mis padres y mi hermana que me alentaron, me brindaron su amor su cariño y que siempre me supieron inculcar los mejores principios éticos y morales…

A mis abuelos y demás familiares que me brindaron su apoyo, su cariño y su ayuda siempre que lo necesite…

A mis compañeros y a todos los que colaboraron para hacer posible la realización de este trabajo…

A todos gracias…

…Yohander Calzadilla Pérez…

Dedicatoria

A mis queridos Padres que con tanto amor y cariño me han sabido guiar para hacer realidad este sueño…

Resumen

Con el paso de la historia los problemas de Control Interno, han centrado la preocupación de la gerencia moderna, así como de los profesionales responsables de implementar nuevas formas de mejorar y perfeccionar dichos controles.

En nuestro país a pesar de que se ha avanzado en el conocimiento de la Resolución 297/2003 del Ministerio de Finanzas y Precios(MFP), en las áreas administrativas de una entidad así como en la entidad son escasos los diagnósticos que se realizan por parte de directivos al Sistema de Control Interno (SCI) de su área, de ahí que la línea central de esta investigación y principal resultado de la misma es la elaboración de un software capaz de crear, diagnosticar y administrar guías metodológicas para la evaluación del SCI en una entidad o área administrativa, lo cual en gran medida favorecerá al control de los procesos para una mejor gestión y calidad de los mismos.

Este software puede llevarse a cabo por directivos, auditores, especialistas en la temática o cualquier otra persona interesada en esta actividad.

Se le propone al Comité de Control y Prevención y al Grupo de Auditoria del Instituto Superior Minero Metalurgico la aplicación del software en el desarrollo de su trabajo como una de las acciones para validar el mismo y proponer su implementación en otras entidades empresariales.

Summary

With the step of story the problems of Internal Control, have focused the concern of modern management, thus as of the professionals responsible for implementing new ways to improve and refine those controls.

In our country, despite the fact that has advanced in the knowledge of the Resolution 297/2003 of the Ministry of Finance and Prices (MFP) in an entity's administrative areas thus as in the entity are few diagnoses are made by directors to the Internal Control System (ICS) in your area, so the center line of this investigation and the main result is the development of a software able to create, manage and diagnose methodological guides for the evaluation of ICS entity or an administrative area, which to a large extent will favor the control of processes for better management and quality of the same ones.

This software can take end by directors, auditors, specialists in the subject or anyone person interested in this activity.

It proposes to the Committee on the Prevention and Control and Audit Group of the Higher Institute of Mining Metallurgical application software in the development of his work as an action to validate it and propose its implementation in other business entities.

Índice

INTRODUCCIÓN **5**

CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA **6**

1.1 INTRODUCCIÓN 6
1.2 ESTADO DEL ARTE 6
1.2.1 Conceptos fundamentales *6*
1.3 SISTEMAS AUTOMATIZADOS EXISTENTES VINCULADOS AL CAMPO DE ACCIÓN 10
1.4 TENDENCIAS Y TECNOLOGÍAS ACTUALES PARA EL DESARROLLO DE APLICACIONES 11
1.4.1 Lenguajes de programación *11*
1.4.2 Sistemas Gestores de Base de Datos (SGBD) *16*
1.4.3 Metodologías para el desarrollo de sistemas informáticos *20*
1.5 FUNDAMENTACIÓN DE LA SELECCIÓN DEL LENGUAJE A UTILIZAR 23
1.6 FUNDAMENTACIÓN DEL SGBD A UTILIZAR 24
1.7 CONCLUSIONES DEL CAPÍTULO 24

CAPITULO 2 MODELO DE DOMINIO **26**

2.1 INTRODUCCIÓN 26
2.2 DEFINICIÓN DE LAS ENTIDADES Y LOS CONCEPTOS PRINCIPALES. 26
2.3 REPRESENTACIÓN DEL MODELO DE DOMINIO. 27
2.4 REQUERIMIENTOS FUNCIONALES Y NO FUNCIONALES DEL SISTEMA. 28
2.4.1 Requerimientos Funcionales. *28*
2.4.2 Requerimientos no Funcionales *28*
2.5 CONCLUSIONES DEL CAPÍTULO 29

CAPITULO 3 DISEÑO E IMPLEMENTACIÓN DEL SISTEMA **30**

3.1 INTRODUCCIÓN 30
3.2 ACTORES DEL SISTEMA 30
3.3 DIAGRAMA DE CASOS DE USO DEL SISTEMA. 30
3.4 DESCRIPCIÓN DE LOS CASOS DE USO 31
3.5 DIAGRAMA DE CLASES DEL DISEÑO 31
3.5.1 Diagrama de clases general. *31*
3.5.2 Descripción general de las clases *32*
3.6 PRINCIPIOS DE DISEÑO 33
3.6.1 Interfaz de usuario "SADCI" *33*
3.6.2 Ayuda *33*
3.7 DISEÑO DE LA BASE DE DATOS 33
3.7.1 Modelo Lógico de Datos *33*
3.7.2 Modelo Físico de Datos *34*

3.8 DIAGRAMAS DE SECUENCIA 35
3.9 DIAGRAMA DE DESPLIEGUE 35
3.10 DIAGRAMA DE COMPONENTES 35
3.11 CONCLUSIONES DEL CAPÍTULO 36

CAPÍTULO 4 ESTUDIO DE FACTIBILIDAD 37

4.1 INTRODUCCIÓN 37
4.2 ESTUDIO DE LA FACTIBILIDAD 37
4.3 PLANIFICACIÓN POR PUNTOS DE FUNCIÓN 37
4.4 ESTIMACIÓN DE LA CANTIDAD DE INSTRUCCIONES FUENTES 40
4.5 APLICACIÓN DE LAS FÓRMULAS DE BOHEM 41
4.5.1 Cálculo del esfuerzo (PM) 41
4.5.2 Cálculo del tiempo de desarrollo (TDEV) 43
4.5.3 Cantidad de hombres 44
4.6 DETERMINAR EL COSTO DEL SOFTWARE 44
4.7 BENEFICIOS TANGIBLES E INTANGIBLES 45
4.8 ANÁLISIS DE COSTOS Y BENEFICIOS 46
4.9 CONCLUSIONES DEL CAPÍTULO 46

CONCLUSIONES GENERALES 47

RECOMENDACIONES 48

REFERENCIAS BIBLIOGRÁFICAS 49

BIBLIOGRAFÍA 50

ANEXO 1 DESCRIPCIÓN DE LOS CASOS DE USOS 51

ANEXO 2 DESCRIPCIÓN DE LAS CLASES 55

ANEXO 3 INTERFAZ DE USUARIO 61

ANEXO 4 DIAGRAMAS DE SECUENCIA 66

Índice de tablas

TABLA 1 ENTIDADES Y CONCEPTOS DEL DOMINIO. 26

TABLA 2 DEFINICIÓN DE LOS ACTORES DEL SISTEMA. 30

TABLA 3 DESCRIPCIÓN DEL CASO DE USO "INSERTAR USUARIOS". 31

TABLA 4 DESCRIPCIÓN DE LA CLASE "EMPRESA". 32

TABLA 5 ENTRADAS EXTERNAS. 37

TABLA 6 SALIDAS EXTERNAS 38

TABLA 7 CONSULTAS EXTERNAS 39

TABLA 8 FICHEROS LÓGICOS INTERNOS 39

TABLA 9 PUNTOS DE FUNCIÓN DESAJUSTADOS 40

TABLA 10 FACTORES DE ESCALA. 41

TABLA 11 MULTIPLICADORES DE ESFUERZO. 42

TABLA 12 CONSTANTES Y FÓRMULAS PARA EL CÁLCULO DE HOMBRES. 44

TABLA 13 CONSTANTES Y FÓRMULAS PARA EL CÁLCULO DE COSTO. 45

TABLA 14 RESULTADOS DE LAS ESTIMACIONES DE ESFUERZO, TIEMPO DE DESARROLLO, CANTIDAD DE HOMBRES Y COSTO DEL PROYECTO. 45

TABLA 15 DESCRIPCIÓN DEL CASO DE USO "MODIFICAR USUARIOS". 51

TABLA 16 DESCRIPCIÓN DEL CASO DE USO "ELIMINAR USUARIOS". 51

TABLA 17 DESCRIPCIÓN DEL CASO DE USO "MOSTRAR LISTADO USUARIOS". 51

TABLA 18 DESCRIPCIÓN DEL CASO DE USO "INSERTAR GUÍA METODOLÓGICA". 51

TABLA 19 DESCRIPCIÓN DEL CASO DE USO "ELIMINAR GUÍA METODOLÓGICA". 51

TABLA 20 DESCRIPCIÓN DEL CASO DE USO "MODIFICAR GUÍA METODOLÓGICA". 52

TABLA 21 DESCRIPCIÓN DEL CASO DE USO "EVALUAR GUÍA METODOLÓGICA". 52

TABLA 22 DESCRIPCIÓN DEL CASO DE USO "INSERTAR MATERIALES DE CONSULTA". 52

TABLA 23 DESCRIPCIÓN DEL CASO DE USO "ELIMINAR MATERIALES DE CONSULTA". 52

TABLA 24 DESCRIPCIÓN DEL CASO DE USO "MOSTRAR LISTADO DE GUÍAS". 53

TABLA 25 DESCRIPCIÓN DEL CASO DE USO "MOSTRAR LISTADO DE EVALUACIONES". 53

TABLA 26 DESCRIPCIÓN DEL CASO DE USO "MOSTRAR LISTADO MATERIALES DE CONSULTA"....53

TABLA 27 DESCRIPCIÓN DEL CASO DE USO "REALIZAR RESUMEN VALORATIVO"....53

TABLA 28 DESCRIPCIÓN DEL CASO DE USO "EXPORTAR RESUMEN"....53

TABLA 29 DESCRIPCIÓN DE LA CLASE "AREA_O_DPTO"....55

TABLA 30 DESCRIPCIÓN DE LA CLASE "COMPONENTES"....55

TABLA 31 DESCRIPCIÓN DE LA CLASE "NORMAS"....55

TABLA 32 DESCRIPCIÓN DE LA CLASE "PREGUNTAS"....56

TABLA 33 DESCRIPCIÓN DE LA CLASE "SUB_PREGUNTAS"....56

TABLA 34 DESCRIPCIÓN DE LA CLASE "TEMATICAS"....56

TABLA 35 DESCRIPCIÓN DE LA CLASE "USUARIOS"....57

TABLA 36 DESCRIPCIÓN DE LA CLASE "MATERIALES_DE_CONSULTA"....57

TABLA 37 DESCRIPCIÓN DE LA CLASE "EVALUACION"....58

TABLA 38 DESCRIPCIÓN DE LA CLASE "GUIA_METODOLOGICA"....58

TABLA 39 DESCRIPCIÓN DE LA CLASE "RESUMEN_VALORATIVO"....59

TABLA 40 DESCRIPCIÓN DE LA CLASE "BASE"....59

Índice de figuras

FIGURA 1 DIAGRAMA DE MODELO DE DOMINIO.....27

FIGURA 2 DIAGRAMA DE CASOS DE USOS DEL SISTEMA.....30

FIGURA 3 DIAGRAMA DE CLASES DEL DISEÑO. 31

FIGURA 4 INTERFAZ DE USUARIO "SADCI".....33

FIGURA 5 DIAGRAMA DE CLASES PERSISTENTE. 33

FIGURA 6 MODELO FÍSICO DE DATOS. 34

FIGURA 7 DIAGRAMA DE SECUENCIA DEL CASO DE USO INSERTAR GUIA METODOLÓGICA.....35

FIGURA 8 DIAGRAMA DE DESPLIEGUE. 35

FIGURA 9 DIAGRAMA DE COMPONENTES.....35

FIGURA 10 INTERFAZ DE USUARIO "INSERTAR GUÍA METODOLÓGICA". 61

FIGURA 11 INTERFAZ DE USUARIO "EVALUAR GUÍA METODOLÓGICA".....61

FIGURA 12 INTERFAZ DE USUARIO "ELIMINAR GUÍA METODOLÓGICA".....62

FIGURA 13 INTERFAZ DE USUARIO "MODIFICAR GUÍA METODOLÓGICA".....62

FIGURA 14 INTERFAZ DE USUARIO "INSERTAR USUARIO". 62

FIGURA 15 INTERFAZ DE USUARIO "ELIMINAR USUARIO".....63

FIGURA 16 INTERFAZ DE USUARIO "MODIFICAR USUARIO".....64

FIGURA 17 INTERFAZ DE USUARIO "INGRESAR MATERIAL DE CONSULTA".....64

FIGURA 18 INTERFAZ DE USUARIO "REALIZAR RESUMEN VALORATIVO".....65

FIGURA 19 DIAGRAMA DE SECUENCIA DEL CASO DE USO "ELIMINAR_GUIA_METODOLÓGICA".....66

FIGURA 20 DIAGRAMA DE SECUENCIA DEL CASO DE USO "ELIMINAR_USUARIO".....66

FIGURA 21 DIAGRAMA DE SECUENCIA DEL CASO DE USO "EXPORTAR_RESUMEN"...67

FIGURA 22 DIAGRAMA DE SECUENCIA DEL CASO DE USO "INSERTAR_GUIA_METODOLÓGICA".....67

FIGURA 23 DIAGRAMA DE SECUENCIA DEL CASO DE USO "INSERTAR_USUARIO".....68

FIGURA 24 DIAGRAMA DE SECUENCIA DEL CASO DE USO "MOSTRAR_LISTADO_EVALUACIONES".....68

FIGURA 25 DIAGRAMA DE SECUENCIA DEL CASO DE USO "MOSTRAR_LISTADO_GUIAS". 68

FIGURA 26 DIAGRAMA DE SECUENCIA DEL CASO DE USO "MOSTRAR_LISTADO_MATERIALES". 69

FIGURA 27 DIAGRAMA DE SECUENCIA DEL CASO DE USO "MOSTRAR_LISTADO_USUARIOS". 69

FIGURA 28 DIAGRAMA DE SECUENCIA DEL CASO DE USO "MODIFICAR_GUIA". 70

FIGURA 29 DIAGRAMA DE SECUENCIA DEL CASO DE USO "MODIFICAR_DATOS_USUARIO". 70

FIGURA 30 DIAGRAMA DE SECUENCIA DEL CASO DE USO "REALIZAR_RESUMEN". ... 71

FIGURA 31 DIAGRAMA DE SECUENCIA DEL CASO DE USO "EVALUAR_GUIA_METODOLÓGICA". 71

Introducción

Desde mediados del siglo XVIII, con la Revolución Industrial, iniciada en Inglaterra, se fueron introduciendo de modo creciente las máquinas que eran operadas por varias personas para la producción de artículos industriales, haciéndose cada vez más complejos los procesos en los cuales intervenían más personas y como consecuencia de esto surge la necesidad de controlar las operaciones.

El origen del Control Interno, suele ubicarse en el tiempo con el surgimiento de la partida doble, que fue una de las medidas de control, pero no fue hasta fines del siglo XIX que los hombres de negocios se preocuparon por formar y establecer sistemas adecuados para la protección de sus intereses. A finales de este siglo, como consecuencia del notable aumento de la producción, los propietarios de los negocios se vieron imposibilitados de continuar atendiendo personalmente los problemas productivos, comerciales y administrativos, viéndose forzados a delegar funciones dentro de la organización conjuntamente con la creación de sistemas y procedimientos que previeran o disminuyeran fraudes o errores, debido a esto comenzó a hacerse sentir la necesidad de llevar a cabo un control sobre la gestión de los negocios, ya que se había prestado más atención a la fase de producción y comercialización que a la fase administrativa u organizativa, reconociéndose la necesidad de crear e implementar sistemas de control como consecuencia del importante crecimiento operado dentro de las entidades. Así nace el control como una función gerencial, para asegurar y constatar que los planes y políticas preestablecidas se cumplan tal como fueron fijadas.

Debido a esto los contadores idearon la comprobación interna, la cual era conocida como: "la organización y coordinación del sistema de contabilidad y los procesos adoptados, que tienen como finalidad brindar a la administración, hasta donde sea posible, práctico y económico el máximo de protección, control e información verídica". [1]

Por mucho tiempo el alcance del sistema de control interno estuvo limitado a las áreas económicas, se hablaba de control interno y se tenía la cultura de que era inherente a las actividades de contabilidad y finanzas; el resto de las áreas

operacionales y de hecho sus trabajadores no se sentían involucrados. No todos los directivos de nuestras organizaciones veían en el Sistema de Control Interno un instrumento de gestión capaz de ser utilizado para lograr la eficiencia y eficacia de las operaciones que se habían propuesto. Otro elemento que atentaba contra la eficiencia de los sistemas de Control Interno era que las actividades de control ya venían establecidas de forma global para todas las entidades del país, minimizando la creatividad de los directivos en el diseño de los objetivos y actividades de control que fueran más eficaces según las características de su entidad. Además, no se contaba con elementos generalizadores que le sirvieran de base a la organización para poder diseñar un sistema de control interno a la medida de sus necesidades. [2]

Con el paso de la historia los problemas de Control Interno, han centrado la preocupación de la gerencia moderna, así como de los profesionales responsables de implementar nuevas formas de mejorar y perfeccionar dichos controles, esto es muy importante por cuanto el Control Interno es fundamental para que una entidad logre alcanzar, a través de una evaluación de su misión y visión, el logro de sus objetivos y metas trazadas, pues de lo contrario sería imposible que se puedan definir las medidas que se deben adoptar para alcanzarlos.

En el proceso de evaluación del Control Interno un auditor debe revisar altos volúmenes de documentos y constatar numerosos criterios, es por esto que el auditor se ve obligado a programar pruebas de carácter selectivo para hacer inferencias sobre la confiabilidad de sus operaciones.

A pesar de que se ha avanzado en el conocimiento de la Resolución 297/2003 del Ministerio de Finanzas y Precios(MFP) en las áreas administrativas de una entidad así como en la entidad son escasos los diagnósticos que se realizan por parte de directivos al Sistema de Control Interno de su área, a la vez que no existen herramientas para poder crear, administrar, evaluar guías de evaluación del Sistema de Control Interno apropiadas a las condiciones y características que se requieran, y mucho menos se muestran evidencias de controles al Sistema de Control Interno de un área.

En la actualidad se llevan a cabo proceso de diagnóstico del Sistema de Control Interno en las entidades a través de la utilización de documentos donde

se recogen los cinco componentes del sistema por medio de guías metodológicas existente en formato digital e impreso, esto se realiza de manera convencional, evaluando y registrando los resultados de igual manera, a partir de la herramienta que se pretende desarrollar se podrá diagnosticar a través de la propia personalización de las distintas normas y actividades dentro de ellas que contienen los componentes del Control Interno y adaptarse a las distintas áreas o departamentos específicos de cada entidad, emitiendo un resultado final, es por ello que se plantea el siguiente **problema de investigación**: ¿Cómo perfeccionar la evaluación del Sistema de Control Interno en las entidades estatales y sus áreas administrativas?

Teniendo en cuenta esta incógnita el **objeto de estudio** se define como el Sistema de Control Interno. De ahí se desprende nuestro **campo de acción** el cual se enmarca en las Guías Metodológicas Automatizadas y Personalizadas del Sistema de Control Interno.

Para darle solución al problema se propone el siguiente **objetivo general**: Desarrollar una aplicación que permita llevar a cabo el diagnóstico del Sistema de Control Interno de forma personalizada para toda entidad o área administrativa.

Como **hipótesis** partimos de la idea de que: Si se desarrolla una herramienta informática para el diagnóstico del Sistema de Control Interno, se perfeccionará la evaluación de Guías metodológicas de forma personalizada.

Para darle cumplimiento al objetivo general, se proponen los siguientes **objetivos específicos**:

- Realizar un estudio y análisis bibliográfico de los antecedentes de los principales conceptos y teorías referidas al objeto de estudio y campo de acción.
- Elaborar los fundamentos teóricos de las principales tecnologías para el sistema a desarrollar.
- Realizar el análisis, diseño e implementación de la herramienta informática que se pretende elaborar.
- Crear un manual de usuario que permita guiar al mismo de una forma sencilla.

Para cumplimentar estos objetivos se han empleado métodos teóricos y empíricos de la investigación científica. Entre los métodos empíricos usados se puede citar la entrevista y el análisis de documentos para la recopilación de la información. Mediante el análisis de la documentación se supo cual era la funcionalidad de los mecanismos actuales de Control Interno, la entrevista nos posibilitó conocer en detalles cuales eran las necesidades de los profesionales para la evaluación del Control Interno y además para determinar los requerimientos que debe de cumplir el sistema a desarrollar.

El presente trabajo consta de introducción, cuatro capítulos, conclusiones, recomendaciones, bibliografía, anexos y glosario de términos:

En el Capítulo 1, Fundamentación Teórica, se ofrece una breve descripción del objeto de estudio y algunos conceptos fundamentales. Se brinda además una panorámica de los sistemas automatizados existentes vinculados al tema y las tendencias y tecnologías actuales así como un análisis crítico de las fuentes utilizadas.

En el Capítulo 2, Modelo de Dominio, se explica el bajo nivel de estructuración del negocio y la necesidad de utilizar un modelo de domino para mostrar la dinámica del sistema, se definen las entidades y conceptos principales, se presenta el modelo de dominio y se muestran los requerimientos con los cuales debe cumplir el sistema.

En el Capítulo 3, Diseño e Implementación del Sistema, se describen en detalles los flujos de trabajos relacionados a estas etapas de diseño e implementación de la metodología utilizada.

En el Capítulo 4, Estudio de Factibilidad, se presenta el estudio de factibilidad de este proyecto, se utilizará la Metodología Costo Efectividad (Beneficio), la cual plantea la conveniencia de la ejecución del proyecto. Se muestran las Conclusiones a las que se arribaron, las recomendaciones que se proponen, la bibliografía utilizada, anexos con información necesaria sobre el trabajo y el glosario de términos.

En el Capítulo 3, Diseño e Implementación del Sistema, se describen en detalles los flujos de trabajos relacionados a estas etapas de diseño e implementación de la metodología utilizada.

En el Capítulo 4, Estudio de Factibilidad, se presenta el estudio de factibilidad de este proyecto, se utilizará la Metodología Costo Efectividad (Beneficio), la cual plantea la conveniencia de la ejecución del proyecto. Se muestran las Conclusiones a las que se arribaron, las recomendaciones que se proponen, la bibliografía utilizada, anexos con información necesaria sobre el trabajo y el glosario de términos.

Capítulo 1 Fundamentación Teórica

1.1 Introducción

En este capítulo se presenta una descripción general del espacio donde radica nuestro objeto de estudio así como del proceso a automatizar además de abordar los conceptos principales asociados al dominio del problema, se realiza un análisis de las herramientas más comunes para el desarrollo de aplicaciones y se fundamenta la elección del lenguaje seleccionado, así como la metodología a utilizar y sus principales características.

1.2 Estado del Arte

El Sistema de Control Interno es de suma importancia para la estructura administrativa de una empresa. El mismo comprende el plan de la organización y todos los métodos coordinados y medidas adoptadas dentro de una empresa con el fin de salvaguardar sus recursos, cuanto mayor y compleja sea una empresa , mayor será la importancia de un adecuado Sistema de Control Interno.

En el Sistema de Control Interno es vital la correcta definición, ejecución y perfeccionamiento de sus cinco componentes. Esto debe formar parte de un mecanismo que se autorregule, operando con información sintética y actualizada, que revise resultados, descubra tendencias e induzca comportamientos en todas las áreas de resultados claves, simultáneamente e interactuando, siguiendo una subyacente cadena causa-efecto entre ellas, donde, el Sistema de Control Interno ha de realizar una importante contribución al desempeño de la organización.

1.2.1 Conceptos fundamentales

Gestión

La gestión es un conjunto de reglas y decisiones dirigidas a incentivar y coordinar acciones, cuyo carácter público está condicionado a que persiguen metas colectivas y se desenvuelve en el marco de unas restricciones jurídico - políticas peculiares.

Del latín gestĭo, el concepto de gestión hace referencia a la acción y al efecto de gestionar o de administrar. Gestionar es realizar diligencias conducentes al

logro de un negocio o de un deseo cualquiera. Administrar, por otra parte, consiste en gobernar, dirigir, ordenar, disponer u organizar.

El término gestión, por lo tanto, implica al conjunto de trámites que se llevan a cabo para resolver un asunto o concretar un proyecto. La gestión es también la dirección o administración de una empresa o de un negocio.

Modelo de gestión

Un modelo de gestión es un esquema o marco de referencia para la administración de una entidad. Los modelos de gestión pueden ser aplicados tanto en las empresas y negocios privados como en la administración pública.

Esto quiere decir que los gobiernos tienen un modelo de gestión en el que se basan para desarrollar sus políticas y acciones, y con el cual pretenden alcanzar sus objetivos.

Sistema o Estructura del Control Interno

Se denomina Sistema de Control Interno al conjunto de objetivos, planes, métodos, políticas, procedimientos y registros que establezcan las instituciones (entidades), incluyendo la actitud de la dirección de una entidad, para ofrecer seguridad razonable respecto al logro de los objetivos del Control Interno con el propósito de:

- Procurar mecanismos adecuados de operación, acordes con las estrategias y fines de las instituciones, que permitan identificar, dar seguimiento y evaluar los riesgos que puedan derivarse de las actividades del negocio, con el propósito de reducir las pérdidas en que puedan incurrir en la realización de actos o hechos voluntarios o involuntarios.
- Delimitar las diferentes funciones y responsabilidades entre sus órganos sociales, unidades administrativas y personal, con el fin de obtener eficiencia y eficacia en la realización de sus actividades.
- Diseñar sistemas de información administrativa y financiera, correcta, precisa, íntegra, confiable y oportuna.
- Coadyuvar permanentemente a la observancia de la normatividad aplicable a las actividades de las instituciones.

Control Interno

Los enfoques dados por diferentes autores que han sido consultados tienen puntos coincidentes al definir el concepto, pues todos destacan que contribuye al logro de los objetivos propuestos por las entidades, que es un plan

estructurado de la organización, que tiene como objetivo primordial salvaguardar los recursos, que contribuye a la información veraz, a promover la eficiencia y en la actualidad se reconoce que involucra a los directivos y trabajadores en general y no solamente a los contadores y auditores.

En Cuba, la Resolución No 297 del 2003 del MFP, expresa cómo el Comité de Normas Cubanas de Contabilidad estableció el concepto de Control Interno acorde con las condiciones de la economía cubana para su aplicación en todas las entidades del país. La referida Resolución define al Control Interno como:

"El proceso integrado a las operaciones efectuado por la dirección y el resto del personal de una entidad para proporcionar una seguridad razonable al logro de los objetivos siguientes:

- Confiabilidad de la información
- Eficiencia y eficacia de las operaciones
- Cumplimiento de las leyes, reglamentos, procedimientos y políticas establecidas
- Control de recursos de todo tipo, a disposición de la entidad". [3]

El Control Interno se divide en:

Control Administrativo, donde se debe observar y aplicar en todas las fases o etapas del proceso administrativo.

Control Financiero - Contable, orientado fundamentalmente hacia el control de las actividades financieras y contables de una entidad.

De la unión de estos dos elementos se deriva el Sistema de Control Interno.

Componentes del Sistema de Control Interno

Los componentes pueden considerarse como un conjunto de normas que son utilizadas para medir el Control Interno y determinar su efectividad. Para operar el Sistema de Control Interno se requiere de los siguientes componentes:

- Ambiente de Control
- Evaluación del Riesgos
- Actividades de Control
- Información y comunicación

- Supervisión y Monitoreo

Guías metodológicas

Las guías metodológicas no es más que una forma de evaluar el Sistema de Control Interno en las cuales las Temáticas y las diferentes tareas en cada una están orientadas, en primer lugar, a que los dirigentes de las entidades logren una mayor identificación con las acciones, métodos, procedimientos, controles, etc. inherentes a cada uno de los cinco componentes del Sistema de Control Interno, a fin de complementar todas las medidas tomadas y, en segundo lugar, efectuar la verificación en una fecha dada el grado de implementación y de aquellos aspectos en los cuales se requiere dedicar mayor esfuerzo y tiempo, tanto por las entidades como por los órganos superiores de dirección empresarial. Estas guías están diseñadas en forma de cuestionarios como método de evaluación los cuales se dividen en tres temáticas:

1. El cumplimiento de las tareas de obligatorio cumplimiento por las entidades dispuestas en la Resolución 297/03.
2. La calidad del cronograma y el cumplimiento en fecha de las tareas programadas.
3. El avance de la implementación del Sistema de Control Interno.

Métodos existentes para la evaluación de Control Interno

El Control Interno comprende el plan de organización y todos los métodos y procedimientos que en forma coordinada se adoptan en una entidad para salvaguardar sus activos, verificar la razonabilidad y confiabilidad de su información financiera y la complementaria administrativa y operacional, promover eficiencia en la operación y provocar adherencia a las políticas prescritas por la administración.

La evaluación del Control Interno se puede realizar a través de diferentes métodos: método descriptivo o de memorándum, método gráfico y método de cuestionarios. Con certeza no puede decirse que cualquiera de los tres métodos es completo o eficaz, aisladamente, en todos los casos. En algunos casos, tal vez sea aplicable el método de gráficos, en otros puede ser conveniente usar el método de cuestionarios y en otros puede ser más fácil o puede ser de mejor interpretación el método descriptivo. En frecuentes ocasiones se usa una combinación de los tres métodos; la recopilación de

antecedentes y la comprobación de los mismos se lleva a efecto usando, tanto gráficos o flujogramas, como cuestionarios o narrativos.

• El método descriptivo consiste, como su nombre lo indica, en describir las diferentes actividades de los departamentos, funcionarios y empleados, y los registros que intervienen en el sistema. Sin embargo, no debe incurrirse en el error de describir las actividades de los departamentos o de los empleados aislada u objetivamente. Debe hacerse la descripción siguiendo el curso de las operaciones a través de su manejo en los departamentos citados, es decir, consiste en la descripción detallada de los procedimientos más importantes y las características del Sistema de Control Interno para las distintas áreas, mencionando los registros y formularios que intervienen.

• El método de flujogramas es aquel que se expone, por medio de cuadros o gráficos. Si se diseña un flujograma del sistema, será preciso que se visualice el flujo de la información y los documentos que se procesan. El flujograma debe elaborarse, usando símbolos estándar, de manera que quiénes conozcan los símbolos puedan extraer información útil relativa al sistema. Si el auditor usa un flujograma elaborado por la entidad, debe ser capaz de leerlo, interpretar sus símbolos y sacar conclusiones útiles respecto al sistema representado por el flujograma.

• El método cuestionarios: Consiste en usar como instrumento para la investigación, cuestionarios previamente formulados que incluyen preguntas acerca de la forma en que se manejan las transacciones u operaciones de las personas que intervienen en su manejo, la forma en que fluyen las operaciones a través de los puestos o lugares donde se definen o se determinan los procedimientos de control para la conducción de las operaciones, es decir, consiste en la evaluación con base en preguntas.

1.3 Sistemas Automatizados existentes vinculados al campo de acción

Con el desarrollo de las nuevas tecnologías el universo de información ha crecido exponencialmente, provocando que los especialistas se vean en la necesidad de hacer uso cada vez más frecuentes de las ventajas que nos brindan los sistemas informáticos para el diagnóstico del Control Interno. En la actualidad existen software que resultan ser eficientes en este aspecto como es el caso de:

Meycor COSO AG, este software ha sido desarrollado por DATASEC. El mismo permite realizar una evaluación del Control Interno según el informe COSO®, evaluación de los riesgos y realizar auditorías de las evaluaciones. Está compuesto por tres módulos: Meycor COSO AG - Eval, Meycor COSO AG - audit y Meycor COSO AG – Web.

El mismo presenta algunas deficiencias:

- Con el producto se pueden realizar evaluaciones según el informe COSO, no según la Resolución 297/2003 emitida por el Ministerio de Finanzas y Precios de nuestro país.
- No permite la personalización de las guías metodológicas según las necesidades de la entidad o el área que se desee evaluar utilizando su propia guía.
- Este producto es un software propietario por tanto no es de libre distribución.

1.4 Tendencias y tecnologías actuales para el desarrollo de aplicaciones

1.4.1 Lenguajes de programación

Los diferentes lenguajes de programación para el desarrollo de aplicaciones de escritorio han adquirido un gran avance en el mundo de la programación, teniendo en cuenta esto y los conocimientos adquiridos durante el estudio de la carrera consideramos como posibles propuestas: Java, C++ y Delphi, siendo estos los más conocidos. Acerca de ellos brindamos una breve descripción y algunas características.

C + +

Es un lenguaje de programación, diseñado a mediados de los años 1980, por Bjarne Stroustrup, como extensión del lenguaje de programación C.

El nombre C++ fue propuesto por Rick Masciatti en el año 1983, cuando el lenguaje fue utilizado por primera vez fuera de un laboratorio científico. Antes se había usado el nombre "C con clases".

Se puede decir que abarca tres paradigmas de la programación:

La programación estructurada.

La programación genérica.

La programación orientada a objetos.

Además posee una serie de propiedades difíciles de encontrar en otros lenguajes de alto nivel

• Posibilidad de redefinir los operadores (sobrecarga de operadores)

• Identificación de tipos en tiempo de ejecución

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

Delphi

Es un entorno de desarrollo de software (IDE) diseñado para la programación de propósito general con énfasis en la programación visual. El Delphi utiliza como lenguaje de programación una versión moderna de Pascal llamada Object Pascal. Es producido comercialmente por la empresa estadounidense CodeGear. En sus diferentes variantes, permite producir archivos ejecutables para Windows, Linux y la plataforma .NET.

Está basado en una versión moderna de Pascal, denominada Object Pascal. Borland en los últimos años defendía que el nombre correcto del lenguaje es también Delphi, posiblemente debido a pretensiones de marca, aunque en sus mismos manuales el nombre del lenguaje aparecía como Object Pascal, por lo que la comunidad de programadores no ha adoptado mayoritariamente este cambio (supuesta aclaración, según Borland). Object Pascal expande las funcionalidades del Pascal estándar:

- Soporte para la programación orientada a objetos (habitualmente llamada POO) también existente desde Turbo Pascal 5.5, pero más evolucionada en cuanto a:
    - Encapsulación: declarando partes privadas, protegidas, públicas y publicadas de las clases.
    - Propiedades: concepto nuevo que luego han adaptado muchos otros lenguajes. Las propiedades permiten usar la sintaxis de asignación para setters y getters.
    - Simplificación de la sintaxis de referencias a clases y punteros.

- Soporte para manejo estructurado de excepciones, mejorando sensiblemente el control de errores de usuario y del sistema.
- Programación activada por eventos (event-driven), posible gracias a la técnica de delegación de eventos. Esta técnica permite asignar el método de un objeto para responder a un evento lanzado sobre otro objeto. Fue adoptada por Niklaus Wirth, autor del Pascal Original, e incorporada a otros de sus lenguajes como Component Pascal.

Java

Java surgió en 1991 cuando un grupo de ingenieros de Sun Microsystems trataron de diseñar un nuevo lenguaje de programación destinado a electrodomésticos. La reducida potencia de cálculo y memoria de los electrodomésticos llevó a desarrollar un lenguaje sencillo capaz de generar código de tamaño muy reducido.

La implementación original y de referencia del compilador, la máquina virtual y las librerías de clases de Java fueron desarrolladas en 1995. Desde entonces, Sun ha controlado las especificaciones, el desarrollo y evolución del lenguaje a través del Java Community Process, si bien otros han desarrollado también implementaciones alternativas de estas tecnologías de Sun, algunas incluso bajo licencias de software libre.

Entre noviembre de 2006 y mayo de 2007, Sun Microsystems liberó la mayor parte de sus tecnologías Java bajo la licencia GNU GPL, de acuerdo con las especificaciones del Java Community Process, de tal forma que prácticamente todo el Java de Sun es ahora software libre (aunque la biblioteca de clases de Sun que se requiere para ejecutar los programas Java todavía no es software libre).

Como características fundamentales de este lenguaje tenemos:

- **Lenguaje simple**

Java posee una curva de aprendizaje muy rápida. Resulta relativamente sencillo escribir applets interesantes desde el principio. Todos aquellos familiarizados con C++ encontrarán que Java es más sencillo, ya que se han eliminado ciertas características, como los punteros. Debido a su semejanza con C y C++, y dado que la mayoría de la gente los conoce aunque sea de

forma elemental, resulta muy fácil aprender Java. Los programadores experimentados en C++ pueden migrar muy rápidamente a Java y ser productivos en poco tiempo.

- **Orientado a objetos**

Java fue diseñado como un lenguaje orientado a objetos desde el principio. Los objetos agrupan en estructuras encapsuladas tanto sus datos como los métodos (o funciones) que manipulan esos datos. La tendencia del futuro, a la que Java se suma, apunta hacia la programación orientada a objetos, especialmente en entornos cada vez más complejos y basados en red.

- **Distribuido**

Java proporciona una colección de clases para su uso en aplicaciones de red, que permiten abrir sockets y establecer y aceptar conexiones con servidores o clientes remotos, facilitando así la creación de aplicaciones distribuidas.

- **Interpretado y compilado a la vez**

Java es compilado, en la medida en que su código fuente se transforma en una especie de código máquina, los bytecodes, semejantes a las instrucciones de ensamblador. Por otra parte, es interpretado, ya que los bytecodes se pueden ejecutar directamente sobre cualquier máquina a la cual se hayan portado el intérprete y el sistema de ejecución en tiempo real (run-time).

- **Robusto**

Java fue diseñado para crear software altamente fiable. Para ello proporciona numerosas comprobaciones en compilación y en tiempo de ejecución. Sus características de memoria liberan a los programadores de una familia entera de errores (la aritmética de punteros), ya que se ha prescindido por completo los punteros, y la recolección de basura elimina la necesidad de liberación explícita de memoria.

- **Seguro**

Dada la naturaleza distribuida de Java, donde las applets se bajan desde cualquier punto de la Red, la seguridad se impuso como una necesidad de vital importancia. A nadie le gustaría ejecutar en su ordenador programas con acceso total a su sistema, procedentes de fuentes desconocidas. Así que se

implementaron barreras de seguridad en el lenguaje y en el sistema de ejecución en tiempo real.

- **Indiferente a la arquitectura**

Java está diseñado para soportar aplicaciones que serán ejecutadas en los más variados entornos de red, desde Unix a Windows Nt, pasando por Mac y estaciones de trabajo, sobre arquitecturas distintas y con sistemas operativos diversos. Para acomodar requisitos de ejecución tan variopintos, el compilador de Java genera bytecodes: un formato intermedio indiferente a la arquitectura diseñado para transportar el código eficientemente a múltiples plataformas hardware y software. El resto de problemas los soluciona el intérprete de Java.

- **Portable**

La indiferencia a la arquitectura representa sólo una parte de su portabilidad. Además, Java especifica los tamaños de sus tipos de datos básicos y el comportamiento de sus operadores aritméticos, de manera que los programas son iguales en todas las plataformas.

Estas dos últimas características se conocen como la Máquina Virtual Java (JVM).

- **Alto rendimiento**

- **Multihebra**

Hoy en día ya se ven como terriblemente limitadas las aplicaciones que sólo pueden ejecutar una acción a la vez. Java soporta sincronización de múltiples hilos de ejecución (multithreading) a nivel de lenguaje, especialmente útiles en la creación de aplicaciones de red distribuidas. Así, mientras un hilo se encarga de la comunicación, otro puede interactuar con el usuario mientras otro presenta una animación en pantalla y otro realiza cálculos.

- **Dinámico**

El lenguaje Java y su sistema de ejecución en tiempo real son dinámicos en la fase de enlazado. Las clases sólo se enlazan a medida que son necesitadas. Se pueden enlazar nuevos módulos de código bajo demanda, procedente de fuentes muy variadas, incluso desde la Red.

- **Produce applets**

Java puede ser usado para crear dos tipos de programas: aplicaciones independientes y applets. Las aplicaciones independientes se comportan como cualquier otro programa escrito en cualquier lenguaje, como por ejemplo el navegador de Web HotJava, escrito íntegramente en Java. Por su parte, las applets son pequeños programas que aparecen embebidos en las páginas Web, como aparecen los gráficos o el texto, pero con la capacidad de ejecutar acciones muy complejas, como animar imágenes, establecer conexiones de red, presentar menús y cuadros de diálogo para luego emprender acciones, etc.

1.4.2 Sistemas Gestores de Base de Datos (SGBD)

Actualmente existen muchos sistemas gestores de bases de datos, entre ellos, analizaremos las características y facilidades que brinda cada uno de los que se han tenido en consideración, los que siguen son: PostgreSQL, Firebird y MySQL.

PostgreSQL

PostGreSQL es un sistema de gestión de bases de datos objeto-relacional (ORDBMS) basado en el proyecto POSTGRES, de la universidad de Berkeley.
Fue el pionero en muchos de los conceptos existentes en el sistema objeto-relacional actual, incluido, más tarde en otros sistemas de gestión comerciales.
PostGreSQL es un sistema objeto-relacional, ya que incluye características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional. A pesar de esto, no es un sistema de gestión de bases de datos puramente orientado a objetos.
PostgreSQL está considerado como la base de datos de código abierto más avanzada del mundo y proporciona un gran número de características que normalmente sólo se encontraban en las bases de datos comerciales tales como DB2 u Oracle.
Entre las que se encuentran:

- Implementación del Standard SQL92/SQL99.
- Permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos.

- Soporta distintos tipos de datos. Además de los tipos base, soporta de tipo fecha, monetarios elementos gráficos etc. Permite la creación de tipos propios.
- Incorpora una estructura de datos array.
- Incorpora funciones de diversas índoles, tales como, manejo de fechas, geométricas, orientadas a operaciones con redes, etc.
- Permite la declaración de funciones propias. Así como la definición de disparadores.
- Soporta el uso de índices, reglas y vistas.
- Incluye herencia en tablas (aunque no entre objetos, ya que no existen) por lo que a este gestor de bases de datos se le incluye entre los gestores objeto – relacionales.
- Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos.

PostGreSQL tiene prácticamente todo lo que tienen los gestores comerciales, haciendo de él una muy buena alternativa GPL, pero es importante señalar que el primer encuentro con este gestor es un poco "duro", ya que la sintaxis de algunos de sus comandos no es nada intuitiva. Además, carece de un conjunto de herramientas que permitan una fácil gestión de los usuarios y de las bases de datos que contenga el sistema. Por otro lado, la velocidad de respuesta que ofrece este gestor con bases de datos relativamente pequeñas puede parecer un poco deficiente, aunque esta misma velocidad la mantiene al gestionar bases de datos realmente grandes.

Firebird

Firebird es una base de datos relacional que ofrece muchas características de SQL ANSI estándar y que funciona en Linux, Windows, MacOSX y una variedad de plataformas UNIX. Firebird ofrece una concurrencia excelente, alto rendimiento y un poderoso lenguaje de procedimientos almacenados y disparadores. Ha estado usándose en producción bajo varios nombres desde 1981. El Proyecto Firebird es un proyecto independiente de programadores de C y C++, asesores técnicos y colaboradores que desarrollan y mejoran a Firebird, la base de datos relacional basada en el código fuente liberado por Inprise Corp (ahora conocida como Borland Software Corp).

Caracteristicas Fundamentales:

- Es multiplataforma, y actualmente puede ejecutarse en los sistemas operativos: Linux, HP-UX, FreeBSD, Mac OS, Solaris y Microsoft Windows.
- Ejecutable pequeño, con requerimientos de hardware bajos.
- Arquitectura Cliente/Servidor sobre protocolo TCP/IP y otros (embedded).
- Soporte de transacciones ACID y claves foráneas.
- Es medianamente escalable.
- Buena seguridad basada en usuarios/roles.
- Diferentes arquitecturas, entre ellas el Firebird incrustado (embedded server) que permite ejecutar aplicaciones monousuario en ordenadores sin instalar el software Firebird.
- Bases de datos de sólo lectura, para aplicaciones que corran desde dispositivos sin capacidad de escritura, como cd-roms.
- Existencia de controladores ODBC, OLEDB, JDBC, PHP, Perl, .net, etc.
- Requisitos de administración bajos, siendo considerada como una base de datos libre de mantenimiento, al margen de la realización de copias de seguridad.
- Pleno soporte del estándar SQL-92, tanto de sintaxis como de tipos de datos.
- Completo lenguaje para la escritura de disparadores y procedimientos almacenados denominado PSQL.
- Capacidad de almacenar elementos BLOB (Binary Large OBjects).
- Soporte de User-Defined Functions (UDFs).
- Versión autoejecutable, sin instalación, excelente para la creación de catálogos en CD-Rom y para crear versiones de evaluación de algunas aplicaciones.

MySQL

MySQL es un sistema de gestión de bases de datos relacional, licenciado bajo la GPL de la GNU. Su diseño multihilo y multiusuario le permite soportar una gran carga de forma muy eficiente. MySQL fue creada por la empresa sueca MySQL AB, que la desarrolla como software libre en un esquema de licenciamiento dual.

Aunque MySQL es software libre, MySQL AB distribuye una versión comercial de MySQL, que no se diferencia de la versión libre más que en el soporte técnico que se ofrece, y la posibilidad de integrar este gestor en un software propietario, ya que de no ser así, se vulneraría la licencia GPL.

Este gestor de bases de datos es, probablemente, el gestor más usado en el mundo del software libre, debido a su gran rapidez y facilidad de uso. Esta gran aceptación es debida, en parte, a que existen infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración.

MySQL funciona sobre múltiples plataformas, incluyendo AIX, BSD, FreeBSD, HP-UX, GNU/Linux, Mac OS X, NetBSD, Novell Netware, OpenBSD, OS/2 Warp, QNX, SGI IRIX, Solaris, SunOS, SCO OpenServer, SCO UnixWare, Tru64, Windows 95, Windows 98, Windows NT, Windows 2000, Windows XP, Windows Vista y otras versiones de Windows. También existe MySQL para OpenVMS.

Las principales características de este gestor de bases de datos son las siguientes:

- Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multihilo.
- Soporta gran cantidad de tipos de datos para las columnas.
- Dispone de API's en gran cantidad de lenguajes (C, C++, Java, PHP, etc).
- Gran portabilidad entre sistemas.
- Soporta hasta 32 índices por tabla.
- Gestión de usuarios y contraseñas, manteniendo un muy buen nivel de seguridad en los datos.
- Múltiples motores de almacenamiento (MyISAM, Merge, InnoDB, BDB, Memory/heap, MySQL Cluster, Federated, Archive, CSV, Blackhole y Example en 5.x), permitiendo al usuario escoger la que sea más adecuada para cada tabla de la base de datos.
- Agrupación de transacciones, reuniendo múltiples transacciones de varias conexiones para incrementar el número de transacciones por segundo.

Ventajas

- Mayor rendimiento. Mayor velocidad tanto al conectar con el servidor como al servir selects y demás consultas.
- Mejores utilidades de administración (backup, recuperación de errores, etc.).
- Aunque se cuelgue, no suele perder información ni corromper los datos.
- No hay límites en el tamaño de los registros.
- Mejor control de acceso, en el sentido de qué usuarios tienen acceso a qué tablas y con qué permisos.
- Se comporta mejor a la hora de modificar o añadir campos a una tabla "en caliente".

1.4.3 Metodologías para el desarrollo de sistemas informáticos

La metodología utilizada para el análisis, implementación y documentación del sistema que se aspira desarrollar fue RUP (Rational Unified Process, Proceso Unificado de Desarrollo de Software). El mismo se utilizó por sus particularidades, facilidades que aporta a todo el proceso, teniendo en cuenta que posee una buena herramienta que soporta cada uno de los procesos que se necesitan. Dicha herramienta es: Rational Rose Enterprise Edition.

Es una herramienta que soporta el modelado basado en UML, se aplica para la etapa de análisis y diseño de sistemas.

Rose es una herramienta con plataforma independiente que ayuda a la comunicación entre los miembros de equipos, a monitorear el tiempo de desarrollo y a entender el entorno de los sistemas.

Una de las grandes ventajas de Rose es que utiliza la notación estándar en la arquitectura de software (UML), la cual permite a los arquitectos de software y desarrolladores visualizar el sistema completo utilizando un lenguaje común, además los diseñadores pueden modelar sus componentes e interfaces en forma individual y luego unirlos con otros componentes del proyecto.

Metodología RUP

El Proceso Unificado de Desarrollo de Software (RUP) brinda un proceso integrado que utiliza el estándar de notación UML (Lenguaje Unificado de Modelado) para desarrollar un producto de software orientado a objetos, el UML constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. No posee

pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada proceso. Entre sus principales características se encuentran:

- Forma disciplinada de asignar tareas y responsabilidades (quién hace qué, cuándo y cómo).
- Pretende implementar las mejores prácticas en Ingeniería de Software.
- Desarrollo iterativo.
- Administración de requisitos.
- Uso de arquitectura basada en componentes.
- Control de cambios.
- Modelado visual del software.
- Verificación de la calidad del software.

Se caracteriza por su enfoque iterativo e incremental, la metodología parte del supuesto de que se trabajará en iteraciones cortas en tiempo y con metas muy claras. Cada iteración tiene entregables claros y en la medida de lo posible, el sistema debe ser funcional desde las primeras iteraciones de desarrollo. Si una funcionalidad no puede terminarse en una iteración, se terminará la iteración sin ella, es más importante terminar la iteración con un sistema funcional y sin errores o detalles incompletos que un sistema que no puede ser evaluado. Incluye artefactos (que son los productos tangibles del proceso como por ejemplo, el modelo de casos de uso, el código fuente, etc.) y roles (papel que desempeña una persona en un determinado momento, la misma puede desempeñar distintos roles a lo largo del proceso).

Lenguaje de modelación UML

Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modeling Language2) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; aún cuando todavía no es un estándar oficial, está respaldado por el OMG (Object Management Group). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. UML ofrece un estándar para describir un "plano" del sistema

(modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables.

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

Es importante recalcar que UML no es una guía para realizar el análisis y diseño orientado a objetos, es decir, no es un proceso. UML es un lenguaje que permite la modelación de sistemas con tecnología orientada a objetos (sistemas de software, sistemas de hardware, y organizaciones del mundo real).

El modelo gráfico de UML tiene un vocabulario en el que se identifican:

1. Elementos: Abstracciones que constituyen los bloques básicos de construcción.
    - Estructurales: Clase, Colaboración, Caso de uso, Clase activa, Componente, Nodo.
    - Comportamiento: Interacción, Máquina.
    - Agrupamiento: Paquete.
    - Anotación.
2. Relaciones: Ligan los elementos.
    - Dependencia.
    - Asociación.
    - Generalización/Especialización.
    - Realización.

3. Diagramas: Es la representación gráfica de un conjunto de elementos.

Visualizan un sistema desde diferentes perspectivas.

- Diagramas de estructura estática:
  - Diagrama de clases: Conjunto de clases, interfaces y colaboraciones; así como sus colaboraciones.
  - Diagrama de objetos: Conjunto de objetos y sus relaciones.
- Diagramas de comportamiento:
  - Diagramas de interacción (secuencia y colaboración): Objetos y sus relaciones, incluyendo los mensajes que pueden ser enviados entre ellos.
  - Diagrama de estados: Muestra una máquina de estado que consta de estados, transiciones, eventos y actividades.
  - Diagrama de actividad: Es un tipo especial de diagrama de estados que muestra el flujo de actividades dentro de un sistema.
- Diagrama de casos de uso: Conjunto de casos de uso y actores y sus relaciones.
- Diagramas de implementación:
  - Diagrama de componentes: Organización y las dependencias entre un conjunto de componentes.
  - Diagrama de despliegue: Configuración de nodos de procesamiento en tiempo de ejecución y los componentes que residen en ellos.

1.5 Fundamentación de la selección del lenguaje a utilizar

¿Por qué Java como lenguaje de programación?

Para el desarrollo de nuestra aplicación hemos seleccionado Java como lenguaje de programación por contar con las siguientes características:

Java reduce en un 50% los errores más comunes de programación con lenguajes como C y C++ al eliminar muchas de las características de éstos, entre las que destacan:

- Aritmética de punteros.

- No existen referencias.
- Registros (struct).
- Definición de tipos (typedef).
- Macros (#define).
- Necesidad de liberar memoria (free).
- Aunque, en realidad, lo que hace es eliminar las palabras reservadas (struct, typedef), ya que las clases son algo parecido.

Con respecto a delphi tiene ventajas en cuanto a seguridad ya que al tener la característica de ser interpretado el código Java pasa a través de un verificador de byte-codes que comprueba el formato de los fragmentos de código y aplica un probador de teoremas para detectar fragmentos de código ilegal -código que falsea punteros, viola derechos de acceso sobre objetos o intenta cambiar el tipo o clase de un objeto. El Cargador de Clases, separando el espacio de nombres del sistema de ficheros local, del de los recursos procedentes de la red. Esto limita cualquier aplicación del tipo Caballo de Troya, ya que las clases se buscan primero entre las locales y luego entre las procedentes del exterior.

1.6 Fundamentación del SGBD a utilizar

¿Por qué MySQL como gestor de base de datos?

Luego de analizar las características y facilidades de los SGBD presentados escogimos MySQL como SGBD por las siguientes razones:

- Actualmente MySQL ha sido comprada por SUN lo que en un futuro lo convertirá en un SGBD más robusto debido a que esta prestigiosa empresa se encarga de darle soporte técnico, además de que sigue siendo de adquisición gratuita.
- El MySQL además de ser multiplataforma, es muy veloz.
- La sintaxis del MySQL es legible.
- Familiarización con el Gestor MySQL.

1.7 Conclusiones del capítulo

Al concluir este capítulo se ha podido apreciar todo lo referente a la base teórica que fundamenta esta investigación, dando ha conocer al lector los principales conceptos que se manejan, las diferentes tecnologías para el desarrollo de aplicaciones, los sistemas automatizados que existen en el

mundo vinculados con el campo donde se proyecta el objeto de estudio así como la fundamentación del leguaje de programación a utilizar, el Sistema Gestos de Base de Datos y la Metodología que se eligió para darle solución al problema planteado.

Capitulo 2 Modelo de Dominio

2.1 Introducción

Al realizar el análisis de los procesos expuestos en el capítulo anterior nos percatamos que nuestro negocio posee un bajo nivel de estructuración. Para tales casos la metodología que se escogió propone realizar un modelo de dominio; que no es más que una representación visual de las clases conceptuales u objetos del mundo real en un dominio de interés, por lo que permite mostrar al usuario los principales conceptos que se manejan en el dominio del sistema en desarrollo. Este modelo contribuirá a identificar algunas clases que se utilizarán en el sistema. Esto incluirá la definición de entidades y los conceptos principales. Las cuales se definirán a partir de su símbolo, su intención, y su extensión.

De modo que se especificará lo siguiente:

1. Definición de entidades y los conceptos principales. Las cuales se definirán a partir de:
   - Símbolo: Palabra que representa el concepto.
   - Intensión: La definición del concepto.
   - Extensión: El conjunto de ejemplos a que se aplica el concepto.
2. Diagrama del Modelo del Dominio.
3. Representación de los requerimientos funcionales y no funcionales del sistema.

2.2 Definición de las entidades y los conceptos principales.

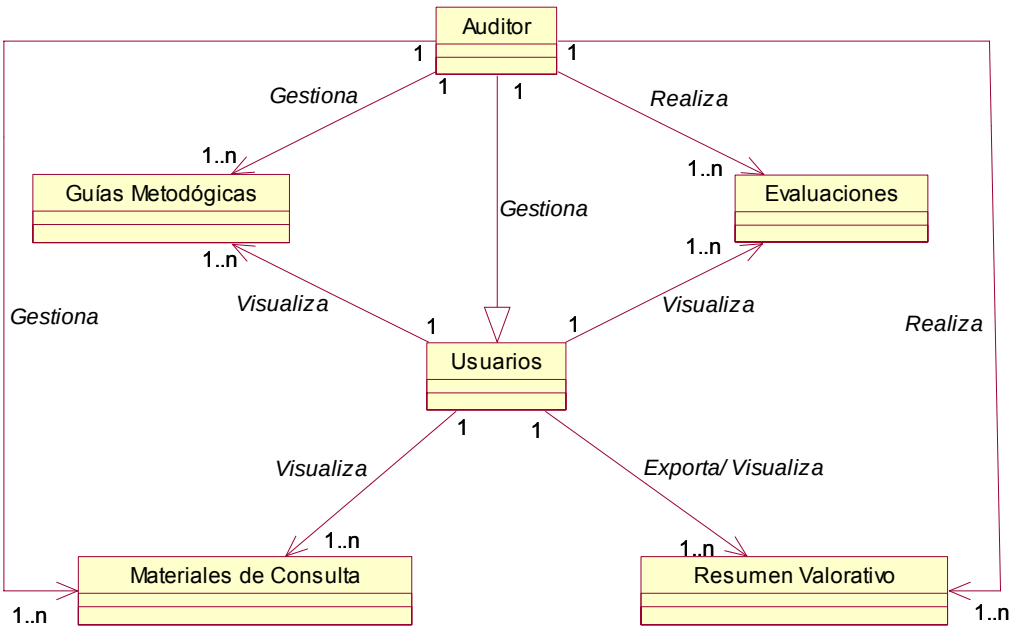
Tabla 1 Entidades y conceptos del dominio.

Símbolo	Intensión	Extensión
Auditor	Se le denomina Auditor a todos aquellos usuarios que se le asignan ciertos privilegios para operar sobre el sistema (puede ser administrador del sistema).	Personas

Usuarios	Se le denomina Usuarios a todos aquellos usuarios que pueden ejecutar las opciones no privilegiadas que brinda el sistema (invitados).	Personas
Guías Metodológicas	Se refiere a las guías por las cuales se realiza el diagnóstico del Sistema de Control Interno.	Cuestionarios
Materiales de consulta	Se refiere a documentos que abarcan temas referentes al Sistema de Control Interno	Documentos
Evaluaciones	Se refiere al los diagnósticos realizados a través de las Guías metodológicas.	_

2.3 Representación del Modelo de dominio.

Figura 1 Diagrama de Modelo de Dominio.



2.4 Requerimientos Funcionales y no Funcionales del Sistema.

2.4.1 Requerimientos Funcionales.

Los requisitos funcionales indican el comportamiento del sistema. Posteriormente estos requisitos son modelados a través del diagrama de casos de uso del sistema.

1. Administrar usuarios.
   1.1 Insertar usuarios.
   1.2 Eliminar usuarios.
   1.3 Modificar Usuario.
2. Administrar Guías Metodológicas (GM).
   2.1 Insertar GM.
   2.2 Eliminar GM.
   2.3 Modificar GM.
   2.4 Evaluar GM.
3. Administrar materiales de consulta.
   3.1 Ingresar materiales de consulta.
   3.2 Eliminar materiales de consulta.
4. Mostrar listado de Materiales de consulta.
5. Mostrar listado de Usuarios.
6. Mostrar listado de Guías.
7. Mostrar listado de Evaluaciones.
8. Realizar resumen valorativo.
9. Exportar resumen.

2.4.2 Requerimientos 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 o confiable.

Apariencia o interfaz externa

La interfaz debe ser sencilla y amigable puesto que los usuarios pueden o no ser personas expertas. La respuesta del sistema ha de ser rápida.

Usabilidad

Para tener acceso a modificar la información se requiere ser usuario Administrador o tener asignado privilegios para ello. El sistema garantizará un acceso fácil y rápido a los usuarios finales y demás usuarios.

Rendimiento

Esta aplicación debe tener un rendimiento óptimo. Su eficiencia estará determinada en cierta medida por la velocidad de las consultas en la base de datos. Los tiempos de respuestas deben ser generalmente rápidos al igual que la velocidad de procesamiento de la información.

Portabilidad

El sistema debe ser multiplataforma, todo depende de la portabilidad que alcance la maquina virtual de java.

Ayuda y documentación

El sistema contará con un manual de usuario en versión digital, la cual guía al usuario en el funcionamiento del sistema.

Software

- Se necesita tener instalada la Máquina Virtual de Java, versión JRE 1.6.0 o superior.
- Se requiere de un servidor de Base de Datos MySQL 5.0.20 o superior.

2.5 Conclusiones del capítulo

En este capítulo se realizó una descripción del funcionamiento del sistema representado por un modelo de dominio, el cual permitió mostrar los roles y conceptos principales que posteriormente permitirán definir las clases conceptuales que se utilizan en la implementación del sistema, las funciones con la cual el sistema debe cumplir así como sus propiedades o características que lo hacen atractivo, usable, rápido o confiable, obteniéndose a partir del análisis de los procesos del dominio, la cual ayudará para la elaboración de los demás diagramas.

Capitulo 3 Diseño e Implementación Del Sistema

3.1 Introducción

En este capítulo se ofrecen los flujos de trabajo correspondiente a la etapa de diseño e implementación del sistema propuesto, donde se definirán los Actores del sistema, se realizará una representación del diagrama de Casos de Uso con los que interactúan todos los usuarios, así como los Diagrama de Clases, de Secuencia, de Despliegue y de Componentes.

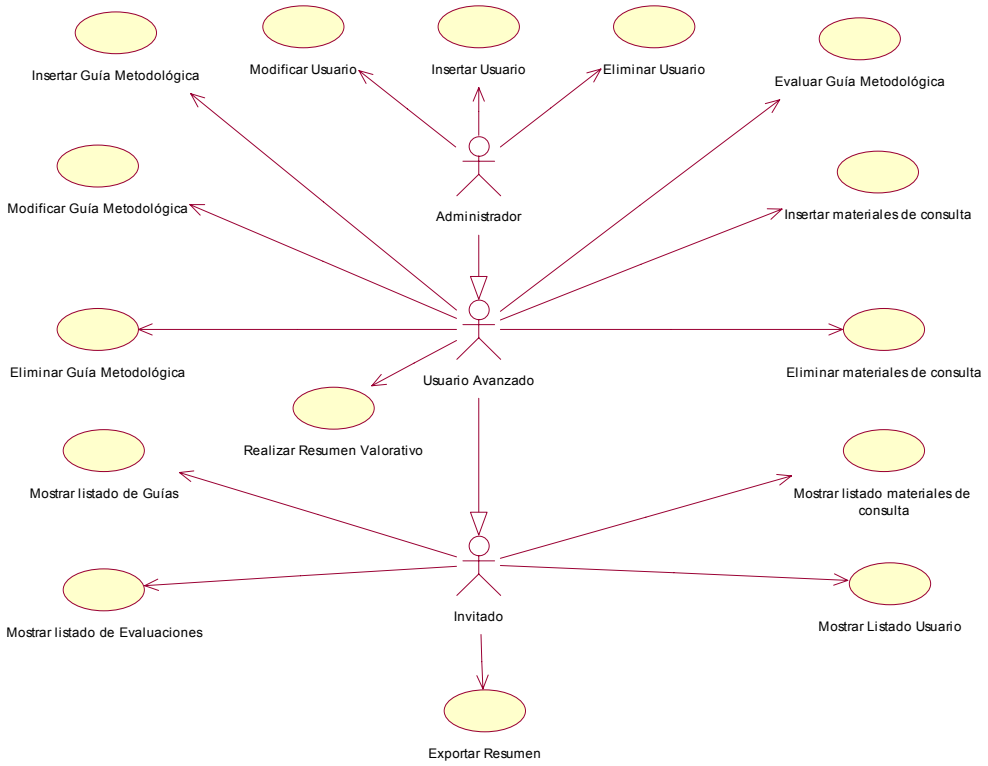
3.2 Actores del sistema

Tabla 2 Definición de los actores del sistema.

Nombre del Actor	Descripción
Administrador	Se le denomina Administrador a todos aquellos usuarios que tienen control total sobre del sistema.
Usuario Avanzado	Se le denomina Usuario Avanzado a todos aquellos usuarios que el Administrador le asigna algunos privilegios para operar sobre el sistema.
Invitado	Se le denomina Invitado a todos aquellos usuarios que pueden ejecutar las opciones no privilegiadas que brinda el sistema.

3.3 Diagrama de casos de uso del sistema.

Figura 2 Diagrama de casos de usos del sistema.



3.4 Descripción de los Casos de Uso.

Tabla 3 Descripción del caso de uso "Insertar Usuarios".

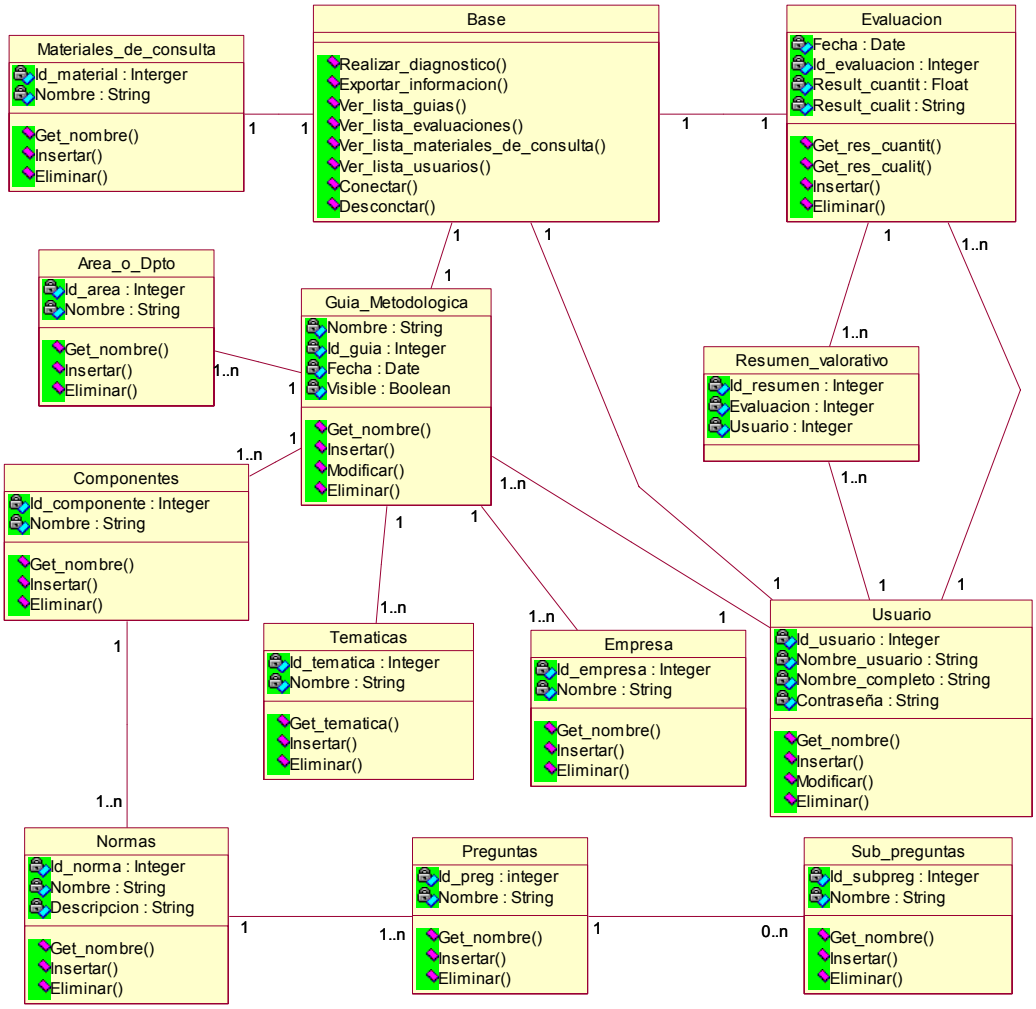
Nombre de caso de uso	Insertar Usuarios
Actor(es)	Administrador(inicia)
Resumen	Este caso de uso lo inicia un administrador cuando decide crear un nuevo usuario.
Referencia	R1.1
Precondiciones	_
Poscondiciones	Queda insertado el usuario listo para ser utilizado.
Requisitos especiales	Se debe validar que sean introducidos todos los datos correspondientes.

En el Anexo 1 se muestra la descripción de los demás casos de uso del sistema.

3.5 Diagrama de clases del diseño

3.5.1 Diagrama de clases general.

Figura 3 Diagrama de clases del diseño.



3.5.2 Descripción general de las clases

Tabla 4 Descripción de la clase "Empresa".

Nombre: Empresa	
Tipo de clase: Entidad	
Atributo	**Tipo**
Id_empresa	Integer
Nombre	String
Para cada responsabilidad	
Nombre	**Descripción**
Get_nombre()	Devuelve el nombre de la empresa
Insertar(int id, String emp)	Se pasa por parámetro el id y el nombre de la empresa y se inserta en la base de datos
Eliminar(String emp)	Se pasa por parámetro el nombre de la empresa que se desea eliminar

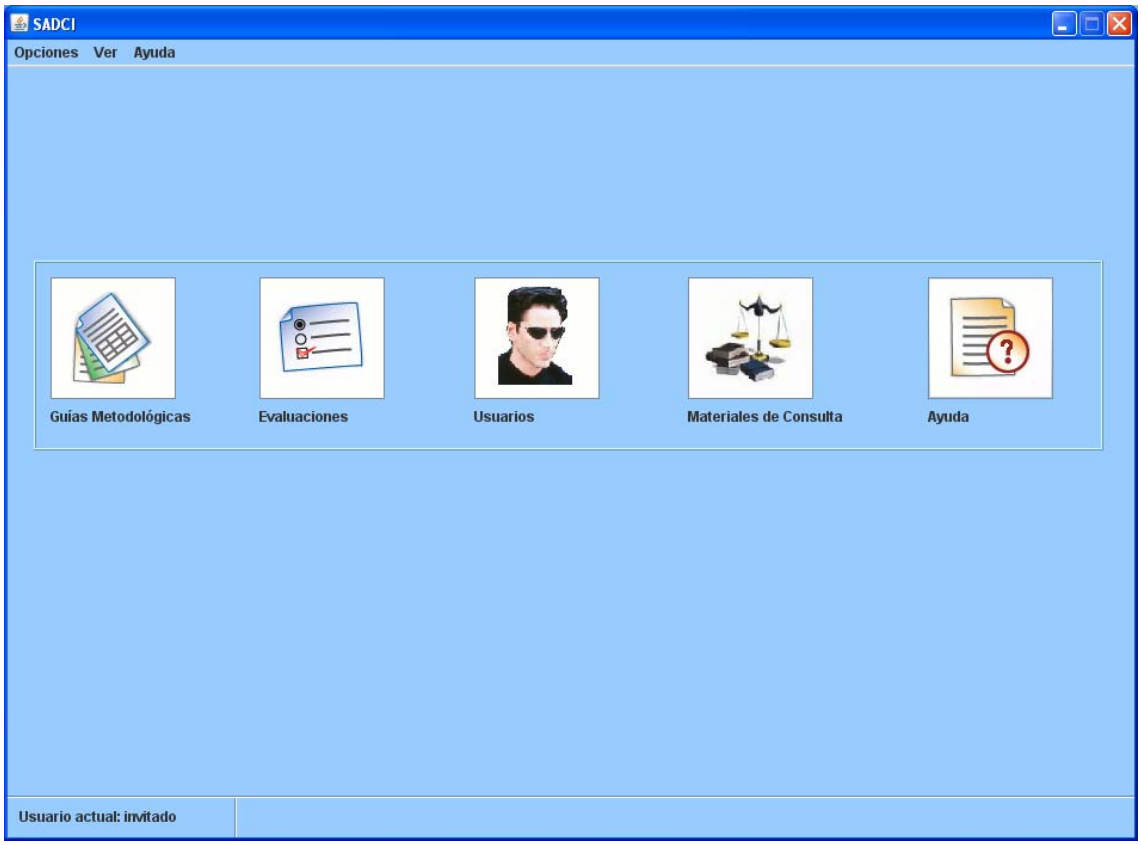
En el Anexo 2 se muestran las demás descripciones de las clases del diseño.

3.6 Principios de diseño

En la construcción de la interfaz de usuario del sistema que se propone se tuvo en cuenta los estándares internacionales definidos para el diseño de interfaces, con el fin de hacer una interfaz sencilla y fácil de utilizar.

3.6.1 Interfaz de usuario "SADCI"

Figura 4 Interfaz de usuario "SADCI".



En el Anexo 3 se muestran mas detalles de la interfaz de usuario.

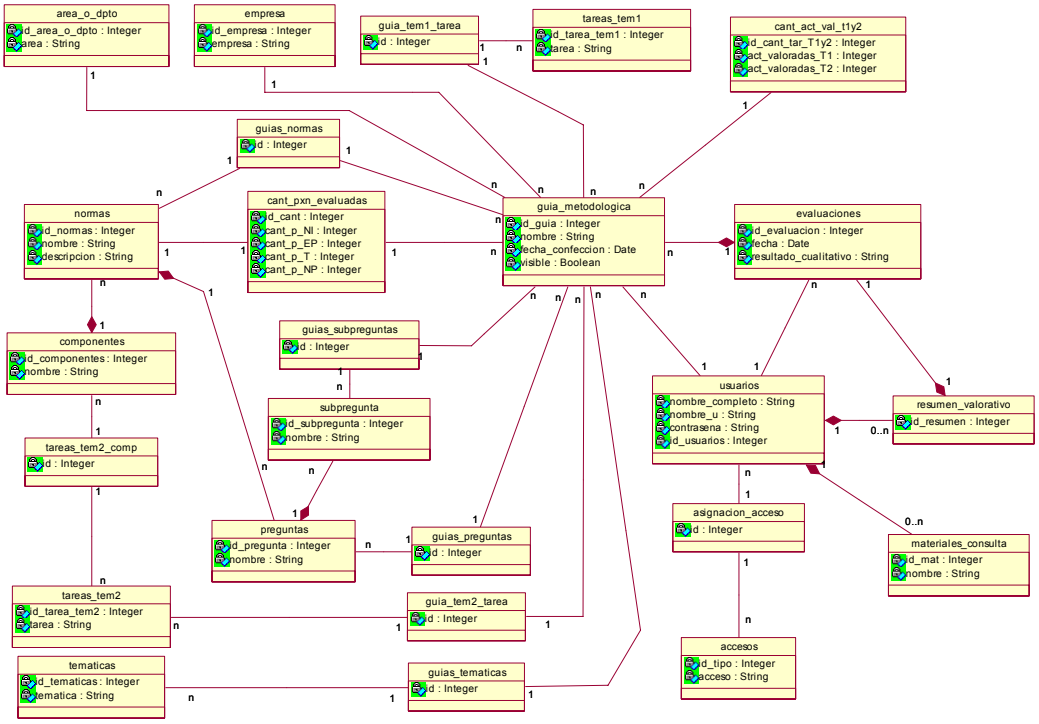
3.6.2 Ayuda

Como ayuda para los usuarios, el sistema solo tendrá el manual de usuario, que contiene toda la información necesaria para usarlo correctamente

3.7 Diseño de la Base de Datos

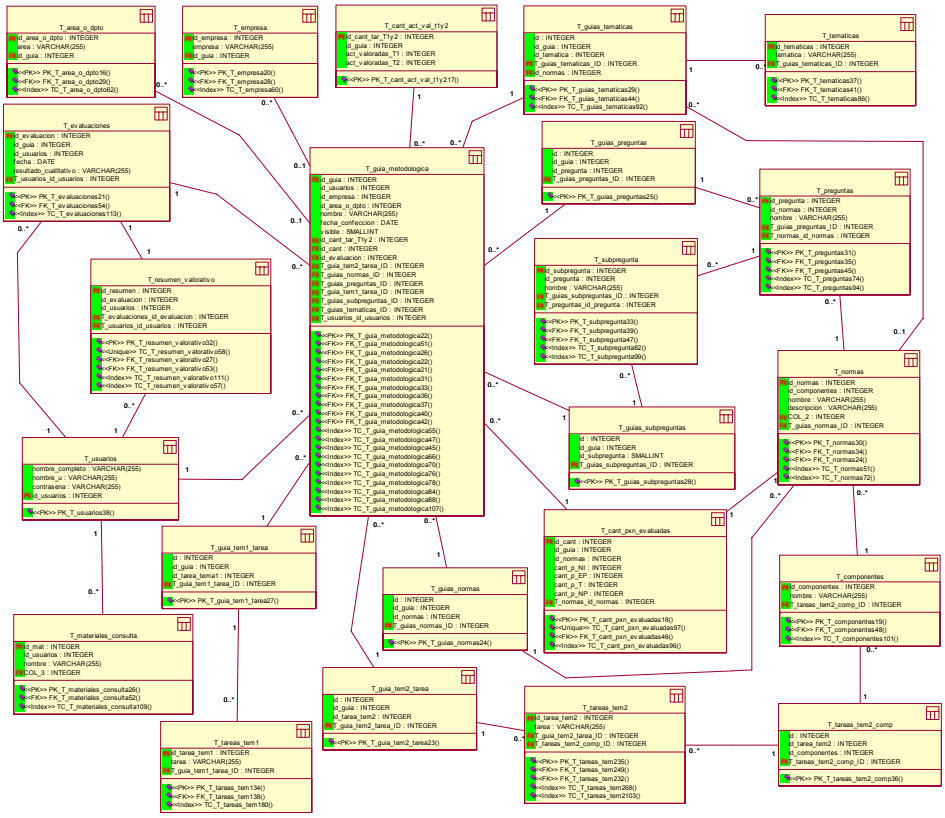
3.7.1 Modelo Lógico de Datos

Figura 5 Diagrama de Clases Persistente.



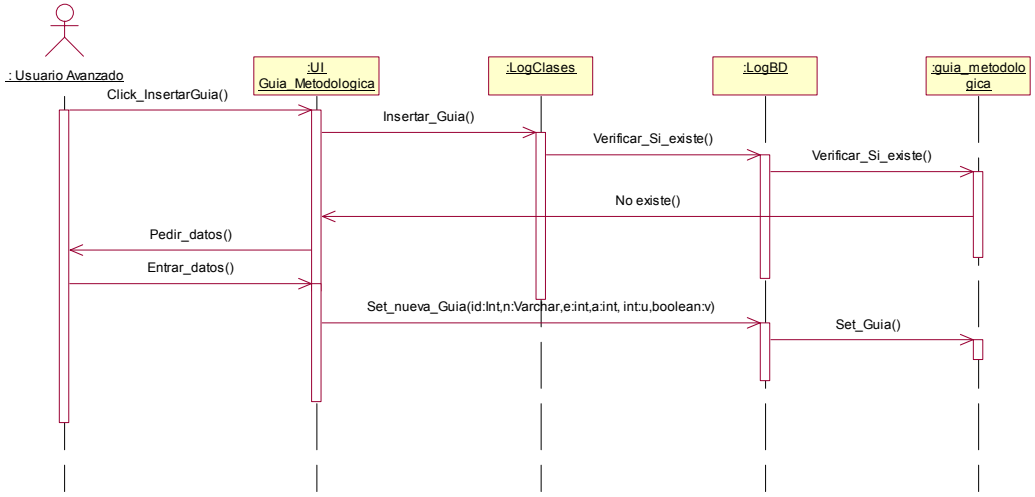
3.7.2 Modelo Físico de Datos

Figura 6 Modelo Físico de Datos.



3.8 Diagramas de Secuencia

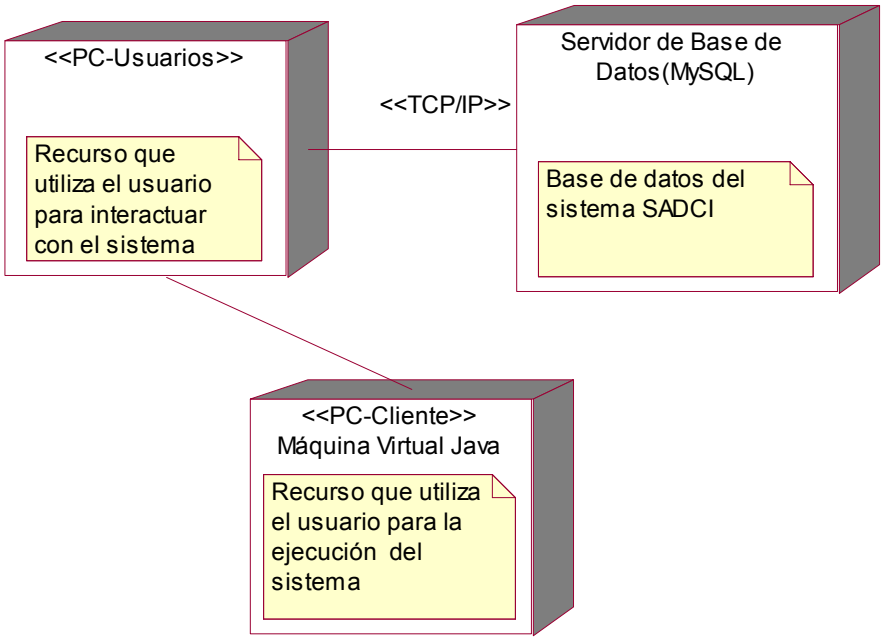
Figura 7 Diagrama de Secuencia del Caso de uso Insertar Guia Metodológica.



En el Anexo 4 se Muestran los restantes diagramas de secuencia de los demás casos de uso.

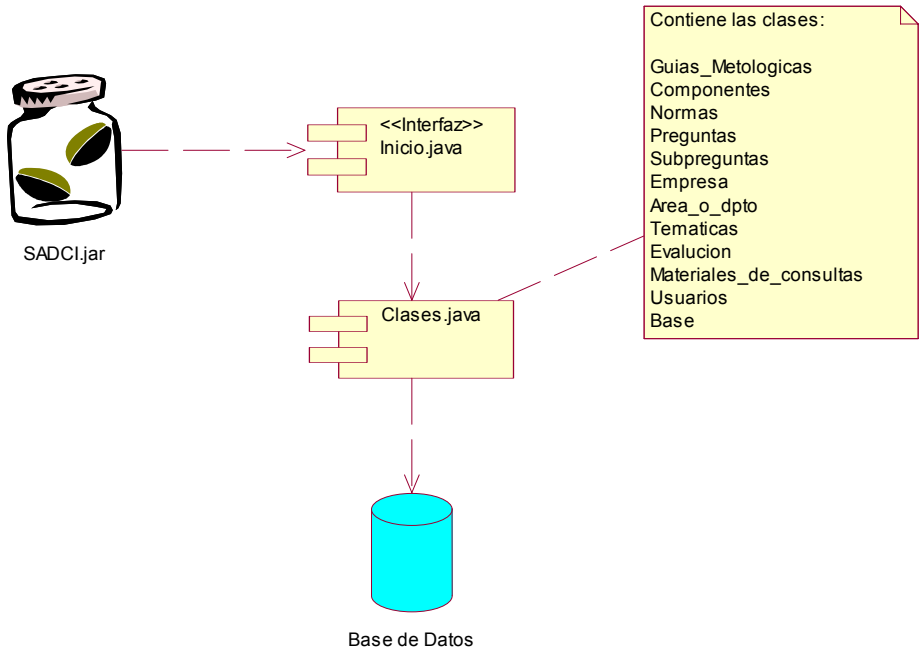
3.9 Diagrama de Despliegue

Figura 8 Diagrama de Despliegue.



3.10 Diagrama de Componentes

Figura 9 Diagrama de Componentes.



3.11 Conclusiones del capítulo

En este capítulo se ofrecieron aspectos que dentro de la metodología de ingeniería de software que se empleó, nos facilitan un correcto entendimiento de la solución propuesta apartir del análisis. Se definieron los actores que intervienen en el sistema, se modeló el Diagrama de Casos de Uso el cual representa las funcionalidades con la que debe cumplir el sistema, se diseñó el Diagrama de Clase para representar la estructura interna del sistema, los Diagramas de Secuencia con los que se logra delinear el orden de los procesos para la ejecución de cada caso de uso, se representó la arquitectura del sistema mediante el Modelo de Despliegue así como otros artefactos que incluye dicha metodología.

Capítulo 4 Estudio de Factibilidad

4.1 Introducción

En este capítulo se expone un estudio de factibilidad basado en los aspectos operativos, técnicos y económicos, utilizando para la determinación del costo las normas establecidas en el modelo COCOMO II, ofreciendo el costo total del software.

4.2 Estudio de la factibilidad

Factibilidad se refiere a la disponibilidad de los recursos necesarios para llevar a cabo los objetivos o metas señalados, apoyándose en tres aspectos básicos: operativo, técnico y económico. El éxito de un proyecto está determinado por el grado de factibilidad que se presente en cada uno de los tres aspectos anteriores. Para esto se realiza un estudio para recopilar datos relevantes sobre el desarrollo de un proyecto y en base a ello tomar la mejor decisión, si procede su estudio, desarrollo o implementación. La investigación de factibilidad es un proyecto que consiste en descubrir cuales son los objetivos de la organización, luego determinar si el proyecto es útil para que la empresa o institución logre sus objetivos.

4.3 Planificación por puntos de función

Entradas externas:

Tabla 5 Entradas externas.

Nombre de la entrada externa	Cantidad de Ficheros	Cantidad de Elementos de datos	Clasificación (Baja, Media y Alta)
Insertar usuario	1	4	Bajo
Modificar Usuario	1	3	Bajo
Cambiar contraseña	1	2	Bajo
Eliminar usuario	12	1	Medio
Insertar Guía Metodológica	15	16+	Alta
Modificar Guía Metodológica	13	16+	Alta
Eliminar Guía Metodológica	10	1	Medio
Ingresar material de consulta	1	3	Bajo
Eliminar material de consulta	1	1	Bajo

Realizar resumen valorativo	12	1	Medio
Exportar Resumen	4	5	Alto
Evaluar Guía Metodológica	3	1	Bajo
Cantidad de ficheros			
Bajo	Medio		Alta
6	3		3

Salidas externas:

Tabla 6 Salidas externas

Nombre de la salida externa	**Cantidad de ficheros**	**Cantidad de Elementos de datos**	**Clasificación (Bajo, Media y Alto)**
Mostrar Listado de Materiales de consulta	2	2	Bajo
Mostrar Listado de Guías Metodológicas	4	5	Medio
Mostrar Listado de Evaluaciones	3	4	Bajo
Mostrar Listado de Usuarios	1	2	Bajo
Cantidad de Ficheros			
Bajo	Medio		Alto
3	1		0

Consultas Externas:

Tabla 7 Consultas externas

Nombre de la Consulta Externa	Cantidad de ficheros	Cantidad de Elementos de datos	Clasificación (Bajo, Media y Alto)
Visualizar Listado de Usuarios	1	2	Bajo
Evaluar Guía Metodológica	14	1	Medio
Visualizar Listado de Materiales de Consulta	1	1	Bajo
Visualizar Listado de Guías Metodológicas	1	5	Bajo
Visualizar Listado de Evaluaciones	3	4	Bajo
Realizar resumen valorativo	12	1	Medio
Cantidad de Ficheros			
Bajo	Medio		Alto
4	2		0

Ficheros Lógicos Internos:

Tabla 8 Ficheros lógicos Internos

Nombre del fichero interno	Cantidad de records	Cantidad de Elementos de datos	Clasificación (Bajo, Media y Alto)
guías metodológicas	6+	7	Medio
usuarios	6+	4	Medio
evaluaciones	6+	5	Medio
empresa	6+	2	Medio
area_o_dpto	6+	2	Medio
cant_act_val_t1y2	6+	4	Medio
cant_pxn_evaluadas	6+	7	Medio
componentes	5	2	Bajo
normas	6+	4	Medio
preguntas	6+	3	Medio
subpreguntas	6+	3	Medio
tematicas	3	2	Bajo
materiales_consulta	6+	3	Medio
resumen_valorativo	6+	3	Medio
guia_tem1_tarea	6+	3	Medio
guia_tem2_tarea	6+	3	Medio
guias_normas	6+	3	Medio

guias_preguntas	6+	3	Medio
guias_subpreguntas	6+	3	Medio
guias_tematicas	6+	3	Medio
tareas_tem1	6+	2	Medio
tareas_tem2	6+	2	Medio
tareas_tem2_comp	6+	3	Medio
accesos	6+	2	Medio
asignación_accesos	6+	3	Medio
usuarios_logueados	6+	2	Medio
ps_val	6+	2	Medio
Cantidad de records			
Bajo	Medio		Alta
2	25		0

Puntos de Función desajustados:

Tabla 9 Puntos de Función desajustados

Elementos	**Bajas**		**Medios**		**Altas**		**Subtotal de puntos de función**
	No	X Peso	No	X Peso	No	X Peso	
Ficheros lógicos internos	2	7	25	10	0	15	264
Entradas externas	5	3	3	4	3	6	45
Salidas externas	3	4	1	5	0	7	17
Consultas externas	4	3	2	4	0	6	20
Total	14		31		3		346

4.4 Estimación de la cantidad de instrucciones fuentes

Para el cálculo de las instrucciones fuentes (SLOC) se utilizó la fórmula siguiente:

SLOC = UFP * ratio.

Donde:

UFP: total de puntos de función desajustados.

- ratio: constante para las SLOC de cada lenguaje de programación en este caso tiene un valor para Java de 53.

Luego

SLOC = 346*53 = 18338

SLOC ≈ líneas de código fuente

KSLOC= (Miles de líneas de código)

KSLOC = 18,338

4.5 Aplicación de las fórmulas de Bohem

Después de realizado el cálculo de la cantidad de las instrucciones fuentes, este valor obtenido es utilizado para el cálculo del esfuerzo por medio dela fórmula de Bohem.

4.5.1 Cálculo del esfuerzo (PM)

$$PM_{NS} = A \times Size^{E} \times \prod_{i=1}^{n} EM_i$$

Donde: $E = B + 0.01 \times \sum_{j=1}^{5} SF_j$

Además se considera:

Size: Tamaño estimado (KSLOC).

SF_j: Factores de escala.

EM_i: Multiplicadores de esfuerzo.

A = 2.94

B = 0.91

Los valores de cada factor de escala (SF_j) y de cada multiplicador de esfuerzo (EM_i) se muestran en las tablas siguientes, los cuales son necesarios para obtener el resultado del cálculo del esfuerzo.

Factores de escala

Tabla 10 Factores de escala.

Factor de Escala	Valor	Justificación
PREC (Precedencia)	3.72	Se considera con aspectos novedosos para el desarrollador
FLEX (Flexibilidad)	2.03	Existe un grado de acuerdo general con los requerimientos pre-establecidos y con las interfaces externas
RESL (Riesgos)	2.83	No se consideran muchos de los riesgos críticos y se establecieron hitos para resolverlos
TEAM (Cohesión del equipo)	0.0	No se requiere más de un desarrollador de la aplicación
PMAT	7.80	Se estima un nivel de madurez alto para el

(Madurez de las Capacidades)		desarrollo del sistema.

Multiplicadores de esfuerzo

Tabla 11 Multiplicadores de Esfuerzo.

Multiplicador	Valor	Justificación
RELI	1.26	Posee una fiabilidad alta exigida al software.
DATA	1	El tamaño de la Base de Datos es normal.
CPLX	1.00	El producto tiene una moderada complejidad. La base de datos que se utiliza tiene un volumen mediano de información por lo que se considera de tamaño moderado.
RUSE	1.07	En la implementación del sistema existe una alta reusabilidad de códigos.
DOCU	1.00	Se considera que la documentación es normal por la cantidad promedio de artefactos que deben ser documentados.
TIME	1.11	Existe una alta exigencias sobre capacidad de ejecución.
STOR	1.05	Se requiere de elevadas exigencias sobre almacenamiento del sistema.
PVOL	0.87	La plataforma es poco volátil producto a que los cambios que ocurren en la misma no son periódicos.
ACAP	0.85	Se considera que la capacidad del analista es elevada.
PCAP	1.00	La capacidad del programador es normal.
PCON	0.81	La volatilidad del personal es elevada.
APEX	1.10	Existe una experiencia previa limitada en área de aplicación.
PLEX	1.09	Existe una experiencia previa limitada en la plataforma.
LTEX	1.09	No existe una experiencia previa avanzada con el lenguaje y herramientas de desarrollo.
TOOL	0.78	Son usadas diferentes herramientas de software por lo tanto es elevado el uso de las mismas.
SITE	1.22	No existe desarrollo de la aplicación en localidades distribuidas.
SCED	1.00	Existen grandes exigencias sobre el calendario.

De los resultados anteriores se obtiene que:

Factores de escala:

$\sum_{j=1}^{5} \mathrm{SF_j}$ = 16,38.

Multiplicadores de esfuerzo:

$\prod_{i=1}^{n} EM_i$ = 1,17.

Por tanto, a partir de ellos se obtiene finalmente el cálculo del esfuerzo:

$$PM_{NS} = A \times Size^{E} \times \prod_{i=1}^{n} EM_i$$

$$E = B + 0.01 \times \sum_{j=1}^{5} SF_j$$

E = 0, 91 + 0,01 * 16, 38 = 1, 0738

$$PM_{NS} = 2{,}94 * 18{,}338^{1{,}0738} * 1{,}17$$

PM_{NS} = 78,18 hombres-mes

El esfuerzo que realize una persona para desarrollar el software en un mes es de aproximadamente 78,18 personas por mes.

4.5.2 Cálculo del tiempo de desarrollo (TDEV)

$$TDEV_{NS} = C \times (PM_{NS})^{F}$$

Donde $F = D + 0.2 \times 0.01 \times \sum_{j=1}^{5} SF_j$

$$= D + 0.2 * (E - B)$$

$$F = 0{,}28 + 0{,}2 * (1{,}0738 - 0{,}91) = 0{,}31276$$

Además se considera:

C = 3.67

D = 0.28

$$TDEV_{NS} = 3{,}67 * (78{,}18)^{0{,}31276}$$

$TDEV_{NS} =$ TDEV$_{NS}$ = **14,35** meses.

El tiempo necesario para desarrollar el producto es de 14,35 meses.

4.5.3 Cantidad de hombres

Teniendo en cuenta el tiempo de desarrollo se calcula la cantidad de personas (CH) necesarios para desarrollar el software, se obtiene la Tabla 4.9.

Tabla 12 Constantes y fórmulas para el cálculo de hombres.

Siglas	Indicador	Valor o Fórmula
CH	Cantidad de hombres por mes	PM/TDES
PM	Esfuerzo	78,18 hombre-mes
TDEV	Tiempo de desarrollo	14,35 meses

$$CH = \frac{PM}{TDEV} = \frac{78{,}18}{14{,}35} = 5{,}44 \text{ personas}$$

Son necesarias 6 personas para realizar el software en 14,35 meses. Como en realidad trabaja una sola persona se reajustan los cálculos para este valor:

$$CH^{*} = 1 \text{ persona}$$

$$TDEV_{REAL} = \frac{PM}{CH^{*}} = \frac{78{,}18}{1} = 78{,}18 \text{ meses}$$

Por tanto trabajando una persona se desarrolla el software en 78,18 meses.

4.6 Determinar el Costo del Software

El costo del software depende del salario promedio de las personas que lo desarrollan y del esfuerzo que ellas realizan para la ejecución del mismo y se calcula a través de la fórmula representada en la tabla.

Tabla 13 Constantes y fórmulas para el cálculo de Costo.

Siglas	Indicador	Valor o Fórmula
C	Costo del proyecto	CHM *PM
CHM	Costo de hombres por mes	CH* * SP
SP	Salario basico de un Ingeniero	$ 315.00
PM	Esfuerzo	78,18 hombre-mes

El software cuesta $ 24626,7.

Tabla 14 Resultados de las estimaciones de Esfuerzo, Tiempo de Desarrollo, Cantidad de hombres y Costo del proyecto.

Cálculo de:	Valor	Justificación
Esfuerzo	78,18 hombres-mes	Cantidad de tiempo que una persona invierte trabajando en el desarrollo de un proyecto
Tiempo de desarrollo	78,18 meses	Cantidad de meses para terminar el proyecto.
Cantidad de personas	1	Cantidad de personas necesarias para terminar el proyecto en 78,18 meses.
Costo	$ 24626,7	Cantidad de dinero que cuesta el proyecto después de terminado.
Salario medio	$ 315.00	Salario basico de un ingeniero

4.7 Beneficios Tangibles e Intangibles

Beneficios Intangibles

Esta aplicación de escritorio se convertirá en una herramienta de uso cotidiano logrando un alto grado de funcionalidad, usabilidad y seguridad; además de ser lo suficientemente flexible como para permitir modificaciones y actualizaciones de la información que en ella se procesa. Se va a mejorar la labor de los directivos en las entidades y áreas administrativas, asi como los auditores podrán desempeñar su trabajo con mayor facilidad a la hora de la evaluación del SCI implementado.

Beneficios Tangibles

A través del uso de este sistema se pueden obtener los siguientes beneficios de forma inmediata:

- Optimiza y mejora el trabajo de auditores, especialistas y directivos interesados en el tema
- La utilización de un gestor de base de datos muy potente que garantiza velocidad en el procesamiento, seguridad y protección de los datos.
- El producto implica además un ahorro del tiempo que se invierte en los procesos de diagnóstico del SCI, de manera que el mayor tiempo posible y los principales esfuerzos en el área estén encaminados al cumplimiento de los objetivos planteados.

4.8 Análisis de costos y beneficios

El costo que significa la implementación del sistema propuesto es un poco elevado sin embargo la tecnología utilizada para su desarrollo es totalmente libre, por tanto no es necesario incurrir en gastos en el pago de licencias de uso. También contribuye al ahorro de ciertos recursos. Por ello se plantea que es factible el desarrollo del mismo.

El sistema es portable por lo que un cambio de plataforma para la puesta en marcha del mismo es viable y factible, y no hay que incurrir en muchos cambios.

Analizando el costo del proyecto, los beneficios que reporta, expuesto en el epígrafe anterior, se puede concluir que su implementación es factible.

4.9 Conclusiones del capítulo

El estudio de factibilidad llevado a cabo en este capítulo ha permitido calcular matemáticamente el comportamiento del desarrollo del proyecto lo cual nos facilitó conocer en detalles tanto el esfuerzo como el tiempo para realizar el software, así como la cantidad de personas necesarias para terminarlo en el tiempo previsto y el costo del producto terminado, permitiendo confirmar la factibilidad de la construcción del sistema propuesto.

Conclusiones Generales

- El desarrollo de la aplicación permitió llevar a cabo la modelación de guías de diagnóstico al Sistema de Control Interno de forma personalizada para toda entidad o área administrativa.
- La descripción del funcionamiento del sistema representado por un modelo de dominio, permitió mostrar los roles y conceptos principales lo que lo hacen atractivo, usable, rápido y confiable.
- La herramienta informática perfecciona la evaluación al Sistema de Control Interno adaptado al área a diagnosticar, mejorando de esta forma la eficacia del resultado a obtener.
- La aplicación constituye una herramienta de alta calidad que optimiza y mejora el trabajo de auditores, especialistas y directivos interesados en el tema.
- El análisis de factibilidad basado en los aspectos operativos, técnicos y económicos permitió confirmar que el producto es factible atendiendo tanto a los beneficios como al esfuerzo y el costo del mismo.

Recomendaciones

- Se le propone al Comité de Control y Prevención y al Grupo de Auditoria del Instituto Superior Minero Metalurgico la aplicación del software en el desarrollo de su trabajo como una de las acciones para validar el mismo y proponer su implementación en otras entidades empresariales.

- Se propone la presentación del software a los organismos centrales del estado para su revisión, validación y generalización.

- Se propone la participación del presente trabajo en eventos científicos relacionados con la materia.

- Hacer una versión del software en plataforma web, lo que permitirá perfeccionar y potenciar las prestaciones del mismo.

- Se recomienda al Departamento de Contabilidad y Finanzas del ISMM Moa la utilización del software mediante la impartición de la asignatura Sistema de Control Interno, perteneciente la disciplina Auditoría.

Referencias Bibliográficas

[1]- http://www.monografias.com/trabajos59/evolucion-control-interno/evolucion-control-interno.shtml

[2]- Segundo Programa de Preparación Económica para Cuadros 2005, página 3, primer párrafo.

[3]- Segundo Programa de Preparación Económica para Cuadros 2005, página 3, tercer párrafo.

Bibliografía

1. Dr C José Carlos del Toro Ríos, MSc Antonio Fonteboa Vizcaíno, Dra C Elvira Armada Trabas, MSc Carlos Manuel Santos Cid. Segundo Programa de Preparación Económica para Cuadros 2005.

2. Elvira Armada Traba, José Carlos del Toro Ríos, Joaquín Infante Ugarte, Sergio Pozo Ceballos, Reyna Gómez Calaña y Grisel Tristá Arbesú, COMPROBACIÓN NACIONAL SOBRE EL GRADO DE IMPLEMENTACIÓN DE LA RESOLUCIÓN No 297/03 SOBRE CONTROL INTERNO. Tabloide 297.

3. Jacobson, G.B., James Rumbaugh., El Proceso Unificado de Desarrollo Software. 1999: Addison Wesley.

4. Antecedentes y evolución del Control Interno. Su desarrollo en Cuba http://www.monografias.com/trabajos59/evolucion-control-interno/evolucion-control-interno.shtml (2/11/2008).

5. El Control Interno http://www.monografias.com/trabajos37/control-interno/control-interno.shtml (25/10/2008).

6. http://es.wikipedia.org/wiki/Proceso_Unificado_de_Rational (10/12/2008).

7. Lenguaje de programación C++ http://es.wikipedia.org/wiki/C%2B%2B (20/01/2009).

8. Lenguaje de programación Delphi http://es.wikipedia.org/wiki/Delphi (17/01/2008).

9. Lenguaje de programación Java http://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n_Java (17/01/2008).

10. Antecedentes y Estructura del Control Interno

http://www.munialajuela.go.cr/app/frmCntrlinterno.htm (17/01/2008).

11. Control Interno

http://www.datasec-oft.com/sp/component/option,com_frontpage/Itemid,1/ (19/01/2008).

12. Sistema Matricial de Control Interno

http://www.monografias.com/trabajos14/control/control.shtml

(8/11/2008).

13. Ministerio de Finanzas y Precios

Anexo Res 297 – 03 control interno

(18/11/2008).

Anexo 1 Descripción de los Casos de Usos

Tabla 15 Descripción del caso de uso "Modificar Usuarios".

Nombre de caso de uso	Modificar Usuarios
Actor(es)	Administrador(inicia)
Resumen	Inicia cuando un administrador desea cambiar la contraseña de cualquier usuario del sistema.
Referencia	R1.3
Precondiciones	Deben existir usuarios.
Poscondiciones	Quedan modificada la contraseña del usuario.
Requisitos especiales	Se debe validar que se hayan introducidos los datos correspondientes.

Tabla 16 Descripción del caso de uso "Eliminar Usuarios".

Nombre de caso de uso	Eliminar Usuarios
Actor(es)	Administrador(inicia)
Resumen	Inicia cuando un administrador decide eliminar un usuario del sistema.
Referencia	R1.2
Precondiciones	Debe existir el usuario.
Poscondiciones	Queda eliminado del sistema el usuario
Requisitos especiales	Se debe validar que se hayan introducido los datos correspondientes.

Tabla 17 Descripción del caso de uso "Mostrar Listado Usuarios".

Nombre de caso de uso	Mostrar Listado Usuarios
Actor(es)	Administrador, Usuario Avanzado, Invitado (inician)
Resumen	Inicia cuando se decide ver los usuarios que están registrados en el sistema.
Referencia	R5
Precondiciones	Deben existir usuarios en el sistema.
Poscondiciones	Quedan listados los usuarios existentes.
Requisitos especiales	_

Tabla 18 Descripción del caso de uso "Insertar Guía Metodológica".

Nombre de caso de uso	Insertar Guía Metodológica
Actor(es)	Administrador, Usuario Avanzado (inician)
Resumen	Inicia cuando un se decide crear una nueva guía.
Referencia	R2.1
Precondiciones	_
Poscondiciones	Quedan insertados los datos listos para ser procesados.
Requisitos especiales	Se debe validar que se hayan introducidos los datos correspondientes.

Tabla 19 Descripción del caso de uso "Eliminar Guía Metodológica".

Nombre de caso de uso	Eliminar Guía Metodológica
Actor(es)	Administrador, Usuario Avanzado (inician)
Resumen	Inicia cuando se decide eliminar una guía del sistema.

Referencia	R2.2
Precondiciones	Debe existir la guía que se quiere eliminar
Poscondiciones	Queda eliminada la guía del sistema.
Requisitos especiales	Se debe validar que se hayan introducidos los datos correspondientes.

Tabla 20 Descripción del caso de uso "Modificar Guía Metodológica".

Nombre de caso de uso	Modificar Guía Metodológica
Actor(es)	Administrador, Usuario Avanzado (inician)
Resumen	Inicia cuando se decide cambiar los datos de una guía.
Referencia	R2.3
Precondiciones	Debe existir la guía que se desea modificar.
Poscondiciones	Quedan modificados los datos de la guía listos para ser procesados.
Requisitos especiales	Se debe validar que se hayan introducidos los datos correspondientes.

Tabla 21 Descripción del caso de uso "Evaluar Guía Metodológica".

Nombre de caso de uso	Evaluar Guía Metodológica
Actor(es)	Administrador, Usuario Avanzado (inician)
Resumen	Inicia cuando se decide realizar una evaluación de una guía cualquiera.
Referencia	R2.4
Precondiciones	Deben existir las guías por la que se decide hacer la evaluación.
Poscondiciones	Se deben mostrar los resultados alcanzados en la evaluación.
Requisitos especiales	_

Tabla 22 Descripción del caso de uso "Insertar materiales de consulta".

Nombre de caso de uso	Insertar materiales de consulta
Actor(es)	Administrador, Usuario Avanzado (inician)
Resumen	Inicia cuando se decide ingresar un nuevo material de consulta.
Referencia	R3.1
Precondiciones	_
Poscondiciones	Quedan insertados los datos listos para ser consultados por los usuarios.
Requisitos especiales	Se debe validar que se hayan introducidos los datos correspondientes.

Tabla 23 Descripción del caso de uso "Eliminar materiales de consulta".

Nombre de caso de uso	Eliminar materiales de consulta
Actor(es)	Administrador, Usuario Avanzado (inician)
Resumen	Inicia cuando se decide eliminar un material de consulta.
Referencia	R3.2
Precondiciones	Debe existir el material de consulta que se quiere eliminar.
Poscondiciones	Queda eliminado el material de consulta del sistema.
Requisitos especiales	Se debe validar que se hayan introducidos los datos del material que se decidió eliminar.

Tabla 24 Descripción del caso de uso "Mostrar listado de Guías".

Nombre de caso de uso	Mostrar listado de Guías
Actor(es)	Administrador, Usuario Avanzado, Invitado (inician)
Resumen	Inicia cuando se desean ver las guías que existen en el sistema.
Referencia	R6
Precondiciones	Deben existir guías creadas en el sistema.
Poscondiciones	Quedan mostradas las guías que existen con sus datos.
Requisitos especiales	_

Tabla 25 Descripción del caso de uso "Mostrar listado de Evaluaciones".

Nombre de caso de uso	Mostrar listado de Evaluaciones
Actor(es)	Administrador, Usuario Avanzado, Invitado (inician)
Resumen	Inicia cuando se decide ver las evaluaciones realizadas.
Referencia	R7
Precondiciones	Debe haberse realizado alguna evaluación.
Poscondiciones	Quedan mostradas todas las evaluaciones realizadas.
Requisitos especiales	_

Tabla 26 Descripción del caso de uso "Mostrar listado materiales de consulta".

Nombre de caso de uso	Mostrar listado materiales de consulta
Actor(es)	Administrador, Usuario Avanzado, Invitado (inician)
Resumen	Inicia cuando se deciden ver los materiales existentes que se pueden consultar.
Referencia	R4
Precondiciones	Deben existir materiales previamente ingresados a la base de datos.
Poscondiciones	Quedan mostrados todos los materiales que existen.
Requisitos especiales	_

Tabla 27 Descripción del caso de uso "Realizar resumen valorativo".

Nombre de caso de uso	Realizar resumen valorativo
Actor(es)	Administrador, Usuario Avanzado (inician)
Resumen	Inicia cuando se decide realizar un resumen valorativo de una evaluación.
Referencia	R8
Precondiciones	Debe haberse realizado alguna evaluación.
Poscondiciones	Queda realizado un resumen de la evaluación y insertado en base de datos.
Requisitos especiales	_

Tabla 28 Descripción del caso de uso "Exportar resumen".

Nombre de caso de uso	Exportar resumen
Actor(es)	Administrador, Usuario Avanzado, Invitado (inician)
Resumen	Inicia cuando se decide exportar un resumen

	valorativo de una evaluación.
Referencia	R8
Precondiciones	El sistema debe tener algún resumen realizado para ser exportado.
Poscondiciones	Los datos que se exporten tendrán salida a los formatos establecidos.
Requisitos especiales	_

Anexo 2 Descripción de las clases

Tabla 29 Descripción de la clase "Area_o_Dpto".

Nombre: Area_o_Dpto	
Tipo de clase: Entidad	
Atributo	**Tipo**
Id_area	Integer
Nombre	String
Para cada responsabilidad	
Nombre	**Descripción**
Get_nombre()	Devuelve el nombre del area
Insertar(int id, String area)	Se pasa por parámetro el id y el nombre del area y se inserta en la base de datos
Eliminar(String area)	Se pasa por parámetro el nombre de la empresa que se desea eliminar

Tabla 30 Descripción de la clase "Componentes".

Nombre: Componentes	
Tipo de clase: Entidad	
Atributo	**Tipo**
Id_componente	Integer
Nombre	String
Para cada responsabilidad	
Nombre	**Descripción**
Get_nombre()	Devuelve el nombre del componente
Insertar(int id, String comp)	Se pasa por parámetro el id y el nombre del componente y se inserta en la base de datos
Eliminar(String comp)	Se pasa por parámetro el nombre del componente que se desea eliminar

Tabla 31 Descripción de la clase "Normas".

Nombre: Normas	
Tipo de clase: Entidad	
Atributo	**Tipo**
Id_norma	Integer
Nombre	String
Para cada responsabilidad	
Nombre	**Descripción**

Get_nombre()	Devuelve el nombre de la norma
Insertar(int id, String norma)	Se pasa por parámetro el id y el nombre de la norma y se inserta en la base de datos
Eliminar(String norma)	Se pasa por parámetro el nombre de la norma que se quiere eliminar

Tabla 32 Descripción de la clase "Preguntas".

Nombre: Preguntas	
Tipo de clase: Entidad	
Atributo	**Tipo**
Id_pregunta	Integer
Nombre	String
Para cada responsabilidad	
Nombre	**Descripción**
Get_nombre()	Devuelve el nombre de la pregunta
Insertar(int id, String preg)	Se pasa por parámetro el id y el nombre de la pregunta y se inserta en la base de datos
Eliminar(String preg)	Se pasa por parámetro el nombre de la pregunta que se desea eliminar

Tabla 33 Descripción de la clase "Sub_preguntas".

Nombre: Sub_preguntas	
Tipo de clase: Entidad	
Atributo	**Tipo**
Id_subpregunta	Integer
Nombre	String
Para cada responsabilidad	
Nombre	**Descripción**
Get_nombre()	Devuelve el nombre de la subpregunta
Insertar(int id, String subpreg)	Se pasa por parámetro el id y el nombre de la subpregunta y se inserta en la base de datos
Eliminar(String subpreg)	Se pasa por parámetro el nombre de la subpregunta que se desea eliminar

Tabla 34 Descripción de la clase "Tematicas".

Nombre: Tematicas	
Tipo de clase: Entidad	
Atributo	**Tipo**

Id_tematica	Integer
Nombre	String
Para cada responsabilidad	
Nombre	**Descripción**
Get_tematica()	Devuelve el nombre de la tematica
Insertar(int id, String tematica)	Se pasa por parámetro el id y el nombre de la tematica y se inserta en la base de datos
Eliminar(String tematica)	Se pasa por parámetro el nombre de la tematica que se desea eliminar

Tabla 35 Descripción de la clase "Usuarios".

Nombre: Usuarios	
Tipo de clase: Entidad	
Atributo	**Tipo**
Id_usuario	Integer
Nombre_usuario	String
Nombre_completo	String
Contrasena	String
Para cada responsabilidad	
Nombre	**Descripción**
Get_nombre()	Devuelve el nombre del usuario
Insertar(int id, String nomb_u, String nomb_completo, String contrasena)	Se pasa por parámetro el id, el nombre de usuario, el nombre completo y la contraseña y se inserta en la base de datos
Modificar(String nomb_u)	Se pasa por parámetro el nombre de usuario que se desea modificar y se guardan los cambios en la base de datos
Eliminar(String usuario)	Se pasa por parámetro el nombre del usuario que se desea eliminar

Tabla 36 Descripción de la clase "Materiales_de_consulta".

Nombre: Materiales_de_consulta	
Tipo de clase: Entidad	
Atributo	**Tipo**
Id_material	Integer
Nombre	String
Para cada responsabilidad	
Nombre	**Descripción**
Get_nombre()	Devuelve el nombre del material

Insertar(int id, String material)	Se pasa por parámetro el id y el nombre del material y se inserta en la base de datos
Eliminar(String material)	Se pasa por parámetro el nombre del material que se desea eliminar

Tabla 37 Descripción de la clase "Evaluacion".

Nombre: Evaluacion	
Tipo de clase: Entidad	
Atributo	**Tipo**
Fecha	Date
Id_evaluacion	Integer
Result_cuantit	Float
Result_cualit	String
Para cada responsabilidad	
Nombre	**Descripción**
Get_result_cuantit()	Devuelve el resultado cuantitativo de la evaluación
Get_result_cualit()	Devuelve el resultado cualitativo de la evaluación
Insertar(Date fecha, int id, Float result_cuant, String result_cualit)	Se pasa por parámetro la fecha, el id, el resultado cuantitativo y el cualitativo de la evaluación y se inserta en la base de datos
Eliminar(int id)	Se pasa por parámetro el id de la evaluación que se desea eliminar

Tabla 38 Descripción de la clase "Guia_Metodologica".

Nombre: Guia_Metodologica	
Tipo de clase: Entidad	
Atributo	**Tipo**
Nombre	String
Id_guia	Integer
Fecha	Date
Visible	Boolean
Para cada responsabilidad	
Nombre	**Descripción**
Get_nombre()	Devuelve el nombre de la guia
Insertar(String nomb, int id_guia, Date fecha, Boolean visible)	Se pasa por parámetro el nombre, el id, la fecha, la visibilidad y se inserta en la base de datos

Modificar(String nomb)	Se pasa por parámetro el nombre de la guia que se desea modificar y se guardan los cambios actualizados en la base de datos
Eliminar(String nomb)	Se pasa por parámetro el nombre de la guia que se desea eliminar de la base de datos

Tabla 39 Descripción de la clase "Resumen_valorativo".

Nombre: Resumen_valorativo	
Tipo de clase: Entidad	
Atributo	**Tipo**
Id_resumen	Integer
Evaluacion	Integer
Usuario	Integer
Para cada responsabilidad	
Nombre	**Descripción**
Get_id()	Devuelve el id del resumen
Insertar(int id, int evaluacion,int usuario)	Se pasa por parámetro el id del resumen, el id de evaluacion, el id del usuario que realiza el resumen y se inserta en la base de datos
Eliminar(int id)	Se pasa por parámetro el id del resumen que se desea eliminar.

Tabla 40 Descripción de la clase "Base".

Nombre: Base	
Tipo de clase: Controladora	
Atributo	**Tipo**
Para cada responsabilidad	
Nombre	**Descripción**
Realizar_diagnostico(Guia_Metodologica guia)	Se pasa por parámetro una guia_metodologica, y luego se evalúa
Exportar_informacion(JTable table)	Se pasa por parámetro una tabla que contiene informacion, la misma es esportada a un fichero pdf, doc o xls.
Ver_lista_guias()	Devuelve un listado con todas las guías que existen en la base de datos, visible a los usuarios finales
Ver_lista_evaluaciones()	Devuelve un listado con todas la evaluaciones que existen el la base de datos
Ver_lista_materiales_de_consulta()	Devuelve un listado con todos los materiales

	de consultas que existen el la base de datos
Ver_lista_usuarios()	Devuelve un listado con todos los usuarios que existen el la base de datos
Conecta()	Establece la conexión con la base de datos
Desconecta()	Cierra la conexión con la base de datos

Anexo 3 Interfaz de Usuario

Figura 10 Interfaz de Usuario “Insertar Guía Metodológica”.

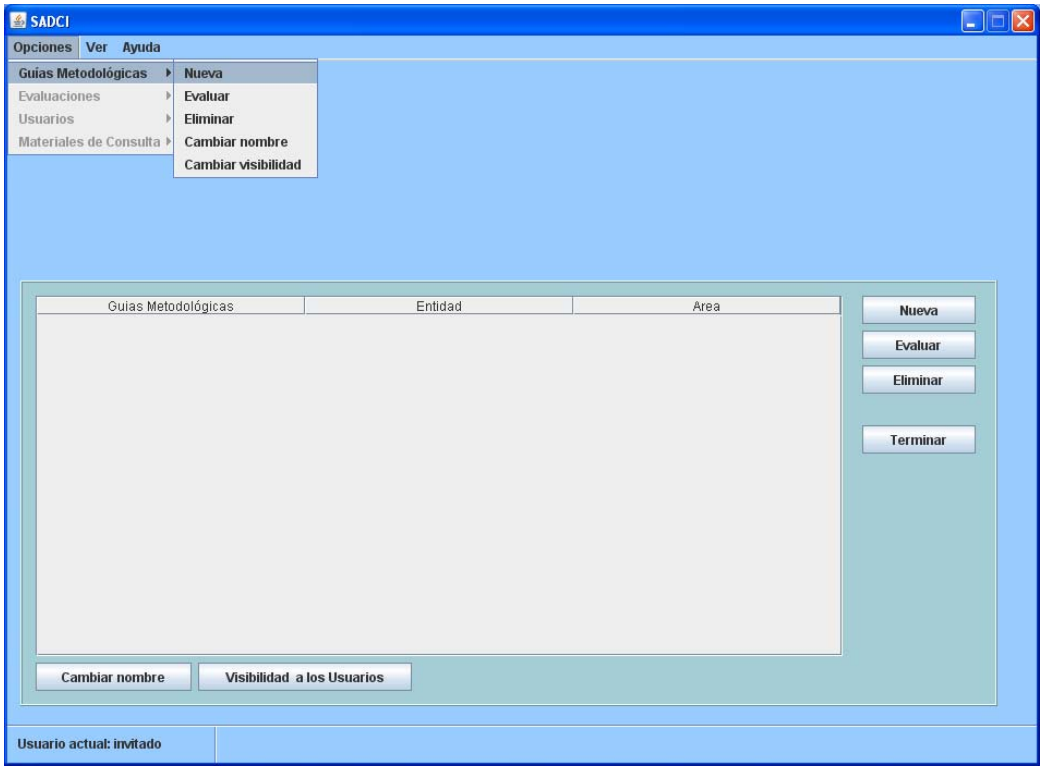


Figura 11 Interfaz de Usuario “Evaluar Guía Metodológica”.

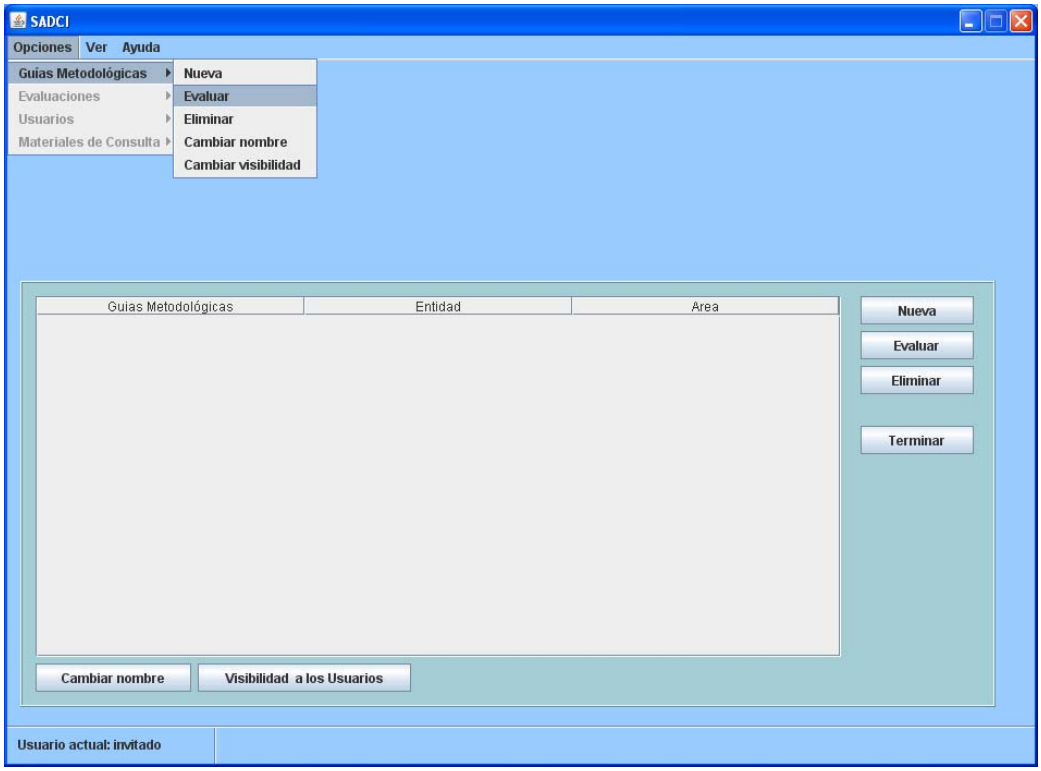


Figura 12 Interfaz de Usuario “Eliminar Guía Metodológica”.

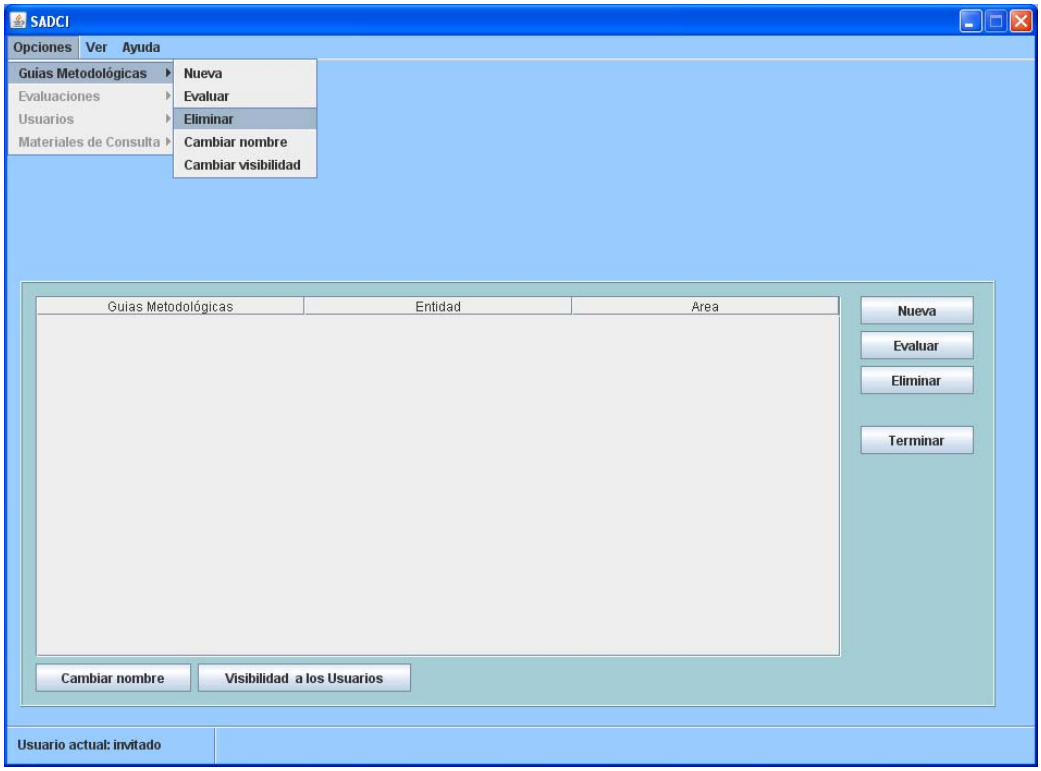


Figura 13 Interfaz de Usuario “Modificar Guía Metodológica”.

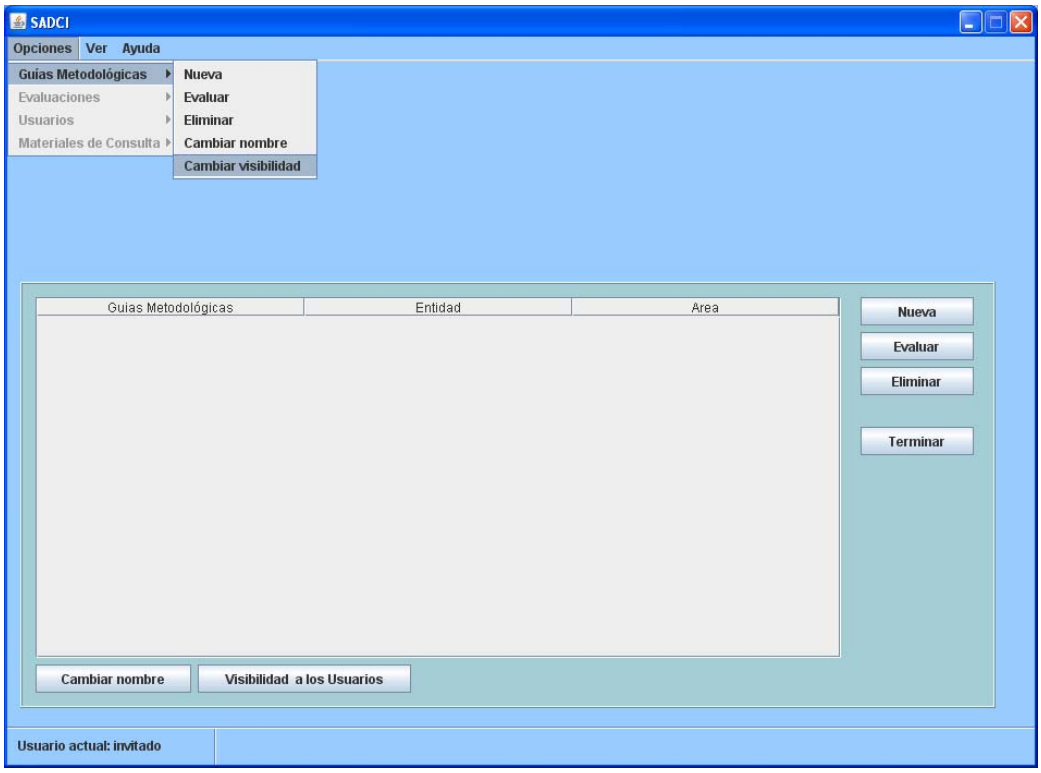


Figura 14 Interfaz de Usuario “Insertar Usuario”.

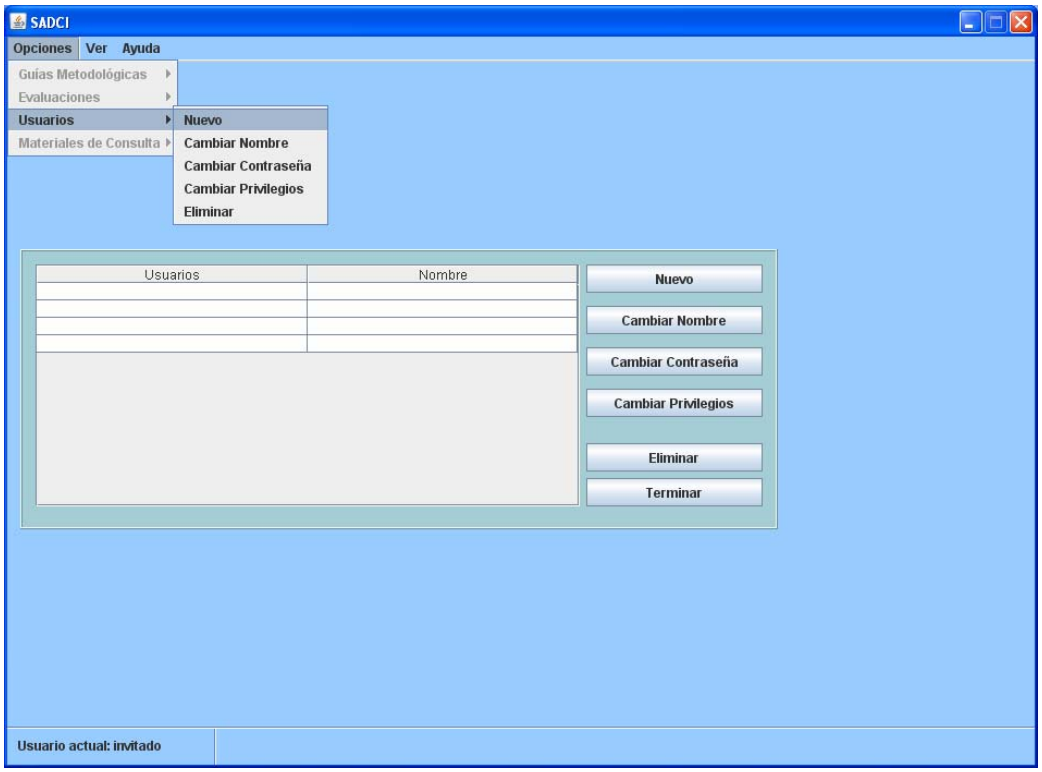


Figura 15 Interfaz de Usuario “Eliminar Usuario”.

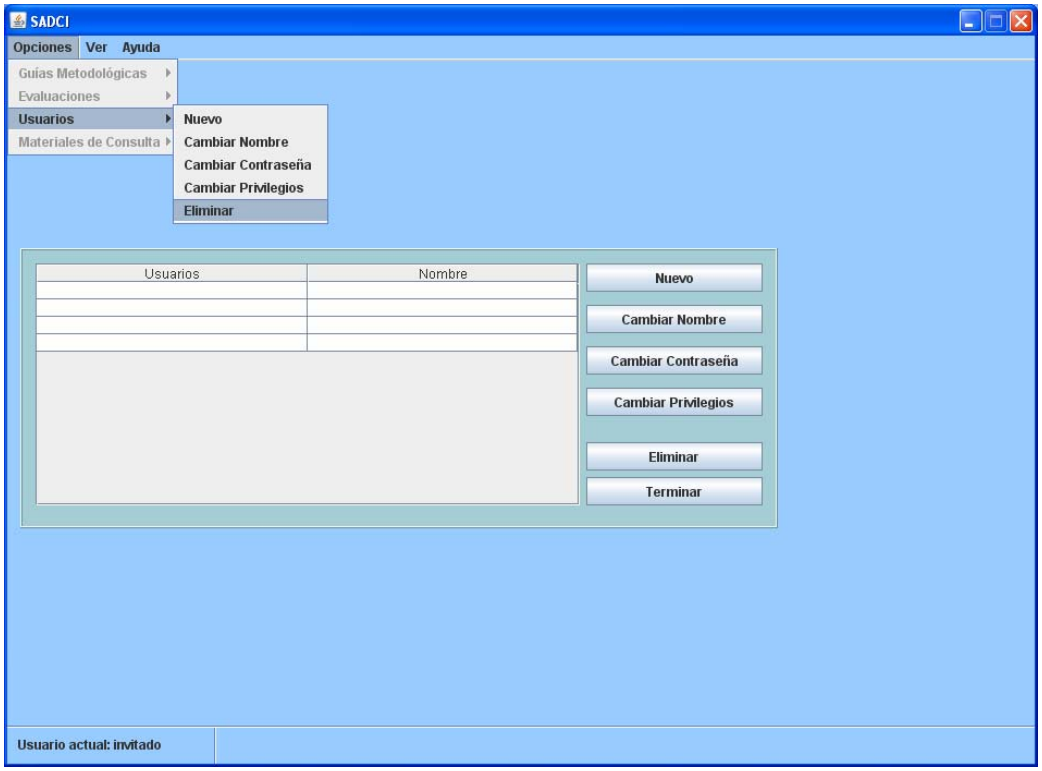


Figura 16 Interfaz de Usuario “Modificar Usuario”.

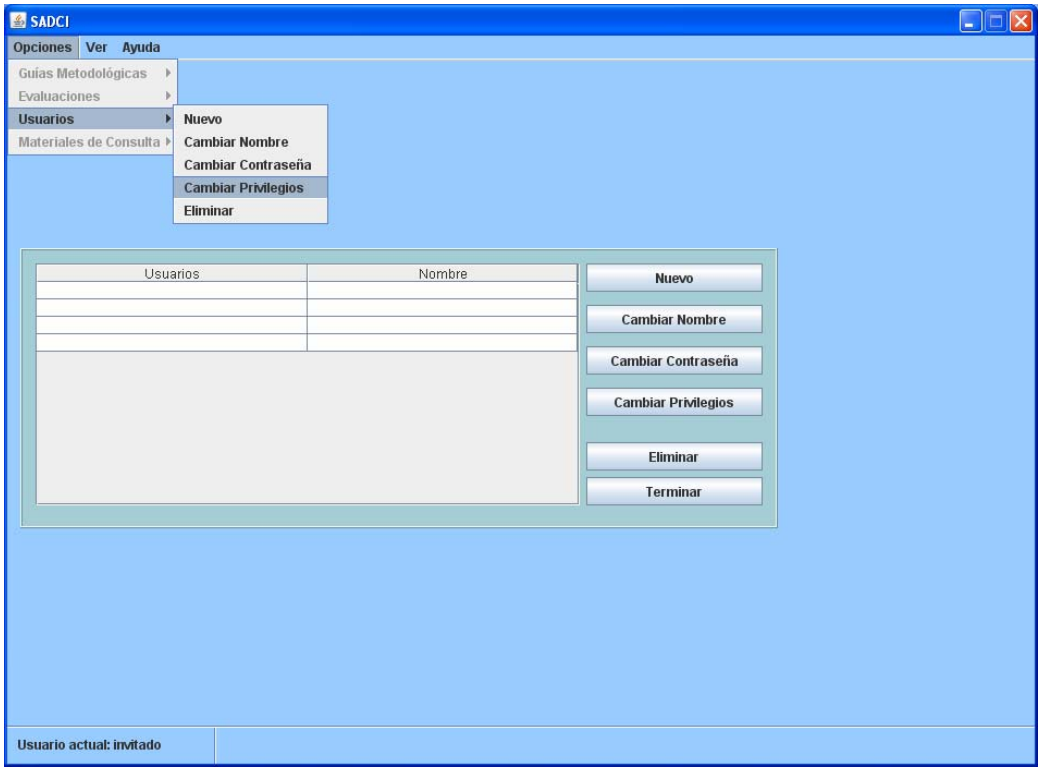


Figura 17 Interfaz de Usuario “Ingresar Material de Consulta”.

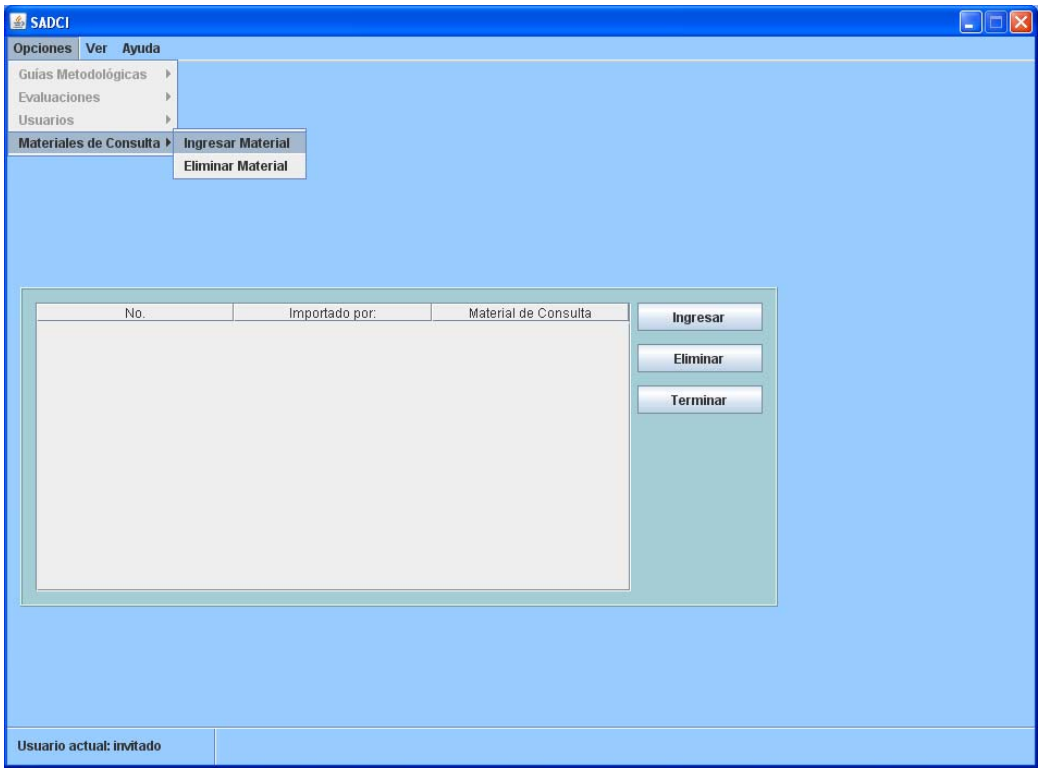
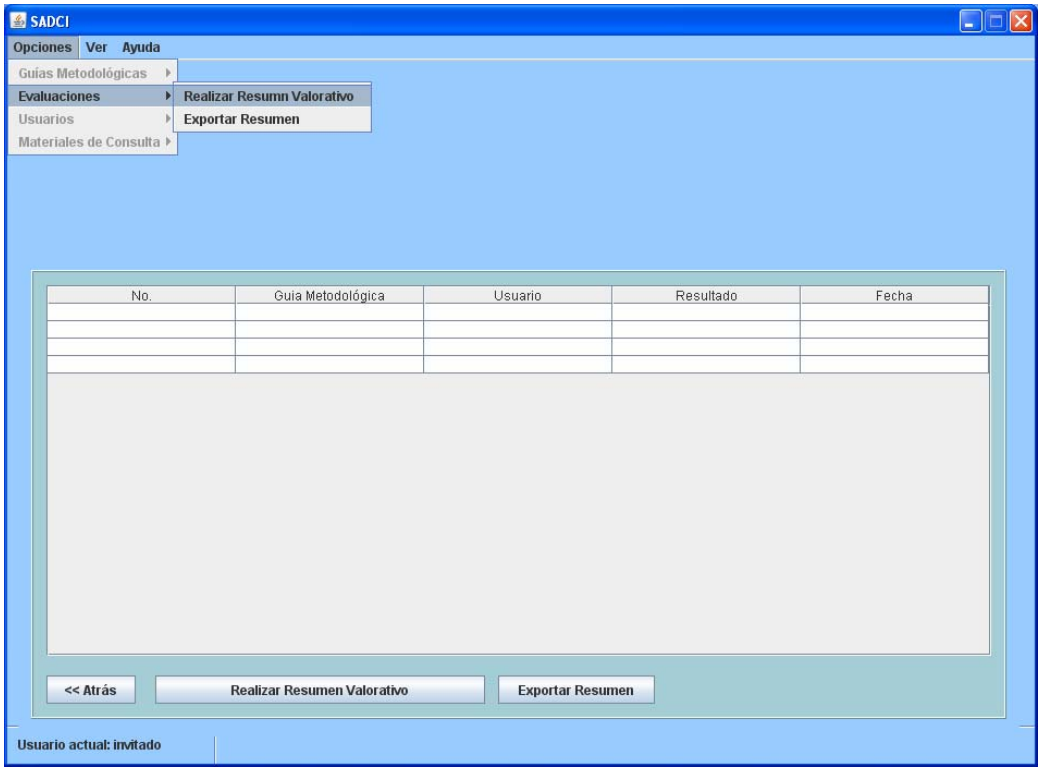


Figura 18 Interfaz de Usuario “Realizar Resumen Valorativo”.



Anexo 4 Diagramas de Secuencia

Figura 19 Diagrama de Secuencia del caso de uso “Eliminar_Guia_Metodológica”.

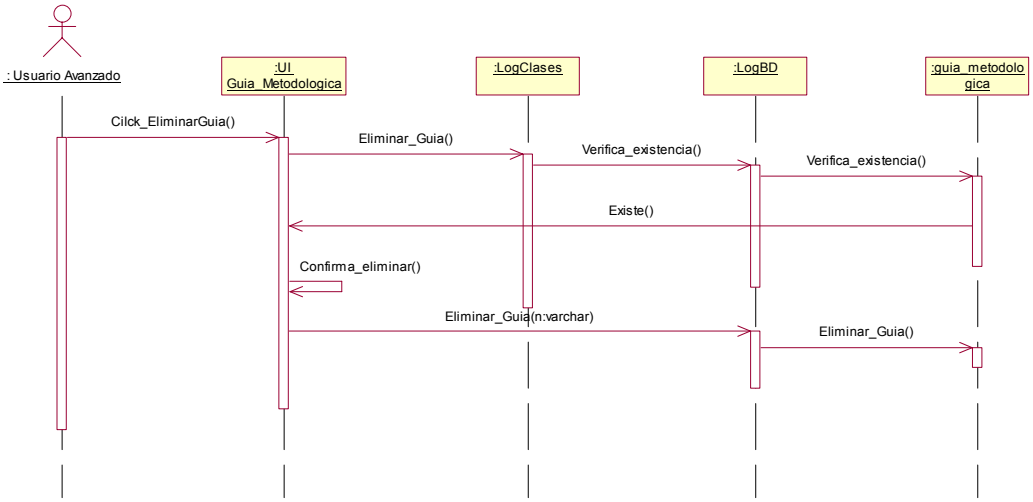


Figura 20 Diagrama de Secuencia del caso de uso “Eliminar_Usuario”.

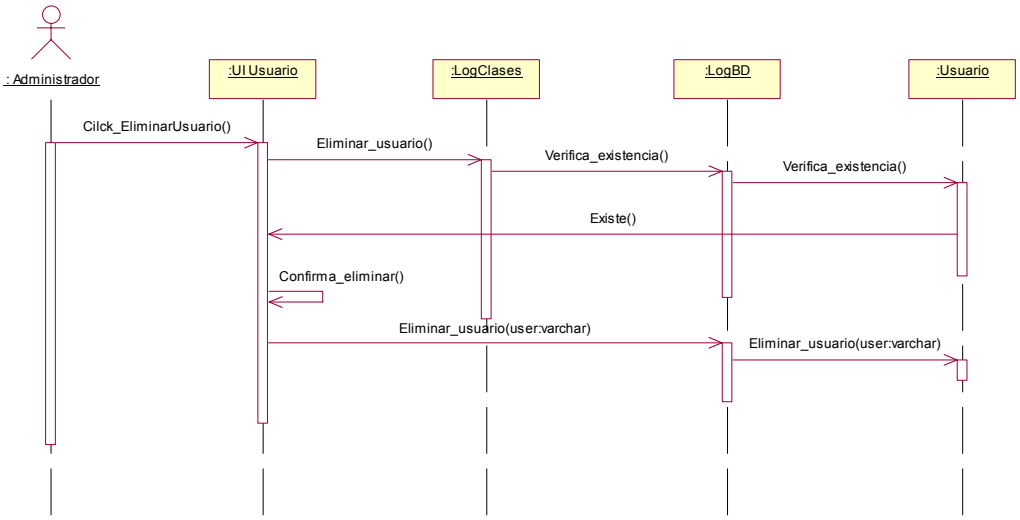


Figura 21 Diagrama de Secuencia del caso de uso “Exportar_Resumen”.

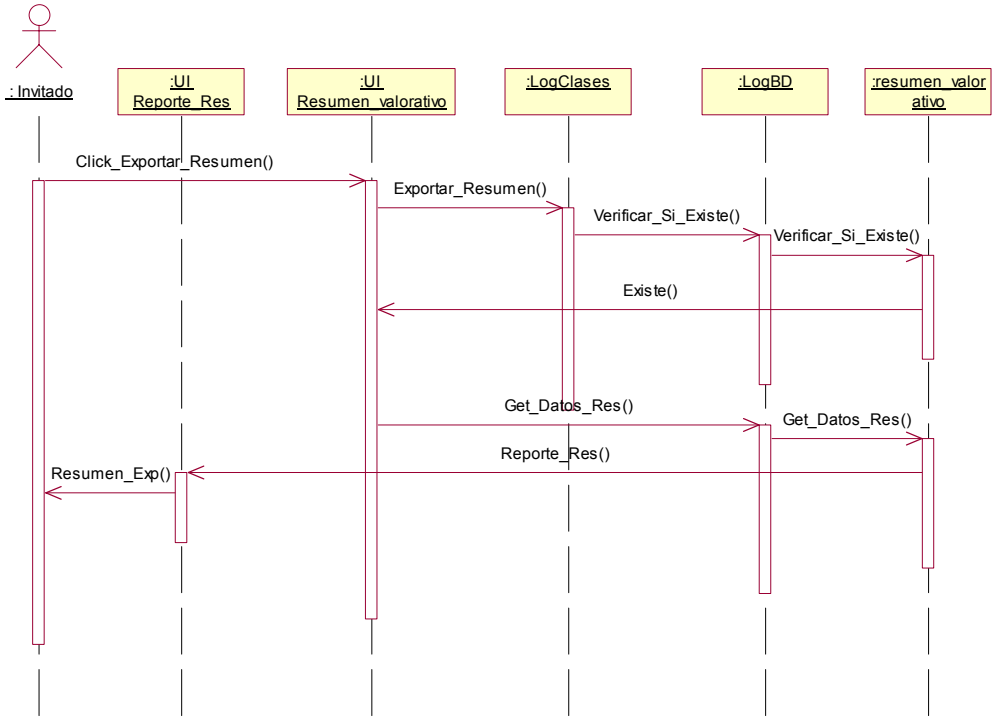


Figura 22 Diagrama de Secuencia del caso de uso “Insertar_Guia_Metodológica”.

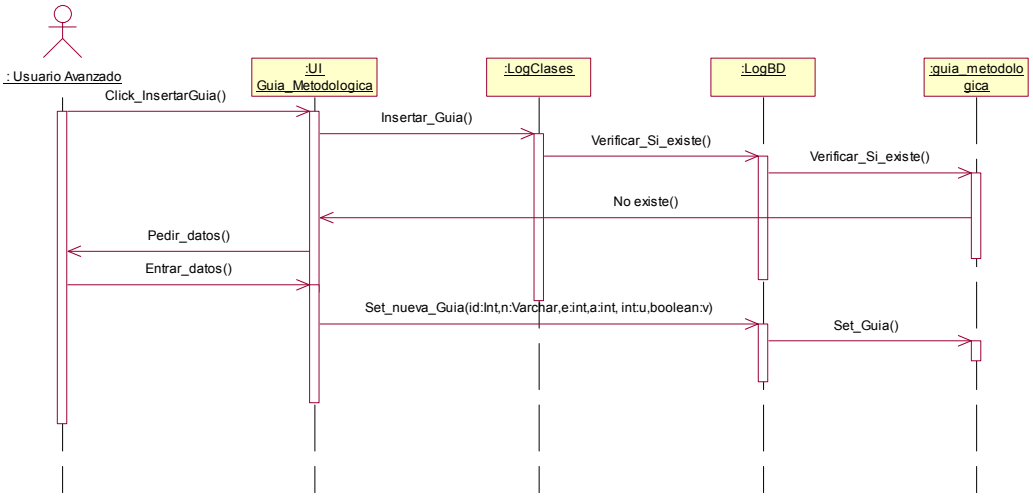


Figura 23 Diagrama de Secuencia del caso de uso “Insertar_Usuario”.

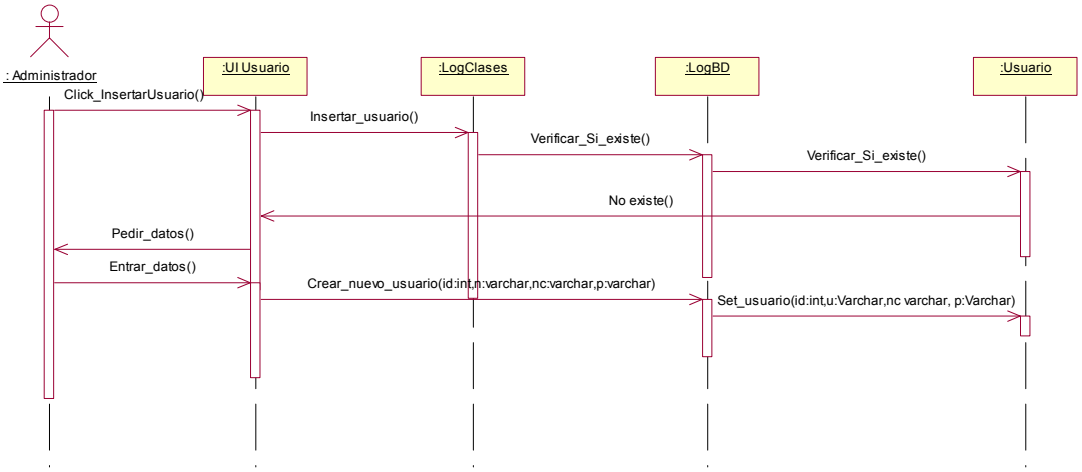


Figura 24 Diagrama de Secuencia del caso de uso “Mostrar_Listado_Evaluaciones”.

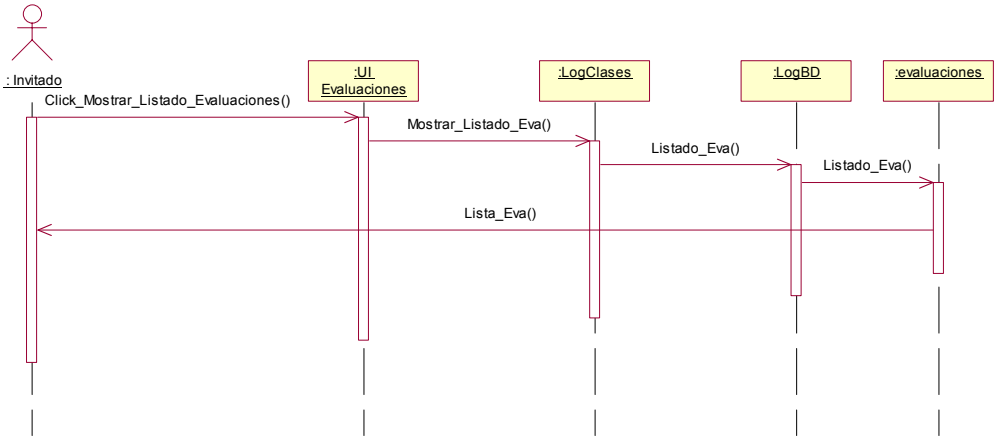


Figura 25 Diagrama de Secuencia del caso de uso “Mostrar_Listado_Guias”.

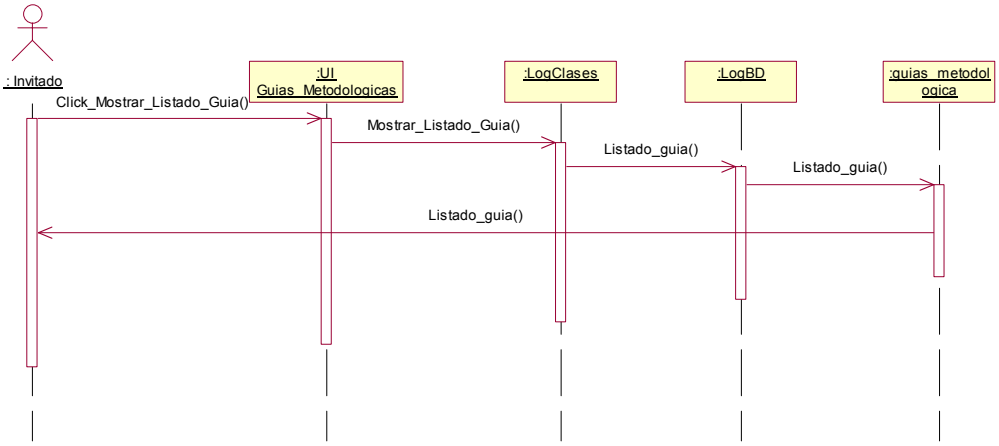


Figura 26 Diagrama de Secuencia del caso de uso “Mostrar_Listado_Materiales”.

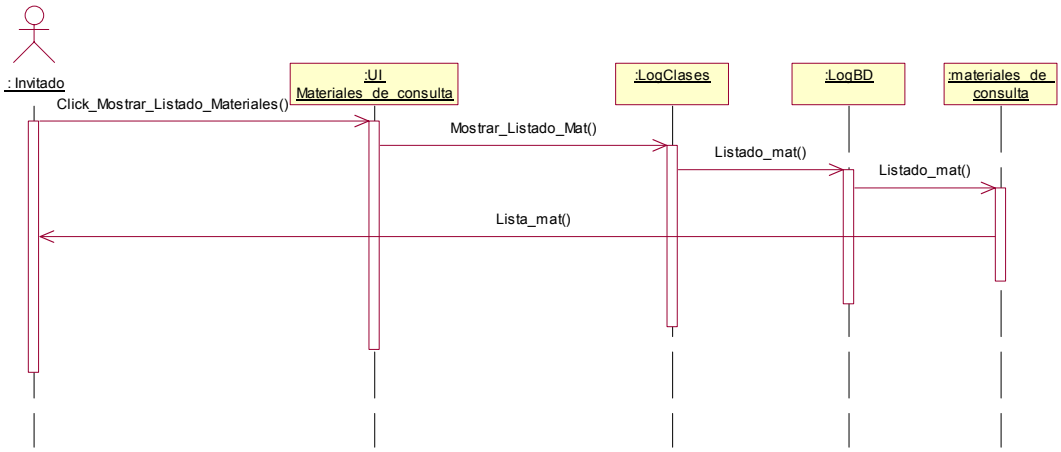


Figura 27 Diagrama de Secuencia del caso de uso “Mostrar_Listado_Usuarios”.

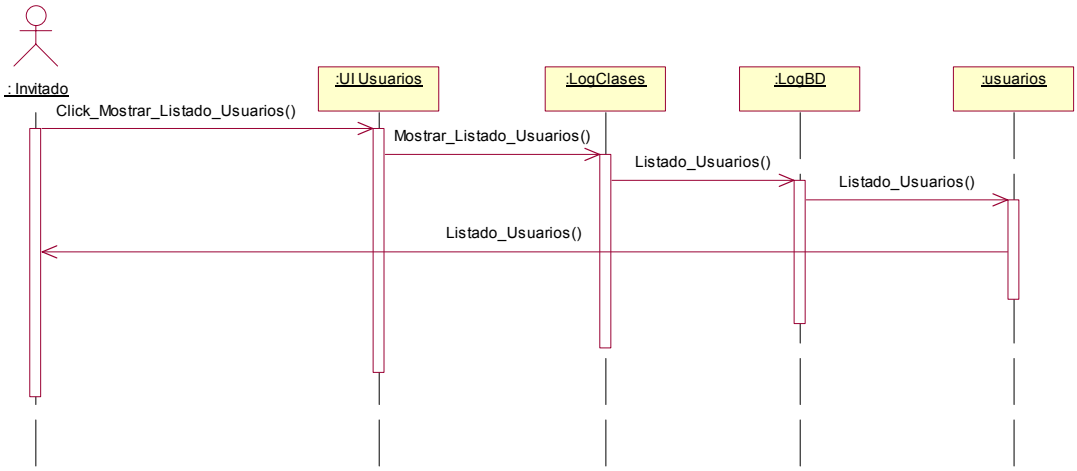


Figura 28 Diagrama de Secuencia del caso de uso “Modificar_Guia”.

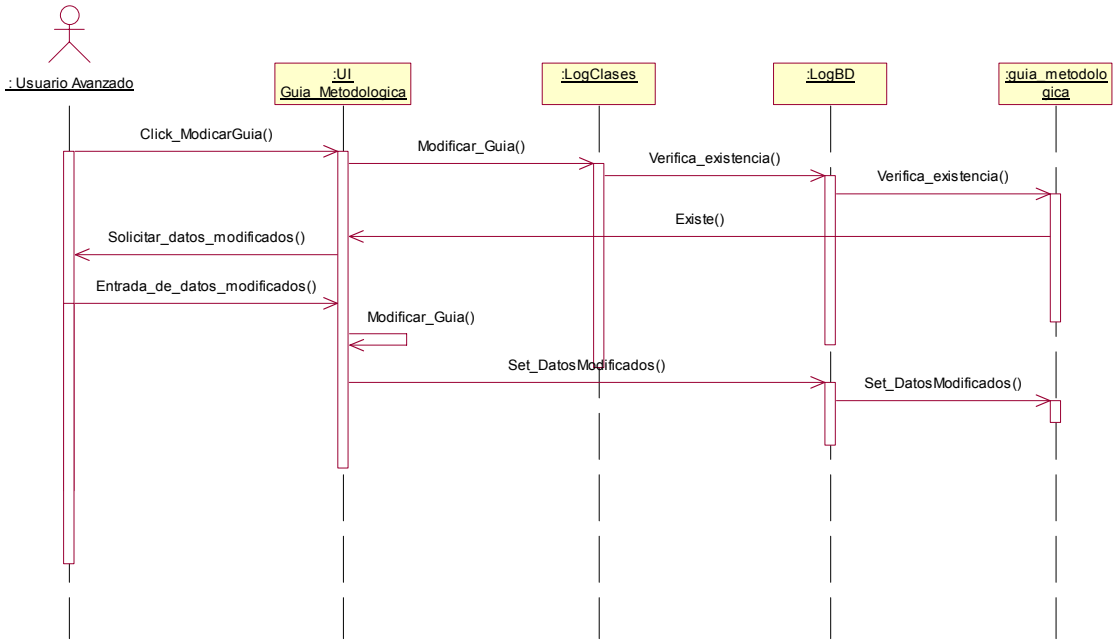


Figura 29 Diagrama de Secuencia del caso de uso “Modificar_Datos_Usuario”.

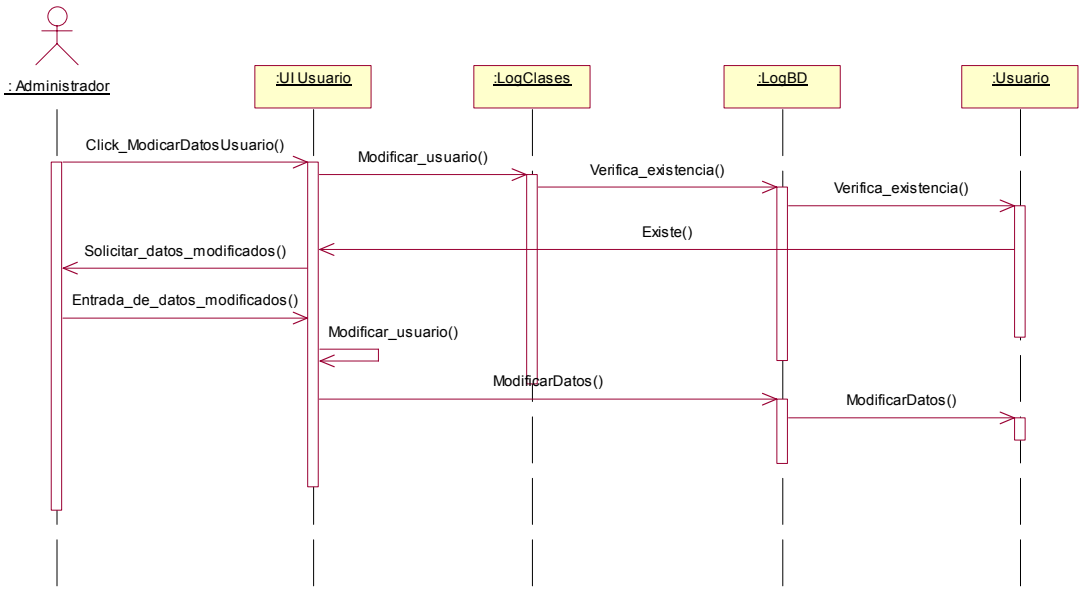


Figura 30 Diagrama de Secuencia del caso de uso “Realizar_Resumen”.

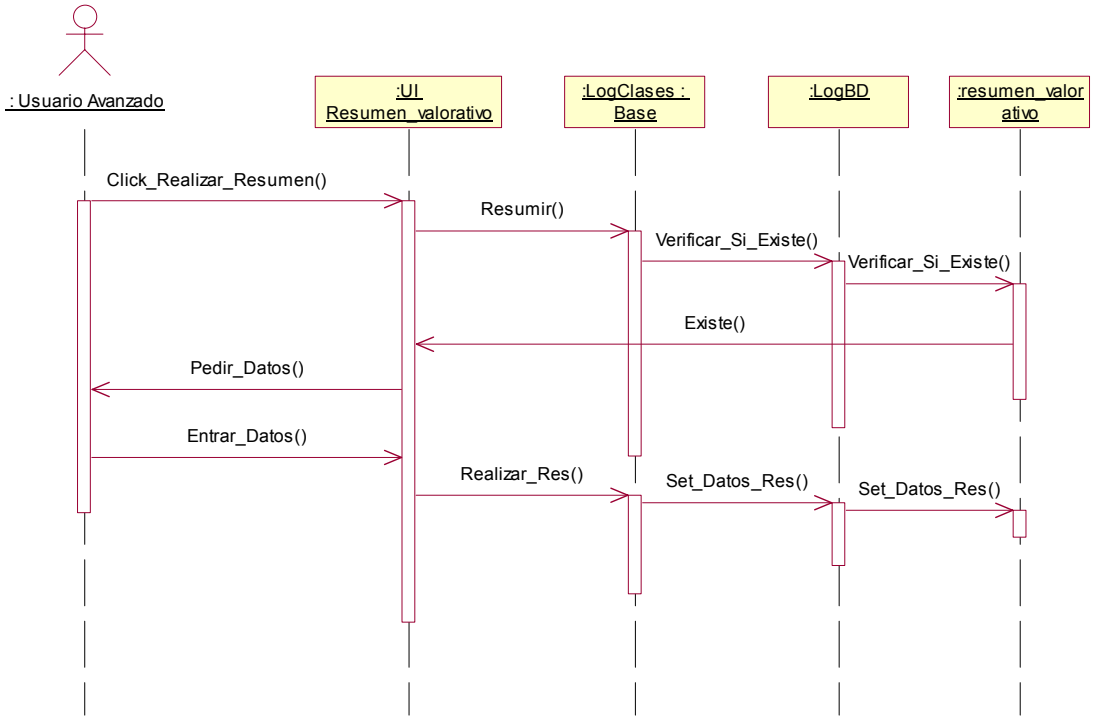
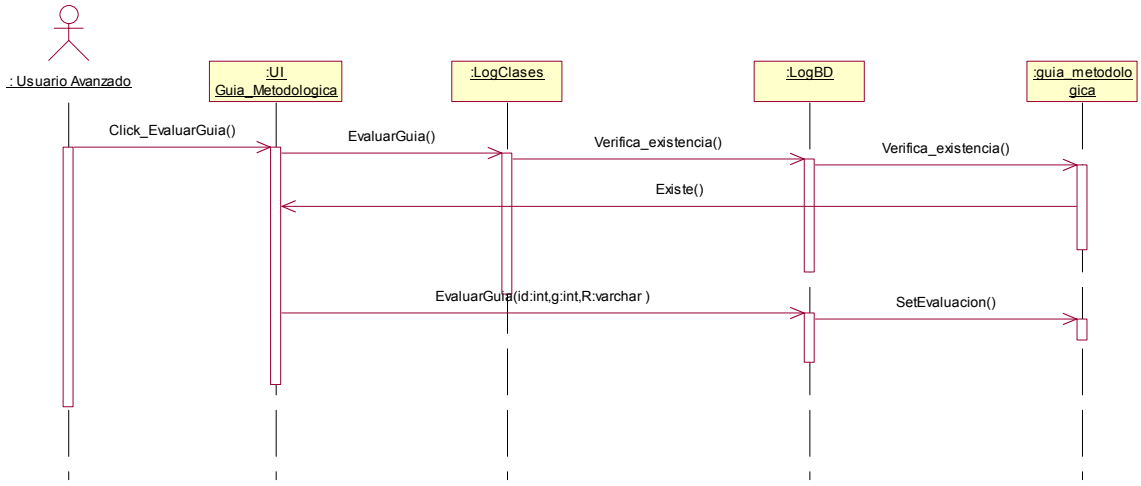



Figura 31 Diagrama de Secuencia del caso de uso “Evaluar_Guia_Metodológica”.