

Universidad de las Ciencias Informáticas

Facultad 1

Análisis y Diseño del Sistema de Control Interno de la Universidad de las Ciencias Informáticas

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

Autor(es): Sailin Trujillo Muzaurieta

Yensy Montalvo Arnet

Tutor(es): Ing. Dasiel Alberto Pérez Suárez

Ing. Reldy Pérez Morales

Ciudad de La Habana, Junio de 2010

DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas (UCI) el derecho de que haga el uso que consideren conveniente con este trabajo.

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

Sailin Trujillo Muzaurieta

Yensy Montalvo Arnet

Ing. Dasiel Alberto Pérez

Ing. Reldy Pérez Morales

AGRADECIMIENTOS

De Sailin Trujillo Muzaurieta:

Le agradezco a Dios y principalmente a mi madre, por todo el amor que me ha dado, por su apoyo y comprensión por ser mi consejera y por apoyarme en los momentos más importantes de mi vida. A mi abuela por todo el apoyo y confianza que me dio y por enseñarme el valor de la familia. A mi querida hermanita y a mi tía Magdalena por todo su cariño. A mis tíos Joel, Anubis y Juan Carlos por creer en mi y darme su apoyo incondicional. A mi abuelo, a mis tías, tíos, primos y toda mi familia por todo lo que hacen por mi y acompañarme en los momentos más difíciles. A Reñier por ayudarme tanto.

A todos mis queridos amigos por la atención y la ayuda que me brindaron a lo largo de la carrera, por haber hecho especiales estos años, a Yoel, a Yasier, a Eli, a Alexander, a Yeny, a Dayana, a José Ángel, a Roilers, a Isuel y especialmente a Liu y por ser mi brazo derecho y dedicarme tanto de su tiempo. A mis tutores Dasiel y Reidy por su paciencia y dedicación. A la Universidad de las Ciencias Informáticas, a la directiva de la facultad, a los profesores y a todos mis compañeros. A todas las personas que de una forma u otra estuvieron presentes y me brindaron su mano.

De Yensy Montalvo Arnet:

Mis agradecimientos empezaran por mis amigos que acudieron al ser llamado. Mis tíos y primos tanto presente como ausentes siempre les estaré agradecido pues de una forma u otra han sabido ayudarme y desearme muchos éxitos. Mis abuelos que se cuanto han hecho por mi y su preocupación infinita por mi porvenir. Mi novia que al parecer será la madre de mis hijos conjuntos sus familiares. Mis padres que han sido lo superlativo en mi vida seguiré honrándolo y son los responsables absolutos de estar donde me encuentro. Y a Dios que mantenga con vida a todos los que tenga una meta que cumplir.

DEDICATORIA

“A toda mi querida familia, en especial a mi madre María Luisa y mi abuela Elsa. Para ti abuela, porque éste fue tu sueño y hoy es realidad.”

Sailin Trujillo Muzaurieta



“A mis padres, familia y todo el que apporto un granito de arena en mi profesión.”

Yensy Montalvo Arnet

RESUMEN

Las nuevas tecnologías de la información y las comunicaciones más conocidas como TIC han impulsado el desarrollo de las naciones a nivel mundial. En Cuba se han insertado en casi todas las esferas ya que constituyen pilares fundamentales para la modernización. En la Universidad de las Ciencias Informáticas (UCI) se hace un uso intensivo de las TIC para su desarrollo y sostenimiento. El objetivo principal de este trabajo es realizar el análisis y diseño de una propuesta de aplicación Web que permita la gestión de los procesos de control interno en la UCI. En el presente trabajo se realiza un estudio de los procesos del negocio que se llevan a cabo en el Departamento de Auditoría de la universidad y se identifican las funcionalidades que el sistema propuesto debe tener. Para ello se investigan las tendencias y tecnologías más usadas para el desarrollo de aplicaciones Web, los lenguajes de programación, las metodologías para el desarrollo del software y los sistemas gestores de base de datos.

Se pretende con la presente investigación facilitar el trabajo de los implementadores, poniendo a su disposición una guía de cómo debe ser el sistema.

Palabras Claves: Control interno, Auditoría, TIC.

ÍNDICE

INTRODUCCIÓN	1
Capítulo 1: Fundamentación del tema.....	5
1.1 Introducción.....	5
1.2 Control Interno	5
1.3 Sistemas vinculados al campo de acción	13
1.4 Tecnologías actuales.....	14
1.4.1 Framework para el desarrollo Web	14
1.4.2 Lenguajes de programación para el desarrollo Web	14
1.4.3 Fundamentación del lenguaje de programación a utilizar.....	16
1.4.4 Apache.....	16
1.4.5 Navegadores.....	17
1.4.6 Lenguaje para la modelación.....	18
1.4.7 Fundamentación del lenguaje para la modelación a utilizar	18
1.4.8 Sistemas gestores de base de datos	19
1.4.9 Fundamentación del Sistema Gestor de Base de Datos a utilizar	19
1.4.10 Metodología para el desarrollo del software	20
1.4.11 Fundamentación de la metodología a utilizar	21
1.4.12 Herramientas de modelado visual.....	21
1.4.13 Fundamentación de la herramienta de modelado visual a utilizar	22
1.5 Patrones	22
1.5.1 Patrones de diseño.....	22

1.5.2 Patrones de Caso de Uso	23
1.5.3 Patrón de Arquitectura	25
1.6 Técnicas empleadas para la captura de los requisitos	25
1.7 Conclusiones.....	26
Capítulo 2: Descripción de la solución propuesta.....	27
2.1 Introducción.....	27
2.2 Principales reglas del negocio	27
2.3 Modelo del negocio.....	28
2.4 Actores y trabajadores del negocio.....	28
2.5 Diagrama de los casos de uso del negocio.....	29
2.6 Descripción de los casos de uso del negocio	30
2.7 Diagrama de actividades por caso de uso del negocio.....	32
2.8 Especificación de los requerimientos de software	35
2.8.1 Requerimientos funcionales:	35
2.8.2 Requerimientos no funcionales	43
2.9 Modelo de casos de uso del sistema.....	46
2.11 Diagrama de casos de uso del sistema.....	48
2.12 Descripciones abreviadas de los casos de uso del sistema.....	49
2.13 Conclusiones.....	51
Capítulo 3: Análisis y diseño del sistema	52
3.1 Introducción.....	52
3.2 Estimación del esfuerzo	52
3.3 Modelo de análisis	59

3.4 Diagrama de clases de análisis.....	59
3.5 Modelo de diseño.....	61
3.6 Diagrama de clases del diseño	62
3.7 Modelo de datos.....	69
3.8 Diagrama de despliegue	70
3.9 Validación de la propuesta de solución	70
3.10 Conclusiones.....	72
CONCLUSIONES	73
RECOMENDACIONES.....	74
BIBLIOGRAFÍA REFERENCIADA.....	75
BIBLIOGRAFÍA CONSULTADA	77
GLOSARIO DE TÉRMINOS	79

INTRODUCCIÓN

Hoy en día el desarrollo del conocimiento y la capacidad de innovación tecnológica son determinantes en los resultados económicos y beneficios sociales de un país. La introducción masiva de las nuevas tecnologías de la información y las comunicaciones, conocidas por las siglas TIC, está influyendo fuertemente en la estructura y dinámica de los procesos económicos y sociales, redefiniendo aceleradamente las formas de producir, vender y competir en prácticamente todos los sectores productivos; también, en las nuevas formas de educar, de interacción y comunicación entre la gente y organismos de la sociedad, y de comunicar y controlar las acciones de gobierno.

Las TIC están transformando profundamente las economías domésticas y mundiales, pues permiten generar riqueza a distancia y en red, superando las fronteras geográficas y políticas. (Rosario, 2005)

Nadie puede permanecer al margen de este proceso revolucionario que supone la incorporación masiva de estas herramientas en la sociedad, puesto que los beneficios que aportan son por demás relevantes. Por el hecho de que las TIC atraviesan transversalmente a todos los sectores sociales y económicos, es que se considera estratégico su desarrollo y su introducción como herramienta e instrumento para promover un desarrollo económico-social sustentable. Esto exige una política de estado con acciones concretas para iniciar un desarrollo sostenido del sector científico-tecnológico de las TIC y contribuir a un crecimiento equitativo y estable del país.

El desarrollo de este campo no ha sido equitativo a escala global. Son varios los factores fundamentales que influyen en la capacidad de un país de acceso a las TIC: infraestructura, el conocimiento y la calidad. Se ha considerado que el hecho de disponer de una infraestructura limitada es el principal obstáculo para colmar la brecha digital. Por ello, principalmente los países desarrollados son los que poseen los mayores índices de explotación y desarrollo de las TIC.

En nuestro país se han introducido las TIC en casi todas las esferas del país, ya que constituyen pilares fundamentales para la modernización, efectividad y eficacia del Estado. En la UCI, se hace un uso intensivo de las mismas, numerosos procesos se han logrado automatizar mediante sitios web, plataformas, base de datos, etcétera. Posee un ambiente muy dinámico que necesita ser supervisado por varias direcciones encargadas de funcionar como un todo.

Gran parte de la información de estas direcciones debe ser protegida convenientemente, brindando solo acceso a quienes trabajan con ella. En la UCI existen varios departamentos que se encargan de dirigir diferentes áreas en la misma, cada una de ellas posee procedimientos, reglamentos, resoluciones y documentos de alta importancia para lograr un buen funcionamiento. La falta de un sistema seguro que permita la gestión documental de información de este tipo puede atentar contra la integridad y disponibilidad de la misma en todo momento. El desarrollo de un sistema contribuye al aumento de la seguridad en la gestión de la información que fluye a través de las direcciones de la UCI.

Entre las principales dificultades con que cuenta actualmente la UCI y que constituyen la **situación problemática** de la presente investigación se encuentran:

1. La UCI es un ambiente muy dinámico que necesita ser supervisado por varias direcciones encargadas de funcionar como un todo, actualmente se necesita un sistema seguro de almacenamiento único de la información, donde la misma se encuentre debidamente relacionada de tal forma que facilite su actualización.
2. Gran parte de la información de estas direcciones debe ser protegida convenientemente, brindando solo acceso a quienes trabajan con ella, evitando que su integridad y disponibilidad puedan ser modificadas.
3. Falta de un sistema seguro que permita la gestión documental rápida y eficiente de la información de este tipo.

Teniendo en cuenta lo anteriormente expuesto, el **problema científico** queda expresado de la siguiente manera:

¿Cómo transformar las necesidades del departamento de auditoría de la UCI en un modelo que facilite la digitalización de los procesos de control interno?

El **objeto de estudio** son los procesos del control interno, siendo el **campo de acción** el control interno en la UCI.

Por todas las dificultades planteadas anteriormente, la dirección de la universidad en vías de mejorar el proceso de control interno decide informatizar este sector para controlar que todas las actividades se realicen cumpliendo los procedimientos y normas fijados, con el objetivo de lograr un mejor control debido

a que toda la información será almacenada en un mismo lugar y accedida por las personas según sus privilegios. Además de que la información estará debidamente relacionada permitiendo brindar un mejor servicio así como colaborar y apoyar el trabajo que realiza el departamento de auditoría.

El **objetivo general** es: Realizar el análisis y diseño de una propuesta del sistema de control interno de la UCI.

Como **idea a defender** se plantea que el análisis y diseño de un sistema informático facilitará la implementación de una aplicación que digitalice la gestión de los procesos de control interno en la UCI.

Para cumplir con el objetivo trazado y resolver el problema planteado, se definen las siguientes tareas:

1. Estudio y descripción de los sistemas de gestión de la información referente al control interno en la UCI y fuera de ella.
2. Estudio de metodologías, herramientas y lenguaje de modelado para el diseño de sistemas informáticos.
3. Modelación y documentación del proceso de desarrollo de software mediante el uso de la metodología RUP, dejando evidencia de la evolución de la investigación.

En el desarrollo de la investigación se hará uso de un sistema de métodos conformado por métodos teóricos y métodos empíricos. En relación con los métodos teóricos se citan:

- Analítico-sintético: Se realizará un estudio al actual sistema y a los documentos, permitiendo la extracción de los elementos más importantes que se relacionan con el objeto de estudio.
- Análisis histórico-lógico: Se realizará un estudio del sistema actual para definir los principales errores y deficiencias en vista de perfeccionar las vulnerabilidades.

Se utilizarán los métodos empíricos siguientes:

- Encuestas y entrevistas: Se realizarán entrevistas a los usuarios de cada una de las diferentes direcciones de la institución, con el objetivo de conocer las principales deficiencias e identificar las funcionalidades que se requieren para erradicar las mismas.

- Observación: Se percibirá la realidad directamente, en vista a esto se eliminarán las vulnerabilidades.
- Estudio documental: Se realizará un estudio a partir de los documentos entregados por el cliente y otros que sean necesarios para estimular la auto superación en temas de interés para el desarrollo del sistema.

Como posible resultado de este trabajo se obtendrá la propuesta del diseño del Sistema de Control Interno de la UCI.

El trabajo está estructurado en 3 capítulos:

Capítulo 1. Fundamentación teórica

Incluye el estado del arte de los sistemas de gestión de información vinculados al objeto de estudio, una caracterización de las herramientas, lenguajes y metodologías más utilizadas para sistemas de este tipo, así como la selección de las tecnologías que se proponen para la realización de este trabajo.

Capítulo 2. Características del sistema

Describe la solución propuesta, muestra el modelo del negocio y el modelo de casos de uso del sistema, donde se recoge la especificación de los requisitos funcionales y no funcionales, definiéndose los actores y los casos de uso del sistema.

Capítulo 3. Análisis y Diseño

Aborda los detalles relacionados con la estimación del esfuerzo, el modelo de análisis y el modelo del diseño, representándose los diagramas de clases del diseño, el modelo de datos así como lo que debería ser la distribución del sistema mediante el diagrama de despliegue.

Capítulo 1: Fundamentación del tema

1.1 Introducción

En el presente capítulo se expone qué es el control interno y se detallan los componentes que intervienen en este proceso. Se identifican las principales deficiencias que existen y que motivaron esta investigación, se estudia una herramienta argentina para la implantación del control interno llamada MEYCOR COSO. Se realiza la selección de las tecnologías actuales que se adapten a las necesidades del sistema propuesto, identificando los lenguajes de programación a utilizar, el sistema gestor de base de datos y la metodología con las cuales se debe desarrollar el Sistema de Control interno para la UCI.

1.2 Control Interno

El control interno ha carecido durante muchos años de un marco referencial común, generando expectativas diferentes entre empresarios y profesionales. Sin estar claramente definido se utiliza en leyes, normas o reglamentos. Para el desarrollo del presente capítulo se tomará como base el llamado Informe COSO (Committee of Sponsoring Organization) “La Comisión de Organizaciones Patrocinadoras de la Comisión Treadway”, publicado en EE.UU. en 1992, surgió como una respuesta a las inquietudes que planteaban la diversidad de conceptos, definiciones e interpretaciones existentes en torno a la temática referida.

Entre los miembros de COSO se cuentan el Instituto Americano de Contadores Públicos (AICPA), el Instituto de Ejecutivos Financieros (FEI), Instituto de Auditores Internos (IIA) y la Asociación Americana de Contabilidad (AAA, que agrupa a profesores universitarios de contabilidad), cuyas instituciones mencionadas gozan de reconocido prestigio y tienen relación directa con el Control interno. En 1992, la publicación del Informe COSO proporcionó, por primera vez, un estándar para diseñar, implementar y medir, sistemas de control interno. Las primeras empresas en adoptarlo fueron las norteamericanas, para luego, extenderse en el resto del mundo.

El Informe COSO internacionalmente se considera hoy en día como un punto de referencia obligado cuando se tratan materias de control interno, tanto en la práctica de las empresas como en los niveles

legislativos y docentes. Cuenta con un objetivo primordial que es establecer una definición de control interno y un desarrollo de los conceptos lo más claro posible, de forma que cuando se hable de control interno todo el mundo esté hablando en un lenguaje unificado. Las definiciones del control interno, el contenido de sus componentes y sus normas, se ponen en vigor en la Resolución 297/2003 del Ministerio de finanzas y precios de nuestro país.

En Colombia se ha institucionalizado la aplicación del control interno en las entidades y organismos del estado mediante la ley 87 de 1993, esta ley establece que los objetivos que busca la administración se cumplan en aras del bienestar común buscando una adecuada gestión y control, conforme a las características en el informe COSO para su adecuado diseño, interpretación, implementación y evaluación.

En el mundo se le ha prestado una gran atención a la necesidad de elevar las exigencias en los controles internos, a partir de la década del 70, por el descubrimiento de muchos pagos ilegales, malversaciones y otras prácticas delictivas en los negocios, por lo que terceras personas quisieron conocer por qué los controles internos de las entidades no habían prevenido a la administración de la ocurrencia de esas ilegalidades. El control interno ha sido preocupación de la mayoría de las entidades, aunque con diferentes enfoques y terminologías, lo cual se puede evidenciar al consultar los libros de texto de auditoría, los artículos publicados por organizaciones profesionales, universidades y autores individuales.

Por lo expuesto anteriormente se considera de vital necesidad abordar el control interno desde su punto de vista conceptual así como su importancia, objetivos, normativas y componentes.

El año 2003 se puede definir como el año del control en Cuba ya que se emitieron por los Ministerios de auditoría y control y de finanzas y precios, disposiciones que tienen como fin, el obligatorio cumplimiento del control de todos los recursos que el estado pone a disposición de las entidades para la creación de bienes y la satisfacción de las necesidades de la sociedad.

La apertura de nuestro país al mercado internacional hace que nuestra economía se tenga que adecuar a las nuevas exigencias en que se está desarrollando, es por ello que la Resolución Económica del V Congreso del PCC plantea que “bajo las condiciones en que opera la economía actual más vinculada a las exigencias de la competencia internacional, el control oportuno y eficaz de la actividad económica es esencial para la dirección”. “Se precisa un proceso de transformaciones del sistema empresarial para la

implantación de fuertes restricciones financieras que hagan que el control del uso eficiente de los recursos sea interno al mecanismo de gestión y no dependa únicamente de comprobaciones externas” (Ministerio de Finanzas y Precios, Resolución No. 297, 2003).

La Resolución No. 297/03 del Ministerio de Finanzas y Precios, revoluciona el concepto de control interno ya que extiende su aplicación a todas las actividades que se desarrollan en las organizaciones proyectándolo a través de cinco componentes:

Componentes de control interno:

- Ambiente de control.
- Evaluación de riesgos.
- Actividades de control.
- Información y comunicación.
- Supervisión o monitoreo.

Ambiente de control

Constituye el andamiaje para el desarrollo de las acciones y refleja la actitud asumida por la alta dirección en relación con la importancia del control interno y su incidencia sobre las actividades de la entidad y resultados, por lo que debe tener presente todas las disposiciones, políticas y regulaciones que se consideren necesarias para su implantación y desarrollo exitoso. El ambiente de control fija el tono de la organización al influir en la conciencia del personal, puede considerarse como la base de los demás componentes del control interno.

Evaluación de riesgos

A través de la investigación y análisis de los riesgos relevantes y el punto hasta el cual el control vigente los neutraliza, se evalúa la vulnerabilidad del sistema. Para ello debe adquirirse un conocimiento práctico de la entidad y sus componentes como manera de identificar los puntos débiles, enfocando los riesgos

tanto de la entidad (internos y externos) como de la actividad. Los objetivos de control deben ser específicos, así como adecuados, completos, razonables e integrados a los globales de la institución.

Actividades de control

Las actividades de control son procedimientos que ayudan a asegurarse que las políticas de la dirección se llevan a cabo, y deben estar relacionadas con los riesgos que ha determinado y asume la dirección. Se ejecutan en todos los niveles de la organización y en cada una de las etapas de la gestión, partiendo de la elaboración de un mapa de riesgos, conociendo los riesgos se disponen los controles destinados a evitarlos o minimizarlos.

Información y comunicación

La información relevante debe ser captada, procesada y transmitida de tal modo que llegue a todos los sectores y permita asumir las responsabilidades individuales. Las personas deben conocer en tiempo, las cuestiones relativas a su responsabilidad de gestión y control. Cada función debe especificarse con claridad. Los informes deben transmitirse adecuadamente a través de una comunicación eficaz, incluyendo una circulación multidireccional de la información. La existencia de líneas abiertas de comunicación y una clara voluntad de escuchar, por parte de los dirigentes, resultan vitales.

Supervisión o monitoreo

Es el proceso que evalúa la calidad del control interno en el tiempo. Es importante monitorear el control interno para determinar si el mismo está operando en la forma esperada y si es necesario hacer modificaciones. Las actividades de monitoreo permanente incluyen actividades de supervisión realizadas de forma permanente, directamente por las distintas estructuras de dirección.

La supervisión es un proceso mediante el cual una persona procesadora de un caudal conocimientos y experiencias, asume la responsabilidad de dirigir a otras para obtener resultados que les son comunes. Supervisar efectivamente requiere, planificar, organizar, dirigir, ejecutar retroalimentar constantemente. Exige constancia, dedicación, perseverancia, siendo necesario poseer características individuales en la persona que cumple esta misión.

Pirámide

El Informe grafica estos componentes en base a figuras geométricas: La pirámide (muestra la estructura del sistema de control interno y de esta manera se puede observar cuál es la base funcional del sistema).



Figura 1 Descripción visual de la estructura del Control Interno

“El Control Interno como proceso clave para el logro de los objetivos de la empresa”, Sergio García Y Roberto de Simone, decano y director de la carrera de contador público de la FCE (Facultad de Ciencias Económicas) y USAL (Universidad del Salvador) respectivamente, eleve 2008.

Objetivos del control interno

En Cuba se ha desarrollado un constante y sostenido esfuerzo por consolidar el control interno en las diferentes entidades a partir del estudio de sus características. Se debe establecer sus acciones y medidas de control y deben cumplirse por todas las personas involucradas y responsabilizadas con su funcionamiento. (Ministerio de finanzas y precios, Resolución No. 297, 2003).

El **control interno** es 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 (Ministerio de finanzas y precios, Resolución No. 297, 2003):

- Confiabilidad de la información.

- Eficiencia y eficacia de las operaciones.
- Cumplimiento de las leyes, reglamentos y políticas establecidas.
- Control de los recursos de todo tipo a disposición de la entidad.

Manual de procedimientos

El manual de procedimientos es un componente del sistema de control interno, el cual se crea para obtener una información detallada, ordenada, sistemática e integral que contiene todas las instrucciones, responsabilidades e información sobre políticas, funciones, sistemas y procedimientos de las distintas operaciones o actividades que se realizan en una organización. Deben cumplir con los criterios establecidos en las normas generales de control interno.

Tipos de manuales

- Organización.
- Identidad corporativa.
- Operaciones.
- Auditoría.
- Analítico.
- Papelería institucional.
- Forma de cada procedimiento.
- Contenido de cada procedimiento.

Manual de organización

En él se precisan las funciones encomendadas a cada unidad. Sirve como orientación al personal de nuevo ingreso, para facilitar su incorporación a las diferentes unidades organizativas y propicia el mejor aprovechamiento de los recursos humanos y materiales en la organización. Define el flujo de información

que permite la aplicación del sistema de información y comunicación. Permite realizar la tarea principal de un directivo: organizar, delegar, supervisar y motivar al personal.

Manual de identidad o imagen corporativa

Es la recopilación de la Imagen y la Identidad corporativa, el cual se encarga de facilitar a nivel global todo lo concerniente a la empresa u organización.

Manual de operaciones

Es el documento que contiene la descripción de actividades que deben seguirse en la realización de las funciones de una unidad administrativa. Incluye además los puestos o unidades administrativas que intervienen precisando su responsabilidad y participación.

Manual de auditoría

Es una guía escrita que orientará y organizará el trabajo del auditor interno en aras de que contribuya al logro de los objetivos institucionales. Contiene las tareas a desarrollar en cada etapa del trabajo, incluye una metodología para la identificación, clasificación y cuantificación de los riesgos elaborada a partir de las exigencias que se establecen en la Resolución 297/03 sobre control interno.

Manual analítico

Contiene los Indicadores estadísticos y analíticos, el control interno contable y administrativo, así como los objetivos de control para la información y tecnologías afines.

Manual de papelería institucional

Contiene todo el sistema informativo interno, para la dirección, con el organismo superior y con terceros. Controla el sistema de la información y los canales de comunicación.

Manual de forma de cada procedimiento

Establece la pauta de Identidad corporativa. Identifica los títulos, códigos del manual, grupo, subgrupo y anexo y es donde se fecha la actualización.

Manual de contenido de cada procedimiento

Contiene la base legislativa aplicable y su objetivo. Describe los procesos de trabajo, las funciones y responsabilidades. Contempla las cualidades de la información implícitas.

Contraloría general

La Contraloría general de la república norma el marco general conceptual del sistema de control interno y a partir del mismo, los jefes de los órganos, organismos y entidades nacionales y los dirigentes a ellos subordinados, lo diseñan e implementan de acuerdo con los principios establecidos en la ley y con las particularidades de su sistema (Ley 107, Artículo 83).

Las direcciones y departamentos de la contraloría general tienen en su esfera de competencia entre las funciones y atribuciones específicas, las siguientes:

Departamento independiente de auditoría interna (Ley 107, Artículo 128).

- Asesorar el diseño e implementación del sistema de control interno y comprobar su efectividad.
- Ejecutar la auditoría interna en las estructuras organizativas de la contraloría general de la república, de acuerdo con el plan aprobado.
- Cualquier otra que deba entenderse referida a su actuación.

Es evidente la importancia que reviste para la recuperación de la economía el control de los recursos materiales, financieros y humanos. La utilización del control interno en las entidades permite que se reduzcan en gran magnitud las ilegalidades y corrupción dentro del marco laboral. Con la aplicación de los sistemas de control en todos sus aspectos se podrá obtener resultados de eficiencia, vigencia y economía en la gestión empresarial.

La UCI está llevando a cabo este proceso para lograr lo anteriormente expuesto por lo que es de vital importancia proporcionar con este trabajo una propuesta de diseño para una mayor organización y confiabilidad de los documentos de uso legal que rigen el sistema de trabajo de la entidad.

1.3 Sistemas vinculados al campo de acción

MEYCOR COSO AG, Versión: 2.0 / Año: 2009

Durante la investigación se descubrió la existencia de una herramienta en Uruguay para la implantación del control interno basado en informe COSO:

El software MEYCOR COSO AG ha sido desarrollado por DATASEC. El producto permite realizar una evaluación del control interno según el informe COSO, evaluando riesgos y auditando dichas evaluaciones. Está compuesto por dos módulos: MEYCOR COSO AG – EVAL y MEYCOR COSOAG – AUDIT. El módulo MEYCOR COSO AG - EVAL constituye la parte de recopilación de información de la organización, así como también de la evaluación del control interno y los riesgos. El producto permite ingresar el organigrama de la organización, sus procesos de negocio, sus objetivos de negocio y las tareas que se realizan para llevar a cabo cada proceso.



Figura 2 Logotipo del Sistema MEYCOR COSO AG

Software para control interno y administración de activos fijos

Durante el estudio de los sistemas vinculados al campo de acción se descubrió este software Mexicano que se encarga de almacenar una base de datos con toda la información de los bienes de su empresa. Sus características permiten tener informes inmediatos en cualquier momento y obtener datos históricos

sobre las asignaciones de cada activo, además de llevar a cabo de ser necesario, la depreciación fiscal de cada uno.

Este proceso puede ser mensual, trimestral, semestral o anual. La clasificación de sus activos en el sistema puede ser tan detallada como sea necesario, haciendo más fácil el control en diversos tipos de compañías. El proceso completo de controlar bienes y activos solo se reducirá a poner una etiqueta de identificación y periódicamente leer un código de barras, una tarea que le ahorrara tiempo y recursos valiosos a su compañía.

1.4 Tecnologías actuales

Hoy día existen numerosas empresas e instituciones desarrolladoras de software las cuales han dado lugar a nuevas tecnologías para el desarrollo web, así como a herramientas, sistemas gestores de base de datos y lenguajes de programación.

1.4.1 Framework para el desarrollo Web

Zend Framework: Zend Framework (ZF), es un framework de código abierto para desarrollar aplicaciones Web y servicios web con PHP5. Es una implementación que usa código 100% orientado a objetos. La estructura de los componentes de ZF es algo único; cada componente está construido con una baja dependencia de otros componentes. Trabaja en 3 capas, usando el Modelo Vista Controlador. Cuenta con módulos para manejar archivos en formato de documento portátil, canales de sindicación de noticias y servicios web. Incluye objetos de las diferentes bases de datos, por lo que es extremadamente simple para consultar la base de datos. Completa documentación y pruebas de alta calidad. Robustas clases para autenticación y filtrado de entrada.

1.4.2 Lenguajes de programación para el desarrollo Web

Lenguajes de programación del lado servidor

PHP: Fue originalmente diseñado en Perl. Es un lenguaje multiplataforma completamente orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una Base de Datos. El código fuente escrito en PHP es invisible al navegador y al cliente. Esto hace que la programación en PHP sea segura y confiable. Posee una amplia documentación en su página oficial Wikipedia 2006.

Se presenta como una alternativa de fácil acceso para todos. Permite aplicar técnicas de programación orientada a objetos. No requiere definición de tipos de variables y posee manejo de excepciones (desde PHP5). El programador puede aplicar en su trabajo cualquier técnica de programación y/o desarrollo que le permita escribir código ordenado, estructurado y manejable.

JSP: Es una tecnología Java que permite generar contenido dinámico para web, en forma de documentos HTML, XML o de otro tipo. Es un lenguaje para la creación de sitios web dinámicos, acrónimo de Java Server Pages. JSP es un lenguaje multiplataforma. Creado para ejecutarse del lado del servidor.

JSP fue desarrollado por Sun Microsystems para la creación de aplicaciones web potentes. Posee un motor de páginas basado en los servlets de Java. Para su funcionamiento se necesita tener instalado un servidor Tomcat.

Lenguajes de programación del lado cliente

Existen diversas tecnologías del lado del cliente que permiten crear sitios según los estándares establecidos, los cuales ayudan a la presentación de los documentos y validaciones del contenido.

JavaScript: Es el lenguaje de programación Web del lado del cliente más extendido, fue creado por Brendan Eich en la empresa Netscape Communications. Con este lenguaje script se pueden generar páginas dinámicamente en función de las preferencias del usuario, validar los datos introducidos en un formulario o modificar dinámicamente el contenido de la página. Fue creado por Netscape para su navegador 2.0 con la versión 1.0 de JavaScript y ya marcha por la 1.5, está basado en objetos.

Por cuestiones de seguridad se ha impedido que JavaScript pueda leer y escribir a disco, ejecutar otros programas y conectarse a otra computadora o enviar correo, producto que pudiera introducirse virus de esta forma cuando un servidor envía una página a un cliente. De la misma forma en el navegador se puede configurar para no ejecutar código script.

HTML: HTML (HyperText Markup Language) es un lenguaje muy sencillo de fácil aprendizaje que permite describir hipertexto, es decir, texto presentado de forma estructurada y agradable, con enlaces que conducen a otros documentos o fuentes de información relacionadas, y con inserciones multimedia (gráficos, sonido).

Cualquier navegador web, independientemente del sistema operativo, puede leer o interpretar una página web, lo admiten todos los exploradores. Sin embargo, el aspecto de dichas páginas depende del tipo de navegador web, del monitor, la velocidad de la conexión de Internet y, por último del navegador. Aunque no todos los navegadores muestran una misma página web de la misma forma.

1.4.3 Fundamentación del lenguaje de programación a utilizar

Se decide PHP como lenguaje luego de realizar el estudio, donde a partir de sus características este lenguaje libre se convierte en la herramienta ideal para la creación de páginas Web dinámicas, se aplica a la propuesta de solución ya que presenta gran velocidad a la hora de procesar los datos, puede operar en la mayoría de los sistemas operativos (GNU/Linux, MacOS, Windows), puede alojarse en cualquier servidor, gestión de memoria automática, rápido, fácil de aprender y por la integración de trabajo con la bases de datos de PostgreSQL. Posee programación orientada a objeto y se cuenta con el Zend Framework que ofrece un gran rendimiento y una robusta implementación Modelo Vista Controlador.

1.4.4 Apache

El servidor HTTP Apache es un software servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Windows, Macintosh y otras que implementa el protocolo HTTP/1.1 y la noción de sitio virtual. Apache presenta entre otras características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude a su configuración. Debido a que la mayor parte de su configuración se realiza en el fichero apache2.conf o httpd.conf según el sistema donde esté corriendo, cualquier cambio en este archivo requiere reiniciar el servidor o forzar la lectura de los archivos de configuración nuevamente.

Algunas de sus ventajas son:

- Modular
- Código abierto
- Multi-plataforma
- Extensible
- Popular (fácil conseguir ayuda/suporte)

La arquitectura del servidor Apache es muy modular. El servidor consta de una sección core y diversos módulos que aportan mucha de la funcionalidad que podría considerarse básica para un servidor web.

1.4.5 Navegadores

Un Navegador o “Browser” esta compuesto por una serie de estructuras programáticas que le permiten desplegar y ejecutar documentos e instrucciones que son cargados en él, este tipo de instrucciones pueden variar desde documentos estáticos(HTML), contenido dinámico(Java script), contenido gráfico(Flash), hasta otro ambiente de navegación por separado al “Browser”.

Uno de los navegadores más destacados por sus características es Mozilla Firefox, que es un navegador de código abierto, multiplataforma. Se ha convertido en el principal foco de desarrollo de la Fundación Mozilla junto con el cliente de correo electrónico y lector de noticias, es por eso que fue escogido para la navegación de la presente investigación.

Entre las Principales opciones de Firefox, se encuentra la utilización de las pestañas, que permiten abrir varias páginas al mismo tiempo, en una misma ventana, hacen que la navegación sea más agradable y parezca más rápida.

Una de las innovaciones del navegador es que contiene dos barras de búsqueda. La de Google y una barra de búsqueda local que puede ocultar.

Las direcciones de sus páginas web favoritas se pueden guardar en marcadores. Además, cada usuario de Firefox puede elaborar su propia lista de marcadores y conservar un historial de las páginas visitadas, creando un perfil personal.

Con el navegador se ha aumentado la seguridad, ya que no permite la instalación automática de programas sin su conocimiento o permiso. Cuando se requiere de un programa adicional para visualizar animaciones o para ejecutar una determinada acción en una página, Firefox le informa de ello y le guía en la instalación de este programa llamado pluing.

1.4.6 Lenguaje para la modelación

El lenguaje de modelado pretende dar apoyo a la mayoría de los procesos de desarrollo de software, son desarrollados con el esfuerzo de simplificar y consolidar el gran número de métodos de desarrollo que han surgido.

UML (Lenguaje Unificado de Modelado)

Es un lenguaje grafico que le ayuda a especificar, visualizar y documentar modelos de sistemas de software, incluida su estructura y diseño, de manera que cumple con todos estos requisitos.

UML es un lenguaje más expresivo, claro y uniforme que otros anteriores que se utilizan para el diseño Orientado a Objetos, permite una nueva y fuerte integración entre las herramientas, los procesos y los dominios.

Diagramas de estructura: Incluyen el diagrama de clases, diagrama de objetos, diagrama de componentes, diagrama de estructura compuesta, diagrama de paquetes, y diagrama de despliegue.

Diagramas de comportamiento: Incluyen el diagrama de casos de uso (algunos métodos utilizados por los requisitos durante la recolección), diagrama de actividad, diagrama de la máquina y el estado.

Diagramas de interacción: Son un subtipo de diagramas de comportamiento, que enfatiza sobre el flujo de control y de datos entre los elementos del sistema modelado, incluyen diagrama de secuencia y diagrama de colaboración.

1.4.7 Fundamentación del lenguaje para la modelación a utilizar

Se selecciona UML para la modelación porque UML es una notación de entendimiento común para todo desarrollador, es vital para describir planos de software, definiendo un conjunto de reglas para representar modelos. Da la posibilidad de permitir documentar y especificar los elementos creados mediante un lenguaje común describiendo modelos. UML se ha convertido en lenguaje estándar para la realización de software.

1.4.8 Sistemas gestores de base de datos

Los sistemas gestores de base de datos son un paquete generalizado de software, que se ejecuta en un sistema computacional anfitrión, centralizando los accesos a los datos y actuando de interfaz entre los datos físicos y el usuario. Permiten la facilidad de manejo de grandes volúmenes de información alcanzando gran velocidad en breve tiempo y aporta independencia del tratamiento de información y seguridad de la información.

PostgreSQL: Es un servidor de base de datos objeto relacional libre, es un motor de base de datos avanzado de código abierto. Es un sistema objeto-relacional, incluye características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional.

Presenta las características esenciales de escalabilidad permitiendo gran cantidad de conexiones simultáneas y gran control de concurrencia al no permitir bloqueos innecesarios. PostgreSQL liberado bajo la licencia BSD, como muchos otros proyectos Open Source, no es manejado por una sola compañía sino que es dirigido por una comunidad de desarrolladores y organizaciones comerciales.

MySQL: Es un sistema de administración de bases de datos (Database Management System) para bases de datos relacionales. Es la base de datos de código fuente abierto más usada del mundo. Presenta condición de open source de MySQL, que hace que su utilización sea gratuita e incluso se pueda modificar con total libertad, pudiendo descargar su código fuente. Esto ha favorecido muy positivamente en su desarrollo y continuas actualizaciones, para hacer de MySQL una de las herramientas más utilizadas por los programadores orientados a Internet.

Utiliza múltiples tablas para almacenar y organizar la información. Posee una adaptación a diferentes entornos de desarrollo, permitiendo su interacción con los lenguajes de programación más utilizados como PHP, Perl y Java y su integración en distintos sistemas operativos.

1.4.9 Fundamentación del Sistema Gestor de Base de Datos a utilizar

Se decide **PostgreSQL** porque presenta es un sistema objeto-relacional, ya que incluye características de la orientación a objetos, entre sus principales ventajas se destacan los ahorros considerables en costos de operación, su estabilidad y confiabilidad legendarias, es extensible y multiplataforma. Constituye una

herramienta muy potente que permite grandes volúmenes de información, además de ser uno de los gestores de base de datos más utilizados en la UCI.

1.4.10 Metodología para el desarrollo del software

En los últimos años la tendencia en el desarrollo web ha tenido gran auge, en la mayoría de casos sin una adecuada sistematización de sus procesos. Por esta razón es necesario ser metódicos para lograr un producto con calidad.

Las metodologías para el desarrollo de software surgen con la idea de mejorar la productividad en el desarrollo y la calidad de los productos. Las mismas guían el proceso de desarrollo y conduce al programador a desarrollar un buen sistema de software.

Proceso Unificado Racional (RUP)

RUP es un proceso de desarrollo de software que en conjunto con el Lenguaje Unificado de Modelado (UML) brindan una mayor facilidad a la hora de analizar, implementar y documentar sistemas orientados a objetos. Es un conjunto de metodologías adaptables al contexto y necesidades de cada organización. Sus tres características esenciales son: está dirigido por los Casos de Uso, está centrado en la arquitectura y es iterativo e incremental. Está compuesto por 4 fases, en las cuales se realizan varias iteraciones, y en cada iteración se realizan diferentes esfuerzos en diferentes actividades. (Jacobson, y otros, 2000)

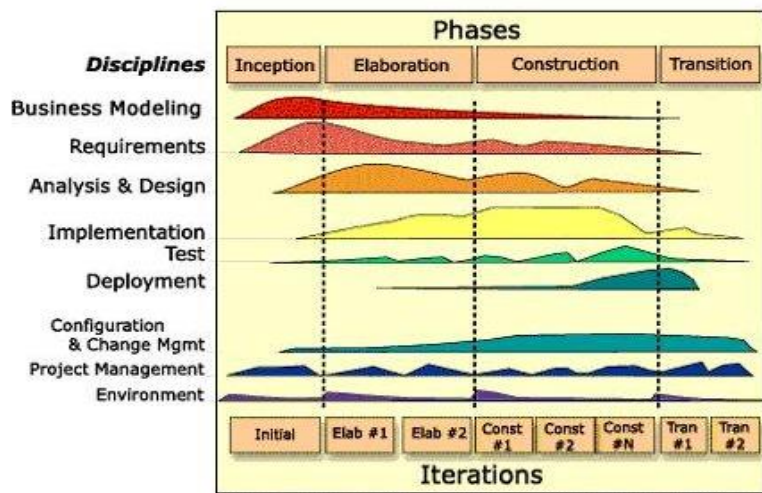


Figura 3 Esquema que representa el ciclo de desarrollo de software definido por RUP

Fases de RUP

- **Inicio:** se hace un plan de fases, se indican los principales casos de uso y se identifican los riesgos. Se define el alcance del proyecto.
- **Elaboración:** se hace un plan de proyecto, se completan los casos de uso y se eliminan los riesgos.
- **Construcción:** se concentra en la elaboración de un producto totalmente operativo y eficiente y el manual de usuario.
- **Transición:** se instala el producto en el cliente y se entrena a los usuarios. Como consecuencia de esto suelen surgir nuevos requerimientos a ser insertados.

1.4.11 Fundamentación de la metodología a utilizar

Se decidió utilizar RUP porque esta metodología mantiene al equipo enfocado en producir incrementalmente software operativo a tiempo, con las características requeridas y con la calidad requerida. Posee mucha documentación, muy organizativo basado en roles. Por todo lo antes dicho y teniendo como base lo expuesto anteriormente se puede concluir que este proceso de desarrollo es el idóneo para llevar a cabo la realización de este software.

1.4.12 Herramientas de modelado visual

Para el modelado visual se utilizan herramientas que ayudan a la construcción de aplicaciones, estas sirven de apoyo para la realización de estos modelos, y por tanto, agilizan la elaboración de aplicaciones consistentes y duraderas.

Visual Paradigm: Es una herramienta UML CASE visual que ayuda a construir aplicaciones rápidamente, mejor y económicamente Visual Paradigm para UML es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. (FREED, 2007). Visual Paradigm es multiplataforma, utiliza UML como lenguaje de modelado, ayudando de una manera más rápida a la construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código

desde diagramas y generar documentación. Esta herramienta también proporciona una mejor interfaz gráfica de usuario y una mayor base de datos de esquema de apoyo, demostraciones interactivas de UML y proyectos UML, soporta generación de código y soporte de ingeniería inversa.

Rational Rose: Es la herramienta CASE desarrollada por los creadores de UML (Booch, Rumbaugh y Jacobson), que cubre todo el ciclo de vida de un proyecto. Facilita el desarrollo de un proceso cooperativo en el que todos los agentes tienen sus propias vistas de información (vista de Casos de Uso, vista Lógica, vista de Componentes y vista de Despliegue), pero utilizan un lenguaje común para comprender y comunicar la estructura y la funcionalidad del sistema en construcción. Rational Rose es una herramienta con plataforma independiente que ayuda a la comunicación entre los miembros de equipo, a la monitorización del tiempo de desarrollo y al entendimiento del entorno de los sistemas.

1.4.13 Fundamentación de la herramienta de modelado visual a utilizar

Se decide utilizar Visual Paradigm como herramienta para el modelado de la propuesta de solución ya que es un producto de calidad que soporta aplicaciones web, es fácil de utilizar y de actualizar, posee compatibilidad entre ediciones, así como licencia gratuita y comercial. Visual Paradigm para UML apoya un conjunto de idiomas tanto en la generación del código, y la ingeniería inversa en Java, C ++, PHP, XML. Es una herramienta muy usada en los proyectos de la UCI para modelar propuestas de solución.

1.5 Patrones

El Patrón es una pareja de problema/solución con un nombre y que es aplicable a otros contextos, con una sugerencia sobre la manera de usarlo en situaciones nuevas.

1.5.1 Patrones de diseño

Los patrones son empleados en los diagramas de clases de diseño para refinar el diseño y hacerlo de forma eficiente. Un patrón es un modelo que podemos seguir para realizar algo. Surgen de la experiencia de seres humanos de tratar de lograr ciertos objetivos capturando la experiencia existente y probada para promover buenas prácticas. Los patrones de diseño son soluciones simples y elegantes a problemas específicos y comunes del diseño orientado a objetos. Son soluciones basadas en la experiencia y que se ha demostrado que funcionan.

Patrones GRASP: Los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones. GRASP es un acrónimo que significa patrones generales de software para asignar responsabilidades. A continuación se explican algunos de ellos:

Experto: Experto es un patrón que se usa más que cualquier otro para asignar responsabilidades; es un principio básico que suele utilizarse en el diseño orientado a objetos. Con él no se pretende designar una idea oscura ni extraña; expresa simplemente la " intuición " de que los objetos hacen cosas relacionadas con la información que poseen. (Larman, 2004).

Creador: El patrón Creador guía la asignación de responsabilidades relacionadas con la creación de objetos, tarea muy frecuente en los sistemas orientados a objetos. El propósito fundamental de este patrón es encontrar un creador que se conecte con el objeto producido en cualquier evento. (Larman, 2004).

Alta Cohesión: Es un principio de que se debe tener presente en todas las decisiones de diseño, es la meta principal que ha de buscarse en todo momento. Es un patrón evaluativo que el desarrollador aplica al valorar sus decisiones de diseño. (Larman, 2004).

Bajo acoplamiento: El Bajo Acoplamiento estimula asignar una responsabilidad de modo que su colocación no incremente el acoplamiento tanto que produzca los resultados negativos propios de un alto acoplamiento, soporta el diseño de clases más independientes que reducen el impacto de los cambios, y también más reutilizables, que acrecientan la oportunidad de una mayor productividad. (Larman, 2004).

Controlador: Un Controlador es un objeto de interfaz no destinada al usuario que se encarga de manejar un evento del sistema. Define además el método de su operación. Las operaciones del sistema deberían manejarse en la capa de dominio de los objetos y no en la interfaz, presentación o aplicación. (Larman, 2004).

1.5.2 Patrones de Caso de Uso

Los Patrones de Caso de Uso capturan técnicas para que el modelo sea mantenible, reusable y entendible, por lo que capturan mejores prácticas para modelar casos de uso. (Cuesta, 2005).

Beneficios

- Aumentar la productividad.
- Reutilizar elementos existentes (en este caso fragmentos de modelos).
- Evitar el retrabajo por errores.
- No invertir tiempo en resolver problemas ya resueltos.
- Aplicar la teoría al trabajo práctico.
- Habilitar las herramientas de soporte para modelar el desarrollo. (Cuesta, 2005).

Otros Patrones de Caso de Uso

- Reglas de Negocio.
- Login.
- Reportes y Explotación de Información.
- Inclusión y Extensión.
- Sistema en Capa.
- Múltiples Actores
- Jerarquía de Comportamiento
- Servicios Opcionales.

Patrón CRUD

El Patrón CRUD Completo consiste en un Caso de Uso para administrar la información (CRUD Información), permite modelar las diferentes operaciones para administrar una entidad de información, tales como crear, leer, cambiar y dar de baja. (Cuesta, 2005).

1.5.3 Patrón de Arquitectura

Patrón Modelo Vista Controlador (MVC)

MVC es un patrón de arquitectura utilizado en sistemas Web para separar los datos de una aplicación, la interfaz de usuario y la lógica de control en tres componentes distintos, permitiendo flexibilidad y facilidad a la hora de hacer futuros cambios.

Vista: Es la información presentada al usuario. Una vista puede ser una página Web o una parte de una página.

Controlador: Actúa como intermediario entre el Modelo, la Vista y cualquier otro recurso necesario para generar una página, es el objeto que proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el Modelo.

Modelo: Representa las estructuras de datos. Típicamente el modelo de clases contendrá funciones para consultar, insertar y actualizar información de la base de datos.

1.6 Técnicas empleadas para la captura de los requisitos

La captura de requisitos es la actividad mediante la que el equipo de desarrollo de un sistema de software extrae de cualquier fuente de información disponible, las necesidades que debe cubrir dicho sistema. El proceso de captura de requisitos puede resultar complejo, principalmente si el entorno de trabajo es desconocido para el equipo de analistas, y depende mucho de las personas que participen en él. Por la complejidad que todo esto puede implicar, la ingeniería de requisitos ha trabajado desde hace años en desarrollar técnicas que permitan hacer este proceso de una forma más eficiente y precisa.

A continuación se presentan un grupo de técnicas que han sido utilizadas para esta actividad en el proceso de desarrollo de todo tipo de software.

La Entrevista es una técnica para recopilar información a partir de un intercambio directo entre personas o grupos (López, 2004). Esta fue la primera técnica empleada para la captura de los requisitos con los que debía cumplir el módulo Otras actividades de Educación y Prevención Jurídica.

La investigación documental consiste en el estudio de documentos escritos sobre un objeto determinado, es decir, son todos aquellos documentos registrados en diferentes dispositivos físicos a los que podemos tener acceso en forma directa o indirecta para su consulta (González). Esta técnica se utilizó para estudiar y analizar las descripciones del negocio hechas por los especialistas.

Brainstorming (Tormenta de ideas) es también una técnica de reuniones en grupo cuyo objetivo es que los participantes muestren sus ideas de forma libre. Consiste en la mera acumulación de ideas y/o información sin evaluar las mismas. (Escalona, y otros, 2002). Para el empleo de esta técnica se reunieron los integrantes del equipo de trabajo, por varios días, donde surgieron ideas renovadoras, lo que permitió optimizar al máximo el trabajo a desarrollar.

1.7 Conclusiones

En el presente capítulo se propone el desarrollo del análisis y diseño de un sistema de control interno para la UCI que tendrá PHP como lenguaje de programación el cual puede trabajar sobre Windows o cualquier servidor, PostgreSQL como gestor de base de datos (BD), Apache como servidor Web, Rational Unified Process (RUP) como metodología de desarrollo donde se despliegan proyectos grandes, RUP posee una aproximación iterativa ayudando a mitigar los riesgos en forma temprana, es adaptable a las necesidades del sistema además permite seleccionar e implantar los componentes específicos de proceso necesarios para facilitar un sistema consistente. Visual Paradigm como herramienta para el modelado visual, Lenguaje Unificado de Modelado (UML) como lenguaje de modelado de sistemas de Software que permitirá la creación de un conjunto de artefactos que servirán en un futuro para la implementación exitosa del sistema. Por estas razones se propone el desarrollo de un sistema que, minimice el esfuerzo del grupo de trabajo del Departamento de Auditoría, una mejor organización y permita detectar los problemas que impiden una buena documentación y monitoreo del proceso.

Capítulo 2: Descripción de la solución propuesta

2.1 Introducción

En el presente capítulo se presentan las características del sistema que brindarán la solución para el problema propuesto, así como una descripción general de cómo debe funcionar el mismo. Se hace un análisis de los procesos del negocio, mediante el modelo de dominio y una descripción de lo que el sistema debe hacer, para la cual se identifican las funcionalidades requeridas y las restricciones que se imponen a través de los requisitos funcionales y no funcionales del sistema que se debe desarrollar. Además se definen los actores y casos de uso del mismo, con las correspondientes especificaciones de los casos de uso.

2.2 Principales reglas del negocio

Reglas de Estructura

- Los procesos involucrados con el Control Interno de la universidad lo realiza el Departamento de Auditoría de la UCI.
- Interesados: Jefe de Expertos, Departamento de Auditoría, Rector, Auditor.
- El informe general culmina el proceso de gestión de riesgos.

Reglas de Restricción

- El Rector es la única persona encargada de solicitar la inspección a un área determinada.
- El Jefe de Expertos solo puede ser asignado a un usuario del sistema.
- Un auditor solo puede evaluar un área de la universidad.

Reglas de Derivación

- Cuando un auditor no realiza su trabajo completo se considera que este incumplió con el mismo.

- Cuando un auditor deja trabajo por hacer se considera que el trabajo es trabajo pendiente.
- Cuando un auditor realiza su trabajo completo se considera que cumplió.

Reglas de Acción

- El rector solicita la evaluación del Control Interno de un área determinada.
- Un área que se encuentra en proceso de Auditoría pasa por varias etapas.
- Si un auditor no asiste a su turno de trabajo en hora y fecha se procede a registrar su inasistencia de trabajo.

2.3 Modelo del negocio

Con el Modelo del Negocio se pretende llegar a un mejor entendimiento de la organización donde se va a implantar el producto. Los objetivos principales de esta actividad son entender la estructura y la dinámica de la organización para la cual el sistema va a ser desarrollado (organización objetivo) y de esta manera poder derivar los requerimientos del sistema necesarios para apoyar la organización objetivo. (Letelier, 2005)

2.4 Actores y trabajadores del negocio

Los actores son aquellos que interactúan con el negocio para beneficiarse de sus resultados y los trabajadores son los que actúan directamente con el mismo. A continuación se muestran los actores y trabajadores definidos en el negocio. Los procesos que se llevan a cabo son representados por casos de uso, los que producen un resultado observable para los actores. Las entidades del negocio, por su parte, representan a los objetos que los trabajadores toman, inspeccionan, manipulan, producen o utilizan durante la realización de los casos de uso del negocio.

Tabla 2.1 Actores del negocio

Actores	Justificación
Rector	Es la persona encargada de solicitar la evaluación del Control Interno de un área determinada.
Dpto. Auditoría	Encargado de llevar a cabo y evaluar todos los procesos involucrados con el Control Interno de la universidad.

Tabla 2.2 Trabajadores del negocio

Trabajador	Justificación
Auditor	Encargado de evaluar determinada área de la universidad.
Jefe de Expertos	Encargado de realizar el informe general en su área al culminar el proceso de gestión de riesgos.
Expertos en Riesgo	Encargado de identificar, evaluar y darle tratamiento a los riesgos.

2.5 Diagrama de los casos de uso del negocio

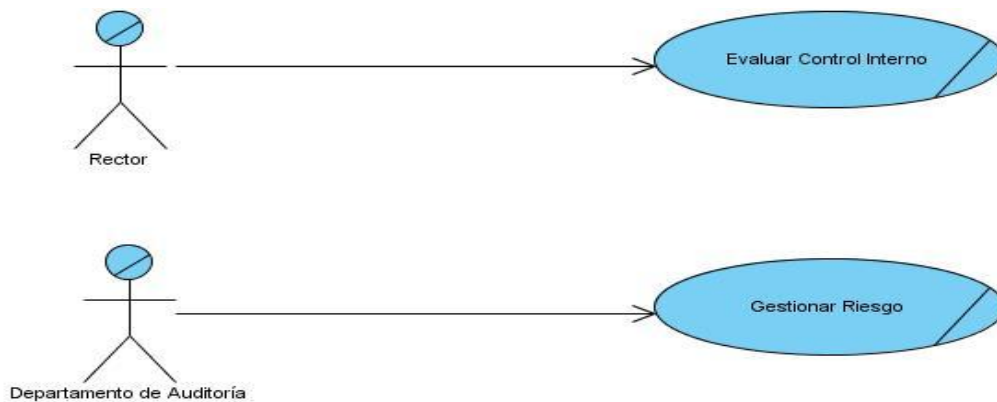


Figura 4 Diagrama de casos de uso del negocio

2.6 Descripción de los casos de uso del negocio

Tabla 2.3 Caso de uso del negocio “Evaluar Control Interno”

Caso Uso	CUN Evaluar Control Interno	
Actores	Departamento de auditoría (Inicia)	
Trabajadores	Auditor	
Resumen	El caso de uso comienza cuando el departamento de auditoría tiene que inspeccionar determinada área de la UCI, el departamento planifica la auditoría, le informa al área la fecha en la cual se le dará cumplimiento a la tarea, posteriormente el auditor va al área y realiza la auditoría, al terminar el auditor emite un informe completo de la actividad realizada y lo discute con el área. Luego de ser procesado por el departamento de auditoría se analiza con el Rector de la Universidad.	
Flujo normal de los eventos		
Acción del Actor	Respuesta del Negocio	
1- El caso de uso se inicia cuando el Rector de la universidad redacta una solicitud de la necesidad de realizar una auditoría en determinada área de la misma y se la envía al departamento de auditoría.	2- El departamento de auditoría planifica la fecha en que se llevará a cabo la auditoría en el área designada y determina un auditor para la realización de la misma.	
	3- El departamento de auditoría le informa al área cuando será evaluada y quien llevará a cabo el proceso.	
	4- El auditor va al área y realiza la auditoría, elabora un informe y discute con el área los resultados obtenidos.	
	5- El departamento de auditoría analiza el informe, lo archiva y lo analiza con el rector.	

6- El rector toma las medidas pertinentes de acuerdo a las dificultades encontradas en el área.	
Prioridad	Crítica

Tabla 2.4 Caso de uso del negocio "Gestionar Riesgo"

Caso Uso	CUN Gestionar Riesgo	
Actores	Departamento de auditoría (Inicia)	
Trabajadores	Expertos en riesgo, jefe de expertos.	
Resumen	El caso de uso comienza cuando el departamento de auditoría le informa a los expertos en riesgos que deben gestionar los riesgos de su área, identificando los mismos, evaluándolos y dándoles tratamiento. Como resultado de éste proceso el jefe de expertos redacta un informe general y lo envía al departamento de auditoría.	
Flujo normal de los eventos		
	Acción del Actor	Respuesta del Negocio
	1- El caso de uso se inicia cuando los expertos en riesgo deciden gestionar los riesgos de su área.	2- Los expertos de cada área identifican los riesgos, los evalúan y les dan tratamiento.
		3- Elaboran un resumen del proceso realizado.

	4- El jefe de expertos conforma un informe general de acuerdo con los resultados del proceso llevado a cabo y lo envía al dpto. de auditoría.
	5- El dpto. de auditoría archiva el informe general.
Prioridad	Crítica

2.7 Diagrama de actividades por caso de uso del negocio

Caso de uso del negocio “Evaluación Control Interno”

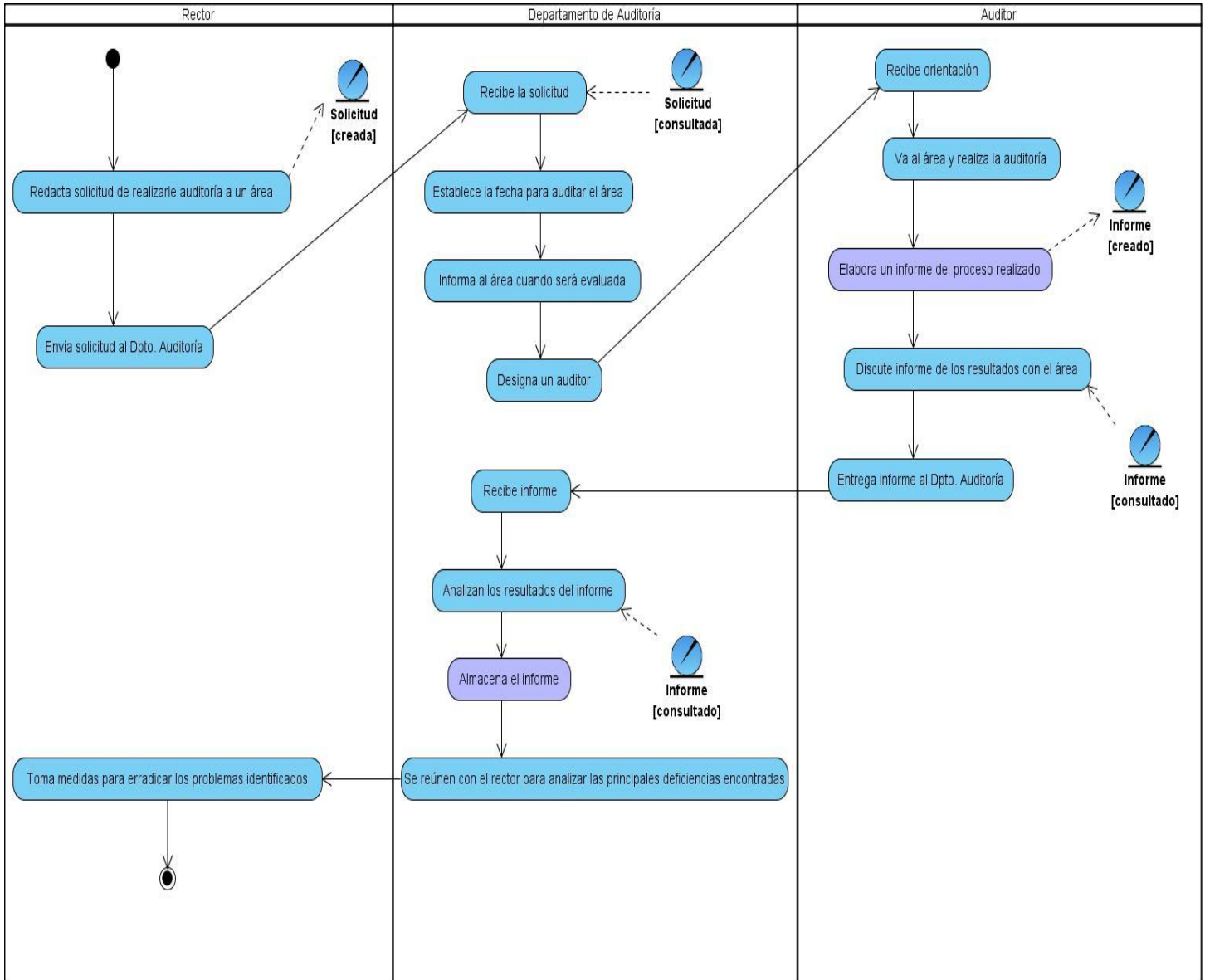


Figura 5 Diagrama de actividades del CUN “Evaluación Control Interno”

Caso de uso del negocio “Gestión de Riesgo”

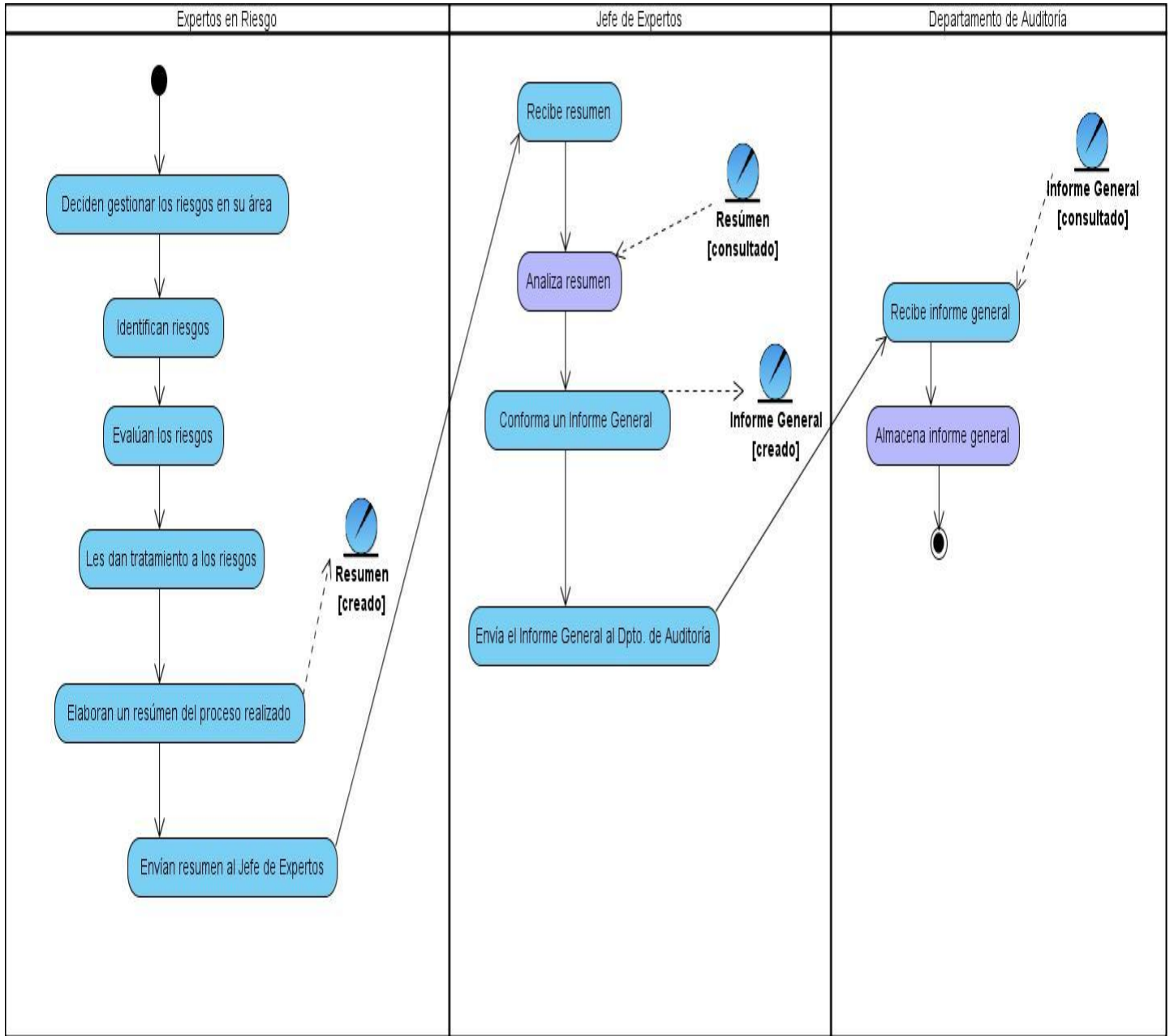


Figura 6 Diagrama de actividades del CUN “Gestión de Riesgo”

2.8 Especificación de los requerimientos de software

La especificación de los requisitos de software es una descripción completa del comportamiento del sistema que se va a desarrollar. Contiene requisitos funcionales que son las capacidades o condiciones que el sistema debe cumplir, en el que se incluye un conjunto de casos de uso que describen las interacciones que tendrán los usuarios con el software y contiene además, requisitos no funcionales que son las propiedades o cualidades que el producto debe tener, que imponen restricciones en el diseño o la implementación. En esta etapa se debe generar una información clara y precisa de los aspectos más relevantes del producto, ya que esta actividad es el hilo conductor de todo el desarrollo del software.

2.8.1 Requerimientos funcionales:

RF 1. Autenticar usuario.

1. Mostrar formulario de autenticación de usuario.
2. Solicitar el nombre de usuario y la clave.
3. Introducir el usuario y la clave.
4. Validar los datos introducidos por el usuario.
 - a) Mostrar mensaje en caso de que existan problemas en la validación.
5. Mostrar las opciones según los privilegios que tenga de acuerdo a su rol.

RF 2. Adicionar usuario.

1. Mostrar formulario para insertar al nuevo usuario.
2. Registrar los datos siguientes:
 - a. Usuario.
 - b. Primer Nombre.
 - c. Segundo Nombre.

- d. Primer Apellido.
- e. Segundo Apellido.
- f. Clave.
- g. Correo.
- h. Rol.

3. Verificar que los datos introducidos estén correctos y no existan campos vacíos, en caso de no estarlos o de existir campos vacíos emitir mensaje de error.
4. Verificar que el usuario no exista, emitir un mensaje de error en caso de existir.
5. Almacenar los datos del nuevo usuario.

RF 3. Modificar usuario.

1. Realizar búsqueda de usuarios. (Para más información de la búsqueda ir al RF 5 Buscar datos de usuarios).
2. Seleccionar el usuario a modificar.
3. Mostrar formulario con los campos del usuario que se pueden modificar:
 - a. Clave.
 - b. Correo.
 - c. Rol.
4. Verificar que los datos sean correctos, en caso de no ser así emitir mensaje de error.
5. Almacenar los nuevos datos de usuario.

RF 4. Eliminar usuario.

1. Realizar búsqueda de usuarios. (Para más información de la búsqueda ir al RF 5 Buscar datos de usuarios).
2. Seleccionar el usuario a eliminar.
3. Mostrar un cartel de confirmación para eliminar el usuario seleccionado.
4. Eliminar el usuario.
5. Mostrar mensaje de éxito en caso de eliminarse satisfactoriamente.

RF 5. Buscar usuario.

1. Realizar una búsqueda de los usuarios de acuerdo a los parámetros siguientes:
 - a. Nombre.
 - b. Apellidos.
 - c. Usuario.
2. Introducir los parámetros de búsqueda.
3. Mostrar resultados.

RF 6. Adicionar plantilla.

1. Realizar búsqueda (Para más información de la búsqueda ir al RF 9 Buscar plantilla).
2. Mostrar formulario para insertar nueva plantilla.
3. Registrar los datos siguientes:
 - a. Título.
 - b. Descripción.
 - c. Documento adjunto.

4. Verificar que los datos introducidos estén correctos y no existan campos vacíos, en caso de no estarlos o de existir campos vacíos emitir mensaje de error.
5. Verificar que la plantilla no exista, emitir un mensaje de error en caso de existir.
6. Almacenar los datos de la nueva plantilla.

RF 7. Modificar plantilla.

1. Realizar búsqueda (Para más información de la búsqueda ir al RF 9 Buscar plantilla).
2. Seleccionar la plantilla a modificar.
3. Mostrar formulario con los campos de la plantilla que se va a modificar:
 - a. Descripción.
 - b. Documento adjunto.
4. Verificar que los datos sean correctos, en caso de no ser así emitir mensaje de error.
5. Almacenar los nuevos datos de la planilla.

RF 8. Eliminar plantilla.

1. Realizar búsqueda. (Para más información de la búsqueda ir al RF 9 Buscar plantilla).
2. Seleccionar la plantilla que se desee eliminar.
3. Mostrar un cartel de confirmación para eliminar la plantilla seleccionada.
4. Eliminar plantilla.
5. Mostrar mensaje de éxito en caso de eliminarse satisfactoriamente.

RF 9. Buscar plantilla.

1. Realizar una búsqueda de las plantillas de acuerdo a los parámetros siguientes:

- a. Título.
 - b. Palabra Clave.
 - c. Fecha.
2. Introducir los parámetros de búsqueda.
 3. Mostrar resultados.

RF 10. Adicionar datos.

1. Realizar búsqueda (Para más información de la búsqueda ir al RF 13 Buscar datos).
 2. Mostrar formulario para insertar datos.
 3. Registrar los datos siguientes:
 - a. Título.
 - b. Descripción.
 - c. Tipo de documento que puede ser:
 - Reportes.
 - Plantillas.
 - Manuales.
 - Leyes.
 - Resoluciones.
 - Artículos.
 - d. Documento adjunto.
-

4. Verificar que los datos introducidos estén correctos y no existan campos vacíos, en caso de no estarlos o de existir campos vacíos emitir mensaje de error.
5. Verificar que el documento no exista, emitir un mensaje de error en caso de existir.
6. Almacenar los datos del nuevo documento.

RF 11. Modificar datos.

1. Realizar búsqueda (Para más información de la búsqueda ir al RF 13 Buscar datos).
2. Seleccionar el documento a modificar.
3. Mostrar formulario con los campos que se pueden modificar:
 - a. Descripción.
 - b. Tipo de documento que pueden ser:
 - Reportes.
 - Plantillas.
 - Manuales.
 - Leyes.
 - Resoluciones.
 - Artículos.
 - c. Documento adjunto.
4. Verificar que los datos sean correctos, en caso de no ser así emitir mensaje de error.
5. Almacenar los nuevos datos.

RF 12. Eliminar datos.

1. Realizar búsqueda. (Para más información de la búsqueda ir al RF 13 Buscar datos).
2. Seleccionar el documento que se desee eliminar.
3. Mostrar un cartel de confirmación para eliminar el documento seleccionado.
4. Eliminar documento.
5. Mostrar mensaje de éxito en caso de eliminarse satisfactoriamente.

RF 13. Buscar datos.

4. Realizar una búsqueda de los documentos de acuerdo a los parámetros siguientes:
 - d. Título.
 - e. Palabra Clave.
 - f. Fecha.
5. Introducir los parámetros de búsqueda.
6. Mostrar resultados.

RF 14. Adicionar reporte.

1. Realizar búsqueda (Para más información de la búsqueda ir al RF 17 Buscar reporte).
2. Mostrar formulario para insertar nuevo reporte.
3. Registrar los datos siguientes:
 - a. Título.
 - b. Descripción.
 - c. Documento adjunto.

4. Verificar que los datos introducidos estén correctos y no existan campos vacíos, en caso de no estarlos o de existir campos vacíos emitir mensaje de error.
5. Verificar que el reporte no exista, emitir un mensaje de error en caso de existir.
6. Almacenar los datos del nuevo reporte.

RF 15. Modificar reporte.

1. Realizar búsqueda (Para más información de la búsqueda ir al RF 17 Buscar reporte).
2. Seleccionar el reporte a modificar.
3. Mostrar formulario con los campos del reporte que se va a modificar:
 - a. Descripción.
 - b. Documento adjunto.
4. Verificar que los datos sean correctos, en caso de no ser así emitir mensaje de error.
5. Almacenar los nuevos datos del reporte.

RF 16. Eliminar reporte:

1. Realizar búsqueda. (Para más información de la búsqueda ir al RF 17 Buscar reporte).
2. Seleccionar el reporte que se desee eliminar.
3. Mostrar un cartel de confirmación para eliminar el reporte seleccionado.
4. Eliminar reporte.
5. Mostrar mensaje de éxito en caso de eliminarse satisfactoriamente.

RF 17. Buscar reporte.

1. Realizar una búsqueda de los reportes de acuerdo a los parámetros siguientes:

- a. Título.
 - b. Palabra Clave.
 - c. Fecha.
2. Introducir los parámetros de búsqueda.
 3. Mostrar resultados.

RF 18. Mostrar Información de últimas actualizaciones.

1. Mostrar la información por categorías de:
 - a. Noticias.
 - b. Lo más relevante.
 - c. Por canales.
2. Seleccionar información.
3. Mostrar información completa, si proviene de los componentes permitir acceso a toda la información.

2.8.2 Requerimientos no funcionales

Los requerimientos no funcionales especifican criterios que puede usarse para juzgar la operación de un sistema en lugar de su funcionalidad. El sistema debe satisfacer las siguientes condiciones o cualidades:

Requerimientos de Software

- El Software se debe ejecutar sobre cualquier plataforma, la PC cliente debe contar con un navegador Web, Mozilla Firefox.

Requerimientos de Hardware

Estaciones de Trabajo:

- Se requiere tarjeta de red.
- Se requiere tengan al menos 256 MB de memoria RAM.
- Se requiere al menos 40 GB de disco duro.
- Procesador 800 MHz como mínimo.

Servidores:

- Se requiere tarjeta de red.
- Se requiere que tengan al menos 512MB de RAM.
- Se requiere al menos 100 GB de disco duro.
- Procesador 2.0 GHz como mínimo.

Requerimientos de Restricciones en el diseño y la implementación

- El sistema se desarrollará con tecnología PHP versión 5.2.5 o superior.
- El análisis y diseño del sistema estará basado en la metodología de desarrollo RUP con el uso del lenguaje de modelado UML.
- Se usara Visual Paradigm como herramienta CASE para el modelado de los artefactos que se generan con cada flujo de trabajo.
- Para el servidor web se utilizará tecnología Apache versión 2.0.50 o superior.
- La base de datos se implementará en PostgreSQL versión 8.2.4 o superior.

Requerimientos de Apariencia o interfaz externa

- Diseño sencillo e interactivo, dando la posibilidad de que no sea necesario tener mucho adiestramiento para utilizar el sistema.

- Diseño perfectamente encuadrado para resoluciones de 800x600, pero preparado para verse en otras resoluciones.
- El diseño debe permitir el uso de los colores azul y blanco, además de elementos relacionados con temas jurídicos. Para la realización del diseño se tendrá en cuenta el control y la transparencia como eje principal en torno al cual girará el sistema.

Requerimientos de Usabilidad

- Los usuarios deben tener un conocimiento básico sobre informática para poder trabajar con el software a desarrollar.

Requerimientos de Portabilidad:

- El sistema debe ser multiplataforma.

Requerimientos de Seguridad

Confidencialidad: La información manejada por el sistema deberá estar protegida de acceso no autorizado y divulgación. Los usuarios accederán a la información correspondiente a cada uno.

Integridad: La información manejada por el sistema será objeto de cuidadosa protección contra estados inconsistentes y corrupción.

Disponibilidad: A los usuarios autorizados se les deberá garantizar el acceso a la información solicitada en todo momento.

Encriptación: La contraseña de la base de datos se encuentra protegida, por el algoritmo de encriptación md5.

Contraseña: Las contraseñas usadas en los servidores y por los usuarios de la aplicación debe ser de 7 o más caracteres. En ambos casos las contraseñas deberán usar la combinación de números, letras y símbolos (".", "-", "+", "_").

Requerimientos Legales

Tanto esta aplicación como la información utilizada para la construcción de la misma, pertenecen al Departamento de Auditoría de la UCI. Debe cumplir con las normas, leyes y políticas de Seguridad establecidas por la Universidad.

2.9 Modelo de casos de uso del sistema

El modelo de casos de uso del sistema permite que los desarrolladores de software y los clientes lleguen a un acuerdo sobre los requisitos, es decir sobre las condiciones y posibilidades que debe cumplir el sistema. Además proporciona la entrada fundamental para el análisis, diseño y pruebas.

2.10 Actores del sistema

Tabla 2.5 Actores del sistema

Actores	Justificación
Administrador	Es el administrador del sistema. Encargado de la gestión de los usuarios del sistema y de gestionar la información referente a las unidades organizativas.
Trabajadores	Usuarios del sistema que solo pueden visualizar la información existente en el mismo, de acuerdo al nivel de acceso que posean.
Jefe de Área	Usuario del sistema que puede visualizar toda la información referente a su área, así como gestionar los usuarios de la misma.
Operador	Usuario del sistema encargado de entrar todos los datos de su área al sistema, así como generar los reportes necesarios. Es el único usuario que puede gestionar toda la información del área.
Sistema de Gestión de Riesgos Operacionales	Componente encargado del proceso de gestión de riesgos, poniendo en práctica los conceptos de probabilidad y consecuencia, así como el grado de exposición del riesgo que presenta el área, evaluando de esta manera el control interno en cada una de ellas. También permitirá a partir de las guías de control realizar revisiones y auditorías a las mismas.
Sistema de Gestión de Auditorías en la UCI	Componente encargado de almacenar todos los expedientes e informes de auditorías que se realicen.

2.11 Diagrama de casos de uso del sistema

Un diagrama de casos de uso del sistema (DCUS) representa gráficamente a los procesos y su interacción con los actores. Además facilita el entendimiento de los procesos realizados por el sistema para el desarrollador.

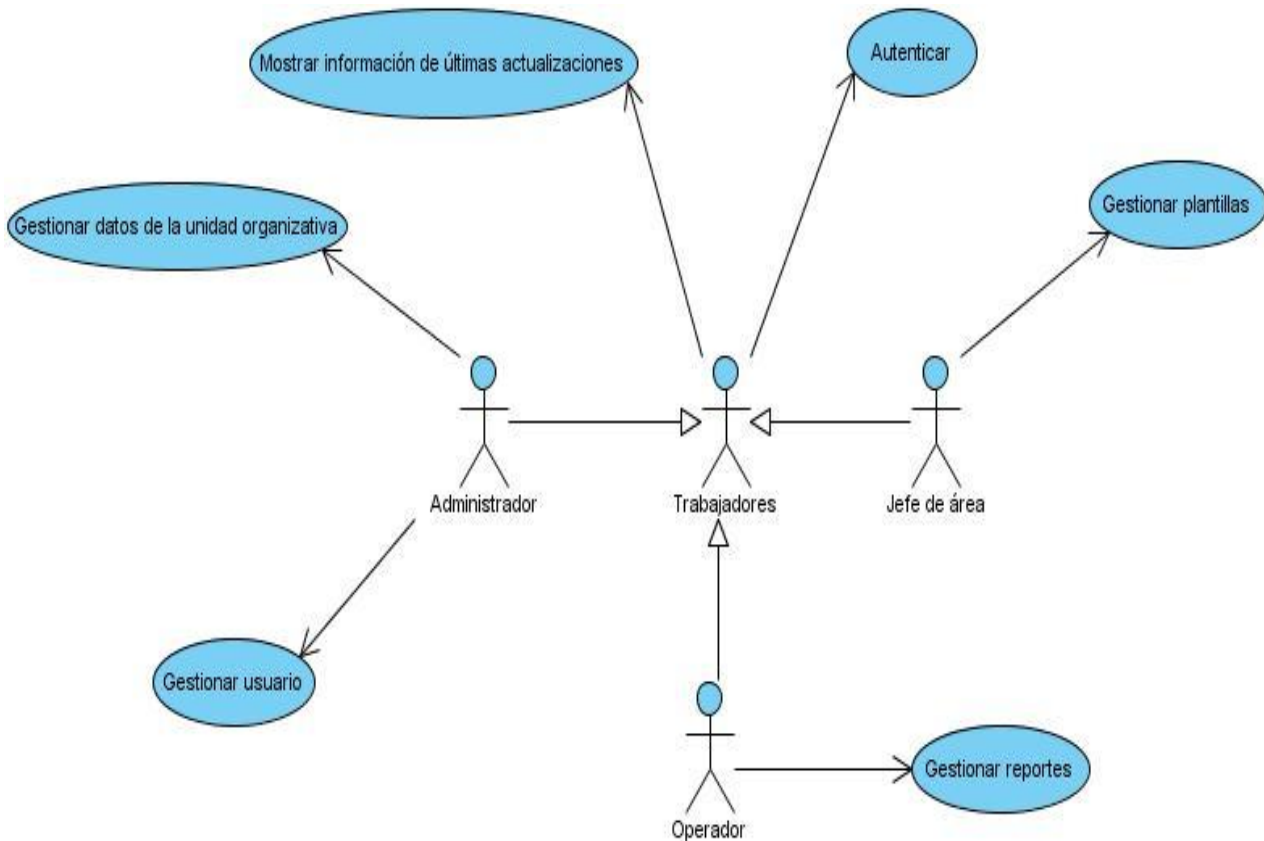


Figura 7 Diagrama de casos de uso del sistema

2.12 Descripciones abreviadas de los casos de uso del sistema

Tabla 2. 6 Resumen del Caso de uso Autenticar usuario

CU-1	Autenticar usuario
Actor	Todos los usuarios
Descripción	El usuario solicita acceder al sistema, el sistema muestra una interfaz para que introduzca los datos, los valida y de ser correctos permite el acceso según su privilegio, si no muestra un mensaje de error y da la opción de intentarlo nuevamente.
Referencia	RF 1.

Tabla 2.7 Resumen del Caso de uso Gestionar usuario

CU-2	Gestionar usuario
Actor	Administrador
Descripción	El administrador del sistema puede gestionar la información referente a todos los usuarios existentes en el sistema, o sea, puede crear un nuevo usuario, modificar los datos de alguno ya existente o eliminarlos.
Referencia	RF 2, RF 2.1, RF 2.2, RF 2.3.

Tabla 2.8 Resumen del Caso de uso Gestionar plantillas

CU-3	Gestionar plantillas
Actor	Jefe de área
Descripción	El jefe de área accede al sistema y selecciona la opción Gestionar Plantillas para habilitar todas las plantillas que se necesiten por el área, así como cualquier documento necesario para el control. A partir de ahí el jefe de área tiene la opción de Adicionar, Modificar, Eliminar y Buscar. Posteriormente introduce los datos necesarios para realizar la operación deseada.
Referencia	RF 3, RF 3.1, RF 3.2, RF 3.3.

Tabla 2.9 Resumen del Caso de uso Gestionar datos de la Unidad Organizativa

CU-4	Gestionar datos de la unidad organizativa
Actor	Administrador
Descripción	El administrador accede al sistema y selecciona la opción Gestionar Datos de la Unidad Organizativa. A partir de ahí el administrador tiene la opción de Adicionar, Modificar, Eliminar y Buscar. Posteriormente introduce los datos necesarios para realizar la operación deseada y finaliza el administrador selecciona la opción salir.
Referencia	RF 4.

Tabla 2.10 Resumen del Caso de uso Gestionar Reportes

CU-5	Gestionar reportes
Actor	Operador
Descripción	El operador selecciona la opción Gestionar Reportes, a partir de ahí tiene la opción de adicionar, eliminar, modificar o buscar un reporte que se genere a partir de la evaluación del control interno en cada una de las áreas de la universidad. Finaliza cuando el operador selecciona la opción salir.
Referencia	RF 5, RF 5.1, RF 5.2, RF 5.3, RF 5.4

Tabla 2.11 Resumen del Caso de uso Mostrar Información de últimas actualizaciones

CU-6	Mostrar Información de últimas actualizaciones
Actor	Trabajadores
Descripción	Los usuarios una vez que acceden al sistema pueden informarse acerca de las últimas actualizaciones hechas en los componentes, así como de lo que acontece acerca del tema del control interno. Finaliza cuando el usuario decida salir de la aplicación.
Referencia	RF 6

2.13 Conclusiones

En el presente capítulo se dio inicio al desarrollo de la propuesta de solución, logrando la descripción detallada de los procesos del negocio. Se definen las principales funcionalidades que tributan al mejoramiento de los procesos del negocio y se obtuvieron los requerimientos funcionales del sistema, además de los actores y trabajadores que intervienen en los mismos, así como el listado de los casos de uso. Se acentuaron las bases del proceso de desarrollo para comenzar con la construcción de la propuesta.

Capítulo 3: Análisis y diseño del sistema

3.1 Introducción

En el presente capítulo se realiza una estimación del esfuerzo mediante el método de estimación por puntos de casos de uso a partir de las características de los requisitos, expresados en los casos de uso. Mediante esta técnica se puede estimar el esfuerzo que empleará el equipo de desarrollo para construir el sistema. El análisis y el diseño de software como flujos de trabajo de la metodología RUP, tienen como principal objetivo transformar los requerimientos de software en lo que el sistema debe hacer y definir una arquitectura de software robusta. A través del análisis se sientan las bases para el diseño. En capítulos anteriores se hizo un análisis de los flujos correspondientes al modelado del negocio y el levantamiento de requisitos, los cuales brindaron la entrada inicial al análisis y el diseño. Con la realización de este capítulo se brindarán los diagramas de clases del análisis, diagrama de clases del diseño, el modelo de datos de la base de datos y el diagrama de despliegue del sistema.

3.2 Estimación del esfuerzo

Paso 1: Cálculo de puntos de casos de uso sin ajustar. Se calcula a partir de la siguiente ecuación:

$UUCP = UAW + UUCW$ donde,

UUCP: Puntos de Casos de Uso sin ajustar

UAW: Factor de Peso de los Actores sin ajustar

UUCW: Factor de Peso de los Casos de Uso sin ajustar

Factor de peso de los actores sin ajustar (UAW)

Este valor se calcula mediante un análisis de la cantidad de actores presentes en el sistema y la complejidad de cada uno de ellos. La complejidad de los actores se establece teniendo en cuenta en primer lugar si se trata de una persona o de otro sistema y en segundo lugar la forma en la que el actor interactúa con el sistema.

Tabla 3.1 Factor de peso de los actores sin ajustar

Actores	Descripción	Complejidad	Factor de Peso
Trabajadores	Una persona que interactúa con el sistema mediante una interfaz gráfica.	Complejo	3
Administrador			
Jefe de Área			
Operador			

UAW = Sumatoria de la multiplicación de la cantidad de actores de un tipo por su factor de peso.

Cantidad de actores de tipo complejo: 4

$$UAW = 4 * 3$$

$$UAW = 12$$

Factor de peso de los casos de uso sin ajustar (UUCW)

Este valor se calcula mediante un análisis de la cantidad de casos de uso presentes en el sistema y la complejidad de cada uno de ellos. La complejidad de los casos de uso se establece teniendo en cuenta la cantidad de transacciones efectuadas en el mismo, donde una transacción se entiende como una secuencia de actividades atómica, es decir, se efectúa la secuencia de actividades completa, o no se efectúa ninguna de las actividades de la secuencia y está representada por uno o más pasos del flujo de eventos principal del caso de uso, pudiendo existir más de una transacción dentro del mismo caso de uso.

Tabla 3.2 Factor de peso de los casos de uso sin ajustar

Caso de Uso	Descripción	Complejidad	Factor de Peso
CUS Autenticar Usuario	Contiene 2 transacciones	Simple	5
CUS Gestionar Usuario	Contiene 3 transacciones	Simple	5
CUS Gestionar Plantillas	Contiene 5 transacciones	Medio	10

CUS Gestionar Datos de la Unidad Organizativa	Contiene 6 transacciones	Medio	10
CUS Gestionar Reportes	Contiene 4 transacciones	Medio	10
CUS Mostrar Información de Ultimas Actualizaciones	Contiene 1 transacción	Simple	5

UUCW = Sumatoria de los casos de uso por su complejidad.

$$UUCW = 3 * 5 + 3 * 10$$

$$UUCW = 45$$

Finalmente, los puntos de casos de uso sin ajustar resultan

$$UUCP = UAW + UUCW$$

$$UUCP = 12 + 45$$

$$UUCP = 57$$

Paso 2: Cálculo de puntos de casos de uso ajustados

$$UCP = UUCP \times TCF \times EF \text{ donde,}$$

UCP: Puntos de casos de uso ajustados

UUCP: Puntos de casos de uso sin ajustar

TCF: Factor de complejidad técnica

EF: Factor de ambiente

Factor de complejidad técnica (TCF)

Para calcular este coeficiente se analizan una serie de factores que pueden determinar la complejidad técnica del sistema, a estos factores se les asocia un valor de 0 a 5 que determina el vínculo del mismo con las características deseadas para el sistema.

Tabla 3.3 Factor de complejidad técnica

Factor	Descripción	Peso	Valor Asignado	Peso * Valor
T1	Sistema distribuido	2	0	0
T2	Objetivos de performance o tiempo de respuesta	1	4	4
T3	Eficiencia del usuario final	1	2	2
T4	Procesamiento interno complejo	1	3	3
T5	El código debe ser reutilizable	1	5	5
T6	Facilidad de instalación	0.5	4	2
T7	Facilidad de uso	0.5	3	1.5
T8	Portabilidad	2	4	8
T9	Facilidad de cambio	1	3	3
T10	Concurrencia	1	5	5
T11	Incluye objetivos especiales de seguridad	1	5	5
T12	Provee acceso directo a terceras partes	1	0	0
T13	Se requieren facilidades especiales de entrenamiento a usuarios	1	3	3

El Factor de complejidad técnica resulta:

$$TCF = 0.6 + 0.01 * \Sigma (\text{Peso} * \text{Valor asignado})$$

$$TCF = 0.6 + 0.01 * (0+4+2+3+5+2+1.5+8+3+5+5+0+3)$$

$$TCF = 0.6 + 0.01 * 41.5$$

$$TCF = 1.015$$

Factor de ambiente (EF)

Para calcular este coeficiente se analizan una serie de factores que pueden determinar el tiempo requerido para el desarrollo del sistema, teniendo en cuenta aspectos como habilidades, conocimientos, etc. de los involucrados en la realización del sistema. A estos factores se les asocia un valor de 0 a 5 que determina el vínculo del mismo con las características deseadas para el sistema.

Tabla 12.4 Factor ambiente

Factor	Descripción	Peso	Valor * Peso
E1	Familiaridad con el modelo de proyecto utilizado	0	0
E2	Experiencia en la aplicación	1.5	1.5
E3	Experiencia en orientación a objetos	4	4
E4	Capacidad del analista líder	2.5	2.5
E5	Motivación	4	4
E6	Estabilidad de los requerimientos	4	4
E7	Personal part-time	0	0
E8	Dificultad del lenguaje de programación	-1	-2

El Factor de ambiente resulta:

$$EF = 1.4 - 0.03 * \Sigma (\text{Peso} * \text{Valor asignado})$$

$$EF = 1.4 - 0.03 * \Sigma (0+1.5+4+2.5+4+4+0-2)$$

$$EF = 1.4 - 0.03 * 14$$

$$EF = 0.98$$

Finalmente, los puntos de casos de uso ajustados resultan:

$$UCP = UUCP * TCF * EF$$

$$UCP = 57 * 1.015 * 0.98$$

$$UCP = 56.6979$$

El esfuerzo en horas-hombre viene dado por:

$$E = UCP * CF \text{ donde,}$$

E: esfuerzo

UCP: puntos de casos de uso ajustados

CF: factor de conversión

Para calcular CF

Total EF = Cantidad EF < 3 (entre E1 – E6) + Cantidad EF > 3 (entre E7, E8)

$$\text{Total EF} = 2 + 0$$

$$\text{Total EF} = 2$$

CF = 20 horas-hombre (si Total EF ≤ 2)

CF = 28 horas-hombre (si Total EF = 3 ó Total EF = 4)

CF = abandonar o cambiar proyecto (si Total EF ≥ 5)

Para este tipo de proyecto 20 horas-hombre/punto de casos de uso, es decir, un punto de caso de uso toma 20 horas-hombre.

$$E = UCP * CF$$

$$E = 56.6979 * 20$$

$$E = 1133.958 \text{ horas-hombres}$$

Distribución del esfuerzo

Tabla 13 Distribución del esfuerzo

Actividad	Porcentaje	Horas-Hombre
Análisis	10%	472.4825
Diseño	20%	944.965
Implementación	40%	1133.958
Pruebas	15%	708.72375
Sobrecarga(otras actividades)	15%	708.72375
Total	100%	3968.853

Esfuerzo Total (horas-hombres) 3968.853.

Para la etapa de análisis y diseño se requiere de un esfuerzo de 1417.4475 horas-hombres, si se considera que trabajan dos personas, 48 horas como promedio en la semana, esta etapa debe terminarse en aproximadamente 15 semanas.

3.3 Modelo de análisis

Con el Modelo de Análisis se pretende identificar las clases que describen la realización de los Casos de Uso, los atributos y las relaciones entre ellas. Con esta información se construyen los diagramas de clase de análisis. Estos diagramas brindan una representación, desde el punto de vista informático de la interacción del usuario con el sistema. El análisis brinda una vista interna, “qué” debe hacer el sistema y el diseño es el que especifica “cómo” lograrlo.

3.4 Diagrama de clases de análisis

El diagrama de clases del análisis es un artefacto que representa las cosas del mundo real. El Análisis se basa en clases, que están catalogadas en “Interfaz”, “Controladora” y “Entidad” en dependencia de las funciones que realicen. A continuación se muestran los diagramas de clase de análisis de los casos de uso del sistema.

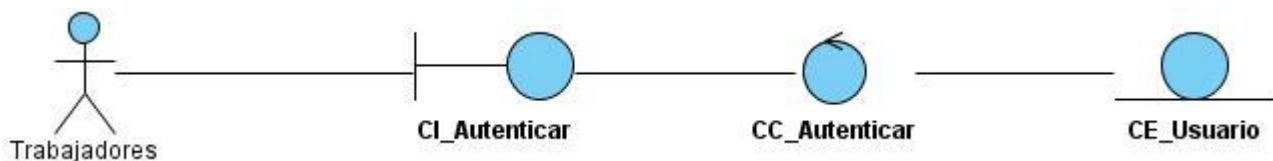


Figura 8 Diagrama de clases del análisis: CUS "Autenticar"

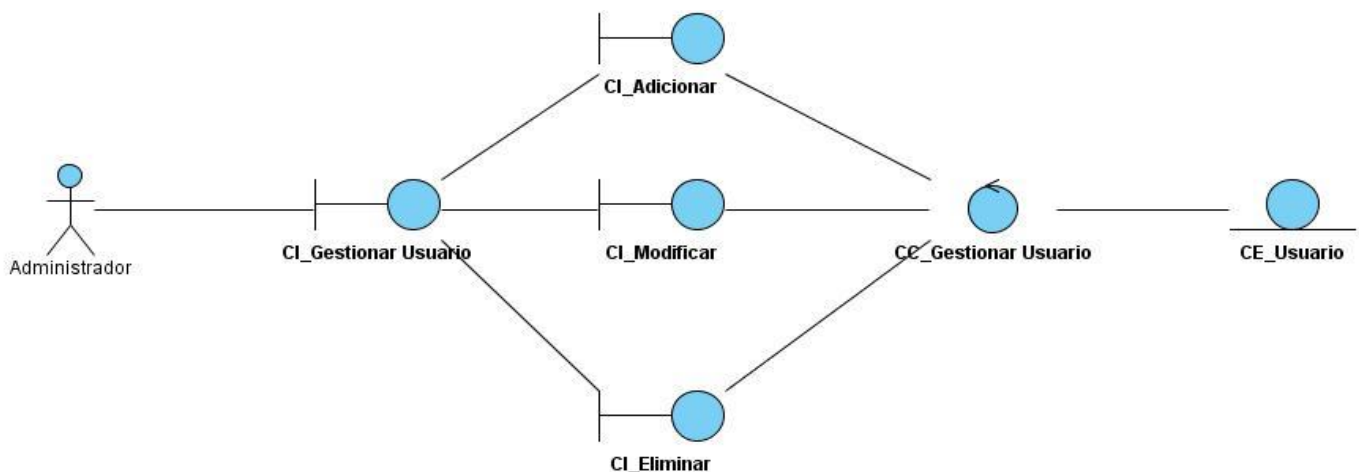


Figura 9 Diagrama de clases del análisis: CUS "Gestionar Usuario"

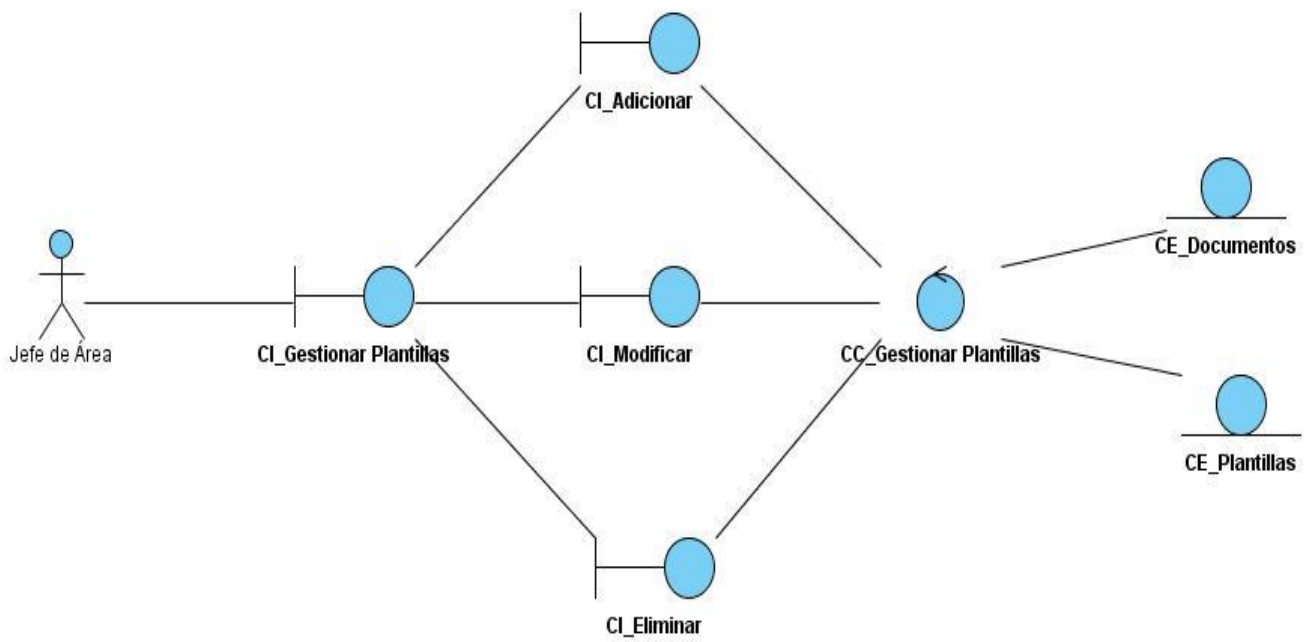


Figura 10 Diagrama de clases del análisis: CUS "Gestionar Plantillas"

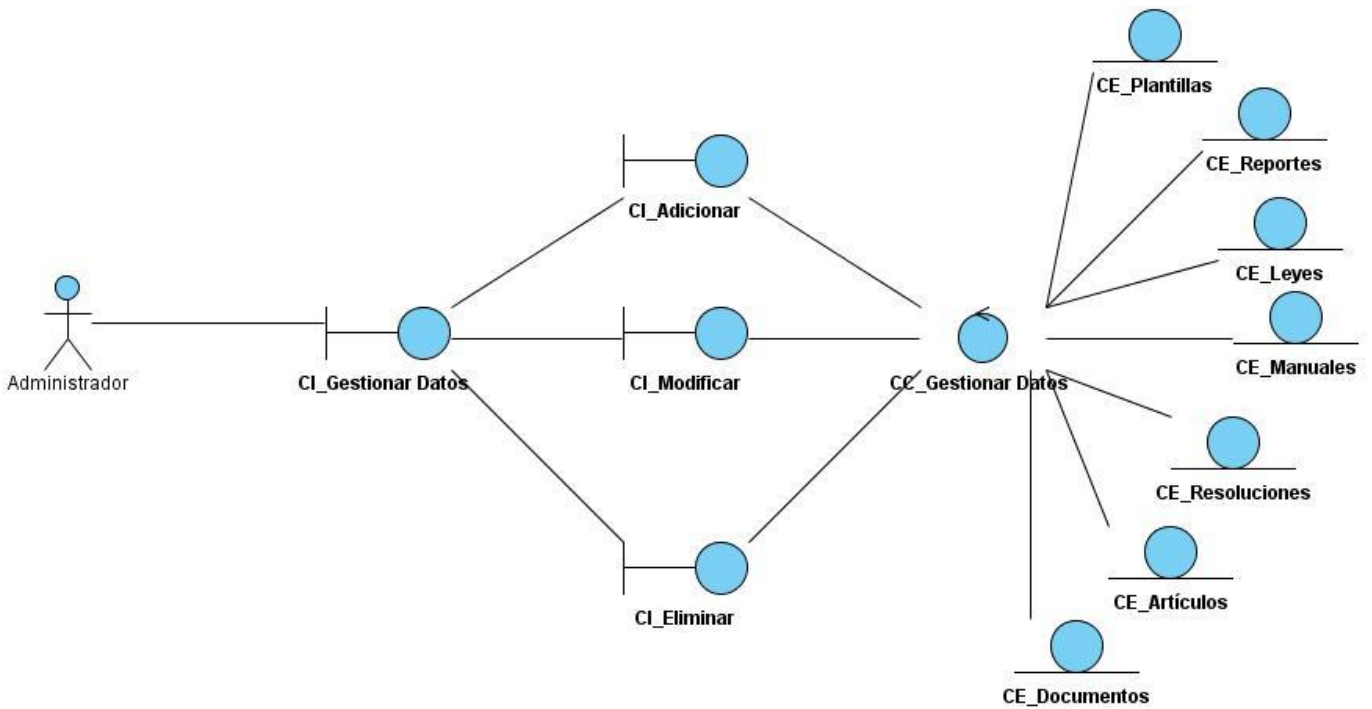


Figura 11 Diagrama de clases del análisis: CUS "Gestionar Datos de la Unidad Organizativa"

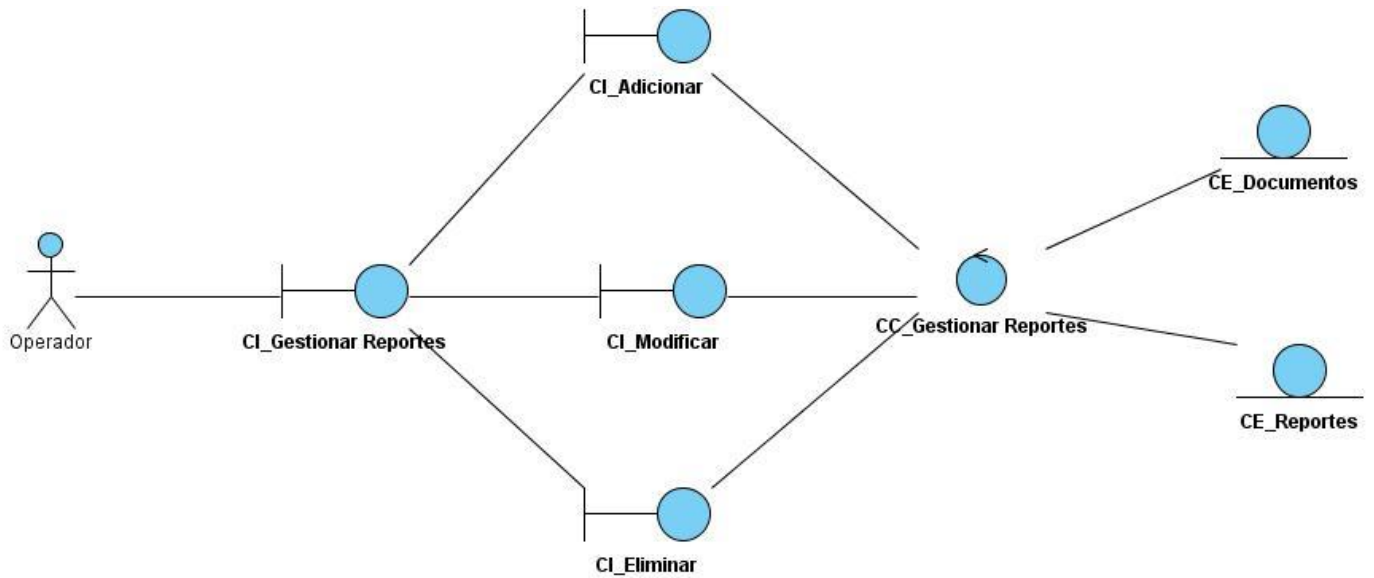


Figura 12 Diagrama de clases del análisis: CUS “Gestionar Reportes”



Figura 13 Diagrama de clases del análisis: CUS “Mostrar Información de últimas actualizaciones”

3.5 Modelo de diseño

El flujo de diseño de un software tiene como principal objetivo decidir cómo el sistema debe ser implementado. Se modela el sistema y se encuentra su forma, incluyendo la arquitectura y sirve de abstracción de la implementación del sistema.

En el diagrama del diseño es donde se describe la estructura del sistema mostrando sus clases, atributos y las relaciones entre ellos. A partir del mismo se crea el diseño conceptual de la información que se manejará en el sistema, los componentes que se encargarán del funcionamiento así como la relación entre uno y otro.

El modelado de sus clases debe estar sujeto al lenguaje de programación, así como al patrón arquitectónico seleccionado. En este caso se ha seleccionado el patrón de arquitectura Modelo Vista Controlador(MVC) el cual es utilizado en sistemas Web para separar los datos de una aplicación, la interfaz de usuario y la lógica de control en tres componentes distintos, permitiendo flexibilidad y facilidad

a la hora de hacer futuros cambios. A continuación se presentan los diagramas de clases del diseño a través de sus estereotipos web.

3.6 Diagrama de clases del diseño

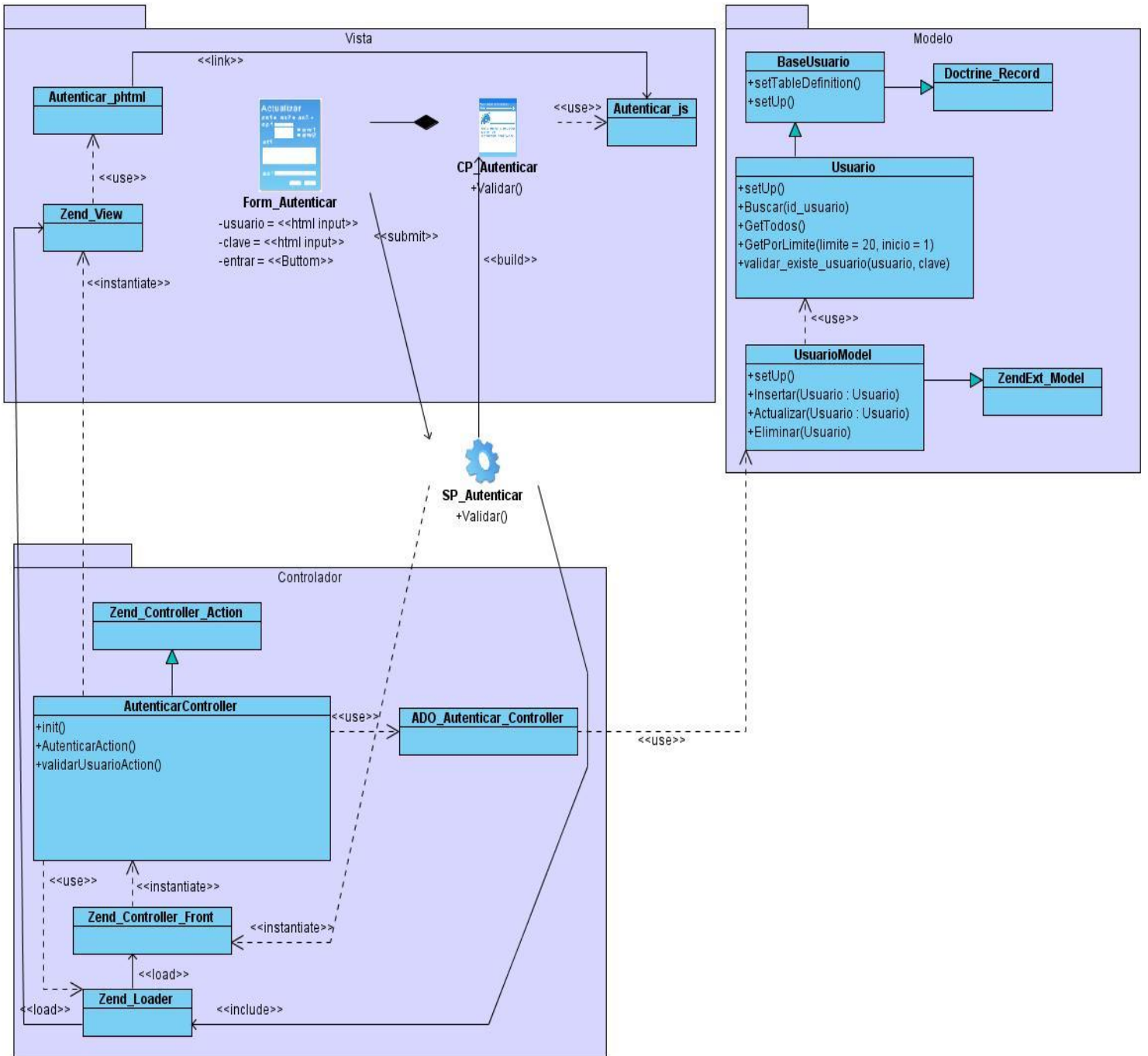


Figura 14 Diagrama de clases Web: CUS "Autenticar"

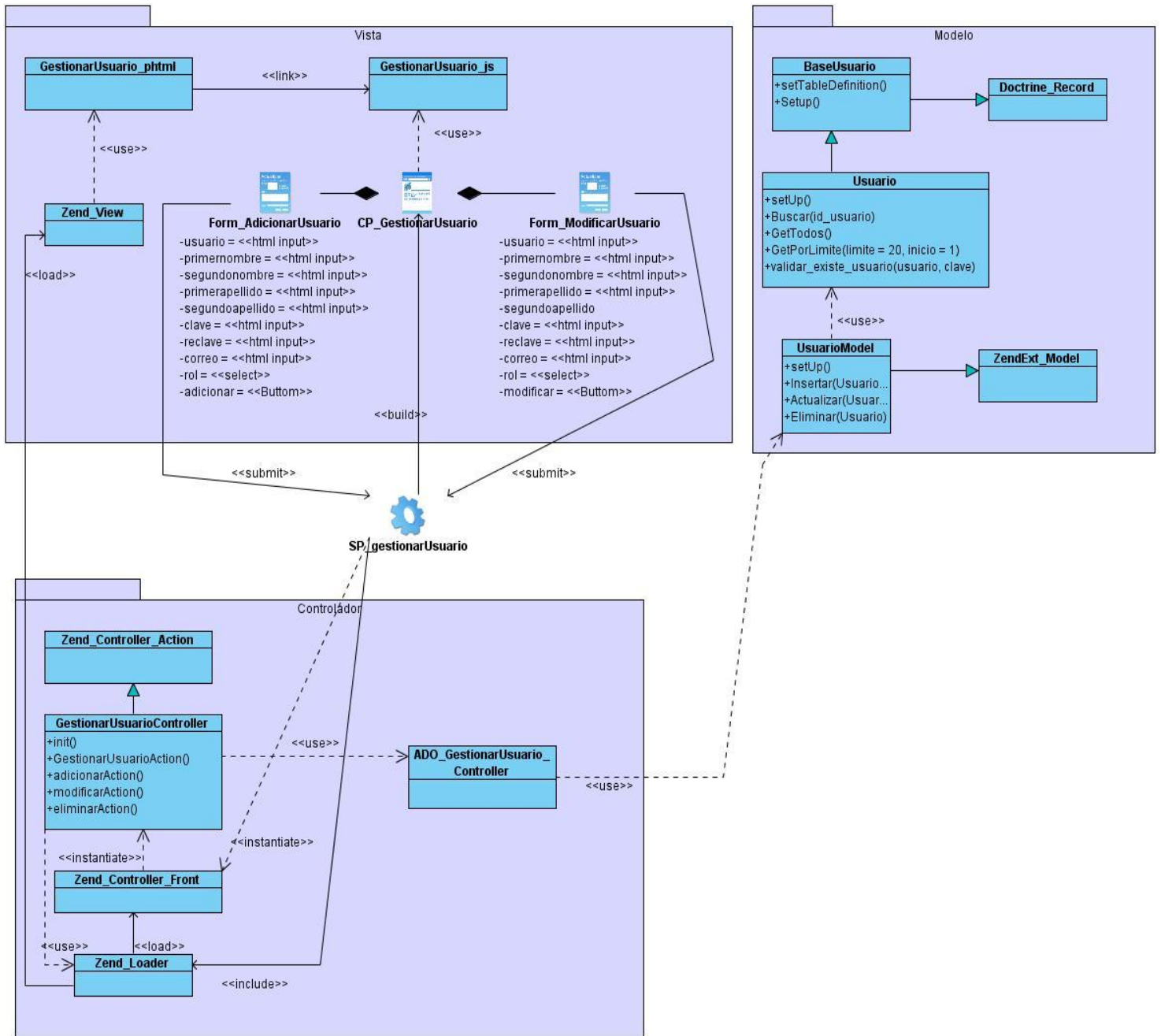


Figura 15 Diagrama de clases Web: CUS "Gestionar Usuario"

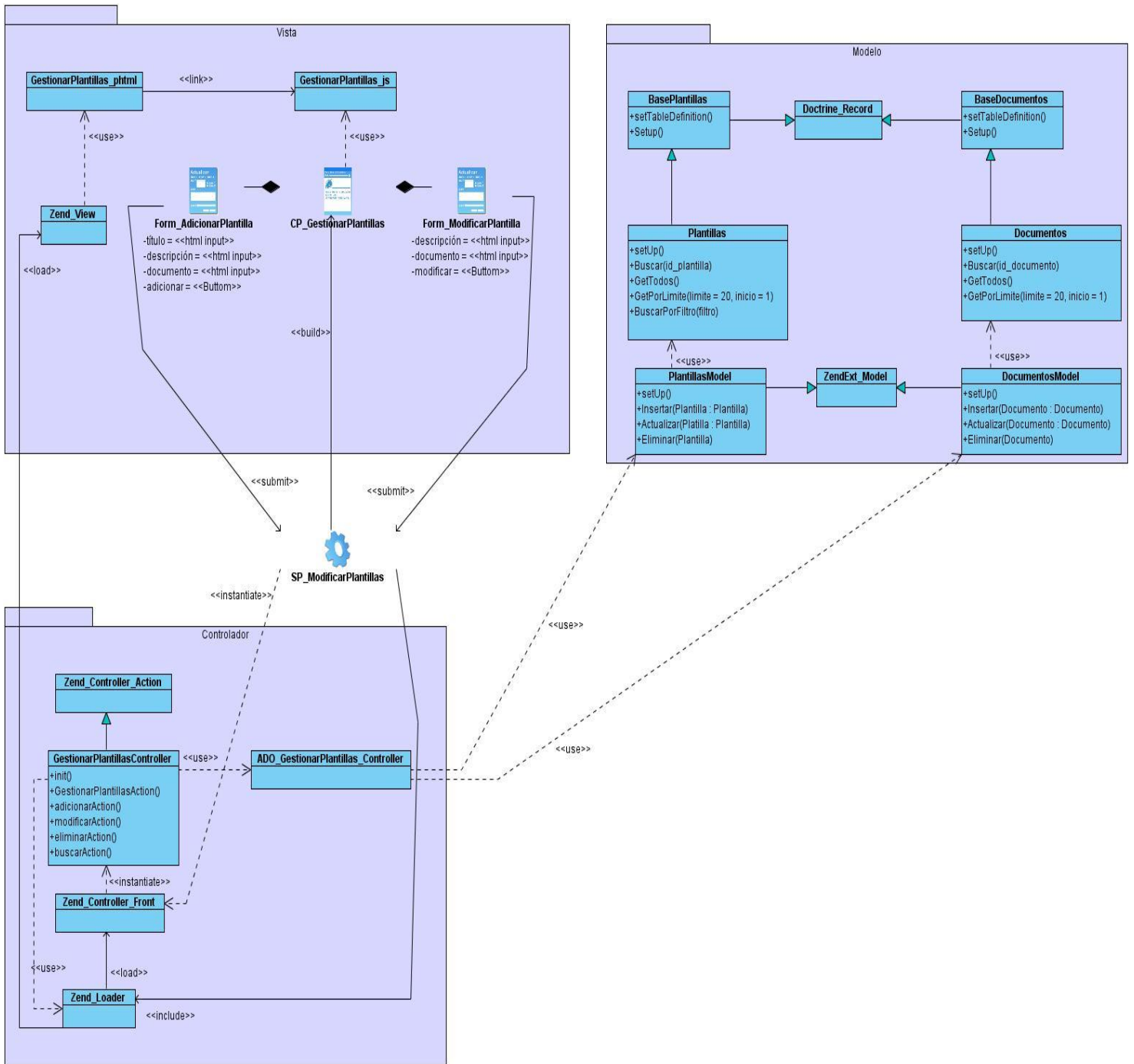


Figura 16 Diagrama de clases Web: CUS “Gestionar Plantillas”

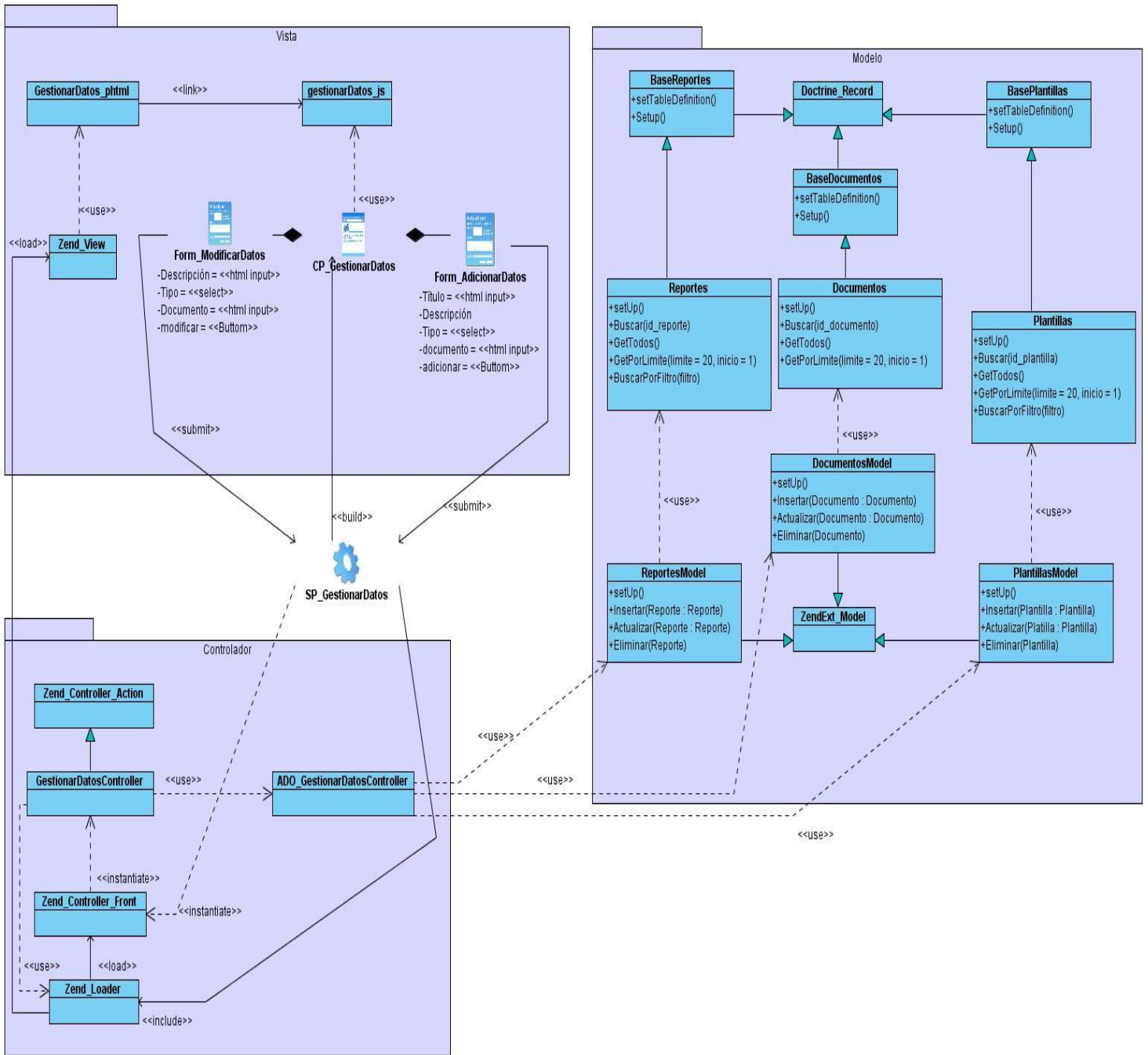


Figura 17 Diagrama de clases Web: CUS “Gestionar Datos de la Unidad Organizativa”

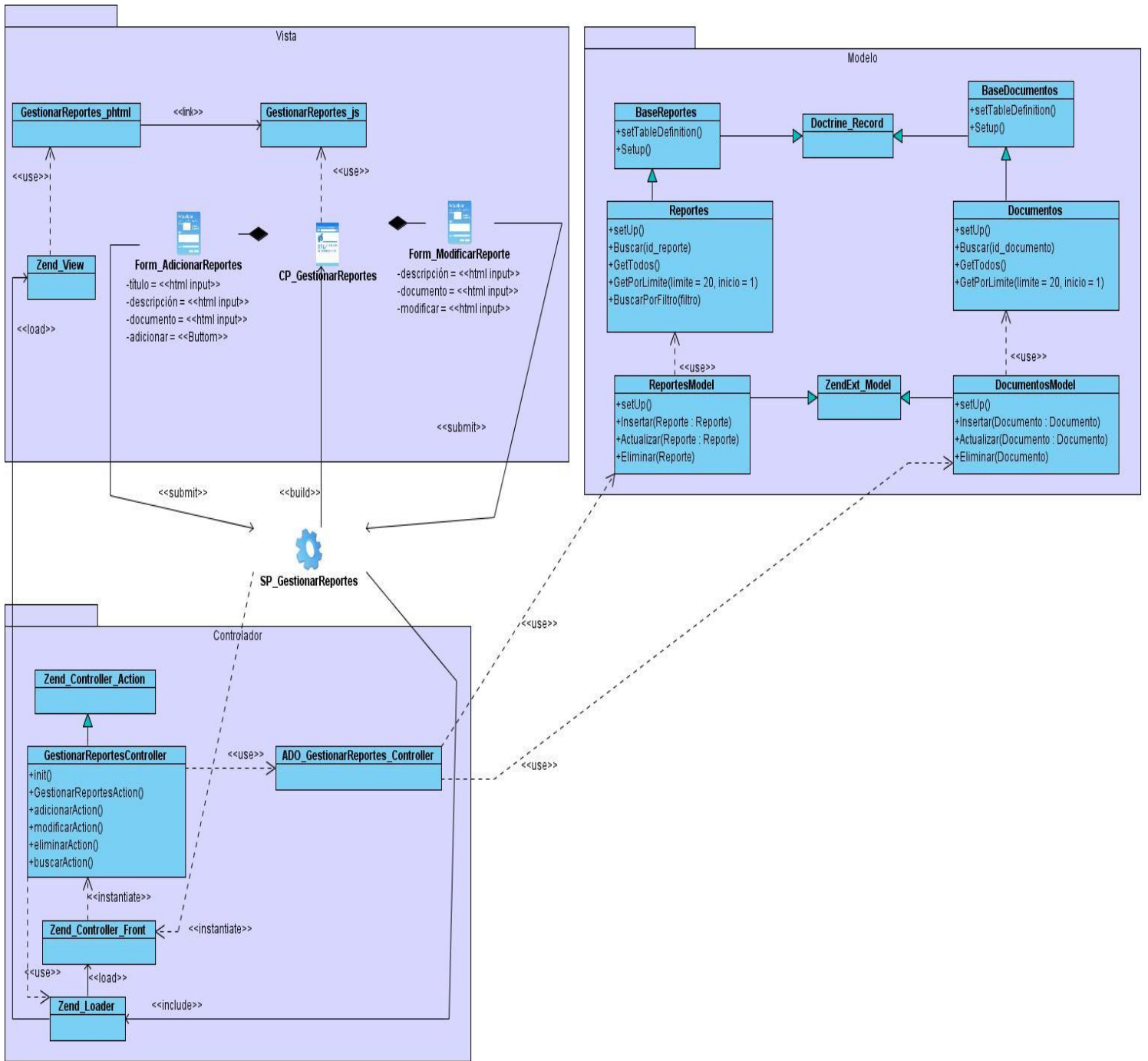


Figura 18 Diagrama de clases Web: CUS "Gestionar Reportes"

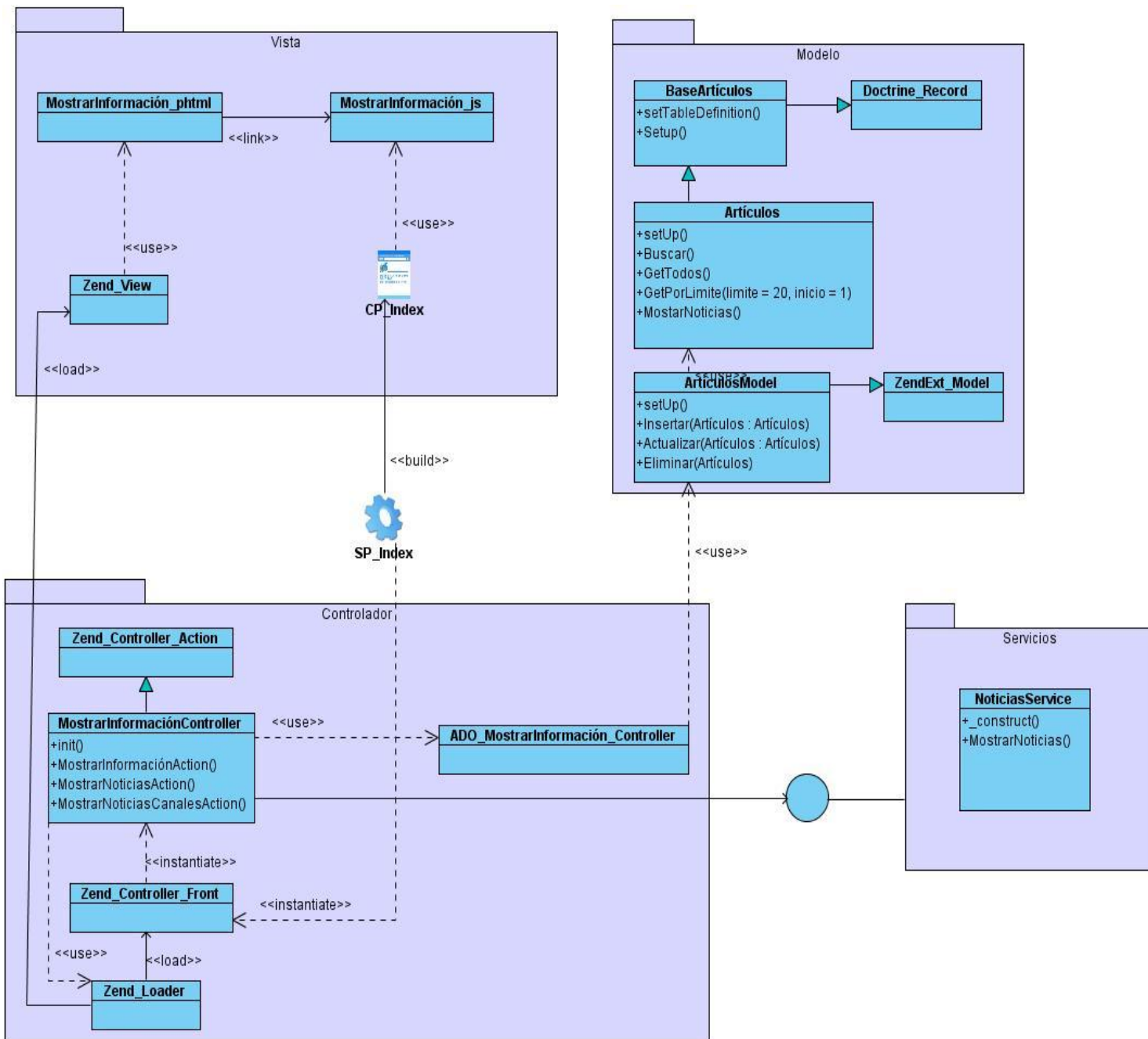


Figura 19 Diagrama de clases Web: CUS “Mostrar Información de Últimas Actualizaciones”

3.7 Modelo de datos

Partiendo del diagrama de clases del diseño se realiza la concepción del diseño de la base de datos, la cual tiene como objetivo asegurar que los datos persistentes sean almacenados de forma segura y eficiente, definiendo cómo se debe implementar una base de datos.

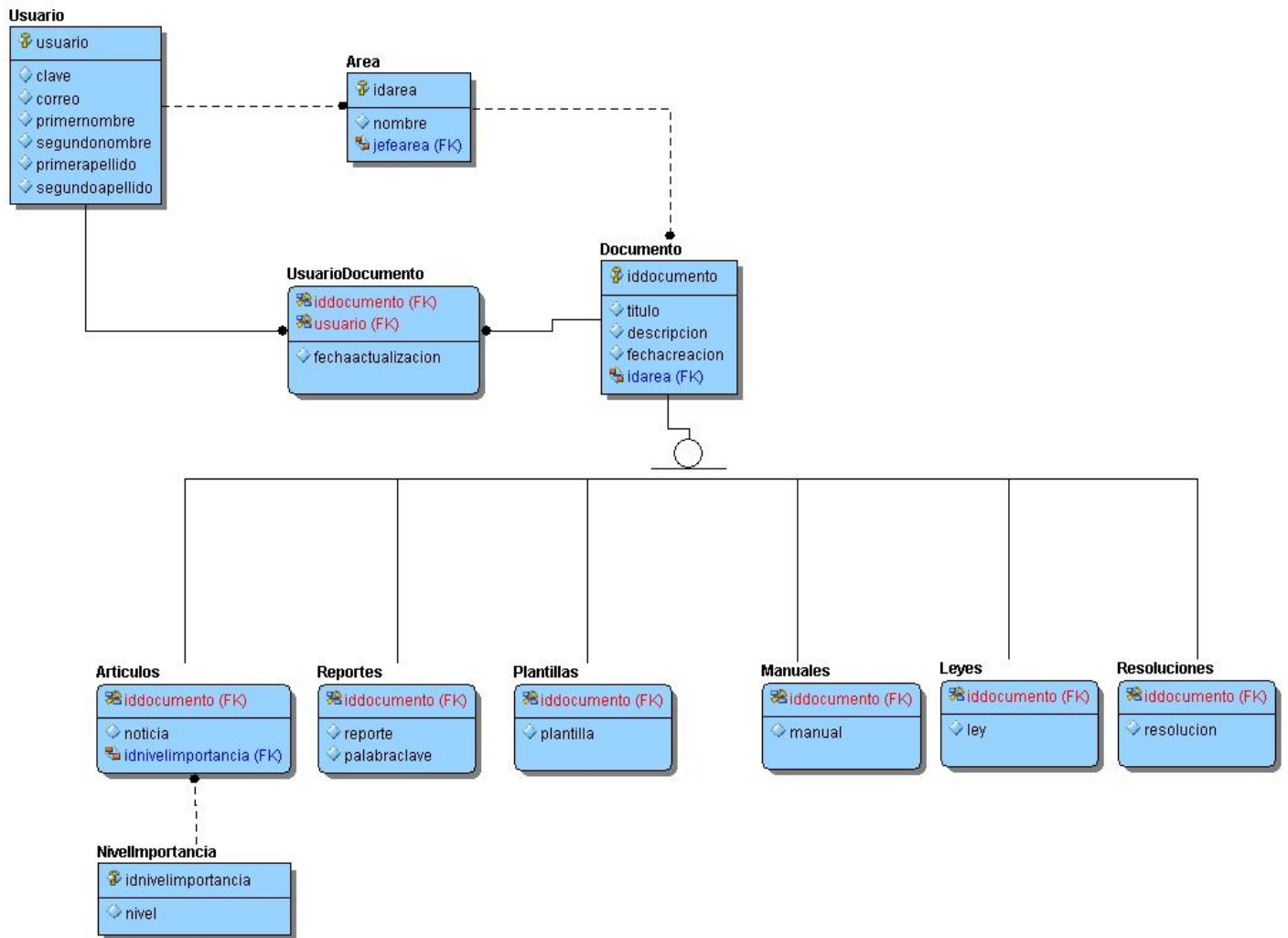


Figura 20 Modelo de datos

3.8 Diagrama de despliegue

El diagrama de despliegue describe la distribución física y espacial del sistema a través de nodos. Mediante el diagrama de despliegue se captura la configuración de los elementos de procesamiento y sus conexiones y se visualiza la distribución de los componentes de software en nodos físicos.

A continuación se presenta el modelo de despliegue, donde se muestra la existencia de los servidores de aplicación y los servidores de bases de datos que se han definido. Se presentan los nodos correspondientes a las PC, así como los dispositivos conectados.

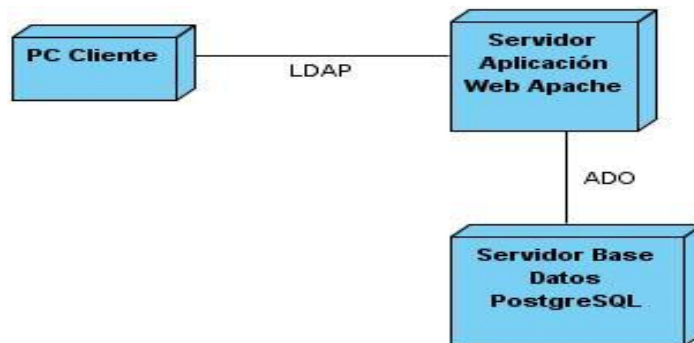


Figura 21 Diagrama de despliegue

3.9 Validación de la propuesta de solución

Técnicas de validación de requisitos:

Los requisitos una vez definidos necesitan ser validados. La validación de requisitos tiene como misión demostrar que la definición de los requisitos define realmente el sistema que el usuario necesita o el cliente desea (Lowe & Hall, 1999). Es necesario asegurar que el análisis realizado y los resultados obtenidos de la etapa de definición de los requisitos son correctos. Pocas son las propuestas existentes que ofrecen técnicas para la realización de la validación y muchas de ellas consisten en revisar los modelos obtenidos en la definición de requisitos con el usuario para detectar errores o inconsistencias.

Aún así, existen algunas técnicas que pueden aplicarse para ello:

Reviews o Walk-throughs: Esta técnica consiste en la lectura y corrección de la completa documentación o modelado de la definición de requisitos. Con ello solamente se puede validar la correcta interpretación de la información transmitida. Más difícil es verificar consistencia de la documentación o información faltante.

Listas de chequeo: Son frecuentemente usadas en inspecciones o revisiones de artefactos generados en el proceso de producción de software; son listas de aspectos que deben ser completados o verificados,

Auditorías: La revisión de la documentación con esta técnica consiste en un chequeo de los resultados contra una checklist (Listas de Chequeo) predefinida o definida a comienzos del proceso, es decir sólo una muestra es revisada.

Prototipos: Algunas propuestas se basan en obtener de la definición de requisitos prototipos que, sin tener la totalidad de la funcionalidad del sistema, permitan al usuario hacerse una idea de la estructura de la interfaz del sistema con el usuario (Olsina, 1999).

- Un prototipo es una versión inicial de un sistema de software que se utiliza para demostrar los conceptos, probar las opciones de diseño y entender mejor el problema y su solución.
- Un prototipo puede revelar errores u omisiones en los requisitos propuestos, favorece la comunicación entre clientes y desarrolladores, da una primera visión del producto.
- Esta técnica tiene el problema de que el usuario debe entender que lo que está viendo es un prototipo y no el sistema final.

La validación de requisitos es una actividad muy importante, pues un levantamiento de requisitos con errores que no se detecten a tiempo, además de no conducir a resultados inesperados provoca costos excesivos y gran pérdida de tiempo.

3.10 Conclusiones

En el presente capítulo se mostraron los principales artefactos del flujo de trabajo definido por RUP, análisis y diseño, los cuales brindan la idea de qué debe hacer y cómo debe hacerse el sistema a desarrollar. Se realizó un modelo detallado de la solución propuesta, además de presentar los respectivos diagramas de clases del análisis y el diseño por cada caso de uso del sistema. También se presentó el modelo de datos, el cual se realiza en base a las clases persistentes del diagrama de clases del diseño, o modelo de diseño. Este modelo físico presenta la información que debe ser almacenada en la base de datos. Se definió el despliegue del sistema, con las especificaciones del hardware y software que debe presentar el entorno para el sistema de control interno de la universidad.

CONCLUSIONES

Con el desarrollo de la presente investigación se concluye lo siguiente:

1. Fueron alcanzados los objetivos propuestos y se demostró la necesidad de diseñar e implementar un sistema Web para el departamento de auditoría de la UCI.
2. Se efectuó una investigación profunda de los sistemas informáticos vinculados al campo de acción y de las fundamentales tendencias y tecnologías utilizadas para la implementación de sistemas de este tipo. Donde se obtuvo como resultado la selección de las herramientas adecuadas y la metodología para el desarrollo de la solución propuesta.
3. Se realizó el análisis y diseño de la propuesta de aplicación Web, definiendo las funcionalidades del mismo y se obtuvieron los distintos artefactos generados a través del uso de la metodología RUP que sirven como guía a los desarrolladores para la posterior implementación del sistema.

RECOMENDACIONES

Es preciso tener en cuenta las siguientes recomendaciones en vías de perfeccionar el trabajo realizado:

1. Implementar el sistema para el control interno de UCI, teniendo como base la labor realizada con la presente investigación con el objetivo de mejorar dicho proceso.
2. Confeccionar la ayuda y el manual de usuario.

BIBLIOGRAFÍA REFERENCIADA

Cuesta, P. 2005. [En línea] 2005. <http://www.latinspin.org>.

El informe COSO y sus repercusiones: un enfoque moderno del control interno. **Ibáñez, P.C. 2008.** 203-220, Valencia : s.n., 2008.

Escalona, María José and Koch, Nora. 2002. *Ingeniería de Requisitos en Aplicaciones para la Web-Un estudio comparativo.* Sevilla : s.n., 2002.

G., B RUBIO. Portal de la Universidad de Islas Baleares. [En línea] [Citado el: 28 de marzo de 2010.] <http://dmi.uib.es/>

González, Jose Luis Hernández. Instituto Tecnológico de Apisaco. [Online] [Cited: 04 09, 2009.]

Jimmy, Rosario. 2005. Archivo de la CiberSociedad. [En línea] 2005. <http://www.cibersociedad.net/archivo.php?art218>.

Larman, C. 2004. UML y Patrones .Introducción al análisis y diseño orientado a objetos.

Letelier, P. Departamento de Sistemas Informáticos y Computación,. Introducción a RUP. [En línea] [Citado el: 5 de junio de 2010.]

<https://pid.dsic.upv.es/C1/Material/DocumentosDisponibles/IntroduccionaRUP.doc>.

López, Jorge Quesada. 2004. La Informática y la Empresa. [Online] 10 2004. [Cited: 04 09, 2009.] <http://www.fec.uh.cu/info3/RECOPIACION.htm>

Santander, José L Estrada. 1987. Diccionario Económico. La Habana : Editora Política, 1987. 158.

Treadway, J.C. Informe de la Comisión Nacional sobre Información Financiera Fraudulenta. [En línea] [Citado el: 11 de marzo de 2010.] <http://www.coso.org/Publications/NCFFR.pdf>.

2003. Resolución del Ministerio de Finanzas y Precios. Habana : s.n., 2003. 297.

2008.Estilos y Patrones en la Estrategia de Arquitectura de Microsoft. [En línea] 2008.
http://eva.uci.cu/mod/resource/view.php?id=4599&subdir=/Materiales_Complementarios_Conf_7..

2009. Ley de la Contraloría General de la República. Habana : s.n., 2009. 107.

BIBLIOGRAFÍA CONSULTADA

Abiertos, Estándares. 2007. *Proyecto Estándares Abiertos.* Europa : s.n., 2007.

Bancaria, Comité de Basilea para la Supervisión. 1998. *Marco para la evaluación de los sistemas de Control Interno.* Basilea : s.n., 1998.

Bilick, Guillermo. 2003. *Sistema de Control Interno.* s.l. : Facultad y Deber, 2003.

Colombia, Autorregulador del Mercado de Valores de. 2008. *Guía de Control Interno en la intermediación del mercado de valores.* Bogota : s.n., 2008.

Coso. 2004. *Gestión de Riesgos Corporativos-Marco Integrado.* Valencia : s.n., 2004.

CONNELL., S. 1997. *Desarrollo y Gestión de Proyectos Informáticos.* 1997.

Jacobson ,G.Booch y J, Rumbaugh. 2000. Madrid : Addison-Wesley, 2000.

Larman, C. 2008. *El mundo informatico.* 2008.

Pressman, Roger S. *Ingeniería de software, un enfoque práctico.*

Marín, María Alejandra. 1998. *NUEVOS CONCEPTOS DE CONTROL INTERNO.* Valencia : s.n., 1998.

Rosales, Humberto Blanco. 2000. *El factor gerencial y el perfeccionamiento de la empresa estatal cubana.* 2000.

Rumbaugh, James. 2007. *El Lenguaje Unificado De Modelado.* Valencia : s.n., 2007.

Schumler, J. 2000. *Aprendiendo UML en 24 horas.* Mexico : Pearson Education, 2000.

Vanstapel, Franki. 2004. *Guía para las normas de control interno del sector público.* Bruselas : s.n., 2004.

Sommerville, Ian. 2005. *Ingeniería de software.* s.l. : Pearson Education, 2005.

Wikimedia Foundation. 2009. Wikipedia. [En línea] 15 de 02 de 2010.
<http://es.wikipedia.org/wiki/PostgreSQL>

Wikimedia Foundation. 2009. Wikipedia. La Enciclopedia Libre [En línea] 03 de 02 de 2010.
http://es.wikipedia.org/wiki/Base_de_datos

Wikimedia Foundation. 2009. Wikipedia. La Enciclopedia Libre [En línea] 20 de 04 de 2010.
<http://es.wikipedia.org/wiki/Framework>

GLOSARIO DE TÉRMINOS

Auditoría: Revisión de las actividades de una organización y de las operaciones para asegurar que éstas están siendo ejecutadas o están funcionando de acuerdo con los objetivos, el presupuesto, las reglas y normas. El objetivo de esta revisión es identificar, en intervalos regulares, desviaciones que pudieran necesitar una acción correctiva.

Base de Datos: Una base de datos o banco de datos (en inglés: database) es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. En este sentido, una biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta. En la actualidad, y debido al desarrollo tecnológico de campos como la informática y la electrónica, la mayoría de las bases de datos están en formato digital (electrónico), que ofrece un amplio rango de soluciones al problema de almacenar datos. (Wikimedia Foundation, 2009).

Casos de Uso: En ingeniería del software, un caso de uso es una técnica para la captura de requisitos potenciales de un nuevo sistema o una actualización de software. Cada caso de uso proporciona uno o más escenarios que indican cómo debería interactuar el sistema con el usuario o con otro sistema para conseguir un objetivo específico.

Componente de Control Interno: Los componentes del control interno de una entidad son: entorno de control, evaluación de riesgo, actividades de control, información y comunicación, y seguimiento. (COSO 1992).

Control: Cualquier acción tomada por la gerencia, el consejo y otras partes para gestionar el riesgo y aumentar las posibilidades de que los objetivos y metas se alcancen. La gerencia planifica, organiza y dirige la gestión con suficientes acciones para proporcionar una garantía razonable de que los objetivos y metas se alcanzan.

COSO: Comité de organizaciones patrocinadas, un grupo de varias organizaciones de contabilidad. En 1992 publicó un estudio significativo sobre el control interno llamado Control Interno-un marco integrado. El informe – que coincide con el llenado de CAECI - es muchas veces llamado Informe COSO.

Entidad: Define la ubicación física, la asignación, el uso, los datos de explotación de uno o más medios técnicos, así como los datos técnicos relacionados a cada uno de estos medios técnicos.

Evaluación de riesgo: La evaluación de riesgo es el proceso de identificación y análisis de los riesgos relevantes al alcanzar los objetivos de la entidad y determinar una respuesta apropiada.

Framework: Un framework, en el desarrollo de software, es una estructura de soporte definida, mediante la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto. Representa una arquitectura de software que modela las relaciones generales de las entidades del dominio. Provee una estructura y una metodología de trabajo la cual extiende o utiliza las aplicaciones del dominio. (Wikimedia Foundation, 2009).

PostgreSQL: Es un sistema de gestión de base de datos relacional orientada a objetos de software libre, publicado bajo la licencia BSD. Como muchos otros proyectos open source, el desarrollo de PostgreSQL no es manejado por una sola compañía sino que es dirigido por una comunidad de desarrolladores y organizaciones comerciales las cuales trabajan en su desarrollo. Dicha comunidad es denominada el PGDG (PostgreSQL Global Development Group). (Wikimedia Foundation, 2009)

Requerimientos Funcionales: Un requisito funcional define el comportamiento interno del software: cálculos, detalles técnicos, manipulación de datos y otras funcionalidades específicas que muestran cómo los casos de uso serán llevados a la práctica. Son complementados por los requisitos no funcionales, que se enfocan en cambio en el diseño o la implementación.

Requerimientos no Funcionales: Un requisito no funcional es, en la ingeniería de sistemas y la ingeniería de software, un requisito que especifica criterios que pueden usarse para juzgar la operación de un sistema en lugar de sus comportamientos específicos, ya que éstos corresponden a los requisitos funcionales. Los requisitos no funcionales más habituales son la estabilidad, la portabilidad y el costo.

RUP: Rational Unified Process.

TIC: Se denominan Tecnologías de la Información y las Comunicaciones (TIC), al conjunto de tecnologías que permiten la adquisición, producción, almacenamiento, tratamiento, comunicación, registro y

presentación de informaciones contenidas en señales de naturaleza acústica (sonidos), óptica (imágenes) o electromagnética (datos alfanuméricos).

UML: Es un estándar general para el propósito del lenguaje de modelado en el campo de la ingeniería de software. Incluye un grupo de técnicas de notación grafica para crear modelos abstractos de sistemas específicos. (Wikimedia Foundation, 2009)