

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

Facultad 3



**Modelamiento del Negocio e Ingeniería de
Requisitos del Sistema Integral de Documentación
e Información Judicial (SIDIJ).**

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

Autor (es): Yebel Fornaris Licea
Yoanna Ibañez Fernández

Tutor(a): Ing. Yanet Pérez Valcárcel
Co-Tutor: Ing. Heiler Fabars Corrales

La Habana, Cuba.
“Año 52 de la Revolución”

DECLARACIÓN DE AUTORÍA

Declaramos ser autores del presente trabajo y autorizamos a la facultad 3 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

Yoanna Ibañez Fernández

(Autor)

Yebel Fornaris Licea

(Autor)

Yanet Pérez Valcárcel

(Tutor)



RESUMEN

El Centro Nacional de Documentación e Información Judicial (CENDIJ) acordó con la Universidad de las Ciencias Informáticas (UCI) la construcción de un sistema informático con el objetivo de prestar un mejor servicio a sus clientes y agilizar sus actividades.

En el presente trabajo de diploma se realiza el modelamiento del negocio para lograr una mejor comprensión del problema y la Ingeniería de Requisitos para establecer las funcionalidades que el futuro sistema debe tener, logrando así satisfacer las expectativas del cliente.

Para ello se hizo un estudio de las etapas de la Ingeniería de Requisitos: elicitación, análisis, especificación, validación y gestión para su posterior puesta en práctica. Además se investigó sobre las metodologías de desarrollo, lenguajes de modelado, herramientas CASE y para el prototipado de interfaz de usuario con el objetivo de seleccionar las más adecuadas para la realización del trabajo.

Se elaboraron los artefactos: Modelo de Negocio, Especificación de Requisitos, Descripción de Casos de Uso del Sistema y se comprobó que cumplen con la calidad requerida mediante el uso de técnicas y métricas de validación.

Palabras clave: Ingeniería de Requisitos, procesos, metodología de desarrollo.



ÍNDICE

INTRODUCCIÓN	6
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	11
1.1 ¿Qué son los requerimientos de software?	11
1.1.1 Clasificación de los requerimientos	11
1.2 Ingeniería de Requisitos.....	13
1.2.1 Importancia de la Ingeniería de Requisitos	13
1.2.2 Etapas de la Ingeniería de Requisitos.....	14
1.2.3 Técnicas para la recopilación de requisitos	16
1.2.4 Técnicas de validación de requisitos	17
1.3 Metodologías de Desarrollo de Software.....	18
1.3.1 Rational Unified Process (RUP)	19
1.3.2 Extreme Programming (XP).....	21
1.3.3 SCRUM.....	22
1.3.4 Dynamic Systems Development Method (DSDM)	23
1.4 Lenguaje de modelado.....	23
1.4.1 Unified Modeling Language (UML).....	24
1.4.2 Business Process Modeling Notation (BPMN)	25
1.5 Herramientas CASE	26
1.5.1 Enterprise Architect (EA).....	26
1.5.2 Rational Rose.....	27
1.5.3 Visual Paradigm para UML (VP-UML)	27
1.6 Herramientas para el prototipado de interfaz de usuario	28
1.6.1 Microsoft Office Visio 2007	28
1.6.2 Axure RP.....	29
1.6.3 Visual Paradigm	30
Conclusiones	30
CAPÍTULO 2: MODELAMIENTO DEL NEGOCIO	31
2.1 Flujo de trabajo Modelamiento del Negocio.	31
2.2 Objetivos del Flujo Modelamiento del Negocio.....	31
2.3 Modelamiento del Negocio del CENDIJ.....	32
2.3.1 Actores del Negocio.....	32
2.3.2 Diagrama de Casos de Uso del Negocio.....	33
2.3.3 Trabajadores del Negocio	34

2.3.4 Realización de los Casos de Uso del Negocio.....	35
2.3.5 Modelo de Objetos del Negocio.....	46
2.3.6 Reglas del Negocio.....	49
Validación del Negocio	50
Conclusiones	50
CAPÍTULO 3: REQUERIMIENTOS DEL SISTEMA	51
3.1 Flujo de Trabajo Requerimientos	51
3.2 Objetivos del Flujo de Requerimientos	51
3.3 Requerimientos Funcionales y no Funcionales	52
3.4 Gestión de requisitos.....	59
3.5 Actores del sistema	60
3.6 Patrones de Casos de Uso	60
3.7 Diagrama de Casos de Uso del Sistema.....	63
3.7.1 Descripción de Casos de Uso del Sistema.....	64
3.8 Validación de Requisitos.....	70
3.8.1 Validación mediante Técnicas.....	70
3.8.2 Métrica para la calidad de la especificación de los requisitos de software.....	71
3.8.3 Métricas para la calidad de la funcionalidad del Diagrama de Casos de Uso del Sistema.....	72
Conclusiones	75
CONCLUSIONES GENERALES.....	76
RECOMENDACIONES	77
BIBLIOGRAFÍA	78
ANEXOS.....	¡Error! Marcador no definido.
GLOSARIO DE TÉRMINOS.....	80

INTRODUCCIÓN

Las Tecnologías de la Información y las Comunicaciones (TIC) contribuyen al mejoramiento de la vida humana debido a que constituyen una herramienta fundamental en el desarrollo científico del hombre. Este es partícipe directo en el creciente y cambiante proceso tecnológico de las últimas décadas que se encuentra enmarcado dentro de la revolución tecnológica, que está en constante dinámica evolutiva y es extraordinariamente innovadora.

En la actualidad se hace verdaderamente imponente la demanda ante la producción de software, donde es imprescindible llevar a cabo un proceso organizado de desarrollo de software para obtener un producto que responda a los requerimientos, el presupuesto y la planificación establecida. Además, las exigencias con respecto al desarrollo de software de calidad crece en la medida que el tamaño y complejidad de los sistemas es mayor; así como su vinculación con dispositivos de hardware u otros componentes electrónicos. Los sistemas toman mayor connotación arquitectónica y los requerimientos son más elevados en correspondencia a los nuevos avances en el mercado industrial.

La conveniente práctica de la Ingeniería de Software (IS) sustenta la idea de obtener un proceso de construcción de software que descansa sobre una organización de calidad y que abarque todo el ciclo de vida del desarrollo de software. Esto evitaría los errores de un proceso de desarrollo inmaduro, centrado en la etapa de implementación y no en todo el ciclo de vida, además de un software difícil de mantener, ya sea por su inaccesibilidad o por su alto costo. (1)

La parte más difícil de construir un sistema es precisamente saber qué construir. No existe otra parte del trabajo conceptual tan difícil como establecer los requerimientos técnicos detallados, incluyendo todas las interfaces con personas, máquinas y otros sistemas. Ninguna otra etapa del proceso afecta tanto el sistema si es hecha mal. Entonces, la tarea más importante que el ingeniero de software hace para el cliente es la extracción iterativa y el refinamiento de los requerimientos del producto. (2)

Es muy frecuente escuchar entre los conoedores del desarrollo de software que un gran número de los proyectos de software fracasan por no realizar una adecuada definición, especificación y gestión de los requerimientos.

La Ingeniería de Requisitos (IR) cumple un papel primordial en el proceso de producción de software, se enfoca en un área fundamental: la definición de lo que se desea producir. Su principal tarea consiste en la generación de especificaciones correctas que describan con

claridad, sin ambigüedades, en forma consistente y compacta, las necesidades de los usuarios o clientes; de esta manera, se pretende minimizar los problemas relacionados por la mala gestión de los requerimientos en el desarrollo de sistemas. A raíz de estos problemas, se requiere cada vez más perfeccionar el proceso de IR enfocando sus procedimientos y técnicas a la resolución y gestión de los requisitos, de forma sistémica a lo largo del ciclo de vida del desarrollo de software.

Cuba no está exenta a la producción de software por lo que ha fomentado la construcción de entidades que permiten una informatización del país para favorecer su desarrollo productivo y social. En este marco y como parte de los programas destinados a la Batalla de Ideas y el profundo proceso de transformaciones educacionales y sociales se crea la Universidad de las Ciencias Informáticas (UCI). Es una universidad productiva, cuya misión es producir software y servicios informáticos no solo para Cuba, sino también para el mundo a partir de la vinculación estudio – trabajo como modelo de formación.

Muchas empresas y organizaciones cubanas luchan cada día en aras de alcanzar un desarrollo tecnológico superior para la mejora de sus actividades laborales. Ejemplo de ello es el Centro Nacional de Documentación e Información Judicial (CENDIJ) que cuenta con una biblioteca que posee los fondos más ricos y completos del país en materia de Derecho y tiene documentos de gran valor histórico. Los fondos con los que cuenta son los siguientes: libros, folletos y publicaciones seriadas que se encuentran en formato papel y disposiciones del consejo de gobierno, legislaciones, sentencias, investigaciones y doctrinas en formato digital.

El centro brinda un conjunto de servicios al público como son: búsqueda de información en ambos formatos, préstamos de libros tanto internos como externos, así como el uso de la sala de lectura. Además de los servicios que se prestan también se realizan una serie de actividades como son: compra de libros, donaciones y procesamiento de la información.

Todo esto se realiza de forma manual y se genera un volumen significativo de documentación que es archivada, lo que trae como consecuencia lentitud en la prestación de los servicios afectando así la satisfacción de los clientes. Se dificulta el acceso a la información por lo que se requiere de una ejecución rápida, oportuna y precisa para facilitar la acertada toma de decisiones con la prontitud requerida. El proceso estadístico se torna complejo porque se hace necesario dedicar tiempo para realizar un análisis exhaustivo de toda la documentación que se obtiene diariamente en la organización. También la entidad cuenta con un registro de constancia de los fondos documentales existentes en la misma; si esta información sufriera alguna pérdida o daño se afectaría su control.

La UCI, en especial la facultad 3, de conjunto con el CENDIJ trabajan en el desarrollo de un software que contribuya a la informatización de los procesos que en esa institución se realizan. Debido a la necesidad de asegurar que el sistema cumpla con los servicios y funcionalidades que espera el usuario se requiere de una eficaz captura de requisitos por ser la pieza fundamental de un proyecto de desarrollo de software, marca el punto de partida para las siguientes actividades del mismo, sirve de base para verificar si se alcanzaron los objetivos establecidos en el proyecto y constituye el hilo conductor de todo el desarrollo del software.

Teniendo en cuenta los argumentos anteriormente expuestos se plantea el siguiente **problema de la investigación**: ¿Cómo lograr una visión común entre clientes y desarrolladores de los requisitos de software para el CENDIJ, que contribuya al diseño de un sistema para dicho centro?

Ante esta interrogante se identifica como **objeto de estudio** de la presente investigación: proceso de desarrollo de software. Con el propósito de proceder a la solución del problema planteado se precisa como **objetivo general**: desarrollar el modelamiento del negocio y la Ingeniería de Requisitos de software para el CENDIJ, para contribuir al diseño de un sistema para dicho centro. Como **campo de acción**: modelamiento del negocio y las etapas de elicitación, análisis, especificación, validación y gestión de la Ingeniería de Requisitos.

Se tiene la siguiente **idea a defender**: el desarrollo de un adecuado modelamiento del negocio e Ingeniería de Requisitos de software para el CENDIJ, contribuirá al diseño de un sistema para dicho centro.

Por la envergadura del problema se han establecido varios **Objetivos Específicos**:

- ❖ Elaborar el marco teórico de la investigación.
- ❖ Realizar el Modelo de Negocio para representar las actividades que se realizan en el Centro Nacional de Documentación e Información Judicial (CENDIJ).
- ❖ Obtener un listado de requerimientos de software que respondan a las funcionalidades que deberá cumplir el sistema a construir.
- ❖ Realizar el Modelo del Sistema donde queden reflejadas todas las actividades a informatizar.
- ❖ Validar la propuesta de solución.

Para dar cumplimiento a los objetivos planteados se hace necesario realizar una serie de **Tareas de investigación**:

- ❖ Estudio previo de las metodologías de desarrollo, lenguajes de modelado y herramientas CASE existentes actualmente.

- ❖ Estudio de las etapas de elicitación, análisis, especificación, validación y gestión de la Ingeniería de Requisitos.
- ❖ Selección de la metodología de desarrollo, lenguaje de modelado y herramienta CASE.
- ❖ Estudio de métricas para realizar la validación.
- ❖ Identificación de los procesos de negocio, con el objetivo de conocer las actividades fundamentales que se realizan en el Centro Nacional de Documentación e Información Judicial (CENDIJ).
- ❖ Realización del Modelo de Negocio, para agrupar actividades relacionadas con un objetivo común.
- ❖ Identificación de los requisitos de software que representarán las funcionalidades del sistema.
- ❖ Realización del Modelo del Sistema para agrupar actividades a informatizar.
- ❖ Validación del Modelo de Negocio.
- ❖ Validación del listado de requisitos obtenido.

Para la realización de las tareas expuestas anteriormente se utilizaron los siguientes métodos científicos de la investigación:

Métodos teóricos

Histórico-Lógico: Se utilizó para estudiar la historia y trayectoria de la Ingeniería de Requisitos.

Análítico-Sintético: Se utilizó para realizar un análisis detallado de varios elementos relacionados con el modelamiento del negocio e Ingeniería de Requisitos a partir de la consulta de bibliografía relacionada con el tema.

Modelación: Posibilitó la creación de modelos que representan abstracciones, con el objetivo de explicar cómo funcionan los procesos. Entre los modelos elaborados pueden encontrarse: modelo de objetos, diagrama de actividades, diagramas de casos de uso.

Métodos empíricos

Observación: Mediante este método se logró percibir de manera directa los problemas existentes en el Centro Nacional de Documentación e Información Judicial (CENDIJ).

Entrevista: Permitió la realización de entrevistas individuales al personal vinculado al Centro Nacional de Documentación e Información Judicial (CENDIJ), lográndose de esta manera conocer y entender los procesos del negocio que allí se efectúan.

Estructura capitular:

La presente investigación queda estructurada en tres capítulos:

Capítulo 1: Fundamentación Teórica

En este capítulo se abordarán temas relacionados con la Ingeniería de Requisitos como son: importancia, actividades que la componen así como algunas técnicas de recopilación y validación de requisitos. Se realizará un estudio para determinar la metodología de desarrollo de software, herramienta CASE y lenguaje de modelado que se utilizarán en el desarrollo de la investigación.

Capítulo 2: Modelamiento del Negocio

En este capítulo se describirán los procesos que se realizan en el CENDIJ, se identificarán actores, trabajadores, entidades del negocio y se obtendrán los artefactos necesarios que darán paso al próximo flujo de trabajo como son el diagrama de casos de uso del negocio, las realizaciones de los casos de uso así como los diagramas de actividades y de objetos del negocio.

Capítulo 3: Requerimientos del Sistema

En este capítulo se presentará el listado de requisitos de software que debe cumplir el sistema a construir, para ello se utilizarán los artefactos generados anteriormente y se aplicarán las actividades de la Ingeniería de Requisitos con el apoyo de sus técnicas más efectivas. Se determinarán los actores del sistema, el diagrama de casos de uso del sistema, descripción de casos de uso y se validarán los requisitos propuestos.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

En el presente capítulo se abordan temas fundamentales relacionados con los requerimientos de software, la definición de Ingeniería de Requisitos como pilar principal en todo proceso de desarrollo de software, sus principales actividades, técnicas para obtener requisitos del cliente así como las técnicas de validación. Se realiza un estudio del estado del arte para identificar las principales metodologías de desarrollo de software, lenguajes de modelado y herramientas CASE existentes en la actualidad.

1.1 ¿Qué son los requerimientos de software?

Los requerimientos son la base fundamental de cualquier software que se desee desarrollar, determinan las capacidades y cualidades que debe cumplir el software para garantizar una buena calidad. Todas las ideas que los clientes, usuarios y miembros del equipo de proyecto tengan acerca de lo que debe hacer el sistema, deben ser analizadas como candidatas a requisitos. A continuación se presentan algunas definiciones de requerimiento:

Ian Sommerville presenta una definición de “Requerimiento”:

- ❖ “Un requerimiento es simplemente una declaración abstracta de alto nivel de un servicio que debe proporcionar el sistema o una restricción de este”. (3)

La IEEE Standard Glossary of Software Engineering Terminology define requerimiento de software como: (4)

- ❖ Condición o capacidad que necesita un usuario para resolver un problema o lograr un objetivo.
- ❖ Condición o capacidad que tiene que ser alcanzada o poseída por un sistema o componente de un sistema para satisfacer un contrato, estándar, u otro documento impuesto formalmente.
- ❖ Una representación documentada de una condición o capacidad como en 1 o 2.

Debido a que incluye y aborda de manera amplia, directa y clara las definiciones de requerimiento funcional y no funcional que serán tratadas con posterioridad, la definición de requerimiento planteada por la IEEE será tomada en la presente investigación.

1.1.1 Clasificación de los requerimientos

Los requisitos se pueden clasificar en: **funcionales** y **no funcionales**.

Requisitos funcionales: son capacidades o condiciones que el sistema debe cumplir.

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

Los requisitos funcionales se mantienen invariables sin importar con que propiedades o cualidades se relacionen.

Requisitos no funcionales: son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable.

Existen múltiples categorías para clasificar a los requisitos no funcionales tales como:

❖ **Requisitos de software**

Este tipo de requerimiento especifica el software del que se debe disponer para utilizar el sistema que se construirá.

❖ **Requisitos de hardware**

Este tipo de requerimiento especifica los elementos de hardware de los que se debe disponer para utilizar el software de manera efectiva.

❖ **Restricciones en el diseño y la implementación**

Este tipo de requerimiento especifica o restringe la codificación o construcción de un sistema, son restricciones que han sido ordenadas y deben ser cumplidas estrictamente.

❖ **Requisitos de apariencia o interfaz externa**

Este tipo de requerimiento describe la apariencia del producto. Es importante destacar que no se trata del diseño de la interfaz en detalle sino que especifican cómo se pretende que sea la interfaz externa del producto.

❖ **Requisitos de seguridad**

Este tipo de requerimiento describe la manera en que se garantizará la disponibilidad, integridad y confidencialidad de la información.

❖ **Requisitos de usabilidad**

Estos requisitos describen los niveles apropiados de usabilidad, dados los usuarios finales del producto, para ello deben revisarse las especificaciones de los perfiles de usuarios y las clasificaciones de sus niveles de experiencia.

❖ Requisitos de soporte

Abarcan todas las acciones a tomar una vez que se ha terminado el desarrollo del software con motivos de asistir a los clientes de este, así como lograr su mejoramiento progresivo y evolución en el tiempo.

1.2 Ingeniería de Requisitos

La Ingeniería de Requisitos (IR) proporciona el mecanismo apropiado para entender lo que el cliente quiere, analizar las necesidades, evaluar la factibilidad y negociar una solución razonable con el mismo. A continuación se proponen algunos conceptos que han formulado diversos autores para describir qué es la Ingeniería de Requisitos y qué funciones cumple en el proceso de desarrollo de software.

Pressman define la “Ingeniería de Requerimientos como la ayuda a los ingenieros de software a entender mejor el problema en cuya solución trabajarán. Incluye el conjunto de tareas que conducen a comprender cuál será el impacto del software sobre el negocio, qué es lo que el cliente quiere y cómo interactuarán los usuarios finales con el software”. (5)

Rational Software Corporation define la “Ingeniería de Requerimientos como el enfoque sistémico para recolectar, organizar y documentar los requerimientos del sistema; es también el proceso que establece y mantiene acuerdos sobre los cambios de requerimientos, entre los clientes y el equipo del proyecto”. (6)

Boehm plantea que la “Ingeniería de Requerimientos es la disciplina para desarrollar una especificación completa, consistente y no ambigua, la cual servirá como base para acuerdos comunes entre todas las partes involucradas y en donde se describen las funciones que realizará el sistema”. (6)

Se concluye que la Ingeniería de Requisitos es una disciplina que provee al ingeniero una vía para comprender las tareas relacionadas con la determinación de las necesidades o condiciones del sistema. Se utiliza para definir todas las actividades involucradas en el descubrimiento, documentación y mantenimiento de los requerimientos para un producto determinado. Su principal meta es entregar una especificación de requisitos de software correcta y completa, de manera que sea consistente con las necesidades de la organización y con las expectativas de los clientes.

1.2.1 Importancia de la Ingeniería de Requisitos

Los principales beneficios que se obtienen al aplicar una correcta Ingeniería de Requisitos son:
(7)

- ❖ Permite gestionar las necesidades del proyecto en forma estructurada: cada actividad de la IR consiste en una serie de pasos organizados y bien definidos.
- ❖ Mejora la capacidad de predecir cronogramas de proyectos, así como sus resultados: la IR proporciona un punto de partida para controles subsecuentes y actividades de mantenimiento, tales como estimación de costos, tiempo y recursos necesarios.
- ❖ Disminuye los costos y retrasos del proyecto: es sabido que reparar errores por un mal desarrollo no descubierto a tiempo, es sumamente caro; especialmente aquellas decisiones tomadas durante la IR, por ser una de las etapas de mayor importancia en el ciclo de desarrollo de software y de las primeras en llevarse a cabo.
- ❖ Mejora la calidad del software: la calidad en el software tiene que ver con cumplir un conjunto de requerimientos.
- ❖ Mejora la comunicación entre equipos: la especificación de requerimientos representa una forma de consenso entre clientes y desarrolladores. Si este consenso no ocurre, el proyecto no será exitoso.
- ❖ Evita rechazos de usuarios finales: la Ingeniería de Requerimientos obliga al cliente a considerar sus requerimientos cuidadosamente y revisarlos dentro del marco del problema, involucrándolo durante todo el desarrollo del proyecto.

1.2.2 Etapas de la Ingeniería de Requisitos

La Ingeniería de Requisitos propone actividades que rigen todo el proceso a partir de la identificación de los requerimientos hasta su exitosa validación. Tienen por objetivo la determinación de las necesidades del sistema, la adquisición por parte del equipo de desarrollo de la información necesaria para desarrollar un producto de calidad y en definitiva, la comunicación inicial entre el grupo de expertos y los clientes y usuarios. Las mismas serán mencionadas a continuación: (7)

Elicitación

Esta fase representa el comienzo de cada ciclo. Extracción es el nombre comúnmente dado a las actividades involucradas en el descubrimiento de los requerimientos del sistema. Aquí, los analistas de requerimientos deben trabajar junto al cliente para descubrir el problema que el sistema debe resolver, los diferentes servicios que el sistema debe prestar, las restricciones que se pueden presentar, etc. La aceptación del sistema dependerá de cuán bien este satisfaga las necesidades del cliente, por lo cual es importante que la extracción sea efectiva.

Análisis

Sobre la base de la extracción realizada previamente, comienza esta fase, la cual se enfoca en descubrir problemas con los requerimientos del sistema identificados hasta el momento.

Usualmente se hace un análisis después de haber producido un bosquejo inicial del documento de requerimientos; en esta etapa se leen los requerimientos, se investigan, se intercambian ideas con el resto del equipo, se resaltan los problemas, se buscan alternativas y soluciones y luego se fijan reuniones con el cliente para discutir los requerimientos.

Especificación

En esta fase se documentan los requerimientos acordados con el cliente en un nivel apropiado de detalle. En la práctica, esta etapa se realiza conjuntamente con el análisis, se puede decir que la especificación es el "pasar en limpio" el análisis realizado previamente aplicando técnicas y/o estándares de documentación, como la notación UML (Lenguaje Unificado de Modelado), que es un estándar para el modelado orientado a objetos, por esta razón los casos de uso y la obtención de requerimientos basada en casos de uso se utiliza cada vez más para la obtención de requerimientos.

Validación

El objetivo de la etapa de validación es ratificar los requerimientos, es decir, verificar todos los requerimientos que aparecen en el documento de especificación para asegurarse que representan una descripción aceptable del sistema que se debe implementar. Esto implica verificar que los requerimientos sean consistentes y que estén completos.

Gestión

La gestión de requisitos es un conjunto de actividades que ayudan al equipo de trabajo a identificar, controlar y seguir los requisitos y los cambios a estos en cualquier momento mientras se desarrolla el proyecto. La gestión formal de requisitos se inicia sólo para proyectos grandes, los cuales tienen cientos de requisitos identificables. En los proyectos pequeños esta función de la Ingeniería de Requisitos es menos formal. (5)

Esta etapa se desarrolla durante toda la vida del producto y plantea que se debe mantener el conjunto de requisitos aprobados y actualizados a lo largo de todo el proyecto. Mediante la gestión se logra:

- ❖ Determinar los requisitos y almacenarlos.
- ❖ Crear la línea base de los requisitos, entendida como el conjunto de requisitos acordados.
- ❖ Gestionar las peticiones de cambio.
- ❖ Mantener las trazas entre los requisitos y otros elementos del proceso de desarrollo.

1.2.3 Técnicas para la recopilación de requisitos

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 IR ha trabajado desde hace años en desarrollar técnicas de recopilación que no son más que procedimientos que contribuyen a recopilar, indagar y averiguar sobre algún tema en específico y que permiten hacer este proceso de una forma más eficiente y precisa.

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

Entrevistas: resulta una técnica muy aceptada dentro de la IR y su uso está ampliamente extendido. Las entrevistas le permiten al analista tomar conocimiento del problema y comprender los objetivos de la solución buscada. A través de esta técnica el equipo de trabajo se acerca al problema de una forma natural.

Existen muchos tipos de entrevistas y son diversos los autores que han trabajado en definir su estructura y dar guías para su correcta realización. Básicamente, la estructura de la entrevista abarca tres pasos: identificación de los entrevistados, preparación de la entrevista, realización de la entrevista y documentación de los resultados. A pesar de que las entrevistas son esenciales en el proceso de la captura de requisitos y con su aplicación el equipo de desarrollo puede obtener una amplia visión del trabajo y las necesidades del usuario, es necesario destacar que no es una técnica sencilla de aplicar. Requiere que el entrevistador sea experimentado y tenga capacidad para elegir bien a los entrevistados y obtener de ellos toda la información posible en un período de tiempo limitado. (8)

Cuestionario: Esta técnica requiere que el analista conozca el ámbito del problema en el que se trabaja. Consiste en redactar un documento con preguntas cuyas respuestas sean cortas y concretas, o incluso cerradas por unas cuantas opciones en el propio cuestionario. Este cuestionario será completado por el grupo de personas entrevistadas o simplemente para recoger información en forma independiente de una entrevista (8). Una de las ventajas que tiene este método es que permite relacionarse con un gran grupo de personas para conocer varios aspectos sobre algún tema en específico, aunque no ofrece la posibilidad de observar las expresiones de quienes lo responden.

Sistemas existentes: esta técnica consiste en analizar distintos sistemas desarrollados que estén relacionados con el sistema a ser construido. Por un lado, se pueden analizar las interfaces de usuario, en dependencia del tipo de información que se maneja y cómo es manejada, por otro lado también es útil analizar las distintas salidas que los sistemas producen porque siempre pueden surgir nuevas ideas sobre la base de estas. (7)

Mapas conceptuales: Los mapas conceptuales son grafos en los que los vértices representan conceptos y las aristas representan posibles relaciones entre dichos conceptos. Estos grafos de relaciones se desarrollan con el usuario y sirven para aclarar los conceptos relacionados con el sistema a desarrollar. Son muy usados dentro de la Ingeniería de Requisitos pues son fáciles de entender por el usuario, más aún si el equipo de desarrollo hace el esfuerzo de elaborarlo en el lenguaje de este. Sin embargo, deben ser usados con cautela porque en algunos casos pueden ser muy subjetivos y pueden llegar a ser ambiguos en casos complejos si no se acompaña de una descripción textual. (8)

Lluvia de ideas (Brainstorm): Es 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. El grupo de personas que participa en estas reuniones no debe ser muy numeroso (máximo 10 personas), una de ellas debe asumir el rol de moderador de la sesión, pero sin carácter de controlador. Como técnica de captura de requisitos es sencilla de usar y de aplicar. Además suele ofrecer una visión general de las necesidades del sistema, pero normalmente no sirve para obtener detalles concretos del sistema, por lo que suele aplicarse en los primeros encuentros. (8)

Las reglas básicas a seguir son: (7)

- ❖ Los participantes deben pertenecer a distintas disciplinas y, preferentemente, deben tener mucha experiencia. Esto trae aparejado la obtención de una mayor cantidad de ideas creativas.
- ❖ Conviene suspender el juicio crítico y se debe permitir la evolución de cada una de las ideas, porque si no se crea un ambiente hostil que no alienta la generación de ideas.
- ❖ Por más locas o salvajes que parezcan algunas ideas, no se las debe descartar, porque luego de maduras probablemente se tornen en un requerimiento sumamente útil.
- ❖ A veces ocurre que una idea resulta en otra idea, y otras veces se puede relacionar varias ideas para generar una nueva.
- ❖ Escribir las ideas sin censura.

1.2.4 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. Es necesario asegurar que el análisis realizado y los resultados obtenidos sean correctos, pues un levantamiento de requisitos con errores que no se detecten a tiempo, además de conducir a resultados inesperados, provoca costos excesivos y gran pérdida de tiempo.

Algunas técnicas que pueden aplicarse para la validación de requisitos son: (8)

- ❖ **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.
- ❖ **Matrices de trazabilidad:** esta técnica consiste en marcar los objetivos del sistema y chequearlos contra los requisitos del mismo. Es necesario determinar qué objetivos cubre cada requisito, de esta forma se podrán detectar inconsistencias u objetivos no cubiertos.
- ❖ **Auditorías:** la revisión de la documentación con esta técnica consiste en un chequeo de los resultados contra una lista 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. Esta técnica tiene el problema de que el usuario debe entender que lo que ve es un prototipo y no el sistema final.

Selección de las técnicas a utilizar

En la presente investigación se selecciona la técnica de validación prototipos con el objetivo de que el usuario tenga una visión temprana de cómo será el sistema y las funcionalidades que este le brindará. También se utilizará la matriz de trazabilidad para que se tengan en cuenta todos los requisitos propuestos a la hora de implementación.

1.3 Metodologías de Desarrollo de Software

Si se quiere construir un software de alta calidad, en el tiempo planificado y con los costos establecidos es necesario enfocarse en trabajar de forma organizada, donde se controle y documente todo lo que tenga relación con el proyecto en cuestión y puedan eliminarse los riesgos que podrían presentarse durante el desarrollo del mismo, lo cual no podría lograrse sin el empleo de una metodología eficaz que se adapte a las características propias del software que se desarrolla. La finalidad de una metodología de desarrollo es garantizar la eficacia y la eficiencia en el proceso de generación de software.

Las metodologías de desarrollo de software se categorizan en dos grandes grupos: las tradicionales que se centran especialmente en el control del proceso, establecen rigurosamente las actividades involucradas, los artefactos que se deben producir, las herramientas y notaciones que se usarán; y las metodologías ágiles las cuales dan mayor valor al individuo, a la colaboración con el cliente y al desarrollo incremental del software con iteraciones muy

cortas. Son usadas en proyectos con requisitos muy cambiantes y cuando se exige reducir drásticamente los tiempos de desarrollo pero sin afectar la calidad.

Las metodologías de desarrollo de software aplican un proceso disciplinado durante todo el ciclo de vida pues indican en cada etapa lo que debe hacerse. No existe una metodología de software universal. Por tanto su elección para la utilización en el desarrollo de un producto de software, debe basarse en las características específicas del proyecto. Se hace necesario además determinar previamente el alcance del proyecto, así como su complejidad.

A continuación se mencionan algunas de las metodologías de desarrollo de software más usadas en el mundo y se da una breve explicación sobre las mismas.

1.3.1 Rational Unified Process (RUP)

Rational Unified Process (Proceso Unificado de Desarrollo) es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema software. Sin embargo, RUP es más que un simple proceso, es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas software, para diferentes áreas de aplicación, tipos de organizaciones, niveles de aptitud y tamaños de proyecto. (9)

El ciclo de vida de RUP se caracteriza por ser:

Dirigido por casos de uso: los casos de uso reflejan lo que los usuarios necesitan y desean. A partir de aquí los casos de uso guían el proceso de desarrollo.

Centrado en la arquitectura: la arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben de estar de acuerdo, por lo que describe los elementos del modelo que son más importantes para su construcción.

Iterativo e incremental: es el proceso de construir sistemas mediante aproximaciones que se acercan progresivamente a la solución ideal. De esta forma se obliga a identificar los riesgos del proyecto en etapas tempranas y es un enfoque de descubrimiento, invención e implementación continuos donde cada iteración termina en una forma predecible y repetible.

RUP en su modelación define como sus principales elementos:

Trabajadores: define el comportamiento y responsabilidades de un individuo, grupo de individuos, sistema informatizado o máquina, que trabajan en conjunto como un equipo. Ellos realizan las actividades y son propietarios de elementos.

Actividades: es una tarea que tiene un propósito claro, es realizada por un trabajador y manipula elementos.

Artefactos: productos tangibles del proyecto que son producidos, modificados y usados por actividades. Pueden ser modelos, elementos dentro del modelo, código fuente y ejecutables.

Flujo de Actividades: secuencia de actividades realizadas por trabajadores y que produce un resultado de valor observable.

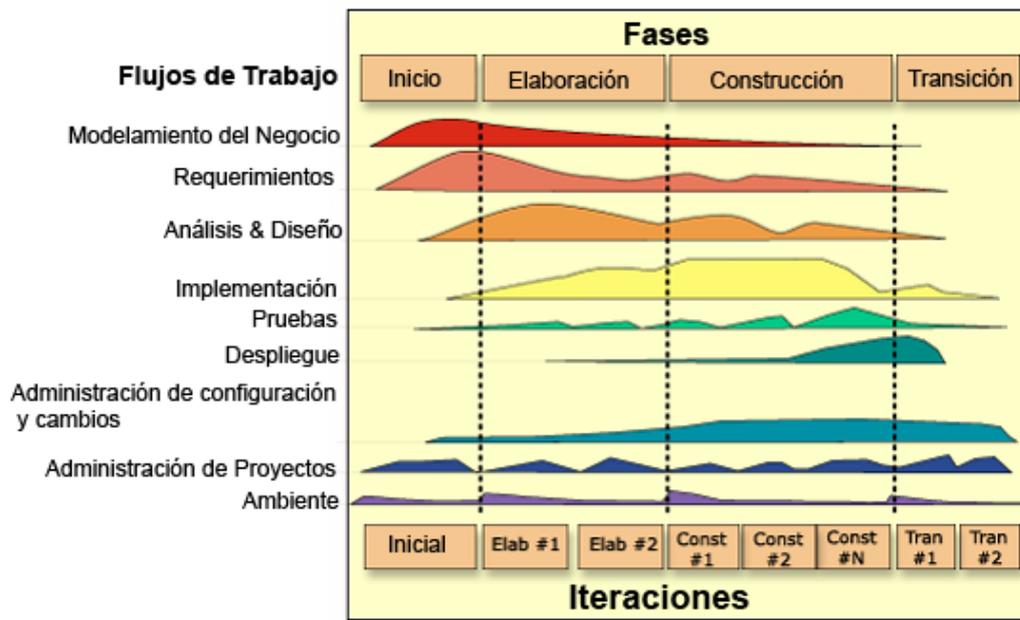


Figura 1: Fases y flujos de Rational Unified Process (RUP)

RUP consta de cuatro fases o etapas que son: (3)

Inicio: el objetivo de la fase de inicio es establecer un caso de negocio para el sistema. Se deben identificar todas las entidades externas (personas y sistemas) que interactúan con el sistema y definir estas interacciones. Esta información se utiliza para evaluar el aporte que el sistema hace al negocio. Si dicho aporte es de poca importancia, se puede cancelar el proyecto después de esta fase.

Elaboración: los objetivos de la fase de elaboración son desarrollar una comprensión del dominio del problema, establecer un marco de trabajo arquitectónico para el sistema, desarrollar el plan del proyecto e identificar los riesgos clave del proyecto. Al terminar esta fase, se debe tener un modelo de los requerimientos del sistema (se especifican los casos de uso UML), una descripción arquitectónica y un plan de desarrollo del software.

Construcción: la fase de construcción fundamentalmente comprende el diseño del sistema, la programación y las pruebas. Durante esta fase se desarrollan e integran las partes del sistema. Al terminar esta fase, se debe tener un sistema software operativo y la documentación correspondiente lista para entregarla a los usuarios.

Transición: la fase final de RUP se ocupa de mover el sistema desde la comunidad de desarrollo a la comunidad del usuario y hacerlo trabajar en un entorno real. Esto se deja de lado en la mayor parte de los modelos de procesos del software pero es, en realidad, una actividad de alto costo y a veces problemática. Al terminar esta fase, se debe tener un sistema software documentado que funcione correctamente en su entorno operativo.

Además de las fases, RUP propone nueve flujos de trabajo en los que se definen las secuencias de actividades, quiénes las deben desarrollar y los artefactos a generar. Los flujos de trabajo son:

Modelamiento del negocio: describe los procesos de negocio, se identifica quiénes participan y las actividades que requieren informatización.

Requerimientos: define qué es lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen.

Análisis y diseño: describe cómo se realizará el sistema a partir de los requerimientos identificados, por lo que indica con precisión lo que se debe programar.

Implementación: define cómo se organizan las clases y objetos en componentes, cuáles nodos se utilizarán, la ubicación en ellos de los componentes y la estructura de capas de la aplicación.

Prueba: busca los defectos a lo largo del ciclo de vida.

Despliegue: produce release del producto y realiza actividades (empaquete, instalación, asistencia a usuarios, etc.) para entregar el software a los usuarios finales.

Administración de configuración y cambios: describe cómo controlar los elementos producidos por todos los integrantes del equipo de proyecto en cuanto a: utilización/actualización concurrente de elementos, control de versiones, etc.

Administración del proyecto: involucra actividades con las que se busca producir un producto que satisfaga las necesidades de los clientes.

Ambiente: contiene actividades que describen los procesos y herramientas que soportarán el equipo de trabajo del proyecto; así como el procedimiento para implementar el proceso en una organización.

1.3.2 Extreme Programming (XP)

Extreme Programming o Programación Extrema es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promueve el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propicia un buen clima de trabajo. Se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para

proyectos con requisitos imprecisos, muy cambiantes, y donde existe un alto riesgo técnico. (10)

Utiliza un enfoque orientado a objetos como su paradigma de desarrollo preferido. Abarca un conjunto de reglas y prácticas que ocurren en el contexto de cuatro actividades del marco de trabajo: planeación, diseño, codificación y pruebas.

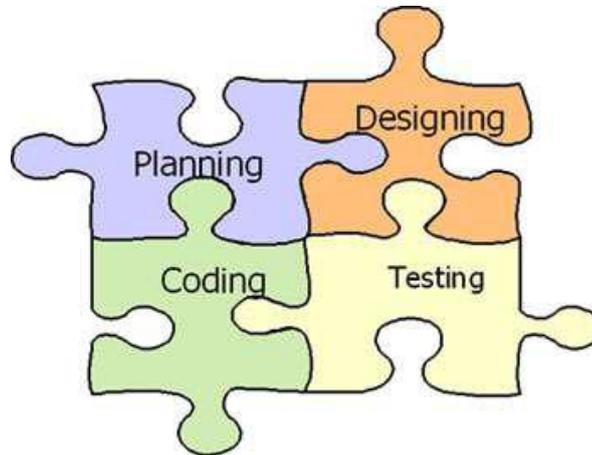


Figura 2: Actividades de Extreme Programming

Características de la metodología de desarrollo XP: (11)

- ❖ Pruebas Unitarias: se basa en las pruebas realizadas a los principales procesos, de tal manera que se adelante en algo hacia el futuro y se hagan pruebas de las fallas que pudieran ocurrir. Es como si se adelantara a obtener los posibles errores.
- ❖ Refabricación: se basa en la reutilización de código, para lo cual se crean patrones o modelos estándares, siendo más flexible al cambio.
- ❖ Programación en pares: una particularidad de esta metodología es que propone la programación en pares, la cual consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo. Cada miembro lleva a cabo la acción que el otro no hace en ese momento. Es como el chofer y el copiloto: mientras uno conduce, el otro consulta el mapa.

1.3.3 SCRUM

Desarrollada por Ken Schwaber, Jeff Sutherland y Mike Beedle. Define un marco para la gestión de proyectos, que se ha utilizado con éxito durante los últimos 10 años. Está especialmente indicada para proyectos con un rápido cambio de requisitos. Sus principales características se pueden resumir en dos. El desarrollo de software se realiza mediante iteraciones, denominadas sprints, con una duración de 30 días. El resultado de cada sprint es un incremento ejecutable que se muestra al cliente. La segunda característica importante son

las reuniones a lo largo del proyecto, entre ellas destaca la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración. (10)

Principios de SCRUM

- ❖ Colaboración estrecha con el cliente.
- ❖ Predisposición y respuesta al cambio.
- ❖ Desarrollo incremental con entregas funcionales frecuentes.
- ❖ Comunicación verbal directa entre los implicados en el proyecto.
- ❖ Motivación y responsabilidad de los equipos por la auto-gestión, auto-organización y compromiso.
- ❖ Simplicidad. Supresión de artefactos innecesarios en la gestión del proyecto.

1.3.4 Dynamic Systems Development Method (DSDM)

Dynamic Systems Development Method o Método de Desarrollo de Sistemas Dinámicos define el marco para desarrollar un proceso de producción de software. Nace en 1994 con el objetivo de crear una metodología unificada. Sus principales características son: es un proceso iterativo e incremental, el equipo de desarrollo y el usuario trabajan juntos. Propone cinco fases: estudio de viabilidad, estudio del negocio, modelado funcional, diseño y construcción e implementación. Las tres últimas son iterativas, además de existir realimentación a todas las fases. (10)

Selección de la metodología

Luego de realizar un estudio de algunas metodologías de desarrollo de software existentes en la actualidad, se llega a la conclusión de que RUP es la más indicada para guiar el proceso de desarrollo del sistema que se desea implementar debido a que provee un marco de proceso adaptable a las necesidades y características del software. Es utilizada frecuentemente en la Universidad de las Ciencias Informáticas, por esta razón el conocimiento que se tiene de la misma es profundo. Tiene como particularidad que cada iteración es superior a la anterior debido a que en cada una se corrigen y refinan los artefactos generados y se obtiene una detallada documentación. RUP constituye además uno de los estándares internacionales de mayor aceptación, soporta varias herramientas CASE y permite el trabajo en equipo.

1.4 Lenguaje de modelado

En la construcción de un software los lenguajes y notación de modelado juegan un papel primordial. Mediante la utilización de los mismos se pueden crear distintos modelos y diagramas que permitan un mayor entendimiento de cómo debe funcionar el sistema por parte de todo el equipo de desarrollo de software.

Tanto los lenguajes y notaciones, como las herramientas de software para la modelación, han creado un canal de comunicación entre las partes involucradas. En un proyecto puede escogerse la utilización de uno u otro lenguaje o notación, o incluso varios de ellos, pero cuando se escoge uno para elaborar un modelo debe cumplirse cabalmente con sus reglas.

Algunos de los lenguajes que permiten el modelado de procesos de una entidad son:

- ❖ Unified Modeling Language (UML)
- ❖ Business Process Modeling Notation (BPMN)

1.4.1 Unified Modeling Language (UML)

Unified Modeling Language (Lenguaje Unificado de Modelado) es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra gran cantidad de software. (12) Es una especificación de notación orientada a objetos y describe lo que supuestamente hará un sistema, pero no dice cómo implementar el mismo.

UML proporciona una forma estándar de representar los planos de un sistema, y comprende tanto elementos conceptuales, como los procesos de negocio y las funciones del sistema. Permite una comunicación sencilla y rápida entre desarrolladores y clientes del software que desarrolla y simplifica el proceso complejo de análisis y diseño de software, facilitando un plano para la construcción. Además provee ventajas en la representación de los artefactos generados durante el ciclo de vida que propone el Proceso Unificado de Desarrollo del Software.

Para poder representar correctamente un sistema, UML ofrece una amplia variedad de diagramas para visualizar el sistema desde varias perspectivas. A continuación se mencionan los mismos:

- ✓ Diagramas de Casos de Uso para modelar los procesos.
- ✓ Diagramas de Secuencia para modelar el paso de mensajes entre objetos.
- ✓ Diagramas de Colaboración para modelar interacciones entre objetos.
- ✓ Diagramas de Estado para modelar el comportamiento de los objetos en el sistema.
- ✓ Diagramas de Actividad para modelar el comportamiento de los Casos de Uso, objetos u operaciones.
- ✓ Diagramas de Clases para modelar la estructura estática de las clases en el sistema.
- ✓ Diagramas de Objetos para modelar la estructura estática de los objetos en el sistema.
- ✓ Diagramas de Componentes para modelar componentes.
- ✓ Diagramas de Despliegue para modelar la distribución del sistema.

Las funciones de UML se pueden sintetizar en: (13)

- ❖ Visualizar: UML permite expresar de una forma gráfica un sistema de forma que otro lo puede entender.
- ❖ Especificar: UML permite especificar cuáles son las características de un sistema antes de su construcción.
- ❖ Construir: a partir de los modelos especificados se pueden construir los sistemas diseñados.
- ❖ Documentar: los propios elementos gráficos sirven como documentación del sistema desarrollado que pueden servir para su futura revisión.

1.4.2 Business Process Modeling Notation (BPMN)

Business Process Modeling Notation (Notación de Modelado de Procesos de Negocio) es una notación gráfica que describe la lógica de los pasos de un proceso de negocio. Esta notación ha sido especialmente diseñada para coordinar la secuencia de los procesos y los mensajes que fluyen entre los participantes de las diferentes actividades. Proporciona un lenguaje común para que las partes involucradas puedan comunicar los procesos de forma clara, completa y eficiente. De esta forma BPMN define la notación y semántica de un Diagrama de Procesos de Negocio (Business Process Diagram, BPD). (14)

BPMN se basa en una técnica de grafos de flujo para crear modelos gráficos de operaciones de procesos de negocio. Un modelo de procesos de negocio, es una red de objetos gráficos, que son actividades y controles de flujo que definen su orden de rendimiento. Se compone de un conjunto de elementos gráficos que facilitan un diagrama entendible tanto por audiencias de negocios como técnicas.

Servir como soporte para la gestión por procesos es su objetivo principal y facilita de esta manera el rápido entendimiento de los procesos de negocio tanto al analista de negocio como a los desarrolladores técnicos responsables de implementar la tecnología que ejecutará estos procesos y personas que ejecutan, monitorean y supervisan los mismos.

BPMN crea un puente estandarizado para disminuir la brecha entre los procesos de negocio y la implementación de estos. Permite además modelar los procesos de una manera unificada y estandarizada al favorecer un entendimiento a todo el personal de una organización. Con su uso es posible detallar qué actividades son automáticas, qué actividades son realizadas con ayuda de un sistema, o cuáles actividades son realizadas manualmente.

Selección del lenguaje

Después de realizar el estudio de algunos de los lenguajes de modelado existentes se decidió utilizar UML por ser un lenguaje de uso universal, consolidado, fácil de aprender y usar. Es una notación gráfica muy expresiva que permite representar en mayor o menor medida todas las fases de un proyecto informático y prácticamente todas las herramientas CASE y de desarrollo lo han adaptado como lenguaje de modelado. Con el uso de UML se obtiene una notable efectividad y productividad en labores de diseño arquitectónico y mantenimiento.

1.5 Herramientas CASE

Las herramientas CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Computadora) son un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del ciclo de vida de desarrollo de un software. Fueron desarrolladas para automatizar los procesos y facilitar las tareas de coordinación de los eventos que necesitan ser mejorados en el ciclo de desarrollo de software.

El propósito de las herramientas CASE es apoyar y facilitar el desarrollo de software. Proporcionan al ingeniero la posibilidad de informatizar actividades manuales y de mejorar su visión general de la ingeniería. Ayudan a asegurar la calidad de un producto desde su diseño antes de construirlo y uno de sus objetivos más importantes es conseguir la generación automática de programas desde una especificación a nivel de diseño.

A continuación se analizan algunas de las herramientas CASE utilizadas en la actualidad:

1.5.1 Enterprise Architect (EA)

Enterprise Architect combina el poder de la última especificación UML 2.1 con alto rendimiento e interfaz intuitiva, de esta manera proporciona modelado avanzado al escritorio para el equipo de desarrollo. Puede equipar al equipo entero, incluye analistas, evaluadores, administradores de proyectos, personal del control de calidad, equipo de desarrollo y más, por una fracción del costo de algunos productos competitivos. (15)

Es una herramienta multiusuario diseñada para trabajar sobre el sistema operativo Windows y ayudar a construir software robusto y fácil de mantener. Cubre el ciclo completo de desarrollo de software, en etapas como captura de requerimientos, análisis, diseño, pruebas y mantenimiento. Soporta la construcción de modelos de sistemas de software rigurosos y donde es posible mantener la trazabilidad de manera consistente en un ambiente fácil de usar, rápido y flexible.

Entre sus características principales se destacan:

- ✓ Posee generación de documentación de gran calidad y herramientas de reportes con un editor de plantilla.
- ✓ Soporta transformaciones de Arquitectura Dirigida por Modelos (MDA) a partir del uso de plantillas con transformaciones fáciles de editar y desarrollar.
- ✓ Ayuda a la visualización de aplicaciones al permitir ingeniería inversa de esquemas de repositorios de base de datos y lenguajes de desarrollo de software como: C++, C#, Java, Delphi, VB.Net, Visual Basic y PHP.

1.5.2 Rational Rose

Es una de las más poderosas herramientas de modelado visual para el análisis y diseño de sistemas basados en objetos. Brinda soporte al modelado visual con UML al permitir a los arquitectos de software y desarrolladores visualizar el sistema completo mediante el uso de un lenguaje común. Está basada en la metodología de desarrollo RUP y utiliza un proceso de desarrollo iterativo controlado. Cubre todo el ciclo de vida de un proyecto y es un software propietario que soporta como única plataforma Windows.

Ayuda a la comunicación entre los miembros del equipo, a monitorear el tiempo de desarrollo y a entender el entorno de los sistemas. Permite a los diseñadores modelar sus componentes e interfaces en forma individual y luego unirlos con otros componentes del proyecto. Soporta la construcción de componentes, la posibilidad de generar código en varios lenguajes a partir de un diseño en UML y proporciona mecanismo para realizar ingeniería inversa.

1.5.3 Visual Paradigm para UML (VP-UML)

Visual Paradigm para UML (VP-UML) es una herramienta profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una rápida construcción de aplicaciones de calidad y a un menor costo. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. La herramienta CASE también proporciona abundantes tutoriales, demostraciones interactivas y proyectos UML. (16)

VP-UML se encuentra disponible en muchas plataformas por lo que se puede utilizar en sistemas operativos como Windows y Linux. Es una herramienta que provee el modelado de procesos del negocio, además de mapeo de objetos relacionales para los lenguajes de programación Java, .NET y PHP. Permite la generación de bases de datos, proporciona un

entorno unificado de diseño de software para el analista de sistemas y desarrollador de software para analizar, diseñar y mantener aplicaciones.

Permite diseñar nuevas notaciones o incorporar formas personalizadas mediante el uso de símbolos incrustados en forma de editor, la importación de imágenes o íconos, con capacidad de incorporar dibujos de Visio en cualquier diagrama UML. Posibilita el análisis textual que es una técnica útil y práctica para la captura de los requisitos del sistema, la identificación de clases candidatas, así como importar diagramas creados previamente por Rational Rose y ERWin.

Selección de la herramienta CASE

Teniendo en cuenta las características antes expuestas de las herramientas analizadas y mediante un profundo estudio de las mismas, se llega a la conclusión que se utilizará como herramienta de desarrollo de software Visual Paradigm para UML por soportar múltiples plataformas y permitir un diseño de mejores aplicaciones a un menor costo. Presenta una interfaz de uso intuitiva y con muchas facilidades a la hora de modelar los diagramas, además de su robustez, usabilidad y portabilidad. Brinda la posibilidad de realizar ingeniería inversa y flexibilidad para integrarse con los principales entornos de desarrollo como el Eclipse y el NetBeans.

1.6 Herramientas para el prototipado de interfaz de usuario

A medida que crece la complejidad y el tamaño de un proyecto, la coordinación se dificulta debido al incremento en la comunicación entre los ingenieros de software, administradores y clientes. Para ello se hace necesaria la aplicación de herramientas para el prototipado de interfaz de usuario debido a que su objetivo principal es mostrar al usuario, desde los momentos iniciales del diseño, el aspecto que tendrá el sistema una vez desarrollado. Ello facilitará la aplicación de los cambios que se consideren necesarios, todavía en la fase de diseño. A continuación se abordan algunos elementos característicos de estas herramientas.

1.6.1 Microsoft Office Visio 2007

Office Visio 2007 permite documentar, diseñar y comprender de forma visual el estado de los sistemas y procesos empresariales con una gran variedad de diagramas: diagramas de flujo de proceso empresarial, de red, de flujo de trabajo, modelos de bases de datos, diagramas de software, entre otros. Para sacar el máximo partido de los mismos se pueden vincular a datos subyacentes para proporcionar una imagen más completa. Office Visio 2007 permite comunicarse con diagramas para maximizar el impacto en los destinatarios de forma más efectiva que mediante palabras y cifras. (17)

Permite examinar de forma visual información compleja para identificar tendencias, excepciones y puntos de vista claves. Constituye una herramienta para analizar, desglosar y crear varias vistas de datos empresariales a fin de obtener una mejor visión empresarial. Con la utilización de su biblioteca de íconos y marcas se puede realizar un seguimiento de las tendencias y marcar excepciones que permiten resaltar fácilmente problemas claves para llevar a cabo acciones efectivas. Brinda la posibilidad de guardar los diagramas y gráficos generados en distintos formatos como: JPG, GIF, HTML y PDF, esto facilita que se puedan compartir incluso con personas que no dispongan de Visio.

Algunas características de la herramienta son:

- ❖ Proporciona una plantilla “Interfaz de usuario de Windows XP” que permite la creación de prototipos con formas diseñadas según las directrices de Microsoft Windows XP.
- ❖ Incluye 2 paletas que son las que básicamente permiten la confección de prototipos de interfaz de usuario: “Windows and Dialogs (US units)” y “Common Controls (US units)”, las cuales están compuestas por varios componentes, entre ellos paneles, cajas de texto y botones, que facilitan la modelación de los prototipos.
- ❖ Sistemas operativos apropiados: Microsoft Windows XP y Windows Server 2003.
- ❖ Permite el acceso rápido a las plantillas que se utilizan con frecuencia.
- ❖ Permite crear diagramas más inteligentes vinculándolos a datos para proporcionar una imagen más completa de un proyecto, proceso o sistema, permitiendo la integración con otras aplicaciones de Microsoft office como Microsoft Office Excel 2007 y Microsoft Office Access 2007.

1.6.2 Axure RP

Axure RP es una aplicación para crear prototipos y especificaciones muy precisas para páginas web. Se trata de una herramienta especializada en esta tarea, así que cuenta con todo lo que se puede necesitar para crear los prototipos de forma eficiente. Axure RP te permite componer la página web visualmente, añadiendo, quitando y modificando los elementos con suma facilidad.

Donde Axure RP demuestra su grado de especialización es en las anotaciones. En este punto, permite especificar el estado de cada elemento (Propuesto, Aceptado, Incorporado), el beneficio esperado (Crítico, Importante, Útil), el riesgo, la estabilidad, a quién va dirigido y a quién se le asignará la tarea. Es una herramienta flexible, fácil de usar y permite crear un documento con especificaciones. Tiene como desventaja que este documento solamente se puede exportar en DOCX. Para su funcionamiento son necesarios los siguientes sistemas operativos Win98/Me/2000/XP. (18)

1.6.3 Visual Paradigm

Visual Paradigm ofrece la funcionalidad de crear prototipos de interfaz de usuario. Contiene una variedad de componentes necesario para diseñar los prototipos. Brinda una fácil navegación por todos los diagramas, clasificados por su tipo, en la ventana Navegación de Diagramas. Provee una Vista Previa al posicionar el mouse sobre algún diagrama con la posibilidad de saber con exactitud a cual nos referimos, incluso antes de abrirlo.

Para el estudio de de esta herramienta se tuvieron en cuenta las características mencionadas en el epígrafe 1.5.

Selección de la herramienta para prototipado de interfaz de usuario

Se decidió utilizar Microsoft Office Visio 2007 para la creación de los prototipos de interfaz de usuario debido a que brinda una amplia variedad de diagramas que permiten comprender, procesar y compartir información sobre los sistemas, recursos y procesos organizativos, siendo estos de fácil actualización. Además esta herramienta ofrece una amplia gama de componentes para la representación clara, concisa y eficaz del tipo de información que se desea visualizar.

Conclusiones

A partir del estudio realizado en este capítulo quedan sentadas las bases para la construcción de un software que informatice los procesos que se llevan a cabo en el Centro Nacional de Documentación e Información Judicial. Para definir este sistema se realizará el modelamiento del negocio y las etapas de elicitación, análisis, especificación, validación y gestión de la Ingeniería de Requisitos. Se considera factible el uso de RUP como metodología de desarrollo, de VP-UML como herramienta CASE y UML como lenguaje de modelado durante el proceso de desarrollo de software del sistema que se desea construir y Microsoft Office Visio 2007 para la creación de los prototipos de interfaz de usuario.

CAPÍTULO 2: MODELAMIENTO DEL NEGOCIO

En este capítulo se tratarán temas relacionados con el flujo de trabajo modelamiento del negocio que plantea la metodología de desarrollo de software RUP como son: en qué consiste y cuáles son sus objetivos. Se expondrán de manera clara y concisa los actores, trabajadores y entidades identificadas en el negocio así como los diagramas, descripciones de casos de uso y reglas del negocio.

2.1 Flujo de trabajo Modelamiento del Negocio.

El personal de una entidad a informatizar domina perfectamente su campo de acción, pero en pocas ocasiones conocen la terminología informática, son los ingenieros de esta especialidad los encargados de entrar en su entorno e identificar el problema para luego ofrecer una solución. El primer paso de todo proceso de desarrollo de software es precisamente alcanzar cierto nivel de conocimiento sobre el problema en cuestión.

En el flujo de modelamiento del negocio el equipo de desarrollo se familiariza con el funcionamiento de la empresa para conocer los procesos que se realizan en la misma, quiénes intervienen en cada uno de ellos y de qué forma lo hacen. Entre sus actividades principales se pueden mencionar la identificación de los procesos, actores y casos de uso del negocio, la especificación de los casos de uso, la identificación de entidades y trabajadores del negocio, así como las responsabilidades de los mismos. Para representar lo antes expuesto se realizan los diagramas: de casos de uso, de actividades y modelo de objetos.

En esta etapa inicial de desarrollo de software se obtiene el modelo de casos de uso del negocio que organiza y presenta los detalles importantes de problemas reales que se vinculan con el sistema informático a construir, tiene como finalidad describir cada proceso del negocio, especificando sus datos, actividades, roles y reglas. El modelado de negocio se utiliza como una técnica para la especificación de los requisitos más importantes del sistema al defender la idea de que los requisitos se definan a partir del propio negocio.

2.2 Objetivos del Flujo Modelamiento del Negocio

La metodología de desarrollo RUP establece un conjunto de objetivos para el flujo de modelamiento del negocio que deben ser cumplidos con el progreso del mismo. A continuación se mencionan dichos objetivos:

- ❖ Comprender la estructura y la dinámica de la organización en la cual se va a implantar un sistema.

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

2.3 Modelamiento del Negocio del CENDIJ

El Centro Nacional de Documentación e Información Judicial tiene como misión dotar a jueces, abogados, fiscales y profesionales en general, de información en correspondencia con sus intereses, para contribuir de manera ética y eficaz a su preparación para la toma de decisiones en la misión de administrar justicia.

En este centro se llevan a cabo diversos procesos, los cuales se caracterizan por una colección de datos que son producidos y manipulados mediante un conjunto de tareas, en las que intervienen trabajadores de la institución. Los procesos que se mencionan a continuación fueron los identificados en el centro:

Uso de la sala de lectura:

Consiste en el uso de la sala disponible para la consulta de documentos por un usuario o lector.

Préstamos de libros:

Consiste en la entrega de documentos disponibles en el fondo de la biblioteca, para su uso interno o externo, previa solicitud del usuario.

Uso de la máquina en sala:

Consiste en el uso de las estaciones de trabajo por parte de los usuarios para la consulta de información.

Procesamiento de información:

Consiste en el análisis, descripción física y almacenamiento de un nuevo libro que ingresa a la institución.

Compra de libros:

Consiste en la compra de nuevos libros para el centro.

Donación de libros:

Donación de libros que recibe la entidad para la prestación de servicios.

Búsqueda de información digital:

Consiste en la obtención de la documentación deseada en formato digital, mediante la utilización de los criterios de búsqueda establecidos para cada tipo de documento.

Búsqueda de información referencial:

Consiste en la obtención de la ubicación de un documento en formato duro para la realización de un préstamo.

2.3.1 Actores del Negocio

Un actor del negocio representa un rol jugado por alguien o algo externo al negocio que interactúa o se relaciona con él para beneficiarse de sus resultados. Es precisamente este rol lo que se modela y simboliza un grupo, individuo, entidad, organización o sistemas de información externos.

En el negocio a modelar se identificaron los siguientes actores:

Actores del Negocio	Descripción
Donante	Persona que realiza una donación al CENDIJ.
Vendedor	Lugares donde el centro puede realizar una compra. Estos pueden ser: editoriales, instituciones, eventos, entre otros.
Funcionario de Desarrollo de Colecciones	Entrega los documentos adquiridos al departamento de procesamiento.
Usuario	Personas externas al CENDIJ que se benefician con los servicios que brinda este.

Tabla 1: Descripción de los Actores del Negocio

2.3.2 Diagrama de Casos de Uso del Negocio

Un proceso de negocio es un grupo de tareas relacionadas lógicamente que se llevan a cabo en una determinada secuencia y manera, que emplean los recursos de la organización. Un caso de uso del negocio representa un proceso específico del negocio que produce un resultado de valor medible. Provee conocimiento acerca de lo que concibe la organización o entidad para satisfacer las expectativas de un actor del negocio.

El diagrama de casos de uso del negocio permite mostrar cada proceso del negocio como un caso de uso. Contiene información de alto nivel y rápida sobre el negocio sin entrar en detalles, visualiza los límites y el entorno de la organización bajo estudio. Este diagrama expresa las relaciones que existen entre los actores y casos de uso del negocio.

La siguiente figura representa el diagrama de casos de uso del negocio definido en la presente investigación.

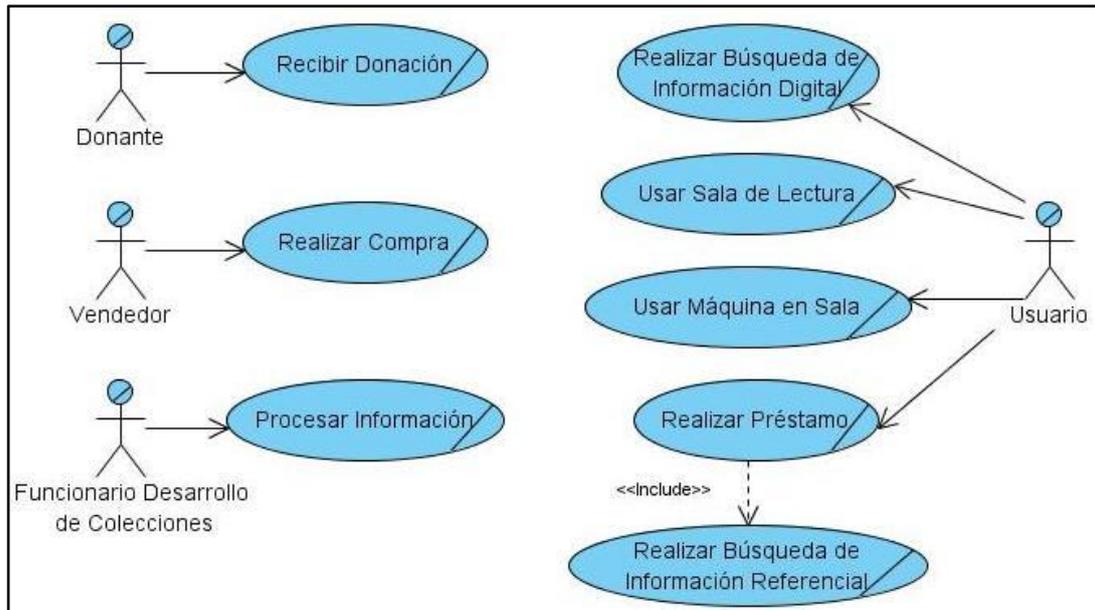


Figura 3: Diagrama de Casos de Uso del Negocio

2.3.3 Trabajadores del Negocio

Un trabajador del negocio es una abstracción de una persona o grupo de personas, una máquina o un sistema automatizado que actúa en el negocio siendo responsable de un conjunto completo de actividades, interactúa con otros trabajadores y manipula entidades del mismo.

Los trabajadores identificados en el modelado del negocio se utilizan como punto de partida para derivar un primer conjunto de actores y casos de uso para el sistema de información en construcción. (9)

Los trabajadores del negocio identificados se describen a continuación:

Trabajadores del Negocio	Descripción
Funcionario de Desarrollo de Colecciones	Persona responsable de realizar compras de nuevos documentos para la institución, recibir documentos donados y realizar las Fichas de Entrada de Información.
Director del CENDIJ	Persona que cuenta con la máxima autoridad para realizar funciones administrativas, siendo entre estas relevante para el negocio la aprobación de las solicitudes de compra.
Departamento de Economía	Departamento que interviene y aprueba todo lo referido al presupuesto del CENDIJ.

Indizador	Especialista que se encarga del proceso de indización de los documentos ingresados al centro.
Catalogador	Especialista que se encarga de realizar la descripción física y almacenamiento de los documentos ingresados al centro.
Especialista en Servicios	Especialista encargado de prestar servicios a los usuarios y de llevar el control de los mismo mediante el almacenamiento de boletas.

Tabla 2: Descripción de los Trabajadores del Negocio

2.3.4 Realización de los Casos de Uso del Negocio

La realización de un caso de uso del negocio muestra cómo colaboran los trabajadores y entidades del negocio para ejecutar un proceso. Cada realización se puede documentar a partir de diagramas de actividad y descripciones textuales.

Un diagrama de actividades se diseña para mostrar una visión simplificada de lo que ocurre durante una operación o proceso. Es una extensión de un diagrama de estados. El diagrama de estados muestra los estados de un objeto y representa las actividades como flechas que conectan a los estados. El diagrama de actividades resalta, precisamente, a las actividades. (18)

A continuación se mostrarán las descripciones textuales de los casos de uso y los diagramas de actividades correspondientes a cada uno de ellos, los cuales representan cómo se realizan los procesos del negocio identificados en el CENDIJ.

Caso de Uso Realizar Compra

Caso de Uso del Negocio	Realizar Compra
Actores	Vendedor
Resumen	El procedimiento comienza cuando el vendedor entrega la prefactura al funcionario, el Departamento de Economía la aprueba, se realiza la compra y el funcionario registra la entrada de los nuevos documentos.
Casos de Uso asociados	
Flujo Básico	
Acción del actor	Respuesta del proceso de negocio
1 El vendedor entrega al funcionario de desarrollo de colecciones una prefactura de lo que desea comprar.	
	2 El funcionario le entrega una solicitud de compra con la prefactura al director del Centro. 3 El director aprueba la solicitud. Si no la aprueba termina el caso de uso.

	<p>4 El director presenta la solicitud aprobada al Departamento de Economía.</p> <p>5 El Departamento de Economía aprueba la compra. De no ser aprobada ver Flujo Alterno 1.</p> <p>6 El Departamento de Economía entrega al funcionario un cheque.</p> <p>7 El funcionario se dirige a realizar la compra.</p>
8 El vendedor le entrega al funcionario los documentos comprados y una factura.	
	9 El funcionario llena la ficha de entrada de información con los siguientes datos: número de entrada, autor, origen, título, fecha de entrada, cantidad real, cantidad fondo, cantidad reserva, costo, vía de adquisición, formato.
Flujo Alterno	
Flujo Alterno 1	
Acción del actor	Respuesta del proceso de negocio
	1 El funcionario archiva la solicitud de compra.

Tabla 3: Descripción del Caso de Uso Realizar Compra

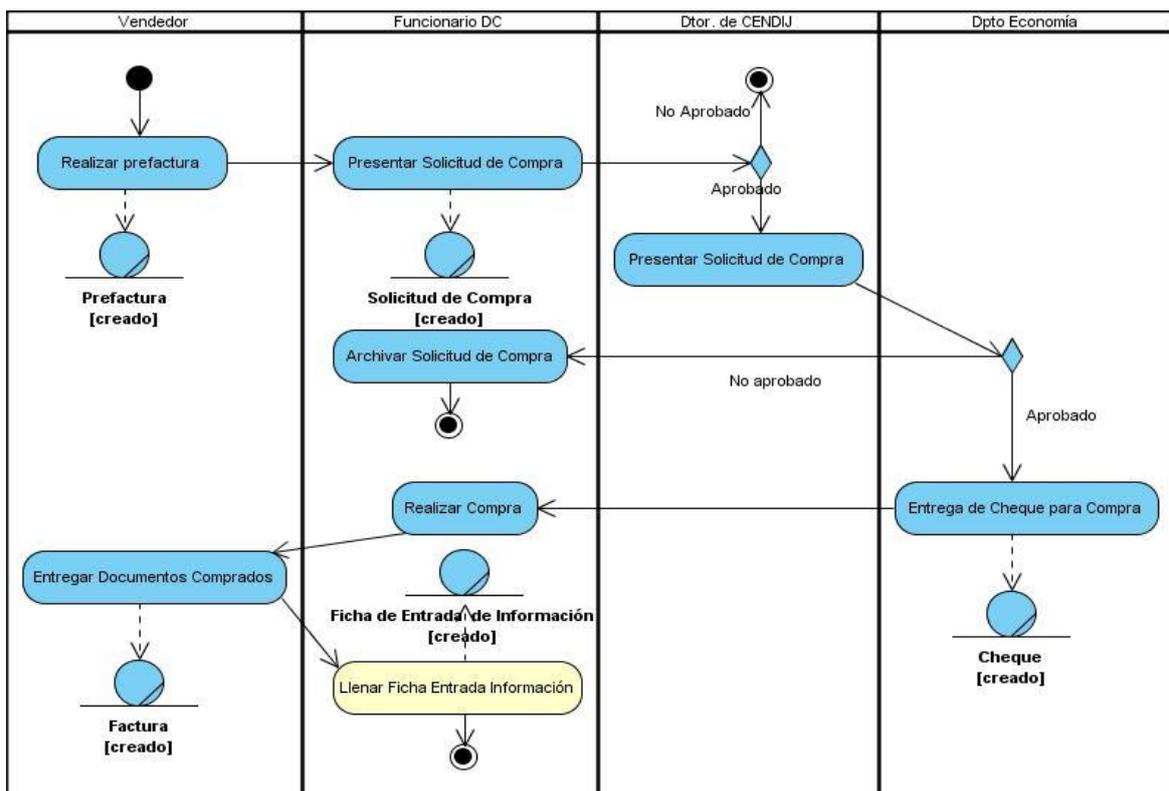


Figura 4: Diagrama de Actividades del Caso de Uso Realizar Compra

Caso de Uso Recibir Donación

Caso de Uso del Negocio	Recibir Donación
Actores	Donante

Resumen	El procedimiento comienza cuando el donante presenta los documentos que desea donar, el funcionario de desarrollo de colecciones le asigna un precio y registra la nueva entrada de información.
Casos de Uso asociados	
Flujo Básico	
Acción del actor	Respuesta del proceso de negocio
1 El donante presenta los documentos que quiere donar al Centro.	2 El funcionario le asigna precios a los documentos. 3 El funcionario llena la ficha de entrada de información con los siguientes datos: # de entrada, autor, origen, título, fecha de entrada, cantidad real, cantidad fondo, cantidad reserva, costo, vía de adquisición, formato. 4 El funcionario envía el listado de precios al director del Centro. 5 El director firma el listado de precios. 6 El director envía el listado firmado al Departamento de Economía. 7 El Departamento de Economía archiva el listado.

Tabla 4: Descripción del Caso de Uso Recibir Donación

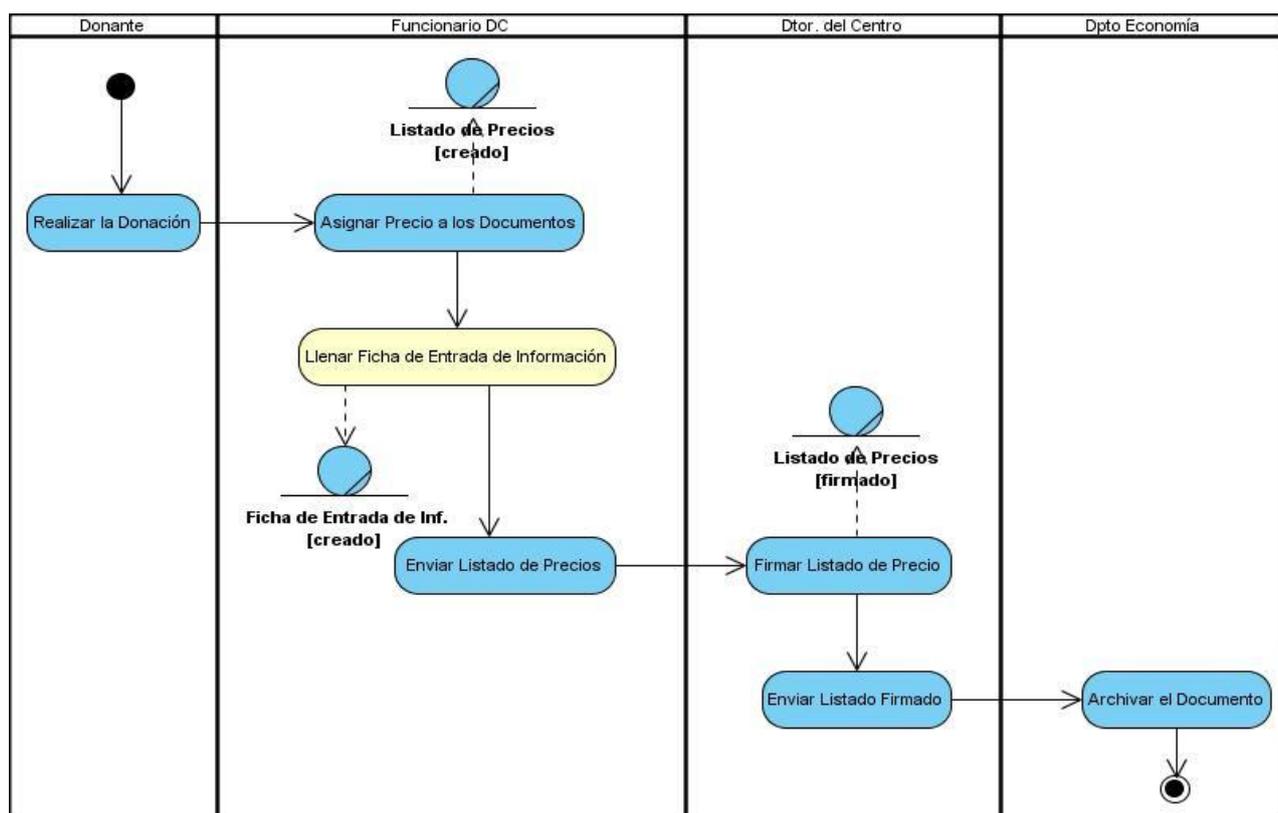


Figura 5: Diagrama de Actividades del Caso de Uso Recibir Donación

Caso de Uso Procesar Información

Caso de Uso del Negocio	Procesar Información
Actores	Funcionario de Desarrollo de Colecciones

Resumen	El procedimiento comienza cuando el funcionario entrega los documentos al indizador. Este analiza el documento para saber qué descriptores le corresponden, el catalogador crea la hoja de trabajo en dependencia del tipo de documento.
Casos de Uso asociados	
Flujo Básico	
Acción del actor	Respuesta del proceso de negocio
1 El funcionario entrega los documentos.	2 El indizador revisa el documento y le asigna los descriptores que le correspondan según su contenido. 3 El catalogador describe el documento físicamente y crea su hoja de trabajo. 4 El catalogador almacena el documento en el fondo.

Tabla 5: Descripción del Caso de Uso Procesar Información

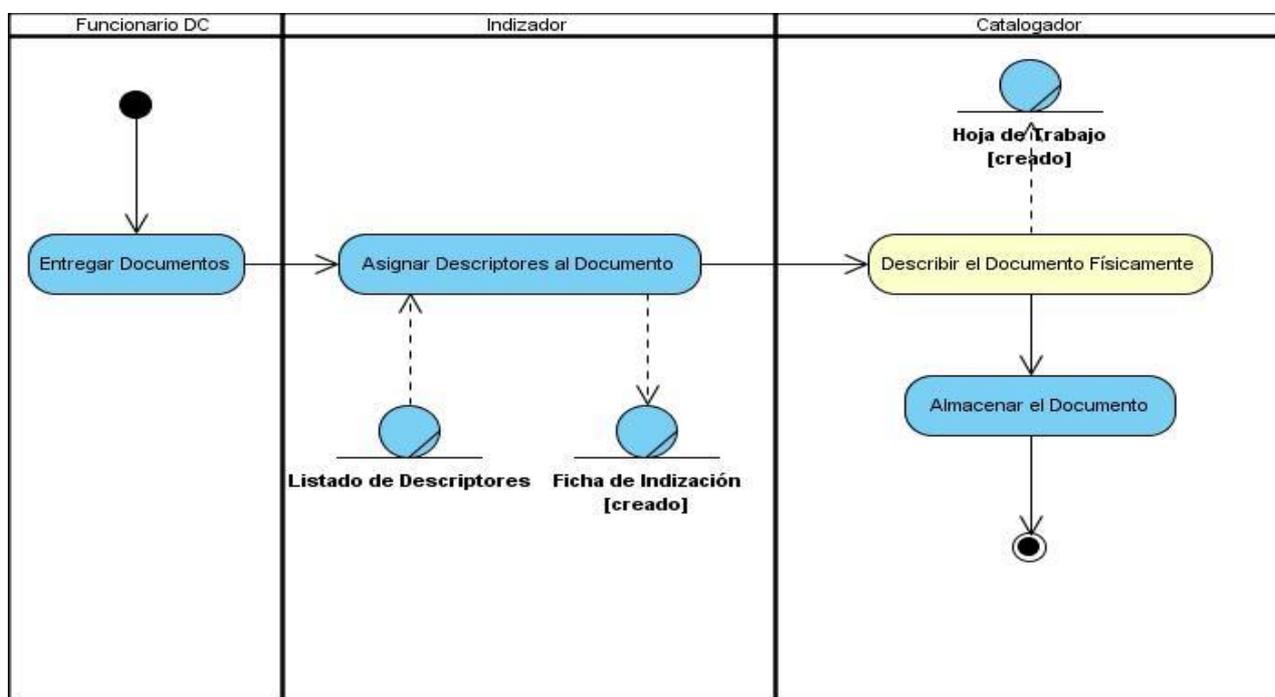


Figura 6: Diagrama de Actividades del Caso de Uso Procesar Información

Caso de Uso Realizar Búsqueda de Información Digital

Caso de Uso del Negocio	Realizar Búsqueda de Información Digital
Actores	Usuario
Resumen	El procedimiento comienza cuando el usuario solicita información digital, la especialista en servicios realiza una búsqueda y la registra.
Casos de Uso asociados	
Flujo Básico	
Acción del actor	Respuesta del proceso de negocio

<p>1 El usuario solicita la información que necesita.</p>	<p>2 El especialista en servicios entrevista al usuario. 3 El especialista en servicios determina la estrategia de búsqueda (criterios de búsqueda). 4 El especialista busca la información solicitada. 5 El especialista en servicios llena la boleta de entrega de información con los siguientes datos: tema, nombre y apellidos, procedencia, fecha de solicitud, e-mail o teléfono, categoría de usuario, nombre del técnico, herramienta de búsqueda, fecha de entrega de información, cantidad de documentos copiados, búsqueda realizada por, fecha. 6 Si encuentra la información se la entrega al usuario. Si no ver Flujo Alterno 1. 7 El especialista guarda la boleta.</p>
Flujos Alternos	
Flujo Alterno 1	
Acción del actor	Respuesta del proceso de negocio
	<p>1 El especialista realiza una búsqueda en internet de la información solicitada. 2 El especialista guarda la búsqueda realizada. 3 El especialista actualiza los siguientes datos en boleta de entrega de información: herramienta de búsqueda, búsqueda realizada por, fecha, cantidad de documentos copiados y fecha de entrega de información. 4 Volver al paso 6 del Flujo Básico.</p>

Tabla 6: Descripción del Caso de Uso Realizar Búsqueda de Información Digital

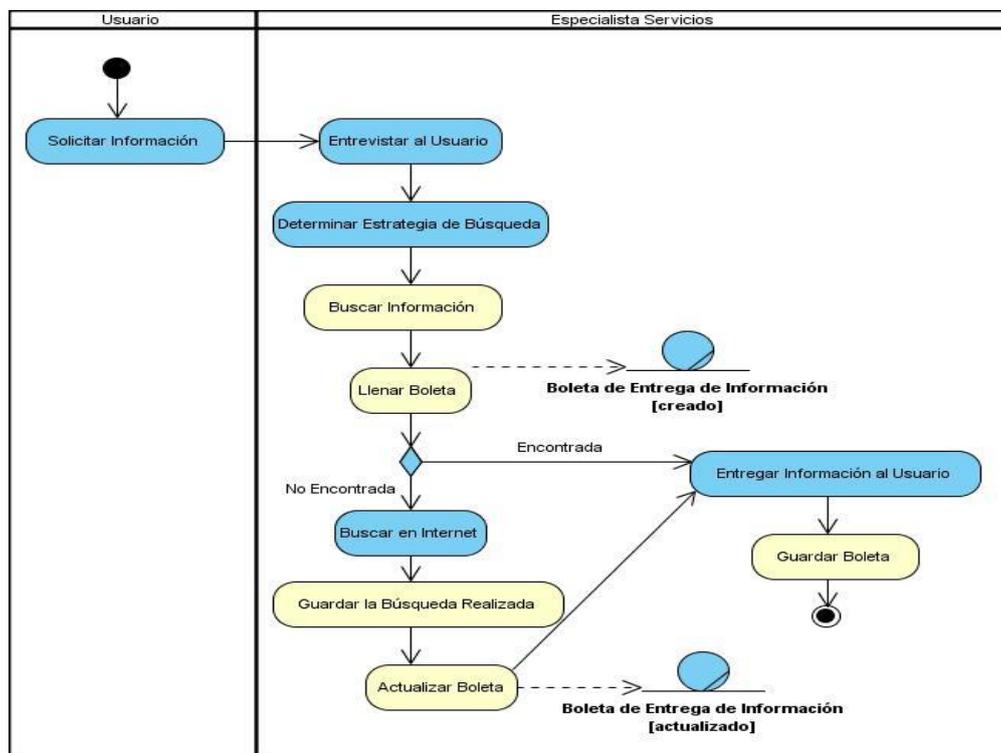


Figura 7: Diagrama de Actividades del Caso de Uso Realizar Búsqueda de Información Digital

Caso de Uso Realizar Búsqueda de Información Referencial

Caso de Uso del Negocio	Realizar Búsqueda de Información Referencial
Actores	Usuario
Resumen	El procedimiento comienza cuando el especialista en servicios entrevista al usuario, busca la información y el usuario escoge qué documento desea consultar.
Casos de Uso asociados	
Flujo Básico	
Acción del actor	Respuesta del proceso de negocio
	<ol style="list-style-type: none"> 1 El especialista en servicios entrevista al usuario. 2 El especialista en servicios determina la estrategia de búsqueda (criterios de búsqueda). 3 El especialista en servicios realiza la búsqueda de la información según el criterio determinado. 4 Si la encuentra le muestra al usuario los resultados de la búsqueda. Si no la encuentra ver Flujo Alterno 1.
5 El usuario escoge el documento a consultar.	
Flujos Alternos	
Flujo Alterno 1	
Acción del actor	Respuesta del proceso de negocio
	1 El especialista en servicios orienta al usuario dónde puede encontrar la información.

Tabla 7: Descripción del Caso de Uso Realizar Búsqueda de Información Referencial

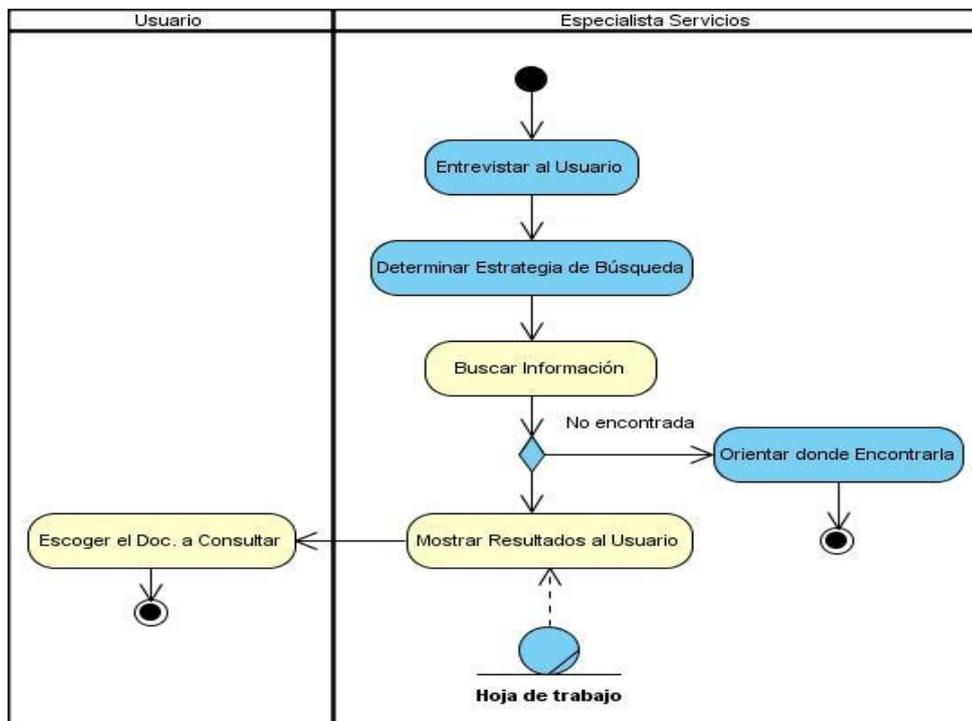


Figura 8: Diagrama de Actividades del Caso de Uso Realizar Búsqueda de Información Referencial

Caso de Uso Realizar Préstamo

Caso de Uso del Negocio	Realizar Préstamo
Actores	Usuario
Resumen	El procedimiento comienza cuando el usuario solicita un préstamo, el especialista en servicios registra los datos del usuario, del documento y presta el mismo.
Casos de Uso asociados	CU Realizar Búsqueda de Información Referencial (Include)
Flujo Básico	
Acción del actor	Respuesta del proceso de negocio
1 El usuario solicita un préstamo.	2 El especialista determina la ubicación del documento mediante el CU Realizar Búsqueda de Información Referencial. 3 El especialista en servicios entrega al usuario la boleta de control de préstamo. Si el préstamo es interno ver Sección 1, si es externo ver Sección 2. 4 El especialista comprueba que los datos del documento sea los mismos que están en la boleta. 5 El especialista guarda la boleta. 6 El especialista coloca el documento en el fondo.
Sección 1 "Préstamo Interno"	
Acción del actor	Respuesta del proceso de negocio
	1 El especialista en servicios entrega al usuario el documento solicitado.
2 El usuario llena los datos de la boleta que son: código del documento, tipo de préstamo, título, autor, tomo o volumen, ejemplar, nombre y apellidos, procedencia, correo, fecha de préstamo, fecha de devolución, categoría de usuario, devuelto, nombre del técnico, BD consultada, prórroga. 3 Entrega la boleta al especialista en servicios. 4 Consulta el documento. 5 Devuelve el documento prestado.	
	6 Ir al paso 4 del Flujo Básico.
Sección 2 " Préstamo Externo"	
Acción del actor	Respuesta del proceso de negocio
1 El usuario llena los datos de la boleta que son: código del documento, tipo de préstamo, título, autor, tomo o volumen, ejemplar, nombre y apellidos, procedencia, fecha de préstamo, fecha de devolución, categoría de usuario, devuelto, nombre del técnico, BD consultada, prórroga. 2 Entrega la boleta al especialista en servicios.	
	3 El especialista guarda la boleta de control de préstamo. 4 Entrega al usuario el documento solicitado.
5 El usuario se lleva el documento para	

ser consultado fuera del Centro.	
6 El usuario entrega el documento en la fecha establecida. Si no lo hace ver Flujo Alterno 1. Si pide prórroga ver Flujo Alterno 2.	
	7 Ir al paso 4 del Flujo Básico.
Flujos Alternos	
Flujo Alterno 1	
Acción del actor	Respuesta del proceso de negocio
	1 El especialista en servicios reclama la entrega del documento al usuario.
2 El usuario devuelve el documento prestado.	
	3 Ir al paso 4 del Flujo Básico.
Flujo Alterno 2	
Acción del actor	Respuesta del proceso de negocio
	1 El especialista en servicios establece la prórroga en la boleta de control de préstamo. 2 Entrega el documento al usuario nuevamente. 3 Volver al paso 5 de la Sección 2.

Tabla 8: Descripción del Caso de Uso Realizar Préstamo

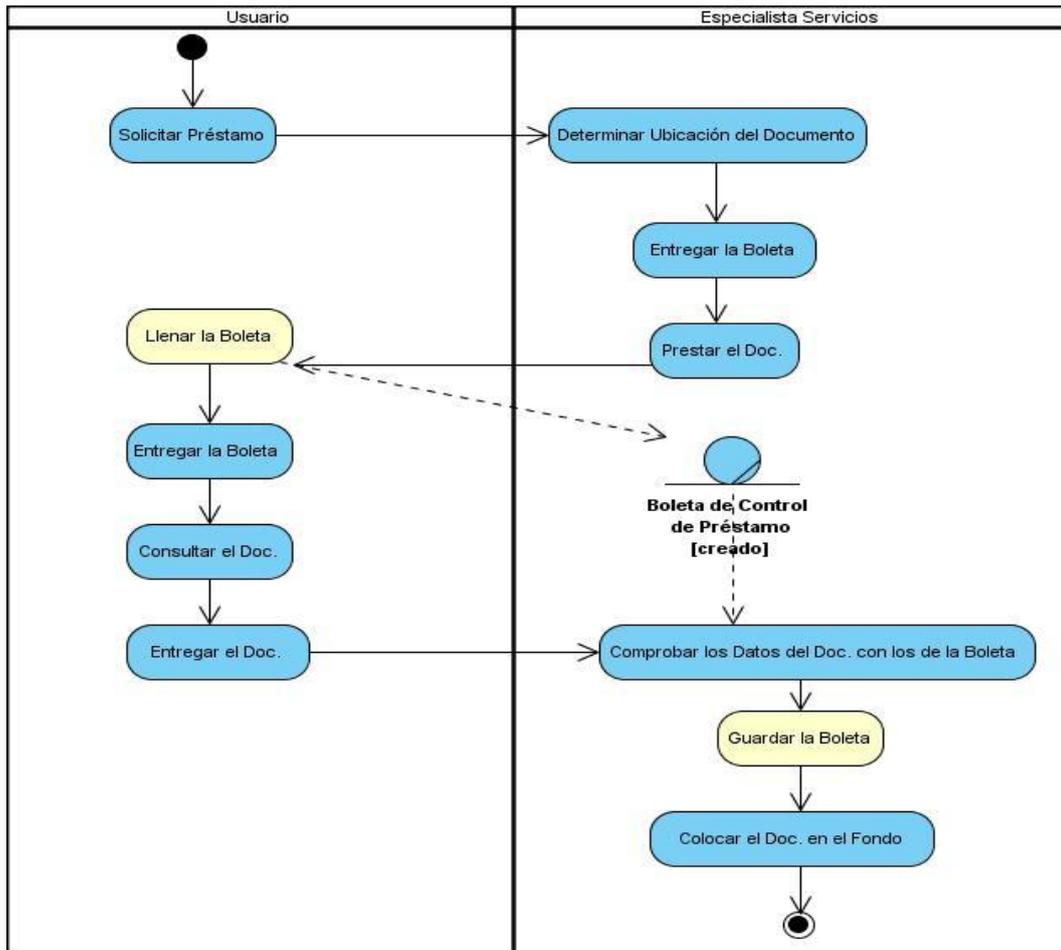


Figura 9: Diagrama de Actividades del Caso de Uso Realizar Préstamo (Sección Préstamo Interno)

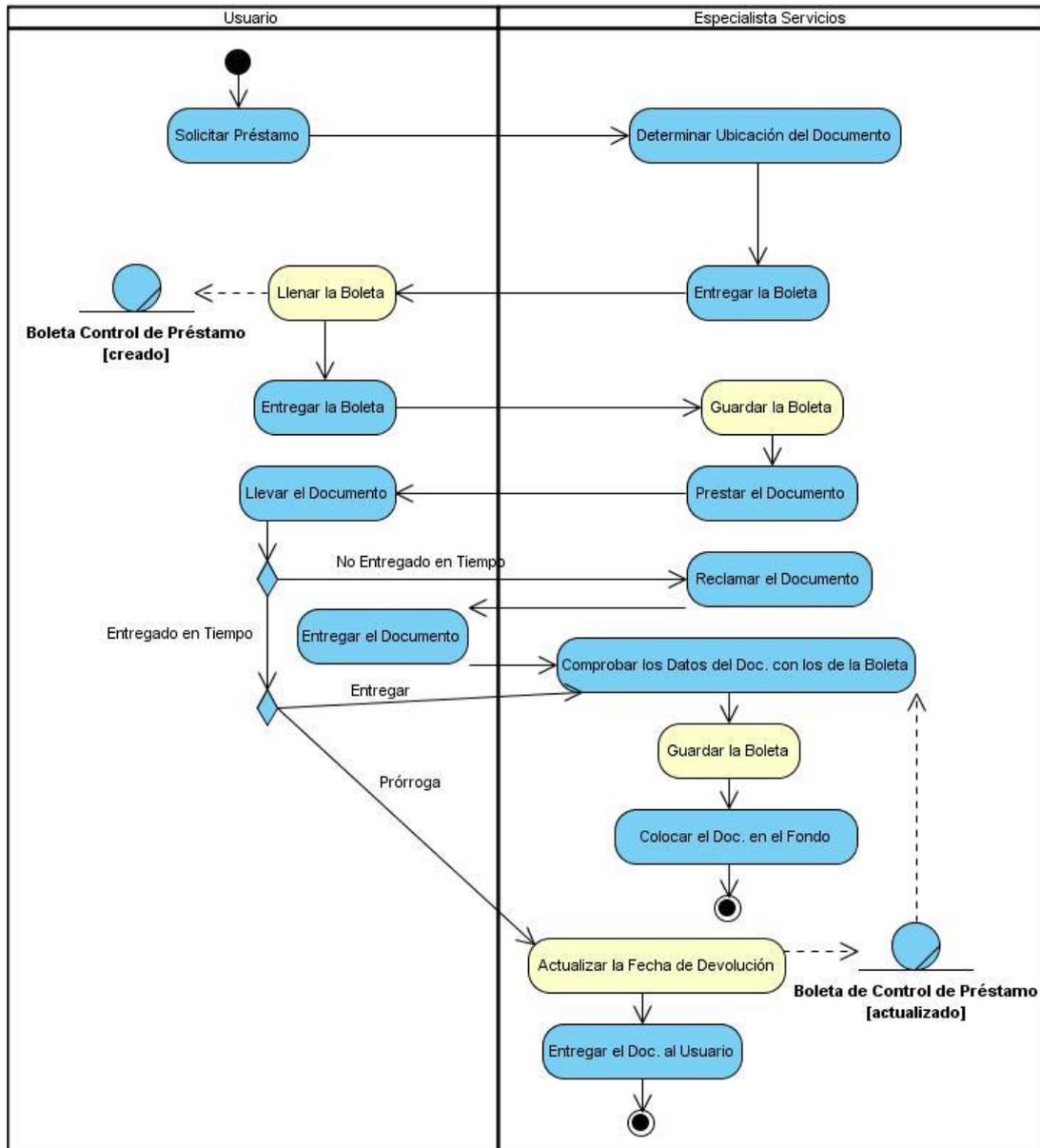


Figura 10: Diagrama de Actividades del Caso de Uso Realizar Préstamo (Sección Préstamo Externo)

Caso de Uso Usar Sala de Lectura

Caso de Uso del Negocio	Usar Sala de Lectura
Actores	Usuario
Resumen	El procedimiento comienza cuando un usuario solicita el servicio, el especialista en servicios registra sus datos y le permite entrar a la sala.
Casos de Uso asociados	
Flujo Básico	
Acción del actor	Respuesta del proceso de negocio
1 El usuario solicita usar la sala de lectura.	2 El especialista en servicios le entrega la boleta de control de sala de lectura.

3 El usuario llena la boleta con los siguientes datos: nombre y apellidos, procedencia, fecha, categoría de usuario y nombre del técnico.	
	4 El especialista guarda la boleta.
5 El usuario hace uso de la sala de lectura.	

Tabla 9: Descripción del Caso de Uso Usar Sala de Lectura

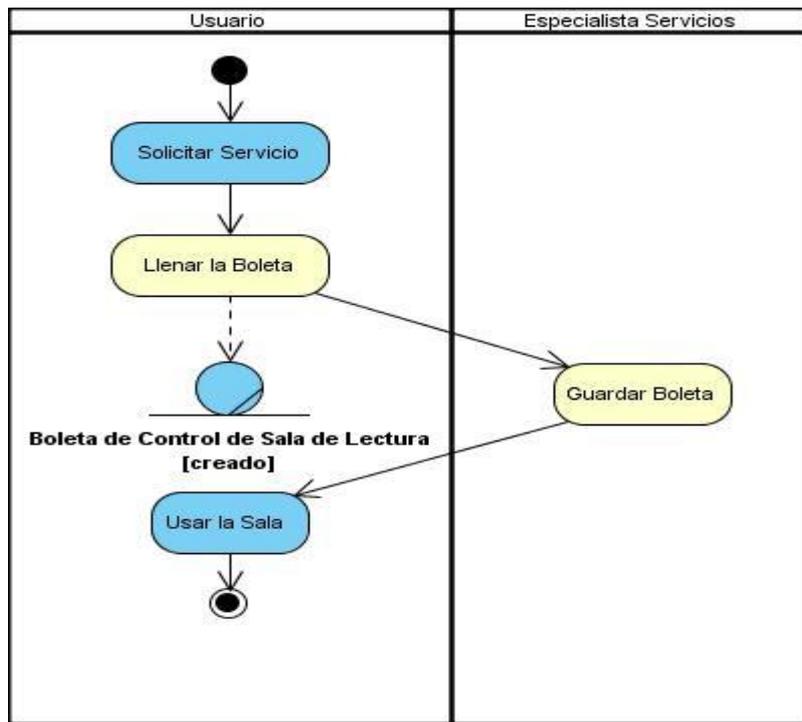


Figura 11: Diagrama de Actividades del Caso de Uso Usar Sala de Lectura

Caso de Uso Usar Máquina en Sala

Caso de Uso del Negocio	Usar la Máquina en Sala
Actores	Usuario
Resumen	El procedimiento comienza cuando un usuario solicita el servicio, el especialista en servicios le permite usar la máquina.
Casos de Uso asociados	
Flujo Básico	
Acción del actor	Respuesta del proceso de negocio
1 El usuario solicita usar la máquina en la sala.	2 El especialista en servicios le entrega la boleta de control de uso de máquina en la sala.
3 El usuario hace uso de la máquina. 4 El usuario llena la boleta con los siguientes datos: nombre y apellidos, procedencia, tipo de servicio, hora de inicio, hora de fin y categoría de usuario, nombre del técnico, cantidad de documentos copiados.	

Tabla 10: Descripción del Caso de Uso Usar Máquina en Sala

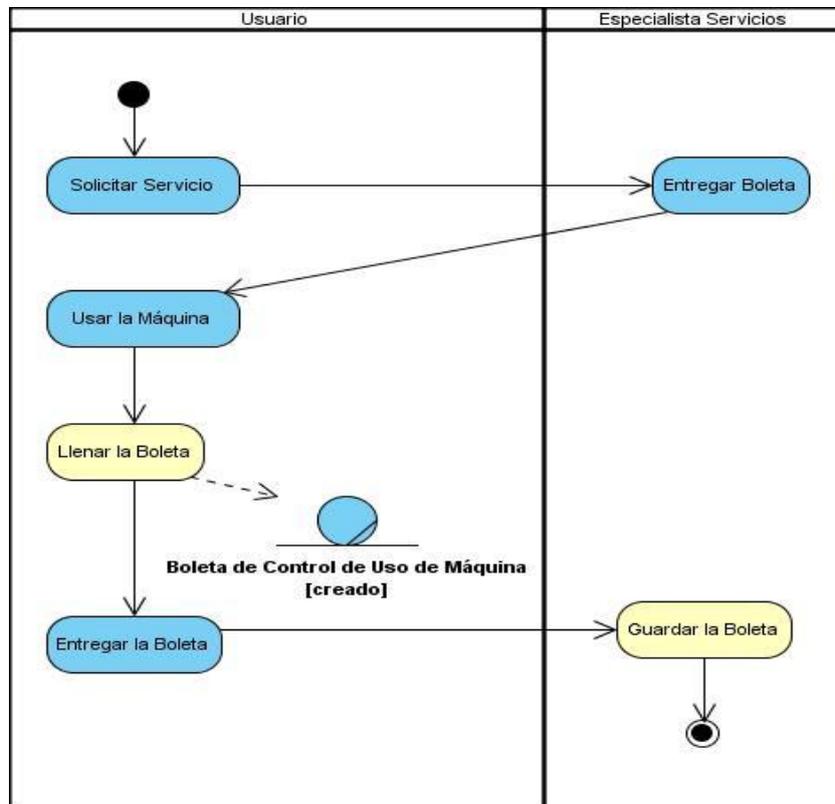


Figura 12: Diagrama de Actividades del Caso de Uso Usar Máquina en Sala

2.3.5 Modelo de Objetos del Negocio

Un modelo de objetos del negocio es un modelo interno a un negocio. Describe cómo cada caso de uso de negocio es llevado a cabo por parte de un conjunto de trabajadores que utilizan las entidades del negocio y las unidades de trabajo. (9)

El modelo de objetos describe cómo colaboran los trabajadores y las entidades del negocio dentro del flujo de trabajo del proceso de negocio. Las entidades representan a los objetos que los trabajadores toman, inspeccionan, manipulan, producen o utilizan durante la realización de los casos de uso de negocio.

A continuación se muestran los Diagramas de Objetos correspondientes a los casos de uso del negocio identificados:

Realizar Compra



Figura 13: Diagrama de Objetos del caso de uso Realizar Compra

Recibir Donación

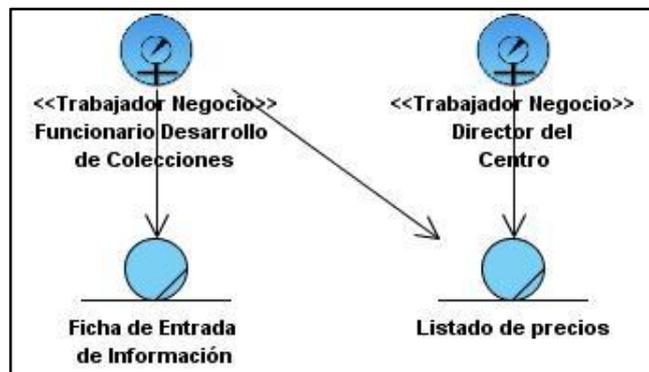


Figura 14: Diagrama de Objetos del caso de uso Recibir Donación

Procesar Información



Figura 15: Diagrama de Objetos del caso de uso Procesar Información

Buscar Información

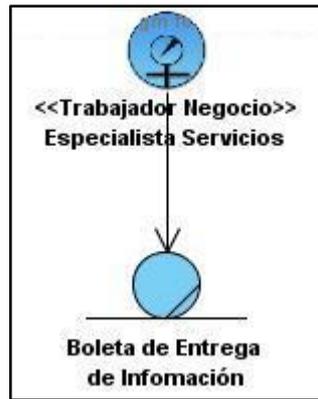


Figura 16: Diagrama de Objetos del caso de uso Realizar Búsqueda de Información Digital

Realizar Búsqueda de Información Referencial

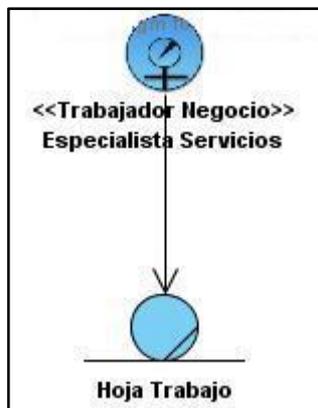


Figura 17: Diagrama de Objetos del caso de uso Realizar Búsqueda de Información Referencial

Realizar Préstamo



Figura 18: Diagrama de Objetos del caso de uso Realizar Préstamo

Usar la Sala de Lectura



Figura 19: Diagrama de Objetos del caso de uso Usar Sala de Lectura

Usar Máquina en Sala



Figura 20: Diagrama de Objetos del caso de uso Usar Máquina en Sala

2.3.6 Reglas del Negocio

Las reglas de negocio describen políticas que deben cumplirse o condiciones que deben satisfacerse, por lo que regulan algún aspecto del negocio. El proceso de especificación implica identificarlas dentro del negocio, evaluar si son relevantes dentro del campo de acción que se modela e implementarlas en la propuesta de solución.

En el CENDIJ se identificaron las siguientes reglas del negocio:

Relación
RN01 El usuario solicita un documento. A la asociación entre el usuario y el documento se le denomina Préstamo.
Restricción
RN02 A todos los documentos donados al centro se les deben asignar un precio.
RN03 En la búsqueda de información, se debe gestionar una boleta por cada tema que se solicite.
RN04 El servicio de préstamo externo de documento solo se brinda a jueces del Tribunal Supremo Popular.

RN05 El tiempo de duración de un préstamo externo de documento es de quince días.
RN06 Las prórrogas de un mismo préstamo externo se realizan solo tres veces.
RN07 El uso de las máquinas en la sala por parte de los usuarios es exclusivamente para la realización de búsquedas de información.
Estímulo y respuesta
RN08 Si se ha llegado a la fecha en que un usuario debe entregar un documento y no lo ha hecho, se le hace una llamada telefónica.

Tabla 10: Reglas del Negocio

Validación del Negocio

La validación del flujo de trabajo modelamiento del negocio se realizó mediante la firma de un acta de aceptación del negocio, donde ambas partes involucradas, tanto los clientes como el equipo de desarrollo acordaron que la descripción realizada de los procesos de negocio llevados a cabo en la entidad se corresponde con el flujo de actividades que ocurren en el centro. Para consultar el acta ver Anexo 1.

Conclusiones

Con la realización del Modelamiento del Negocio se pudo identificar y comprender con claridad los procesos que se desarrollan en el CENDIJ. Como resultado de este flujo de trabajo se obtuvieron artefactos de gran importancia como son: la realización de los casos de uso, los diagramas de actividades, el diagrama de casos de uso del negocio, el modelo de objetos y se definieron las reglas que regulan el negocio. A partir de las actividades identificadas que pueden ser informatizadas se inicia la definición de los requisitos que debe cumplir el sistema a construir, dando paso al flujo de trabajo de requerimientos.

CAPÍTULO 3: REQUERIMIENTOS DEL SISTEMA

En este capítulo se tratarán temas relacionados con el flujo de trabajo requerimientos que plantea la metodología de desarrollo de software RUP como son: en qué consiste, cuáles son sus objetivos, patrones de casos de uso así como métricas de validación de requisitos. Quedará explícito el listado de requisitos funcionales y no funcionales, el diagrama de casos de uso del sistema, las descripciones y la validación de los requisitos propuestos.

3.1 Flujo de Trabajo Requerimientos

El propósito fundamental del flujo de trabajo requerimientos es guiar el desarrollo hacia el sistema correcto. Esto se consigue mediante una descripción de los requisitos del sistema suficientemente buena para que pueda llegarse a un acuerdo entre el cliente y los desarrolladores sobre qué debe y qué no debe hacer el sistema. (9)

Las principales actividades que se realizan en este flujo de trabajo son:

- ❖ Identificar y clasificar requisitos.
- ❖ Encontrar actores y casos de uso.
- ❖ Priorizar casos de uso.
- ❖ Detallar casos de uso.
- ❖ Estructurar el modelo de casos de uso.

En este flujo de trabajo hay que analizar el problema, comprender las necesidades de los interesados y expresarlas en forma de requisitos; construir el diagrama de casos de uso para los requisitos funcionales y los no funcionales describirlos textualmente en especificaciones suplementarias. Además hay que gestionar los cambios en los requisitos a lo largo de todo el proceso. (19)

3.2 Objetivos del Flujo de Requerimientos

El objetivo principal del flujo de trabajo requerimientos es establecer las funcionalidades que debe prestar el sistema que se quiere desarrollar. Es en esta etapa del ciclo de vida de un software donde se capturan los requisitos de manera que sean consistentes con las necesidades del usuario mediante el uso de técnicas de captura de requisitos, se analizan para descubrir problemas que puedan presentar los mismos, se describen para lograr una mejor comprensión por parte del equipo de desarrollo, se aprueban con el cliente y se validan para confirmar que no sean ambiguos y que estén completos.

Por lo anteriormente expuesto los objetivos específicos del flujo de trabajo requerimientos son:

- ❖ Definir el ámbito del sistema.
- ❖ Definir una interfaz de usuarios para el sistema, enfocada a las necesidades y metas del usuario.
- ❖ Establecer y mantener un acuerdo entre clientes y otros involucrados sobre lo que el sistema debería hacer.
- ❖ Proveer a los desarrolladores un mejor entendimiento de los requisitos del sistema.
- ❖ Proveer una base para estimar recursos y tiempo de desarrollo del sistema.
- ❖ Proveer una base para la planeación de los contenidos técnicos de las iteraciones.

3.3 Requerimientos Funcionales y no Funcionales

Requerimientos Funcionales:

RF01 Autenticar Usuario.

El sistema debe permitir autenticar un usuario con los siguientes parámetros: usuario y contraseña.

RF02 Adicionar Usuario.

El sistema debe permitir adicionar un usuario mediante los siguientes datos: nombre y apellidos, carné de identidad, usuario, contraseña y rol.

RF03 Modificar Usuario.

El sistema debe permitir modificar todos los datos del usuario especificado.

RF04 Eliminar Usuario.

El sistema debe permitir eliminar uno o varios usuarios seleccionados.

RF05 Buscar Usuario.

El sistema debe permitir realizar una búsqueda de usuarios por el parámetro: usuario.

El resultado de la búsqueda debe mostrar los datos del usuario que coincidan con el criterio de búsqueda establecido. Los datos que se deben mostrar como resultado son: nombre y apellidos, carné de identidad, usuario, contraseña y rol.

RF06 Insertar Descriptor.

El sistema debe permitir adicionar un descriptor.

RF07 Modificar Descriptor.

El sistema debe permitir modificar un descriptor.

RF08 Eliminar Descriptor.

El sistema debe permitir eliminar un descriptor.

RF09 Visualizar Listado de Descriptores.

El sistema debe permitir visualizar el listado de descriptores.

RF10 Insertar Ficha de Entrada de Información.

El sistema debe permitir adicionar una Ficha de Entrada de Información con los siguientes datos: número de acceso, autor, origen, título, fecha de entrada, cantidad real, cantidad fondo, cantidad reserva, costo, vía de adquisición, formato.

RF11 Modificar Ficha de Entrada de Información.

El sistema debe permitir modificar los datos de una ficha de entrada de información.

RF12 Insertar Boleta de Entrega de Información.

El sistema debe permitir adicionar una Boleta de Entrega de Información con los siguientes datos: tema, nombre y apellidos, procedencia, fecha de solicitud, e-mail o teléfono, categoría de usuario, nombre del técnico, herramienta de búsqueda, cantidad de documentos copiados, búsqueda realizada por, fecha de búsqueda, fecha de entrega de información.

RF13 Modificar Boleta de Entrega de Información.

El sistema debe permitir modificar los datos de una boleta de entrega de información.

RF14 Insertar Boleta de Control de Préstamo.

El sistema debe permitir adicionar una boleta de Control de Préstamo con los siguientes datos: código del documento, tipo de préstamo, título, autor, tomo o volumen, ejemplar, nombre y apellidos, procedencia, correo, fecha de préstamo, fecha de devolución, categoría de usuario, nombre del técnico, devuelto, base de datos consultada, prórroga.

RF15 Modificar Boleta de Control de Préstamo.

El sistema debe permitir modificarlos datos de una boleta de control de préstamo.

RF16 Insertar Boleta de Control de Sala de Lectura.

El sistema debe permitir adicionar una boleta de Control de Sala de Lectura con los siguientes datos: nombre y apellidos, procedencia, fecha, categoría de usuario, nombre del técnico.

RF17 Modificar Boleta de Control de Sala de Lectura.

El sistema debe permitir modificar los datos de una boleta de control de sala de lectura.

RF18 Insertar Boleta de Control de Uso de Máquina.

El sistema debe permitir adicionar una boleta de Control de Uso de Máquina con los siguientes datos: nombre y apellidos, procedencia, tipo de servicios, hora de inicio, hora de término, categoría de usuario, nombre de técnico, cantidad de documentos copiados.

RF19 Modificar Boleta de Control de Uso de Máquina.

El sistema debe permitir modificar los datos de una boleta de control de uso de máquina.

RF20 Buscar Boleta.

El sistema debe permitir realizar una búsqueda de las distintas boletas y fichas.

- ❖ Si la búsqueda es de Boletas de Control de Sala de Lectura se tendrán los siguientes criterios de búsqueda: nombre y apellidos, fecha y categoría de usuario.
- ❖ En el caso que sea de Boletas de Control de Máquina en Sala: nombre y apellidos, tipo de servicio, fecha y categoría de usuario.
- ❖ En el caso que sea de Boletas de Control de Préstamo: código, título, autor, nombre y apellidos, fecha de préstamo y categoría de usuario.
- ❖ En el caso que sea de Boletas de Entrega de Información: tema, nombre y apellidos, fecha de solicitud y categoría de usuario.
- ❖ En el caso que sea de Fichas de Entrada de Información: número de acceso, autor, título, fecha de entrada y vía de adquisición.

RF21 Insertar Documento.

El sistema debe permitir insertar un documento mediante los siguientes campos:

- ❖ Cuando los documentos que se desean agregar sean Publicaciones Seriadas, estos deben contar con los siguientes datos: número de acceso, ubicación física, autor personal, autor institucional, título de publicación seriada, mención de responsabilidad, edición, volumen, número, mes, lugar de publicación, editorial, año de publicación, idioma, páginas, ilustraciones, material acompañante, serie, notas, descriptores, número normalizado, cantidad de ejemplares, fecha de entrada al sistema, especialista.
Ficha analítica: número de acceso, ubicación física, autor personal, autor institucional, título, mención de responsabilidad, mes, idioma, páginas, ilustraciones, notas, descriptores, cantidad de ejemplares, fecha de entrada al sistema, especialista.
- ❖ En el caso que los documentos que se desean agregar sean Libros y Folletos, Fondo General o Referencias, estos deben contar con los siguientes datos: número de acceso, ubicación física, autor personal, autor institucional, título, mención de responsabilidad, edición, lugar de publicación, país, editorial, año de publicación, idioma, tomo o volumen, páginas, ilustraciones, material acompañante, serie, notas, descriptores, número normalizado, cantidad de ejemplares, especialista, fecha de entrada al sistema.
Ficha analítica: número de acceso, ubicación física, autor personal, autor institucional, título, mención de responsabilidad, tomo o volumen, páginas, ilustraciones, notas, descriptores, cantidad de ejemplares, fecha de entrada al sistema.
- ❖ En el caso que los documentos que se desean agregar sean de Flujo Ascendente, estos deben contar con los siguientes datos: número de acceso, ubicación física, autor,

procedencia del autor, categoría del autor, país del autor, título del trabajo, mención de responsabilidad, año de presentación, páginas, ilustraciones, presentado en, tipo de trabajo, notas, descriptores, cantidad de ejemplares, fecha de entrada al sistema, especialista.

- ❖ En el caso que se deseen agregar CD, estos deben contar con los siguientes datos: número de acceso, título, tipo de documento, mención de responsabilidad, designación del recurso, idioma, lugar de publicación, editorial, año de publicación, extensión del recurso, designación específica, material acompañante, ilustraciones, serie, notas, número normalizado, descriptores, cantidad de ejemplares, fecha de entrada al sistema, especialista.
- ❖ El sistema debe permitir adicionar documentos con los siguientes datos en caso que fuera Legislación serían los siguientes campos: tipo de norma, materia, número de la norma, nombre de la norma, órgano emisor, fecha de emisión, gaceta oficial, año de publicación, número de la gaceta, mes de la gaceta, notas, descriptores, fecha de entrada al sistema, especialista.
- ❖ En caso que el documento sea Investigación, se introducen los siguientes datos: autor, procedencia del autor, categoría del autor, país del autor, título del trabajo, año de presentación, ilustraciones, presentado en, tipo de trabajo, notas, descriptores, fecha de entrada al sistema, especialista.
- ❖ En caso que el documento sea Doctrina, se introducen los siguientes datos: autor personal, autor institucional, título, tipo de soporte, edición, lugar de publicación, país, editor, año de publicación, fecha de actualización/revisión, fecha de cita, notas, disponible en, número normalizado, descriptores, fecha de entrada al sistema, especialista.
- ❖ En caso que el documento sea Disposición del Consejo de Gobierno, se introducen los siguientes datos: tipo de disposición, número, año, descriptores, fecha de entrada al sistema, especialista.
- ❖ En caso que el documento sea Sentencia, se introducen los siguientes datos: sentencia, año de sentencia, número de rollo, año de rollo, ponente, proceso, año de proceso, tribunal, sala, fallo, asuntos, descriptores.

RF22 Modificar Documento.

El sistema debe permitir modificar todos los datos de un documento seleccionado.

RF23 Eliminar Documento.

El sistema debe permitir eliminar uno o varios documentos seleccionados.

RF24 Buscar Información Referencial.

El sistema debe permitir realizar una búsqueda de la información referencial a partir de los siguientes criterios de búsqueda:

- ❖ En caso que la búsqueda fuera de Publicaciones Seriadas y la realice un usuario se hará mediante los siguientes datos: ubicación física, número de acceso, autor personal, autor institucional, título publicación seriada, título, mención de responsabilidad, editorial, año de publicación, serie, notas y descriptores.
- ❖ En caso que la búsqueda se realice por un especialista se deben adicionar además los campos siguientes: volumen, número, mes, ilustraciones, material acompañante, número normalizado, cantidad de ejemplares, fecha de entrada al sistema, especialista.
- ❖ En el caso que la búsqueda fuera de Libros y Folletos, Fondo General o Referencias y la realice un usuario se hará mediante los siguientes criterios de búsqueda: ubicación física, número de acceso, autor personal, autor institucional, título, mención de responsabilidad, país, editorial, año de publicación, serie, fuente, notas, descriptores.
- ❖ En caso que la búsqueda se realice por un especialista se deben adicionar además los campos siguientes: lugar de publicación, ilustraciones, material acompañante, número normalizado, cantidad de ejemplares, especialista, fecha de entrada al sistema.
- ❖ En el caso que la búsqueda fuera de Flujo Ascendente y la realice un usuario se hará mediante los siguientes criterios de búsqueda: número de acceso, ubicación física, autor, país del autor, título del trabajo, año de presentación, presentado en, tipo de trabajo, notas, descriptores, texto o frase.
- ❖ En caso que la búsqueda se realice por un especialista se deben adicionar además los campos siguientes: procedencia del autor, categoría del autor, ilustraciones, cantidad de ejemplares, fecha de entrada al sistema, especialista.
- ❖ En el caso que la búsqueda fuera de CD y la realice un usuario se hará mediante los siguientes criterios de búsqueda: número de acceso, título, tipo de documento, mención de responsabilidad, designación específica, material acompañante, serie, notas, descriptores, texto o frase.
- ❖ En caso que la búsqueda se realice por un especialista se deben adicionar además los campos siguientes: designación del recurso, idioma, ilustraciones, cantidad de ejemplares, fecha de entrada al sistema, especialista.

El resultado de la búsqueda debe mostrar los documentos que coinciden con el o los criterios de búsqueda establecidos.

RF25 Buscar Información Digital.

El sistema debe permitir realizar una búsqueda de la información digital a partir de los siguientes criterios de búsqueda:

- ❖ En caso de que la búsqueda fuera de Legislación y la realice un usuario se hará mediante los siguientes parámetros: tipo de norma, materia, número de la norma, nombre de la norma, órgano emisor, fecha de emisión, notas, descriptores, texto o frase.
- ❖ En caso que la búsqueda se realice por un especialista se deben adicionar además los datos siguientes: gaceta oficial, año de publicación, número de la gaceta, fecha de entrada al sistema, especialista.
- ❖ En caso que la búsqueda fuera de Investigación y la realice un usuario se hará mediante los siguientes parámetros: autor, país del autor, título del trabajo, año de presentación, presentado en, tipo de trabajo, notas, descriptores, texto o frase.
- ❖ En caso que la búsqueda se realice por un especialista se deben adicionar además los datos siguientes: procedencia del autor, categoría del autor, ilustraciones, fecha de entrada al sistema, especialista.
- ❖ En caso que la búsqueda fuera de Doctrina y la realice un usuario se hará mediante los siguientes parámetros: autor personal, autor institucional, título, país, fecha de cita, notas, descriptores, texto o frase.
- ❖ En caso que la búsqueda se realice por un especialista se deben adicionar además los datos siguientes: tipo de soporte, año de publicación, disponible en, fecha de entrada al sistema, especialista.
- ❖ En caso que la búsqueda fuera de Disposición del consejo de Gobierno y la realice un usuario se hará mediante los siguientes parámetros: tipo de disposición, número, año, descriptores, texto o frase.
- ❖ En caso que la búsqueda se realice por un especialista se deben adicionar además los datos siguientes: fecha de entrada al sistema, especialista.
- ❖ En caso que la búsqueda fuera de Sentencias se realizará mediante los siguientes parámetros: número, año, ponente, tribunal, sala, texto, fallo, causales, descriptores.

El resultado de la búsqueda debe mostrar los documentos que coinciden con el o los criterios de búsqueda establecidos.

Requerimientos no Funcionales:

Restricciones en el diseño e implementación

RNF01 Se utilizará Visual Paradigm como herramienta CASE para el modelado y obtención de los distintos diagramas necesarios para el desarrollo del software.

RNF02 Se utilizará Microsoft Office Visio 2007 como herramienta para la creación de los prototipos de interfaz de usuarios.

RNF03 El sistema será implementado con el lenguaje de programación PHP 5.

RNF04 El Entorno de Desarrollo Integrado que se utilizará para la implementación es NetBeans 6.9 mediante el uso del Framework Simfony 1.4.

RNF05 Se utilizará Propel como Framework para el trabajo con la base de datos.

RNF06 Se utilizará Ext JS como Framework para el trabajo con las interfaces de usuarios.

Requisitos de software

RNF07 En el servidor central debe estar instalado el servidor web Apache.

RNF08 Las estaciones de trabajo deben tener instalado un navegador web para acceder a la aplicación y Microsoft Office 2003 o superior para visualizar los documentos en formato digital.

Requisitos de apariencia o interfaz externa

RNF09 El sistema deberá tener una interfaz gráfica uniforme que incluirá un menú en forma de árbol a la izquierda con las funcionalidades que ofrecerá el sistema, a la derecha del mismo estará ubicada la información, la parte superior contará con un banner que incluirá el logotipo de la aplicación.

Requisitos de seguridad

RNF10 La seguridad se establecerá por roles que se le asignarán a los usuarios que interactúen con el sistema para garantizar que la información almacenada solo sea modificada y/o visualizada por los usuarios autorizados.

RNF11 Los usuarios deben autenticarse en el sistema para verificar que tienen acceso al mismo y permitirle la interacción con las funcionalidades en dependencia de los roles que tenga el usuario.

RNF12 El sistema debe permitir que cada usuario pueda cambia su contraseña, así como debe exigir que la misma tenga un nivel aceptable de complejidad.

RNF13 Se deben almacenar de manera cifrada las claves de los usuarios en la base de datos, para evitar que se pueda reemplazar la identidad de un usuario en el sistema.

Requisitos de hardware

RNF14 Para que un cliente de la aplicación pueda ejecutar procesos, en línea, considerados en el sistema el punto de acceso deberá cumplir con los siguientes requisitos mínimos: 512 Mb de memoria RAM y 20 Gb de disco duro.

RNF15 Para que se pueda ejecutar la aplicación el servidor debe contar con los requisitos mínimos: 1 Gb de memoria RAM, 120 de disco duro y un procesador Pentium IV.

Requisitos de usabilidad

RNF16 El sistema deberá facilitar la entrada de datos a los usuarios al presentar campos de selección que permitan escoger los valores. Estos campos contendrán los datos posibles con los que se podrá llenar un determinado elemento en la interfaz.

3.4 Gestión de requisitos

Para la gestión se crearon matrices de trazabilidad de requisitos donde se muestran los cambios que se realizaron en los mismos y que casos de uso se vieron afectados. Pressman define cinco tablas de rastreabilidad para el seguimiento y control, que ayudan a entender cómo el cambio en un requisito afectará diferentes aspectos del sistema que se construirá. Estas son: (5)

- ❖ **Tabla de rastreabilidad de las características:** Muestra la manera en que los requisitos se relacionan con las características del sistema/producto observables para el cliente.
- ❖ **Tabla de rastreabilidad de la fuente:** Identifica la fuente de cada requisito.
- ❖ **Tabla de rastreabilidad de dependencia:** Indica la forma en que los requisitos están relacionados entre sí.
- ❖ **Tabla de rastreabilidad del subsistema:** establece categorías entre los requisitos de acuerdo con el(los) subsistema(s) que gobierna(n).
- ❖ **Tabla de rastreabilidad de la interfaz:** Muestra la forma en que los requisitos se relacionan con las interfaces internas o externas del sistema.

La tabla de rastreabilidad que se utilizará es la de dependencia la cual permite determinar cómo afecta un cambio entre requerimientos.

RF	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	-																								
2		-	X	X																					
3			-																						
4				-																					
5			X	X	-																				
6						-	X	X																	
7							-																		
8								-																	
9									-																
10										-	X														
11											-														
12												-	X												

Tabla 11: Porción de la tabla rastreabilidad de dependencia entre requisitos.

En el Anexo 2 se pueden encontrar las matrices de trazabilidad que muestran los cambios realizados en los requisitos en dos períodos de tiempo y la tabla de rastreabilidad de dependencia.

3.5 Actores del sistema

Un actor del sistema representa el rol que juega una o varias personas, un equipo o un sistema automatizado que interactúa con el mismo. Los trabajadores del negocio que realizan actividades a informatizar son candidatos a actores del sistema, en caso de que algún actor del negocio interactúe con el sistema, entonces también será un actor del mismo. Además los actores del sistema no son parte de él y pueden intercambiar información con él.

A partir de lo anterior y del modelamiento del negocio realizado se identificaron los siguientes actores del sistema:

Actores del Sistema	Descripción
Administrador	Persona responsable de insertar, modificar o eliminar usuarios del sistema.
Lingüista	Persona responsable de insertar, modificar o eliminar descriptores de la lista de descriptores.
Catalogador	Persona encargada de insertar, modificar o eliminar los datos de la descripción física de los documentos al sistema.
Funcionario de Desarrollo de Colecciones	Persona que inserta o modifica las fichas de entrada de información en el sistema.
Especialista Servicios	Persona encargada de insertar o modificar las boletas que registran los servicios que se prestan en el Centro.
Usuario general	Representa a todos los usuarios del sistema teniendo las funcionalidades de visualizar el listado de descriptores, realizar búsquedas y autenticarse.

Tabla 11: Descripción de los Actores del Sistema

3.6 Patrones de Casos de Uso

Un patrón es una solución a un problema en un contexto, codifica conocimiento específico acumulado por la experiencia en un dominio. Para que una solución sea considerada un patrón debe poseer ciertas características. Una de ellas es que debe haber comprobada su efectividad

al resolver problemas similares en ocasiones anteriores. Otra es que debe ser reusable, lo que significa que es aplicable a diferentes problemas en distintas circunstancias.

Algunos de los patrones de casos de uso más utilizados en la actualidad se abordan a continuación: (20)

Patrón Reglas de Negocio

Se basa en la extracción de información originada de las políticas, reglas y regulaciones del negocio de la descripción del flujo y describe la información como una colección de reglas del negocio referenciadas a partir de las descripciones de los casos de uso.

Definición Estática

Este patrón es de tipo descripción, por lo que no influye sobre la estructura del modelo de casos de uso. Es aplicado a todos los casos de uso que modelan servicios que son afectados por reglas de negocio definidas en la organización. Las reglas son descritas en un documento por separado, referenciado por la descripción del caso de uso. Es apropiado cuando no se necesita cambios dinámicos en las reglas del negocio mientras el sistema está en uso.

Patrón Concordancia

Extrae una subsecuencia de acciones que aparecen en diferentes lugares del flujo de casos de uso y es expresado por separado.

Reusabilidad

Es un patrón de estructura que consta de tres casos de uso. El primero llamado subsecuencia común, modela una secuencia de acciones que aparecerán en múltiples casos de uso en el modelo. Los otros casos de uso modelan el uso del sistema que comparte la subsecuencia común de acciones (deben existir al menos dos de ellos).

Adición

En el caso de este patrón alternativo, la subsecuencia común de casos de uso, extiende los casos de uso que comparten la subsecuencia de acciones. Los otros casos de uso modelan el flujo que será expandido con la subsecuencia. Este patrón es preferible usarlo cuando otros casos de uso se encuentran propiamente completos, o sea, que no requieren de una subsecuencia común de acciones para modelar los usos completos del sistema.

Especialización

Contiene casos de uso del mismo tipo. En este caso, estos son modelados como una especialización de casos de uso de tipo de uso común. Todas las acciones en estos casos de uso son heredadas por los casos de uso hijos, donde otras acciones serán adicionadas o acciones heredadas que serán especializadas. Este patrón es aplicable cuando la utilización de

los casos de uso que han sido modelados son del mismo tipo y este tipo debe hacerse visible en el modelo.

Patrón Inclusión Concreta

Inclusión Concreta es un patrón de estructura. Consiste en dos casos de uso y una relación de inclusión entre el caso de uso base y el caso de uso incluido. Este último puede ser instanciado por sí solo. El caso de uso base puede ser concreto o abstracto. Se utiliza este patrón cuando un flujo de datos puede ser incluido en el flujo de otro caso de uso y también puede ejecutarse por sí solo.

Patrón Extensión Concreta

Extensión Concreta es un patrón de estructura. Consiste en dos casos de uso y una relación de extensión entre ellos. El caso de uso extendido es concreto, lo que quiere decir que este puede ser instanciado por sí solo, ser una extensión del caso de uso base. El caso de uso base puede ser concreto o abstracto. Es aplicable cuando un flujo de un caso de uso puede extender el flujo de otro caso de uso, así como ser ejecutado por sí solo.

Patrón CRUD

Este patrón se basa en la fusión de casos de uso simples para formar una unidad conceptual.

Completo

CRUD cuyas siglas significan Creating, Reading, Updating and Deleting, es un patrón de estructura. Propone identificar un caso de uso, llamado “Información CRUD” o “Gestionar Información”, que modela todas las diferentes operaciones que se pueden realizar sobre una parte de información de cierto tipo tal como crearla, leerla, actualizarla y eliminarla. Debe ser usado cuando todos los flujos contribuyen al mismo valor de negocio y estos a su vez son cortos y simples.

Parcial

Este patrón alternativo modela una de las vías de los casos de uso como un caso de uso separado. Es preferiblemente utilizado cuando una de las alternativas de los casos de uso es más significativa, larga o más compleja que las otras.

Patrón Múltiples Actores

Roles diferentes

Captura la concordancia entre actores manteniendo roles separados. Consiste en un caso de uso y por lo menos dos actores. Es utilizado cuando dos actores juegan diferentes roles en un caso de uso, o sea, interactúan de forma diferente con el mismo.

Rol común

Es un patrón de estructura que plantea que cuando dos actores juegan el mismo papel hacia un caso de uso se representa otro actor, del que heredan los actores que comparten este rol. Este patrón es aplicable cuando, desde el punto de vista de un caso de uso, hay solo una entidad externa que interactúe con cada instancia del caso de uso.

3.7 Diagrama de Casos de Uso del Sistema

Cada forma en que los actores usan el sistema se representa como un caso de uso. Los casos de uso son fragmentos de funcionalidades que el sistema ofrece para aportar un resultado de valor para sus actores. De manera más precisa, un caso de uso especifica una secuencia de acciones que el sistema puede llevar a cabo al interactuar con sus actores e incluye alternativas dentro de la secuencia. (9)

Un diagrama de casos de uso del sistema (DCUS) representa gráficamente a los procesos y su interacción con los actores. Se utiliza para ilustrar los requerimientos del sistema y sirve para especificar la comunicación y el comportamiento de un sistema mediante su interacción con los usuarios y otros sistemas.

La siguiente figura muestra el diagrama de casos de uso del sistema a construir:

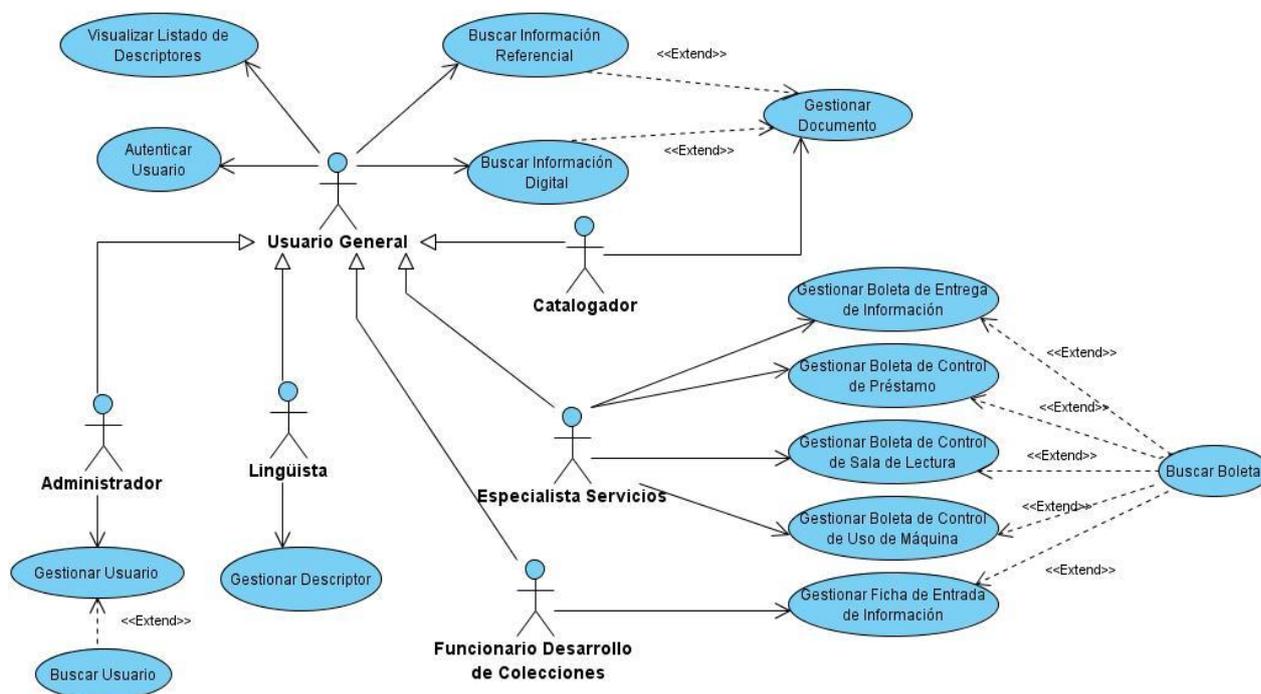


Figura 21: Diagrama de Casos de Uso del Sistema

Para la confección del diagrama se agruparon los requerimientos del sistema en casos de uso y se utilizaron los siguientes patrones de casos de uso:

❖ Reglas del Negocio-Definición Estática: Este patrón fue utilizado porque las reglas del

negocio no cambian dinámicamente cuando el sistema esté en uso.

- ❖ **CRUD-Completo:** Se utilizó este patrón para agrupar funcionalidades que se realizan sobre una unidad de información de un tipo determinado. Su aplicación se evidencia en los siguientes casos de uso: Gestionar Usuario, Gestionar Descriptor, Gestionar Documento, Gestionar Ficha de Entrada de Información, Gestionar Boleta de Control de Préstamo, Gestionar Boleta de Control de Sala de Lectura, Gestionar Boleta de Control de Uso de Máquina y Gestionar Boleta de Entrega de Información.
- ❖ **Múltiples Actores-Roles Comunes:** Este es usado porque existen varios actores que juegan el mismo rol sobre algunos casos de uso. Su aplicación se ilustra con la representación del actor Usuario General del que heredan los actores Administrador, Funcionario de Desarrollo de Colecciones, Catalogador, Lingüista y Especialista de servicios.
- ❖ **Extensión Concreta:** La utilización de este patrón se evidencia en los casos de uso Buscar Información Digital y Buscar Información Referencial pues son casos de uso extendidos que pueden ser instanciados por si solos.
- ❖ **Concordancia-Adición:** La aplicación de este patrón se refleja en el caso de uso Buscar Boleta por ser extendido de varios casos de usos.

3.7.1 Descripción de Casos de Uso del Sistema

A continuación se presentan algunas descripciones de casos de uso significativos del sistema, así como las interfaces de prototipo correspondientes a los mismos. Las restantes descripciones pueden encontrarse a partir del Anexo 3 hasta el 14.

Caso de Uso Gestionar Usuario

Caso de Uso:	Gestionar Usuario	
Actores:	Administrador	
Resumen:	El administrador tiene las opciones de adicionar, modificar y eliminar usuarios del sistema.	
Precondiciones:	El administrador tiene que estar autenticado en el sistema.	
Referencias	RF02, RF03, RF04, CU Buscar Usuario (Extend)	
Prioridad	Alta	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1 El administrador escoge la opción Gestionar Usuario.	2 Muestra un listado con los usuarios existentes en el sistema. 3 Brinda las opciones de Adicionar, Modificar y Eliminar Usuario.	
4 El administrador escoge una de las opciones. Si selecciona: <ul style="list-style-type: none"> - Adicionar Usuario ver Sección 1. - Modificar Usuario ver Sección 2. - Eliminar Usuario ver Sección 3. 		
Sección 1 "Adicionar Usuario"		
Acción del Actor	Respuesta del Sistema	
	1 Muestra una página con los siguientes	

	campos: nombre y apellidos, carné de identidad, usuario, contraseña y rol.
2 El administrador llena los campos.	3 Verificar que no hayan campos en blanco. Si los hay ver Flujo Alterno 1. 4 Validar usuario y contraseña. Si no están correctos ver Flujo Alterno 2. 5 Guarda el Usuario en la base de datos.
Sección 2 "Modificar Usuario"	
Acción del Actor	Respuesta del Sistema
	1 Muestra los datos del usuario llamando al CU Buscar Usuario.
2 El administrador modifica los datos.	3 Verificar que no hayan campos en blanco. Si los hay ver Flujo Alterno 1. 4 Validar usuario y contraseña. Si no están correctos ver Flujo Alterno 2. 5 Guardar los cambios realizados.
Sección 3 "Eliminar Usuario"	
Acción del Actor	Respuesta del Sistema
	1 Muestra los datos del usuario llamando al CU Buscar Usuario.
2 El administrador selecciona la opción eliminar.	3 Muestra un mensaje: ¿Desea eliminar el usuario seleccionado? y las opciones: Aceptar o Cancelar. En caso de escoger cancelar se termina el caso de uso. 4 Elimina el usuario de la base de datos.

Prototipo de Interfaz

Adicionar Usuario

Nombre y apellidos:

Carné de identidad:

Usuario:

Contraseña:

Roles

Rol: ▼

Modificar Usuario

Nombre y apellidos:

Carné de identidad:

Usuario:

Contraseña:

Roles

Rol:

Eliminar Usuario

Nombre y apellidos:

Carné de identidad:

Usuario:

Contraseña:

Roles:

Flujos Alternos

Flujo Alternativo 1

Acción del Actor	Respuesta del Sistema
	1 Muestra un mensaje: Debe llenar todos los campos. 2 Volver al paso 2 de la Sección 1 ó 2.

Flujo Alternativo 2

Acción del Actor	Respuesta del Sistema
	1 Muestra un mensaje: Usuario o contraseña incorrectos. 2 Volver al paso 2 de la Sección 1 ó 2.

Poscondiciones	Queda insertado, modificado o eliminado un usuario en el sistema.
-----------------------	---

Caso de Uso Buscar Información Digital

Caso de Uso:	Buscar Información Digital
Actores:	Usuario General
Resumen:	El usuario especifica los criterios por los que quiere buscar un documento determinado y el sistema le muestra los resultados de la búsqueda.
Precondiciones:	
Referencias	RF25
Prioridad	Alta

Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1 El usuario selecciona la opción Buscar Información Digital.	2 Muestra una ventana con los criterios de búsqueda. Si el documento es de tipo: <ul style="list-style-type: none"> - Disposición del Consejo de Gobierno ver Sección 1. - Legislación ver Sección 2. - Sentencia ver Sección 3. - Investigación ver Sección 4. - Doctrina ver Sección 5.
3 El usuario llena los criterios seleccionados.	4 Muestra una lista con los documentos que cumplan con los criterios establecidos.
Sección 1 “Buscar Disposición del Consejo de Gobierno”	
Acción del Actor	Respuesta del Sistema
	1 Muestra una ventana con los siguientes criterios de búsqueda: tipo de disposición, número, año, descriptores, texto o frase. Si el usuario es un especialista se muestra: tipo de disposición, número, año, descriptores, fecha de entrada al sistema, especialista, texto o frase. 2 Ir al paso 3 del Flujo Normal de Eventos.
Sección 2 “Buscar Legislación”	
Acción del Actor	Respuesta del Sistema
	1 Muestra una ventana con los siguientes criterios de búsqueda: tipo de norma, materia, número de la norma, nombre de la norma, órgano emisor, fecha de emisión, notas, descriptores, texto o frase. Si el usuario es un especialista se muestra: tipo de norma, materia, número de la norma, nombre de la norma, órgano emisor, fecha de emisión, gaceta oficial, año de publicación, número de la gaceta, descriptores, texto o frase, fecha de entrada al sistema, especialista. 2 Ir al paso 3 del Flujo Normal de Eventos.
Sección 3 “Buscar Sentencia”	
Acción del Actor	Respuesta del Sistema
	1 Muestra una ventana con los siguientes criterios de búsqueda: tipo de sentencia, número, año, ponente, tribunal, sala, texto, fallo, causales, descriptores. 2 Ir al paso 3 del Flujo Normal de Eventos.
Sección 4 “Buscar Investigación”	
Acción del Actor	Respuesta del Sistema
	1 Muestra una ventana con los siguientes criterios de búsqueda: autor, país del autor, título del trabajo, año de presentación, presentado en, tipo de trabajo, notas, descriptores, texto o frase. Si el usuario es un especialista se muestra: autor, procedencia del autor, categoría del autor, país del autor, título del trabajo,

Una de las técnicas seleccionadas para realizar la validación de los requerimientos del sistema a construir fue la de prototipos. Se crearon uno o varios prototipos de interfaz de usuario por cada caso de uso, son estos un conjunto de interfaces que muestran una simulación de lo que sería el futuro sistema a implementar.

Los prototipos fueron diseñados de forma exitosa con la herramienta Microsoft Office Visio 2007. Después de confeccionados, se realizó una reunión con los clientes que tuvo como objetivo y resultado la aceptación por parte de los mismos de la propuesta realizada. Los prototipos asociados a cada caso de uso del sistema pueden encontrarse en sus descripciones (Ver anexos del 3 al 14).

❖ Matriz de trazabilidad

Otra técnica que se utilizó para la validación de requisitos fue la matriz de trazabilidad, en este caso para verificar que no existan inconsistencias en la inclusión de los requisitos en los casos de uso. Esto es de gran importancia porque cada requerimiento debe quedar reflejado en al menos un caso de uso para evitar que no se satisfaga alguna de las necesidades del cliente.

Esta matriz muestra todos los requisitos funcionales del sistema de forma vertical y los casos de uso del sistema de forma horizontal, de manera que puedan hacerse corresponder entre ellos. A continuación se muestra la matriz de trazabilidad realizada en la que se aprecia como resultado que todos los requisitos fueron asociados a un caso de uso para cubrir una determinada funcionalidad del sistema. (Ver Anexo 15).

3.8.2 Métrica para la calidad de la especificación de los requisitos de software

La métrica para la calidad de la especificación de los requisitos de software se basa en la consistencia de las interpretaciones del grupo de revisión para cada uno de ellos. Mediante el uso de la misma se realizó la validación de los requisitos para comprobar que no exista ambigüedad.

Esta métrica consiste en realizar una división entre número de requisitos para los que todos los revisores tuvieron interpretaciones idénticas (n_{ui}) y la cantidad de requisitos de software (n_r):

$$Q_1 = n_{ui} / n_r, \quad n_r = n_f + n_{nf}$$

Donde n_f es el número de requisitos funcionales y n_{nf} es el número de requisitos no funcionales. Cuanto más cerca de uno esté el valor de Q_1 menor será la ambigüedad de la especificación. (21)

La siguiente tabla muestra los nombres y el rol de cada uno de los miembros del equipo de revisión:

Nombre y Apellidos	Rol desempeñado
Ing. Heiler Fabars Corrales	Jefe del Proyecto CENDIJ
Yordanka Carralero Domínguez	Analista
Anais Louit Moreno	Analista

Tabla 12: Descripción del Grupo de Revisores

A continuación se muestra un resumen de los resultados obtenidos:

Atributo de Calidad	Tipo de Requisito	Interpretaciones	
		Iguales	Desiguales
Especificidad	Funcionales	24	1
	No Funcionales	14	2
	Total	38	3

Tabla 13: Representación de los resultados de las interpretaciones

$$n_f=25 \quad n_{nf}=16 \quad n_r = n_f + n_{nf} = 25+16=41$$

$$Q_1 = n_{ui} / n_r = 38 / 41 = \mathbf{0.926}$$

Q_1 resultó ser un valor cercano a 1. Con este resultado queda demostrado que existe un bajo nivel de ambigüedad en la especificación de los requisitos de software.

3.8.3 Métricas para la calidad de la funcionalidad del Diagrama de Casos de Uso del Sistema

Las métricas para la calidad de la funcionalidad del diagrama de casos de uso del sistema proponen un conjunto de factores con métricas asociadas a cada uno de ellos que permiten evaluar cuatro atributos fundamentales de calidad. Estos atributos son:

- ❖ **Compleitud:** grado en que se ha incluido de forma clara y concisa todos los elementos necesarios para la descripción.
- ❖ **Consistencia:** grado en que los elementos del DCUS representan en forma única y no contradictoria un aspecto del problema.
- ❖ **Correctitud:** grado de adecuación del DCUS para satisfacer los requisitos.
- ❖ **Complejidad:** grado de claridad y re-uso del DCUS.

Factores	Métricas asociadas
Compleitud	
1. ¿Están definidos todos los requisitos que justifican la funcionalidad del caso de uso?	1. Número de requisitos omitidos por caso de uso. 2. Número de casos de uso que tienen requisitos omitidos.
2. ¿Existen requisitos que no han sido considerados en algún caso de uso?	3. Número de requisitos que no son considerados en ningún caso de uso.
3. ¿Todos los casos de uso del sistema han sido clasificados de acuerdo a su relevancia (alto, medio o bajo)?	4. Número de casos de uso que no han sido clasificados.
4. ¿Están todas las acciones del flujo de eventos redactadas en función del responsable?	5. Número de acciones del flujo de eventos que no están redactadas en función del responsable. 6. Número de casos de uso que tienen acciones del flujo de eventos no redactados en función del responsable.
5. ¿Han sido definidos todos los roles relevantes de usuario encargados de generar/modificar o consultar información?	7. Número de roles relevantes omitidos.
Consistencia	
6. ¿El nombre dado a los casos de uso es una expresión verbal que describe alguna funcionalidad relevante en el contexto del usuario?	8. Número de casos de uso que tienen un nombre incorrecto.
7. ¿La descripción del flujo de eventos se inicia con la descripción de una acción externa originada por un actor o por una condición interna del sistema claramente identificable?	9. Número de casos de uso cuya descripción extendida no inicia con una acción externa o con una condición monitoreada por el sistema.
8. ¿Existen acciones en el flujo de eventos asignadas a un responsable que no le corresponde?	10. Número de acciones del flujo de eventos que no se corresponde con la definición del responsable 11. Número de casos de uso que tienen acciones del flujo de eventos asignados a un responsable que no le corresponde.
9. ¿Está adecuadamente redactado (en el lenguaje del usuario) el flujo de eventos?	12. Número de casos de uso no aceptados.
10. ¿Existe una adecuada separación entre el flujo básico de eventos y los flujos alternos y/o	13. Número de casos de uso complejos que no tienen separación del flujo básico y de flujos alternos.

flujos subordinados?	
Correctitud	
11. ¿Representa el caso de uso requisitos comprensibles por el usuario?	14. Número de casos de uso en que los requisitos representados no son comprensibles por el usuario.
12. ¿Se ajusta la representación del diagrama del caso de uso de acuerdo a lo normado en la metodología?	15. Grado en que se ajusta el diagrama del caso de uso a la metodología.
13. ¿Las interacciones definidas describen la funcionalidad requerida del sistema?	16. Número de casos de uso que deben ser modificados para adecuarlos a la funcionalidad del sistema.
Complejidad	
14. ¿Los elementos dentro del diagrama están adecuadamente ubicados de manera que facilitan su interpretación?	17. Número de elementos del diagrama que requieren reubicación.

Tabla 14: Métricas para la calidad de la funcionalidad del Diagrama de Casos de Uso de Sistema

Como el diagrama de casos de uso del sistema es uno de los artefactos más importante del flujo de trabajo requerimiento, se realizaron dos revisiones en aras de asegurar que el sistema tenga la calidad requerida.

La primera revisión se realizó cuando se efectuaron los cambios en los requisitos que se evidencian en la matriz del primer período de la gestión. En esta se comprobó que el diagrama construido no cumplía con algunas métricas que evalúan los atributos de calidad. Estas métricas fueron:

- ❖ Para el atributo completitud: 4, 5, 6.
- ❖ Para el atributo consistencia: 10, 11, 13.
- ❖ Para el atributo correctitud: 14.

Las métricas se aplicaron por segunda ocasión una vez realizadas las modificaciones en los requerimientos en el segundo período de la gestión, demostrándose que el diagrama cuenta con un 100% de calidad de su funcionalidad.

A continuación se muestra una gráfica donde se aprecian los resultados obtenidos de la evaluación de los atributos de calidad:

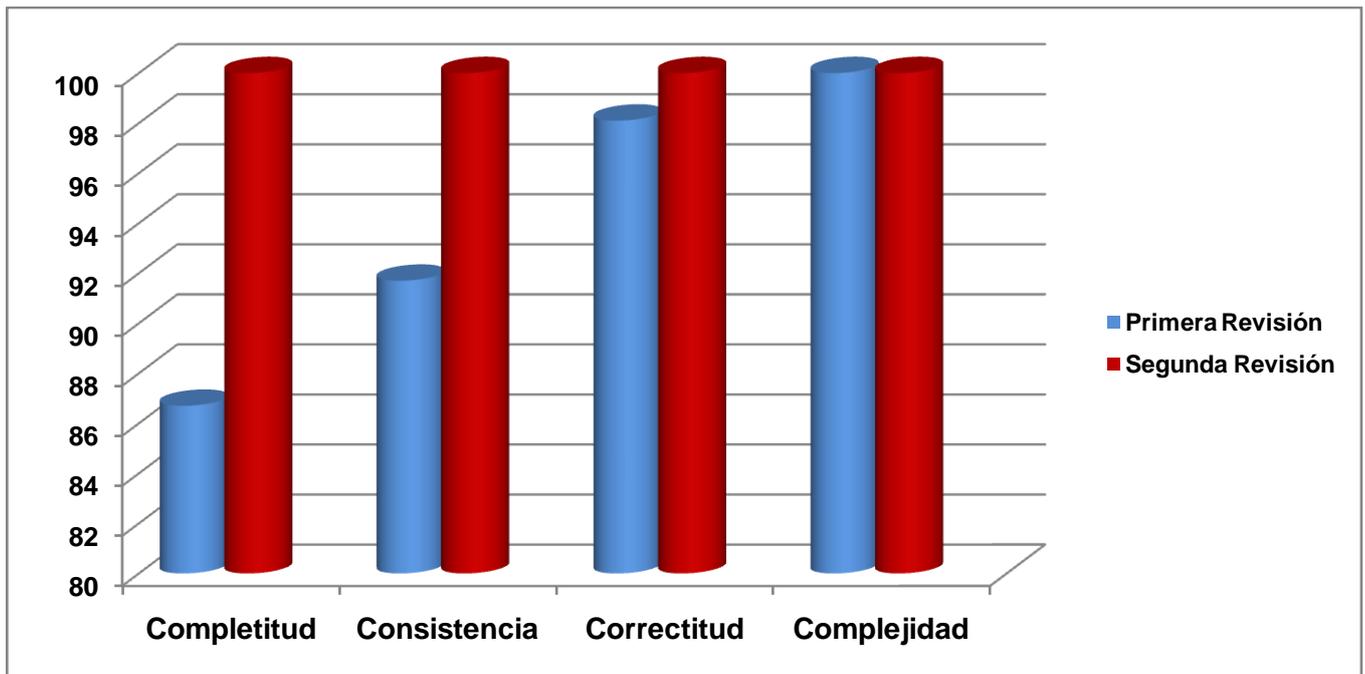


Figura 22: Resultados de la evaluación

Conclusiones

En este capítulo, mediante la puesta en práctica de algunas etapas de la Ingeniería de Requisitos, se obtuvo un listado de los requisitos que representan las necesidades de los clientes. Se construyó el diagrama de casos de uso del sistema y se realizaron las descripciones de los casos de uso para comprender cómo interactuarán los usuarios con el sistema. Otro punto abordado fue la validación de la solución del sistema propuesta con el objetivo de evitar errores que pudieran resultar altamente costosos en una etapa posterior del desarrollo del sistema, para lo cual se utilizaron prototipos de interfaz de usuario, matriz de trazabilidad y métricas.

CONCLUSIONES GENERALES

Después de realizada la presente investigación se arriba a las siguientes conclusiones:

- ❖ La realización del modelamiento del negocio permitió comprender la estructura y dinámica de la organización, así como los principales problemas que allí existen. Posibilitó además la identificación de las actividades a informatizar.
- ❖ Aplicando las etapas de la Ingeniería de Requisitos se identificaron las necesidades de los clientes para obtener los requerimientos de software que representan las funcionalidades del sistema a construir.
- ❖ La aplicación de técnicas y métricas de validación posibilitó verificar que la especificación de requisitos y el DCUS fueron elaborados con la calidad requerida.
- ❖ Con la confección de los artefactos: Modelo de Negocio, Especificación de Requisitos y Descripción de Casos de Uso del Sistema se logró una visión común entre clientes y desarrolladores.

RECOMENDACIONES

Después de finalizar la presente investigación se sugieren las siguientes recomendaciones:

- ❖ Desarrollar los restantes flujos de trabajo propuestos por RUP, de manera que se obtenga como resultado un sistema capaz de informatizar los procesos del CENDIJ.
- ❖ Continuar la gestión de requisitos para llevar un seguimiento y control de los requerimientos de software en iteraciones posteriores.

BIBLIOGRAFÍA

1. **Fernández Sánchez, Ing. Leidy.** *Procedimiento para el desarrollo del proceso de ingeniería de requisitos en un proyecto software (PROCIR)*. 2006.
2. **Frederick P. Brooks, Jr.** *The Mythical Man-Month*. s.l. : Addison-Wesley, 1995.
3. **Sommerville, Ian.** *Ingeniería del Software*. Madrid : Addison Wesley, 2005.
4. **IEEE, Std 610.12-1990.** *IEEE Standard Glossary of Software Engineering Terminology*.
5. **Pressman, Roger S.** *Ingeniería del Software: Un enfoque práctico*. 2005.
6. **Jurado, Ing. Jose Luis.** *INGENIERIA DE REQUERIMIENTOS (IR)*. Facultad de Ingeniería Electrónica y Telecomunicaciones, Universidad del Cauca. Colombia : s.n. Disponible en http://pis.unicauca.edu.co/moodle/file.php/291/Ingenieria_de_Requerimientos.pdf.
7. **Arias Chávez, Michel.** *LA INGENIERÍA DE REQUERIMIENTOS Y SU IMPORTANCIA EN EL DESARROLLO DE PROYECTOS DE SOFTWARE*. Universidad de Costa Rica : s.n., 2006.
8. **José Escalona, María y Koch, Nora.** *Ingeniería de Requisitos en Aplicaciones para la Web. Un estudio comparativo*. Lenguajes y Sistemas Informáticos, Escuela Técnica Superior de Ingeniería Informática. Universidad de Sevilla : s.n., 2002.
9. **Rumbaugh, James, Jacobson, Ivar y Booch, Grady.** *El proceso unificado de desarrollo de software*. s.l. : Addison Wesley, 2000.
10. **Canós, José H., Letelier, Patricio y Penadés, M. Carmen.** *Metodologías Agiles en el desarrollo de software*. Universidad Politécnica de Valencia : s.n.
11. **Mendoza Sánchez, María A.** *Metodologías De Desarrollo De Software*. 2004.
12. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James.** *El Lenguaje Unificado de Modelado. Manual de Referencia*. s.l. : Addison Wesley, 2000.
13. **Hernández Orallo, Enrique.** *El Lenguaje Unificado de Modelado (UML)*. Disponible en <http://www.disca.upv.es/enheror/pdf/ActaUML.PDF>.
14. **BizAgi.** bizagi. [En línea] [Citado el: 17 de Febrero de 2011.] Disponible en <http://www.bizagi.com/docs/BPMNbyExampleSPA.pdf>. <http://www.bizagi.com/>.
15. **Enterprise Architect .** Spark Systems. [En línea] 2007. [Citado el: 13 de Febrero de 2011.] [Disponible en <http://www.sparxsystems.com.ar/products/ea.html>]. <http://www.sparxsystems.com.ar>.
16. **Visual Paradigm International Ltd.** Free Download Manager - Sitio de descargas de software. [En línea] 5 de Marzo de 2007. [Citado el: 13 de Febrero de 2011.] [Disponible en http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%28M%C3%8D%29_14720_p/]. <http://www.freedownloadmanager.org/es>.

17. **Microsoft Corporation.** Microsoft Office Online. [En línea] 23 de Mayo de 2006. [Citado el: 18 de Febrero de 2011.] Disponible en <http://www.microsoft.com/latam/office/preview/programs/visio/overview.mspx>.
<http://www.microsoft.com/>.
18. **Gómez, Julián.** Softonic. *Softonic*. [En línea] [Citado el: 21 de febrero de 2011.] Disponible en: <http://axure-rp.softonic.com/> . <http://www.softonic.com/>.
19. **Schmuller, Joseph.** *Aprendiendo UML en 24 horas*. Mexico : PEARSON EDUCACION LATINOAMERICA, 2000.
20. **Martínez, Alejandro y Martínez, Raúl.** *Guía a Rational Unified Process*. Universidad de Castilla la Mancha.
21. **Övergaard, Gunnar y Palmkvist, Karin.** *Use Cases Patterns and Blueprints*. s.l. : Addison-Wesley Professional, 2004 .
22. **Davis, Alan, y otros.** Identifying and Measuring Quality in a Software Requirements Specification. *Software Requirements Specification*. Segunda Edición. Baltimore : s.n., 1993, Capítulo 3.
23. **WHITE, STEPHEN A., MIERS, DEREK y PhD.** *BPMN Guía de Referencia y Modelado*. s.l. : Future Strategies, 2009.

GLOSARIO DE TÉRMINOS

CENDIJ: Centro Nacional de Documentación e Información Judicial.

Doctrina: Conjunto de ideas u opiniones sustentadas por una persona o un grupo.

Doctrina Legal: La doctrina establecida por los tribunales al interpretar y aplicar de modo reiterado e idéntico las leyes, y que referida al Tribunal Supremo, configura la jurisprudencia.

Legislación: Conjunto de leyes por las cuales se gobierna un Estado o una materia determinada.

Flujo Ascendente: Investigaciones que no llegan a tener el rótulo de informe científico pero cuyo nivel de profundidad merece que sean tenidas en cuenta. Ejemplo: tesis, actas de congresos, boletines, cuadernos de trabajo, informes técnicos y autobiografías.

Sentencia: Acto del Órgano jurisdiccional que pone término al proceso resolviendo todas las cuestiones litigiosas planteadas por las partes.

Metodología de desarrollo de software: Conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a los desarrolladores a estructurar, planificar y controlar el proceso de desarrollo de software.

Ingeniería de Requisitos: Disciplina para desarrollar una especificación completa, consistente y no ambigua de los requisitos, la cual servirá como base para acuerdos comunes entre todas las partes involucradas y en dónde se describen las funciones que realizará el sistema.

CASE: Acrónimo de Computer Aided Software Engineering (Ingeniería de Software Asistida por Ordenador), son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software y reducir el costo de las mismas en términos de tiempo y de dinero.

IEEE: Corresponde a las siglas de The Institute of Electrical and Electronics Engineers (Instituto de Ingenieros Eléctricos y Electrónicos), el cual se desempeña como asociación técnico-profesional, sin fines de lucro, más grande del mundo formada por profesionales de todas las disciplinas de la ingeniería, dedicada a la estandarización.

Métricas de calidad de software: Proporcionan una indicación de cómo se ajusta el software a los requisitos implícitos y explícitos del cliente.