

Universidad de las Ciencias Informáticas

Facultad 3



**Título: Análisis del Módulo de Soporte del
Proyecto Informatización del
Convenio Cuba Venezuela**

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

Autor: Yurisleidys Leiva Zuñiga

Tutores: Ing. Dairo Reyes Rodríguez

Ing. Yaniet Piñeiro Pérez

Junio 2008

DECLARACIÓN DE AUTORÍA

Declaro ser autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Yurisleidys Leiva Zuñiga

Dairo Reyes Rodríguez

Yaniet Piñeiro Pérez

Firma del Autor

Firma del Tutor

Firma del Tutor

“Si una persona es perseverante, aunque sea dura de entendimiento, se hará inteligente; y aunque sea débil se transformará en fuerte.”

Leonardo Da Vinci

AGRADECIMIENTOS

A la Revolución por crear esta maravillosa universidad y permitir formarme en esta profesión.

A mi mamá, mi papá y Magalys por confiar y estar pendientes de mí en todo momento.

A mi Pepito por apoyarme durante todo este tiempo.

A mis amigos Yenicé, Fernandito y Enrique por mantenerse en contacto y compartir las nuevas experiencias de estos 5 años.

A mis tutores por preocuparse tanto, especialmente a Dairo por ayudarme en todo.

Y a todas aquellas personas que de una forma u otra me han ayudado a llegar hasta donde estoy hoy.

RESUMEN

En el año 2000 se firmó entre Cuba y Venezuela un Convenio Integral de Cooperación, con el propósito de desarrollar programas y proyectos en conjunto que beneficiaran a ambos países, para ser ejecutados por organismos y entidades de los sectores públicos de estos. Estos proyectos se firman durante las Comisiones Mixtas que se realizan cada cierto período. Actualmente no se encuentra automatizado todo el proceso que se lleva a cabo desde la firma hasta la culminación de dichos proyectos, y es por esto que en la Universidad de las Ciencias Informáticas surge el Proyecto Informatización del Convenio Cuba Venezuela, cuyo objetivo es desarrollar un sistema que permita tener el control de lo anteriormente expuesto.

El presente trabajo muestra el análisis del Módulo Soporte, necesario para posteriormente lograr una correcta implementación del mismo, que cumpla con las necesidades reales del cliente. Para ello se realizó un estudio sobre las metodologías y herramientas para el desarrollo de software que son más utilizadas en la actualidad, las actividades que realiza el analista de sistemas, los objetivos de la Ingeniería de Requisitos, las técnicas para la captura de requisitos y los patrones de casos de usos que pueden ser aplicados para lograr una correcta modelación de estos. Finalmente se generaron los artefactos como resultado del proceso de análisis del módulo, los cuales constituyeron la entrada del siguiente flujo de trabajo.

PALABRAS CLAVE

Convenio Cuba Venezuela, proyectos, análisis, módulo, soporte, artefactos.

TABLA DE CONTENIDOS

AGRADECIMIENTOS.....	I
RESUMEN.....	II
INTRODUCCIÓN.....	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	6
1.1 Metodologías para el Desarrollo de Software.....	7
1.1.1 XP (Extreme Programming).....	8
1.1.2 RUP (Rational Unified Process).....	10
1.1.3 Selección de la metodología de desarrollo de software.....	15
1.2 Analista de Sistemas.....	16
1.3 Ingeniería de Requisitos.....	17
1.4 Técnicas para la Captura de Requisitos.....	20
1.4.1 Entrevistas.....	20
1.4.2 Cuestionarios.....	20
1.4.3 Tormenta de ideas.....	21
1.4.4 JAD (Joint Application Development).....	21
1.5 Patrones de Casos de Uso.....	22
1.5.1 Reglas de negocio.....	22
1.5.2 Concordancia.....	23
1.5.3 Extensión concreta o inclusión.....	24
1.5.4 CRUD (Creating, Reading, Updating, Deleting).....	25
1.5.5 Múltiples actores.....	26
1.6 Herramientas para el Modelado.....	27
1.6.1 Rational Rose.....	28
1.6.2 Enterprise Architect.....	29
1.6.3 Visual Paradigm.....	29
1.6.4 Selección de la herramienta de modelado.....	30
1.7 Conclusiones Parciales.....	30

CAPÍTULO 2: SOLUCIÓN PROPUESTA	31
2.1 Captura de Requisitos. Técnicas, Métodos y Plantillas a utilizar	32
2.2 Modelo del Dominio	33
2.2.1 Descripción general del dominio	33
2.2.2 Especificación de las clases conceptuales del modelo del dominio.....	35
2.2.2.1 Descripción de la clase: Convenio Cuba Venezuela	35
2.2.2.2 Descripción de la clase: Comisión Mixta.....	35
2.2.2.3 Descripción de la clase: País	35
2.2.2.4 Descripción de la clase: Secretaría Técnica	35
2.2.2.5 Descripción de la clase: Ministerio.....	35
2.2.2.6 Descripción de la clase: Ente Ejecutor.....	36
2.2.2.7 Descripción de la clase: Persona.....	36
2.2.2.8 Descripción de la clase: Responsable	36
2.2.2.9 Descripción de la clase: Directivo de Gobierno	36
2.2.2.10 Descripción de la clase: Directivo de Convenio	36
2.2.2.11 Descripción de la clase: Proyecto	36
2.3 Especificación de los Requisitos de Software.....	37
2.3.1 Requisitos funcionales	37
2.3.2 Requisitos no funcionales	39
2.4 Modelo de Casos de Uso del Sistema	41
2.4.1 Descripción general del sistema	41
2.4.2 Aplicación de patrones del análisis	44
2.4.3 Especificación de los casos de uso del sistema.....	45
2.4.3.1 Descripción del caso de uso: Autenticar	45
2.4.3.2 Descripción del caso de uso: Gestionar Usuario	47
2.4.3.3 Descripción del caso de uso: Gestionar Usuario a Nivel de Secretaría Técnica	57
2.4.3.4 Descripción del caso de uso: Gestionar Usuario a Nivel de Ministerio.....	59
2.4.3.5 Descripción del caso de uso: Modificar Datos Personales	61
2.4.3.6 Descripción del caso de uso: Gestionar Ministerio	62
2.4.3.7 Descripción del caso de uso: Gestionar Ente Ejecutor	69
2.4.3.8 Descripción del caso de uso: Listar Usuarios	77
2.4.3.9 Descripción del caso de uso: Listar Usuarios a Nivel de Ministerio.....	79

2.4.3.10 Descripción del caso de uso: Listar Usuarios a Nivel de Ente Ejecutor.....	80
2.4.3.11 Descripción del caso de uso: Listar Ministerios	81
2.4.3.12 Descripción del caso de uso: Listar Entes Ejecutores	82
2.4.3.13 Descripción del caso de uso: Listar Entes Ejecutores a Nivel de Ministerio	84
2.5 Conclusiones Parciales.....	85
CAPÍTULO 3: RESULTADOS OBTENIDOS	86
3.1 Resultados del Modelo del Dominio	86
3.2 Resultados de la Especificación de Requisitos	86
3.3 Resultados del Modelo de Casos de Uso del Sistema	87
3.3.1 Prototipo del sistema.....	87
3.3.2 Evaluación de la calidad.....	88
3.4 Aceptación del Cliente	92
3.5 Conclusiones Parciales.....	95
CONCLUSIONES	96
RECOMENDACIONES.....	97
BIBLIOGRAFÍA.....	98
ANEXOS	101
Anexo 1: Plantilla para el Modelo del Dominio	101
Anexo 2: Plantilla para la Especificación de Requisitos	103
Anexo 3: Plantilla para el Modelo de Casos de Uso del Sistema	107
Anexo 4: Plantilla para el Glosario de Términos.....	110
Anexo 5: Factores y Métricas del Modelo de Casos de Uso del Sistema.....	112
GLOSARIO	117

ÍNDICE DE FIGURAS

Figura 1.1: Capas de la Ingeniería de Software	6
Figura 1.2: Hitos y Fases de RUP	12
Figura 1.3: RUP en Dos Dimensiones	14
Figura 1.4: Características de un Analista de Sistemas	16
Figura 2.1: Diagrama de Clases Conceptuales del Dominio	34
Figura 2.2: Diagrama de Casos de Uso del Sistema	43
Figura 3.1: Paradigma de Construcción de Prototipos	88
Figura 3.2: Grado de Calidad de la Funcionalidad del Modelo de Casos de Uso del Sistema.....	91

ÍNDICE DE TABLAS

Tabla 2.1: Descripción del Caso de Uso Autenticar	46
Tabla 2.2: Descripción del Caso de Uso Gestionar Usuario	56
Tabla 2.3: Descripción del Caso de Uso Gestionar Usuario a Nivel de Secretaría Técnica	58
Tabla 2.4: Descripción del Caso de Uso Gestionar Usuario a Nivel de Ministerio	60
Tabla 2.5: Descripción del Caso de Uso Modificar Datos Personales	61
Tabla 2.6: Descripción del Caso de Uso Gestionar Ministerio	68
Tabla 2.7: Descripción del Caso de Uso Gestionar Ente Ejecutor	76
Tabla 2.8: Descripción del Caso de Uso Listar Usuarios	78
Tabla 2.9: Descripción del Caso de Uso Listar Usuarios a Nivel de Ministerio	79
Tabla 2.10: Descripción del Caso de Uso Listar Usuarios a Nivel de Ente Ejecutor	80
Tabla 2.11: Descripción del Caso de Uso Listar Ministerios	81
Tabla 2.12: Descripción del Caso de Uso Listar Entes Ejecutores	83
Tabla 2.13: Descripción del Caso de Uso Listar Entes Ejecutores a Nivel de Ministerio	84
Tabla 3.1: Atributos y Factores para la Evaluación de la Calidad del Modelo de Casos de Uso del Sistema	90

INTRODUCCIÓN

La República de Cuba y la República Bolivariana de Venezuela mantienen desde hace algunos años estrechos lazos de amistad entre sus pueblos y excelentes relaciones entre sus gobiernos. En el marco de la Alternativa Bolivariana para las Américas (ALBA), ambas naciones, conscientes de su interés común por promover y fomentar el progreso de sus respectivas economías, y las ventajas recíprocas que resultan de una cooperación que tenga resultados efectivos en el avance económico y social de ambos países y la integración de América Latina y el Caribe, se comprometieron a elaborar de común acuerdo, programas y proyectos de cooperación. Para esto existe un Convenio donde estos países firman acuerdos para desarrollar proyectos conjuntos que beneficien a ambas partes. Estos proyectos se firman cada cierto tiempo durante las Comisiones Mixtas. Estas Comisiones Mixtas son el mecanismo financiero para los acuerdos a nivel de gobierno, que traen como consecuencia pasar a una fase posterior.

Para la ejecución de estos programas y proyectos de cooperación, se considera la participación de organismos y entidades de los sectores públicos de ambos países. Cuando es necesario, también se incluye la participación de las universidades, organismos de investigación y de organizaciones no gubernamentales.

Existen por la parte cubana y la venezolana toda una estructura administrativa que permite la coordinación para que fluyan los proyectos aprobados en las Comisiones Mixtas. El primer nivel está constituido por los Directivos del Gobierno seguido de los Directivos del Convenio. En un nivel inferior están las Secretarías Técnicas, que se componen de un coordinador de la Secretaría Técnica por cada una de las partes y un grupo de trabajo de esta. Dicho grupo está formado por un jefe de grupo, especialistas que atienden a los diferentes Ministerios u Organismos, un especialista que atiende la parte financiera y uno que atiende las misiones. Subordinados a este nivel de Secretaria Técnica se encuentran los Ministerios correspondientes a cada país, en estos se incluye tanto el ministro como el responsable ejecutor. Así mismo, los Ministerios incluyen a un conjunto de entidades que se desempeñan como ejecutoras de los proyectos, estos Entes Ejecutores se componen de un director y un responsable que atiende la cooperación en el marco de la Comisión Mixta. Las entidades dirigen a un grupo de equipos que desarrollan cada uno de los proyectos, siendo este el último nivel en la jerarquía.

El Convenio ha crecido y ganado en diversidad. Ha provocado un impacto positivo en importantes sectores del desarrollo económico y social de Cuba y Venezuela.

Actualmente este proceso se realiza de forma manual, lo que hace más engorroso y difícil el trabajo de ambos países en cuanto a este aspecto. Además, esto trae como consecuencia que no se tenga un control estricto y centralizado de los proyectos que se han firmado, ni del monto en efectivo relativo a estos. Por todo lo anteriormente expuesto ha sido propuesto iniciar en la Universidad de las Ciencias Informáticas, un proyecto para automatizar las actividades que se efectúan antes, durante y después de realizada la Comisión Mixta de cada año. Para esto, se desarrollará una aplicación Web donde puedan acceder las dos partes y de esta forma mejorar el modo en que se manejan los proyectos firmados dentro del marco del Convenio.

El proyecto Informatización del Convenio Cuba Venezuela consta de dos fases. La primera, tiene como objetivo desarrollar la Aplicación Inicial del Sistema de Gestión para el Seguimiento de los Proyectos, que se basa principalmente en controlar de forma general el seguimiento de los proyectos de la VI, VII y VIII Comisión Mixta. En la segunda fase se desarrollará una aplicación que permita la automatización de los procesos de definición de la propuesta, conceptualización y alcance, contratación, gestión de ejecución y control de la evolución de los proyectos de colaboración y sus diferentes partes, la gestión de las misiones y del pago o cancelación de obligaciones de las partes involucradas según lo acordado en los contratos, entre otras actividades. Esta fase inicial consta de los módulos Soporte y Entrada de Datos para el desarrollo del sistema.

En el proceso de desarrollo de un software intervienen de forma colaborativa varios roles que hacen posible la confección del producto final, entre los que se encuentran: analistas, arquitectos, diseñadores y desarrolladores. Dentro de este proceso la inexistencia de un analista del sistema puede provocar situaciones como: que no exista una visión detallada por parte del equipo del proyecto de los procesos que se llevan a cabo, ni se comprenda la estructura y dinámica de la organización para la que se implementará el sistema, y además, que el software finalmente implementado no cumpla con las necesidades reales que tiene el cliente.

A partir de esta información surge el siguiente **problema**: ¿Cómo realizar el análisis del Módulo Soporte del proyecto Informatización del Convenio Cuba Venezuela, para la implementación de un software que cumpla con las necesidades del cliente?

El **objeto de estudio** lo constituye la Ingeniería de Requisitos dentro del proceso de desarrollo de proyectos productivos.

El **campo de acción** determinado por el objeto de estudio planteado es la identificación, análisis, especificación y validación de los requisitos del Módulo Soporte del proyecto Informatización del Convenio Cuba Venezuela.

El **objetivo general** de este trabajo es la realización del análisis del Módulo Soporte del proyecto Informatización del Convenio Cuba Venezuela, para lograr una posterior implementación de un software que cumpla con las necesidades que posee el cliente. Para darle cumplimiento a este objetivo se plantean los siguientes **objetivos específicos**:

- Desarrollar un estudio sobre el estado del arte de las metodologías y las herramientas que pueden ser utilizadas.
- Documentar los artefactos relacionados con el análisis del sistema.
- Evaluar los artefactos obtenidos durante el proceso de análisis.

Esta investigación está basada en la siguiente **hipótesis**: si se realiza un correcto análisis del Módulo Soporte del proyecto Informatización del Convenio Cuba Venezuela, entonces se logrará la implementación de un software que cumpla con las necesidades del cliente.

Se establecen entonces las siguientes **tareas de la investigación**:

- Estudiar las metodologías y herramientas de modelado para el desarrollo de software.
- Definir la estrategia para la captura de los requisitos del sistema.
- Realizar la modelación del dominio.
- Identificar los requisitos funcionales y no funcionales del sistema.
- Estructurar el modelo de casos de uso del sistema.
- Especificar los casos de uso del sistema.
- Definir el prototipo no funcional del sistema.

La estrategia de investigación a seguir es la exploratoria, pues se realizará un estudio acerca de las diferentes metodologías para el desarrollo de software, herramientas para el modelado, patrones que pueden ser utilizados por el analista y otros. Esto con el fin de adquirir los conocimientos necesarios para darle solución de la mejor forma posible al problema planteado.

Los métodos científicos de investigación a utilizar son el teórico, el empírico y los particulares. Dentro de los métodos teóricos se utilizará el histórico, al realizar un estudio del estado del arte sobre todo lo relacionado con el analista de sistemas. El hipotético deductivo es otro método a poner en práctica, pues a partir del problema se plantearon objetivos específicos para darles solución a lo largo de la investigación. Para el análisis y modelación del dominio y sistema es necesario la elaboración de diagramas, figuras y otros artefactos importantes, por lo que se hará uso del método de modelación, pues mediante este se pueden crear abstracciones con el propósito de explicar la realidad. Dentro de los métodos empíricos otro a seguir es el de la observación, pues se va a definir dentro de la Ingeniería de Software la parte que específicamente se tratará durante la investigación. Dentro de los métodos particulares se utilizará la entrevista como técnica fundamental para la obtención de la información referente a los clientes y así realizar la captura de requisitos del software a implementar.

A partir del análisis del sistema se esperan obtener los modelos correspondientes a esta actividad, de forma que permitan un entendimiento tanto para los desarrolladores como para los clientes, en cuanto a las funcionalidades que debe tener el software. Los artefactos generados durante este proceso constituirán dentro del equipo de desarrollo, la entrada para el diseñador del sistema.

La presente tesis está estructurada de la siguiente forma:

En el Capítulo 1 se realiza un estudio sobre algunas de las metodologías para el desarrollo de software más usadas en la actualidad. Se muestran los objetivos del analista de sistemas y de la Ingeniería de Requisitos, técnicas para la captura de requisitos y patrones de casos de usos que son de importancia para lograr una correcta modelación de estos. Así mismo se pueden observar las características principales de algunas herramientas para el modelado.

En el Capítulo 2 se muestra el análisis y la modelación a través de la metodología RUP, derivada de la actividad del analista de sistemas. Primeramente se describen las técnicas usadas en la captura de los requisitos. En segundo lugar se presenta el Modelo del Dominio, donde se refleja de forma conceptual el área de interés. Luego, se pueden observar los requisitos funcionales y no funcionales del sistema. Por último, se muestra el Modelo de Casos de Uso del Sistema, donde se encuentran las especificaciones de los casos de uso y el prototipo no funcional.

En el Capítulo 3 se analizan los resultados obtenidos durante el desarrollo del proceso de análisis. Se expone la calidad del Modelo de Casos de Uso del Sistema mediante el uso de métricas. Además, se expresa mediante una carta de aceptación la conformidad del cliente con la aplicación final, mostrando que la misma cumplió con sus necesidades y expectativas.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

En la actualidad la industria del software está causando un fuerte impacto sobre el mundo tecnológico. La construcción de un producto de software constituye un proceso de gran complejidad, en el que es necesario tener en cuenta una serie de aspectos para lograr que el resultado final posea la calidad requerida. Dentro de este proceso se encuentran todo un conjunto que actividades que deben ser efectuadas. Pressman (1) define un proceso de software como un marco de trabajo de las tareas que se requieren para construir software de alta calidad. Según Jacobson, Booch y Rumbaugh (2), un proceso de desarrollo de software es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de software.

La Ingeniería de Software juega un papel fundamental dentro de este campo. Es una disciplina de las Ciencias de la Computación que ofrece métodos y técnicas para desarrollar y mantener software de calidad que resuelven problemas de todo tipo (1). Se define como la aplicación de un enfoque sistemático, disciplinado y cuantificable hacia el desarrollo, funcionamiento y mantenimiento del software, es decir la aplicación de ingeniería al software (3).

Pressman caracteriza la Ingeniería de Software como una tecnología multicapas, como se puede apreciar en la Figura 1.1.



Figura 1.1: Capas de la Ingeniería de Software (1).

Estas capas pueden ser descritas de la siguiente forma: (1)

- Enfoque de calidad: la gestión de la calidad y las filosofías similares fomentan una cultura continua de mejoras de los procesos, que conduce al desarrollo de enfoques cada vez más robustos para la ingeniería del software.
- Proceso: define un marco de trabajo para un conjunto de áreas claves, que forman la base del control de gestión de proyectos de software y establecen el contexto en el cual: se aplican los métodos técnicos, se producen resultados de trabajo, se establecen hitos, se asegura la calidad y se gestiona el cambio adecuadamente.

- **Métodos:** indican cómo construir técnicamente el software. Abarcan una gran gama de tareas que incluyen análisis de requisitos, diseño, construcción de programas, pruebas y mantenimiento. Dependen de un conjunto de principios básicos que gobiernan cada área de la tecnología e incluyen actividades de modelado y otras técnicas descriptivas.
- **Herramientas:** proporcionan un soporte automático o semiautomático para el proceso y los métodos, estas son las llamadas herramientas CASE (Computer Aided Software Engineering).

Por lo anteriormente expresado, el objetivo de la Ingeniería de Software es lograr productos de software de calidad, mediante un proceso apoyado por métodos y herramientas.

1.1 Metodologías para el Desarrollo de Software

Dada la complejidad del proceso de desarrollo de un software surge la necesidad de que exista un método que rija todas las acciones que deben ser realizadas, con el fin de tener una mejor organización. Por esto han surgido un considerable número de metodologías que constituyen una guía para todo el proceso de creación de un software.

Las metodologías de desarrollo son necesarias para obtener un producto que satisfaga a los clientes, y al mismo tiempo sea capaz de mantener motivados a los desarrolladores. El desarrollo de software es riesgoso y difícil de controlar, por lo que si no se aplica una metodología, ambas partes pueden quedar insatisfechas con el resultado. Muchas veces no se utiliza ninguna metodología, principalmente cuando se trata de proyectos con tiempo de desarrollo demasiado corto, en este caso lo que se hace es descomponer el software en procesos, cada proceso en funciones, y por cada función determinar un tiempo aproximado de desarrollo. Cuando los proyectos que se van a desarrollar son de mediana o mayor envergadura, entonces sí cobra sentido hacer uso de una metodología de desarrollo y seleccionar la más apropiada para lo que se desea lograr. Muchas veces no se encuentra la más adecuada y se termina por adaptar una metodología existente o diseñar una propia, lo cual no es para nada erróneo, siempre y cuando cumpla con el objetivo.

A partir de esto se distinguen dos grandes grupos de metodologías, las ágiles y las pesadas. Dentro del primero se encuentran: XP (Extreme Programming), FDD (Feature Driven Development), DSDM (Dynamic Systems Development Method), AUP (Agile Unified Process), Scrum, Crystal, Adaptive Software Development y otras. Así mismo se puede mencionar RUP (Rational Unified Process) como metodología pesada. (4) (5) (6)

A continuación se abordarán características fundamentales de dos de las metodologías más usadas actualmente; por una parte, XP como principal exponente de las metodologías ágiles, y por el otro, RUP como una de las más generalizadas y usadas entre las metodologías pesadas.

1.1.1 XP (Extreme Programming)

XP es una de las metodologías ágiles para el desarrollo de software más exitosas de la actualidad. Se utiliza en proyectos con pequeños equipos de desarrollo y con corto plazo de entrega. Se basa en la retroalimentación entre el cliente y el equipo de desarrollo, buena comunicación entre todos los participantes y simplicidad en las soluciones implementadas. Consiste en una programación rápida, cuya particularidad es que tiene como miembro del equipo al usuario final. Es una metodología adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. (7)

En XP se sigue la idea de la programación en pares, dado las ventajas que ofrece respecto a la creación del código, pues se pueden evitar errores y malos diseños al controlar cada línea de código y decisión de diseño instantáneamente. La interacción entre ambos desarrolladores puede generar discusiones que lleven a mejores estructuras y algoritmos, aumentando así la calidad del software. (8) (9)

Fases de la metodología XP

El ciclo de vida ideal de XP está compuesto por las siguientes seis fases: (4) (10) (11)

- Exploración: en esta fase los clientes plantean a grandes rasgos las historias de los usuarios que son de interés para la primera entrega del producto. Las historias de los usuarios establecen los requisitos del cliente y son la base para las pruebas de aceptación. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Esta fase toma de pocas semanas a pocos meses, dependiendo del tamaño y familiaridad que tengan los programadores con la tecnología.
- Planificación de la entrega: en esta fase el cliente establece la prioridad de cada historia de usuario, y los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente. Esta fase dura unos pocos días.

- Iteraciones: esta fase incluye varias iteraciones sobre el sistema antes de ser entregado. Todo el trabajo de la iteración es expresado en tareas de programación, cada una de ellas es asignada a un programador como responsable, pero llevadas a cabo por parejas de programadores.
- Producción: esta fase requiere de pruebas adicionales y revisiones de rendimiento antes de que el sistema sea trasladado al entorno del cliente. Al mismo tiempo, se deben tomar decisiones sobre la inclusión de nuevas características a la versión actual, debido a cambios durante esta fase.
- Mantenimiento: mientras la primera versión se encuentra en producción, el proyecto debe mantener el sistema en funcionamiento al mismo tiempo que desarrolla nuevas iteraciones. Para realizar esto se requiere de tareas de soporte para el cliente. De esta forma, la velocidad de desarrollo puede bajar después de la puesta del sistema en producción.
- Muerte del proyecto: ocurre cuando el cliente no tiene más historias para ser incluidas en el sistema. Se genera la documentación final y no se realizan más cambios en la arquitectura. La muerte del proyecto también ocurre cuando el sistema no genera los beneficios esperados por el cliente o cuando no hay presupuesto para mantenerlo.

Roles definidos dentro de la metodología XP

Como parte del proceso de desarrollo siguiendo esta metodología son identificados 7 roles, cada uno con sus funciones específicas a cumplir: (4) (11) (12)

- Programador: escribe las pruebas y produce el código del sistema.
- Cliente: escribe las historias de usuario y las pruebas funcionales para validar su implementación. Asigna la prioridad a los requerimientos y define el orden en que estos serán implementados. Decide cuándo cada requisito está cumplido.
- Encargado de pruebas: ayuda al cliente a escribir las pruebas funcionales. Ejecuta las mismas, difunde los resultados en el equipo y es responsable de las herramientas de soporte para pruebas.

- Encargado de seguimiento: proporciona retroalimentación al equipo. Realiza el seguimiento del progreso de cada iteración y evalúa si los objetivos pueden ser alcanzados con las restricciones de tiempo y recursos presentes o si es necesario realizar algún cambio en el proceso.
- Entrenador: responsable del proceso global. Es necesario que conozca a fondo todo el proceso para que pueda guiar a los miembros del equipo de forma que se siga el proceso correctamente.
- Consultor: es un miembro externo del equipo con un conocimiento específico en algún tema necesario para el proyecto. Guía al equipo para resolver un problema específico.
- Director: toma las decisiones referentes al proceso, para esto, él se comunica con el equipo del proyecto para determinar la situación actual y distinguir cualquier dificultad que exista.

1.1.2 RUP (Rational Unified Process)

RUP es una metodología para el desarrollo de software que es el resultado de varios años de trabajo y uso práctico, en el que se han unificado varias técnicas partiendo del Lenguaje Unificado de Modelado (UML). Su desarrollo como producto ha seguido un camino desde el Proceso Objectory de 1987, pasando por el Proceso Objectory de Rational de 1997, hasta la versión que se ha estandarizado, el Proceso Unificado de Rational publicado en 1998. Al igual que cualquier notación, el Proceso Unificado actúa como un modelo que puede adaptarse a cualquier tipo de proyecto y empresa, ya sean grandes o pequeñas. Es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas de software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyectos. (2)

RUP en su modelación define como sus principales elementos: (2) (13)

- Trabajadores (“quién”): define habilidades y responsabilidades (rol) de un individuo o grupo de individuos, sistema automatizado o máquina, que trabajan en conjunto como un equipo. Ellos realizan las actividades y son propietarios de elementos.
- Actividades (“cómo”): es una tarea que tiene un propósito claro, es realizada por un trabajador y manipula elementos.

- Artefactos ("qué"): productos tangibles del proyecto que son creados, modificados y usados por los trabajadores al realizar actividades. Pueden ser modelos, elementos dentro del modelo, código fuente y ejecutables.
- Flujo de actividades ("cuándo"): secuencia de actividades realizadas por trabajadores y que produce un resultado de valor observable.

Fases de la metodología RUP

RUP divide el proceso de desarrollo en una serie de ciclos que constituyen la vida de un sistema. El ciclo de vida de RUP se caracteriza por ser: (2) (13) (14)

1. Dirigido por casos de uso: los casos de uso reflejan lo que los usuarios desean, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos. A partir de aquí, los casos de uso guían el proceso de desarrollo, pues los modelos que se obtienen como resultado de los diferentes flujos de trabajo representan la realización de los casos de uso.
2. Centrado en la arquitectura: la arquitectura muestra la visión común del sistema completo, en la que el equipo de proyecto y los usuarios deben estar de acuerdo. Describe los elementos del modelo que son más importantes para su construcción, los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente. Las iteraciones que se realizan en RUP comienzan por los casos de uso relevantes desde el punto de vista de la arquitectura.
3. Iterativo e incremental: cada fase de RUP se desarrolla en iteraciones. Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros. Por ejemplo, una iteración de Elaboración centra su atención en el análisis y diseño, aunque refina los requerimientos y obtiene un producto con un determinado nivel, pero que irá creciendo incrementalmente en cada iteración.

Cada ciclo está compuesto por cuatro fases, y dentro de cada una, el trabajo se puede descomponer en iteraciones. Cada fase finaliza con un hito, como se observa en la Figura 1.2, y cada uno de estos se determina por la disponibilidad de un conjunto de artefactos. Los hitos tienen muchos objetivos, entre ellos se encuentran: la toma de decisiones por parte de los directivos antes de que el trabajo pueda continuar en la siguiente fase, y además, permiten controlar el progreso del trabajo a la dirección del proyecto y a los mismos desarrolladores. (2) (14)

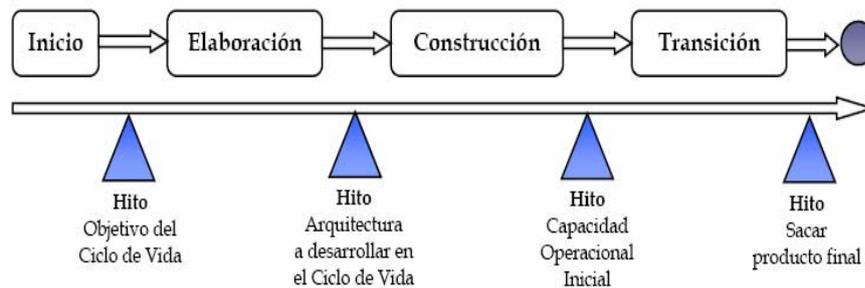


Figura 1.2: Hitos y Fases de RUP (2).

La Figura 1.3 muestra una representación gráfica de los flujos de trabajo y las fases de RUP, mostrando la dinámica expresada en iteraciones y puntos de control. Las fases definidas son: (2) (13) (14) (15)

- Inicio: en esta fase se establece un acuerdo entre todos los interesados acerca de los objetivos del proyecto. Su significación para el software es primaria, pues se asegura de identificar los riesgos relacionados con el negocio y requerimientos. Se describe el negocio y se delimita el proyecto, exponiendo sus alcances con la identificación de los casos de uso del sistema.
- Elaboración: en esta fase se establece la arquitectura base del sistema para proveer bases estables para el diseño e implementación en la siguiente fase, y se obtiene una aplicación ejecutable que responde a los casos de uso que la comprometen. La arquitectura debe abarcar todas las consideraciones de mayor importancia de los requerimientos y una evaluación del riesgo.
- Construcción: el objetivo general de esta fase es obtener un producto listo para su utilización, que esté documentado y tenga un manual de usuario. Se obtienen una o varias liberaciones del producto que han pasado las pruebas y se ponen a consideración de un subconjunto de usuarios. Esta fase es la más compleja de todas pues se necesita de más personal y de más tiempo, por esto requiere de más iteraciones que las fases anteriores.
- Transición: esta fase se centra en implantar el producto en su entorno de operación. La forma en que el proyecto lleva a cabo este objetivo varía con la naturaleza de la relación del producto con su mercado.

Para cada una de las cuatro fases se debe trabajar en 9 disciplinas o flujos de trabajo. Los 6 primeros son conocidos como flujos de ingeniería y los tres últimos como flujos de apoyo. Estos flujos de trabajos son: (13) (15)

- Modelamiento del negocio: describe los procesos del negocio, identificando quiénes participan y las actividades que requieren automatización.
- 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 el sistema será realizado a partir de la funcionalidad prevista y las restricciones impuestas (requerimientos), 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 y la ubicación en ellos de los componentes y la estructura de capas de la aplicación.
- Pruebas: busca los defectos a lo largo del ciclo de vida del producto de software.
- Despliegue: produce liberaciones 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: control de versiones, utilización y actualización concurrente de elementos y otros.
- 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.

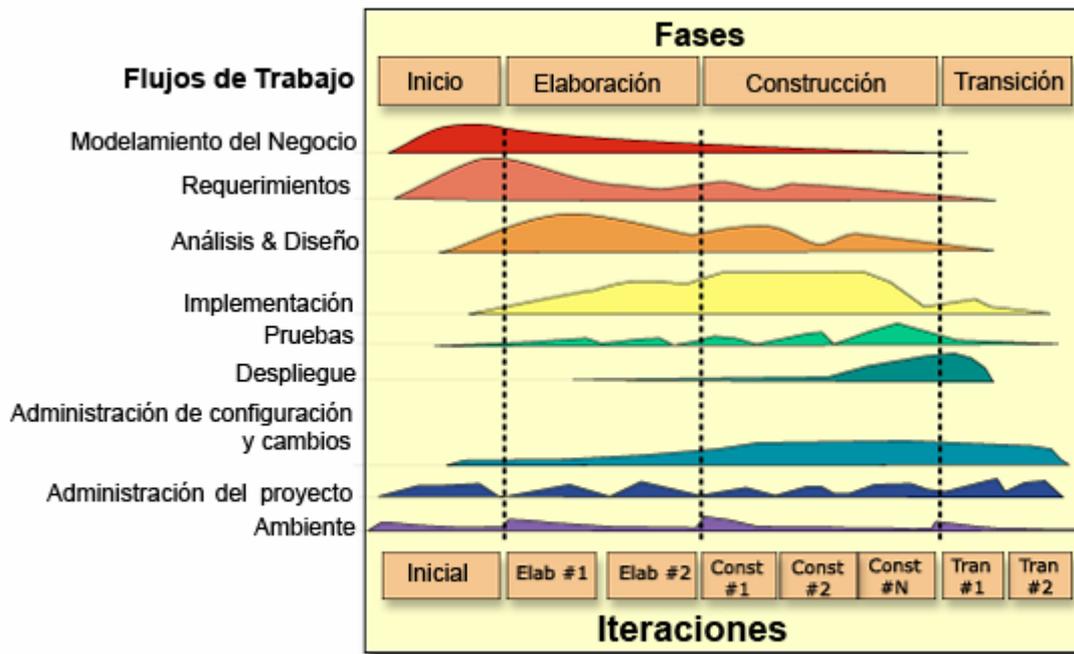


Figura 1.3: RUP en Dos Dimensiones (2).

Roles definidos dentro de la metodología RUP

RUP define grupos de roles, que están agrupados por la participación en actividades relacionadas. Estos grupos son: (15)

- **Analistas:** grupo de roles que están involucrados en la obtención e investigación de los requerimientos del software. Ellos son: analista de procesos de negocio, diseñador del negocio, analista de sistema y especificador de requisitos.
- **Desarrolladores:** grupo de roles que están involucrados en el diseño y la implementación del software. Ellos son: arquitecto de software, diseñador, diseñador de interfaz de usuario, diseñador de paquetes, diseñador de base de datos, implementador e integrador.
- **Gestores:** grupo de roles que están involucrados en la gestión y configuración de los procesos de Ingeniería de Software. Ellos son: jefe de proyecto, jefe de control de cambios, jefe de configuración, jefe de pruebas, jefe de despliegue, ingeniero de procesos, revisor de gestión del proyecto y gestor de pruebas.

- Apoyo: grupo de roles que son necesarios para apoyar el proceso de desarrollo o para producir materiales adicionales requeridos por el producto final. Ellos son: documentador técnico, administrador de sistema, especialista en herramientas, desarrollador de cursos y diseñador gráfico.
- Especialistas en pruebas: grupo de roles que trabajan con habilidades específicas para la realización de las pruebas del software. Ellos son: especialista en pruebas, analista de pruebas y diseñador de pruebas.
- Otros roles: stakeholders¹, revisor, coordinación de revisiones y revisor técnico.

1.1.3 Selección de la metodología de desarrollo de software

Luego de la realización de un estudio de las metodologías más usadas en la actualidad, se puede concluir que RUP es la más indicada para el desarrollo de un sistema de gestión como el que se implementará. Esta metodología constituye uno de los estándares internacionales que más aceptación ha tenido en los últimos años en el desarrollo informático. Además, varias herramientas CASE soportan dicha metodología, permitiendo el trabajo en equipo y con la capacidad de generar código en distintos lenguajes de programación a partir de un diseño UML.

A parte de esto, RUP es una metodología que se encarga de: (13)

- Asegurar la producción de un software de alta calidad que reúna las necesidades de los usuarios finales dentro de un plan y un presupuesto predecible.
- Proveer un enfoque disciplinado para asignar tareas y responsabilidades dentro del desarrollo del sistema.
- Proveer un camino metódico, sistemático para desarrollar, diseñar y validar una arquitectura.
- Reducir en gran medida el riesgo que representa la construcción de sistemas complejos, porque evoluciona de forma incremental partiendo de sistemas más pequeños en los que ya se tiene confianza.

¹ Personas u organizaciones que están activamente implicadas en el negocio, ya sea porque participan en él o porque sus intereses se ven afectados.

1.2 Analista de Sistemas

El analista de sistemas surge producto de la necesidad de analizar, identificar y separar en procesos toda la información referente al software que se desea construir o mejorar. Es el encargado de proponer soluciones y seleccionar la idea más idónea para el problema en cuestión. Este rol se encarga de investigar lo referente al sistema que se quiere desarrollar, y además de definir los diferentes artefactos que intervienen en el proceso de desarrollo y la estrategia de captura de requisitos.

Su labor comienza desde el inicio de la construcción del software, y tiene como objetivo: identificar las necesidades del cliente y evaluar los conceptos que tiene el negocio mediante la entrevista, para realizar la captura de los requisitos y así asignar las funcionalidades al software (2). Es responsable de delimitar el sistema, encontrando los actores y los casos de uso, y asegurando que el modelo de casos de uso sea completo y consistente. Para la consistencia, el analista puede utilizar un glosario para conseguir un acuerdo en los términos comunes, nociones y conceptos durante la captura de los requisitos. (16)

Como la persona que debe establecer la relación con la organización o empresa que solicita el sistema, el analista debe tener una formación integral en informática, y además debe contar con determinadas habilidades personales que se muestran en la Figura 1.4. (16)

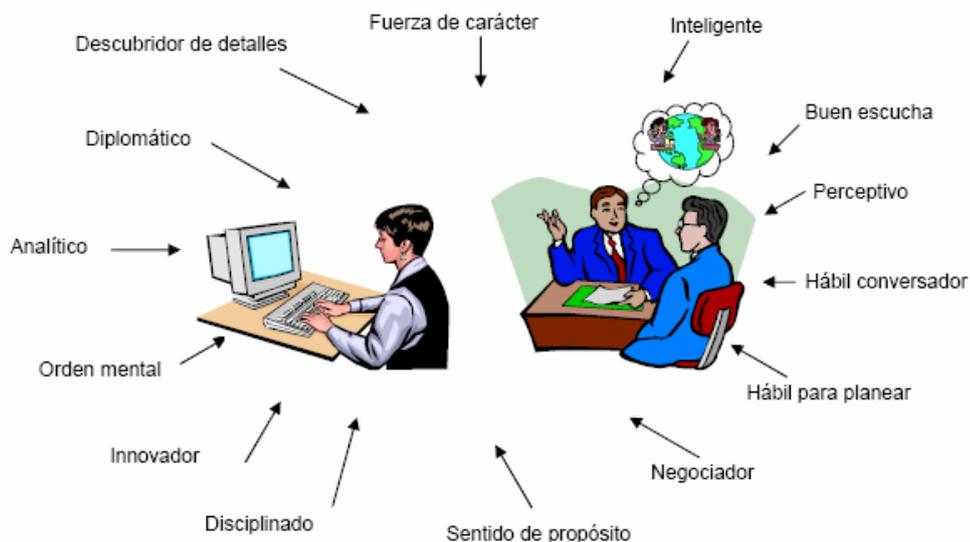


Figura 1.4: Características de un Analista de Sistemas (16).

1.3 Ingeniería de Requisitos

La Ingeniería de Requisitos es una disciplina de la Ingeniería de Software, donde se identifica el propósito del sistema, dirección y alcance del mismo. Es un conjunto de actividades y transformaciones que intentan comprender las necesidades de un sistema de software y convertir la declaración de estas necesidades en una descripción completa, precisa y documentada de los requerimientos del sistema siguiendo un determinado estándar. Facilita el mecanismo apropiado para comprender lo que quiere el cliente, analizando, confirmando su viabilidad, negociando una solución razonable, especificando la solución sin ambigüedad, validando la especificación y gestionando los requisitos para que se transformen en un sistema operacional. Se encuentra vigente durante todo el ciclo de desarrollo, al agregar un nuevo requerimiento en cualquiera de las fases. Una mala captura de requisitos al inicio del proyecto es la razón fundamental por la que una parte de los proyectos de desarrollo de software fracasan o terminan con un tiempo y un presupuesto mayor que el estimado. (1) (17)

Los principales beneficios que se obtienen de la Ingeniería de Requisitos son: (18)

- Permite gestionar las necesidades del proyecto en forma estructurada, debido a que cada actividad de consta de una serie de pasos organizados y bien definidos.
- Mejora la capacidad de predecir cronogramas de proyectos, así como sus resultados.
- Disminuye los costos y retrasos del proyecto, pues reparar errores por un mal desarrollo no descubierto a tiempo, es sumamente caro, especialmente aquellas decisiones tomadas durante la especificación de los requisitos.
- Mejora la calidad del software, al encargarse de velar por el cumplimiento de un conjunto de requerimientos como funcionalidad, facilidad de uso, confiabilidad, desempeño y la correspondencia entre la aplicación y los requisitos del sistema.
- Mejora la comunicación entre equipos, pues la especificación de requerimientos representa una forma de consenso entre clientes y desarrolladores.
- Evita rechazos de usuarios finales, al obligar al cliente a considerar sus requerimientos cuidadosamente y revisarlos dentro del marco del problema.

El proceso de Ingeniería de Requisitos puede ser descrito en seis pasos: (1)

1. Identificación de requisitos

La identificación de los requisitos es una tarea indispensable durante el comienzo del desarrollo de un software. Para esto se deben realizar preguntas al cliente, los usuarios y a los que están involucrados en los objetivos del producto y sean expertos, investigar cómo los sistemas o productos se ajustan a las necesidades del negocio, y finalmente cómo va a ser utilizado el sistema. Para la obtención de requisitos existen un conjunto de actuaciones, que están descritas en las tareas siguientes:

- Valorar el impacto en el negocio y la viabilidad técnica del sistema propuesto.
- Identificar las personas que ayudarán a especificar requisitos y contrastar su papel en la organización.
- Definir el entorno técnico (arquitectura de computación, sistema operativo, necesidades de telecomunicaciones) en el sistema o producto a desarrollar e integrar.
- Identificar restricciones de dominio (características específicas del entorno de negocio en el dominio de la aplicación) que limiten la funcionalidad y rendimientos del sistema a construir.
- Definir uno o más métodos de obtención de requisitos (entrevistas, grupos de trabajo, etc.).
- Solicitar la participación de muchas personas para que los requisitos se definan desde diferentes puntos de vista.
- Identificar requisitos ambiguos como candidatos para el prototipado, y crear escenarios de uso para ayudar a los clientes/usuarios a identificar mejor los requisitos fundamentales.

2. Análisis y negociación de requisitos

Una vez recopilados los requisitos, son agrupados por categorías y organizados en subconjuntos, se estudia cada uno en relación con el resto y se clasifican en base a las necesidades del cliente. Es usual en usuarios y clientes solicitar más de lo que puede realizarse y proponer requisitos contradictorios, argumentando que su versión es esencial por necesidades especiales. El ingeniero del sistema debe resolver estos conflictos a través de un proceso de negociación. Los clientes, usuarios y el resto del personal que interviene deberán clasificar sus requisitos y discutir los posibles conflictos según su prioridad. Se efectúan estimaciones del esfuerzo de desarrollo que se

usan para valorar el impacto de cada requisito en el costo del proyecto y en el plazo de entrega. Utilizando un procedimiento iterativo se irán eliminando, combinando y/o modificando requisitos para conseguir satisfacer los objetivos planteados.

3. Especificación de requisitos

La especificación del sistema es el producto final sobre los requisitos del sistema obtenidos por el ingeniero. Sirve como fundamento para la ingeniería del hardware, ingeniería del software, la ingeniería de bases de datos y la ingeniería humana. Describe la función y características de un sistema de computación y las restricciones que rigen su desarrollo. La especificación delimita cada elemento del sistema y describe la información (datos y control) que entra y sale de este.

4. Modelado del sistema

Todos los sistemas basados en computadoras pueden modelarse como una transformación de la información empleando una arquitectura del tipo entrada – proceso – salida. Como casi todas las técnicas de modelado usadas en la ingeniería del software y de sistemas, el esquema del modelado del sistema permite al analista crear una jerarquía en detalle.

5. Validación de requisitos

La validación de requisitos examina las especificaciones para asegurar que todos los requisitos del sistema han sido establecidos sin ambigüedades, inconsistencias u omisiones, que los errores detectados hayan sido corregidos y que el resultado del trabajo se ajusta a los estándares establecidos para el proceso, el proyecto y el producto. El primer mecanismo para la validación de requisitos son las Revisiones Técnicas Formales.

6. Gestión de requisitos

La gestión de requisitos es un conjunto de actividades que ayudan al equipo de trabajo a identificar, controlar y seguir los requisitos y sus cambios en cualquier momento. Comienza con la identificación de los requisitos para luego darles seguimiento con un grupo de matrices, entre ellas están las matrices de seguimiento de: características, orígenes, dependencias e interfaces. Cada una de estas identifica los requisitos relacionados con uno o más aspectos del sistema o su entorno.

1.4 Técnicas para la Captura de Requisitos

La identificación de los requisitos que debe cumplir un software es una actividad que se lleva a cabo desde el inicio del desarrollo del sistema. En este proceso los analistas extraen de diferentes fuentes de información los datos que son necesarios para conocer las funcionalidades que implementará el sistema. Uno de los riesgos a tener en cuenta es el desconocimiento parcial o total de los analistas sobre el entorno de trabajo de la organización para la que se va a desarrollar el software, bajo este principio, es de vital importancia seleccionar personas que conozcan el negocio de la entidad. Por esto han surgido técnicas que permiten realizar el proceso de captura de requisitos de una forma más eficiente y segura.

Algunas de estas técnicas son: entrevistas, cuestionarios, listas de verificación (Checklist), grabaciones de video y de audio, tormenta de ideas (Brainstorming), JAD (Joint Application Development o Desarrollo Conjunto de Aplicaciones), arqueología de documentos, observación, prototipos y talleres de trabajo basados en los casos de uso. A continuación se describen algunas de ellas. (16) (18) (19)

1.4.1 Entrevistas

Es una de las técnicas más usadas en la captura de requisitos. Consiste en establecer una conversación entre personas de ambas partes. Las entrevistas son dirigidas normalmente por el personal más experto del equipo, quienes junto al equipo de profesionales de otras áreas, como la psicología y el derecho, son los encargados de orientar las entrevistas de forma que la información obtenida a través de ellas sea relevante al proceso. Al analizar las características del sistema con el personal seleccionado cuidadosamente por sus conocimientos sobre este, los analistas pueden conocer los datos que no están disponibles en ninguna otra forma. En esta técnica se pueden identificar tres fases: la preparación, la realización y el análisis de la información obtenida.

1.4.2 Cuestionarios

Consiste en redactar un documento con preguntas, cuyas respuestas sean cortas y concretas, o incluso cerradas por algunas opciones en el propio cuestionario. Este se le realiza a un grupo de personas para que le den respuesta. Esta técnica requiere que el analista conozca el ámbito del problema en el que está trabajando. La forma de la pregunta puede influir en las respuestas, por lo que deben ser planeadas cuidadosamente.

Existen dos categorías: abiertos y cerrados. Los cuestionarios abiertos permiten que los encuestados respondan con su propia terminología. Generalmente estos son más reveladores, pues los interrogados no están limitados en sus respuestas. Por otra parte, los cuestionarios cerrados predeterminan todas las posibles respuestas y el interrogado elige entre las opciones presentadas. Estos cuestionarios se pueden utilizar por ejemplo, cuando se está estableciendo con el cliente el criterio de priorización de los casos de uso. También se puede utilizar cuando se tiene que negociar algún requerimiento con el cliente.

1.4.3 Tormenta de ideas

El propósito de esta técnica es la generación de ideas en un ambiente libre de críticas o juicios. La intención de su aplicación es la de generar la mayor cantidad posible de requerimientos para el sistema. Esta técnica tiene la ventaja de que es muy fácil de aprender y requiere poca organización. Por otro lado, al ser un proceso poco estructurado, puede no producir resultados con la misma calidad o nivel de detalle que otras técnicas.

Las reglas básicas a seguir son:

- Los participantes deben pertenecer a distintas disciplinas y, preferentemente deben tener mucha experiencia. Esto trae aparejado la obtención de una cantidad mayor de ideas creativas.
- Conviene suspender el juicio crítico y se debe permitir la evolución de cada una de las ideas, para que no se cree un ambiente hostil que no aliente la generación de ideas.
- No se deben descartar ideas, por más locas que parezcan, 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 pueden relacionar varias ideas para generar una nueva.

1.4.4 JAD (Joint Application Development)

La técnica para la captura de requisitos JAD (Joint Application Development o Desarrollo Conjunto de Aplicaciones), es una alternativa a las entrevistas. Se desarrolla a lo largo de un conjunto de reuniones en grupo durante un período de 2 a 4 días. En estas reuniones se ayuda a los clientes y usuarios a formular problemas y explorar posibles soluciones, involucrándolos y haciéndolos sentirse partícipes del desarrollo.

Está basada en cuatro principios fundamentales: dinámica de grupo, el uso de ayudas visuales para mejorar la comunicación (diagramas, multimedia, herramientas CASE), mantener un proceso organizado y racional y una filosofía de documentación WYSIWYG (What You See Is What You Get, lo que ve es lo que obtiene), es decir, durante la entrevista se trabajará sobre lo que se generará.

En comparación con las entrevistas, JAD presenta varias ventajas como: el ahorro de tiempo al evitar que las opiniones de los clientes se contrasten por separado, todo el grupo, incluyendo los clientes y los futuros usuarios, revisa la documentación generada y además implica más a los clientes y usuarios en el desarrollo. Dentro de esta técnica se distinguen tres fases: adaptación, celebración de las sesiones JAD y conclusión.

1.5 Patrones de Casos de Uso

Un patrón es una pareja de problema / solución con un nombre y que es aplicable a otros contextos, que estandariza buenos principios y da sugerencias sobre la manera de usarlo en situaciones nuevas (20). Describe un problema de diseño y una solución general para este, en un contexto particular. En muchas disciplinas ha sido reconocida la importancia de la utilización de patrones en la creación de sistemas complejos, pues cada patrón describe un problema que ocurre una y otra vez, y da una solución a este, de manera que dicha solución pueda ser usada un gran número de veces sin hacerlo necesariamente dos veces del mismo modo. (21)

Existen diversos tipos de patrones, entre ellos se encuentran los de arquitectura, de diseño (creacionales, estructurales y de comportamiento) y los de casos de uso. Un patrón de caso de uso es un diseño probado en un modelo de casos de uso, junto con una descripción del contexto en el cual será usado y las consecuencias que tendrá su aplicación en el modelo. A continuación se expone una descripción de algunos de ellos. (22)

1.5.1 Reglas de negocio

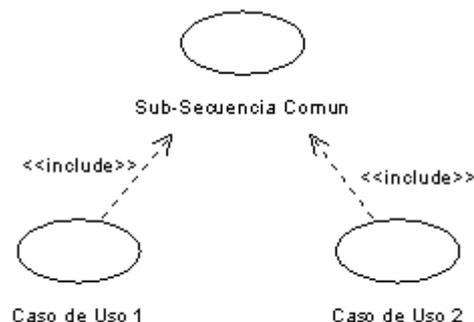
Se basan en la extracción de información originada de las políticas, reglas y regulaciones del negocio de la descripción del flujo y describe la información como una colección de reglas del negocio referenciadas a partir de las descripciones de los casos de uso.

- **Definición estática:** este patrón se aplica a todos los casos de uso que modelan los servicios que son afectados por las reglas de negocio definidas en la organización. Sin embargo, no tiene influencia en la estructura del modelo de casos de uso. Las reglas son descritas en un documento separado, referenciadas por las descripciones de los casos de usos relevantes. Este patrón es apropiado utilizarlo cuando no hay necesidad de cambiar dinámicamente las reglas del negocio mientras el sistema se esté utilizando.
- **Modificación dinámica:** este modelo del patrón contiene un caso de uso llamado Gestionar Regla, que se encarga de crear, actualizar y eliminar las reglas del negocio (Ver 1.5.4). Este patrón es útil cuando la colección de reglas deba ser modificada dinámicamente, o sea, estas pueden ser modificadas mientras el sistema este corriendo.

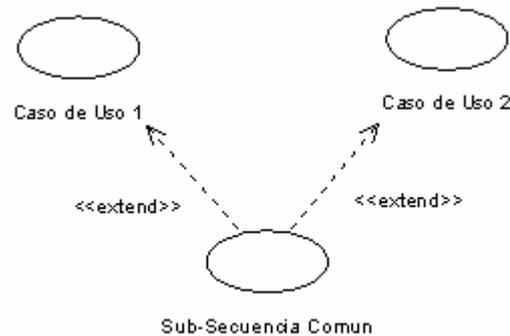
1.5.2 Concordancia

Se basa en extraer una sub-secuencia de acciones que aparecen en diferentes lugares del flujo de casos de uso y es expresada por separado.

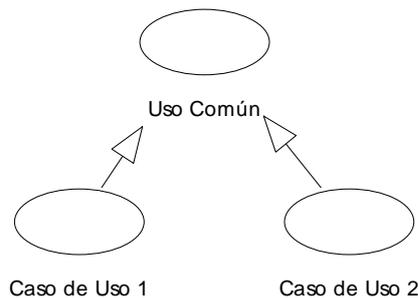
- **Reutilización:** consta de 3 casos de uso. El primero llamado sub-secuencia común, que modela una secuencia de acciones que aparecerán en múltiples casos de uso del modelo; los demás modelan el uso del sistema que comparte la sub-secuencia común de acciones.



- **Adición:** en este caso la sub-secuencia común de casos de uso, extiende los casos de uso compartiendo la sub-secuencia de acciones. Los demás casos de uso modelan el flujo que será expandido con la sub-secuencia. Este patrón es preferible usarlo cuando otros casos de uso se encuentran propiamente completos, o sea, que no requieren de una sub-secuencia común de acciones para modelar los usos completos del sistema.



- **Especialización:** es donde los casos de uso son modelados como una especialización de un tipo común de caso de uso. Todas las acciones en el caso de uso de tipo común son heredadas por los específicos, donde otras acciones pueden ser agregadas o las acciones heredadas pueden ser 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.

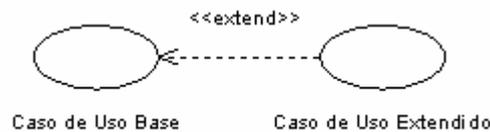


- **Reutilización interna:** si la sub-secuencia de acciones es utilizada en diferentes lugares en un solo caso de uso, no existe la necesidad de extraer la sub-secuencia dentro de un caso de uso separado. En cambio, este debe ser descrito en una sub-sección separada en la descripción del caso de uso. Esta sub-sección será referenciada desde diferentes partes en la descripción del caso de uso donde las sub-secuencias de acciones sean realizadas.

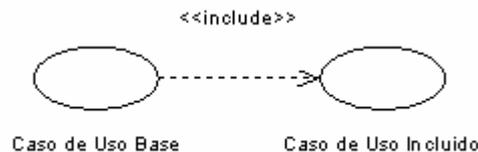
1.5.3 Extensión concreta o inclusión

Se basa en modelar ambos flujos de trabajo como parte de un caso de uso y como separado, completando el caso de uso por sí solo.

- **Extensión:** consiste en la existencia de una relación de extensión entre dos casos de uso. El caso de uso extendido puede ser o no instanciado por el caso de uso base. El caso de uso base puede ser concreto o abstracto. Este patrón se utiliza cuando un flujo puede extender el flujo de otro caso de uso o bien puede ejecutarse dentro de este.



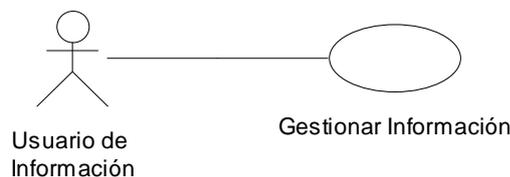
- **Inclusión:** existe una relación de inclusión del caso de uso base con el caso de uso incluido. Este último puede ser instanciado como el mismo. El caso de uso base puede ser concreto o abstracto. Este patrón se utiliza cuando un flujo puede ser incluido en el flujo de un caso de uso y también puede ejecutarse dentro de este.



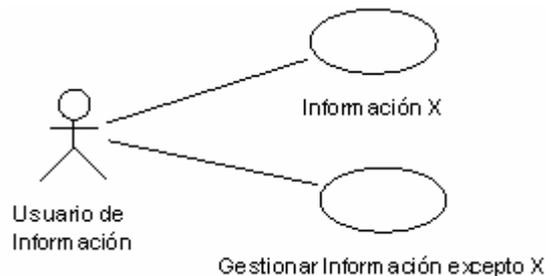
1.5.4 CRUD (Creating, Reading, Updating, Deleting)

Se basa en la fusión de casos de uso simples para formar una unidad conceptual.

- **Completo:** consta de un caso de uso llamado Gestionar Información, que modela todas las operaciones que pueden ser realizadas sobre una parte de la información de un tipo específico, tales como creación, lectura, actualización y eliminación. Suele ser utilizado cuando todos los flujos contribuyen al mismo valor del negocio, y estos a su vez son cortos y simples.



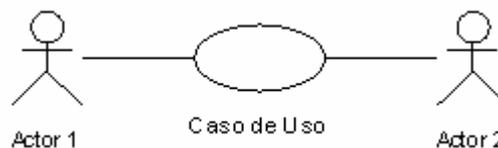
- Parcial: modela una de las vías de los casos de uso como un caso de uso separado. Es preferiblemente utilizado cuando una de las alternativas de los casos de uso es más significativa, larga o más compleja que las otras.



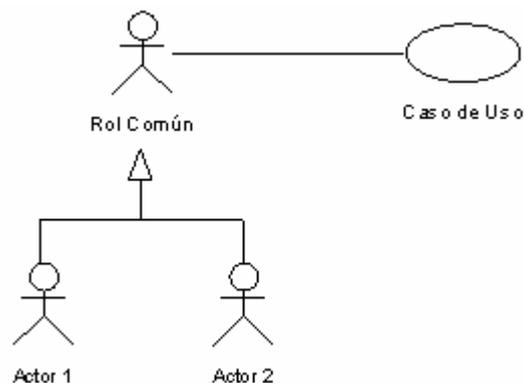
1.5.5 Múltiples actores

Captura puntos comunes entre actores manteniendo separadas otras funciones.

- 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.



- Roles comunes: puede suceder que los dos actores jueguen el mismo rol sobre el caso de uso. Este rol es representado por otro actor, heredado por los actores que comparten este rol. Es aplicable cuando, desde el punto de vista del caso de uso, solo exista una entidad externa interactuando con cada una de las instancias del caso de uso.



1.6 Herramientas para el Modelado

En la medida que evoluciona el mundo de la informática y las comunicaciones, y dentro de este, la rama del desarrollo de sistemas informáticos, se hace de gran interés la obtención de productos de mayor calidad, con la mayor eficiencia posible y los menores costos. Se trata de usar, la metodología correcta para el desarrollo de un software en equipo, con el objetivo de evitar el fracaso de los proyectos o el retraso de los mismos. Uno de los factores determinantes en el éxito de un proyecto software lo constituye una adecuada selección del entorno de trabajo, que incluye las herramientas que permitirán el eficiente desarrollo del mismo. Por esta razón la selección de las herramientas a utilizar se ha convertido en una tarea primordial y de vital importancia.

Las herramientas CASE son aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software, reduciendo el costo del mismo en términos de tiempo y presupuesto. Pueden ayudar en todos los aspectos del ciclo de vida de desarrollo del software, en tareas como el proceso de realizar un diseño del proyecto, cálculo de costos, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación, detección de errores y otros. (23)

Entre los principales objetivos de las herramientas CASE se encuentran: (24)

- Aumentar la calidad del software.
- Mejorar la productividad en el desarrollo y mantenimiento del software.
- Mejorar el tiempo, costo de desarrollo y mantenimiento de los sistemas informáticos.
- Mejorar la planificación de un proyecto.
- Automatizar el desarrollo del software, documentación, generación de código, pruebas de errores y gestión del proyecto.
- Ayuda a la reutilización del software, portabilidad y estandarización de la documentación.
- Gestión global en todas las fases del desarrollo de software con una misma herramienta.
- Facilitar el uso de las distintas metodologías propias de la ingeniería del software.

Las herramientas CASE de modelado con UML permiten aplicar una metodología de análisis y diseño orientados a objetos y hacer una abstracción del código fuente, en un nivel donde la arquitectura y el diseño se tornan más obvios y fáciles de entender y modificar. Cuanto más grande es un proyecto, es más aconsejable utilizar este tipo de herramientas. La herramienta CASE utilizada por el proyecto, debe ser capaz de ayudar al equipo en la mayor parte de las etapas, y posibilitar que las salidas o resultados obtenidos en cada etapa, complementen o sean la entrada de la etapa siguiente. Mediante el uso de las herramientas CASE, los analistas de sistemas pueden capturar los requisitos del sistema con un modelo de casos de uso. De igual forma, los diseñadores y arquitectos pueden producir el modelo de diseño y los desarrolladores pueden transformar los modelos en una aplicación funcional.

1.6.1 Rational Rose

La herramienta CASE Rational Rose, es actualmente una de las más potentes que existe en el mercado para el modelado. Esto se sustenta en el nivel de integración que tiene esta con el resto de las herramientas que la acompañan en la Rational Suite, entre las que se pueden citar el Rational Clear CASE para el control de versiones, el Rational Clear Quest para el control de cambios y el Rational Requisite Pro para la gestión de los requisitos que tendrá el futuro sistema. Otro aspecto a destacar es la posibilidad de generar y realizar ingeniería inversa en varios lenguajes de programación y el número de frameworks que tienen predefinidos, entre los cuales están J2EE, .Net, Visual Basic 6 y C++. (25)

Rational Rose Enterprise Edition propone la utilización de cuatro tipos de modelo para realizar un diseño del sistema, utilizando vistas estática, dinámica, lógica y física de los modelos del sistema. Permite realizar y refinar estas vistas creando de esta forma un modelo completo que representa el dominio del problema y el sistema de software.

Permite que hayan varias personas trabajando a la vez en el proceso iterativo controlado, para ello posibilita que cada desarrollador opere en un espacio de trabajo privado que contiene el modelo completo y tenga un control exclusivo sobre la propagación de los cambios en ese espacio de trabajo. Además, permite visualizar, entender, refinar los requerimientos y la arquitectura antes del enfrentamiento con el código. Esto permite evitar esfuerzos innecesarios en el ciclo de desarrollo en caso de detectarse errores, primero se pueden identificar los riesgos y después se prueba la aplicación para que éstos se hagan mínimos. (16)

1.6.2 Enterprise Architect

La herramienta Enterprise Architect posee las últimas especificaciones UML. Puede servir como herramienta de modelado a todo el equipo, incluyendo analistas, evaluadores, jefes de proyectos, personal del control de calidad, equipo de desarrollo y otros. Incluye características novedosas, con un ambiente rápido, flexible y una interfaz intuitiva para todo el equipo de desarrollo de software. Es una herramienta de diseño y análisis UML comprensible, cubre el desarrollo de software desde los requerimientos, hasta la etapa del análisis, modelos de diseño, pruebas y mantenimiento. Es multiusuario y se encuentra disponible para la plataforma Windows y Linux. Ofrece salida de documentación flexible y de alta calidad. (26)

Ha sido construido sobre las bases de UML 2.1 y combina procesos de negocio, información y flujos de trabajo en un modelo, usando extensiones gratuitas para BPMN (Business Process Modeling Notation) y el perfil Eriksson-Penkerd. Ayuda a disminuir la complejidad con herramientas para rastrear las dependencias, permite soporte para modelos muy grandes, control de versiones con proveedores CVS o SCC y línea base por cada punto del tiempo. Soporta generación e ingeniería inversa de código fuente para muchos lenguajes de programación, incluyendo C++, C#, Java, Delphi, VB.Net, Visual Basic y PHP. También existen componentes gratis para CORBA y Python. Contiene un editor de código fuente con resaltador de sintaxis incorporado. Permite navegar y explorar su modelo de código fuente en el mismo ambiente. Se integra fácilmente con los entornos de desarrollo Eclipse y Visual Studio.Net. (27)

1.6.3 Visual Paradigm

Es una herramienta para el modelado, que posee 6 ediciones: Academia Edition, Community Edition, Personal, Modeler, Standard y Professional Edition. Soporta UML 2.1 completo, BPMN, SysML (Systems Modeling Language), DFD (Data Flow Diagram) y ERD (Entity Relationship Diagram). Permite realizar ingeniería tanto directa como inversa. A partir de un modelo relacional en Sql Server, MySql y otros, es capaz de desplegar todas las clases asociadas a las tablas. Para gestionar la persistencia y el mapeo de estas clases con la base de datos utiliza Hibernate para Java y NHibernate para la plataforma .Net. Está desarrollada con el lenguaje de programación Java, por lo que contiene versiones en los sistemas operativos más conocidos. (28)

En las ediciones comerciales, se brindan un gran conjunto de plug-ins para integrarse con entornos de desarrollo, entre los cuales se pueden citar Eclipse, NetBeans, JBuilder, IntelliJ, WebSphere y Oracle JDeveloper. Visual Paradigm permite realizar mediante un análisis textual técnicas para la captura y análisis de requisitos, identificando casos de uso, actores y clases, como elementos de modelado candidatos. La herramienta es colaborativa, es decir, soporta múltiples usuarios trabajando sobre el mismo proyecto, además permite control de versiones. Genera la documentación del proyecto automáticamente en varios formatos como Web o PDF. (28)

1.6.4 Selección de la herramienta de modelado

Las herramientas CASE de UML, brindan una forma de representar sistemas complejos para facilitar una mejor comprensión a través de su código fuente subyacente y permiten desarrollar la solución de software correcta más rápido y más económicamente. Visual Paradigm for UML 6.0 Enterprise Edition, es la herramienta CASE seleccionada para modelar todos los artefactos que se obtendrán a partir del análisis del negocio y el sistema. Esta herramienta es de gran utilidad para el proyecto, por su nivel de integración con el entorno de desarrollo Eclipse y por la posibilidad de trabajar ambos sobre la plataforma libre GNU/Linux, sistema operativo sobre el cual se desarrolla el proyecto.

1.7 Conclusiones Parciales

En este capítulo se realizó un análisis entre dos de las distintas metodologías de desarrollo, donde se selecciona la utilizada para el desarrollo del proyecto, en este caso RUP. Se brindó información sobre temas de vital importancia a comprender para el trabajo de un analista como son: los objetivos y fases de la Ingeniería de Requisitos, la definición de analista como rol del proceso productivo y las principales técnicas para la captura de requisitos. Además se abordaron patrones que pueden ser utilizados en el modelo de caso de uso y por último se realizó un estudio de las herramientas de modelado más utilizadas en la actualidad, definiéndose la que se utilizará en la realización de los artefactos.

CAPÍTULO 2: SOLUCIÓN PROPUESTA

Como solución al sistema solicitado por el cliente, se realizará una aplicación web. El hecho de contar con dos países involucrados en este proceso, que además necesitan mantener comunicación, hizo de una aplicación de este tipo una opción interesante y candidata segura por encima de una aplicación de escritorio, debido a los beneficios que esta proporciona, entre ellos se encuentran (29):

- Una empresa puede migrar de un sistema operativo a otro, o cambiar el hardware libremente sin afectar el funcionamiento de las aplicaciones del servidor.
- No se requieren complicadas combinaciones de hardware y software para utilizar estas aplicaciones, solo una computadora con un navegador web.
- Facilita el trabajo a distancia, al poder trabajar desde cualquier computadora con conexión a Internet y desde cualquier sitio o país.
- Actualizar o hacer cambios en el software es sencillo y sin riesgos de incompatibilidades.
- Existe solo una versión en el servidor, lo que implica que no hay que distribuirla entre las demás computadoras.
- Al funcionar en un navegador, se requiere un conocimiento básico de informática para utilizar una aplicación web.

La confección de un proyecto de software encierra un conjunto de actividades para lograr el término del mismo, una de estas, es la captura y modelado de los requerimientos por parte del analista de sistemas. En este capítulo se describen las técnicas utilizadas para lograr la captura de requisitos, obteniendo de esta forma las funcionalidades que se automatizarán. Además se muestran los artefactos generados durante el proceso de análisis: el Modelo del Dominio, la Especificación de los Requisitos de Software y el Modelo de Casos de Uso del Sistema, así como los patrones tomados en cuenta en la confección de este último.

2.1 Captura de Requisitos. Técnicas, Métodos y Plantillas a utilizar

La captura de los requisitos del usuario es una de las etapas fundamentales dentro del ciclo de desarrollo de un producto de software, pues a partir de lo que se obtenga en esta es que se realizará el sistema informático. En el proyecto Informatización del Convenio Cuba Venezuela primeramente cada analista se preparó individualmente, con el objetivo de conocer a grandes rasgos el entorno de negocio en el que se enmarca el proyecto a desarrollar, identificar dudas que pudiesen aparecer en este sentido y preparar un grupo de preguntas que pudieran esclarecer algunos temas generales.

Posterior a esto se efectuaron entrevistas con el cliente, como una de las formas más efectivas de obtención de información referente al negocio y de las necesidades específicas de este. Se realizaron con el objetivo de identificar los procesos, los requisitos y objetivos del sistema a construir. Luego de cada una, se realizaron talleres de requisitos tomando como referencia, técnicas como la “Tormenta de ideas”, donde se propiciaron discusiones para lograr esclarecer en el equipo los requisitos del cliente y permitir la confección de los artefactos. Estas actividades se efectuaron con carácter iterativo e incremental, hasta que quedaron bien definidos los requisitos, los casos de uso y sus descripciones, el prototipo no funcional y finalmente estos fueron validados por el cliente.

Para la elaboración de los diferentes artefactos se utilizaron las plantillas definidas por la metodología RUP y adaptadas por la Dirección Central de Software de la UCI. Mediante el uso de estas plantillas se puede lograr una mayor organización del trabajo, quedando constancia de todo lo realizado y facilitando el trabajo al personal del equipo de calidad, permitiendo la realización de las revisiones necesarias para la obtención de un producto de alta calidad.

Las plantillas utilizadas son las siguientes:

- Modelo del Dominio (Anexo 1)
- Especificación de Requisitos (Anexo 2)
- Modelo de Casos de Uso del Sistema (Anexo 3)
- Glosario de Términos (Anexo 4)

2.2 Modelo del Dominio

El Modelo de Dominio o Modelo Conceptual es una representación visual de los conceptos u objetos del mundo real significativos para un problema o área de interés. Se representa en UML con un diagrama de clases, en el que se muestran: conceptos u objetos del dominio del problema (clases conceptuales), atributos y asociaciones entre las clases conceptuales. Es factible su realización cuando los eventos del negocio en cuestión no tienen fronteras completamente definidas y se hace difícil delimitar los procesos del negocio, pero si es posible identificar las personas involucradas o que se benefician en el mismo. El modelo del dominio se considera en la metodología RUP un subconjunto del llamado modelo de objetos del negocio. (2) (20)

2.2.1 Descripción general del dominio

Entre Cuba y Venezuela existe un Convenio de cooperación donde se ejecutan cada año una gran cantidad de proyectos con un importante impacto social, político y económico. Cada cierto tiempo se realizan Comisiones Mixtas en las que se firman acuerdos para desarrollar de forma conjunta estos programas y proyectos de cooperación, que son ejecutados por organismos y entidades de los sectores públicos de ambos países.

Para toda la coordinación, administración y ejecución de los proyectos aprobados en estas Comisiones Mixtas, existe una estructura que está constituida en primer lugar por los directivos del gobierno y los directivos del convenio, que son los responsables de todo lo que acontece que esté relacionado con los proyectos y con cada tema que tenga que ver con la colaboración. En cada uno de los países hay una secretaría técnica, que son las encargadas de coordinar todas las actividades necesarias para el buen desarrollo de cada proyecto. Subordinados a ellas, están los ministerios correspondientes a cada país, que a su vez controlan un grupo de entidades que se desempeñan como entes ejecutores de estos proyectos.

Como parte del Modelo de Dominio, el diagrama de clases conceptuales quedó estructurado como se muestra a continuación:

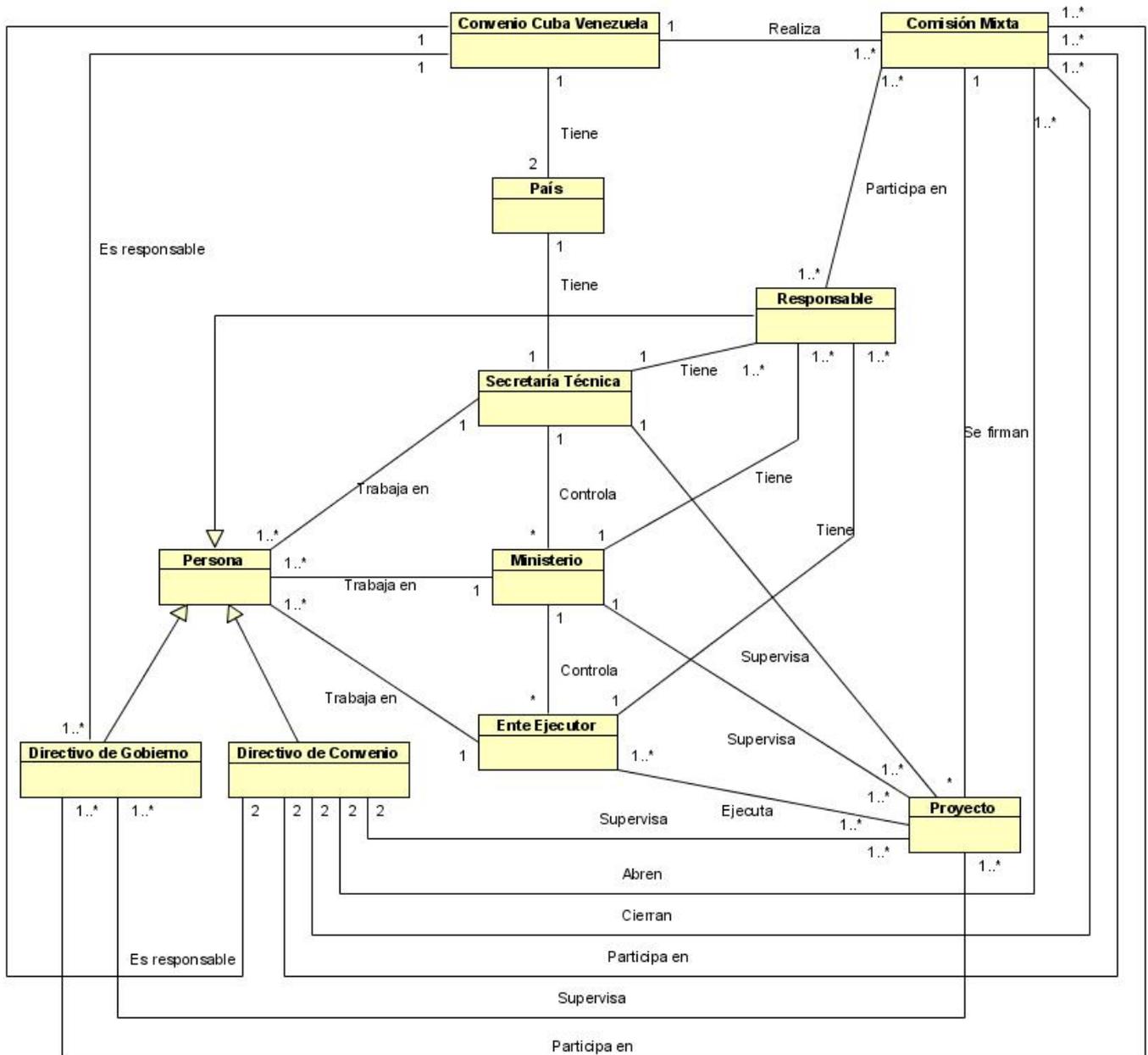


Figura 2.1: Diagrama de Clases Conceptuales del Dominio.

2.2.2 Especificación de las clases conceptuales del modelo del dominio

2.2.2.1 Descripción de la clase: Convenio Cuba Venezuela

Acuerdo de cooperación integral firmado entre la República de Cuba y la República Bolivariana de Venezuela, con el objetivo de elaborar de común acuerdo, programas y proyectos que beneficien a ambos países.

2.2.2.2 Descripción de la clase: Comisión Mixta

Reunión que se realiza cada año con representantes de ambos países, con el fin de servir como mecanismo para el cumplimiento y seguimiento de las acciones de cooperación previstas en el Convenio. Está presidida por la parte cubana por el Ministerio para la Inversión Extranjera y la Colaboración Económica (MINVEC) y por la parte venezolana por el Ministerio del Poder Popular para la Energía y Petróleo (MENPET). Cada una de las partes puede proponer a la otra en cualquier momento, nuevos sectores y proyectos específicos de cooperación para su estudio y aprobación. Así mismo, las partes pueden convocar de común acuerdo y cuando lo consideren necesario, reuniones extraordinarias de la Comisión Mixta.

2.2.2.3 Descripción de la clase: País

Representa las partes involucradas en el Convenio, en este caso Cuba o Venezuela.

2.2.2.4 Descripción de la clase: Secretaría Técnica

Encargadas de coordinar en ambas partes, todas las actividades necesarias para el buen desarrollo de cada proyecto. Llevan un control estadístico de la evolución de cada uno de los proyectos, aceptan las propuestas y cambios en la ejecución de estos.

2.2.2.5 Descripción de la clase: Ministerio

Controlan y aprueban las acciones relacionadas con los proyectos, dirigen las actividades de sus entes ejecutores y por tanto aceptan las propuestas que se presenten.

2.2.2.6 Descripción de la clase: Ente Ejecutor

Es una empresa o institución, que forma parte de los responsables del proyecto. Son los encargados de coordinar y ejecutar cada una de las acciones relacionadas con los proyectos correspondientes a esa entidad.

2.2.2.7 Descripción de la clase: Persona

Individuos que realizan las actividades correspondientes a cada uno de los entes, ministerios y secretarías, que están involucrados en el Convenio.

2.2.2.8 Descripción de la clase: Responsable

Personas que realizan actividades de dirección o se encuentran al frente de cada uno de los entes, ministerios y secretarías, que están involucrados en el Convenio.

2.2.2.9 Descripción de la clase: Directivo de Gobierno

Personas que a nivel de nación son los responsables de todo lo que acontece y está estrechamente relacionado con el Convenio. Estos directivos están al tanto de todos los movimientos llevados a cabo por los proyectos.

2.2.2.10 Descripción de la clase: Directivo de Convenio

Ministros de los ministerios rectores por las partes cubana y venezolana. Tienen todo tipo de facultades dentro de cada tema que tenga que ver con la colaboración, garantizando el control de toda la información relacionada con la Comisión Mixta. Estos directivos están al tanto de todos los movimientos llevados a cabo por los proyectos. Además son los encargados de abrir y cerrar las Comisiones Mixtas.

2.2.2.11 Descripción de la clase: Proyecto

Acuerdo que cumple con un conjunto de actividades que siguen un objetivo específico y son ejecutadas entre dos o más empresas o instituciones. Estos proyectos son firmados en el marco del Convenio durante las Comisiones Mixtas y aportan beneficios tanto sociales como económicos a ambos países.

2.3 Especificación de los Requisitos de Software

Los requisitos generales del producto a construir se obtienen del cliente. Estos requisitos comprenden necesidades de información y control, funcionalidad del producto y comportamiento, rendimiento general del producto, diseño, restricciones de interfaz y otras necesidades especiales (1). Después que se conocen dichos requisitos la misión del analista es otorgarle la funcionalidad a los componentes que conforman el sistema.

2.3.1 Requisitos funcionales

El sistema debe ser capaz de:

- RF01 Autenticar: permitir que cada usuario pueda acceder solo a la parte del sistema que le corresponda según su rol y darle los derechos para realizar las acciones que requiera. Los roles que deben existir en el sistema son Administrador, Directivo de Gobierno, Directivo de Convenio, Secretaría Técnica, Ministerio y Ente Ejecutor.
- RF02 Crear usuario: permitir introducir los datos de un nuevo usuario. En esta acción la persona que se encuentre en el sistema como Administrador, puede crear usuarios de todos los roles; como Secretaría Técnica, puede crear usuarios de ambos países con los roles Ministerio y Ente Ejecutor; y como Ministerio, solo puede crear usuarios con el rol de Ente Ejecutor y que pertenezcan a los entes del ministerio de la persona que está creando el nuevo usuario. En caso que el rol del nuevo usuario sea Ministerio o Ente Ejecutor debe aparecer una lista de entidades para seleccionar una, con los ministerios o entes ejecutores que se encuentran en el sistema respectivamente.
- RF03 Buscar personas: permitir dado el nombre y/o el identificador listar las personas con esas coincidencias. Según el rol del usuario autenticado se deben listar: si es Administrador todas las coincidencias, si es Secretaría Técnica los usuarios pertenecientes a los Ministerios y Entes Ejecutores de ambos países, si es Ministerio los usuarios pertenecientes a sus entes ejecutores y si coincide con el propio usuario este también se mostrará en el listado.

- RF04 Modificar usuario: permitir modificar los datos de un usuario seleccionado. Si el rol del usuario autenticado es Administrador puede modificar todos los usuarios, si es Secretaría Técnica puede modificar los usuarios de los ministerios y entes ejecutores en ambos países y si es Ministerio puede modificar los de sus entes ejecutores. Todos los usuarios del sistema pueden modificar sus propios datos personales.
- RF05 Crear ministerio: permitir introducir los datos de un nuevo ministerio.
- RF06 Buscar ministerios: permitir dado el nombre y/o las siglas listar los ministerios con esas coincidencias.
- RF07 Modificar ministerios: permitir modificar los datos de un ministerio seleccionado.
- RF08 Crear ente ejecutor: permitir introducir los datos de un nuevo ente ejecutor.
- RF09 Buscar ente ejecutor: permitir dado el nombre y/o las siglas listar los entes ejecutores con esas coincidencias.
- RF10 Modificar ente ejecutor: permitir modificar los datos de un ente ejecutor seleccionado.
- RF11 Listar usuarios: permitir obtener un listado con el nombre, apellidos, número de identidad, entidad a la que pertenece y cargo, de los usuarios existentes en el sistema. Si el rol del usuario autenticado es Administrador, Secretaría Técnica, Directivo de Gobierno o Directivo de Convenio se listarán todos los usuarios que existan, si es Ministerio se deben mostrar solamente los usuarios pertenecientes a su ministerio y los de sus entes ejecutores, si es Ente Ejecutor se deben mostrar los usuarios que pertenecen a su ente ejecutor.
- RF12 Listar ministerios: permitir obtener un listado con el nombre, las siglas y el país de todos los ministerios existentes en el sistema.
- RF13 Listar entes ejecutores: permitir obtener un listado con el nombre, las siglas y el país de todos los entes ejecutores existentes en el sistema. Si el rol del usuario autenticado es Administrador, Secretaría Técnica, Directivo de Gobierno o Directivo de Convenio se listarán todos los entes ejecutores que existan; si es Ministerio solo se deben listar los que pertenezcan a su ministerio.

2.3.2 Requisitos no funcionales

Requerimientos no funcionales de usabilidad:

- Permitir uso del teclado para realizar operaciones sobre el sistema.
- Poseer una interfaz agradable para el cliente de acuerdo a los estándares de diseño.
- Mostrar la información de forma lógica y correctamente estructurada.

Requerimientos no funcionales de fiabilidad:

- Permitir autenticación obligatoria y segura.
- Permitir que los datos que circulan por la red no viajen en texto plano.
- Acceder a la información según el rol del usuario.
- Manejar las sesiones de los usuarios, estas deben expirar en 10 minutos.
- Permitir ocultar la información que aparece en la URL.
- No permitir SQL Injection.
- Realizar salvadas periódicamente de la información contenida en la base de datos.
- Permitir que el sistema se recupere ante fallos, ya sea por pérdida de conexión, alimentación u otras causas.

Requerimientos no funcionales de eficiencia:

- Cargar las páginas de la aplicación en un tiempo inferior a 10 segundos.
- Garantizar que con 300 usuarios conectados concurrentemente no disminuya el rendimiento y rapidez de la aplicación.

Requerimientos no funcionales de soporte:

- Utilizar en la implementación del sistema el estándar de codificación definido en el proyecto.

Requerimientos no funcionales de restricciones de diseño:

- Realizar la implementación del sistema mediante el uso de la Programación Orientada a Objetos, específicamente con el lenguaje Java.
- Utilizar diferentes herramientas de software libre para el desarrollo del sistema como: Eclipse para la programación y PostgreSQL como gestor de bases de datos.
- Utilizar el sistema operativo GNU/Linux para la implementación del sistema.

Requerimientos no funcionales de software:

- El sistema operativo sobre el que se va a trabajar debe ser transparente para la aplicación.
- Las computadoras donde se ejecute el sistema requieren tener instalado un navegador Web y un lector de PDF.
- En la computadora donde se encuentre el servidor de bases de datos debe estar instalado el gestor de bases de datos PostgreSQL sobre un sistema operativo Debian GNU/Linux.

Requerimientos no funcionales de hardware:

- Es necesario que se cuente con una red local y con conexión externa.
- Deben existir dos servidores, por razones de seguridad, en uno se ejecutará el sistema y en el otro se encontrará el servidor de base de datos.
- La comunicación entre las máquinas donde se encuentre el sistema y el servidor de base de datos será a través del protocolo de TCP/IP.

2.4 Modelo de Casos de Uso del Sistema

El modelo de casos de uso del sistema sirve como acuerdo entre clientes y desarrolladores para delimitar las funcionalidades con que dispondrá el sistema luego de ser implementado, además proporciona la entrada fundamental para el análisis, el diseño, la implementación y las pruebas. Este modelo está formado por los diagramas de casos de uso y las descripciones de los mismos. El modelado de casos de uso es una técnica efectiva y a la vez simple para modelar los requerimientos del sistema desde la perspectiva del usuario.

2.4.1 Descripción general del sistema

La Aplicación Inicial del Sistema de Gestión para el Seguimiento de los Proyectos del Convenio Integral de Cooperación Cuba – Venezuela, es una herramienta de gestión para el manejo de la información de la actividad colaborativa que se realiza entre las dos naciones, facilitando el seguimiento y control de los proyectos en ejecución, firmados durante las Comisiones Mixtas. Dentro de este sistema, el Módulo Soporte es el encargado de la gestión de los usuarios, ministerios y entes ejecutores, permitiendo que el nivel de acceso a la información esté determinado por el rol desempeñado en el marco del Convenio.

Los roles con que cuenta el sistema son: Administrador, Directivo de Gobierno, Directivo de Convenio, Secretaría Técnica, Ministerio y Ente Ejecutor, ordenados por el nivel de jerarquía que poseen. Para entrar al sistema, primeramente este solicita la autenticación del usuario y luego le da acceso, habilitando las funcionalidades que le corresponden según su rol. Una vez dentro de este, aparecerán deshabilitadas las tareas que no le conciernen. A la hora de crear o modificar usuarios, los administradores no tendrán ningún tipo de limitaciones; los demás roles podrán hacerlo pero con los usuarios que se encuentren por debajo de ellos en la jerarquía, menos los directivos, pues ellos sólo pueden observar los listados que ofrece el sistema. Los usuarios con los roles de administración o secretaria, serán los encargados de insertar o hacer cualquier tipo de modificación de los ministerios o entes ejecutores que deben existir. Todos los usuarios podrán modificar sus datos personales, exceptuando su rol, país, entidad (en caso de pertenecer a alguna) y si es lector o escritor.

Como parte del Modelo de Casos de Uso del Sistema, los actores identificados son los siguientes:

- Usuario: puede autenticarse en el sistema.
- Usuario Autenticado: usuario que se encuentra autenticado en el sistema con alguno de los roles y puede modificar sus datos personales.
- Ente Ejecutor: puede visualizar el listado de los usuarios que pertenecen a su mismo ente ejecutor y modificar sus datos personales.
- Ministerio: puede gestionar usuarios que pertenezcan a sus entes ejecutores, visualizar los listados de sus entes ejecutores y de los usuarios de sus entes y su ministerio, además de modificar sus datos personales.
- Directivo: usuario que se encuentra autenticado en el sistema con el rol de Directivo de Gobierno, Directivo de Convenio, Secretaría Técnica o Administrador; puede visualizar los listados de usuarios, ministerios y entes ejecutores que se ofrecen y además modificar sus datos personales.
- Usuario Administrativo: usuario que se encuentra autenticado en el sistema con el rol de Secretaría Técnica o Administrador y puede gestionar ministerios y entes ejecutores.
- Secretaría Técnica: puede gestionar ministerios, entes ejecutores y usuarios con el rol de Ministerio o Ente Ejecutor. Además, puede visualizar los listados de usuarios, ministerios y entes ejecutores que ofrece el sistema, y modificar sus datos personales.
- Administrador: puede realizar todas las acciones del sistema, o sea, listar o gestionar usuarios, ministerios y entes ejecutores, sin ninguna restricción.

El diagrama de casos de uso del sistema quedó estructurado como se muestra a continuación:

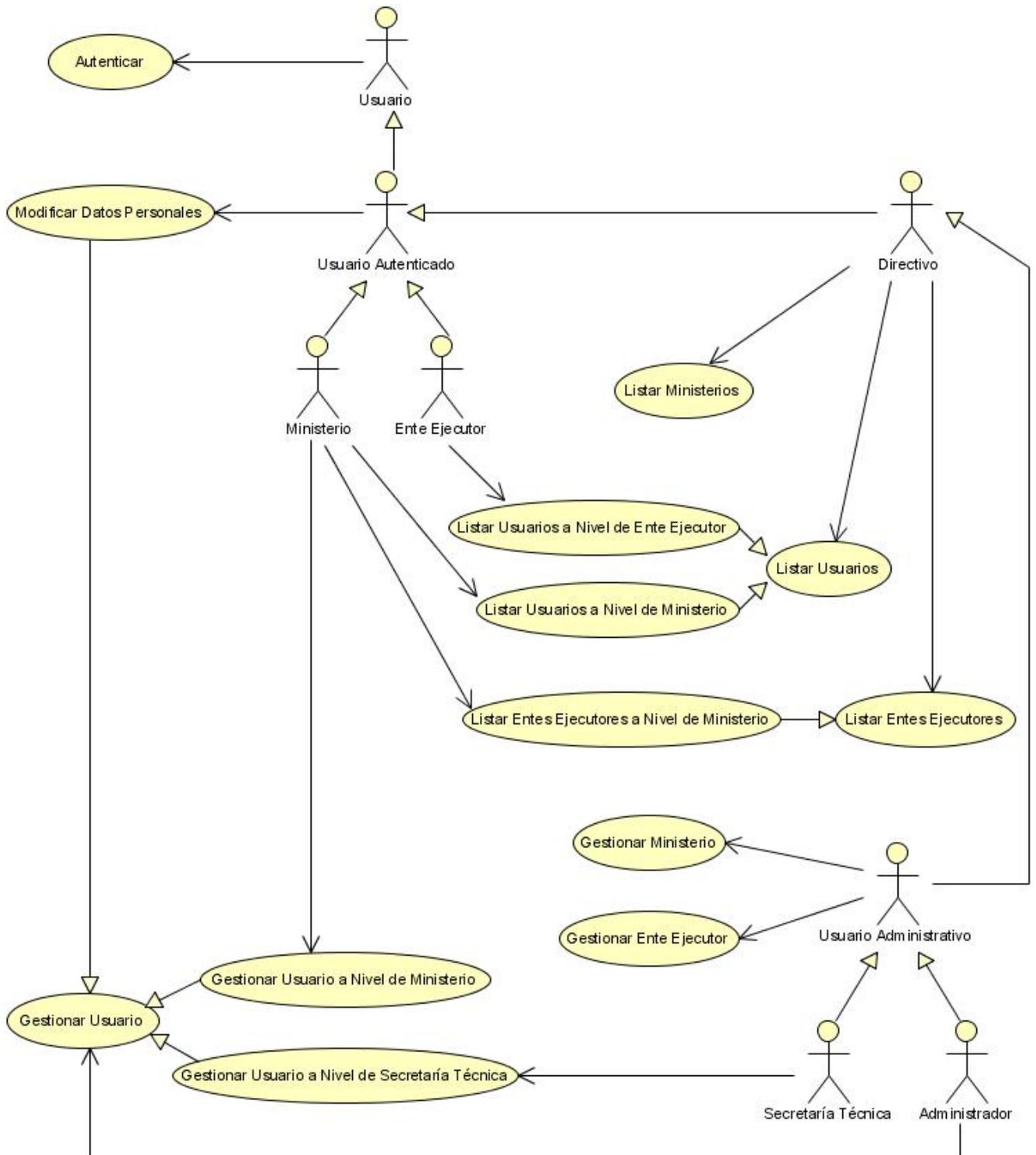


Figura 2.2: Diagrama de Casos de Uso del Sistema.

2.4.2 Aplicación de patrones del análisis

Para desarrollar el modelo casos de uso del sistema se aplicaron patrones de casos de uso, vistos en la sesión 1.5, por la importancia que tienen a la hora de conformar estos. Esto se realizó con el objetivo de reflejar los requisitos funcionales definidos por el cliente y los desarrolladores, de la forma más exacta posible y así lograr mejores resultados en el producto a construir.

Los patrones aplicados son los siguientes:

- Concordancia: especialización

El patrón especialización se aplicó mediante la agrupación de casos de uso de una misma clase que heredan todas las acciones del caso de uso de tipo común. Los casos de uso son: *Modificar Datos Personales*, *Gestionar Usuario a Nivel de Ministerio* y *Gestionar Usuario a Nivel de Secretaría Técnica*, como especializaciones de *Gestionar Usuario*; los casos de uso *Listar Usuarios a Nivel de Ente Ejecutor* y *Listar Usuarios a Nivel de Ministerio*, como especializaciones de *Listar Usuarios*; y el caso de uso *Listar Entes Ejecutores a Nivel de Ministerio*, como especialización de *Listar Entes Ejecutores*.

- CRUD: parcial

Este patrón se aplicó en los casos de uso *Gestionar Usuario*, *Gestionar Ministerio* y *Gestionar Ente Ejecutor*, pues en ellos se agruparon en una sola las funciones básicas de crear y modificar, dejando aparte la funcionalidad de listar en los casos de uso *Listar Usuarios*, *Listar Ministerios* y *Listar Entes Ejecutores*.

- Múltiples actores: rol común

Se aplicó este patrón al existir varios actores del sistema desempeñando el mismo papel hacia un caso de uso y se hizo necesario representarlos con otro actor, donde los primeros heredan las funcionalidades del general. En este caso se encuentran: todos los actores al darle inicio al caso de uso *Autenticar* y a *Modificar Datos Personales*; los actores *Administrador* y *Secretaría Técnica* heredando de *Usuario Administrativo* con los casos de uso *Gestionar Ministerio* y *Gestionar Ente Ejecutor*; y los actores *Administrador*, *Secretaría Técnica* y *Usuario Administrativo* heredando de *Directivo* con los casos de uso *Listar Usuarios*, *Listar Ministerios* y *Listar Entes Ejecutores*.

2.4.3 Especificación de los casos de uso del sistema

2.4.3.1 Descripción del caso de uso: Autenticar

Caso de Uso:	Autenticar
Actores:	Usuario
Resumen:	El caso de uso se inicia cuando se introduce un usuario y contraseña en el sistema para autenticarse, de ser correctos estos datos el sistema le asigna al usuario sus privilegios y en caso contrario muestra un mensaje de error.
Referencias:	RF01
Prioridad:	Crítico

Flujo Normal de Eventos

Acción del Actor	Respuesta del Sistema
1. El usuario entra sus datos al sistema.	1.1 El sistema verifica los datos. 1.2 Si los datos están correctos el sistema le da entrada y le asigna los privilegios según su rol.

Prototipo de Interfaz

El prototipo de interfaz muestra un formulario de autenticación centrado en la pantalla. El formulario tiene un borde gris y contiene los siguientes elementos:

- Un campo de texto etiquetado "Usuario:".
- Un campo de texto etiquetado "Contraseña:".
- Un botón rectangular etiquetado "Aceptar" situado debajo de los campos de texto.

Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	1.2 Si los datos son incorrectos, el sistema muestra un mensaje de error.
Prototipo de Interfaz	
 <p>The screenshot shows a login form with two input fields: 'Usuario:' and 'Contraseña:'. Below the fields, a red error message reads 'Usuario o contraseña incorrecta.' and there is an 'Aceptar' button.</p>	
Poscondiciones:	La autenticación se realiza de forma correcta y se muestran las funcionalidades según el rol del usuario.

Tabla 2.1: Descripción del Caso de Uso Autenticar.

2.4.3.2 Descripción del caso de uso: Gestionar Usuario

Caso de Uso:	Gestionar Usuario
Actores:	Administrador, Secretaría Técnica, Ministerio, Usuario Autenticado
Resumen:	El caso de uso se inicia cuando el usuario selecciona la opción de crear o modificar un usuario, el sistema muestra una interfaz para que entre los datos y si estos están correctos el sistema los guarda o los actualiza.
Precondiciones:	El usuario debe estar autenticado y debe estar registrada la entidad (ministerio o ente ejecutor) a la que va a pertenecer el nuevo usuario o deben existir usuarios en el sistema para el caso de modificar.
Referencias:	RF02, RF03, RF04
Prioridad:	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El usuario selecciona en el menú de la página la opción Administración.	1.1 El sistema muestra una interfaz visual con las opciones: a) Para crear un nuevo ver sección "Crear Usuario". b) Para modificar un usuario existente ver sección "Modificar Usuario".
Flujo Normal de Eventos	
Sección "Crear Usuario"	
Acción del Actor	Respuesta del Sistema
2. El usuario selecciona la opción Crear Usuario.	2.1 El sistema muestra una interfaz donde aparecen los datos que deben ser introducidos. El campo Entidad, debe estar en dependencia del rol y el país seleccionado previamente, de forma tal que: si el rol es Ministerio se listen los todos ministerios pertenecientes a ese país y si es Ente Ejecutor se listen todos los entes ejecutores de dicho país; no deberá aparecer este campo si el rol seleccionado no es ninguno de los mencionados anteriormente. En caso de que se introduzca el Pasaporte, se debe seleccionar el Tipo de Pasaporte y viceversa. El campo Lector/Escritor debe deshabilitarse en caso de que el rol seleccionado sea Directivo de Gobierno, Directivo de Convenio

	<p>o Ente Ejecutor (pues será lector por defecto), y si es Administrador (pues será escritor por defecto).</p> <p>En caso que el usuario autenticado posea el rol:</p> <p>a) Secretaría Técnica, ver Caso de Uso “Gestionar Usuario a Nivel de Secretaría Técnica Secretaría Técnica”, sección “Crear Usuario a Nivel de Secretaría Técnica”.</p> <p>b) Ministerio, ver Caso de Uso “Gestionar Usuario a Nivel de Ministerio”, sección “Crear Usuario a Nivel de Ministerio”.</p>
<p>3. El usuario llena los datos.</p>	<p>3.1 El sistema verifica que los datos sean correctos.</p> <p>3.2 Si los datos son correctos el sistema los almacena y muestra el mensaje “Se insertó correctamente”.</p>

Prototipo de Interfaz

Administración | Proyectos | Reportes Inicio | Salir

CREAR USUARIO

Datos de usuario

Nombre:

Apellidos:

Número de Identidad:

Rol:

País:

Entidad:

Cargo:

Correo Electrónico:

Pasaporte:

Tipo de Pasaporte:

Teléfono:

Fax:

Datos de la persona

Usuario:

Contraseña:

Confirmar:

Lector Escritor

Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	3.2 Si los datos introducidos por el usuario son incorrectos, el sistema muestra un mensaje de error indicando dónde está el dato erróneo y retorna a la acción 3.

Prototipo de Interfaz

Administración | Proyectos | Reportes
Inicio | Salir

CREAR USUARIO

Datos de la Persona

Nombre: No se admiten campos vacíos.

Apellidos: No se admiten campos vacíos.

Número de Identidad: No se admiten campos vacíos.

Rol: Debe seleccionar un elemento.

País: Debe seleccionar un elemento.

Entidad: Debe seleccionar un elemento.

Cargo: No se admiten campos vacíos.

Correo Electrónico: No se admiten campos vacíos.

Pasaporte:

Tipo de Pasaporte:

Teléfono: No se admiten campos vacíos.

Fax:

Datos de Usuario

Usuario: No se admiten campos vacíos.

Contraseña: No se admiten campos vacíos.

Confirmar: No se admiten campos vacíos.

Lector Escritor

CREAR USUARIO

Datos de la Persona

Nombre: No se admiten campos vacíos.

Apellidos: No se admiten campos vacíos.

Número de Identidad: No se admiten campos vacíos.

Rol: Debe seleccionar un elemento.

País: Debe seleccionar un elemento.

Entidad: Debe seleccionar un elemento.

Cargo: No se admiten campos vacíos.

Correo Electrónico: No se admiten campos vacíos.

Pasaporte: Debe especificar el Pasaporte.

Tipo de Pasaporte:

Teléfono: No se admiten campos vacíos.

Fax:

Datos de Usuario

Usuario: Validar No se admiten campos vacíos.

Contraseña: No se admiten campos vacíos.

Confirmar: No se admiten campos vacíos.

Lector Escritor

Aceptar Cancelar

CREAR USUARIO

Datos de la Persona

Nombre: No se admiten campos vacíos.

Apellidos: No se admiten campos vacíos.

Número de Identidad: No se admiten campos vacíos.

Rol: Debe seleccionar un elemento.

País: Debe seleccionar un elemento.

Entidad: Debe seleccionar un elemento.

Cargo: No se admiten campos vacíos.

Correo Electrónico: No se admiten campos vacíos.

Pasaporte:

Tipo de Pasaporte: Debe seleccionar el Tipo de Pasaporte.

Teléfono: No se admiten campos vacíos.

Fax:

Datos de Usuario

Usuario: Validar No se admiten campos vacíos.

Contraseña: No se admiten campos vacíos.

Confirmar: No se admiten campos vacíos.

Lector Escritor

Aceptar Cancelar

Flujos Alternos	
Acción del Actor	Respuesta del Sistema
3. El usuario llena los datos y da en la opción Validar.	3.1 El sistema verifica que el usuario introducido sea correcto. 3.2 Si el usuario es correcto el sistema muestra el mensaje "Usuario Correcto".
Acción del Actor	Respuesta del Sistema
	3.2 Si el usuario ya está en uso, el sistema muestra un mensaje de error y retorna a la acción 3.

Prototipo de Interfaz

Administración | Proyectos | Reportes
Inicio | Salir

MODIFICAR USUARIO

Datos de usuario

Nombre:

Apellidos:

Número de Identidad:

Rol:

País:

Entidad:

Cargo:

Correo Electrónico:

Pasaporte:

Tipo de Pasaporte:

Teléfono:

Fax:

Datos de la persona

Usuario: Usuario existente.

Contraseña:

Confirmar:

Lector Escritor

Flujo Normal de Eventos	
Sección "Modificar Usuario"	
Acción del Actor	Respuesta del Sistema
2. El usuario selecciona la opción Modificar Usuario.	2.1 El sistema muestra una interfaz que permite hacer una búsqueda por nombre o número de identificación.
3. El usuario introduce los datos para la búsqueda.	<p>3.1 El sistema verifica los datos introducidos.</p> <p>3.2 Si los datos están correctos se listan todas las personas con esas coincidencias.</p> <p>a) Si coincide con el propio usuario, ver Caso de Uso "Modificar Datos Personales".</p> <p>En caso que el usuario autenticado posea el rol:</p> <p>b) Secretaría Técnica, ver Caso de Uso "Gestionar Usuario a Nivel de Secretaría Técnica Secretaría Técnica", sección "Modificar Usuario a Nivel de Secretaría Técnica".</p> <p>c) Ministerio, ver Caso de Uso "Gestionar Usuario a Nivel de Ministerio", sección "Modificar Usuario a Nivel de Ministerio".</p>
4. El usuario selecciona una persona.	4.1 El sistema muestra una interfaz con todos los datos de la persona seleccionada. Se cumplen las mismas restricciones que al crear un usuario.
5. El usuario modifica los datos necesarios.	<p>5.1 El sistema verifica los datos introducidos.</p> <p>5.2 Si los datos están correctos el sistema los actualiza y muestra el mensaje "Se modificó correctamente".</p>

Prototipo de Interfaz

Administración | Proyectos | Reportes
Inicio | Salir

BUSCAR PERSONA

Datos de Búsqueda

Nombre:

Número de Identidad:

Lista de Personas

Nombre	Apellidos	Número de Identidad	Entidad	Cargo
Nombre 1	Apellidos 1	XXXXXXXXX 1	Entidad 1	Cargo 1
Nombre 2	Apellidos 2	XXXXXXXXX 2	Entidad 2	Cargo 2
Nombre 3	Apellidos 3	XXXXXXXXX 3	Entidad 3	Cargo 3
Nombre 4	Apellidos 4	XXXXXXXXX 4	Entidad 4	Cargo 4
Nombre 5	Apellidos 5	XXXXXXXXX 5	Entidad 5	Cargo 5
Nombre 6	Apellidos 6	XXXXXXXXX 6	Entidad 6	Cargo 6
Nombre 7	Apellidos 7	XXXXXXXXX 7	Entidad 7	Cargo 7
Nombre 8	Apellidos 8	XXXXXXXXX 8	Entidad 8	Cargo 8
Nombre 9	Apellidos 9	XXXXXXXXX 9	Entidad 9	Cargo 9
Nombre 10	Apellidos 10	XXXXXXXXX 10	Entidad 10	Cargo 10

Anterior [1](#) [2](#) [3](#) Siguiente

Administración | Proyectos | Reportes
Inicio | Salir

MODIFICAR USUARIO

Datos de usuario

Nombre:

Apellidos:

Número de Identidad:

Rol:

País:

Entidad:

Cargo:

Correo Electrónico:

Pasaporte:

Tipo de Pasaporte:

Teléfono:

Fax:

Datos de la persona

Usuario:

Contraseña:

Confirmar:

Lector Escritor

Flujos Alternos

Acción del Actor	Respuesta del Sistema
3. El usuario selecciona la opción Listar Todos.	3.1 El sistema muestra un listado con las personas existentes en el sistema, según el nivel de privilegio que tenga el usuario autenticado, como se especifica en la acción 3.1.

Flujos Alternos

Acción del Actor	Respuesta del Sistema
	3.2 Si el usuario no introduce todos los datos para la búsqueda, el sistema muestra un mensaje de error indicando dónde está el dato faltante y retorna a la acción 3.

Prototipo de Interfaz

Administración | Proyectos | Reportes Inicio | Salir

BUSCAR PERSONA

Datos de Búsqueda

Nombre: Debe al menos especificar un campo de búsqueda.

Número de Identidad: Debe al menos especificar un campo de búsqueda.

Flujos Alternos

Acción del Actor	Respuesta del Sistema
	3.2 Si los datos introducidos por el usuario para la búsqueda son incorrectos, el sistema muestra un mensaje de error indicando dónde está el dato erróneo y retorna a la acción 3.

Prototipo de Interfaz

Administración | Proyectos | Reportes Inicio | Salir

BUSCAR PERSONA

Datos de Búsqueda

Nombre: * Solo se admiten caracteres alfabéticos.

Número de Identidad: * Solo se admiten caracteres alfanuméricos.

Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	5.2 Si los datos introducidos por el usuario son incorrectos, el sistema muestra un mensaje de error indicando dónde está el dato erróneo y retorna a la acción 5.
Prototipo de Interfaz	
Poscondiciones:	Se crea un nuevo usuario. Se modifica un usuario existente.

Tabla 2.2: Descripción del Caso de Uso Gestionar Usuario.

2.4.3.3 Descripción del caso de uso: Gestionar Usuario a Nivel de Secretaría Técnica

Caso de Uso:	Gestionar Usuario a Nivel de Secretaría Técnica
Actores:	Secretaría Técnica
Resumen:	El caso de uso se inicia cuando el usuario selecciona la opción de crear o modificar un usuario, el sistema muestra una interfaz para que entre los datos y si estos están correctos el sistema los guarda o los actualiza.
Precondiciones:	El usuario debe estar autenticado y debe estar registrada la entidad (ministerio o ente ejecutor) a la que va a pertenecer el nuevo usuario o deben existir usuarios en el sistema para el caso de modificar.
Referencias:	RF02, RF03, RF04
Prioridad:	Crítico
Flujo Normal de Eventos	
Sección “Crear Usuario a Nivel de Secretaría Técnica”	
Acción del Actor	Respuesta del Sistema
	2.1 El sistema muestra una interfaz donde aparecen los datos que deben ser introducidos. En este caso los roles que se podrán seleccionar son Ministerio y Ente Ejecutor, y en el campo Entidad se listarán en dependencia del rol seleccionado, todos los ministerios o entes ejecutores que pertenezcan al país seleccionado previamente. En caso de que se introduzca el Pasaporte, se debe seleccionar el Tipo de Pasaporte y viceversa. Si el rol seleccionado es Ente Ejecutor el campo Lector/Escritor aparecerá deshabilitado.
3. El usuario llena los datos.	3.1 El sistema verifica que los datos sean correctos. 3.2 Si los datos son correctos el sistema los almacena y muestra el mensaje “Se insertó correctamente”.

Sección “Modificar Usuario a Nivel de Secretaría Técnica”	
Acción del Actor	Respuesta del Sistema
	3.2 El sistema lista las personas con esas coincidencias pertenecientes a los ministerios y entes ejecutores de ambos países.
4. El usuario selecciona una persona.	4.1 El sistema muestra una interfaz con los datos de la persona seleccionada. Se cumplen las mismas restricciones que al crear un usuario a nivel de secretaría técnica.
5. El usuario modifica los datos necesarios.	5.1 El sistema verifica los datos introducidos. 5.2 Si los datos están correctos el sistema los actualiza y muestra el mensaje “Se modificó correctamente”.
Poscondiciones:	Se crea un nuevo usuario, según el rol de Secretaría Técnica. Se modifica un usuario existente, según el rol de Secretaría Técnica.

Tabla 2.3: Descripción del Caso de Uso Gestionar Usuario a Nivel de Secretaría Técnica.

2.4.3.4 Descripción del caso de uso: Gestionar Usuario a Nivel de Ministerio

Caso de Uso:	Gestionar Usuario a Nivel de Ministerio
Actores:	Ministerio
Resumen:	El caso de uso se inicia cuando el usuario selecciona la opción de crear o modificar un usuario, el sistema muestra una interfaz para que entre los datos y si estos están correctos el sistema los guarda o los actualiza.
Precondiciones:	El usuario debe estar autenticado y debe estar registrada la entidad (ministerio o ente ejecutor) a la que va a pertenecer el nuevo usuario o deben existir usuarios en el sistema para el caso de modificar.
Referencias:	RF02, RF03, RF04
Prioridad:	Crítico
Flujo Normal de Eventos	
Sección "Crear Usuario a Nivel de Ministerio"	
Acción del Actor	Respuesta del Sistema
	2.1 El sistema muestra una interfaz donde aparecen los datos que deben ser introducidos. En este caso solo se podrá seleccionar el rol Ente Ejecutor y en el campo Entidad se listarán todos los entes ejecutores que pertenecen al ministerio del usuario autenticado. El campo Lector/Escritor aparecerá deshabilitado.
3. El usuario llena los datos.	3.1 El sistema verifica que los datos sean correctos. 3.2 Si los datos son correctos el sistema los almacena y muestra el mensaje "Se insertó correctamente".

Sección “Modificar Usuario a Nivel de Ministerio”	
Acción del Actor	Respuesta del Sistema
	3.2 El sistema lista las personas con esas coincidencias que correspondan a los entes ejecutores del ministerio al que pertenece el usuario autenticado.
4. El usuario selecciona una persona.	4.1 El sistema muestra una interfaz con los datos de la persona seleccionada. Se cumplen las mismas restricciones que al crear un usuario a nivel de ministerio.
5. El usuario modifica los datos necesarios.	5.1 El sistema verifica los datos introducidos. 5.2 Si los datos están correctos el sistema los actualiza y muestra el mensaje “Se modificó correctamente”.
Poscondiciones:	Se crea un nuevo usuario, según el rol de Ministerio. Se modifica un usuario existente, según el rol de Ministerio.

Tabla 2.4: Descripción del Caso de Uso Gestionar Usuario a Nivel de Ministerio.

2.4.3.5 Descripción del caso de uso: Modificar Datos Personales

Caso de Uso:	Modificar Datos Personales	
Actores:	Usuario Autenticado	
Resumen:	El caso de uso se inicia cuando el usuario selecciona la opción de modificar usuario, el sistema muestra una interfaz para que entre los datos y si estos están correctos, son actualizados.	
Precondiciones:	El usuario debe estar autenticado.	
Referencias:	RF03, RF04	
Prioridad:	Crítico	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
	3.2 El sistema muestra el usuario que está autenticado.	
4. El usuario selecciona la persona.	4.1 El sistema muestra una interfaz con todos los datos de la persona, donde no pueden ser cambiados los campos Rol, País, Entidad (solo en caso de existir) y Lector/Escritor. Si el rol del usuario autenticado es Administrador, este podrá modificar todos sus datos.	
5. El usuario modifica los datos necesarios.	5.1 El sistema verifica los datos introducidos. 5.2 Si los datos están correctos el sistema los actualiza y muestra el mensaje "Se modificó correctamente".	
Poscondiciones:	Se modifican los datos del usuario autenticado.	

Tabla 2.5: Descripción del Caso de Uso Modificar Datos Personales.

2.4.3.6 Descripción del caso de uso: Gestionar Ministerio

Caso de Uso:	Gestionar Ministerio
Actores:	Usuario Administrativo
Resumen:	El caso de uso se inicia cuando el usuario selecciona la opción de crear o de modificar un ministerio, el sistema muestra una interfaz para que el usuario entre los datos y si estos están correctos el sistema los guarda o los actualiza.
Precondiciones:	El usuario debe estar autenticado y para el caso de modificar debe haber ministerios registrados en el sistema.
Referencias:	RF05, RF06, RF07
Prioridad:	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El usuario selecciona en el menú de la página la opción Administración.	1.1 El sistema muestra una interfaz visual con las opciones: a) Para crear un nuevo ministerio ver sección "Crear Ministerio". b) Para modificar un ministerio existente ver sección "Modificar Ministerio".

Flujo Normal de Eventos	
Sección "Crear Ministerio"	
Acción del Actor	Respuesta del Sistema
2. El usuario selecciona la opción Crear Ministerio.	2.1 El sistema muestra una interfaz donde aparecen los datos que deben ser introducidos. En el campo país se podrá seleccionar Cuba o Venezuela.
3. El usuario llena los datos.	3.1 El sistema verifica que los datos sean correctos. 3.2 Si los datos son correctos el sistema los almacena y muestra el mensaje "Se insertó correctamente".

Prototipo de Interfaz

The screenshot shows a web interface for creating a ministry. At the top, there are navigation links: 'Administración | Proyectos | Reportes' on the left and 'Inicio | Salir' on the right. Below this is a header 'CREAR MINISTERIO' and a sub-header 'Datos del Ministerio'. The form contains the following fields:

- Nombre: A text input field.
- Siglas: A text input field.
- Nombre del Jefe: A text input field.
- Cargo: A text input field.
- Teléfono: A text input field.
- Fax: A text input field.
- País: A dropdown menu with 'Selecciona' as the current selection.

 At the bottom of the form are two buttons: 'Aceptar' and 'Cancelar'.

Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	3.2 Si los datos introducidos por el usuario son incorrectos, el sistema muestra un mensaje de error indicando dónde está el dato erróneo y retorna a la acción 3.

Prototipo de Interfaz

Prototipo de Interfaz

Administración | Proyectos | Reportes
Inicio | Salir

CREAR MINISTERIO

Datos del Ministerio

Nombre: No se admiten campos vacíos.

Siglas: No se admiten campos vacíos.

Nombre del Jefe: No se admiten campos vacíos.

Cargo: No se admiten campos vacíos.

Teléfono: No se admiten campos vacíos.

Fax:

Pais: Debe seleccionar un elemento.

Flujo Normal de Eventos

Sección “Modificar Ministerio”

Acción del Actor	Respuesta del Sistema
2. El usuario selecciona la opción Modificar Ministerio.	2.1 El sistema muestra una interfaz que permite hacer una búsqueda por nombre y siglas.
3. El usuario introduce los datos para la búsqueda.	3.1 El sistema verifica los datos introducidos. 3.2 Si los datos están correctos se listan los ministerios con esas coincidencias.
4. El usuario selecciona un ministerio.	4.1 El sistema muestra una interfaz con todos los datos del ministerio seleccionado.
5. El usuario modifica los datos necesarios.	5.1 El sistema verifica los datos introducidos. 5.2 Si los datos están correctos el sistema los actualiza y muestra el mensaje “Se modificó correctamente”.

Prototipo de Interfaz

Administración | Proyectos | Reportes
Inicio | Salir

BUSCAR MINISTERIO

Datos de Búsqueda

Nombre:

Siglas:

Lista de Ministerios

Nombre	Siglas	País
Nombre del Ministerio 1	Sigla 1	País 1
Nombre del Ministerio 2	Sigla 2	País 1
Nombre del Ministerio 3	Sigla 3	País 1
Nombre del Ministerio 4	Sigla 4	País 1
Nombre del Ministerio 5	Sigla 5	País 1
Nombre del Ministerio 6	Sigla 6	País 2
Nombre del Ministerio 7	Sigla 7	País 2
Nombre del Ministerio 8	Sigla 8	País 2
Nombre del Ministerio 9	Sigla 9	País 2
Nombre del Ministerio 10	Sigla 10	País 2

Anterior [1](#) [2](#) [3](#) Siguiente

Administración | Proyectos | Reportes
Inicio | Salir

MODIFICAR MINISTERIO

Datos del Ministerio

Nombre:

Siglas:

Nombre del Jefe:

Cargo:

Teléfono:

Fax:

País: ▼

Flujos Alternos	
Acción del Actor	Respuesta del Sistema
3. El usuario selecciona la opción Listar Todos.	3.1 El sistema muestra un listado con todos los ministerios existentes.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	3.2 Si el usuario no introduce todos los datos para la búsqueda, el sistema muestra un mensaje de error indicando dónde está el dato faltante y retorna a la acción 3.

Prototipo de Interfaz

The screenshot shows a web interface with a navigation bar at the top containing 'Administración | Proyectos | Reportes' on the left and 'Inicio | Salir' on the right. Below the navigation bar is a section titled 'BUSCAR MINISTERIO'. Under this title, there is a sub-section 'Datos de Búsqueda' which contains two input fields: 'Nombre:' and 'Siglas:'. Both fields are empty and have a red error message next to them that reads 'Debe al menos especificar un campo de búsqueda.'. Below the input fields are three buttons: 'Listar Todos', 'Buscar', and 'Cancelar'.

Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	3.2 Si los datos introducidos por el usuario para la búsqueda son incorrectos, el sistema muestra un mensaje de error indicando dónde está el dato erróneo y retorna a la acción 3.

Prototipo de Interfaz

Administración | Proyectos | Reportes Inicio | Salir

BUSCAR MINISTERIO

Datos de Búsqueda

Nombre: Solo se admiten caracteres alfabéticos.

Siglas: Solo se admiten caracteres alfabéticos.

Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	5.2 Si los datos introducidos por el usuario son incorrectos, el sistema muestra un mensaje de error indicando dónde está el dato erróneo y retorna a la acción 5.

Prototipo de Interfaz	
Poscondiciones:	Se crea un nuevo ministerio. Se modifica un ministerio existente.

Tabla 2.6: Descripción del Caso de Uso Gestionar Ministerio.

2.4.3.7 Descripción del caso de uso: Gestionar Ente Ejecutor

Caso de Uso:	Gestionar Ente Ejecutor	
Actores:	Usuario Administrativo	
Resumen:	El caso de uso se inicia cuando el usuario selecciona la opción de crear o de modificar un ente ejecutor, el sistema muestra una interfaz para que el usuario entre los datos y si estos están correctos el sistema los guarda o los actualiza.	
Precondiciones:	El usuario debe estar autenticado y debe estar previamente creado el ministerio al cual pertenece el nuevo ente ejecutor o debe haber entes ejecutores registrados en el sistema para el caso de modificar.	
Referencias:	RF08, RF09, RF10	
Prioridad:	Crítico	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El usuario selecciona en el menú de la página la opción Administración.	1.1 El sistema muestra una interfaz visual con las opciones: a) Para crear un nuevo ente ejecutor ver sección "Crear Ente Ejecutor". b) Para modificar un ente ejecutor existente ver sección "Modificar Ente Ejecutor".	

Flujo Normal de Eventos	
Sección "Crear Ente Ejecutor"	
Acción del Actor	Respuesta del Sistema
2. El usuario selecciona la opción Crear Ente Ejecutor.	2.1 El sistema muestra una interfaz donde aparecen los datos que deben ser introducidos. En este caso, en el campo correspondiente a los ministerios solo se listarán los que pertenezcan al país (Cuba o Venezuela) que haya sido seleccionado previamente. Si se selecciona un banco, se debe introducir una cuenta bancaria y viceversa.
3. El usuario llena los datos.	3.1 El sistema verifica que los datos sean correctos. 3.2 Si los datos son correctos el sistema los almacena y muestra el mensaje "Se insertó correctamente".

Prototipo de Interfaz

Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	3.2 Si los datos introducidos son incorrectos, el sistema muestra un mensaje indicando dónde está el dato erróneo y retorna a la acción 3.

Prototipo de Interfaz

Administración | Proyectos | Reportes
Inicio | Salir

CREAR ENTE EJECUTOR

Datos del Ente Ejecutor

Nombre:

Siglas:

Nombre del Jefe:

Cargo:

Teléfono:

Fax:

Correo Electrónico:

Banco:

Cuenta Bancaria:

País:

Ministerio:

No se admiten campos vacíos.

Debe seleccionar un elemento.

Debe seleccionar un elemento.

CREAR ENTE EJECUTOR
Datos del Ente Ejecutor

Nombre:	<input type="text"/>	No se admiten campos vacíos.
Siglas:	<input type="text"/>	No se admiten campos vacíos.
Nombre del Jefe:	<input type="text"/>	No se admiten campos vacíos.
Cargo:	<input type="text"/>	No se admiten campos vacíos.
Teléfono:	<input type="text"/>	No se admiten campos vacíos.
Fax:	<input type="text"/>	
Correo Electrónico:	<input type="text"/>	No se admiten campos vacíos.
Banco:	<input type="text" value="Banco X"/>	
Cuenta Bancaria:	<input type="text"/>	Debe especificar la Cuenta Bancaria.
País:	<input type="text" value="Seleccione"/>	Debe seleccionar un elemento.
Ministerio:	<input type="text" value="Seleccione"/>	Debe seleccionar un elemento.

CREAR ENTE EJECUTOR
Datos del Ente Ejecutor

Nombre:	<input type="text"/>	No se admiten campos vacíos.
Siglas:	<input type="text"/>	No se admiten campos vacíos.
Nombre del Jefe:	<input type="text"/>	No se admiten campos vacíos.
Cargo:	<input type="text"/>	No se admiten campos vacíos.
Teléfono:	<input type="text"/>	No se admiten campos vacíos.
Fax:	<input type="text"/>	
Correo Electrónico:	<input type="text"/>	No se admiten campos vacíos.
Banco:	<input type="text" value="No Especificado"/>	Debe seleccionar el Banco.
Cuenta Bancaria:	<input type="text" value="000000000000"/>	
País:	<input type="text" value="Seleccione"/>	Debe seleccionar un elemento.
Ministerio:	<input type="text" value="Seleccione"/>	Debe seleccionar un elemento.

Sección "Modificar Ente Ejecutor"	
Acción del Actor	Respuesta del Sistema
2. El usuario selecciona la opción Modificar Ente Ejecutor.	2.1 El sistema muestra una interfaz que permite realizar una búsqueda por nombre y siglas.
3. El usuario introduce los datos para la búsqueda.	3.1 El sistema verifica los datos introducidos. 3.2 Si los datos están correctos se listan los entes ejecutores con esas coincidencias.
4. El usuario selecciona un ente ejecutor.	4.1 El sistema muestra una interfaz con los datos del ente ejecutor seleccionado. Se cumplen las mismas restricciones de crear un ente ejecutor.
5. El usuario modifica los datos necesarios.	5.1 El sistema verifica los datos introducidos. 5.2 Si los datos están correctos el sistema los actualiza y muestra el mensaje "Se modificó correctamente".

Prototipo de Interfaz

Administración | Proyectos | Reportes
Inicio | Salir

BUSCAR ENTE EJECUTOR

Datos de Búsqueda

Nombre:

Siglas:

Lista de Entes Ejecutores

Nombre	Siglas	País
Nombre del Ente Ejecutor 1	Sigla 1	País 1
Nombre del Ente Ejecutor 2	Sigla 2	País 1
Nombre del Ente Ejecutor 3	Sigla 3	País 1
Nombre del Ente Ejecutor 4	Sigla 4	País 1
Nombre del Ente Ejecutor 5	Sigla 5	País 1
Nombre del Ente Ejecutor 6	Sigla 6	País 2
Nombre del Ente Ejecutor 7	Sigla 7	País 2
Nombre del Ente Ejecutor 8	Sigla 8	País 2
Nombre del Ente Ejecutor 9	Sigla 9	País 2
Nombre del Ente Ejecutor 10	Sigla 10	País 2

Anterior [1](#) [2](#) [3](#) Siguiente

Administración | Proyectos | Reportes Inicio | Salir

MODIFICAR ENTE EJECUTOR

Datos del Ente Ejecutor

Nombre:

Siglas:

Nombre del Jefe:

Cargo:

Teléfono:

Fax:

Correo Electrónico:

Banco:
 ▼

Cuenta Bancaria:

País:
 ▼

Ministerio:
 ▼

Flujos Alternos

Acción del Actor	Respuesta del Sistema
3. El usuario selecciona la opción Listar Todos.	3.1 El sistema muestra un listado con todos los entes ejecutores existentes.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	3.2 Si el usuario no introduce todos los datos para la búsqueda, el sistema muestra un mensaje de error indicando dónde está el dato faltante y retorna a la acción 3.

Prototipo de Interfaz

Flujos Alternos

Acción del Actor	Respuesta del Sistema
	3.2 Si los datos introducidos por el usuario para la búsqueda son incorrectos, el sistema muestra un mensaje de error indicando dónde está el dato erróneo y retorna a la acción 3.

Prototipo de Interfaz

Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	5.2 Si los datos introducidos por el usuario son incorrectos, el sistema muestra un mensaje de error indicando dónde está el dato erróneo y retorna a la acción 5.

Prototipo de Interfaz

Poscondiciones:	Se crea un nuevo ente ejecutor. Se modifica un ente ejecutor existente.
------------------------	--

Tabla 2.7: Descripción del Caso de Uso Gestionar Ente Ejecutor.

2.4.3.8 Descripción del caso de uso: Listar Usuarios

Caso de Uso:	Listar Usuarios
Actores:	Directivo, Ministerio, Ente Ejecutor
Resumen:	El caso de uso se inicia cuando el usuario selecciona la opción de listar usuarios, luego el sistema le muestra el listado de los usuarios existentes.
Precondiciones:	El usuario debe estar autenticado.
Referencias:	RF11
Prioridad:	Auxiliar
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El usuario selecciona en el menú de la página la opción Administración y luego la opción Listar Usuarios.	1.1 El sistema muestra un listado ordenado alfabéticamente de todos los usuarios registrados (en caso de que el rol del usuario sea Directivo de Gobierno, Directivo de Convenio, Secretaría Técnica o Administrador). En caso que el usuario autenticado posea el rol: a) Ministerio, ver Caso de Uso “Listar Usuarios a Nivel de Ministerio”. b) Ente Ejecutor, ver Caso de Uso “Listar Usuarios a Nivel de Ente Ejecutor”.

Prototipo de Interfaz

Administración Proyectos Reportes				Inicio Salir
Lista de Personas ✕				
Nombre	Apellidos	Número de Identidad	Entidad	Cargo
Nombre 1	Apellidos 1	XXXXXXXXX 1	Entidad 1	Cargo 1
Nombre 2	Apellidos 2	XXXXXXXXX 2	Entidad 2	Cargo 2
Nombre 3	Apellidos 3	XXXXXXXXX 3	Entidad 3	Cargo 3
Nombre 4	Apellidos 4	XXXXXXXXX 4	Entidad 4	Cargo 4
Nombre 5	Apellidos 5	XXXXXXXXX 5	Entidad 5	Cargo 5
Nombre 6	Apellidos 6	XXXXXXXXX 6	Entidad 6	Cargo 6
Nombre 7	Apellidos 7	XXXXXXXXX 7	Entidad 7	Cargo 7
Nombre 8	Apellidos 8	XXXXXXXXX 8	Entidad 8	Cargo 8
Nombre 9	Apellidos 9	XXXXXXXXX 9	Entidad 9	Cargo 9
Nombre 10	Apellidos 10	XXXXXXXXX 10	Entidad 10	Cargo 10
Anterior 1 2 3 Siguiete				

Poscondiciones:

Los usuarios registrados en el sistema son listados.

Tabla 2.8: Descripción del Caso de Uso Listar Usuarios.

2.4.3.9 Descripción del caso de uso: Listar Usuarios a Nivel de Ministerio

Caso de Uso:	Listar Usuarios a Nivel de Ministerio
Actores:	Ministerio
Resumen:	El caso de uso se inicia cuando el usuario selecciona la opción de listar usuarios, luego el sistema le muestra el listado de los usuarios existentes.
Precondiciones:	El usuario debe estar autenticado.
Referencias:	RF11
Prioridad:	Auxiliar
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
	1.1 El sistema muestra un listado ordenado alfabéticamente con los usuarios registrados que pertenecen al ministerio de la persona que está autenticada, y a los entes ejecutores que corresponden a dicho ministerio.
Prototipo de Interfaz	
Poscondiciones:	Los usuarios registrados en el sistema son listados, según el rol de Ministerio.

Tabla 2.9: Descripción del Caso de Uso Listar Usuarios a Nivel de Ministerio.

2.4.3.10 Descripción del caso de uso: Listar Usuarios a Nivel de Ente Ejecutor

Caso de Uso:	Listar Usuarios a Nivel de Ente Ejecutor
Actores:	Ente Ejecutor
Resumen:	El caso de uso se inicia cuando el usuario selecciona la opción de listar usuarios, luego el sistema le muestra el listado de los usuarios existentes.
Precondiciones:	El usuario debe estar autenticado.
Referencias:	RF11
Prioridad:	Auxiliar
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
	1.1 El sistema muestra un listado ordenado alfabéticamente con los usuarios registrados que pertenecen al ente ejecutor de la persona que está autenticada.
Prototipo de Interfaz	
Poscondiciones:	Los usuarios registrados en el sistema son listados, según el rol de Ente Ejecutor.

Tabla 2.10: Descripción del Caso de Uso Listar Usuarios a Nivel de Ente Ejecutor.

2.4.3.11 Descripción del caso de uso: Listar Ministerios

Caso de Uso:	Listar Ministerios
Actores:	Directivo
Resumen:	El caso de uso se inicia cuando el usuario selecciona la opción de listar ministerios, luego el sistema le muestra un listado con todos los ministerios registrados.
Precondiciones:	El usuario debe estar autenticado.
Referencias:	RF12
Prioridad:	Auxiliar
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El usuario selecciona en el menú de la página la opción Administración y luego la opción Listar Ministerios.	1.1 El sistema muestra un listado de todos los ministerios existentes. Estos deberán aparecer ordenados alfabéticamente, mostrando primero los de Venezuela y luego los de Cuba.
Prototipo de Interfaz	
Poscondiciones:	Los ministerios registrados en el sistema son listados.

Tabla 2.11: Descripción del Caso de Uso Listar Ministerios.

2.4.3.12 Descripción del caso de uso: Listar Entes Ejecutores

Caso de Uso:	Listar Entes Ejecutores
Actores:	Directivo, Ministerio
Resumen:	El caso de uso se inicia cuando el usuario selecciona la opción de listar entes ejecutores, luego el sistema le muestra el listado de los entes ejecutores registrados.
Precondiciones:	El usuario debe estar autenticado.
Referencias:	RF13
Prioridad:	Auxiliar
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El usuario selecciona en el menú de la página la opción Administración y luego la opción Listar Entes Ejecutores.	<p>1.1 El sistema muestra un listado de todos los entes ejecutores registrados. Estos deberán aparecer ordenados alfabéticamente, mostrando primero los de Venezuela y luego los de Cuba. (en caso de que el rol del usuario sea Directivo de Gobierno, Directivo de Convenio, Secretaría Técnica o Administrador).</p> <p>En caso que el usuario autenticado posea el rol:</p> <p>a) Ministerio, ver Caso de Uso “Listar Entes Ejecutores a Nivel de Ministerio”.</p>

Prototipo de Interfaz

Administración Proyectos Reportes			Inicio Salir
Lista de Entes Ejecutores			X
Nombre	Siglas	País	
Nombre del Ente Ejecutor 1	Sigla 1	País 1	
Nombre del Ente Ejecutor 2	Sigla 2	País 1	
Nombre del Ente Ejecutor 3	Sigla 3	País 1	
Nombre del Ente Ejecutor 4	Sigla 4	País 1	
Nombre del Ente Ejecutor 5	Sigla 5	País 1	
Nombre del Ente Ejecutor 6	Sigla 6	País 2	
Nombre del Ente Ejecutor 7	Sigla 7	País 2	
Nombre del Ente Ejecutor 8	Sigla 8	País 2	
Nombre del Ente Ejecutor 9	Sigla 9	País 2	
Nombre del Ente Ejecutor 10	Sigla 10	País 2	
Anterior 1 2 3 Siguiente			

Poscondiciones:

Los entes ejecutores registrados en el sistema son listados.

Tabla 2.12: Descripción del Caso de Uso Listar Entes Ejecutores.

2.4.3.13 Descripción del caso de uso: Listar Entes Ejecutores a Nivel de Ministerio

Caso de Uso:	Listar Entes Ejecutores a Nivel de Ministerio
Actores:	Ministerio
Resumen:	El caso de uso se inicia cuando el usuario selecciona la opción de listar entes ejecutores, luego el sistema le muestra el listado de los entes ejecutores registrados.
Precondiciones:	El usuario debe estar autenticado.
Referencias:	RF13
Prioridad:	Auxiliar
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
	1.1 El sistema muestra un listado de todos los entes ejecutores que correspondan al ministerio de la persona autenticada. Estos deberán aparecer ordenados alfabéticamente, mostrando primero los de Venezuela y luego los de Cuba.
Prototipo de Interfaz	
	
Poscondiciones:	Los entes ejecutores registrados en el sistema son listados, según el rol de Ministerio.

Tabla 2.13: Descripción del Caso de Uso Listar Entes Ejecutores a Nivel de Ministerio.

2.5 Conclusiones Parciales

En este capítulo se mostró el Modelo del Dominio utilizado para comprender mejor el funcionamiento del área de interés, haciendo énfasis en su descripción y en la especificación de sus clases conceptuales. Se especificaron los requisitos funcionales y no funcionales del software. Por último, se presentó el Modelo de Casos de Uso del Sistema, donde se realizó la especificación formal de los casos de uso identificados, mostrándose los prototipos no funcionales para cada uno y se aplicaron algunos patrones de casos de uso al modelo planteado.

CAPÍTULO 3: RESULTADOS OBTENIDOS

La obtención de los artefactos desarrollados durante este trabajo, sirvió como entrada al proceso de diseño del software, que posteriormente se llevó a cabo por otro de los integrantes del equipo del proyecto. Luego de transcurrida la etapa de implementación y pruebas del producto, este se puso a consideración del cliente hasta que finalmente fue aceptado por el mismo, mostrando que se logró cumplir con sus necesidades. Para asegurar la validez del Modelo de Casos de Uso del Sistema se aplicaron métricas que permitieron evaluar la calidad del mismo, siendo el paso final luego del proceso de refinamiento de dicho modelo.

3.1 Resultados del Modelo del Dominio

La realización del Modelo del Dominio facilitó que se comprendiera la estructura de la organización y los conceptos u objetos que intervienen en el desarrollo del Convenio Cuba Venezuela y que son de interés para el Módulo Soporte. Proporcionó además, una noción de cómo debería ser desplegado el sistema. Un resultado valioso de esta actividad fue que se logró que los clientes y el equipo de desarrollo llegaran a un entendimiento común sobre el funcionamiento del negocio.

En general, en esta etapa se obtuvo el Diagrama de Clases del Dominio y la Definición de las Clases Conceptuales que intervienen en este. Además, el intercambio de ideas entre los especialistas y analistas permitió la derivación de los requerimientos del sistema y la identificación de actividades a automatizar en cada proceso, junto a las personas involucradas en cada una de ellas. El modelado del dominio constituyó una entrada fundamental para el flujo de trabajo Levantamiento de Requisitos.

3.2 Resultados de la Especificación de Requisitos

El Levantamiento de Requisitos es una de las etapas fundamentales por las que debe pasar un software. Para alcanzar los resultados esperados, fue importante primeramente, entender la definición y el alcance del problema que se estaba tratando de solucionar con el sistema.

Durante esta etapa se identificaron cada una de las necesidades del cliente, teniendo presente la filosofía del negocio en cuestión y respetando los roles de cada persona dentro de la organización, obteniendo como resultado los requisitos funcionales y no funcionales del sistema. Una vez terminada la captura de requisitos se procedió a la realización del Modelo de Casos de Uso del Sistema, donde fueron agrupados varios requisitos en un caso de uso hasta conformar el modelo.

3.3 Resultados del Modelo de Casos de Uso del Sistema

El Modelo de Casos de Uso del Sistema sirvió como medio de comunicación entre el cliente y los desarrolladores del sistema en cuanto a las funcionalidades del mismo. Este modelo describe los requisitos funcionales de un actor, en términos de las interacciones que realiza con el sistema. Dichas interacciones se describen por medio de uno o más flujos de eventos que ocurren para llevar a cabo una tarea.

En esta etapa se obtuvieron los actores del sistema, el diagrama de casos de uso del sistema y las descripciones de los casos de uso. En estas últimas se presentó una descripción detallada del flujo de eventos en función del responsable y las condiciones de excepción que contemplan estos flujos, conjuntamente con los prototipos de cada uno.

3.3.1 Prototipo del sistema

Al realizar la captura de requisitos, el cliente define un conjunto de objetivos generales para el software, pero no identifica los requisitos detallados de entrada, proceso o salida. Por esta razón es de importancia la construcción de prototipos al comienzo de la recolección de los requisitos. Se procede a la creación de un diseño rápido, que se centra en una representación de los aspectos del software que serán visibles para el cliente, como: enfoques de entrada y formatos de salida. (1)

El prototipo lo evalúa el cliente y se utiliza para refinar los requisitos del software a desarrollar. La iteración ocurre cuando el prototipo se pone a punto para satisfacer las necesidades del cliente, siguiendo el ciclo que se muestra en la Figura 3.1, permitiendo al mismo tiempo que el desarrollador comprenda mejor lo que necesita hacer. (1)

Sin importar la forma en que éste se aplique, el paradigma de construcción de prototipos ayuda al desarrollador de software y al cliente a entender mejor cuál será el resultado de la construcción cuando los requisitos estén satisfechos.

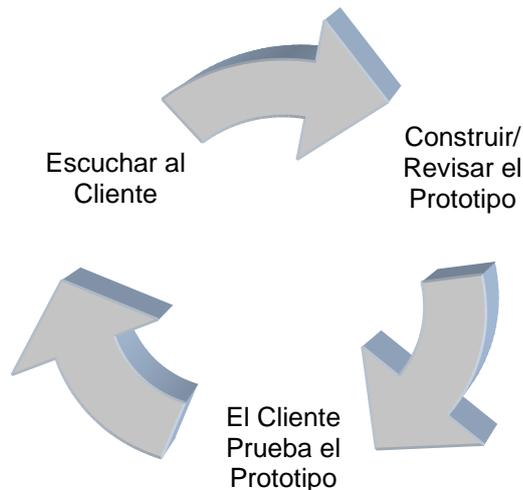


Figura 3.1: Paradigma de Construcción de Prototipos (1).

En esta etapa, luego de definir los objetivos del software e identificar los requisitos junto al cliente, se procedió a la realización del prototipo, que sería la guía en el desarrollo posterior del diseño de la aplicación. Para la realización de este, se hizo una división estableciéndose prototipos de acuerdo a un área funcional determinada, que es representada por los casos de uso definidos en el Diagrama de Casos de Uso del Sistema, mostrado en el Capítulo 2.

La realización de los prototipos de interfaz de usuario aportó ventajas, pues sirvió como retroalimentación con los clientes, quienes ayudaron en el refinamiento de los mismos. Por cada caso de uso del sistema se crearon nuevos prototipos de interfaces no funcionales de la aplicación. Estas interfaces mostraron las funcionalidades que podría activar cada actor al interactuar con un caso de uso específico a través de formularios. La generación de los prototipos de interfaz fue una forma de validar los requisitos funcionales y de usabilidad que fueron identificados durante el levantamiento de los requisitos.

3.3.2 Evaluación de la calidad

En los últimos años se ha hecho necesaria la evaluación de productos y procesos de software en distintos dominios de aplicación, es por ello que han surgido numerosas propuestas que definen conjuntos de métricas de calidad. Se define métrica como una medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo dado (3). Las métricas de software se refieren a un amplio rango de medidas para el software de computadoras, pueden ser aplicadas al proceso de software para mejorarlo y utilizadas en proyectos para ayudar en la estimación, el control de la calidad y la evaluación de la productividad (1).

Para evaluar la validez del Modelo de Casos de Uso del Sistema obtenido en el Capítulo 2, se empleó un Modelo de Métricas para el Análisis y Diseño Orientado a Objetos Basado en UML propuesto por la Universidad EAFIT de Colombia. Este modelo tiene como objetivo medir la calidad de la funcionalidad del sistema a partir del diagrama de casos de uso generado. Define cuatro atributos genéricos de propiedades de calidad: consistencia, correctitud, completitud y complejidad, que tienen un significado concreto de acuerdo al tipo de artefacto software y al nivel de abstracción que este describe. Estos atributos tienen un conjunto de factores y cada uno de estos tiene asociado una métrica. (16) (30)

Los atributos se definen de la siguiente forma:

- **Completitud:** Grado en que se ha logrado detallar todos los casos de uso relevantes.
- **Consistencia:** Grado en que los casos de uso del sistema describen las interacciones adecuadas entre el usuario y el sistema.
- **Correctitud:** Grado en que las interacciones actor / sistema soportan adecuadamente el proceso del negocio.
- **Complejidad:** Grado de claridad en la presentación de los elementos que describen el contexto y la claridad del sistema.

A continuación se muestran cada uno de los atributos con sus factores asociados:

Atributo	Factores
Completitud	<p>¿Han sido involucradas todas las áreas funcionales a las cuales apoyará el sistema?</p> <p>¿Han sido definidos todos los roles de usuario encargados de generar / modificar o consultar información?</p> <p>¿Se presenta una descripción detallada (descripción extendida esencial) de todos los casos de uso del sistema?</p> <p>¿Se presenta una descripción resumida (descripción de alto nivel) de todos los casos de uso?</p> <p>¿Están definidos todos los requisitos que justifican la funcionalidad de los casos de uso?</p> <p>¿Todos los requisitos han sido considerados en algún caso de uso?</p>

	<p>¿Están todas las acciones del flujo de eventos redactadas en función del responsable?</p> <p>¿Se describen las condiciones de excepción que debe contemplar cada flujo de eventos?</p> <p>¿Todos los casos de uso han sido clasificados de acuerdo a su relevancia (crítico, secundario, auxiliar, opcional)?</p>
Consistencia	<p>¿El nombre dado a los casos de uso es una expresión verbal que describe alguna funcionalidad relevante en el contexto del usuario?</p> <p>¿Representan los casos de uso una interacción observable por un actor?</p> <p>¿Está adecuadamente redactado (en el lenguaje del usuario) el flujo de eventos?</p> <p>¿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?</p> <p>¿Existe una adecuada separación entre el flujo básico de eventos y los flujos alternos?</p>
Correctitud	<p>¿Existe para cada caso de uso por lo menos un usuario responsable?</p> <p>¿Representa el caso de uso requisitos comprensibles por el usuario?</p> <p>¿Las interacciones definidas describen la funcionalidad requerida del sistema?</p>
Complejidad	<p>¿Los elementos dentro del diagrama están adecuadamente ubicados de manera que facilitan su interpretación?</p>

Tabla 3.1: Atributos y Factores para la Evaluación de la Calidad del Modelo de Casos de Uso del Sistema.

Del Modelo de Métricas empleado se tomaron solamente aquellos factores que se consideraron fundamentales a la hora de evaluar el Modelo de Casos de Uso del Sistema, de acuerdo a las particularidades del mismo. Para cada métrica está determinado su nombre, significado, umbral² y una acción sugerida³ (Anexo 5).

La Figura 3.2 muestra los resultados obtenidos luego de la aplicación de las métricas.

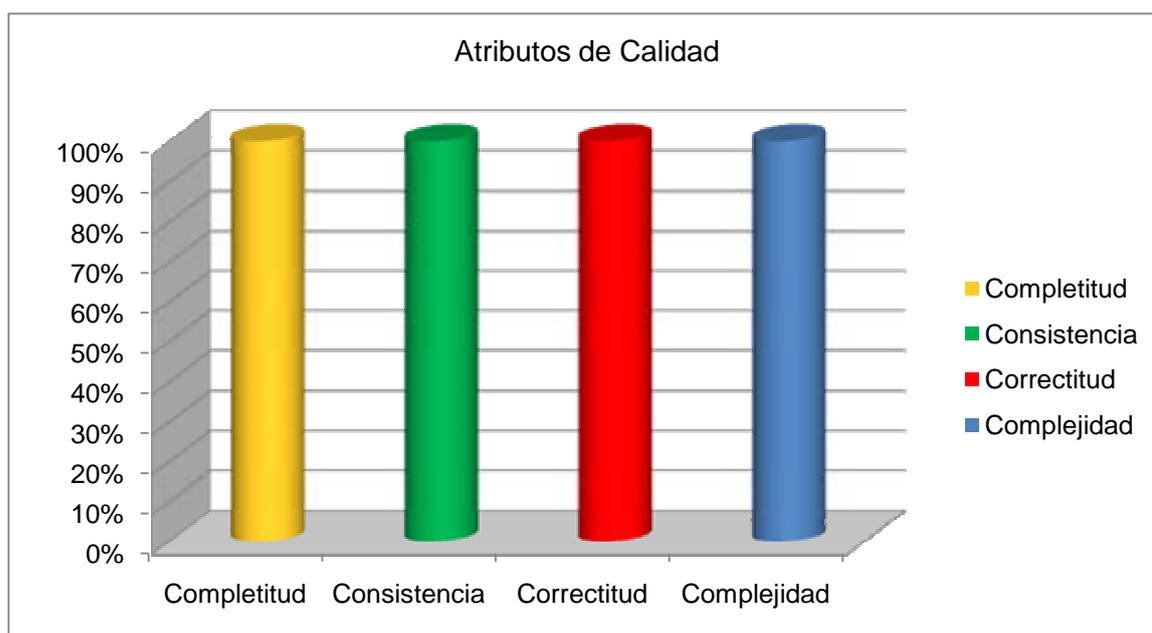


Figura 3.2: Grado de Calidad de la Funcionalidad del Modelo de Casos de Uso del Sistema.

En la evaluación efectuada se alcanzó una calificación superior de 100% de funcionalidad, pues todos los factores de los 4 atributos de calidad que se midieron, se encontraban presentes en el Modelo de Casos de Uso del Sistema realizado.

² El umbral es un valor heurístico para establecer un conjunto de rangos deseables e indeseables de los valores de las métricas para la fase medida. Estos permiten identificar las anomalías en el proceso de desarrollo de una aplicación.

³ Las acciones sugeridas son observaciones sobre aspectos a considerar para mejorar el comportamiento del factor medido. Básicamente los resultados de la medición son avisos para establecer un plan de acción cuando la métrica tiene un valor fuera del umbral establecido.

Los resultados de la evaluación muestran que el modelo posee la calidad requerida. El mismo soporta todas las operaciones de las áreas funcionales que fueron identificadas, muestra la interacción de los roles de usuario encargados de generar, consultar o modificar la información y da cumplimiento a través de los casos de uso a todos los requisitos identificados. Los casos de uso han sido clasificados de acuerdo a su relevancia, presentan una descripción resumida y otra detallada de las acciones del flujo de eventos y las condiciones de excepción que debe contemplar cada uno. Para una mejor comprensión, los casos de uso se nombraron con una expresión verbal que describe una funcionalidad relevante para el usuario, representando así una interacción observable para un actor del sistema. Los elementos dentro del diagrama están ubicados de manera que facilitan su interpretación.

3.4 Aceptación del Cliente

Luego de concluir el proceso de análisis del Módulo Soporte, se desarrollaron las demás fases del proceso de confección del software, que fueron llevadas a cabo por los demás integrantes del equipo de desarrollo, hasta que quedó completamente integrado con el otro módulo en un solo sistema. El producto final fue probado por el equipo de calidad durante varias iteraciones, hasta que finalmente fue liberado y aceptado por el cliente, mostrando la validez del trabajo realizado.

La Aplicación Inicial del Sistema de Gestión para el Seguimiento de los Proyectos del Convenio Integral de Cooperación Cuba – Venezuela, actualmente se encuentra publicada en internet, siendo utilizada por el personal de ambas partes que está involucrado en la actividad colaborativa.

El Acta de Aceptación por parte del cliente y el Acta de Entrega de la solución informática desarrollada se presentan a continuación:

Acta de Aceptación



Producto: Sistema de Gestión para Seguimiento de los Proyectos del Convenio Integral de Cooperación Cuba-Venezuela.

Categoría de las pruebas: Pruebas de Aceptación del Sistema

Fecha de la conciliación: 03/04/2008

Involucrados en el proceso:

Por la parte del Cliente (MENPET): Sandra Cortés

Por la parte del Suministrador (ALBET): Ing. Raykenler Yzquierdo

Observador Independiente (CALISOFT): Ing. Yeniset León

Observaciones del proceso:

Por acuerdo entre las partes involucradas en el proceso de las Pruebas de Aceptación, se ha tomado el acuerdo de Aceptar "La Aplicación Inicial del Sistema de Gestión para el Seguimiento de los Proyectos" y "El Manual de Usuario", con fecha 3 de abril de 2008.

Para que conste la aceptación de los resultados de las pruebas, presentes en este documento, y por tanto la aceptación de "La Aplicación Inicial del Sistema de Gestión para el Seguimiento de los Proyectos" y "Manual de Usuario", dando fe del acuerdo firman la presente, los principales representantes de las Partes.

Sandra Cortés
Representante MENPET

Ing. Raykenler Yzquierdo
Representante Parte
Suministradora
(ALBET)

Ing. Yeniset León
Observador independiente
(CALISOFT)





ACTA DE ENTREGA

En cumplimiento del CONTRATO E08-001 -000 DE SOLUCIÓN TECNOLÓGICA INTEGRAL PARA EL DESARROLLO DEL SISTEMA DE GESTIÓN PARA SEGUIMIENTO DE LOS PROYECTOS DEL CONVENIO INTEGRAL DE COOPERACIÓN CUBA VENEZUELA, se hace entrega de la documentación que se relaciona a continuación.

Referencia	Documento
	Aplicación Inicial del Sistema de Gestión para el Seguimiento de los Proyectos
CV - SW - DR - 001	Manual de Usuario

Entrega

NOMBRE Y APELLIDO

Ing. Yudier Cervantes Puga

CARGO

Jefe de Proyecto Solución Integral de Software

FIRMA



Recibe

NOMBRE Y APELLIDO

Sandra Cortés

CARGO

Gerente general

FIRMA

FECHA

03/04/2008

Referencia:

ALBET, S.A.

Centro de Negocios Miramar. Edificio
Barcelona, Oficina 322. Avenida 5ta e/ 76 y
78. Miramar. Playa, Ciudad Habana, Cuba

Tel/Fax: +53 (7) 837 2407

E-mail: albet@albet.cu

3.5 Conclusiones Parciales

En este capítulo se explicaron los resultados alcanzados como parte de toda la investigación. Se utilizó como método para justificar la validez del trabajo realizado una carta de aceptación del cliente, que avala por sí misma los resultados que se debían conseguir con el análisis del sistema. A parte de esto, se aplicó un modelo de métricas que permitió obtener una medida cuantitativa de la calidad del Modelo de Casos de Uso del Sistema propuesto.

CONCLUSIONES

Al término de este trabajo se puede arribar a las siguientes conclusiones:

- El estudio y selección de la metodología para el desarrollo de software y de las principales estrategias para la captura de requisitos permitió la modelación del sistema teniendo presente las necesidades del cliente.
- La metodología RUP y la herramienta Visual Paradigm fueron utilizadas para modelar los artefactos correspondientes, definiendo las clases conceptuales del dominio, los actores y los casos de uso del sistema a implementar posteriormente.
- La realización de la modelación del dominio facilitó tener una mayor claridad sobre el entorno de la organización.
- La Especificación de los Requisitos permitió definir las características principales y las funcionalidades con que debía contar el sistema.
- La realización del Modelo de Casos de Uso del Sistema posibilitó obtener una descripción detallada de los flujos de eventos de debía seguir el sistema.
- Los artefactos generados durante el proceso de análisis, constituyeron la entrada para el trabajo del diseñador del sistema y fueron imprescindibles para que el equipo de desarrollo lograra cumplir con su propósito de entregar un producto terminado al cliente.
- La evaluación de la calidad del proceso de análisis mediante un conjunto de métricas, realizado al Modelo de Casos de Uso del Sistema, obtuvo un resultado satisfactorio.
- El cliente expresó su aceptación con el trabajo realizado hasta la fecha.

RECOMENDACIONES

Para la obtención de mejores resultados en la siguiente fase del proyecto se recomienda incluir las siguientes funcionalidades al sistema:

- Eliminar personas, ministerios o entes ejecutores que estén registrados.
- Agregar más campos para la búsqueda de personas, ministerios y entes ejecutores en el sistema, para de esta forma mejorar las mismas.
- Gestionar los roles en el sistema, es decir, crear, modificar y eliminar los roles con los cuales podrán ser conformados los usuarios que tendrán acceso a la aplicación.
- Asignar a un rol creado, niveles de acceso con distintas funcionalidades según las actividades que realice la persona que interactúa con el sistema.
- Posibilitar la generación automática de la contraseña al crear un nuevo usuario en el sistema y enviarla por correo electrónico.

BIBLIOGRAFÍA

1. Pressman, Roger S. *Ingeniería del Software. Un Enfoque Práctico*. Quinta Edición, 2002.
2. Jacobson, Ivar, Booch, Grady y Rumbaugh, James. *El Proceso Unificado de Desarrollo de Software*. Madrid, España: Addison Wesley, 2000. ISBN 84-7829-036-2.
3. IEEE. IEEE Standard 610.12-1990 IEEE Standard Glossary of Software Engineering Terminology. [En línea] <http://standards.ieee.org/reading/ieee/std/se/610.12-1990.pdf>.
4. Abrahamsson, Pekka, et al. Agile Software Development Methods. Review and Analysis. [En línea] 2002. <http://www.inf.vtt.fi/pdf/publications/2002/P478.pdf>. ISBN 951-38-6010-8.
5. Molpeceres, Alberto. Procesos de desarrollo: RUP, XP y FDD. [En línea] Diciembre 15, 2002. <http://www.javahispano.org/contenidos.downloadatt.action?id=71>.
6. Palacio, Juan. Gestión y Modelos para la Eficiencia en Empresas de Desarrollo de Software. [En línea] http://www.baquia.com/marketing/Gestion_y_modelos_eficiencia_software.pdf.
7. Aguilar, Catherine. Aplicación de Conceptos de Gestión de Proyectos y Gestión de Riesgos en el Desarrollo de Productos Nuevos en el Campo de la Tecnología de Información. Universidad de Puerto Rico, Diciembre 2005.
8. Jeffries, Ron, Anderson, Ann y Hendrickson, Chet. *Extreme Programming Installed*. Addison Wesley, Octubre 2000. ISBN 0-201-70842-6.
9. Marches, Michele, et al. *Extreme Programming Perspectives*. Addison Wesley, Agosto 2002. ISBN 0-201-77005-9.
10. Beck, Kent. *Extreme Programming Explained*. Addison Wesley, Septiembre 1999. ISBN 0201616416.
11. Smith, John. A comparison of RUP and XP. [En línea] 2001. <ftp://ftp.software.ibm.com/software/rational/web/whitepapers/2003/TP167.pdf>.
12. Crispin, Lisa y House, Tip. *Testing Extreme Programming*. Addison Wesley, Octubre 2002. ISBN 0-321-11355-1.

13. Kruchten, Philippe. *The Rational Unified Process An Introduction, Second Edition*. Addison Wesley, Marzo 2000. ISBN 0-201-70710-1.
14. Kroll, Per y Kruchten, Philippe. *The Rational Unified Process Made Easy: A Practitioner's Guide to the RUP*. Addison Wesley, Abril 2003. ISBN 0-321-16609-4.
15. Rational Software Corporation. *Rational Unified Process*. 2003.
16. Plaza, Mairelys y Piñeiro, Yaniet. *Predictor: Sistema de Descarga y Procesamiento Automatizado de Patentes. Rol Analista de Sistemas*. Universidad de la Ciencias Informáticas, Ciudad de La Habana, Mayo 2007.
17. Robertson, Suzanne y Robertson, James. *Mastering the Requirements Process Second Edition*. Addison Wesley Professional, Marzo 2006. ISBN 0-321-41949-9.
18. Bacallao, Maylín y Pino, Yaumarys. *Propuesta de un Proceso para la Captura de Requisitos de la Segunda Fase del Desarrollo del Proyecto de Modernización de los Registro y Notarías de la República Bolivariana de Venezuela*. Universidad de las Ciencias Informáticas, Ciudad de La Habana, Mayo 2007.
19. Durán, Amador y Bernárdez, Beatriz. *Metodología para la Elicitación de Requisitos de Sistemas Software*. Universidad de Sevilla, España, Octubre 2001.
20. Larman, Craig. *UML y Patrones. Introducción al Análisis y Diseño Orientado a Objetos*. México : Addison Wesley, 1999. ISBN 970-17-0261-1.
21. Ridaó, Marcela, Doorn, Jorge y Sampaio, Julio C. *Uso de Patrones en la Construcción de Escenarios*. [En línea] http://wer.inf.puc-rio.br/WERpapers/artigos/artigos_WER00/ridao.pdf.
22. Övergaard, Gunnar y Palmkvist, Karin. *Use Cases Patterns and Blueprints*. Addison Wesley Professional, Noviembre 2004. ISBN 0-13-145134-0.
23. Instituto Nacional de Estadística e Informática. *Herramientas CASE*. [En línea] Noviembre 1999. <http://www.inei.gov.pe/biblioineipub/bancopub/Inf/Lib5103/Libro.pdf>. 875-99-OI-OTDETI-INEI.
24. Manrique, Jorge. *Introducción a las Herramientas Case*. [En línea] <http://www.iesjorgemanrique.es/hda/descargas/tema1.ppt>.

25. IBM. IBM Rational Suite family of products. [En línea] <ftp://ftp.software.ibm.com/software/rational/web/datasheets/version6/suite.pdf>.
26. Sparx Systems. Características de la herramienta de diseño UML Enterprise Architect. [En línea] http://www.sparxsystems.com.ar/products/ea_features.html.
27. Enterprise Architect - Herramienta de diseño UML y herramienta CASE UML para desarrollo de software. [En línea] <http://www.sparxsystems.com.ar/products/ea.html>.
28. Visual Paradigm. UML CASE Tools - Free for Learning UML, Cost-Effective for Business Solutions. [En línea] <http://www.visual-paradigm.com/product/vpuml/>.
29. DIMAGEN. Diseño Web Programación y Desarrollo Web Aplicaciones Web. [En línea] http://www.dimagin.net/es/contenido.php?t_id=6.
30. Macias, Danaysa. Sistema Integrado de Gestión Estadística, Rol Analista de Sistemas, Módulo Generador de Modelos. Universidad de la Ciencias Informáticas, Ciudad de La Habana, Junio 2007.

ANEXOS

Anexo 1: Plantilla para el Modelo del Dominio



Modelo del Dominio

Modelo del Dominio

<Nombre del Proyecto>

<Nombre del producto>

<Versión>

Control de versiones

Fecha	Versión	Descripción	Autor
<dd/mmm/yy>	<x.x>	<detalles>	<nombre>

Reglas de Confidencialidad

Clasificación: <<Clasificación>>

Este documento contiene información propietaria de **ALBET Ingeniería y Sistemas y/o** "**<<Empresa Cliente>>**", y es emitido confidencialmente para un propósito específico.

El que recibe el documento asume la custodia y control, comprometiéndose a no reproducir, divulgar, difundir o de cualquier manera hacer de conocimiento público su contenido, excepto para cumplir el propósito para el cual se ha generado.

Estas reglas son aplicables a las **X** páginas de este documento.

1. Introducción

1.1 Propósito

[\[Resumen del propósito de este documento\]](#)

1.2 Alcance

[\[Breve descripción del alcance del modelo del dominio\]](#)

1.3 Definiciones, Acrónimos y Abreviaturas

1.4 Referencias

[\[Lista de documentos a los que se hace referencia\]](#)

2. Diagrama de clases del modelo del dominio

3. Definición de las clases del modelo del dominio

3.1 <Primera clase>

3.1.1 Descripción de la clase

Anexo 2: Plantilla para la Especificación de Requisitos



Especificación de Requisitos

Especificación de Requisitos

<Nombre del Proyecto>

<Nombre del Producto>

<Versión>

Control de versiones

Fecha	Versión	Descripción	Autor
<dd/mmm/yy>	<x.x>	<detalles>	<nombre>

Reglas de Confidencialidad

Clasificación: <<Clasificación>>

Este documento contiene información propietaria de **ALBET Ingeniería y Sistemas y/o "<<Empresa Cliente>>"**, y es emitido confidencialmente para un propósito específico.

El que recibe el documento asume la custodia y control, comprometiéndose a no reproducir, divulgar, difundir o de cualquier manera hacer de conocimiento público su contenido, excepto para cumplir el propósito para el cual se ha generado.

Estas reglas son aplicables a las **X** páginas de este documento.

1. Introducción

1.1 Propósito

[Resumen del propósito de este documento]

1.2 Alcance

[Proyectos con los que se involucra la Especificación]

1.3 Definiciones, Acrónimos y Abreviaturas

1.4 Referencias

[Lista de documentos a los que se hace referencia en la Especificación]

Código	Título
[1]	Documento 1
[2]	Documento 2
[3]	Modelo de Diseño - Módulo de Administración v0.0

2. Funcionalidad

[Esta sección describe los requisitos funcionales del sistema expresados en lenguaje natural. Típicamente se organiza por características pero también son apropiados métodos alternativos de organización como por ejemplo por usuario o subsistema.]

2.1 <Requisito Funcional 1..n>

[Descripción del requisito]

3. Usabilidad

[Esta sección incluye todos los requisitos que afectan la usabilidad. Ejemplos:

- Especificar el tiempo de entrenamiento requerido para que usuarios normales y avanzados sean productivos operando el sistema.
- Especificar requisitos acordes con estándares de usabilidad establecidos]

3.1 < Requisito de Usabilidad 1..n>

[Descripción del requisito]

4. Fiabilidad

[En esta sección se especifican los requisitos relacionados con la Fiabilidad. Ejemplos:

- Disponibilidad – especificar porcentaje de tiempo disponible (xx.xx%), horas de uso, acceso para mantenimiento, modo de funcionamiento degradado, etc.
- Tiempo medio entre fallos – usualmente se especifica en horas pero puede también especificarse en términos de días, meses o años.
- Tiempo medio de reparación – ¿Cuánto tiempo está permitido que el sistema quede fuera de operación luego de haber fallado?
- Exactitud – especificar la precisión y exactitud requerida en las salidas del sistema.
- Máximo de errores – usualmente es expresado en términos de errores/MLC (miles de líneas de código) o errores/puntos de función.
- Errores – categorizar los errores en términos de menores, significativos y críticos: los requisitos deben definir qué se entiende por error crítico (ej. Pérdida total de los datos o inhabilitadas para el uso ciertas partes del funcionamiento del sistema).]

4.1 < Requisito de Fiabilidad 1..n>

[Descripción del requisito]

5. Eficiencia

[Deben perfilarse en esta sección las características de la eficiencia del sistema. Incluir los tiempos de respuesta específicos. Donde sea aplicable, hacer referencia a los Casos de Uso por el nombre.

- Tiempo de respuesta por transacción (promedio, máximo).
- Rendimiento (ej. transacciones por segundo, cantidad de datos que pueden ser transferidos en un segundo).
- Capacidad (ej. número de clientes o transacciones que el sistema puede alojar).
- Modos de degradación (cuál es el modo de operación aceptable cuando el sistema de alguna forma ha sido degradado).
- Utilización de recursos (memoria, disco, comunicaciones, etc.)

5.1 < Requisito de Eficiencia 1..n>

[Descripción del requisito]

6. Soporte

[Esta sección indica cualquier requisito que refuerce el soporte o mantenimiento del sistema a construir, incluyendo normas de codificación, convenciones para nombrado, bibliotecas de clase, el acceso y utilidades de mantenimiento.]

6.1 < Requisito de Soporte 1..n>

[Descripción del requisito]

7. Restricciones de diseño

[Esta sección debe indicar cualquier restricción de diseño en el sistema a construir. Las restricciones representan decisiones de diseño que se han tomado y a las cuales es necesario adherirse (ej.: lenguajes de programación, requisitos de proceso de software, el uso prescrito de herramientas de desarrollo, restricciones de arquitectura y diseño, componentes comprados, las bibliotecas de la clase, etc.)]

7.1 < Requisito de Diseño 1..n>

[Descripción del requisito]

Anexo 3: Plantilla para el Modelo de Casos de Uso del Sistema



Modelo de Casos de Uso del Sistema

Modelo de Casos de Uso del sistema

<Nombre del Proyecto>

<Nombre del Producto>

<Versión>

Control de versiones

Fecha	Versión	Descripción	Autor
<dd/mmm/yy>	<x.x>	<detalles>	<nombre>

Reglas de Confidencialidad

Clasificación: <<Clasificación>>

Este documento contiene información propietaria de **ALBET Ingeniería y Sistemas y/o "<<Empresa Cliente>>"**, y es emitido confidencialmente para un propósito específico.

El que recibe el documento asume la custodia y control, comprometiéndose a no reproducir, divulgar, difundir o de cualquier manera hacer de conocimiento público su contenido, excepto para cumplir el propósito para el cual se ha generado.

Estas reglas son aplicables a las **X** páginas de este documento.

1. Introducción

1.1 Propósito

[Resumen del propósito de este documento]

1.2 Alcance

[Breve descripción del alcance del modelo de casos de uso del sistema]

1.3 Definiciones, Acrónimos y Abreviaturas

1.4 Referencias

[Lista de documentos a los que se hace referencia]

2. Actores del Sistema

[Se especifican todos los actores del sistema y se le asocia una descripción simple de cada uno de ellos]

Actor	Descripción

3. Diagrama de Casos de Uso del Sistema

[Figura que ilustre del modelo de casos de uso del sistema]

4. Especificación de los Casos de Uso

4.1 <Primer Caso de Uso del Sistema>

4.1.1 Descripción de Casos de Uso

[Se describe en la tabla los detalles del caso de uso en fusión de acción del actor y respuesta del sistema.

Si se decide tener un documento independiente para la definición de cada Caso de Uso, en esta sección se haría referencia a ese documento]

Caso de Uso:	
Actores:	
Resumen:	
Precondiciones:	
Referencias:	
Prioridad:	
Flujo Normal de Eventos	
Sección “”	
Acción del Actor	Respuesta del Negocio
<i>Prototipo de Interfaz</i>	
Flujos Alternos	
Acción del Actor	Respuesta del Negocio
<i>Prototipo de Interfaz</i>	
Poscondiciones:	

Anexo 4: Plantilla para el Glosario de Términos



Glosario de términos

Glosario de términos

<Nombre del Proyecto>

<Nombre del Producto>

<Versión>

Control de versiones

Fecha	Versión	Descripción	Autor
<dd/mmm/yy>	<x.x>	<detalles>	<nombre>

Reglas de Confidencialidad

Clasificación: <<Clasificación>>

Este documento contiene información propietaria de **ALBET Ingeniería y Sistemas y/o "<<Empresa Cliente>>"**, y es emitido confidencialmente para un propósito específico.

El que recibe el documento asume la custodia y control, comprometiéndose a no reproducir, divulgar, difundir o de cualquier manera hacer de conocimiento público su contenido, excepto para cumplir el propósito para el cual se ha generado.

Estas reglas son aplicables a las **X** páginas de este documento.

1. Introducción

1.1 Propósito

[Definir términos generales establecidos por la Dirección de Calidad y Normas de la UCI para los proyectos Productivos]

1.2 Alcance

[Todos los proyectos de la UCI]

1.3 Referencias

[Lista de documentos a los que se hace referencia]

Código	Título
[1]	Documento 1
[2]	Documento 2
[3]	Modelo de Diseño - Módulo de Administración v0.0

1.4 Resumen

[En el glosario aparecen un grupo de términos básicos para los proyectos productivos]

2. Definiciones

2.1 < Definición 1..n >

Anexo 5: Factores y Métricas del Modelo de Casos de Uso del Sistema

Factores de Completitud	Métricas Asociadas
Factor 1: ¿Han sido involucradas todas las áreas funcionales a las cuales apoyará el sistema?	<p>Métrica 1: Número de áreas funcionales omitidas.</p> <p>Umbral: < 10%</p> <p>Acciones sugeridas: Revisar los requerimientos del negocio.</p>
Factor 2: ¿Han sido definidos todos los roles de usuario encargados de generar / modificar o consultar información?	<p>Métrica 2: Número de roles omitidos.</p> <p>Umbral: < 10%</p> <p>Acciones sugeridas: Revisar el alcance del sistema e involucrar tipos de usuarios representativos de cada una de las áreas funcionales.</p>
Factor 3: ¿Se presenta una descripción resumida (descripción de alto nivel) de todos los casos de uso?	<p>Métrica 3: Número de casos de uso que no tiene descripción resumida.</p> <p>Umbral: < 10%</p> <p>Acción sugerida: Completar la descripción resumida del caso de uso.</p>
Factor 4: ¿Se presenta una descripción detallada (descripción extendida esencial) de todos los casos de uso?	<p>Métrica 4: Número de casos de uso que no poseen una descripción extendida.</p> <p>Umbral < 20%</p> <p>Acción sugerida: Interactuar con el cliente para realizar la definición extendida del caso de uso que sea consistente con la definición a alto nivel.</p>

<p>Factor 5: ¿Están definidos todos los requisitos que justifican la funcionalidad del caso de uso?</p>	<p>Métrica 5: Número de requisitos omitidos por caso de uso.</p> <p>Umbral < 10%</p> <p>Acción sugerida: Revisar la lista de requisitos para determinar cuáles serán apoyados por cada caso de uso.</p>
<p>Factor 6: ¿Existen requisitos que no han sido considerados en algún caso de uso?</p>	<p>Métrica 6: Número de requisitos que no son considerados en ningún caso de uso.</p> <p>Umbral < 10%</p> <p>Acción sugerida: Revisar la lista de requisitos para determinar cuáles deben ser incorporados en algún caso de uso.</p>
<p>Factor 7: ¿Están todas las acciones del flujo de eventos redactadas en función del responsable?</p>	<p>Métrica 7: Número de acciones del flujo de eventos que no están redactadas en función del responsable.</p> <p>Umbral < 20%</p> <p>Acción sugerida: Revisar las responsabilidades tanto del actor (actores) como del sistema.</p>
<p>Factor 8: ¿Se describen las condiciones de excepción que debe contemplar cada flujo de eventos?</p>	<p>Métrica 8: Número de casos de uso que no describen condiciones de excepción.</p> <p>Umbral < 20%</p> <p>Acción sugerida: Revisar las excepciones presentadas en el flujo de eventos que producen un mensaje de error al usuario.</p>

<p>Factor 9: ¿Todos los casos de uso han sido clasificados de acuerdo a su relevancia (crítico, secundario, auxiliar, opcional)?</p>	<p>Métrica 9: Número de casos de uso que no han sido clasificados.</p> <p>Umbral: < 10%</p> <p>Acción sugerida: Hacer reuniones con los usuarios para analizar y priorizar los requisitos de acuerdo a su relevancia.</p>
Factores de Consistencia	Métricas Asociadas
<p>Factor 10: ¿El nombre dado a los casos de uso es una expresión verbal que describe alguna funcionalidad relevante en el contexto del usuario?</p>	<p>Métrica 10: Número de casos de uso que tienen un nombre incorrecto.</p> <p>Umbral < 20%</p> <p>Acción sugerida: Modificar el nombre del caso de uso de tal manera que signifique una acción desde el punto de vista del usuario.</p>
<p>Factor 11: ¿Representa el caso de uso una interacción observable por un actor?</p>	<p>Métrica 11: Número de casos de uso que no representan una interacción observable por un actor.</p> <p>Umbral < 5%</p> <p>Acción sugerida: Eliminar el caso de uso e incorporar su funcionalidad como una responsabilidad del sistema dentro de otro caso de uso.</p>
<p>Factor 12: ¿Está adecuadamente redactado (en el lenguaje del usuario) el flujo de eventos?</p>	<p>Métrica 12: Número de casos de uso que no está redactado adecuadamente el flujo de eventos.</p> <p>Umbral < 10%</p> <p>Acción sugerida: Corregir la descripción del caso de uso.</p>

<p>Factor 13: ¿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?</p>	<p>Métrica 13: Número de casos de uso cuya descripción extendida no inicia con una acción externa o con una condición monitoreada por el sistema.</p> <p>Umbral: < 10%</p> <p>Acción sugerida: Completar la definición del caso de uso incluyendo la acción fuera del sistema que da inicio al caso de uso o la condición interna que el sistema tiene controlar para dar inicio al caso de uso.</p>
<p>Factor 14: ¿Existe una adecuada separación entre el flujo básico de eventos y los flujos alternos y/o flujos subordinados?</p>	<p>Métrica 14: Número de casos de uso complejos que no tienen separación del flujo básico y de flujos alternos.</p> <p>Umbral: < 20%</p> <p>Acción sugerida: Estructurar el caso de uso de manera que separe su funcionalidad básica (caso de uso base) de la funcionalidad repetitiva o alternativa. Si hay pasos repetitivos, formar un caso de uso que lo incluye y con los pasos alternativos formar un caso de uso que lo extienda.</p>
<p>Factores de Correctitud</p>	<p>Métricas Asociadas</p>
<p>Factor 15: ¿Existe para cada caso de uso por lo menos un usuario responsable?</p>	<p>Métrica 15: Número de casos de uso que no tienen un usuario responsable.</p> <p>Umbral: < 10%</p> <p>Acción sugerida: Analizar la responsabilidad que representa el caso de uso y acordar con los usuarios cuál es el responsable directo de éste.</p>

<p>Factor 16: ¿Representa el caso de uso requisitos comprensibles por el usuario?</p>	<p>Métrica 16: Número de casos de uso en que los requisitos representados no son comprensibles por el usuario.</p> <p>Umbral: < 5%</p> <p>Acción sugerida: Discutir con el usuario la interacción que describe el caso de uso y ajustar dicha descripción de manera que sea comprensible por el usuario.</p>
<p>Factor 17: ¿Las interacciones definidas describen la funcionalidad requerida del sistema?</p>	<p>Métrica 17: Número de casos de uso que deben ser modificados para adecuarlos a la funcionalidad del sistema.</p> <p>Umbral: < 10%</p>
<p>Factores de Complejidad</p>	<p>Métricas Asociadas</p>
<p>Factor 18: ¿Los elementos dentro del diagrama están adecuadamente ubicados de manera que facilitan su interpretación?</p>	<p>Métrica 18: Número de elementos del diagrama que requieren reubicación.</p> <p>Umbral: < 30%</p> <p>Acción sugerida: Modificar la ubicación de los elementos del diagrama de manera que los elementos relacionados se encuentren lo más cercano posible.</p>

GLOSARIO

Actor: Conjunto coherente de roles que los usuarios de casos de uso desempeñan cuando interactúan con estos.

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

Captura de Requisitos: Proceso durante el cual se identifica un problema y se especifican los requisitos que debe cumplir un producto de software.

Caso de Uso: Secuencias de acciones que el sistema puede llevar a cabo interactuando con sus actores, incluyendo alternativas dentro de las secuencias.

Cliente: Persona o empresa que contrata al desarrollador de software.

Comisión Mixta: Reunión que se realiza cada año con representantes de ambos países, con el fin de servir como mecanismo para el cumplimiento y seguimiento de las acciones de cooperación previstas en el Convenio.

Convenio Cuba Venezuela: Acuerdo de cooperación integral firmado entre la República de Cuba y la República Bolivariana de Venezuela, con el objetivo de elaborar de común acuerdo, programas y proyectos que beneficien a ambos países.

Diagrama: Representación gráfica de un conjunto de elementos, usualmente representado como un grafo conectado de vértices (elementos) y arcos (relaciones).

Diagrama de casos de uso del sistema: Representa la estructura del sistema en términos de casos de uso agrupando los requisitos funcionales. Muestra las relaciones entre los casos de uso del sistema y los actores.

Diagrama de clases del dominio: Representa los conceptos u objetos del mundo real significativos para un problema o área de interés. Muestra las relaciones entre los conceptos u objetos del dominio del problema con sus atributos.

Ente Ejecutor: Empresa o institución que lleva a cabo proyectos que se firman en las Comisiones Mixtas.

Especificación de Requisitos: Documento que describe lo que hace un sistema de software: sus funciones y sus atributos. Generalmente escritas desde el punto de vista del usuario.

Herramienta CASE (Computer Aided Software Engineering): Diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software, reduciendo el costo de las mismas en términos de tiempo y de dinero.

IEEE (Institute of Electrical and Electronics Engineers): Importante asociación de técnicos y profesionales, con sede en los Estados Unidos, que favorece la investigación en campos diversos, como la tecnología aeroespacial, la computación, las comunicaciones y la tecnología biomédica. Promueve la estandarización de normas.

Ingeniería de Requisitos: Conjunto de actividades en las cuales, utilizando técnicas y herramientas, se analiza un problema y se concluye con la especificación de una solución (a veces más de una).

Ingeniería de Software: Tecnología multicapa en la que se pueden identificar los métodos, el proceso y las herramientas.

Ministerio: Dirigen las actividades de los entes ejecutores y controlan los proyectos que estos ejecutan.

Poscondición: Restricción que ha de ser cierta al completarse una operación.

Precondición: Restricción que ha de ser cierta cuando una operación es invocada.

Proceso de Desarrollo de Software: Definición del conjunto de actividades que guían los esfuerzos de las personas implicadas en el proyecto, a modo de plantilla, que explica los pasos necesarios para terminar el proyecto. Un proceso define: “quién”, “qué”, “cuándo” y “cómo” hay que realizar las cosas para alcanzar un determinado producto de software.

Prototipo: Maqueta visual funcional o no de la futura aplicación. Este puede ser una imagen o una aplicación software que simule funcionalidades del software.

RUP (Rational Unified Process): Proceso que de manera ordenada define tareas y quién, cómo y cuándo el equipo de desarrollo las hará. Define un ciclo de vida iterativo, dirigidos por casos de uso y centrado en la arquitectura.

Requerimiento: Condición o capacidad necesaria para un usuario para resolver un problema o alcanzar un objetivo.

Requerimiento funcional: Capacidades o condiciones que el sistema debe cumplir.

Requerimiento no funcional: Propiedades o cualidades que el producto debe tener. Limitación de hardware o software bajo la cual el sistema debe operar.

Rol: Conjunto de expectativas de conducta asociadas a una persona, un patrón de comportamiento que se espera de quién desempeñe cada puesto, con cierta independencia de la persona que sea.

Secretaría Técnica: Entidad en cada país encargada de coordinar todas las actividades necesarias para el buen desarrollo de cada proyecto.

UML (Unified Modeling Language): Lenguaje gráfico que brinda un vocabulario y reglas para especificar, construir, visualizar y documentar los artefactos de un sistema utilizando el enfoque orientado a objetos.

Usuario: Persona o grupo de personas dentro de la empresa que utilizan el software desarrollado.

XP (Extreme Programming): El más destacado de los procesos ágiles para el desarrollo de software. Utilizada para proyectos de corto plazo. Consiste en desarrollos rápidos e iterativos, cuya particularidad es tener como parte del equipo al usuario final.