

# Universidad de las Ciencias Informáticas

## Facultad 3



**Título: Desarrollo de los procesos de Registro de Agencias de Viajes y Registro de Empresas Importadoras y Exportadoras del proyecto Sistema de Información Registral de la Cámara de Comercio.**

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

Autor

Yasmany García Granada

Tutor

Ing. Yurisleidys Leiva Zuñiga

La Habana, Cuba

2012

## 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 2012.

---

Yasmany García Granado

---

Ing. Yurisleidys Leiva Zuñiga

## AGRADECIMIENTOS

*A mis padres, por preocuparse por mí, educarme y apoyarme incondicionalmente en todas las decisiones que he tomado en la vida.*

*A mi madre Dolores, por haberme enseñado a luchar por las cosas que quiero, por apoyarme en todo momento, por ser la mejor madre del mundo, por ser lo mejor que he tenido en la vida, te quiero mima.*

*A mi padre Gerardo, que aunque desde hace poco tiempo ya no está presentemente físicamente, hasta hace apenas un año y pocos meses me apoyó en todo momento.*

*A mi querida hermana Mey, por compartir, por ayudarme, por entenderme, por quererme, por preocuparse.*

*A mi familia en general que siempre se ha preocupado por mí y me ha apoyado en todo momento. A mi abuela María que la quiero con la vida, a mi abuela Ana y a mi abuelo Gerardo. A mi "corazón de carne" mi tía Teresa que la amo, por quererme, ayudarme, apoyarme, estar pendiente, por ser mi segunda madre. A mi tía abuela Alejandrina, que es como si fuera mi abuela, por ser tan especial y brindarme su cariño y su apoyo. A mis tíos Pedro, Elisa, Blanca, Ariel y Danilo, a mi prima Marbelis.*

*A todos mis maestros y profesores de la primaria, la secundaria y el preuniversitario por influir en mi formación, especialmente a Angelita, Zenia, Leonor Aladro, Mercedes Esteris, Martha Hernández. A mis profesores de la universidad, especialmente Rubén de León, Esther Cristina, Mónica Cortina y Yoan Martínez.*

*A mi jefe de proyecto, Jorrín, por toda su preocupación y ayuda en el tema de la investigación.*

*A mis amigos, los que estuvieron a mi lado en los buenos y malos momentos, los que me supieron escuchar, entender, ayudar, a los que lloraron conmigo, a Yurita, Linnet, Albert Garnache, Mairovis, Milena, Liván, Héctor, Yaudel, Yunior, Pepe, Albert la Rosa, Alberto Milán, Yudier, Martín...*

**DEDICATORIA**

*A mi madre Dolores, por llenar todos mis días y ser lo más sagrado del mundo para mí.*

*A mi padre Gerardo, te recuerdo siempre papi y te quiero mucho.*

*A mi hermana Mey, por tener la dicha de tenerla a mi lado, te adoro mi hermanita.*

## **RESUMEN**

La Cámara de Comercio de la República de Cuba es una institución vinculada al comercio, la industria y los servicios reconocida ante el estado cubano. En la misma se realizan cuatro procesos principales, relacionados fundamentalmente con las actividades registrales de sus entidades asociadas. Actualmente la información que se genera en la realización de estos procesos es gestionada de forma manual, por lo que surgen varios problemas relacionados con el seguimiento y actualización de los datos. Es por ello que surgió la necesidad de contar con un sistema informático para la gestión de la información manejada en dichos procesos, para que se realice de manera centralizada, estandarizada y accesible, y que permita realizar búsquedas y generar reportes.

El presente trabajo aborda elementos del proceso de desarrollo de software llevado a cabo para el desarrollo de los procesos de Registro de Agencias de Viajes y Registro de Empresas Importadoras y Exportadoras del Sistema de Información Registral de la Cámara de Comercio.

Se realiza un estudio de las herramientas y tecnologías que fueron usadas en la implementación de la solución. Las mismas fueron escogidas de acuerdo a varios criterios de selección establecidos por el equipo de desarrollo. Se tuvieron en cuenta los beneficios de usarlas y las limitaciones o restricciones existentes en el contexto del proyecto, el centro de desarrollo, la universidad y el país. También fueron documentados, verificados y validados los artefactos obtenidos durante las etapas del proceso de desarrollo del software.

## **PALABRAS CLAVE**

Cámara de Comercio, procesos de negocio, desarrollo de software, Registro de Agencias de Viaje, Registro de Empresas Importadoras y Exportadoras.

ÍNDICE

<b>INTRODUCCIÓN</b> .....	<b>1</b>
<b>CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA</b> .....	<b>6</b>
1.1 Sistemas para resolver problemáticas similares en otros países .....	6
1.2 Criterios de selección de herramientas y tecnologías .....	7
1.3 Notación de Modelado de Procesos de Negocio .....	10
1.4 Metodología de desarrollo de software.....	10
1.5 Proceso Unificado de Desarrollo de Software .....	11
1.5.1 Lenguaje Unificado de Modelado .....	14
1.6 Gestor de procesos del negocio .....	14
1.6.1 Bonita Open Solution .....	15
1.7 Herramientas CASE .....	16
1.7.1 Visual Paradigm for UML .....	16
1.8 Lenguajes de programación.....	17
1.8.1 El lenguaje de programación Java .....	17
1.9 La plataforma empresarial de Java.....	18
1.9.1 APIs JEE .....	19
1.10 Servidores de aplicaciones .....	21
1.10.1 Servidor de aplicaciones Tomcat.....	22
1.11 Entornos de desarrollo .....	22
1.11.1 Netbeans .....	23
1.12 Frameworks de desarrollo.....	23
1.13 EclipseLink.....	24
1.13.1 Spring .....	24
1.14 Servidor de base de datos .....	26
1.14.1 PostgreSQL.....	26
1.15 Conclusiones parciales .....	27
<b>CAPÍTULO 2: SOLUCIÓN PROPUESTA</b> .....	<b>28</b>
2.1 Modelado de procesos de negocio .....	28
2.2 Requisitos de software.....	32
2.2.1 Requisitos funcionales .....	32
2.2.2 Requisitos no funcionales .....	34
2.3 Actores del sistema.....	36
2.4 Diagrama de caso de uso del sistema.....	37
2.4.1 Descripción de casos de uso del sistema .....	38
2.5 Elementos arquitectónicos .....	44
2.6 Diagrama de clases del diseño.....	48
2.7 Diagrama de secuencia.....	49
2.9 Conclusiones parciales .....	51
<b>CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA</b> .....	<b>52</b>
3.1 Diagrama de componentes .....	52

3.2	Validación del modelado de los procesos de negocio .....	53
3.3	Validación de requisitos .....	54
3.4	Métricas de software .....	54
3.4.1	Métrica de la calidad de la especificación .....	55
3.4.2	Métrica para la calidad de la funcionalidad del diagrama de casos de uso del sistema .....	56
3.5	Listas de chequeo .....	59
3.6	Revisión por pares.....	60
3.7	Pruebas de caja negra .....	61
3.8	Conclusiones parciales .....	62
	<b>CONCLUSIONES GENERALES .....</b>	<b>63</b>
	<b>RECOMENDACIONES.....</b>	<b>64</b>
	<b>BIBLIOGRAFÍA.....</b>	<b>65</b>
	<b>ANEXOS .....</b>	<b>69</b>
	<b>GLOSARIO DE TÉRMINOS.....</b>	<b>72</b>

## ÍNDICE DE FIGURAS

Figura 1: Fases, iteraciones y disciplinas de RUP.....	12
Figura 2: Esquema general de la JEE.....	21
Figura 3: Módulos del marco de trabajo Spring.....	26
Figura 4: DPN Inscripción de Agencia de Viaje Nacional.....	30
Figura 5: Diagrama de caso de uso del sistema.....	38
Figura 6: Arquitectura del sistema.....	46
Figura 7: Diagrama de clases del diseño del CUS Inscribir Agencia.....	49
Figura 8: Diagrama de secuencia del CUS Inscribir Agencia.....	50
Figura 9: Diagrama de componentes del sistema.....	53
Figura 10: Resultados de la métrica de la calidad de la especificación.....	56
Figura 11: Resultados de la evaluación de la calidad en los diagramas de casos de usos.....	59
Figura 12: Resultados obtenidos en la revisión por pares.....	60

## ÍNDICE DE TABLAS

Tabla 1: Actores del sistema.....	37
Tabla 2: Descripción del caso de uso Inscribir Agencia.....	39
Tabla 3: Equipo de revisión.....	55
Tabla 4: Interpretación de los revisores para cada requisito.....	56
Tabla 5: Métricas correspondientes a los atributos de calidad.....	57
Tabla 6: Resultados de la aplicación de la métrica para la calidad de la funcionalidad del diagrama de casos de uso del sistema.....	58
Tabla 7: Resultados de la revisión por pares.....	60

## INTRODUCCIÓN

En el transcurso de la historia las actividades comerciales han sido un factor de vital importancia para el desarrollo de las sociedades a nivel mundial. Surgieron entonces los primeros grupos de personas dedicados a labores relacionadas con artes y oficios varios, con lo que se lograron agrupar personas que ejercían una misma actividad, creándose además mecanismos para la regulación de productos.

Las sociedades establecidas en el medio oriente hacia el año 4000 a.C. organizaron sus actividades comerciales estableciendo oficinas destinadas a esa finalidad, además de promover sus derechos en el contexto de tales actividades. De manera que la actividad comercial no fue solo interés de personas aisladas, sino que existía la intención establecer una institución social que fuera un organismo representativo para regular el comercio.

Con estos antecedentes surgen las cámaras de comercio como instituciones o asociaciones que operan en determinada región de las divisiones político – administrativas, y que tienen como función proteger los intereses de los comerciantes que desarrollan actividades industriales, de importación, exportación, mayoristas, minoristas y otras. Una de sus funciones fundamentales es el mejoramiento de las actividades comerciales así como la vida social del área donde se desarrollan. Varias fuentes coinciden en afirmar que la primera Cámara de Comercio surgió en Francia, específicamente en Marsella en el año 1559. La misma protegía los derechos de comerciantes y manufactureros que se asociaban a esta, promoviendo además sus productos y estimulando las transacciones comerciales.

Desde el siglo XIX Cuba no ha estado exenta a la organización de las actividades comerciales que se realizan a nivel mundial. Las instituciones relacionadas con la actividad comercial en el país han ido evolucionando y transformándose hasta que en el año 1963, después del triunfo revolucionario de 1959, se creó la Cámara de Comercio de la República de Cuba que mantiene su vigencia hasta la actualidad.

La Cámara de Comercio de la República de Cuba es una asociación de empresas vinculadas al comercio, la industria y los servicios reconocida ante el estado cubano. Tiene como uno de sus objetivos el desarrollo de la actividad empresarial de sus entidades asociadas. Para ello les ofrece las mejores alternativas disponibles. Para el país constituye una poderosa herramienta para la inserción de la economía cubana en el mundo de las relaciones económicas internacionales, pues potencia e intercambia información valiosa en torno a las posibilidades de

negocios a escala mundial. Esta entidad promueve productos y servicios exportables, así como las oportunidades de negocios e inversión de la empresa cubana y la sustitución de importaciones en beneficio de la economía nacional.

La actividad económica debe ir siempre en ascenso a través del surgimiento de nuevos servicios. Se hace necesaria por consiguiente la actualización constante para estar acorde con el nuevo contexto; el uso de aplicaciones informáticas es una excelente forma de trabajar en este sentido. La Cámara de Comercio de la República de Cuba lleva la responsabilidad de encontrar las mejores formas de lograr mercados convenientes haciendo uso de habilidades empresariales, así como de las personas que administran la información y el conocimiento de las políticas comerciales.

Con bases y aspiraciones superiores, desde los principios de la sociedad cubana, la Cámara de Comercio de la República de Cuba tiene como Misión (1):

- Promover el desarrollo de la empresa cubana asociada, en beneficio de la economía nacional.
- En el marco del avance que se va obteniendo en el desarrollo de la economía del país, se hace más importante el crecimiento de los intercambios, así como ambientes y resultados ventajosos para todas las partes que en ellos intervienen. Como espacio legal, la Cámara ha de convertir en realidad las aspiraciones de Cuba de llegar a ser una potencia en el área, a través del incremento del comercio, el turismo, y las inversiones.

La Cámara de Comercio desarrolla numerosos procesos relacionados con la actividad comercial de cualquier institución que genere actividad económica en el país. Entre otras instituciones de este tipo se tienen a las agencias de viajes nacionales, las sucursales de agencias de viajes extranjeras, así como empresas importadoras y exportadoras; las mismas tienen la obligación de registrarse ante la Cámara de Comercio.

Teniendo en cuenta la importancia que tienen las actividades que realiza la Cámara de Comercio para el país, tanto en el ámbito nacional como internacional, se hace necesario que la realización de los procesos de esta entidad deban ser objeto de informatización en aras de mejorar la gestión que realiza dicha institución. Pero la realidad es que tomando en consideración el avance de las nuevas tecnologías de la informática y las comunicaciones, la forma en que se realizan los procesos registrales de las agencias de viajes e importadores y

exportadores no es el más idóneo. La información que se maneja en la realización de estos procesos es generada y almacenada en formato duro o manualmente, agrupando una numerosa cantidad de expedientes y demás documentos que presentan dificultades respecto a su accesibilidad, gestión, fiabilidad, redundancia, actualización, deterioro, inconsistencia y otros. Informalmente se llevan algunos registros utilizando herramientas no destinadas a esta actividad y que no pueden manejar grandes volúmenes de datos tales como Microsoft Excel. Todos los documentos oficiales que son generados en los procesos anteriormente mencionados, como las Licencias de inscripción de las Agencias de Viajes Nacionales, son escritos e impresos usando Microsoft Word. Por lo anteriormente descrito se presentan numerosas dificultades en el seguimiento y actualización de los datos que constituyen operaciones fundamentales dentro de las actividades que realiza la institución.

Es por ello que se hace necesario disponer de un sistema informatizado que permita gestionar toda la información referente a los procesos registrales de las agencias de viajes nacionales, sucursales de agencias de viajes extranjeras e importadores y exportadores. De esta manera se logrará gestionar de forma rápida, estandarizada y centralizada toda la información procesada durante la ejecución de los mismos.

Con los elementos mencionados anteriormente se define el **problema** de la presente investigación: la forma en que se registran las agencias de viajes y las empresas importadoras y exportadoras en la Cámara de Comercio, no permite gestionar la información de las mismas de manera que se encuentre centralizada, estandarizada y accesible, realizar búsquedas rápidas y generar reportes.

El **objeto de estudio** será el Proceso de Desarrollo de Software y el **objetivo general**: desarrollar los procesos de Registro de Agencias de Viajes y Registro de Empresas Importadoras y Exportadoras del proyecto Sistema de Información Registral de la Cámara de Comercio, que permita gestionar la información de los mismos de manera centralizada, estandarizada y accesible, realizar búsquedas rápidas y generar reportes. El **campo de acción** estará determinado por la ingeniería de requisitos, diseño, implementación y pruebas.

En la presente investigación se sustenta la siguiente **idea a defender**: con el desarrollo de los procesos de Registro de Agencias de Viajes y Registro de Empresas Importadoras y Exportadoras del proyecto Sistema de Información Registral de la Cámara de Comercio, se logrará gestionar la información de los mismos de manera centralizada, estandarizada y accesible, realizar búsquedas rápidas y generar reportes.

Los **objetivos específicos** de la investigación son:

- Elaborar el marco teórico de la investigación.
- Desarrollar los requisitos de los procesos de Registro de Agencias de Viajes y Registro de Empresas Importadoras y Exportadoras.
- Implementar los requisitos obtenidos.
- Verificar y validar la solución propuesta.

Para garantizar el cumplimiento de los objetivos propuestos se llevarán a cabo las siguientes **tareas de la investigación**:

- Estudio de sistemas existentes para resolver problemáticas similares en países foráneos.
- Dominio y aplicación de las metodologías, herramientas y lenguajes más utilizados a nivel mundial para el desarrollo de este tipo de aplicaciones.
- Identificación y modelación de los procesos de Registro de Agencias de Viajes y Registro de Empresas Importadoras y Exportadoras.
- Descripción del flujo de los procesos de Registro de Agencias de Viajes y Registro de Empresas Importadoras y Exportadoras.
- Identificación, análisis, especificación y validación de los requisitos de los procesos de Registro de Agencias de Viajes y Registro de Empresas Importadoras y Exportadoras.
- Diseño de las clases de los procesos de Registro de Agencias de Viajes y Registro de Empresas Importadoras y Exportadoras.
- Implementación de los procesos de Registro de Agencias de Viajes y Registro de Empresas Importadoras y Exportadoras.
- Pruebas de calidad de software a través de métricas y diseños de casos de prueba basados en requisitos.

Los **métodos científicos** a utilizar en la investigación son los siguientes:

### **Teóricos**

- *Análisis Histórico – Lógico*: Es utilizado para el estudio del desarrollo que han tenido los procesos Registro de Agencias de Viajes y Registro de Empresas Importadoras y Exportadoras, realizando un análisis sobre la información recuperada para llegar a conclusiones formales.

- *Analítico – Sintético*: Utilizado para analizar la información existente, determinar los aspectos esenciales de la misma y sintetizar dichos aspectos en elementos relevantes de utilidad para la investigación.
- *Modelación*: El modelado es usado en el Proceso de Desarrollo de Software con la intención de abstraer la descripción o el funcionamiento de los sistemas en aras de comprender mejor estos y guiar otras actividades haciendo uso de los modelos.

### **Empíricos**

- *Entrevistas*: Se realizan a los clientes con la finalidad de recoger información de utilidad acerca de los procesos Registro de Agencias de Viajes y Registro de Empresas Importadoras y Exportadoras de la Cámara de Comercio.
- *Medición*: Se evidencia en la aplicación de métricas de calidad y la realización de pruebas que aseguren la calidad de los requisitos y artefactos generados en el proceso de desarrollo de software.

La investigación está estructurada en tres capítulos. El contenido de los mismos se describe a continuación:

**Capítulo 1. Fundamentación Teórica:** Se abordan los elementos teóricos y de carácter técnico relacionados con la implementación de la solución. Se seleccionan, de acuerdo a criterios establecidos en el proyecto, diferentes herramientas y tecnologías que facilitarán las actividades de implementación durante el proceso de desarrollo. Las mismas cubren todas las áreas del desarrollo tales como: metodología de desarrollo, lenguajes de modelado, gestor de procesos, herramientas CASE, lenguaje de programación, servidor de aplicaciones, entorno de desarrollo, marcos de trabajo, servidor de base de datos y otros.

**Capítulo 2. Solución Propuesta:** Se definen los elementos que plantea la metodología de desarrollo para la implementación del sistema. Se muestran y describen los diagramas de procesos, requisitos funcionales y no funcionales, actores del sistema, diagrama de casos de uso, diagrama de clases del diseño y diagrama de secuencia.

**Capítulo 3. Implementación y Prueba:** Se muestra el diagrama de componentes del sistema y se exponen los resultados obtenidos a partir de las técnicas de validación de modelos de procesos y requisitos, así como las métricas para evaluar la calidad de los artefactos generados.

## **CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA**

En el presente capítulo se realiza un estudio de los aspectos a tener en cuenta en el desarrollo del Sistema de Información Registral de la Cámara de Comercio. La información obtenida resume el estado en que se encuentran los diferentes elementos teóricos y técnicos necesarios para definir los criterios de selección, sistemas, metodologías, lenguajes de programación y modelado así como herramientas y tecnologías que serán usados en la implementación. Los elementos mencionados anteriormente son ilustrados de manera general y se especifican cuáles son las características que tributan específicamente al desarrollo del sistema en cuestión.

A continuación se muestra el estudio de aplicaciones similares en países foráneos y las consideraciones de su utilización o no en la implementación de la aplicación. Se definen las pautas a seguir en la selección de las herramientas y tecnologías y la forma en que serán usadas. Se introduce el enfoque a procesos como uno de los aspectos fundamentales en el desarrollo del proyecto, para ello, se resume la información de elementos relacionados tales como la metodología para la gestión de procesos de negocio, notación del modelado de procesos y gestores de procesos de negocio.

### **1.1 Sistemas para resolver problemáticas similares en otros países**

La red cameral española se compone de 85 cámaras de comercio brindando servicios al público en 138 puntos del país. Estas forman parte de la red de cámaras europeas denominada “Eurocámara” a la que están integradas más de 1200 cámaras con aproximadamente 14 millones de empresas asociadas. Dentro de las tres líneas trazadas en el ejercicio de la actividad de dichas instituciones se encuentra la información y asesoramiento que es ejecutada a través de una red de servicios telemáticos denominada Carmerdata. Las empresas pueden obtener y acceder en línea a bases de datos sobre las empresas españolas. Existe el sistema Cámara Net a través del cual las cámaras y las empresas pueden conectarse a través de Internet para exponer sus productos al mundo. También se ofrecen servicios de asesoría jurídica, centro de documentación e información y asesoramiento sobre medio ambiente. Con esta finalidad se ha creado la Bolsa Subproductos, servicio informático para gestionar la compra de materiales reciclables. (2)

La red de Cámaras de Comercio de Colombia posee servicios en línea como el registro mercantil, línea de respuesta inmediata, así como chat y llamadas virtuales. Otros servicios brindados son la gestión de trámites, inscripciones y consulta de información (3). Han

incorporado el uso la firma digital garantizando la autenticidad, integridad y no repudio teniendo la misma validez jurídica y probatoria de un certificado físico impreso en papel de seguridad.

El portal de la Cámara de Comercio del municipio español Bilbao permite obtener información de corte empresarial relacionados con el comercio internacional (4). La página web del portal de la Cámara de Comercio de Madrid ofrece servicios de búsquedas sobre bases de datos relacionadas con el quehacer de las asociaciones empresariales y otras (2). En Ecuador, Quito, el portal de la Cámara de Comercio brinda algunos servicios tales como guía comercial, incremento del capital, chequeo del estado de cuentas, gestión de formularios de débitos así como una guía electrónica (5). La Cámara de Comercio de Lima desarrolló un sistema basado en una plataforma de servicios integrados, los cuales permiten a las empresas hacer negocios a nivel local e internacional, apoyadas con herramientas de primer nivel y a bajo costo. Se trata de más de 30 servicios implementados en una plataforma web ([www.e-camara.net](http://www.e-camara.net)). (6)

La Cámara de Comercio de la República de Cuba requiere de un sistema informático que permita gestionar la información de todas las áreas de la misma, ajustado a las regulaciones establecidas legalmente en el país para el desarrollo de sus actividades. Debido a las limitaciones existentes en el país y en la entidad respecto al uso de Internet, se hace imposible ofrecer una gama completa de servicios en línea siguiendo la práctica que se realiza a nivel internacional, por lo que la solución a desarrollar será solo para uso interno de la entidad en la gestión de sus procesos.

Después de haber indagado acerca de los sistemas para resolver problemáticas similares en países foráneos, se tiene que actualmente a nivel internacional existen sistemas informatizados para la gestión de los procesos de las Cámaras de Comercio que no cumplen con los requerimientos y limitaciones de la entidad cubana. En la mayoría de los casos analizados dichas instituciones ofrecen sus servicios a través de portales web, de manera informativa fundamentalmente, o constituyen sistemas usados en redes internas los cuales no son accesibles ni de código abierto.

## **1.2 Criterios de selección de herramientas y tecnologías**

En la implementación de la solución se hará uso de diferentes herramientas y tecnologías que guiarán y facilitarán el proceso de desarrollo. Para la selección de las mismas se consideró por el equipo de desarrollo establecer diferentes criterios que debían cumplir. A continuación se resumen brevemente dichos criterios de selección.

### *Mantenibilidad*

El IEEE (1990) define mantenibilidad como: “La facilidad con la que un sistema o componente software puede ser modificado para corregir fallos, mejorar su funcionamiento u otros atributos o adaptarse a cambios en el entorno”. El proceso de desarrollo de un software en muchas ocasiones acarrea la aparición de errores en la captura de requisitos o cambios de los mismos por parte del cliente debido a cambios en los procesos que realiza la organización. Es importante por consiguiente que el sistema permita dentro de lo posible reducir el costo de mantenimiento del mismo debido a modificaciones que se tengan que realizar. (7)

### *Independencia tecnológica*

La política en el país y la universidad es el uso de herramientas que garanticen la independencia tecnológica. El uso de herramientas propias o de código abierto que permitan usarlas, estudiarlas, compartirlas y modificarlas es uno de los principios fundamentales a seguir en aras del cumplimiento de este objetivo.

### *Portabilidad*

Se define como: “La capacidad del software para ser trasladado de un entorno a otro. El entorno puede incluir entornos organizacionales, de hardware o de software”. Una de las dificultades que se presenta a menudo es la plataforma hacia la que está destinado el software, ya que los mismos requieren de implementaciones específicas para cada una de ellas. Se persigue entonces que las herramientas, tecnologías y la solución sean independientes de la plataforma. (8)

### *Documentación*

Bajo este término genérico se agrupan todos los manuales, guías de referencia, libros de ayuda, etcétera, que suelen entregarse con cada programa, de manera que el usuario pueda aprender su manejo y consultar cualquier duda ante un problema desconocido. Se persigue que todos los elementos que sean usados en la implementación de la solución estén formal y correctamente documentados y que se puedan solucionar los problemas que puedan aparecer en el proceso de desarrollo. (9)

### *Enfoque a procesos*

El enfoque a procesos es un camino para organizar y gestionar las actividades que crean valor en la empresa. Este permite obtener eficientemente un resultado cuando las actividades y los recursos se gestionan como un proceso, es decir, un conjunto de actividades mutuamente relacionadas o que interactúan, las cuales transforman elementos de entrada en resultados (10). Las empresas están organizadas como áreas dentro de una jerarquía funcional. El enfoque de procesos elimina las barreras entre diferentes áreas funcionales y unifica sus enfoques hacia las metas principales de la organización. (11)

Las organizaciones tienen la visión de usar las tecnologías de la información para informatizar sus procesos, sin embargo, este objetivo no es tan fácil de lograr. Se depende cada vez más de sistemas informáticos utilizados en la consecución de sus metas, pero dichos sistemas son usados independientemente o de forma separada estableciendo dependencias complejas entre los mismos.

Se hace necesario por consiguiente que los procesos sean definidos y gestionados por las personas que tienen el conocimiento sobre ello. La informatización de dichos procesos debe ser realizada sobre la base de esa definición de acuerdo a los cambios de la tecnología y minimizando el impacto de los cambios en los procesos en su implementación (12). Dicha implementación, de acuerdo a este enfoque será más económica ya que aunque un proceso afecte a otros, es mucho más sencillo cambiar o mejorar el proceso o partes del mismo sin afectar al resto (13).

Este último aspecto es muy importante ya que actualmente existe mucha dinámica en el área empresarial. Las empresas deben cambiar y/o mejorar sus procesos con el objetivo de tener un lugar en la competencia o brindar mejores servicios al cliente. Comúnmente en el desarrollo de software se identifican cuáles son las actividades de un determinado proceso que serán objeto de informatización, el resultado es un software que informatiza partes del proceso de negocio y no el proceso en sí. Por consiguiente, al cambiar la definición del proceso de negocio el software queda obsoleto o es necesario realizar una actualización que conlleva gastos y esfuerzo adicional.

Debido a las razones planteadas anteriormente es objetivo del proyecto emplear un enfoque a procesos durante todo el proceso de desarrollo quedando evidenciado este en la solución final.

#### 1.4 Notación de Modelado de Procesos de Negocio

Se selecciona como lenguaje para el modelado de los procesos la Notación de Modelado de Procesos de Negocio (Business Process Modeling Notation, BPMN) en su versión 2.0. El objetivo de la notación del modelado de procesos del negocio es muy similar a lo que se intenta conseguir con el Lenguaje Unificado de Modelado (Unified Modeling Language, UML) para el diseño y análisis orientado a objetos. La idea es identificar las mejores prácticas de los enfoques existentes para combinarlos y obtener como resultado un nuevo lenguaje. En los documentos del estándar se plantea que:

*“El principal objetivo de BPMN es proveer una notación que es fácilmente entendible por todos los usuarios del negocio, desde los analistas de negocios que crean los borradores iniciales de los procesos, hasta los desarrolladores responsables de la implementación y de la tecnología que permitirá la ejecución de estos procesos, y finalmente, las personas del negocio que administrarán y monitorearán estos procesos.” (14)*

Los modelos de procesos de negocio son expresados en diagramas de procesos de negocio. Cada uno de estos diagramas consiste en un conjunto de elementos de modelado. Los principales elementos de esta notación son: eventos, actividades, elementos de decisión y de secuencia en el flujo. La notación gráfica se complementa con atributos que pueden ser asociados a elementos particulares del diagrama o al diagrama del proceso del negocio en su totalidad. (14)

#### 1.5 Metodología de desarrollo de software

En el ciclo de vida del software se deben completar una serie de tareas para obtener un producto de software. Los distintos componentes de software deben pasar por distintas fases o etapas durante el ciclo de vida. Las metodologías proponen un proceso disciplinado sobre el desarrollo de software con el objetivo de hacerlo más predecible y eficiente. Ofrecen un conjunto de técnicas, herramientas, procedimientos y soporte documentales que trazan las pautas para asegurar la eficiencia y calidad en el desarrollo del producto. Lo hacen desarrollando un proceso detallado con un fuerte énfasis en la planificación (15). La elección de una metodología de desarrollo de software para un proyecto es un asunto de vital importancia, pues se deben tener en cuenta las características del proyecto en cuestión para garantizar que la selección de la misma se adecue con las necesidades reales para el desarrollo y para la satisfacción de los clientes.

## 1.6 Proceso Unificado de Desarrollo de Software

Se escogió por parte del equipo de desarrollo del proyecto el Proceso Unificado de Desarrollo de Software (Rational Unified Process, RUP). Dicha selección está sustentada en el hecho de que esta metodología es una de las más reconocidas a nivel internacional, el equipo de desarrollo tiene un amplio conocimiento sobre la misma y los miembros están familiarizados con los artefactos y actividades que propone para el desarrollo. Cabe destacar que existen diferentes herramientas CASE que la soportan ya que la misma hace uso de UML. Un aspecto muy importante a tener en cuenta es el proceso de mejoras que se lleva a cabo en la universidad. Este plantea diferentes pautas en la realización de los procesos de la institución. RUP es perfectamente ajustable al mismo por lo que para el proceso de desarrollo de software se utilizarán los flujos de trabajo de Requisitos, Diseño, Implementación y Pruebas. Para el modelado del negocio se utilizará la notación BPMN y en el flujo de Implementación se hará uso de la misma como parte del diseño de los procesos en el gestor de procesos seleccionado.

RUP es un proceso de desarrollo de software y como tal define un conjunto de actividades que son necesarias para transformar los requisitos del usuario en un software. Sin embargo, no sólo contempla los aspectos relacionados particularmente al proceso de desarrollo sino que constituye un marco de trabajo genérico que puede especializarse para ser adaptado a sistemas de software, diferentes áreas de aplicación, diferentes tipos de organizaciones y diferentes tamaños de proyecto. Es un proceso que define “quién” está haciendo “qué”, “cuándo” y “cómo” para alcanzar un determinado objetivo. (16)

Dicha metodología está basada en componentes, por lo que el sistema quedará estructurado en diferentes componentes de software que se interrelacionan a través de interfaces. Lo anterior es reconocido como una buena práctica en la implementación de un sistema añadiendo aspectos como la reusabilidad y la seguridad entre otros (16). RUP posee algunas características de referencia para su estudio:

- *Dirigido por casos de uso:* La interacción entre un usuario y un sistema se puede resumir en una acción por parte del usuario y una secuencia de acciones por parte del sistema para ofrecer al primero un resultado importante. Los casos de uso representan requisitos funcionales y se agrupan en modelos de casos de uso. Estos guían el proceso de desarrollo al dirigir su diseño, implementación y prueba. Basado en los modelos de casos de uso se crean otros modelos de diseño e implementación que son

probados a partir de la descripción los primeros. En otras palabras los casos de uso inician el proceso de desarrollo y le proporcionan un hilo conductor.

- *Centrado en la arquitectura:* La arquitectura de un sistema de software describe a través de diferentes vistas el sistema que se quiere implementar. Incluye los aspectos estáticos y dinámicos más significativos de un sistema que es reflejada en los casos de uso. Es una vista del diseño completo resaltando las características principales solamente.
- *Iterativo e incremental:* RUP es iterativo e incremental en la medida en que se subdivide el trabajo en partes más pequeñas que constituyen iteraciones o pasos en el flujo de trabajo, e incremental porque cada iteración aporta algo que se traduce en el crecimiento o desarrollo del producto final. En cada iteración se identifican y especifican casos de uso relevantes basándose en la arquitectura seleccionada, se implementa el diseño en componentes y se verifica que estos últimos satisfacen los casos de uso.

RUP divide el proceso de desarrollo de software en fases, en las cuales se realizan varias iteraciones en dependencia de proyecto y se ejecutan con mayor o menor carga de trabajo las diferentes actividades de los flujos de trabajo o disciplinas que se describirán brevemente más adelante. (17)

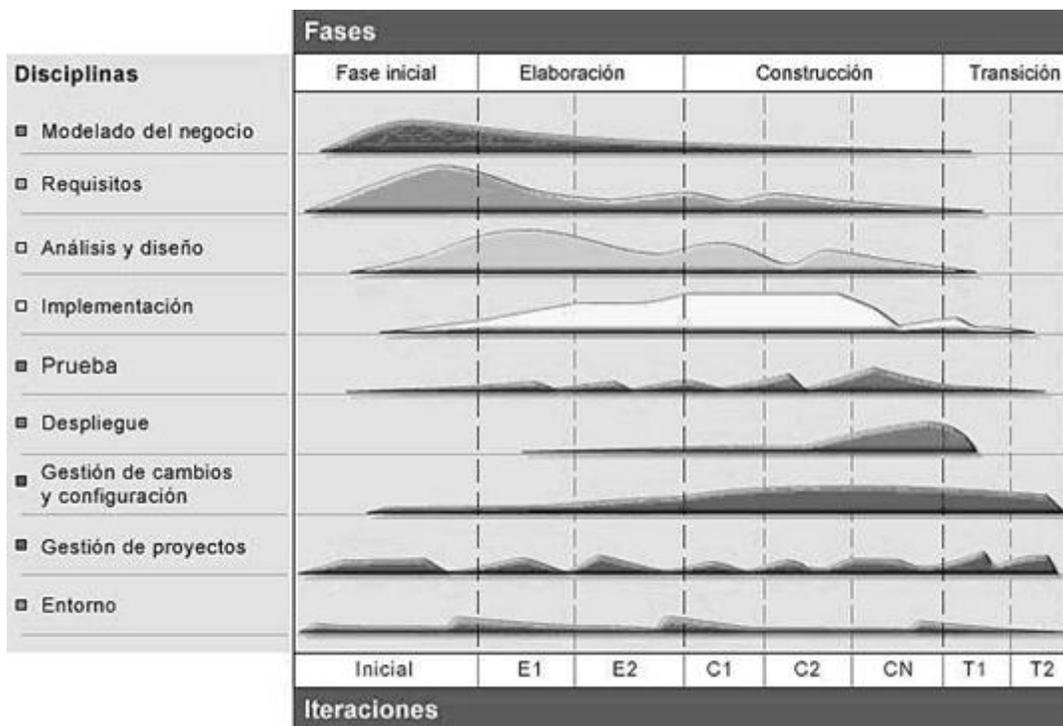


Figura 1: Fases, iteraciones y disciplinas de RUP. (15)

- *Inicio*: En la fase de inicio se define el objetivo, el ámbito del proyecto y sus límites, si es factible o no la implementación del sistema, casos de usos críticos, arquitectura candidata, estimación de riesgos, etcétera. Como productos de esta fase se tiene el documento visión del negocio, modelo de casos de uso, requisitos no funcionales, glosario de términos, lista de riesgos y planes de contingencia, prototipos así como el plan de iteración de la fase de elaboración.
- *Elaboración*: El objetivo de esta fase es analizar el dominio del problema, establecer los cimientos de la arquitectura y desarrollar el plan de proyecto. Se define una arquitectura basada en los casos de uso críticos identificados y se elabora un plan para la siguiente fase. Esta fase es de gran importancia pues cuando termina se llega a un punto denominado línea base de la arquitectura donde un retroceso en el proceso es significativamente costoso, además de que se da paso a la siguiente fase que es costosa y arriesgada.
- *Construcción*: El objetivo principal de esta fase es obtener un producto completamente funcional resultado de las iteraciones realizadas en la misma. Todos los componentes, características y requisitos son implementados, integrados y probados en aras de conseguir una calidad adecuada en el producto.
- *Transición*: En esta fase se pone el producto en manos de los usuarios finales con sus respectivas y sucesivas actualizaciones, se completa la documentación, y se entrena al usuario en el uso, ajuste, configuración, instalación y usabilidad del producto.

A través de las fases de RUP se desarrollan en mayor o menor medida las actividades definidas en los nueve flujos de trabajo de la metodología. Los mismos están separados en dos grupos: los de ingeniería (Modelado del negocio, Requisitos, Análisis y diseño, Implementación, Prueba y Entrega) y los de apoyo (Administración de proyecto, Configuración de Cambios y Administración y Ambiente).

- *Modelado del negocio*: Se describen los procesos del negocio, las actividades que se requieren informatización y quiénes las realizan.
- *Requisitos*: Se definen las funcionalidades que tendrá el sistema así como las restricciones del mismo.
- *Análisis y diseño*: A partir de los requerimientos se define cómo será implementado el sistema.
- *Implementación*: Define como se organizan las clases y objetos en componentes y subsistemas de implementación.

- *Prueba*: Busca defectos en todos los artefactos generados así como en la solución final a lo largo de todo el proceso de desarrollo.
- *Despliegue*: Se realizan liberaciones del producto y otras actividades para hacer la entrega del software a los usuarios.
- *Gestión de cambios y configuración*: Entre otras actividades, se establecen las pautas para controlar el versionado de los artefactos generados por el equipo de desarrollo.
- *Gestión de proyectos*: Relacionado con las acciones a emprender en aras de lograr la satisfacción del cliente.
- *Entorno*: Relacionado con los procesos y herramientas que involucran y serán usadas respectivamente a los usuarios y por los mismos.

### **1.6.1 Lenguaje Unificado de Modelado**

Para realizar los diferentes diagramas propuestos en la metodología seleccionada para guiar el proceso de desarrollo se utilizará el estándar para el modelado de software UML. El mismo se utiliza en la visualización, especialización, construcción y documentación de los artefactos de sistemas de software. Es usado para entender, diseñar, examinar, configurar, mantener y controlar la información de dichos sistemas. El mismo incluye conceptos semánticos asociados a las representaciones gráficas, la notación como tal y guías de uso. Uno de sus objetivos es ser soportado por herramientas de modelado que poseen además generadores de código. Está diseñado para que pueda usarse en la mayoría de los procesos de desarrollo orientados a objetos. Este lenguaje de modelado incluye además elementos que permiten organizar el modelo, dividiéndolo en paquetes permitiendo particionar grandes sistemas en pequeñas partes funcionales con sus respectivas dependencias. (18)

### **1.7 Gestor de procesos del negocio**

Los gestores de procesos del negocio transforman elementos de entrada en elementos de salida haciendo uso de un conjunto de recursos y actividades interrelacionadas. Soportan estructuras orientadas a procesos y flujos empresariales, poseen una gestión de cambio bastante rápida, cuentan con diferentes estándares para la generación de reportes que permiten o facilitan la toma de decisiones a nivel empresarial, y automatizan el ciclo de vida de los procesos. (19)

### 1.7.1 Bonita Open Solution

Como se había mencionado anteriormente el sistema a desarrollar estará caracterizado por un enfoque a procesos, por lo que para la implementación se decidió utilizar el gestor de procesos de negocio Bonita Open Solution (BOS) en su versión 5.6. El mismo ofrece ventajas significativas en el ámbito del proyecto:

- *Cambios en los procesos:* Los procesos de las organizaciones cambian continuamente con el objetivo de mejorar los mismos o por decisiones a nivel legal, institucional o gubernamental. BOS permite modificar el flujo de un proceso previamente diseñado con mucha facilidad, concretamente, modificando un diagrama de procesos en notación BPMN.
- *Generación de interfaces y manejo básico de datos:* El gestor de procesos seleccionado permite generar automáticamente las interfaces de usuario (particularmente los formularios de entrada de datos) y el flujo básico de movimiento entre las mismas. Los formularios de captura y recuperación de datos llevan también asociadas variables para la captura de los datos a nivel de actividades y a nivel de procesos. Las plantillas predeterminadas siempre pueden ser modificadas haciendo uso de HTML, CCS y Java Script.

BOS es una herramienta de código abierto, editor y solución de referencia de su tipo, multiplataforma y compatible con el estándar BPMN. Posee ciertas facilidades propias de un entorno de desarrollo ya que está desarrollado sobre la reconocida plataforma Eclipse. Su código puede ser descargado bajo la licencia GPL v2. (20)

BOS está compuesto por cuatro módulos (21):

1. *Bonita Execution Engine:* Es el motor de BPM de Bonita y se encarga del despliegue y ejecución de los procesos.
2. *Bonita Studio:* Es la aplicación gráfica que tiene como función el diseño de los procesos BPM usando la notación BPMN.
3. *Bonita Form Builder:* Destinado al diseño y configuración de los formularios de entrada o consultas de datos del sistema.
4. *Bonita User Experience (User XP):* Es un módulo para el usuario. Permite la ejecución e intervención en los procesos de negocio desplegados distinguiendo entre roles y tipos de usuarios. La interfaz se muestra en forma de correo electrónico donde en la bandeja

de entrada se tendrán los procesos suspendidos, respuestas a eventos y procesos que requieren la intervención del usuario.

Dentro del mundo del software libre BOS es el primer editor y solución BPM de referencia. Brinda diferentes ventajas para las diferentes partes involucradas o actores de proyectos BPM. Permite aprovechar en gran medida el estándar BPMN y diseñar relativa y fácilmente los proyectos así como desarrollar aplicaciones escribiendo poco o ningún código. Ofrece un conjunto de APIs sólidas para esta finalidad y permite la personalización de las interfaces web. Para el usuario final ofrece aplicaciones autónomas e intuitivas, visualizar fácilmente todas las tareas de un determinado proceso, gestión de la lista de tareas de todos los procesos, etcétera. Al ser software libre permite una reducción considerable en aspectos como el tiempo, el esfuerzo y el presupuesto destinado a la ejecución del proyecto. (20)

## 1.8 Herramientas CASE

La definición básica de una herramienta CASE plantea que es una herramienta de software que automatiza una parte del ciclo de desarrollo de software. (22)

Las herramientas CASE (*Computer Aided Software Engineering*) constituyen un conjunto de programas y ayudas que facilitan las tareas de los analistas, ingenieros de software y desarrolladores durante el ciclo de vida de desarrollo de un software. También CASE se define como un conjunto de utilidades y técnicas que facilitan parcial o completamente el desarrollo de sistemas de información. Otros la entienden como una filosofía de desarrollo de software que representa una innovación en la organización, una unión entre herramientas de software automáticas y las metodologías de desarrollo de software. (22)

El proceso de desarrollo de un software necesita la organización de las tareas de forma eficiente para su correcto completamiento. Las herramientas CASE tienen como función la automatización de este proceso permitiendo a los analistas emplear más tiempo en el análisis y el diseño minimizando el tiempo para codificar y realizar pruebas.

### 1.8.1 Visual Paradigm for UML

La herramienta CASE seleccionada para el desarrollo del proyecto es Visual Paradigm for UML 8.0. Esta selección se realizó teniendo en cuenta las características que posee que deben ir en correspondencia con los criterios de selección especificados anteriormente. La misma soporta la metodología de desarrollo de software seleccionada, pudiéndose por consiguiente, generar todos los diagramas y esquemas que define la misma en notación UML y BPMN. Está

disponible para la plataforma Linux y se integra con el entorno de desarrollo que será usado en la fase de implementación, cubre todas las fases del ciclo de vida de desarrollo del software, tiene soporte para aplicaciones web, permite la generación de código en el lenguaje Java, generación de base de datos entre otras funcionalidades. Es una herramienta CASE de diseño orientada a facilitar el desarrollo de software. Ofrece una colección de herramientas de desarrollo de software para la captura de requisitos, planeación de software, pruebas, modelamiento de clases, modelado de datos y otros (23).

Posee características gráficas muy cómodas que facilitan la realización de diagramas de modelado basado en el estándar UML tales como los diagramas de clases, casos de uso, comunicación, secuencia, estado, actividades, componentes, etcétera. Está disponible en varias ediciones, cada una destinada a necesidades específicas: Enterprise, Professional, Community, Standard, Modeler y Personal. La CE o CommunityEdition es gratuita aunque presenta algunas limitaciones respecto a las versiones propietarias. (24) (25)

La versión abierta CE es perfectamente integrable con el entorno de desarrollo seleccionado para el proyecto (Netbeans) permitiendo el diseño en UML, diseño de bases de datos, generación de código, generación de base de datos y generación de la documentación. (26)

## **1.9 Lenguajes de programación**

En computación, un programa es una secuencia de instrucciones que permiten a un ordenador procesar una información conocida como datos de entrada para producir una información de salida o resultados (27). Estas instrucciones se encuentran escritas en un lenguaje particular de programación. Los lenguajes de programación se pueden clasificar como de bajo nivel y de alto nivel. Entre los lenguajes de alto nivel, que son los que interesan en función de darle solución al problema se encuentran C, C++, C#, Java y otros.

### **1.9.1 El lenguaje de programación Java**

Después de un análisis de los diferentes lenguajes de programación que se pudieran utilizar en la solución, se llegó a la conclusión de que Java era la mejor opción, pues además de las propias características que posee el lenguaje, otras herramientas y tecnologías seleccionadas para la ejecución del proyecto están en correspondencia con dicho lenguaje o se integran perfectamente con este.

Java es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero

tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria. (28)

Algunas de las características del lenguaje son las siguientes (27):

- *Orientado a objetos*: Java ratifica el paradigma de la programación orientada a objetos (POO) desde el mismo punto en que ofrece numerosas bibliotecas de clases denominadas APIs Core Java que no son más que una colección de componentes.
- *Interpretado*: El código escrito en Java es primeramente compilado a un lenguaje intermedio denominado “bytecode”, que luego es interpretado por una máquina virtual (Java Virtual Machine, JVM) que sirve como una abstracción entre la máquina y el lenguaje permitiendo que se pueda ejecutar la aplicación en cualquier arquitectura.

### 1.10 La plataforma empresarial de Java

Las plataformas Java existen en varias versiones (29):

- *JSE (Java Standard Edition)*: Para el desarrollo de aplicaciones de escritorio y applets de Java. Es la base de la JEE y tecnologías Java Web Services. Posee el compilador, herramientas, módulos de ejecución y la API de Java que le permiten escribir, probar, implementar y ejecutar applets y aplicaciones. (30)
- *JEE (Java Enterprise Edition)*: Para el desarrollo de aplicaciones multicapa sobre la web del lado del servidor. Define el estándar para el desarrollo basado en componentes aplicaciones empresariales. Está basada en la Java SE y otros servicios adicionales. (30)
- *JME (Java MicroEdition)*: Conjunto de tecnologías y especificaciones dirigidas a los consumidores y dispositivos embebidos, como teléfonos móviles, asistentes personales digitales (PDA), impresoras y TV. Esta no es más que una versión simplificada de la Java SE y la inclusión de algunas APIs específicas para este tipo de aplicaciones. (30)

JEE es un conjunto de especificaciones de APIs de Java que describe el conjunto de paquetes, interfaces y clases que deben estar presentes en un marco de trabajo JEE. Este conjunto de servicios, APIs y protocolos brindan las funcionalidades requeridas en el desarrollo de aplicaciones web centrándose fundamentalmente en escribir la lógica empresarial. Actualmente existen implementaciones propietarias (Oracle WebLogic Server, Sun Application Server, Oracle Application Server, IBM WebSphere Application Server) y libres (JBoss, GlashFish, Tomcat + OpenEJB). (31) (32)

La plataforma define un estándar para el desarrollo de aplicaciones portables, robustas, escalables y seguras usando tecnología Java en el lado del servidor. Establece una infraestructura común básica para el acceso a bases de datos, gestión de la persistencia, control de seguridad, gestión de transacciones, etcétera. Muestra una separación lo suficientemente clara entre la presentación (interfaz), el modelo (lógica) del negocio y los datos. (33) (34) (35)

### 1.10.1 APIs JEE

Al estar Java EE basado en Java SE todas las APIs de esta última versión también están disponibles en la JEE, añadiéndose otras que son necesarias específicamente para el desarrollo de sistemas distribuidos multicapas las cuales proveen funcionalidades para publicar, distribuir, clusterizar y permitir la tolerancia a fallos (35). Algunas APIs como JDBC y Java Persistence API (JPA) han sido especificadas finalmente en la Java SE (33). A continuación se describen algunas de las APIs más relevantes de la JEE.

#### *Enterprise JavaBeans*

El sistema de componentes JavaBean es una infraestructura portátil e independiente de la plataforma que utiliza el lenguaje de programación Java. Este sistema acopla los componentes de software más sofisticados para el desarrollo basado en componentes. El mismo permite a los desarrolladores (15):

- Analizar el funcionamiento de los Beans (componentes).
- Personalizar su comportamiento y aspecto.
- Establecer mecanismos de comunicación y coordinación.
- Desarrollar Beans personalizados para su utilización en una aplicación en específico.
- Probar y evaluar el comportamiento de un Bean.

Los EJB constituyen los componentes básicos para el manejo del encapsulamiento de la lógica del negocio en la plataforma JEE. Estos facilitan la portabilidad, escalabilidad y disponibilidad brindando accesos para obtener y gestionar componentes que manejan los datos y las reglas del negocio. Proveen un modelo simplificado de programación para objetos distribuidos. (36)

Para la ejecución de estos componentes se requiere el contenedor “EJB Container” que normalmente viene incluido en la mayoría de los servidores de aplicaciones que se pueden encontrar actualmente (35). El mismo implementa las interfaces de la JEE que permiten el

manejo de sesiones, la persistencia, la seguridad, las transacciones, el “multi-threading”<sup>1</sup> y los accesos concurrentes (36).

#### *Java Servlets, Java Server Pages y JSP Standar Tag Libraries*

Constituyen los elementos usados en la capa de presentación (interfaz gráfica) en la web de la plataforma JEE, surgieron como una respuesta al estándar CGI (Common Gateway Interface) (35).

#### *Java Servlets*

Un Java Servlet se ejecuta en un servidor web y que construye una página web que es devuelta al usuario. Esta página, construida dinámicamente, puede contener información procedente de bases de datos, ser una respuesta a los datos introducidos por el usuario, etcétera. (37)

Los servlets son clases Java que manejan muchos de los protocolos presentes en Internet como son HTTP, SMTP, FTP, POP y otros a través de la implementación de interfaces específicas para esta finalidad. Para su ejecución requieren un contenedor denominado “Servlet and JSP Container” que posee acceso a todos los recursos de red que se encuentren disponibles. (35)

Ofrecen una infraestructura de desarrollo de aplicaciones web ya que brindan, entre otras, funciones de análisis automático y decodificación de los datos de los formularios de HTML, acceso a las cabeceras de las peticiones HTTP, manejo de cookies, seguimiento, control y gestión de sesiones. Pueden compartir los datos entre sí, conexiones a bases de datos, mantener información de solicitud en solicitud facilitando tareas como el seguimiento de las sesiones de usuario, etcétera. (37)

#### *Java Server Pages*

Las JSP son una tecnología que permite mezclar HTML estático con HTML generado dinámicamente mediante código Java embebido en las páginas (37). Fue adquiriendo paulatinamente algunas características como el acceso a recursos Java Beans, uso de bibliotecas de etiquetas XML (JSP Tag Libs), que luego se estandarizaron en JSTL (JSP Standar Tag Libraries). De la misma manera que los Java Servlets requieren de un contenedor

---

<sup>1</sup> Capacidad de ejecutar las diferentes partes de un programa (threads) simultáneamente.

“Servlet and JSP” para su funcionamiento. Las JSP se han difundido en gran medida por la facilidad de su uso. (35)

### Java Data Base Connection

JDBC es una API que permite el trabajo con bases de datos relacionales. Mediante esta el desarrollador puede realizar consultas, actualizaciones, inserciones y borrados, así como ejecutar las consultas bajo transacciones. (38)

A continuación se muestra el esquema general de la JEE donde se muestran las diferentes APIs de la plataforma distribuidas por áreas lógicas.

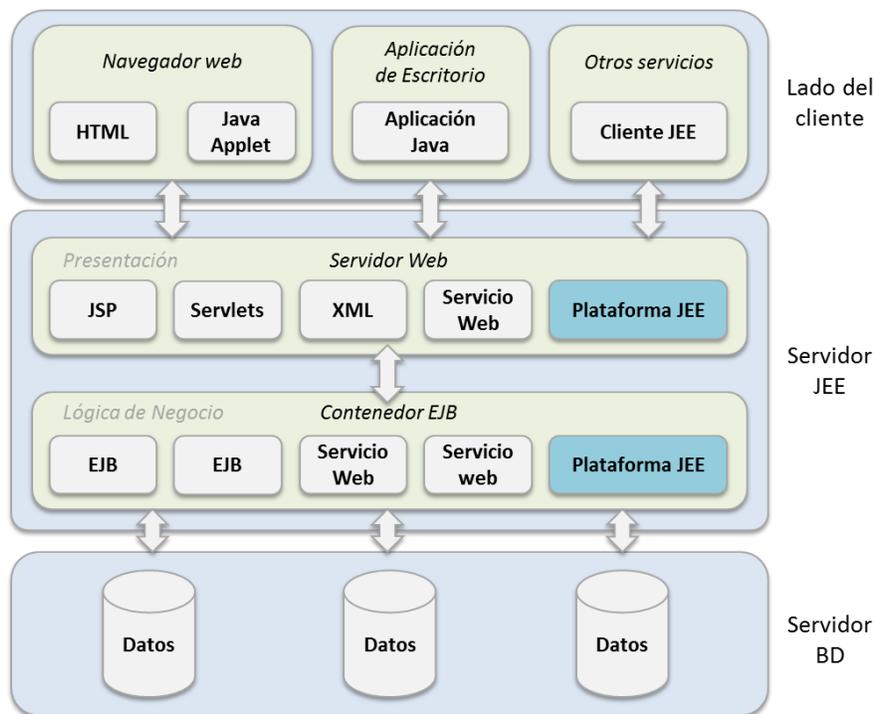


Figura 2: Esquema general de la JEE. (39)

### 1.11 Servidores de aplicaciones

Un servidor de aplicaciones es un servidor que en una red ejecuta determinadas aplicaciones, tales servidores suelen escribirse usando un lenguaje de programación tal como Java o C++ (15). Es un software que ofrece servicios a los dispositivos clientes fundamentalmente a través de Internet y usando el protocolo HTTP. La presencia de contenido dinámico y su integración con bases de datos son características que los distinguen de los servidores web. Manejan la mayoría de las transacciones relacionadas con la lógica del negocio y el acceso a datos. (40)

Los servidores de aplicaciones proporcionan la infraestructura y los servicios claves a las aplicaciones alojadas en un sistema, entre algunas de sus funciones encontramos: agrupación de recursos, administración de transacciones en un ambiente distribuido, comunicación asíncrona entre programas, interfaces de servicios web XML, detección de errores y seguridad. (41)

### **1.11.1 Servidor de aplicaciones Tomcat**

Tomcat es la implementación de referencia oficial para las especificaciones Servlet y JSP a partir de las versiones (2.2 y 1.1 respectivamente). Es un producto muy robusto, altamente eficiente y uno de los más potentes contenedores de Servlets existentes. Su único punto débil reside en la complejidad de su configuración, dado el gran número de opciones existentes. El mismo incluye un pequeño servidor HTTP para atender las peticiones estáticas y redirigir el resto al motor de ejecución de Servlets. (37)

Actualmente Tomcat puede funcionar como servidor de aplicaciones por sí solo pero también puede especificarse como el manejador de las peticiones de JSP y servlets recibidas por servidores Web populares, como el servidor HTTP Apache de la Fundación Apache. Es una solución de servidor sin coste de licencia, de fácil mantenimiento y que permite el establecimiento de clústeres fácilmente aprovechando el potencial del servidor HTTP Apache. Actualmente es el servidor más utilizado en Internet para el despliegue de aplicaciones web. De forma nativa no soporta EJBs ni Web Services, pero esto se puede lograr con su integración con otras soluciones que brindan estos servicios. El módulo que da soporte para la ejecución dinámica de código Java es Catalina, que se inicializa en el arranque y Jasper es el módulo que permite procesar e interpretar ficheros JSP. (42)

## **1.12 Entornos de desarrollo**

Un entorno de desarrollo integrado (Integrated Development Environment, IDE) es una aplicación compuesta por diferentes herramientas que son usadas por los desarrolladores de software para facilitar las tareas del desarrollo. Puede ser destinado al trabajo con un lenguaje de programación en particular o también para varios de estos. Los elementos que comúnmente forman parte de un entorno de desarrollo son el editor de texto, el compilador o intérprete, depurador, diseñador de interfaz de usuario y pueden ser integrados con sistemas de control de versiones.

### 1.12.1 Netbeans

El equipo de desarrollo decidió emplear en la implementación el IDE de desarrollo Netbeans en su versión 7.1. Esta decisión estuvo basada en los criterios de selección establecidos para las herramientas y tecnologías. Netbeans es un proyecto de código abierto en constante crecimiento que cuenta con una amplia comunidad de usuarios y cerca de cien socios alrededor del mundo. Sun Microsystems fundó el proyecto de código abierto NetBeans en junio de 2000 y continúa siendo el patrocinador principal de los proyectos. (43)

Netbeans IDE es un entorno de desarrollo escrito en Java, pero se puede utilizar para desarrollar en otros lenguajes de programación. El mismo puede extender sus funcionalidades mediante la incorporación de módulos gracias a su arquitectura modular. Es un producto libre y gratuito sin restricciones de uso. (43)

En Netbeans el soporte para el lenguaje de programación es creado en forma de módulos sobre la plataforma como tal, esto permite, que pueda ser adaptado para necesidades específicas de los usuarios permitiendo que módulos que no son usados puedan ser desactivados o desinstalados. Es un entorno de desarrollo que posee un alto grado de robustez y valor para el usuario final, que se traduce en una considerable reducción en el tiempo de desarrollo, consistencia en la interfaz de usuario, independencia de la plataforma, reusabilidad y confiabilidad. (44)

### 1.13 Frameworks de desarrollo

Un framework se define como *“una estructura de soporte definida para proyectos de software en donde su principal actividad es organizar el desarrollo. Incluye soporte para programas, bibliotecas de clases, lenguajes de programación y otras aplicaciones que acompañan al desarrollo de alguna aplicación y proyectos complejos.”* (45)

Un framework representa una arquitectura de software que modela las relaciones generales, provee una estructura y una metodología de trabajo. Su objetivo es fundamentalmente facilitar en gran medida el desarrollo de software abstrayendo el bajo nivel permitiendo concentrar esfuerzos, hacer posible la reutilización de software y a su vez ganar tiempo en la identificación de los requerimientos de software. (45)

## 1.14 EclipseLink

EclipseLink es un proyecto de código abierto que constituye la implementación de referencia de Java Persistence API (JPA) (46). JPA fue especificada sobre muchos de los conceptos principales que han sido implementados y probados en el campo del mapeo objeto relacional (Object Relational Mapping, ORM) tales como TopLink e Hibernate (47). Es el resultado de la evolución de varios marcos de trabajo de Java que proveen una capa de acceso simple a bases de datos para POJOs (Plain Old Java Objects). (48)

El objetivo de JPA es la especificación de un modelo de persistencia estandarizado, simple y usable. Se distingue fundamentalmente por ser ligero. En JPA las consultas SQL son especificadas en forma de Java Persistence Query (JPQL), tanto de forma dinámica como de forma estática. Comprende tres aspectos diferentes: la API como tal, el lenguaje de consultas JPQL y las anotaciones para la definición relacional de la información. (44)

Esta API describe un mecanismo para estructurar los datos en términos de clases de Java. Puede ser usada para definir y aplicar esquemas de datos consistentes y/o servir o entenderse como una capa de modelado de datos. Ofrece una capa de portabilidad pues este estándar es soportado por muchas soluciones de almacenamiento de datos. En JPA se definen las clases como POJOs<sup>2</sup> y se usan anotaciones para especificar qué clases serán persistentes para su almacenamiento en una determinada base de datos. La definición de los datos en términos de clases de Java solamente facilita de forma considerable la manipulación de la persistencia de los datos, al mismo tiempo que facilita probar la aplicación usando mocks<sup>3</sup>. (49)

### 1.14.1 Spring

Spring es un marco de trabajo de código abierto creado por Rod Johnson. Fue creado con el objetivo de guiar el desarrollo de aplicaciones empresariales complejas. La utilidad de Spring no se limita solo al desarrollo del lado del servidor, cualquier aplicación Java puede recibir el beneficio de usar Spring en aras de la simplicidad, facilidad de probar las aplicaciones y el bajo acoplamiento. (50)

Con Spring se pueden automatizar un gran número de tareas pero cuando se analiza el marco en sí, descomponiéndolo en sus diferentes partes, las principales características de Spring son la inyección de dependencias (Dependency Injection, DI) y la programación orientada a

---

<sup>2</sup> Acrónimo de Plain Old Java Object. Objetos simples que no heredan ni implementan otras clases.

<sup>3</sup> Objeto de reemplazo u objeto simulado. Son objetos que imitan el comportamiento de objetos reales evitando usar estos últimos y su implementación. Usados en la realización de pruebas de unidad. (50)

aspectos (Aspect-oriented programming, AOP), lo cual convierte a Spring en un contenedor de estos elementos. A continuación se describen brevemente algunas de las características, patrones y definiciones relacionadas con el marco de trabajo. (50)

- *Inyección de dependencias*: Spring promueve el bajo acoplamiento a través de la técnica conocida como inyección de dependencias. Esto significa que los objetos “reciben” o “se le pasan” sus dependencias en vez de crearlas o buscarlas por ellos mismos. En vez del objeto buscar sus dependencias en un contenedor, el contenedor le pasa las dependencias en el momento de su instanciación.
- *Orientado a aspectos*: Posee un excelente soporte para programación orientada a aspectos que permite un desarrollo cohesionado separando la lógica del negocio de los servicios brindados por el sistema, como el manejo de transacciones por ejemplo. Con esto se logra que los objetos de la aplicación realicen solo las funciones que deberían realizar la lógica del negocio y nada más, eliminando dentro de las funcionalidades de estos objetos la responsabilidad de aspectos relacionados con el sistema en general como las trazas o el soporte de transacciones.
- *Contenedor*: Spring se puede considerar como un contenedor ya que contiene y maneja el ciclo de vida y la configuración de los objetos de la aplicación. Con Spring se puede especificar como cada objeto de la aplicación será creado, cómo será configurado y cómo será relacionado con lo demás.
- *Marco de trabajo*: Hace posible configurar y desarrollar aplicaciones complejas desde los componentes más simples. En Spring los objetos son definidos de forma declarativa, generalmente a través de ficheros XML. El marco de trabajo como tal ofrece otras funcionalidades como el manejo de transacciones, integración con marcos de trabajo que manejan la persistencia de los datos, y otras, dejando para el desarrollador solo la parte de la lógica de la aplicación.

Los objetos de Spring en una determinada aplicación a menudo no tienen dependencia con alguna clase específica del marco de trabajo evidenciando la cualidad de no intrusivo (50). Mientras que muchas aplicaciones fuerzan al código de la aplicación a estar al tanto del marco de trabajo implementando interfaces específicas del mismo o extendiendo las funcionalidades de clases específicas del mismo, Spring se enfoca en la dependencia del código de la aplicación del marco de trabajo configurando objetos de la aplicación que no importen las APIs del marco de trabajo. (51)

Spring está compuesto por una serie de módulos que ofrecen la mayoría de los aspectos que son necesarios para el desarrollo de aplicaciones empresariales. No obstante, las aplicaciones no tienen que hacer uso de todo el marco de trabajo sino que pueden escogerse los módulos que serán utilizados, además de que posee puntos de integración con una gran cantidad de marcos de trabajos y bibliotecas de clase, de manera que no es necesario reescribir estas funciones de nuevo (50). El esquema a continuación muestra los módulos del marco de trabajo Spring.

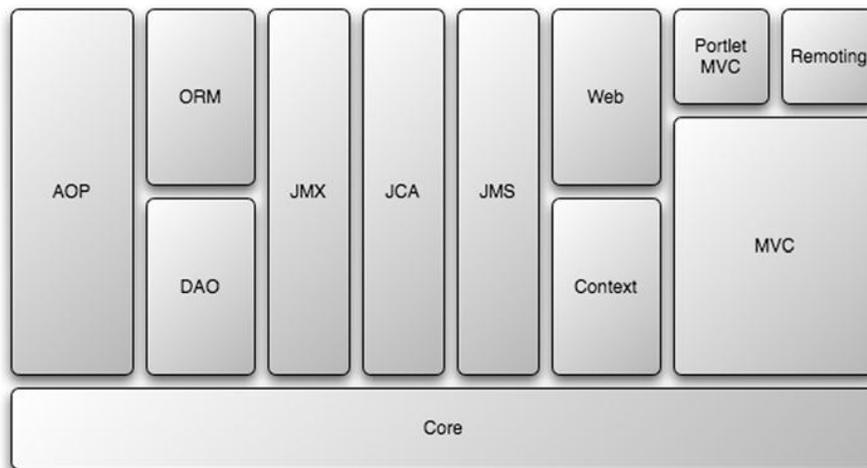


Figura 3: Módulos del marco de trabajo Spring. (50)

## 1.15 Servidor de base de datos

Los servidores de bases de datos son dispositivos que almacenan grandes colecciones de datos de forma estructurada. Son utilizados en todo el mundo en una gran cantidad de aplicaciones. Permiten realizar almacenamiento, acceso y análisis de datos al estar los mismos almacenados de forma estructurada. La mayoría de los servidores ofrecen una interfaz de texto para interactuar con el servidor usando SQL o alguna de sus variantes. (52)

### 1.15.1 PostgreSQL

PostgreSQL es un sistema de administración de base de datos relacional desarrollado en el departamento de Ciencias de Computación de la Universidad de California. Es un proyecto de código abierto y posee un excelente ajuste al estándar SQL. Es distribuido bajo una licencia similar a la BSD o MIT, que permite a los usuarios disponer del código (puede ser usado, modificado y distribuido por cualquiera libremente), incluso revender el producto compilado sin el código fuente. La única restricción es la imposibilidad de responsabilizar legalmente a los

desarrolladores por problemas con el software. PostgreSQL debería poder funcionar sin problemas en cualquier plataforma compatible con UNIX además de ejecutarse en los sistemas operativos Windows. (53) (54)

El proyecto de Postgres se enfoca fundamentalmente en lograr que tenga las siguientes características: robusto, software de alta calidad, mantenible, código documentado, poco trabajo en la administración del mantenimiento, soporte para el estándar SQL, interoperabilidad y compatibilidad, rendimiento, seguridad y amplia disponibilidad. Se ha destacado precisamente por el amplio espectro de características avanzadas que tiene. (54)

Está estructurado en una arquitectura cliente – servidor, se pueden realizar consultas con alto grado de complejidad, existencia y manejo de llaves foráneas, disparadores (triggers), vistas, integridad transaccional, un diseño enfocado en soportar configurable y extensible, es escalable y tiene un alto rendimiento con muchas opciones para optimizarlo. Permite una alta concurrencia donde las operaciones de lectura y escritura no se bloquean entre sí. Esta característica es denominada “Multi-Version Concurrency Control”. (54)

Actualmente es desarrollado y mantenido por un grupo considerable de desarrolladores y contribuyentes. Por consiguiente posee una amplia comunidad de usuarios que le da soporte, por lo que cualquier problema puede ser solucionado con relativa facilidad además de poder utilizar listas de distribución de correos con esta finalidad. (54) (53)

### **1.16 Conclusiones parciales**

Se plantea la necesidad de la construcción de un sistema propio de uso interno para la gestión de los procesos de la Cámara de Comercio de la República de Cuba, debido a los requerimientos y limitaciones de la entidad cubana. Los criterios de mantenibilidad, independencia tecnológica, portabilidad, documentación y el enfoque a procesos fueron la base para la selección de las herramientas y tecnologías a emplear en el desarrollo de la solución. Lo anterior permitió que las mismas estuvieran ajustadas a las necesidades y restricciones de la universidad, el centro productivo y el equipo de desarrollo. Como uno de los aspectos más importantes se tiene el enfoque a procesos que tendrá la solución final, basado en las ventajas que el mismo ofrece para el proyecto y para los clientes. El mismo será garantizado mediante el uso del gestor de procesos Bonita Open Solution, aprovechando al mismo tiempo los beneficios de su uso, tales como la facilidad en el cambio de los procesos de negocio así como la generación de interfaces y manejo básico de datos.

## CAPÍTULO 2: SOLUCIÓN PROPUESTA

Con el objetivo de desarrollar una solución de software a la medida, es necesario realizar un estudio detallado de los procesos del negocio que sustentan los objetivos de la entidad y cualquier procedimiento asociado a estos. Se deben identificar puntualmente las operaciones que serán objeto de informatización así como los requisitos funcionales y no funcionales, definiendo de esta manera, las características que debe tener el sistema a desarrollar. Se deben tener en cuenta los sistemas automatizados que ya existen en la entidad o aquellas operaciones que poseen cierto grado de automatización. Es importante recopilar datos sobre la información que se manipula, los documentos que son utilizados, normas, procedimientos, así como aquellos que serán generados por el sistema.

Siguiendo las pautas establecidas para el modelado de procesos de negocio y los artefactos de RUP se procede a la modelación y especificación de diferentes artefactos tales como los procesos de negocio, modelos de procesos, requisitos funcionales y no funcionales, modelos de casos de uso y otros, que en su conjunto permiten modelar el sistema completamente.

### 2.1 Modelado de procesos de negocio

La Cámara de Comercio de la República realiza diferentes actividades y gestiones enmarcadas en el contexto comercial de sucursales y agentes de sociedades mercantiles extranjeras, agencias de viajes nacionales, sucursales de agencias de viajes extranjeras así como empresas importadoras y exportadoras. Los procesos identificados que serán objeto de informatización son el Registro Nacional de Agencias de Viaje y el Registro de Empresas Importadoras y Exportadoras. A continuación se describen los procesos de negocio en cuestión; para mayor información consultar el Anexo # 1.

#### *Descripción del proceso Registro de Agencias de Viaje*

El Registro de Agencias de Viaje abarca todo el flujo de actividades que son realizadas en aras de registrar toda la información concerniente a las agencias de viajes nacionales y sucursales extranjeras en Cuba, así como contratos de representación de agencias de viajes extranjeras. El proceso puede subdividirse para su mejor comprensión en tres áreas principales:

- Agencias de Viajes Nacionales
- Sucursales de Agencias de Viajes Extranjeras
- Contratos de Representación

En cada una de estas áreas se realizan básicamente las mismas operaciones, las cuales consisten en registrar o inscribir en el registro las agencias, sucursales y contratos de representación, renovar el registro o inscripción de los mismos y cancelar dicho registro. Los procedimientos mencionados anteriormente involucran a las agencias de viaje, la Dirección Jurídica de la Cámara de Comercio y al Ministerio del Turismo (MINTUR).

Las operaciones de registro de las diferentes áreas son realizadas por el encargado del registro de la Dirección Jurídica de la Cámara de Comercio cuando las partes interesadas así lo solicitan o a solicitud del MINTUR. Las agencias de viajes nacionales deben contar con un documento o aprobación del MINTUR que autoriza su constitución además de los documentos que deben presentar ante la cámara, iniciando de este modo el subproceso registral de este tipo de agencias. Se creará el expediente asociado a la agencia donde se registrarán todas las informaciones pertinentes, incluida la licencia emitida en la operación que será entregada a la misma, con lo que se concluye la operación. Para el registro de las sucursales de agencias de viajes extranjeras, el proceso de registro es básicamente el mismo que para las agencias de viajes nacionales, solo que en este caso en particular son necesarios documentos que acrediten la actividad de la misma en su país de origen, relaciones con empresas cubanas y extranjeras así como otros requerimientos legales y económicos. En el caso de los contratos de representación, los mismos son inscritos teniendo como representante a la Agencia Nacional de Viajes. Esta última presenta ante la cámara los documentos necesarios para la inscripción. El encargado del registro de la Dirección Jurídica revisará que dichos documentos se cumplan las formalidades establecidas para ello y formalizará la inscripción emitiendo como parte de la operación la Certificación de la inscripción, archivando además una copia de la misma.

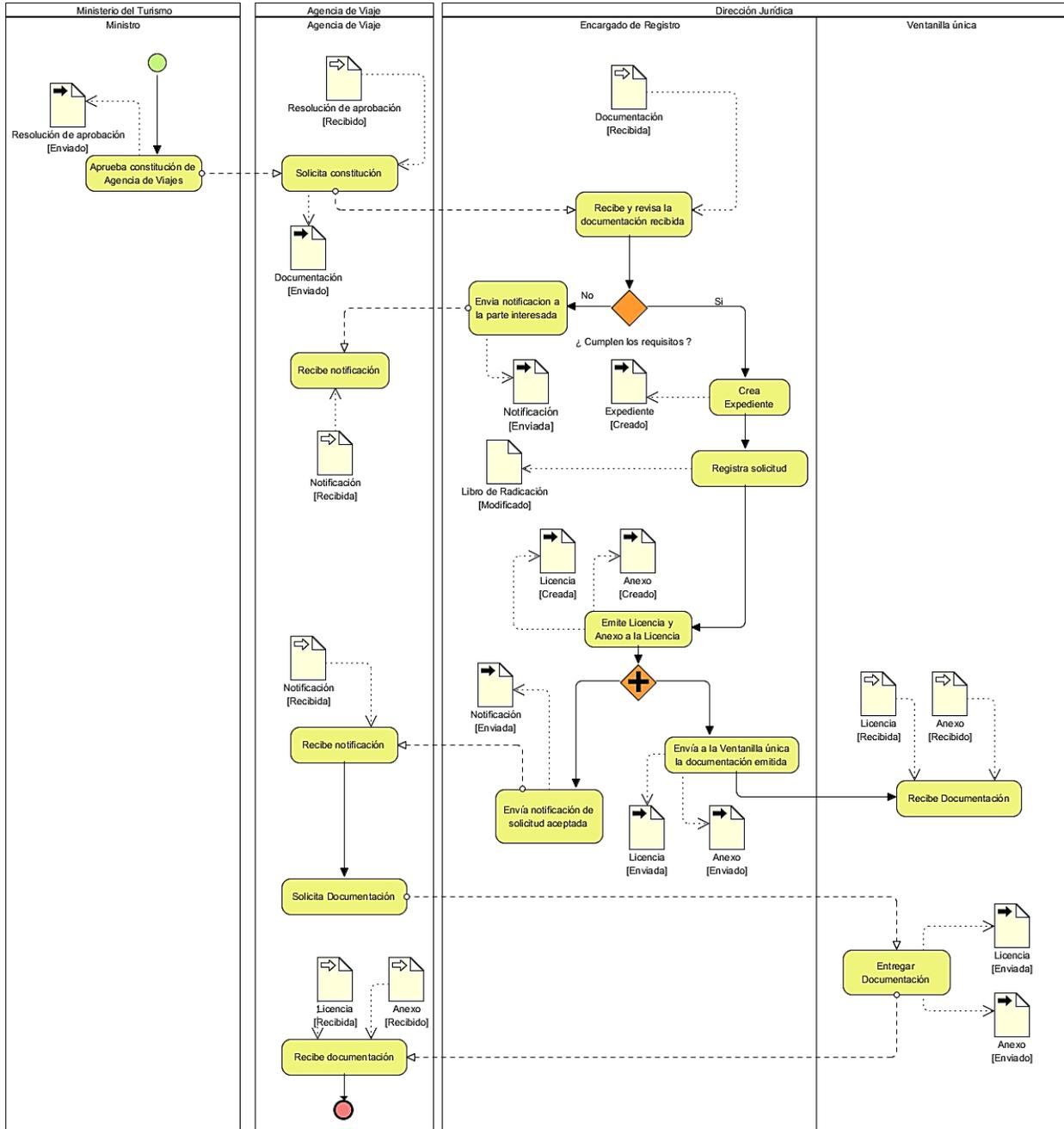


Figura 4: DPN Inscripción de Agencia de Viaje Nacional.

Para la renovación de las inscripciones de las agencias de viajes nacionales, se debe presentar la documentación requerida ante el encargado del registro de la Dirección Jurídica de la Cámara de Comercio dentro de los 30 días hábiles anteriores al vencimiento de la Licencia emitida en la inscripción. En este paso resulta necesario esperar por la decisión del MINTUR para aprobar o denegar dicha renovación. En caso de que sea aprobada, se registra la operación y se emite una nueva Licencia que tiene una vigencia de 3 años en este caso y se

entrega la documentación a la agencia de viaje. En caso de que la renovación sea denegada se procede a ejecutar el procedimiento más adelante descrito como cancelación de la inscripción de las agencias de viajes nacionales.

Para el caso de las sucursales de agencias de viajes extranjeras, el encargado del registro efectúa la renovación verificando que los documentos presentados por la Sucursal cumplan con los requerimientos establecidos. En este caso en particular la Cámara de Comercio envía una comunicación al MINTUR junto al traslado de la solicitud con sus recomendaciones. Para la renovación de los Contratos de Representación, la solicitud es presentada por la Agencia Nacional de Viajes ante la Cámara de Comercio. El encargado del registro registra la operación y emite la certificación correspondiente.

La cancelación o modificación de la inscripción de una agencia de viaje nacional es facultad del MINTUR, quien presentará ante el encargado del registro de la Dirección Jurídica la solicitud de cancelación. Este a su vez registra la operación y se entregan los documentos constitutivos a la agencia de viajes si esta así lo solicita. La cancelación de la inscripción de sucursales de agencias de viaje extranjeras es realizada en el momento en que la misma presenta la solicitud. En el caso de los contratos de representación, la solicitud de cancelación es presentada por la Agencia Nacional de Viajes o la operación es realizada como parte de las funciones del encargado del Registro cuando ha vencido el término establecido en la inscripción o renovación anterior.

#### *Descripción del proceso Registro de Empresas Importadoras y Exportadoras*

Para el registro de las empresas importadoras y exportadoras, las mismas deben estar previamente autorizadas por resolución emitida por el Ministerio de Comercio Exterior. El encargado del registro revisa la documentación que presenta la entidad directamente ante la Cámara de Comercio verificando que cumpla con las formalidades establecidas, registrará los datos de la empresa, creará el expediente correspondiente y registrará la fecha de asiento. Registrará además las disposiciones establecidas en la Resolución del Ministerio de Comercio Exterior. Seguidamente emitirá la certificación correspondiente y se enviará una notificación a la empresa informando de la disponibilidad de los documentos.

Para el proceso de modificación de facultades, el Ministerio de Comercio Exterior emite una resolución modificando las facultades que fueron concedidas a la empresa, el encargado del registro registrará la nueva información emitiendo una nueva certificación en este caso y en

caso de que cambie el código que se había emitido en la resolución. Se notifica a la entidad interesada la disponibilidad de los documentos. En el caso de cancelación de las facultades que le han sido conferidas a una empresa el encargado del registro registrará lo así dispuesto en la resolución enviada por el Ministerio de Comercio Exterior.

## **2.2 Requisitos de software**

Los requisitos de software se encuentran clasificados en dos grupos: funcionales y no funcionales. Los requisitos funcionales son capacidades o condiciones que el sistema debe cumplir sin alterar la funcionalidad del producto y se mantienen invariables sin importar con qué propiedades o cualidades se relacionen. Los requisitos no funcionales son propiedades o cualidades que el producto debe tener; especifican propiedades como restricciones del entorno o de la implementación, rendimiento, dependencia de la plataforma, facilidad de mantenimiento, extensibilidad y fiabilidad. Los requisitos de manera general recogen las necesidades del cliente añadiendo el valor esperado por estos. Deben ser descritos de modo que sean comprendidos por los usuarios y clientes. (15)

### **2.2.1 Requisitos funcionales**

A continuación se muestran las funcionalidades críticas del sistema de los procesos de Registro de Agencias de Viaje y Registro de Empresas Importadoras y Exportadoras. Para obtener más detalles ver Anexo # 2.

#### *Proceso Registro de Agencias de Viaje*

**RF\_1:** Inscribir Agencia de Viaje: El sistema permite capturar, validar y registrar los datos de una Agencia de Viajes o Sucursal Extranjera.

**RF\_2:** Modificar inscripción de Agencia de Viaje: Permite modificar los datos previamente registrados en el sistema de la Agencia de Viaje Nacional o Sucursal Extranjera.

**RF\_3:** Eliminar inscripción de Agencia de Viaje: Permite eliminar los datos registrados previamente en el sistema de una Agencia de Viaje Nacional o Sucursal Extranjera.

**RF\_4:** Generar Licencia de Agencia de Viaje: Permite generar un documento en formato PDF recuperando los datos necesarios que fueron previamente almacenados.

**RF\_5:** Generar Anexo a la Licencia de Agencia de Viaje: Permite generar un documento en formato PDF recuperando los datos necesarios que fueron previamente almacenados y especificando nuevos datos.

**RF\_6:** Enviar notificación: El sistema debe permitir notificar de las diferentes operaciones registrales vía correo electrónico.

**RF\_7:** Renovar Inscripción de Agencia de Viaje: Permite guardar en el sistema los datos legales de la renovación, así como establecer un nuevo estado sobre la entidad.

**RF\_8:** Buscar Agencia de Viaje: Permite realizar una búsqueda sobre las Agencias de Viajes Nacional o Sucursales Extranjeras registradas previamente mediante uno o varios términos de búsqueda.

**RF\_9:** Visualizar datos de Agencia de Viaje: Permite visualizar todos los datos de determinada Agencia de Viaje Nacional o Sucursal Extranjera previamente almacenados en el sistema.

**RF\_10:** Generar Solicitud de Renovación: Generar un documento en formato PDF recuperando los datos necesarios que fueron previamente almacenados

**RF\_11:** Cancelar Agencia de Viaje: Registrar datos de la operación de cancelación y establecer el estado de “cancelada” a la Agencia de Viajes.

**RF\_12:** Registrar Contrato de Representación: El sistema permite capturar, validar y registrar los datos de un Contrato de Representación.

**RF\_13:** Modificar Contrato de Representación: Permite modificar los datos guardados en el sistema del Contrato de Representación

**RF\_14:** Eliminar Contrato de Representación: Permite eliminar los datos guardados en el sistema del Contrato de Representación

**RF\_15:** Visualizar Contratos de Representación: Permite visualizar todos los datos de determinado Contrato de Representación previamente almacenados en el sistema

**RF\_16:** Generar Certificación de Contrato de Representación: Generar un documento en formato PDF recuperando los datos necesarios que fueron previamente almacenados

**RF\_17:** Renovar Contrato de Representación: Permite guardar en el sistema los datos legales de la renovación, así como establecer un nuevo estado sobre el contrato de representación.

**RF\_18:** Buscar Contrato de Representación: Permite realizar una búsqueda sobre los Contrato de Representación registrados previamente mediante uno o varios términos de búsqueda.

**RF\_19:** Cancelar Contrato de Representación: Registrar datos de la operación de cancelación y establecer el estado de “cancelado” al Contrato de Representación.

**RF\_20:** Revisar Contratos de Representación vencidos: Permite consultar cuáles contratos han vencido el término previsto.

### *Proceso Registro de Empresas Importadoras y Exportadoras*

**RF\_21:** Inscribir empresa importadora – exportadora: El sistema permite capturar, validar y registrar los datos de una empresa importadora – exportadora.

**RF\_22:** Procesar Documento Nomenclador de Productos: El sistema debe procesar automáticamente un documento subido en formato doc y almacenar automáticamente la información acerca de las actividades autorizadas en el mismo a una empresa importadora – exportadora.

**RF\_23:** Modificar empresa importadora/exportadora: Permite modificar los datos guardados en el sistema de una empresa importadora – exportadora.

**RF\_24:** Eliminar empresa importadora – exportadora: Permite eliminar los datos guardados en el sistema de una empresa importadora – exportadora.

**RF\_25:** Visualizar datos de empresa: Permite visualizar todos los datos de determinada empresa importadora – exportadora previamente almacenados en el sistema

**RF\_26:** Generar Certificación: Permite generar un documento en formato PDF recuperando los datos necesarios que fueron previamente almacenados sobre la empresa importadora – exportadora

**RF\_27:** Buscar empresa: Permite realizar una búsqueda sobre las empresas importadoras – exportadoras registradas previamente mediante uno o varios términos de búsqueda.

**RF\_28:** Modificar facultades de empresa importadora – exportadora: Permite registrar que se ha hecho una modificación a las facultades de una empresa importadora – exportadora y registrar las nuevas facultades.

**RF\_29:** Cancelar facultades de empresa importadora – exportadora: Registrar datos de la operación de cancelación y establecer el estado de “cancelada” a una empresa importadora – exportadora.

### **2.2.2 Requisitos no funcionales**

A continuación se relacionan los requisitos no funcionales identificados.

#### *Usabilidad:*

**RNF\_1:** El sistema será usado por el personal de la Cámara de Comercio que se encuentre involucrado en los procesos de Registro de Agencias de Viaje y Registro de Empresas Importadoras y Exportadoras.

#### *Tipo de Usuario*

**RNF\_2:** El sistema podrá ser utilizado por los usuarios que se describen a continuación:  
Director Jurídico.

#### *Tipo de Aplicación Informática*

**RNF\_3:** El producto constituirá una aplicación web enfocada en las necesidades de la Cámara de Comercio de la República de Cuba.

**RNF\_4:** El sistema estará enfocado a la gestión de información relacionada con los procesos registrales que se llevan a cabo en la Cámara de Comercio.

#### *Finalidad*

**RNF\_5:** Este sistema está enfocado a la gestión de información relacionada con los procesos registrales que se llevan a cabo en la Cámara de Comercio.

#### *Ambiente*

**RNF\_6:** El sistema deberá presentar una interfaz de usuario fácil de entender y usar.

**RNF\_7:** El tiempo de respuesta brindado por el sistema será menor de 3 segundos.

#### *Confiability*

**RNF\_8:** El tiempo máximo de inactividad es de 5 min, y una vez que el sistema quede inactivo el usuario deberá autenticarse nuevamente.

**RNF\_9:** El sistema deberá estar disponible al menos durante toda la jornada laboral.

#### *Eficiencia*

**RNF\_10:** El número de usuarios que el sistema puede alojar es 30.

#### *Soporte*

**RNF\_11:** El uso del sistema requerirá un tiempo de preparación previa por los usuarios finales para su correcta explotación.

**RNF\_12:** Se deberá recibir entrenamiento para la configuración y el mantenimiento del sistema.

#### *Restricciones de diseño*

**RNF\_13:** Para el montaje del sistema se requerirá del sistema gestor de bases de datos PostgreSQL 9.1 y del contenedor web Apache Tomcat 6.0.

**RNF\_14:** Para la utilización del sistema se requerirá Mozilla Firefox 1.5 o superior.

#### *Interfaz*

**RNF\_15:** El sistema presentará una interfaz legible, simple de usar, interactiva.

**RNF\_16:** El sistema deberá tener un diseño de interfaz de usuario basada en las características de BonitaSoft.

**RNF\_17:** El sistema deberá ser independiente de la plataforma.

#### *Requisitos de Licencia*

**RNF\_18:** El software debe cumplir los siguientes requisitos de licencias: Licencia GPL 2.0 para la herramienta Bonita y licencia PostgreSQL para el gestor de base de datos Postgres.

#### *Estándares Aplicables*

**RNF\_19:** Se aplicará el estándar de calidad CMMI 2 como parte del Proceso de mejora que se lleva a cabo en la universidad.

#### *Seguridad*

**RNF\_20:** El sistema podrá ser utilizado solamente por usuarios autenticados en el mismo.

**RNF\_21:** El sistema brindará la posibilidad de establecer permisos sobre acciones, garantizando que solo acceda a la información quien esté autorizado.

**RNF\_22:** El sistema mostrará las funcionalidades del sistema de acuerdo a quien esté autenticado en el mismo.

**RNF\_23:** El sistema deberá garantizar la protección ante acciones no autorizadas.

### **2.3 Actores del sistema**

Un actor es algo que comunica con el sistema o producto y que es externo al sistema en sí mismo. Representa una clase de entidades externas, generalmente una persona aunque no siempre, que lleva a cabo una actividad en el sistema. Luego de definir los actores se tiene además identificado el entorno externo del sistema. (15)

En la tabla siguiente se especifica y describe el actor principal que invoca las funcionalidades definidas para el subproceso de Inscripción de Agencias de Viajes, de forma general, este actor es quien realiza todas las operaciones en los procesos abordados.

Tabla 1: Actores del sistema.

Actor	Descripción
Encargado del registro	Se encarga de realizar todas las operaciones registrales en el sistema como el registro, renovación y cancelación de inscripción de las agencias de viajes nacionales, sucursales de agencias de viajes extranjeras así como contratos de representación. En cada una de estas operaciones emite las licencias, anexos y certificados correspondientes y envía una notificación vía correo electrónico para la recogida de los documentos.

## 2.4 Diagrama de caso de uso del sistema

Un diagrama de casos de uso muestra un grupo de casos de uso y actores y sus relaciones. Los diagramas de casos de uso representan la vista estática de un sistema. Estos son especialmente importantes en la organización y modelado del comportamiento del sistema. Los diagramas de casos de uso son útiles para visualizar, especificar y documentar el comportamiento de un elemento. Ayudan al acercamiento y entendimiento de sistemas, subsistemas y clases mediante la presentación de la vista externa de cómo esos elementos son usados en contexto. Son importantes además, para probar sistemas mediante ingeniería directa y para la comprensión de sistemas ejecutables mediante ingeniería inversa. (55)

Se identificaron 22 casos de uso que fueron representados en el diagrama de casos de uso del sistema como se muestra en la Figura 5, En la realización del mismo se aplicaron los patrones de casos de uso CRUD Parcial, Extensión o Inclusión Concreta: Extensión y Concordancia: Reutilización.

- *CRUD Parcial*: Es un patrón estructural que modela una de las alternativas del CRUD Completo como un caso de uso separado. Es utilizado cuando una de las alternativas del caso de uso es más significativa, larga o mucho más compleja que las demás alternativas. (56)
- *Extensión o Inclusión Concreta: Extensión*: Es un patrón estructural que se aplica cuando un caso de uso puede extender el flujo de otro caso de uso, o puede ser instanciado por el mismo. (56)
- *Concordancia: Reutilización*: Es un patrón estructural que consiste en tres casos de uso. El primero llamado "Sub-secuencia Común", modela la secuencia de acciones que aparecen en múltiples casos de uso del modelo. Los otros dos casos de uso comparten

de esta sub-secuencia común de acciones (dos es la menor cantidad que puede existir).  
(56)

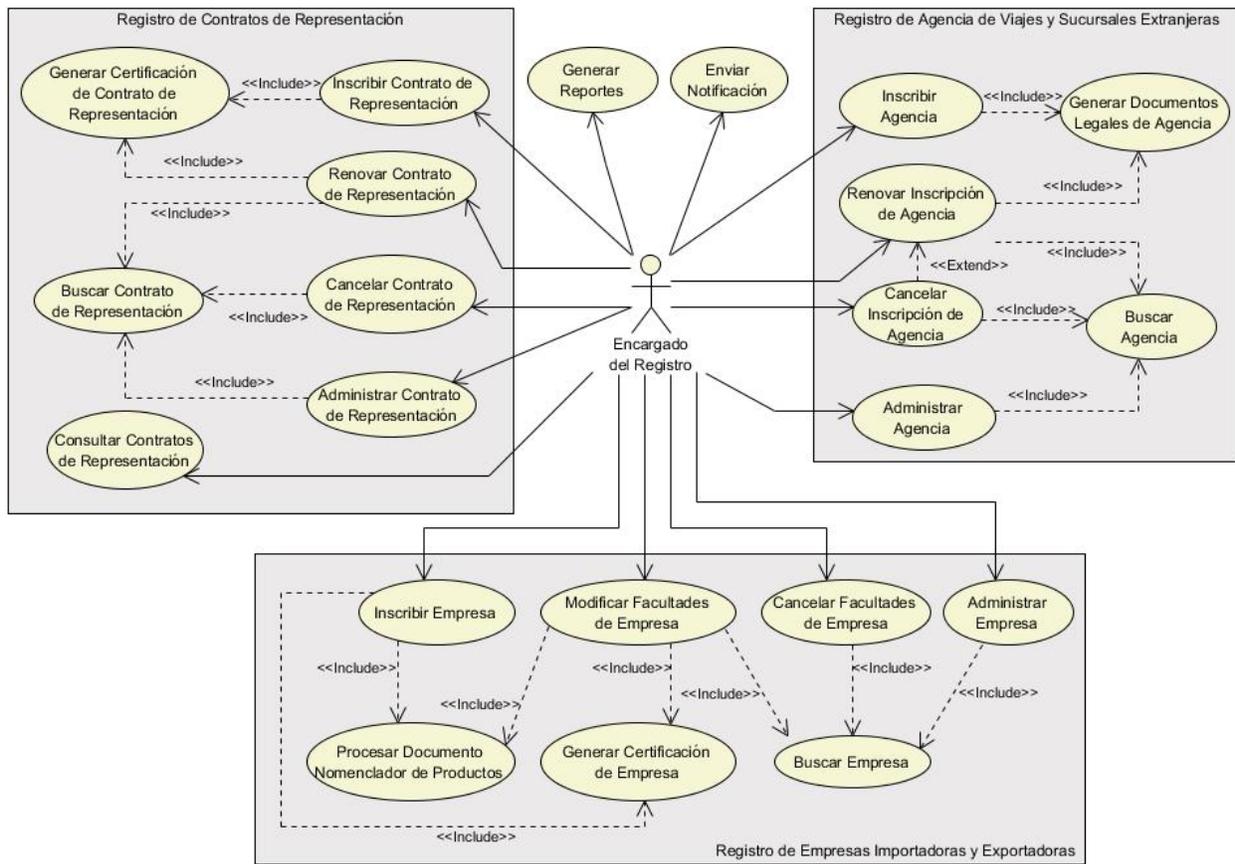


Figura 5: Diagrama de caso de uso del sistema.

### 2.4.1 Descripción de casos de uso del sistema

La descripción de la realización de un caso de uso detalla paso por paso el flujo de eventos necesarios para realizar el camino básico de una de las funcionalidades del sistema correspondiente a los procesos del negocio. Para consultar las descripciones de casos de uso, revisar el Anexo # 3. A continuación se describe la realización del caso de uso Inscribir Agencia.

Tabla 2: Descripción del caso de uso Inscribir Agencia.

<b>Caso de Uso:</b>	Inscribir Agencia	
<b>Actores:</b>	Encargado del registro	
<b>Resumen:</b>	El caso de uso inicia cuando el actor (encargado del registro) solicita registrar una nueva agencia de viajes introduciendo en el sistema los datos necesarios. Al concretar la operación el sistema salvará la entrada de información y generará dos documentos en formato PDF: la licencia de inscripción y el anexo a la misma. Adicionalmente el sistema enviará un correo electrónico a la agencia o sucursal registrada para notificarle de la disponibilidad de los documentos.	
<b>Precondiciones:</b>	El sistema debe estar instalado y ejecutado correctamente. El actor debe estar autenticado con los permisos necesarios.	
<b>Prioridad:</b>	Crítico	
<b>Flujo de eventos</b>		
<b>Sección “Inscripción Agencia”</b>		
<b>Actor</b>	<b>Sistema</b>	
1. El caso de uso se inicia cuando el usuario accede a la sección “Inscribir Agencia”.	2. El sistema muestra la interfaz para entrar los datos de la agencia: Nombre, Otras denominaciones, Domicilio Casa Matriz, Dirección, Localización, Lugar de constitución, Presidente, Representante, Expediente, Correo(s) electrónico(s), Teléfono, Fax, Tomo, Folio, Fecha de constitución, Fecha de vencimiento. 3. El sistema muestra las opciones “Enviar notificación” (seleccionable), “Guardar” y “Cancelar”.	
4. El usuario introduce los datos que se requieren para el registro y selecciona la opción “Guardar”. Si el usuario desea cancelar la operación ver flujo alterno 1. Si el usuario selecciona la opción	5. El sistema guarda los datos de la agencia y muestra la interfaz para entrar los datos del anexo a la licencia: Otras anotaciones.	

<p>“Enviar notificación” ver sección “Inscribir Agencia – Enviar notificación”.</p>	
<p>6. El usuario introduce los datos que se requieren para el registro y selecciona la opción “Guardar”.</p>	<p>7. El sistema guarda los datos del anexo y muestra la interfaz para la descarga de los documentos generados.</p>
<p>8. El usuario selecciona el vínculo “Descargar documento” correspondiente a la licencia.</p>	<p>1. El sistema genera la licencia de inscripción en formato PDF y le da al usuario la posibilidad de descargarla o visualizarla en el navegador según la configuración del navegador web.</p>
<p>9. El usuario selecciona el vínculo “Descargar documento” correspondiente al anexo a la licencia.</p>	<p>2. El sistema genera el anexo a la licencia de inscripción en formato PDF y le da al usuario la posibilidad de descargarla o visualizarla en el navegador según la configuración del navegador web del usuario finalizando el caso de uso.</p>
<p><b>Prototipo de Interfaz</b></p>	

 **Cámara de Comercio de la República de Cuba** usuario  
Cerrar sesión

Inscribir Agencia

Nombre:

Otras denominaciones:

País:

Domicilio Casa Matriz:

Dirección:

Localización:

Lugar de constitución:

Presidente:

Representante:

Expediente:

Correo(s) electrónico(s):

Teléfono:  Fax:

Tomos:  Folios:

Fecha de constitución:  Fecha de vencimiento:

Enviar notificación

 **Cámara de Comercio de la República de Cuba**

usuario  
Cerrar sesión

Datos de Licencia

*Datos del Anexo a la Licencia*

Otras anotaciones:

Guardar      Cancelar



**Sección “Inscribir Agencia – Enviar notificación”**

Actor	Sistema
	1. El sistema envía una notificación vía correo electrónico a la agencia notificando la disponibilidad de los documentos y retorna a la interfaz principal finalizando el caso de uso.
Flujo alternativo 1	
Actor	Sistema
2. El usuario elige cancelar la	3. El sistema cancela la operación

operación.	retornando a la interfaz principal terminando el caso de uso.
<b>Poscondiciones</b>	<ul style="list-style-type: none"> <li>• Quedan registrados en el sistema los datos de la agencia de viajes.</li> <li>• Quedan registrados en el sistema los datos del anexo a la licencia de inscripción de la agencia de viajes.</li> <li>• La agencia de viajes es notificada de la disponibilidad de los documentos mediante un correo electrónico.</li> </ul>

## 2.5 Elementos arquitectónicos

La arquitectura de un sistema comprende varios elementos tales como estilos arquitectónicos, patrones arquitectónicos y patrones de diseño, cuyos niveles de abstracción entre los mismos están definidos en el mismo orden en que se mencionaron (57). Se abordará a continuación el patrón arquitectónico usado en la implementación del sistema describiendo la arquitectura del mismo así como los patrones de diseños empleados.

En la bibliografía consultada un patrón se define como una regla que consta de tres partes, la cual expresa una relación entre un contexto, un problema y una solución. A partir de esta definición se plantea que un patrón arquitectónico es una descripción de un problema particular y recurrente de diseño, que aparece en un contexto específico y presenta una solución demostrada. Los patrones arquitectónicos expresan el esquema de organización estructural fundamental para sistemas de software. (58)

El patrón arquitectónico usado en la implementación de la solución es Arquitectura en Capas, el mismo fue seleccionado después de estudiar la medida en que se adapta a los requerimientos de calidad que debe cumplir el sistema. Este patrón consiste en estructurar aplicaciones que pueden ser descompuestas en grupos de subtarear, las cuales se clasifican de acuerdo a un nivel particular de abstracción. En otras palabras las capas particionan el software, teniendo cada una de estas un grupo de interfaces públicas que pueden ser invocadas por otras capas o software, disponiendo por consiguiente de un conjunto de servicios cohesionados. Del mismo se pueden desatacar atributos tales como la reusabilidad, la portabilidad y la facilidad de pruebas. (59)

En la Figura 6 se representa la arquitectura del sistema y a continuación se describen brevemente cada una de las capas lógicas que conforman la misma:

- *Capa Base*: Es transversal a las demás capas, ya que maneja aspectos que son utilizados en toda la aplicación tales como las trazas, el manejo de las excepciones y las transacciones. Para ello, en la misma se hace uso del framework Spring, aprovechando las ventajas que ofrece el mismo tales como la orientación a aspectos y la inversión de control, este último, para hacer uso de la inyección de dependencias en aras de lograr un bajo acoplamiento o dependencia.
- *Capa de Servicios*: Ofrece un conjunto de servicios u operaciones de entrada y salida de datos así como invocaciones a métodos sin valor de retorno a los que se acceden a través de interfaces. Estos servicios serán “consumidos” o invocados a través de elementos denominados conectores en las diferentes actividades del flujo del proceso. De esta manera brinda el mecanismo que permite conectar o complementar la lógica de un flujo de proceso manejada por el gestor, con la lógica implementada específicamente para el negocio en cuestión.
- *Capa de Negocio*: Ofrece un conjunto de interfaces que son usadas por la capa de servicios para invocar las funcionalidades que responden a la lógica del negocio. La misma contiene a los Business Objects (BO) u objetos de negocio, los que típicamente son referidos como clases controladoras. El objetivo es extraer las funcionalidades de la aplicación de las clases entidades utilizando estas solo como objetos de transferencia de datos.
- *Capa de Acceso a datos*: Interactúa con la capa de negocio a través de interfaces. La misma contiene las clases que ejecutan la lógica de la persistencia de las entidades del negocio. En la misma se hace uso de la especificación de JPA para manejar la persistencia de los objetos. Las operaciones que básicamente se realizan son las de insertar o hacer persistir un objeto, actualizar o modificar la información almacenada del mismo y recuperar dicha información.

Las entidades del negocio son objetos persistentes del dominio que representan conceptos reales tales como una empresa o una persona. Las mismas son agrupadas y representadas transversalmente a las capas de negocio y acceso a datos pues ambas las utilizan.

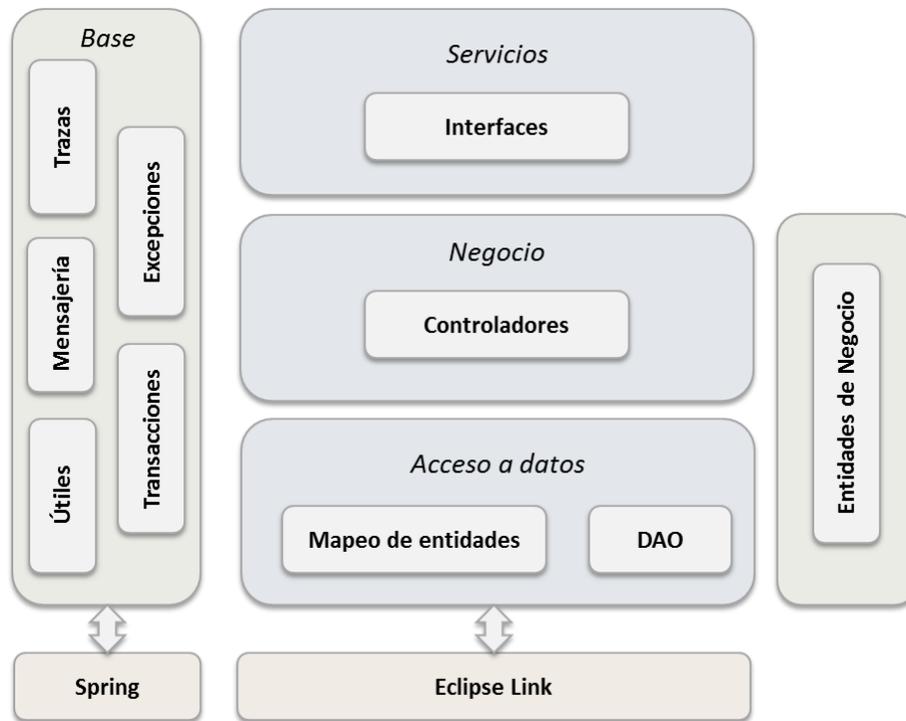


Figura 6: Arquitectura del sistema.

A un nivel menor de abstracción de los patrones arquitectónicos se encuentran los patrones de diseño. Un patrón de diseño provee un esquema para refinar los subsistemas o componentes de un sistema de software, o las relaciones entre ellos. Describe la estructura comúnmente recurrente de los componentes en comunicación, que resuelve un problema general de diseño en un contexto particular. Tienden a ser independientes de los lenguajes y paradigmas de programación y su aplicación no afecta necesariamente al sistema completo pero si a un subsistema o parte del mismo. (58)

### Patrones de diseño

A continuación se resumen brevemente los **patrones de diseño** empleados en el desarrollo del sistema clasificando los mismos de acuerdo a su propósito:

#### *Patrones para asignar responsabilidades (GRASP)*

- *Experto*: Asignar una responsabilidad al experto en información, la clase que cuenta con la información necesaria para cumplir la responsabilidad.
- *Creador*: Asignar a las clases la responsabilidad de crear instancias de otras clases.

- *Controlador*: Asignar la responsabilidad del manejo de un mensaje de los eventos de un sistema a una clase.
- *Bajo Acoplamiento*: Asignar una responsabilidad para mantener bajo acoplamiento. El grado de acoplamiento no puede considerarse aisladamente de otros principios como Experto y Alta Cohesión. Sin embargo, es un factor a considerar cuando se intente mejorar el diseño.
- *Alta Cohesión*: Asignar una responsabilidad de modo que la cohesión siga siendo alta.

### *Patrones GOF*

*Patrones de Creación*: Tratan la creación de instancias o sobre qué objetos un objeto delegará responsabilidades. (60)

- *Fábrica Abstracta*: Proporciona una interfaz para crear familias de objetos relacionados o dependientes sin especificar su clase concreta. El uso de este patrón se evidencia en la “fabricación” de los objetos de las capas de negocio y acceso a datos, así como algunos objetos del *core* o núcleo de la aplicación. Dicha instanciación se realiza mediante los ficheros de configuración XML que utiliza Spring sin conocer los detalles concretos de su instanciación.
- *Instancia Única*: Garantiza que solamente se crea una instancia de la clase y provee un punto de acceso global a este.

*Patrones Estructurales*: Tratan la relación entre clases, la combinación de clases y la formación de estructuras de mayor complejidad, describiendo así la forma en que diferentes tipos de objetos pueden ser organizados para trabajar unos con otros. (60)

- *Composición*: Permite gestionar objetos complejos e individuales de forma uniforme. Es utilizado para representar entidades de negocios que tienen una asociación recursiva o dicho de otra forma, clases que contienen clases de su mismo tipo.
- *Decorador*: Extiende la funcionalidad de un objeto dinamizante de tal modo que es transparente a sus clientes. Es utilizado en el sistema para recuperar información de objetos en un formato determinado.
- *Fachada*: Simplifica los accesos a un conjunto de objetos relacionados proporcionando un objeto de comunicación. Se evidencia en la existencia de un punto de acceso para la invocación de las interfaces definidas en la capa de servicios. Para cada uno de los procesos en cuestión (Registro de Agencias de Viaje y Registro de Empresas

Importadoras y Exportadoras) existirá un objeto de comunicación que implementa la interfaz correspondiente, agrupando las operaciones relacionadas con dicho proceso.

- *Proxy*: Los clientes asociados a un componente se comunican con un representante de éste, en lugar del componente en sí mismo. Es utilizado internamente por el framework Spring en la intercepción de las llamadas a métodos utilizando la programación orientada a aspectos.

*Patrones de Comportamiento*: Tratan la interacción y cooperación entre clases. Organizan, manejan y combinan comportamientos. (60)

- *Iterador*: Acceso secuencial a los elementos de una colección. Es utilizado para el acceder a todos los objetos que representan entidades del negocio definidos como colecciones en las declaraciones de las clases.
- *Estado*: Modifica el comportamiento de un objeto cuando su estado interno cambia. Usado para cambiar los estados por los cuáles deben transitar las diferentes entidades del negocio en la realización de los procesos de negocio.
- *Estrategia*: Define una familia de algoritmos, encapsula cada uno y los hace intercambiables. Utilizado para aplicar diferentes algoritmos de ordenamiento fundamentalmente sobre los objetos en colecciones, sobre los cuáles se realizan operación de la lógica del negocio de los procesos en cuestión.

## 2.6 Diagrama de clases del diseño

Un diagrama de clases muestra un conjunto de clases, interfaces, colaboraciones y sus relaciones. Representan el diseño estático de un sistema. Generalmente incluyen el modelado del vocabulario del sistema, modelado de las colaboraciones o el modelado de esquemas. Son importantes para visualizar, especificar y documentar modelos estructurados así como construir sistemas ejecutables usando ingeniería directa e inversa. (55)

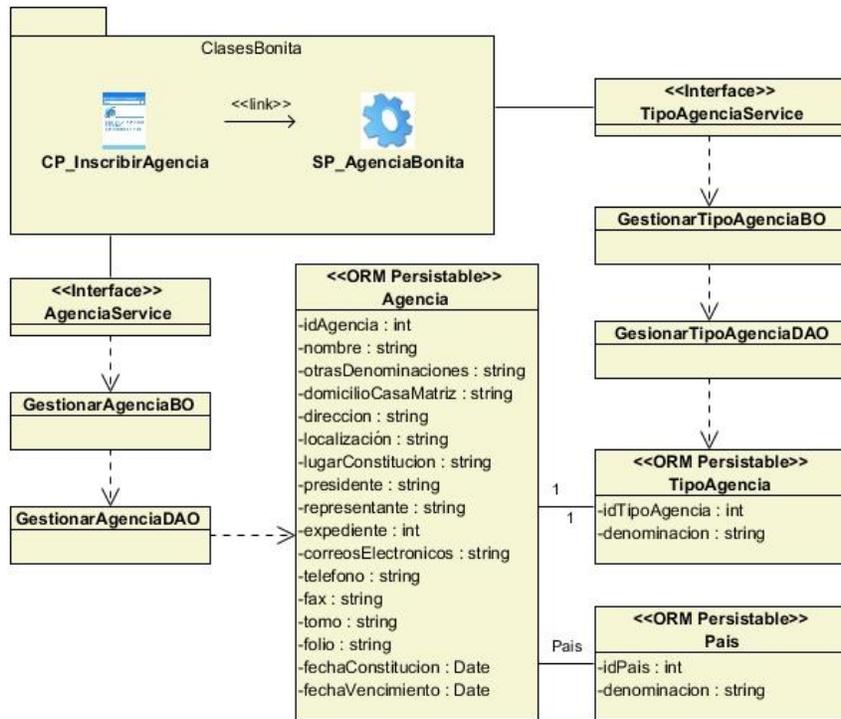


Figura 7: Diagrama de clases del diseño del CUS Inscribir Agencia.

## 2.7 Diagrama de secuencia

Los diagramas de secuencias pertenecen al grupo de los diagramas de interacción. Representa la secuencia de acciones ordenadas y detalladas en el tiempo, que ocurre cuando un actor inicializa un caso de uso mediante el envío de algún tipo de mensaje al sistema. Generalizando lo anterior, un diagrama de secuencia muestra la interacción que ocurre entre un conjunto de objetos y sus relaciones incluyendo los mensajes que pueden ser intercambiados entre los mismos. Este tipo de diagrama enfatiza en el orden en que se envían y reciben los mensajes. Son usados para modelar aspectos dinámicos de un sistema; incluyen elementos concretos o prototipos de clases, interfaces, componentes y nodos. Permiten visualizar, especificar, construir y documentar la dinámica de un conjunto de objetos o modelar el flujo de un caso de uso particular. (55)

A continuación se muestra el diagrama de secuencia que describe el flujo del caso de uso Inscribir Agencia.

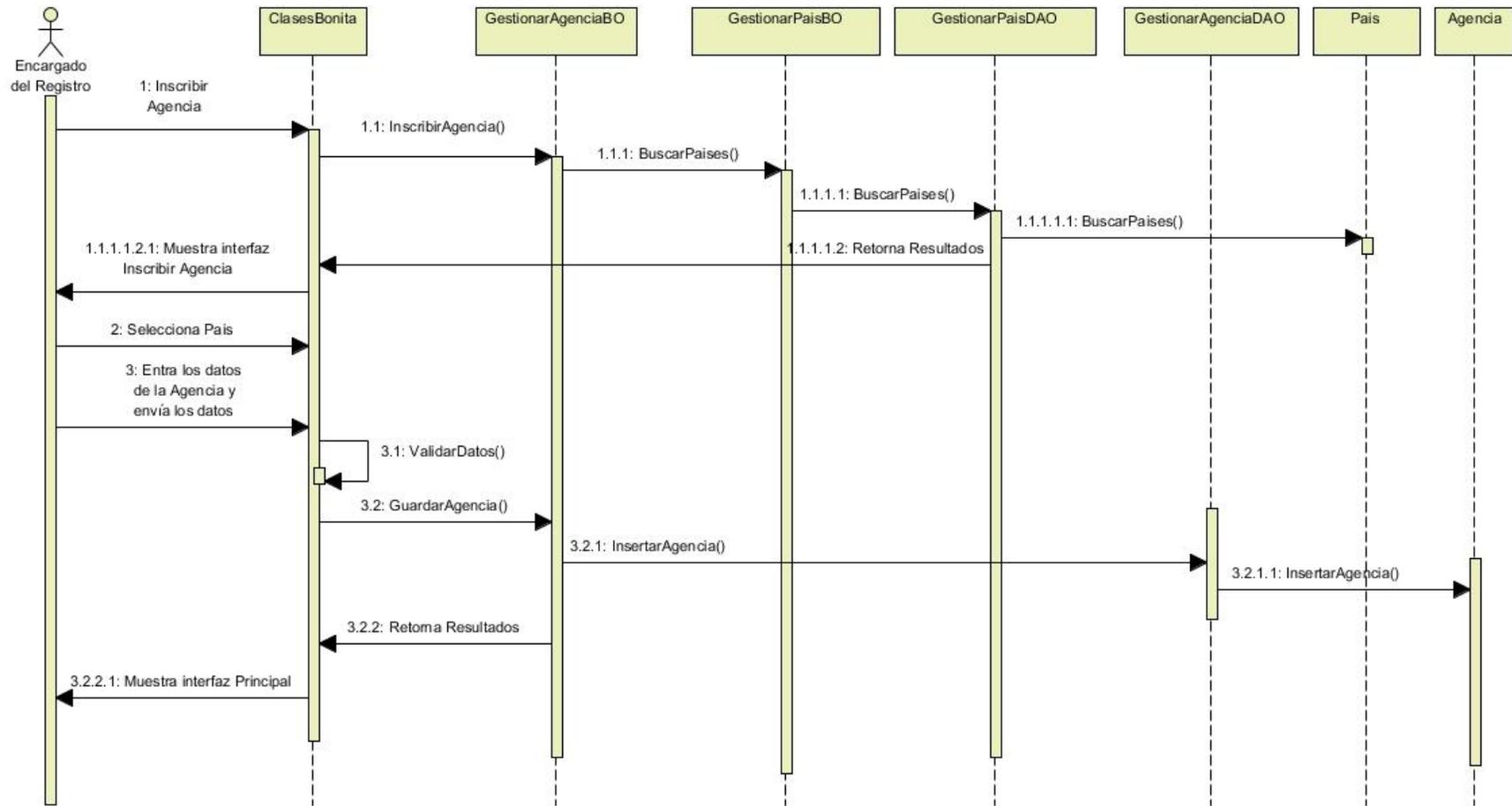


Figura 8: Diagrama de secuencia del CUS Inscribir Agencia.

## 2.9 Conclusiones parciales

En este capítulo se abordaron los elementos correspondientes a las características y el diseño del sistema. Se realizó la identificación y especificación de los procesos de Registro de Agencias de Viajes y Registro de Empresas Importadoras y Exportadoras de la Cámara de Comercio, realizando como parte de estas actividades los diagramas de procesos. Esto permitió un mejor entendimiento de los procesos de la entidad y con ello la identificación de las actividades automatizables. Tomando como entrada el estudio realizado se procedió a la identificación de los requisitos funcionales y no funcionales del sistema. Los elementos anteriores permitieron la estructuración del diagrama de casos de uso del sistema aplicando los patrones de casos de uso CRUD Parcial, Extensión o Inclusión Concreta: Extensión y Concordancia: Reutilización. Se definieron las características arquitectónicas del sistema a partir de la selección de elementos tales como patrones arquitectónicos y de diseño. Como parte de la realización de los casos de uso y del modelado de la estructura y comportamiento del sistema, se realizaron los diagramas de clases del diseño y de secuencia mostrando a un nivel más bajo la forma en que será implementado el sistema. De manera general se generaron y describieron los artefactos necesarios para dar paso a la fase de implementación del sistema.

## **CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA**

En la descripción de la metodología de desarrollo usada para guiar el proceso de desarrollo se planteó que el software debía quedar estructurado en componentes. En el presente capítulo se muestran cuáles fueron estos componentes a través de un diagrama para su mejor entendimiento.

Después del diseño del sistema se efectúa la implementación del sistema para posteriormente verificar la solución. Los elementos teóricos empleados para dicha verificación y los resultados de la misma son añadidos a las comprobaciones realizadas a los requisitos y el diseño.

### **3.1 Diagrama de componentes**

Un diagrama de componentes muestra la organización y dependencias entre los diferentes componentes de un software. Son usados para mostrar la implementación estática de un sistema. Incluye el modelado de elementos concretos tales como ejecutables, bibliotecas de clases, tablas, archivos y documentos. En esencia, son diagramas de clases enfocados en los componentes de un sistema. Son importantes para la visualización, especificación y documentación de sistemas basados en componentes y para implementación de sistemas ejecutables (55). En la Figura 9 se muestra el diagrama de componentes del sistema, evidenciándose cómo quedan lógicamente separados en las diferentes capas arquitectónicas mencionadas y descritas anteriormente.

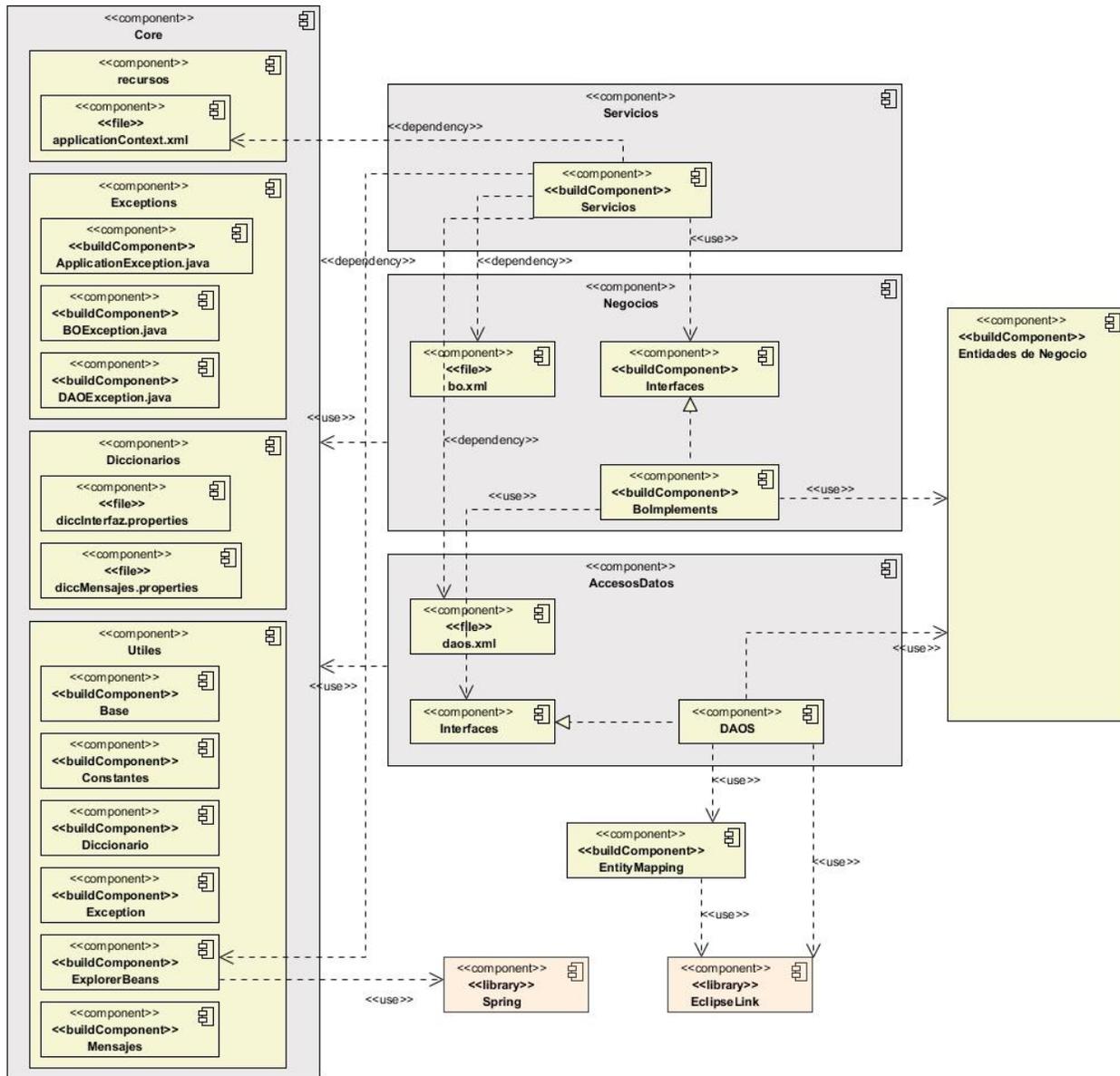


Figura 9: Diagrama de componentes del sistema.

### 3.2 Validación del modelado de los procesos de negocio.

Después de la realización del diseño o modelado de un proceso de negocio es necesario validar el mismo. Una de las formas o instrumentos más utilizados para llevar a cabo esta tarea es la realización de talleres, mediante el cual las personas involucradas discuten el diseño realizado. Los participantes en estos talleres revisan si todas las especificaciones de los procesos de negocio han sido correctamente representadas en los modelos de procesos del negocio. (14)

*Resultados de la validación del modelado de los procesos de negocio.*

Utilizando la técnica descrita anteriormente se realizaron varios encuentros con los clientes en los que sucesivamente se fueron modelando y refinando los procesos de negocio identificados. Finalmente quedaron correctamente modelados los procesos de acuerdo a la conformidad expresada por parte de los clientes y el equipo de desarrollo. Como constancia se tienen las cartas de aceptación por parte del cliente correspondientes a los procesos en cuestión, ver Anexos # 6 y 7.

### **3.3 Validación de requisitos**

Según Pressman: “La validación de requisitos examina las especificaciones para asegurar que todos los requisitos del sistema han sido establecidos sin ambigüedad sin inconsistencias, sin 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.” (15)

#### **Validación de requisitos mediante prototipos**

La construcción de prototipos comienza con la recolección de los requisitos de software. Durante este momento en el proceso de desarrollo, se realiza una representación de los aspectos del software que serán visibles por el usuario, por ejemplo: formatos de entrada y salida de datos. Los prototipos son evaluados por los clientes con el objetivo de refinar los requisitos. (15)

Como parte de las técnicas empleadas en la validación de requisitos se emplearon los prototipos de interfaz de usuario. Se diseñó una propuesta inicial utilizando la herramienta Visual Paradigm. Dicha propuesta incluyó un prototipo por cada una de las interfaces que se definieron para cada requisito. A partir de esta versión inicial, se realizaron sucesivos encuentros con los clientes en los que se fueron refinando los requisitos conjuntamente con los prototipos. Finalmente se llegó a un acuerdo con los clientes, donde éstos aceptaron los prototipos que resultaron del proceso mencionado anteriormente. Para más detalles, consultar los Anexos # 2, 6 y 7.

### **3.4 Métricas de software**

En el campo de la Ingeniería de Software los ingenieros deben emplear diferentes métodos que les permita evaluar la calidad del software de forma concreta. Esta medición de los resultados es uno de los aspectos más importantes y por tanto de tener en cuenta. Los datos obtenidos de

la aplicación de métricas de software asisten la toma de decisiones, definen con gran exactitud la calidad del producto, evalúan la productividad de las personas y los beneficios por el uso de nuevos métodos y herramientas. (61)

### 3.4.1 Métrica de la calidad de la especificación

Se utilizó la métrica de la calidad de la especificación donde los requisitos del sistema fueron revisados en varias iteraciones por los miembros del equipo de calidad del proyecto.

*Métrica de la calidad de la especificación.*

Para determinar la especificidad de los requisitos se sugiere una métrica basada en la consistencia de la interpretación de los revisores para cada requisito:

$$nr = nf + nnf \quad \begin{array}{l} nf : \text{Número de requisitos funcionales} \\ nnf : \text{Número de requisitos no funcionales.} \end{array}$$

$$Q1 = \frac{nui}{nr}$$

nui : Número de requisitos para los que todos los revisores tuvieron interpretaciones idénticas. Cuanto más cerca de uno este el valor de Q1 menor será la ambigüedad de la especificación.

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

*Tabla 3: Equipo de revisión.*

<b>Nombre y apellidos</b>	<b>Rol desempeñado</b>
Jorge Yuniel Jorin Perdomo	Jefe de Proyecto
Ana Cecilia Labrador	Analista Principal
Daniel Varona	Administrador de Calidad
Yoenia María	Analista
Marel Orta Pérez	Director Jurídico de la Cámara de Comercio

*Resultados de la aplicación de la métrica de la calidad de la especificación.*

A continuación se muestra un resumen de los resultados obtenidos:

Tabla 4: Interpretación de los revisores para cada requisito.

Atributo de calidad	Tipo de requisito	Interpretaciones	
		Iguales	Desiguales
Especificidad	Funcionales	25	4
	No funcionales	20	3
	Total	45	7

$$Q1 = \frac{nui}{nf + nnf} = \frac{45}{29 + 23} = 0.8$$

El resultado obtenido es bastante cercano a la unidad por lo que se puede afirmar que existe poca ambigüedad en la especificación de los requisitos. Los resultados obtenidos se muestran en el siguiente gráfico:

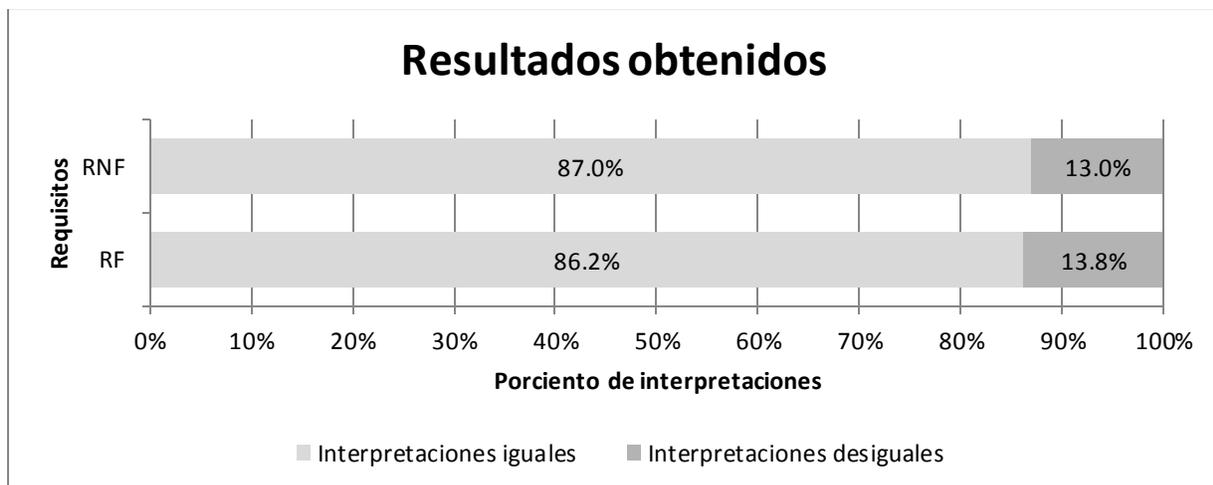


Figura 10: Resultados de la métrica de la calidad de la especificación.

### 3.4.2 Métrica para la calidad de la funcionalidad del diagrama de casos de uso del sistema

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

- *Compleitud*: Grado en que se ha incluido de forma clara y concisa todos los elementos necesarios para la descripción.

- *Consistencia*: Grado en que los elementos del DCUS representan en forma única y no contradictoria un aspecto del problema.
- *Correctitud*: Grado de adecuación del DCUS para satisfacer los requisitos.
- *Complejidad*: Grado de claridad y reuso del DCUS.

Cada uno de estos atributos es evaluado a través de un conjunto especificado de métricas. A continuación se muestra la tabla de las métricas correspondientes a cada uno de los atributos especificados anteriormente:

Tabla 5: Métricas correspondientes a los atributos de calidad.

Atributo	No.	Métricas
<b>Complejidad</b>	1	Número de casos de uso que no tienen descripción resumida.
	2	Número de casos de uso que tienen requerimientos omitidos.
	3	Número de casos de uso que no poseen una descripción extendida.
	4	Número de casos de uso que tienen acciones del flujo de eventos no redactados en función del responsable.
	5	Número de casos de uso que no describen condiciones de excepción relevantes.
	6	Número de casos de uso que no han sido clasificados.
<b>Consistencia</b>	7	Número de casos de uso que tienen un nombre incorrecto.
	8	Número de casos de uso que no representan una interacción observable por un actor.
	9	Número de casos de uso que tienen acciones del flujo de eventos asignados a un responsable que no le corresponde.
	10	Número de casos de uso no aceptados.
	11	Número de casos de uso complejos que no tienen separación del flujo básico y de flujos alternos.
	12	Número de casos de uso que no tienen un usuario responsable.
<b>Correctitud</b>	13	Número de casos de uso en que los requerimientos representados no son comprensibles por el usuario.
	14	Número de casos de uso que deben ser modificados para adecuarlos a la funcionalidad del sistema.
	15	Número de casos de uso que deben ser modificados para mejorar el proceso actual.
<b>Complejidad</b>	16	Número de elementos del diagrama que requieren reubicación.

Resultados de la aplicación de la métrica para la calidad de la funcionalidad del diagrama de casos de uso del sistema.

A continuación se muestra una tabla con los resultados después de aplicar esta técnica de validación:

Tabla 6: Resultados de la aplicación de la métrica para la calidad de la funcionalidad del diagrama de casos de uso del sistema.

Métrica	Cantidad de CU analizados	Resultado	Porcentaje de error
1	22	Actualmente todos poseen una descripción resumida.	0%
2	22	Se detectó un CU que presentaba requerimientos omitidos.	4.5%
3	22	Se detectaron 3 casos de uso que no presentaba su descripción extendida.	13.6%
4	22	Todos están redactados en función del responsable que le corresponde a cada uno.	0%
5	22	No se detectó ningún caso de uso que no describiera condiciones de excepción relevantes.	0%
6	22	Todos han sido clasificados según los diferentes tipos de prioridad.	0%
7	22	Se detectaron 10 casos de uso que presentaron problemas en cuanto al nombre dado.	45.5%
8	22	Todos representan correctamente la interacción con un actor.	0%
9	22	Ninguno presentó acciones del flujo de eventos que estuvieran asignados a un responsable que no le corresponde.	0%
10	22	Todos fueron aceptados.	0%
11	22	Se detectaron tres casos de uso que no tienen separación del flujo básico y de flujos alternos separadas.	13.6%
12	22	Todos poseen un usuario responsable.	0%
13	22	Todos son comprensibles por el usuario.	0%
14	22	Un caso de uso debe ser modificado para adecuarlo a la funcionalidad del sistema.	4.5%
15	22	Tres casos de uso tuvieron que ser modificados para mejorar el proceso actual.	0%
16	22	Ningún elemento requiere reubicación.	0%

Se realizó una primera revisión al diagrama de casos de uso del sistema con el objetivo asegurar que el mismo tenga la calidad requerida. En esta primera revisión se detectó que el diagrama presentaba algunas dificultades en los atributos de correctitud, consistencia y completitud, específicamente en las métricas 2, 3, 7, 11, 14 y 15.

Se realizó por ende una segunda revisión después de efectuar las modificaciones pertinentes en base a los problemas detectados. Los resultados fueron positivos garantizando de esta manera la calidad de la funcionalidad del diagrama de casos de uso del sistema.

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

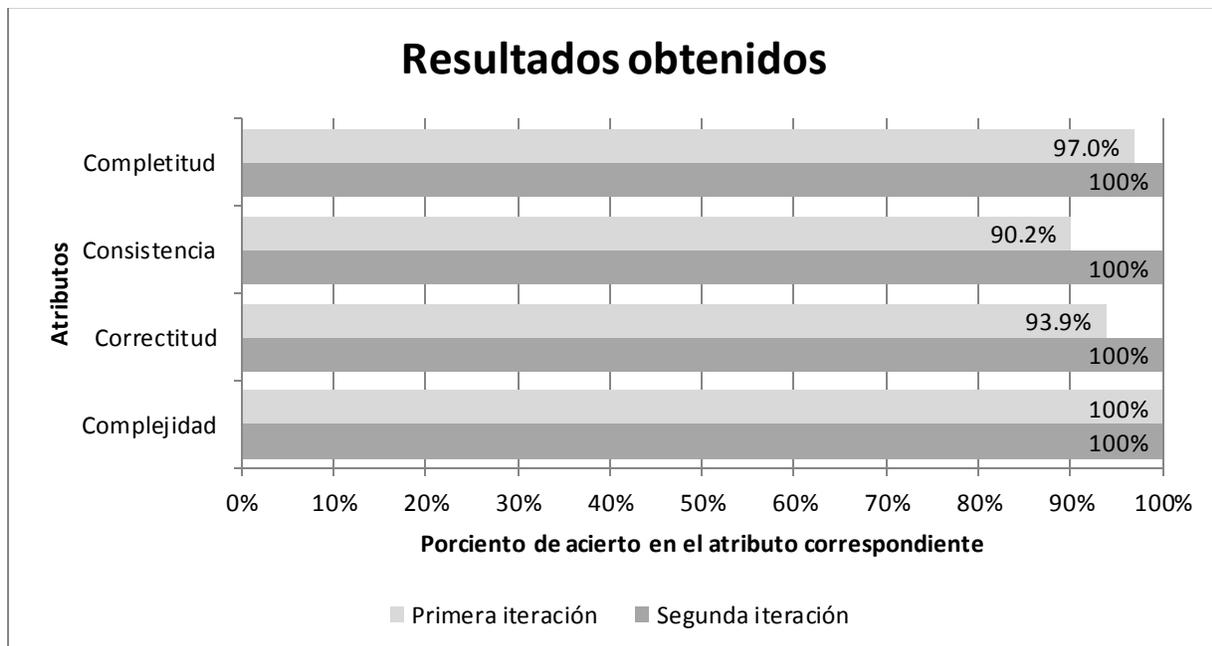


Figura 11: Resultados de la evaluación de la calidad en los diagramas de casos de usos.

### 3.5 Listas de chequeo

Se utilizaron listas de chequeo para garantizar la fiabilidad de los requisitos de software, así como la descripción de los modelos de procesos de negocio. Se realizaron tres iteraciones de revisiones por parte del equipo de calidad a nivel de proyecto, detectándose un total de 7 no conformidades en los modelos de procesos y 13 en la especificación de requisitos. Todas las no conformidades encontradas durante las iteraciones fueron solucionadas. Las listas de chequeo aplicadas pueden ser consultadas en el Anexo # 5.

### 3.6 Revisión por pares

La revisión por pares es una técnica para validar los documentos generados en el proceso de desarrollo de software. La comprobación es realizada por los miembros del proyecto que posean igual rango o rol.

*Resultados de la revisión por pares:*

Tabla 7: Resultados de la revisión por pares.

Documentos	Cantidad de no conformidades	Resultados por iteraciones		
		1ra Iteración	2da Iteración	3ra Iteración
Descripción de los procesos de negocio	4	2 resueltas	1 resuelta	1 resuelta
Especificación de requisitos de software	16	9 resueltas	5 resueltas	2 resueltas
Descripción de casos de uso del sistema	10	7 resueltas	2 resueltas	1 resuelta
Prototipos de interfaz de usuario	8	5 resueltas	3 resueltas	-

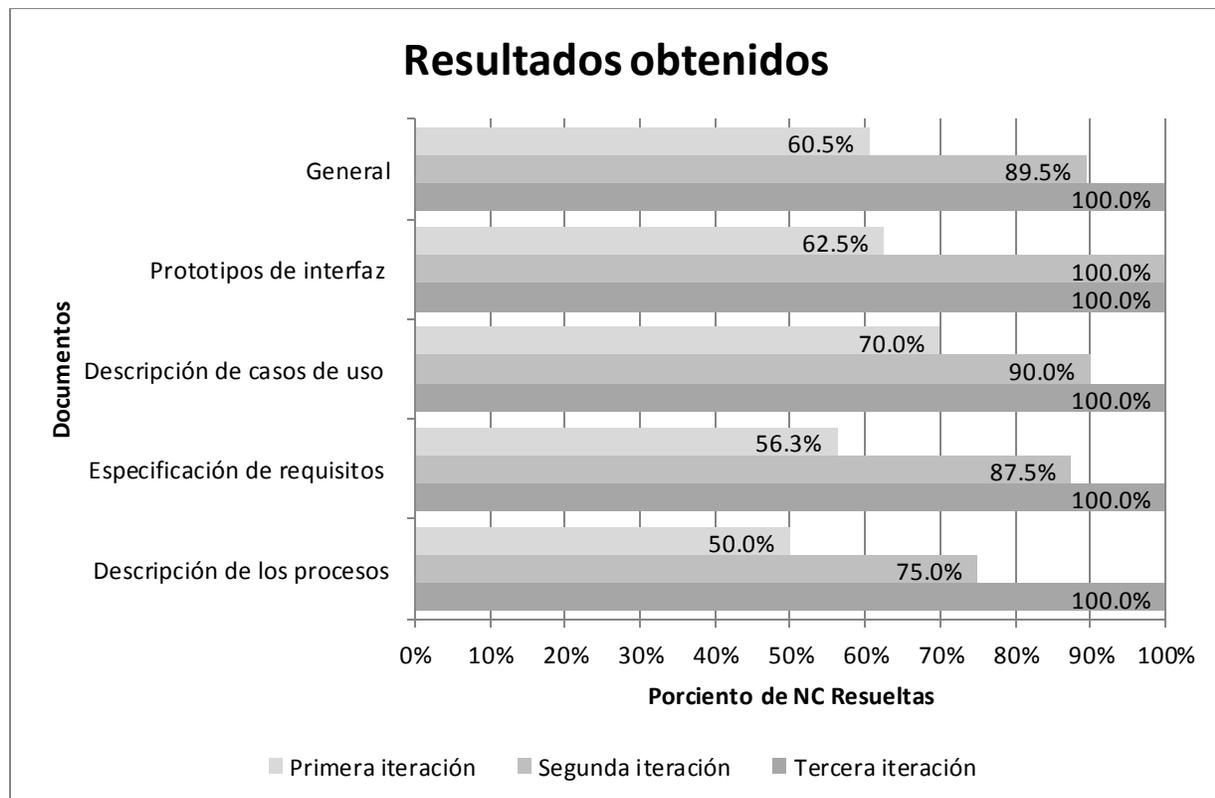


Figura 12: Resultados obtenidos en la revisión por pares.

### 3.7 Pruebas de caja negra

Las pruebas de caja negra o de comportamiento se enfocan en los requisitos funcionales del software. Permiten ejercitar y probar todos los requisitos funcionales a través de un conjunto de datos de entrada seleccionado con esa finalidad. Las pruebas de caja negra intentan encontrar errores que no pueden ser detectados en las de caja blanca. Con el objetivo de llevar a cabo este tipo de pruebas se realizan un conjunto de casos de prueba que permiten tanto la detección de errores como la reducción de la cantidad de casos de prueba a realizar. Este tipo de comprobaciones se realiza fundamentalmente sobre las interfaces de la aplicación con vista al usuario, por tanto, no consideran la estructura interna del sistema. (15)

Las pruebas de caja negra se llevan a cabo a través de diferentes técnicas, las mismas son:

- *Los métodos de pruebas basados en grafos:* Verifican que todos los objetos tienen entre ellos las relaciones esperadas.
- *La partición equivalente:* Divide el campo de entrada de datos en clases de datos de los que se pueden derivar casos de prueba, la idea es encontrar de manera más rápida errores que normalmente serían detectados después de la ejecución de muchos casos de prueba o de una cantidad relativamente mayor.
- *El análisis de valores límite:* Complementan la técnica de partición equivalente, se centran en probar los casos “extremos” de las clases y se obtienen casos de prueba no solo para los campos de entrada, sino también para los campos de salida.
- *Prueba de comparación:* Se utilizan en los casos donde se prueban y comparan los resultados de dichas pruebas, de varias implementaciones del mismo software. Lo anterior se debe a que determinados tipos de software necesitan ser completamente fiables, fundamentalmente, sistemas que deben ofrecer una respuesta en tiempo real.
- *La prueba de la tabla ortogonal:* Suministra un método sistemático y eficiente para probar sistemas con un número reducido de parámetros de entrada.

De las técnicas mencionadas anteriormente se utilizará la de particiones equivalentes para la realización de casos de prueba que permitan comprobar la funcionalidad del sistema.

#### **Diseño de casos de prueba basado en requisitos.**

Un caso de prueba especifica una forma de probar un sistema indicando la entrada y las condiciones bajo las cuales se ejecutará. Existen diferentes casos de prueba que permiten

probar un caso de uso o un escenario específico de un caso de uso. Incluyen la verificación del resultado de la interacción entre los actores y el sistema. (16)

Un buen caso de prueba es aquel que tiene una alta probabilidad de mostrar un error no descubierto hasta el momento, o de encontrar el mayor número de errores con la mínima cantidad de esfuerzo y tiempo posible. (15)

#### *Resultados de las pruebas a través de los casos de prueba basados en requisitos*

Durante la realización de los casos de prueba por parte del equipo de calidad del proyecto para el requisito “Inscribir Agencia de Viajes”, se detectaron un total de 10 no conformidades, 9 de ellas, clasificadas como no críticas, en la entrada y validación de datos y una no conformidad crítica en el envío de notificaciones vía correo electrónico. Las mismas fueron solucionadas en su totalidad por el equipo de desarrollo. Los casos de prueba diseñados podrán ser consultados en el Anexo # 4.

### **3.8 Conclusiones parciales**

Se aplicaron diferentes técnicas y métricas para la validación de los modelos de procesos así como los requisitos y artefactos del software. Los talleres realizados con los clientes permitieron revisar y aprobar los modelos de procesos de negocio. También se validaron los requisitos a través de prototipos de interfaz de usuario y la aplicación de métrica de la calidad de la especificación, la cual demostró que existe poca ambigüedad en la especificación de los mismos. Durante la aplicación de la métrica para la calidad de la funcionalidad del diagrama de casos de uso del sistema se detectaron varias dificultades que fueron corregidas por el equipo de desarrollo, garantizando de esta manera la funcionalidad de este artefacto. Las listas de chequeo permitieron garantizar la fiabilidad de los requisitos, casos de uso y modelos de procesos de negocio. Las mismas fueron aplicadas en varias iteraciones para revisar la eliminación de las no conformidades detectadas. La técnica de revisión por pares fue muy útil para homogeneizar la forma de trabajo del equipo de desarrollo comprobando al mismo tiempo la correcta realización de los documentos generados en el proceso de desarrollo de software. Con el empleo de casos de prueba basado en requisitos utilizando la técnica de particiones equivalentes, se detectaron y corrigieron las no conformidades encontradas fundamentalmente en la entrada y validación de los datos en las interfaces gráficas de la aplicación. De manera general, las métricas de software y técnicas relacionadas anteriormente permitieron identificar y solucionar las inconsistencias detectadas elevando la calidad de los artefactos generados, por lo que se obtuvieron resultados satisfactorios en el proceso.

## CONCLUSIONES GENERALES

Una vez concluida la investigación se arribaron a las siguientes conclusiones:

- El estudio de las aplicaciones en otros países para resolver problemáticas similares planteó la necesidad de la construcción de un sistema propio de uso interno para la gestión de los procesos de la Cámara de Comercio. Dichos sistemas no cumplen con los requerimientos y limitaciones de la entidad cubana y/o las diferentes políticas establecidas en la universidad y el centro de desarrollo que fueron expresadas en los criterios de selección de las herramientas y tecnologías a usar en la implementación de la solución.
- Se realizó el modelado de los procesos de negocio como punto de partida para posteriormente identificar los requisitos siguiendo la guía para el proceso de desarrollo de software RUP. Se identificaron un total 29 requisitos funcionales y 23 requisitos no funcionales. Lo anterior permitió definir y especificar el diagrama de casos de uso del sistema en el que fueron representados 22 casos de uso, a partir del cual se definieron los elementos arquitectónicos y de diseño de la solución.
- Se realizó la implementación de los requisitos de los procesos Registro de Agencias de Viajes y Registro de Empresas Importadoras y Exportadoras.
- Se aplicaron pruebas y métricas que permitieron detectar y solucionar dificultades encontradas en los modelos de procesos de negocio, los requisitos, el diagrama de casos de uso y en la entrada y validación de datos a la aplicación. La calidad de los modelos de procesos de negocios y la especificación de requisitos fue validada validados las cartas de aceptación del cliente.

## RECOMENDACIONES

Vencido el objetivo general de la investigación y teniendo en cuenta la experiencia y el conocimiento adquirido durante la misma, se plantean las siguientes ideas como recomendaciones de este trabajo:

- Desarrollar e incorporar al sistema un manual de usuario a modo de documentación en línea o ayuda.
- Realizar otra serie de casos de prueba basado en requisitos para comprobar las funcionalidades a nivel de negocio de la aplicación así como pruebas de unidad e integración.
- Utilizar el enfoque a procesos en el desarrollo de nuevas soluciones en el Centro de Gobierno Electrónico (CEGEL) y en la universidad. Ello implica la forma de funcionamiento del sistema que se especifica en los casos de uso y el uso de un gestor de procesos de negocio para facilitar el proceso de desarrollo.
- Tomar el presente trabajo como material de estudio y documentación acerca del Sistema de Información Registral de la Cámara de Comercio.

**BIBLIOGRAFÍA**

1. **Cámara de Comercio de la República de Cuba.** *Portal de la Cámara de Comercio de la República de Cuba.* [En línea] 2012. [Citado el: 10 de Febrero de 2012.] <http://www.camaracuba.cu/>.
2. **Cámara de Comercio de Madrid.** Cámara de Comercio de Madrid. *Sitio web de la Cámara de Comercio de Madrid.* [En línea] [Citado el: 21 de Febrero de 2012.] <http://www.camaramadrid.es>.
3. **Cámara de Comercio de Bogotá.** Cámara de Comercio de Bogotá. *Sitio web de la Cámara de Comercio de Bogotá.* [En línea] [Citado el: 21 de Febrero de 2012.] [http://aplica.ccb.org.co/ccbconsultas/consultas/RUE/consulta\\_empresa.aspx](http://aplica.ccb.org.co/ccbconsultas/consultas/RUE/consulta_empresa.aspx).
4. **Cámara de Comercio Bilbao.** Cámara de Comercio Bilbao. *Sitio web de la Cámara de Comercio Bilbao.* [En línea] [Citado el: 21 de Febrero de 2012.] <http://www.camarabilbao.com>.
5. **Cámara de Comercio de Quito.** Cámara de Comercio de Quito. *Sitio web de la Cámara de Comercio de Quito.* [En línea] [Citado el: 21 de Febrero de 2012.] <http://www.lacamaradequito.com>.
6. **Cámara de Comercio de Lima.** *Sitio web de la Cámara de Comercio de Lima.* [En línea] 2012. [Citado el: 21 de Febrero de 2012.] <http://www.camaralima.org.pe/>.
7. **Veronica de la Morena.** Connexions. [En línea] 27 de Noviembre de 2008. [Citado el: 13 de Enero de 2012.] <http://cnx.org/content/m17457/latest/>.
8. **ISO 9126.** "Software product evaluation – Quality characteristics and guidelines for their use". 2005.
9. **MasterMagazine.** MasterMagazine. [En línea] 2004. [Citado el: 12 de Enero de 2012.] <http://www.mastermagazine.info/termino/4720.php>.
10. **ISO.** *Orientación acerca del enfoque basado en procesos para los sistemas de gestión de la calidad.* 2001.
11. **Grupo Kaizen Calidad.** GestioPolis.com. *sitio web de GestioPolis.com.* [En línea] Octubre de 2005. <http://www.gestiopolis.com/canales5/ger/gksa/61.htm>.
12. **Delgado, Andrea.** *Desarrollo de Software con enfoque en el Negocio.* Montevideo : s.n., 2007.
13. **Fernández Pereda, Héctor.** BuscarPortal. [En línea] 2012. [http://www.buscarportal.com/articulos/iso\\_9001\\_enfoque\\_procesos.html](http://www.buscarportal.com/articulos/iso_9001_enfoque_procesos.html).
14. **Weske, Mathias.** *Business Process Management.* Berlin Heidelberg : Springer-Verlag, 2007. 978-3-540-73521-2.
15. **S. Pressman, Roge.** *Ingeniería del Software. Un enfoque práctico.* s.l. : McGraw Hill Higher Education, 2002. 9701054733.
16. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James.** *El Proceso Unificado de Desarrollo de Software.* Madrid : Addison Wesley, 2000. 84-7829-036-2.

17. **Kruchten, Philippe.** *The Rational Unified Process An Introduction, Second Edition.* s.l. : Addison Wesley, 2000. 0-201-70710-1.
18. **Rumbaugh, James, Jacobson, Ivar y Booch, Grady.** *The Unified Modeling Language Reference Manual.* s.l. : Addison Wesley Longman, Inc., 1999. 0-201-30998-X.
19. **Muzaurieta Ladrón de Guevara, Denise y Sánchez-Parodi Montoto, Ramón.** GestioPolis.com. [En línea] 2004. [Citado el: 28 de Febrero de 2012.] <http://www.gestiopolis.com/recursos3/docs/ger/procymodela.htm>.
20. **Alvarado Castillo, Alejandra.** *BONITA SOFT: Gestor de procesos de negocios BPM.* Colombia : s.n., 2011.
21. **Víctor Javier Madrid.** Adictos al trabajo. [En línea] 13 de 08 de 2010. [Citado el: 14 de 02 de 2012.] <http://www.adictosaltrabajo.com/index.php>.
22. **INSTITUTO NACIONAL DE ESTADISTICA E INFORMATICA.** *Herramientas Case.* Peru : Oficina Técnica de Difusión Estadística y Tecnología Informática del Instituto Nacional de Estadística e Informática, 1999. 875-99-OI-OTDETI-INEI.
23. **Visual Paradigm International.** Visual Paradigm International. *Sitio web de Visual Paradigm.* [En línea] [Citado el: 23 de Febrero de 2012.] <http://www.visual-paradigm.com/product/vpuml/>.
24. **Valdez Altamirano, Ing. Alfonso.** Ubicuos. [En línea] <http://ubicuos.com/wp-content/uploads/2009/05/comparativoides.pdf>.
25. Version Cero. [En línea] [Citado el: 24 de Febrero de 2012.] <http://www.versioncero.com/noticia/210/visual-paradigm-for-uml>.
26. **Visual Paradigm International.** Visual Paradigm. *Sitio web de Visual Paradigm.* [En línea] [Citado el: 24 de Febrero de 2012.] [www.visual-paradigm.com/netbeansIntegrationTutorial.php](http://www.visual-paradigm.com/netbeansIntegrationTutorial.php).
27. **Enciclopedia Libre Universal en Español .** Enciclopedia Libre Universal en Español . [En línea] [Citado el: 27 de Febrero de 2012.] [http://enciclopedia.us.es/index.php/Lenguaje\\_de\\_programaci%C3%B3n](http://enciclopedia.us.es/index.php/Lenguaje_de_programaci%C3%B3n).
28. **Holzner, Steven.** *La Biblia de Java 2.* España : Anaya Multimedia, 2003. 84-415-1037-7.
29. **Escuela de Informática de Ourense.** Área de Ciencias de la Computación e Inteligencia. *Sitio web del Área de Ciencias de la Computación e Inteligencia.* [En línea] 2009. [Citado el: 26 de Febrero de 2012.] <http://ccia.ei.uvigo.es/docencia/SCS/Tema5-1.pdf>.
30. **Oracle.** [En línea] [Citado el: 2012 de Febrero de 2012.] <http://java.sun.com/j2ee/overview.html>.
31. —. Java. [En línea] [Citado el: 25 de Febrero de 2012.] <http://java.com/es/download/faq/techinfo.xml>.
32. —. Java. [En línea] [Citado el: 26 de Febrero de 2012.] <http://java.sun.com/javase/overview/compatibility.jsp>.
33. **Área de Ciencias de la Computación e Inteligencia.** [En línea] [Citado el: 27 de Febrero de 2012.] <http://ccia.ei.uvigo.es/docencia/SCS/Tema5-1.pdf>.

34. **Universidad de Huelva.** [En línea] [Citado el: 26 de Febrero de 2012.] [http://www.uhu.es/josel\\_alvarez/NvasTecnProg/recursos/tTema1.pdf](http://www.uhu.es/josel_alvarez/NvasTecnProg/recursos/tTema1.pdf).
35. **Rivera Pineda, Julian.** [En línea] [Citado el: 26 de Febrero de 2012.] <http://xtecuan.org/pdfs/IntroAplicacionesWebJava-UGB.pdf>.
36. **Alfredo Voos, Javier.** [En línea] [Citado el: 20 de Febrero de 2012.] <http://neurotests.frc.utn.edu.ar/Tests/Docu/Tests.pdf>.
37. **Mateu, Carles.** *Desarrollo de aplicaciones web.* Barcelona : Eureka Media, SL, 2004. 84-9788-118-4.
38. Castro, Jon; Escolano, Jonathan. [En línea] [Citado el: 23 de Febrero de 2012.] <http://studies.ac.upc.edu/FIB/CASO/seminaris/2q0203/T7%20-%20J2EE.pdf>.
39. **Universidad de Huelva.** Universidad de Huelva. [En línea] 2007. [Citado el: 23 de Febrero de 2012.] [http://www.uhu.es/josel\\_alvarez/NvasTecnProg/recursos/tTema1.pdf](http://www.uhu.es/josel_alvarez/NvasTecnProg/recursos/tTema1.pdf).
40. **editum.** editum. [En línea] [Citado el: 20 de Febrero de 2012.] <http://www.editum.org/Que-Es-Un-Servidor-De-Aplicaciones-p-473.html>.
41. **Microsoft Coporation.** Microsoft. [En línea] [Citado el: 25 de Febrero de 2012.] [http://technet.microsoft.com/es-es/library/cc728199\(v=ws.10\).aspx](http://technet.microsoft.com/es-es/library/cc728199(v=ws.10).aspx).
42. **Consultor Java.** Consultor Java. [En línea] [Citado el: 19 de Febrero de 2012.] <http://www.consultoriajava.com/tools/tomcat.shtml>.
43. **Netbeans.** Netbeans. [En línea] [Citado el: 25 de Febero de 2012.] [http://netbeans.org/index\\_es.html](http://netbeans.org/index_es.html).
44. **Böck, Heiko.** *The Definitive Guide to NetBeans Platform.* s.l. : Apress, 2009. 978-1-4302-2418-1.
45. **Aplicaciones Empresariales.** Aplicaciones Empresariales. [En línea] [Citado el: 20 de Febero de 2012.] <http://www.aplicacionesempresariales.com/utilizando-framework-de-programacion-en-las-empresas.html>.
46. **Keith, Mike, Schincariol y Merrick.** *Pro JPA 2. Mastering the Java Persistence API.* New York : Apress, 2009. 978-1-4302-1957-6.
47. **Tepper Fisher, Paul y D. Murphy, Brian.** *Spring Persistence with Hibernate.* s.l. : Apress, 2010. 978-1-4302-2633-8.
48. **Chen-Becker, Derek, Danciu, Marius y Weir, Tyler.** *The Definitive Guide to Lift.* s.l. : Apress, 2009. 978-1-4302-2422-8.
49. **Sanderson, Dan.** *Programming Google App Engine.* s.l. : O'Reilly Media, Inc., 2010. 978-0-596-52272-8.
50. **Walls, Craig y Breidenbach, Ryan.** *Spring in action.* s.l. : Manning Publications Co., 2008. 1-933988-13-4.
51. **Johnson et al., Rod.** *Professional Java Development with the Spring Framework.* s.l. : John Wiley & Sons, 2005. 0764574833.

- 
52. **S. Mullins, Craig.** *Database Administration: The Complete Guide to Practices and Procedures.* s.l. : Addison Wesley, 2002. 0-201-74129-6.
53. **The PostgreSQL Global Development Group.** *PostgreSQL 9.0.1 Documentation.* California : Regents of the University of California, 2010.
54. **Riggs, Simon y Krosing, Hannu.** *PostgreSQL 9 Administration Cookbook.* Birmingham : Packt Publishing Ltd., 2010. 978-1-849510-28-8.
55. **Booch, Grady, Rumbaugh, James y Jacobson, Ivar.** *The Unified Modeling Language User Guide.* Massachusetts : Addison-Wesley Longman Inc., 1998. 0-201-57168-4.
56. **Övergaard, Gunnar y Palmkvist, Karin.** *Use Cases Patterns and Blueprints.* s.l. : Addison Wesley Professional, 2004. 0-13-145134-0.
57. **Camacho, Erika, Cardeso, Fabio y Núñez, Gabriel.** *Arquitecturas de Software.* 2004.
58. **Buschmann, Frank, y otros, y otros.** *Pattern – Oriented Software Architecture. A System of Patterns.* Inglaterra : John Wiley & Sons, Inc., 1196. 978-0471958697.
59. **Bachmann, Felix, y otros, y otros.** *Software Architecture Documentation in Practice: Documenting Architectural Layers.* Pittsburgh : Carnegie Mellon University, 2000.
60. **E, Gamma, y otros, y otros.** *Design Patterns. Elements of Reusable Object-Oriented.* s.l. : Addison Wesley, 2003.
61. **Perez, Isabel.** *Métricas para el control de proyectos de software.* Ciudad de la Habana : s.n., 2002.

## **ANEXOS**

### **ANEXO 1. Documento Modelo de Procesos de Negocio con BPM**

Consultar los documentos de nombre “0115\_Modelo de Procesos de Negocio con BPM” en la carpeta Artefactos de los procesos de Registro de Agencias de Viajes y Registro de Empresas Importadoras y Exportadoras.

### **ANEXO 2. Documento Especificación de Requisitos de Software**

Consultar documentos con el nombre “0113\_Especificación de Requisitos de Software.doc” en la carpeta Artefactos.

### **ANEXO 3. Especificación de casos de uso**

Consultar documentos con el nombre “0114 Especificación de casos de uso.doc” en la carpeta Artefactos.

### **ANEXO 4. Diseño de casos de pruebas basado en requisitos**

Consultar documentos con el nombre “0122 Diseño de casos de pruebas basado en requisitos” en la carpeta Artefactos.

### **ANEXO 5. Listas de chequeo**

Consultar los documentos con el nombre “Listas de chequeo” en la carpeta Artefactos.

## ANEXO 6. Acta de aceptación del cliente. Proceso Registro de Agencias de Viajes.



## Acta de aceptación

## ACTA DE ACEPTACIÓN

En cumplimiento del **Convenio de colaboración con la Cámara de Comercio** y en función de la ejecución del Proyecto Sistema Informatización Registral de la Cámara de Comercio, se hace entrega del producto que se relaciona a continuación.

Lista de productos que serán aceptados:

- Módulo Registro de Agencias de Viajes.
- Modelo de proceso de negocio con BPMN.
  - Especificación de Requisitos de Software.

<b>Entrega</b>	<b>Recibe</b>
Proyecto Sistema Informatización Registral de la Cámara de Comercio.	Cámara de Comercio
<b>Nombre y Apellidos:</b> <i>Ara Cecilia Labrador Valdés</i>	<b>Nombre y Apellidos:</b> <i>Marel Orta Pérez</i>
<b>Cargo:</b> Analista Principal	<b>Cargo:</b> <i>Directora Jurídica</i>
<b>Firma:</b> <i>[Firma]</i>	<b>Firma:</b> <i>[Firma]</i>
<b>Fecha:</b> <i>31/5/2012</i>	
<b>Comentarios:</b>	

## ANEXO 7. Acta de aceptación del cliente. Proceso Registro de Empresas Importadoras y Exportadoras.



### Acta de aceptación

#### ACTA DE ACEPTACIÓN

En cumplimiento del **Convenio de colaboración con la Cámara de Comercio** y en función de la ejecución del Proyecto Sistema Informatización Registral de la Cámara de Comercio, se hace entrega del producto que se relaciona a continuación.

Lista de productos que serán aceptados:

Módulo Registro de Importadores y Exportadores.

- Modelo de proceso de negocio con BPMN.
- Especificación de Requisitos de Software.

<b>Entrega</b> Proyecto Sistema Informatización Registral de la Cámara de Comercio.	<b>Recibe</b> Cámara de Comercio
<b>Nombre y Apellidos:</b> Ana Cecilia Labrador Valdés	<b>Nombre y Apellidos:</b> Marel Orta Pérez
<b>Cargo:</b> Analista Principal	<b>Cargo:</b> Directora Jurídica
<b>Firma:</b> 	<b>Firma:</b> 
<b>Fecha:</b> 31/5/2012	
<b>Comentarios:</b>	

## GLOSARIO DE TÉRMINOS

**BPM:** *Business Process Management*, Gestión de Procesos de negocio.

**BPMN:** *Business Process Modeling Notation*, notación para el modelado de procesos de negocio.

**CASE:** *ComputerAided Software Engineering*, Ingeniería de Software Asistida por Ordenador.

**Requerimientos:** Condición o necesidad de un usuario para resolver un problema o alcanzar un objetivo.

**RUP:** *Rational Unified Process*, metodología que guía el proceso de desarrollo de software.

**UML:** *Unified Modeling Language*, lenguaje de modelado de sistemas de software. Permite visualizar, especificar, construir y documentar un sistema.

**RF:** Requisito funcional.

**RNF:** Requisito no funcional.

**CU:** Caso de uso.

**CUS:** Casos de uso del Sistema

**DCUS:** Diagrama de casos de uso del Sistema

**DCD:** Diagrama de Clase del Diseño

**DPN:** Diagrama de Procesos de Negocio

**DS:** Diagrama de Secuencia