

Universidad de las Ciencias Informáticas

Facultad #3



*Título: Aplicación para la generación de documentos XML para la
Ventanilla Única de Comercio Exterior de Cuba*

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

Autor: Helen Polanco Rodríguez.

Tutores: Ing. Yaniris Blanco Zamora.

Ing. Juan Javier Dans Moreno.

La Habana, Junio de 2013

Año 55 de la Revolución

PENSAMIENTO



"El futuro de nuestra patria tiene que ser necesariamente un futuro de hombres de ciencia, tiene que ser un futuro de hombres de pensamiento, porque precisamente es lo que más estamos sembrando; lo que más estamos sembrando son oportunidades a la inteligencia (...)"

Fidel Castro Ruz

DECLARACIÓN DE AUTORÍA

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos 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 _Junio_ del año __2013__.

Helen Polanco Rodríguez

Ing. Yaniris Blanco Zamora

Firma del Autor

Firma del Tutor

Ing. Juan Javier Dans Moreno

Firma del Tutor

DATOS DE CONTACTO

Autor: Helen Polanco Rodríguez

- Estudiante de la Universidad de las Ciencias Informáticas.
- Correo electrónico hpolanco@estudiantes.uci.cu.

Tutor(a): Ing. Yaniris Blanco Zamora.

- Ingeniera en Ciencias Informáticas graduada en el 2008 en la Universidad de las Ciencias Informáticas.
- Correo electrónico ybzamora@uci.cu

Tutor(a): Ing. Juan Javier Dans Moreno

- Ingeniero en Ciencias Informáticas graduada en el 2011 en la Universidad de las Ciencias Informáticas.
- Correo electrónico jjdans@uci.cu.

AGRADECIMIENTOS

Antes que todo y que todos le agradezco a mi madre Yoandra Rodríguez Díaz y Noemí Díaz Sánchez por haberme convertido en la mujer que hoy soy.

A mi tío Bartolo por ser el mejor padre que la vida me pudo dar.....

A su esposa María Elena por su apoyo incondicional en todo momento....

A toda mi familia que de una manera u otra ha aportado un granito de arena para que haya llegado este día...

A mis tutores J y Yaniris por su eterna paciencia....

A mi tribunal por haber hecho con sus críticas constructivas en todo momento (cosa que no imagine) que me formara de la mejor manera posible y llegara ante esta defensa con todos los argumentos necesarios para defenderme..

A Adrian y a Weeden por su ayuda incondicional y su extrema paciencia, nunca me voy a olvidar de ustedes (que dió), a Yorlen, a Lazaro, a todo mi grupo de proyecto....

A mi mejor amiga en estos 5 años Nolvís un millón de gracias por estar ahí cuando más lo necesite, con personas como tú se puede andar confiado...

A Danny por su comprensión y apoyo en todo momento de esta recta final y más difícil, muchas gracias mi vida, nunca voy a olvidar lo que has hecho por mí...

A Disnel, por ser un gran amigo, a Polanco, Ramón (contigo sin palabras), Pepe, Yanerki, Analie, Gisel, Joisel y su potrico, Tania, Yanet...A todo mi grupo de manera general y todos los que aportaron su granito para que llegara este día les estaré eternamente agradecida

Un millón de gracias.

DEDICATORIA

Les dedico este trabajo de diploma a las tres personas más importantes del mundo hoy para mí...

A mi madre Yoandra, a mi segunda madre: mi tía Noemí y a mi hermanito que es la cosa más grande que dios me ha dado, los amo a los tres y hoy más que por mí esto va por ustedes ...

Resumen

La Aduana General de la República de Cuba ha tomado conciencia de la necesidad del uso intensivo de internet para la gestión de sus procesos de negocios, razón por la cual se desarrolla en la Universidad de las Ciencias Informáticas (UCI) el Sistema de Gestión Integral de Aduana (GINA), con el objetivo de automatizar los procesos aduaneros del país.

En la realización de los trámites para efectuar una operación comercial, los involucrados deben acudir a varias instituciones, organismos y entidades solicitando permisos y/o liberaciones que le avalen la operación frente a las autoridades comerciales del país, provocando la pérdida de tiempo, el gasto de divisas, así como la insatisfacción de los clientes. Dada esta situación surge la Ventanilla Única de Comercio Exterior (VUCE) que se define como un mecanismo de facilitación que permite a las partes involucradas, gestionar a través de medios electrónicos los trámites requeridos por las entidades para el tránsito, ingreso o salida del territorio nacional de mercancías.

El objetivo principal de la presente investigación es desarrollar un sistema que permita crear la documentación asociada a la VUCE para satisfacer las necesidades del cliente. Para ello, se emplearon las tecnologías y herramientas establecidas por el Departamento de Soluciones para la Aduana: Visual Paradigm para UML, BPMN, Axure RP, NetBeans y Java fue el lenguaje escogido para su desarrollo. Con este propósito, se realiza un estudio acerca de la gestión de la información que realizan algunos sistemas de Ventanilla Única existentes en el mundo.

Como resultado de la presente investigación se obtuvo un sistema que permite la creación y envío de la documentación referente a Despacho Comercial de la VUCE de Cuba. Además garantiza una mayor calidad en los procesos de recepción de documentos de la Ventanilla Única de Comercio Exterior del país.

Palabras claves: informatización de procesos aduaneros, generación de documentos XML, Ventanilla Única.

Índice

Capítulo I Fundamentación Teórica.....	15
1.1 Introducción	15
1.2 Estado actual.....	15
1.2.1 Ventanilla Única de Comercio Exterior de la República del Ecuador	16
1.2.2 Ventanilla Única de Comercio Exterior de Colombia.....	16
1.2.3 Ventanilla Única de Comercio Exterior de la República de Corea	17
1.2.4 Ventanilla Única de Comercio Exterior de España	18
1.2.5 Ventanilla Única de Comercio Exterior de Canadá.....	18
1.3 Valoraciones de los sistemas de Ventanilla Única estudiados.....	19
1.4 Documento digital.....	19
1.5 Lenguaje de marcado	20
1.5.1 XML	20
1.6 Servicios Web	21
1.6.2 SOAP	22
1.6.3 Web Service Description Language (WSDL).....	22
1.7 Aplicación de escritorio.....	23
1.7.1 Ventajas de las aplicaciones de escritorio.	23
1.8 Modelo de desarrollo de software.....	24
1.8.1 Inicio o Estudio preliminar:.....	24
1.8.2 Desarrollo:	25
1.9 Técnicas, lenguajes y herramientas utilizadas.	26
1.9.1 Programación Orientada a Objetos (POO)	26
1.9.2 Lenguajes de Programación.	26
1.10 Herramientas de desarrollo	30
1.10.1 Visual Paradigm para UML.....	30
1.10.2 NetBeans IDE 7.2.1	30
Conclusiones parciales	30

Capítulo II Descripción de la solución propuesta.....	32
2.1 Introducción	32
2.3 Especificación de los requisitos del sistema.....	33
2.3.1. Técnicas para la captura de requisitos.....	33
2.3.2. Requisitos funcionales	33
2.3.3 Técnicas para la Validación de requisitos.....	47
2.3.4 Requisitos no funcionales.....	48
2.4 Patrones de diseño utilizados.....	48
2.4.1 Patrones GRASP	49
2.5 Arquitectura de software.....	51
2.6 Diseño del sistema.....	52
2.6.1 Diagrama de clase de la aplicación.....	52
2.6.2 Diagrama de secuencia.....	53
2.6.3 Diagrama de paquetes.....	55
2.7 Métricas para la evaluación del diseño.....	56
2.7.1 Métricas propuestas por Lorenz y Kidd.....	56
2.7.2 Métrica de Tamaño Operacional de Clase (TOC).....	57
2.8 Aportes de la solución y beneficios esperados.....	58
Conclusiones Parciales	58
Capítulo III: Implementación y pruebas al sistema.....	60
3.1 Introducción	60
3.2 Estándares de codificación para java.....	60
3.3 Tratamiento de Errores	63
3.4 Diagrama de despliegue.....	64
3.5 Validación de la solución	65
3.5.1 Técnicos de prueba.....	65
3.6 Escenario de caso de prueba.....	66
3.7 Conclusiones parciales.....	68
Conclusiones.....	69
Referencias Bibliográficas	71
Anexos	75

Glosario de términos.....	78
---------------------------	----

Introducción

La ARG fue fundada el 5 de febrero de 1963, constituye el órgano que vela y regula el control aduanero aplicable a la entrada, el tránsito, el cabotaje, el trasbordo, el depósito y la salida del territorio nacional de mercancías, viajeros y sus equipajes, bienes y valores sujetos a regulaciones especiales, así como los medios en los que se transporten. Constituye además parte de la Administración Central del Estado y se subordina al Consejo de Ministros. (1)

Dado el auge actual de las Tecnologías de la Información y las Comunicaciones (TIC), las entidades que manejan gran cantidad de información y procesos, se han sumado a una carrera de informatización de sus mecanismos de trabajo, la AGR es una de ellas, pues con el crecimiento continuo actual del comercio internacional y la interdependencia económica que vincula y norma las relaciones entre estados, se hace necesario para una buena aplicación de las políticas aduaneras, contar con un personal altamente calificado junto a una tecnología y sistemas adecuados, que se correspondan con las necesidades actuales.

Con el surgimiento de la Universidad de las Ciencias Informáticas (UCI) en el año 2002, se abrieron nuevas puertas a varias instituciones nacionales y extranjeras para su incursión en la informatización de sus procesos, entre ellas se encuentra la AGR, la cual a pesar de tener el apoyo del Centro de Automatización para la Dirección y la Información (CADI)¹; solicita la colaboración de la UCI en el marco de la ardua labor que realizan en pos de mejorar las negociaciones de comercio.

Las negociaciones de Facilitación del Comercio son las más avanzadas en el contexto del Programa de Doha para el Desarrollo de la Organización Mundial del Comercio (OMC). Una de las medidas en negociación es el establecimiento de una Ventanilla Única de Comercio Exterior (VUCE).

Esta se define como el sistema integrado que permite a los involucrados en las operaciones de comercio exterior y transporte internacional gestionar, a través de Internet, por una sola vía y en un solo lugar, los trámites requeridos por las entidades gubernamentales competentes para la exportación, importación y tránsito de mercancías. Se trata de una herramienta digital donde el usuario puede realizar –en línea y sin papeles- la totalidad de sus trámites y pagos de las

¹CADI: Centro de Automatización para la Dirección y la Información, centro dedicado al desarrollo de soluciones informáticas para la aduana, en él se concentran los especialistas de la rama y que tienen precisamente como propósito la creación de aplicaciones que intervengan en los diferentes procesos aduaneros.

operaciones de comercio exterior, servicio virtual que está disponible las veinte y cuatro (24) horas del día, los siete (7) días de la semana, durante todo el año. A partir de un encuentro realizado en el año 2011 entre los directivos del Departamento de Soluciones para la Aduana perteneciente al Centro de Informatización de la Gestión de Entidades (CEIGE) de la UCI y el CADI se concluyó iniciar el proyecto de desarrollo “Ventanilla Única de Comercio Exterior de Cuba (VUCE)”.

La realización de los trámites en la VUCE se efectúa mediante formularios y presentación de documentos XML, lo cual hace necesario que las entidades involucradas en las operaciones de comercio exterior y transporte internacional cuenten con acceso a Internet y sistemas que le permitan la creación de documentos XML según el estándar definido por la VUCE. En la actualidad dichas entidades poseen un conjunto de soluciones informáticas para garantizar la gestión de sus procesos pero ninguna cuenta con una funcionalidad que permita la creación de los documentos XML necesarios para la realización de los trámites. Como consecuencia la obtención de los documentos XML por parte de las entidades se vuelve un proceso engorroso e ineficiente, en numerables ocasiones la información no se tiene en el momento preciso y hace necesario contar con trabajadores capacitados provocando gastos en recursos.

Partiendo de la situación expuesta con anterioridad se ha identificado como **Problema a resolver:** El proceso de creación de documentos XML en las entidades involucradas en las operaciones de comercio exterior y transporte internacional presenta deficiencias en su ejecución.

Se plantea como **objetivo general:** Desarrollar una aplicación que permita a las entidades crear los documentos XML para la Ventanilla Única de Comercio Exterior de Cuba facilitando la realización de los trámites requeridos.

A partir de la problemática planteada anteriormente se puede establecer el **objeto de estudio** de la presente investigación, el cual se centra en los proceso de creación de documentos XML.

Se determina como **campo de acción:** el proceso de creación de documentos XML para la Ventanilla Única de Comercio Exterior.

Objetivos específicos:

- ✓ Elaborar el marco teórico referente a las soluciones de software que gestionan la

creación de documentos XML.

- ✓ Identificar, los requisitos de la aplicación.
- ✓ Describir los requisitos de la aplicación.
- ✓ Validar los requisitos de la aplicación.
- ✓ Modelar el diseño de la aplicación para definir las clases y componentes necesarios para su implementación.
- ✓ Implementar una aplicación que gestione la creación de documentos XML.
- ✓ Realizar pruebas a la aplicación.

Para dar cumplimiento a estos objetivos fueron trazadas las siguientes **tareas en la investigación:**

- ✓ Análisis del estado del arte de las soluciones de software que gestionan la creación de documentos XML.
- ✓ Análisis de la documentación referente al lenguaje de marcado extensible (XML).
- ✓ Análisis de la metodología de desarrollo de software, lenguajes, herramientas de modelado y tecnologías utilizadas en el proyecto.
- ✓ Definición de los requisitos funcionales del sistema a desarrollar.
- ✓ Realización de la descripción de los requisitos.
- ✓ Diseño de la solución descrita.
- ✓ Validación técnica y funcional del diseño obtenido.
- ✓ Implementación de la aplicación.
- ✓ Diseño de casos de prueba.
- ✓ Verificación la solución mediante pruebas funcionales.

Idea a defender:

Si se desarrolla la aplicación que permita crearlos documentos XML para la Ventanilla Única de Comercio Exterior de Cuba, se facilitará la realización de los trámites requeridos.

Para el desarrollo exitoso de este trabajo de diploma se utilizaron algunos de los **métodos científicos de la investigación**, como los **métodos teóricos** que seguidamente se relacionan.

- ✓ **Histórico-Lógico** se aplica para investigar la existencia de plataformas que generen ficheros con características similares a la solución que se propone en el presente trabajo.
- ✓ **Analítico – sintético:** con el objetivo de realizar un estudio detallado de las teorías, tendencias y documentos relacionados con una aplicación que facilite la creación de los documentos por las entidades y los usuarios externos.
- ✓ **Método de Modelación** :se emplea para mostrar los diferentes diagramas y componentes que se construyen como resultado del proceso de Ingeniería de Software

Métodos Empíricos

- ✓ **Entrevista:** con el objetivo de obtener información sobre las necesidades reales que tiene la AGR, para que el sistema cumpla de forma completa con las mismas.

De la presente investigación, se espera como resultado una aplicación de escritorio que permita la generación de documentos en formato XML para la realización de operaciones de Despacho Comercial y Medios de Transporte Internacional a través de la VUCE.

Capítulo 1: abarca lo relacionado con la fundamentación teórica de la investigación. Se enuncian los principales conceptos que se abordan en la investigación. Se realiza además un estudio de las soluciones al problema planteado, así como de las herramientas, metodologías y lenguajes a usar en el desarrollo de la solución.

Capítulo 2: en este capítulo se aborda el diseño de la solución, describiendo a su vez varios artefactos que se generan durante dichas fases. Se describe además la definición de los requerimientos del sistema, los diagramas de secuencia, el diagrama de clase del sistema, entre otras puntualizaciones.

Capítulo 3: implementación y validación de la solución propuesta; teniendo como base los artefactos concebidos en la etapa de diseño para la implementación de la solución y posteriormente haciendo uso de las distintas técnicas de evaluación de software, para determinar posibles errores y corregirlos así como saber si se cumplió o no con los objetivos propuestos.

Capítulo I Fundamentación Teórica.

1.1 Introducción

En el presente capítulo se describen los elementos principales que sirven de fundamentación para el contenido de la investigación. Se realiza un estudio de las actuales soluciones informáticas para las ventanillas únicas a nivel internacional en la generación de los documentos legales. Se hará una breve descripción de las herramientas, lenguajes y notación de modelado para el desarrollo del producto.

1.2 Estado actual

El desarrollo de los sistemas de VUCE en el mundo es un tema que ha venido madurando con el paso del tiempo. Estos sistemas brindan servicios de importación y exportación mediante trámites legales sin necesidad de la presencia física en el órgano oficial. Para esto se precisa la autenticación que sería para estos casos una firma digital, teniendo la necesidad de usar mayormente la vía de formularios para la realización de algún tipo de trámite ya que un gran número de personas no cuentan con la experiencia necesaria para general los documentos con el formato oficial y usar la vía del envío de archivo XML. Esta opción hoy es brindada por la VUCE de Cuba se ve dificultado su uso.

Otras soluciones

El comercio exterior es un pilar importante para el desarrollo económico mundial, por lo que cada país busca fortalecerlo a partir de acciones que permitan incrementar su competitividad y crecimiento. Los estados miembros del Sistema Económico Latinoamericano y del Caribe (SELA) realizan un significativo esfuerzo por fomentar sistemas que impulsen la agilidad y transparencia del comercio exterior y en este punto vale resaltar el desarrollo de las VUCE en los países que lo integran.

Teniendo en cuenta que el mayor número de relaciones comerciales de Cuba es con los países pertenecientes al SELA entre los que se encuentran: Colombia, Ecuador y Chile se realiza un estudio de las VUCE de estos países en su mayoría sin dejar de tener en cuenta países como Canadá y España. En el caso específico de la Ventanilla Única de Ecuador, está recibiendo apoyo del gobierno coreano a través de la firma KTNET². Es precisamente la Ventanilla Única

²KTNET: por sus siglas en inglés KoreaTrade Network. Es un proveedor de servicio de Ventanilla Única.

(VU) de Corea uno de los líderes mundiales en esta área, razón por la cual se toma en cuenta para el estudio a realizar. Los sistemas seleccionados implementan en su totalidad el mismo modelo de Ventanilla Única.

Para un correcto enfoque del estudio, se determinan los siguientes indicadores:

- ✓ Codificación de la información
- ✓ Forma de entrada de la información.
- ✓ Formato en que se recibe la información.
- ✓ Forma de envío de notificaciones.

1.2.1 Ventanilla Única de Comercio Exterior de la República del Ecuador

Para la gestión de la información, la elaboración y entrega de los documentos electrónicos en esta VU podrán realizarse de dos formas: mediante el acceso a los formularios que brinda el portal o a través de un software propio que poseerá el declarante. Dicho software presenta al usuario formularios que le permiten crear los documentos electrónicos, envía toda la información cifrada y firmada digitalmente. Se recibe en formato XML y con un número de entrega, el cual está compuesto por el Código del Operador de Comercio Exterior (8) + Año (4) + Series (8) + Clasificación de medio de transmisión de la declaración: Software o Portal (1): en total 21 dígitos (ejemplo: **01000880 2011 12345678 P**). (2)

Una vez recibida la información la VU se encarga de verificar los documentos de envío, descifrar, verificar la firma electrónica y tramitar la recepción de los documentos electrónicos. El trámite de la recepción de los documentos electrónicos consiste en la interconexión con la Base de Datos Interna de la VU, validación de los documentos, trámite operativo y generación de notificaciones de error o datos.

1.2.2 Ventanilla Única de Comercio Exterior de Colombia

La evolución de la VUCE en Colombia ha sido notable desde el 2004, contándose al momento con una estrategia comunicacional para su aplicación. Mediante la VUCE se realizan trámites relacionados con los requisitos, certificaciones, permisos y vistos buenos requeridos en las operaciones de importación y exportación. La misma emplea la firma electrónica avanzada para todas las transacciones que se realizan a través de ella. Además de utilizar un intercambio estándar de documentos a través del formato XML. (3)

Existen dos formas de transmisión de la información encaminadas a satisfacer las necesidades del usuario.

CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA

- *Sistema VUCE Web*: funciona vía Internet; desde éste el usuario consulta el arancel de vistos buenos, radica las solicitudes, firma electrónicamente en línea, realiza los pagos en línea y hace seguimiento al trámite para determinar el estado del mismo.
- *Aplicativo VUCE*: herramienta de intercambio de datos entre los sistemas de información de los importadores y el sistema VUCE Web. Los importadores generan desde sus propios aplicativos un archivo texto de acuerdo con una estructura válida; el aplicativo VUCE toma los datos de estos archivos planos, los convierte a estructura XML y luego lo firma, para transmitirlo al Sistema VUCE Web.

A través de la aplicación que utilizan los usuarios se realiza la conversión a formato XML de los archivos planos, se firma y se transmiten al Sistema VUCE Web. Vía correo se le informa al usuario del estado de las operaciones.

Existen estados predefinidos según las operaciones que se realizan, estos pueden ser:

- ✓ Transmisión vía aplicación (Pendiente de pago, Estado de error).
- ✓ Trámites y documentos para entidades externas (Requisitos al importador, No requiere, Negada, Aprobada).
- ✓ Trámites de exportaciones (Iniciar trámites, Trámites en proceso, Trámites concluidos).

1.2.3 Ventanilla Única de Comercio Exterior de la República de Corea

Como resultado de los esfuerzos realizados por parte de los Servicios Aduaneros de Corea, en 1998, el KCS³ construyó un exitoso sistema en los que todas las aplicaciones relacionadas con el despacho y la declaración de documentos, incluidos los de importación/exportación, gestión del transporte, devoluciones, entre otros procesos que se llevan a cabo en la Aduana y se procesan electrónicamente. (4)

El proceso de gestión de la información es similar a otras VUCE estudiadas. Los agentes de carga envían los informes de arribo/partida en formato XML, la ventanilla se encarga de verificar los certificados de estos agentes y enviar los informes hacia los organismos receptores para que los aprueben. Por otro lado, los agentes importadores y exportadores realizan la entrega de las declaraciones en la VU, esta se encarga de verificarlas y confirmarlas con las agencias pertinentes. Al sistema administrativo aduanero se le notifica el resultado de la declaración/arribo; este emite una notificación para que se le comunique a los agentes de carga

³KCS: por sus siglas en inglés Korea Customs Service.

que fue aceptado el informe y a los agentes importadores y exportadores el resultado del despacho.

1.2.4 Ventanilla Única de Comercio Exterior de España

En España, el servicio de VUCE fue impulsado gracias al acuerdo del Consejo de Ministros del 15 de julio del 2005, el cual establece que “el ministro de Administraciones Públicas, en ejercicio de sus competencias, debe coordinar las acciones para establecer progresivamente una red de espacios comunes de atención al ciudadano que permitiera ofrecer sobre determinados aspectos de la actividad administrativa, servicios integrados de información, orientación, atención y tramitación en el ejercicio de las competencias respectivas entre departamentos, organismos o administraciones”.(5)

La VUCE de España pone a su disposición los siguientes servicios:

Ofrece un centro de atención en Internet, a través del cual los ciudadanos y las empresas pueden efectuar las gestiones que venían realizando de forma presencial, en cualquiera de las oficinas del gobierno. El conjunto de servicios que se brindan, corresponde a los diversos niveles administrativos existentes en España: estatales, autónomos y locales.

En el caso de esta VUCE no presentan un método para la realización de trámites de otra manera que no sea mediante la atención online que ofrece el portal o por la presencia física de un representante de la entidad ante una oficina de VU. Por tal razón no será tomado en consideración el mecanismo de recensión de documento de esta ventanilla.

1.2.5 Ventanilla Única de Comercio Exterior de Canadá.

Los servicios ofrecidos van desde la oferta de información hasta la posibilidad de llevar a cabo una interacción completa en tiempo real. El 30% de los ciudadanos o empresas que utilizan los servicios del gobierno de Canadá recurren a Internet y otras tecnologías para llevar a cabo sus trámites.

En el año 2003, el éxito se plasmaba en el aumento de un 38% en las visitas en relación con el 2002, así como los avances alcanzados en los trabajos interdepartamentales. Algo especialmente relevante fue la implementación del portal inalámbrico, el cual permite el acceso a los servicios por medio de teléfonos celulares. Estos son los medios con que cuenta dicha ventanilla, para realizar sus trámites debido a que la conectividad para ellos no constituye un problema no se ven en la necesidad de implementar un sistema como la propuesta de solución que se plantea en el presente trabajo.(6)

1.3 Valoraciones de los sistemas de Ventanilla Única estudiados

A partir del estudio realizado y de acuerdo al desarrollo del comercio electrónico que posee Cuba, se pueden reutilizar características como es el envío de la información encriptada mencionada, solo que nuestra aplicación es propiamente de las leyes cubanas, quedando desechadas las soluciones estudiadas por el motivo de que toda la configuración de interfaz de usuario y lógica del negocio cambiaría adecuándola a las necesidades para la realización de las operaciones de Despacho Comercial y Medios de Transporte Internacional específicamente. La gestión de los trámites a través de software es algo propio de las VUCE estudiadas por lo que a partir de los mismos se determinaron características similares que poseen estos sistemas pudiéndose utilizar en la aplicación propuesta, entre ellas se encuentran:

- La información viaja codificada.
- Para el procesamiento de la información, el documento debe encontrarse en formato XML.
- Según el proceso en el que se encuentra inmerso el documento, existen estados predefinidos para mostrar como notificaciones.
- Las notificaciones se muestran mediante el sistema de VU y mensajes de correo electrónico.

1.4 Documento digital

Son aquellos en los cuales la información está registrada en formato electrónico, sobre un soporte electrónico, que requiere de dispositivos informáticos para la consulta.(7)

Para empezar se puede tomar lo que se considera documento, entonces si se toma la definición expuesta por el reglamento general de archivos que define documento como “Información registrada, cualquiera sea su forma o el medio utilizado.”(8) Entonces si se dice “que una información es digital cuando está codificada en el formato que puede interpretar un ordenador y suele decirse que consiste en series de ceros y de unos (0, 1).”(9), Entonces se podría decir que el documento digital es todo aquel que registra y conserva la información en lenguaje digital, es decir, en el formato que puede interpretar un ordenador que consiste en series de unos y ceros.

En este caso es necesario un formato estandarizado que precisamente permita consultarlo independientemente del medio de almacenamiento en el que se ubique y que sea además multiplataforma. A partir del estudio de las VU, se determina usar el estándar XML para la

transferencia de documento digital entre el portal y la aplicación.

1.5 Lenguaje de marcado

1.5.1 XML

XML se define como Lenguaje de Marcado Extensible (***Extensible MarkupLanguage***) es un lenguaje de marcado de carácter general. Creado en 1998 por W3C (***World Wide Web Consortium***), estándar internacional libre y gratuito. (8)

Los lenguajes de marcado son lenguajes informáticos que utilizan marcas o etiquetas para definir la estructura, presentación y/o formato de los textos. El marcado se lleva a cabo añadiendo al contenido original del texto etiquetas con una función o significado determinados, y gracias a ellas es posible que los programas informáticos puedan procesar o interpretar adecuadamente los textos .XML; proponiendo un estándar para el intercambio de información estructurada entre diferentes plataformas. Tiene un papel muy importante en la actualidad ya que permite la compatibilidad entre sistemas para compartir la información de una manera segura y fiable.(8)

Los documentos han de seguir una estructura estrictamente jerárquica con lo que respecta a las etiquetas que delimitan sus elementos. Una etiqueta debe estar correctamente incluida en otra, es decir, las etiquetas deben estar correctamente anidadas. Los elementos con contenido deben estar correctamente cerrados.

- Los documentos XML sólo permiten un elemento raíz del que todos los demás sean parte, es decir, solo pueden tener un elemento inicial.
- Los valores y/o atributos en XML siempre deben estar encerrados entre comillas simples o dobles.
- El XML es sensible a mayúsculas y minúsculas. Existe un conjunto de caracteres llamados espacios en blanco (espacios, tabuladores, saltos de línea) que los procesadores XML tratan de forma diferente en el marcado XML.
- Es necesario asignar nombres a las estructuras, asignar tipos de elementos, entidades y elementos particulares.
- Las construcciones como etiquetas, referencias de entidad y declaraciones se denominan marcas; son partes del documento que el procesador XML espera entender. El resto del documento entre marcas son los datos “entendibles” por las personas.

- XML es extensible pues ofrece al usuario la posibilidad de definir sus propios elementos: el usuario puede definir marcas o etiquetas de su gusto, y estructurar el documento en función de dichas etiquetas., XML más que un lenguaje es un metalenguaje, ya que ofrece la oportunidad de definir y utilizar diferentes lenguajes o aplicaciones.(9)

Objetivo fundamental del uso de XML

El objetivo es intercambiar datos estructurados entre sistemas de información, fundamentalmente a través de Internet. Se trata de un formato de texto plano, lo que facilita enormemente la transferencia de información, logrando independencia con respecto a las diferentes plataformas. Utilizando XML se puede marcar cualquier documento. Además, como acepta codificación Unicode, es válido para cualquier idioma. El estándar Unicode permite codificar electrónicamente caracteres escritos en cualquier lengua. XML es un formato autodocumentado que, además de por máquinas, puede ser leído e interpretado por personas.

Los documentos XML deben tener una correcta formación (**well-formed**), cumpliendo con todas las normas. Este tipo de documentos presenta un modelo de datos jerárquicos el cual está formado por dos estructuras; elementos y atributos. Los atributos se utilizan para la descripción de la información complementaria, lo que es llamado meta información y en el caso de los elementos tienen dos clasificaciones: simples y complejos. Los simples están formados por valores como cadenas de caracteres.(10) En el caso de los complejos están compuesto por varios simples.

1.6 Servicios Web

Existen múltiples definiciones sobre lo que son los Servicios Web; una sería hablar de ellos como un conjunto de aplicaciones o de tecnologías con capacidad para interoperar en la Web. Estas aplicaciones o tecnologías intercambian datos entre sí con el objetivo de ofrecer servicios. Los proveedores ofrecen sus servicios como procedimientos remotos y los usuarios solicitan un servicio llamando a estos procedimientos a través de la Web.(11)

En resumen, son aplicaciones que permiten el intercambio de datos en formato XML. Estos datos pueden ser consumidos por cualquier aplicación, independientemente de las tecnologías usadas tanto para la implementación del servicio como de la aplicación cliente, esto los hace ser multiplataforma y portables. Constituyen la base para establecer la relación de la aplicación con la VUCE. En la **figura 1.1** se evidencia este proceso



Figura 1.1 Servicio Web entre la aplicación y la VUCE.

En todo este proceso intervienen una serie de tecnologías que hacen posible esta circulación de información, por un lado, estaría WSDL y por otro lado SOAP.

1.6.2 SOAP

(Protocolo de Simple Acceso a Objetos); se trata de un protocolo basado en XML, que permite la interacción entre varios dispositivos y que tiene la capacidad de transmitir información compleja. Los datos pueden ser transmitidos a través de HTTP, SMTP, TCP.

SOAP especifica el formato de los mensajes. El mensaje SOAP está compuesto por un sobre, cuya estructura está formada por los siguientes elementos: **header** (cabecera) y **body** (cuerpo).(12)

Características principales:

- ✓ Extensibilidad: Seguridad y WS-routing son extensiones aplicadas en el desarrollo.
- ✓ Neutralidad: SOAP puede ser utilizado sobre cualquier protocolo de transporte como HTTP, SMTP, TCP o JMS.
- ✓ Independencia: SOAP permite cualquier modelo de programación.

1.6.3 Web Service Description Language (WSDL)

WSDL (*Web ServiceDescriptionLanguage*) describe la interfaz pública a los servicios Web. Está basado en XML y describe la forma de comunicación, es decir, los requisitos del protocolo y los formatos de los mensajes necesarios para interactuar con los servicios listados en su catálogo. Las operaciones y mensajes que soporta se describen en abstracto y se ligan después al protocolo concreto de red y al formato del mensaje. Por lo que a menudo es usado WSDL en combinación con SOAP y XML Schema. Un programa cliente que se conecta a un servicio web puede leer el WSDL para determinar que funciones están disponibles en el

servidor. El cliente puede usar SOAP para hacer la llamada a una de las funciones listadas en el WSDL(13)

Estos servicios proporcionan mecanismos de comunicación estándares entre diferentes aplicaciones, que interactúan entre sí presentándole información dinámica al usuario. Para proporcionar interoperabilidad y extensibilidad entre estas aplicaciones, y que al mismo tiempo sea posible su combinación para realizar cualquier tipo de operaciones.

1.7 Aplicación de escritorio

Una aplicación desktop (también llamada aplicación de escritorio) es aquella que está instalada en el ordenador del usuario, que es ejecutada directamente por el sistema operativo, ya sea Microsoft Windows, Mac OS X, Linux o Solaris, y cuyo rendimiento depende de diversas configuraciones de hardware como memoria RAM, disco duro, memoria de video.(14)

1.7.1 Ventajas de las aplicaciones de escritorio.

- ✓ Resuelve el problema de la generación de los ficheros XML de las solicitudes de Despacho comercial y Medios de Transporte Internacional que presentan los usuarios.
- ✓ La aplicación será capaz de realizar muchas de las operaciones de la VUCE sin necesidad de estar conectado a ella; ventaja que le permitirá al usuario subir el archivo cuando lo desee.
- ✓ El usuario está más acostumbrado a la interacción con aplicaciones de este tipo ya que su diseño no está lejos de las aplicaciones propias de cualquier sistema operativo.
- ✓ Para el caso de nuestra aplicación no será necesario una conexión con un gestor de base de datos ya que la aplicación gestiona sus datos a través de ficheros descargados de la VUCE.
- ✓ Aplicaciones muy pesadas como tienden a ser las de escritorios son recomendadas que sean ejecutadas en el lado del cliente.

Una de las funcionalidades que deberá cumplir la aplicación será permitir el envío mediante servicio web la información necesaria para realizar un tipo de trámite determinado. Esto será posible por los servicios que brinda hoy la VUCE de recibir información por esta vía. La información viaja cifrada con una llave pública, descriptando solamente por la persona que contenga la llave privada que en este caso será el proveedor de servicios de la VUCE. La

información tendrá un formato JSON⁴ en dependencia del trámite que se realice será el formato JSON que se le dé.

1.8 Modelo de desarrollo de software.

Para el buen funcionamiento del Centro de Informatización de la Gestión de Entidades (CEIGE) y de los proyectos que pertenecen al mismo se ha establecido el siguiente modelo de desarrollo en el cual se especifican las actividades de cada una de las fases del ciclo de vida de los proyectos. También se detallan los artefactos a generar en cada momento independientemente de las herramientas o métodos que se utilicen para ello. Debido a la buena organización que te brinda este modelo para el desarrollo de un proyecto y por estar establecida su utilización en el proyecto GINA, será utilizada en el desarrollo del presente trabajo. (CEIGE Modelo de desarrollo). A continuación se describen las fases del ciclo de vida de los proyectos.

1.8.1 Inicio o Estudio preliminar:

Los equipos de desarrollo de software que llevan varios años de experiencia, como es el caso del Departamento de Soluciones para la Aduana y el proyecto ERP⁵ de la UCI, han desarrollado sus propios modelos de desarrollo, basados en los distintos tipos de metodologías existentes, ajustándolas y modificándolas, extrayendo de estas aquellos aspectos que se ajustan a las necesidades del proyecto para guiar, agilizar y mejorar la organización del desarrollo productivo.

Durante el inicio del proyecto se llevan a cabo las actividades relacionadas con la planeación del proyecto a un alto nivel, la evaluación de la factibilidad del proyecto y el registro de este. En esta fase se realiza un estudio inicial de la organización cliente que permite obtener información fundamental acerca del alcance del proyecto, realizar estimaciones de tiempo, esfuerzo y costo, y decidir si se ejecuta o no el proyecto.

Los objetivos de la fase son:

- Asegurar la factibilidad del proyecto.

⁴JSON: (**JavaScript ObjectNotation** - Notación de Objetos de JavaScript) es un formato ligero de intercambio de datos.

⁵ERP: Por sus siglas en inglés Enterprise ResourcePlanning.

- Establecer un plan para la ejecución del proyecto.

Hitos:

- Plan de desarrollo de software.
- Acta de inicio del proyecto firmada.

1.8.2 Desarrollo:

En esta fase se ejecutan las actividades requeridas para desarrollar el software, incluyendo el ajuste de los planes del proyecto considerando los requisitos y la arquitectura. Durante el desarrollo se refinan los requisitos, se elaboran la arquitectura y el diseño, se implementa y se libera el producto.

El objetivo de esta fase es:

- Obtener un sistema que satisfaga las necesidades de los usuarios finales.

Hito:

- Producto liberado por entidad certificadora de calidad.

En esta fase se ejecutan las disciplinas modelado de negocio, requisitos, análisis y diseño, implementación, pruebas internas y pruebas de liberación.

En el caso específico de la aplicación como resultados de estas fases se generaron una serie de artefactos para una mejor comprensión del negocio como lo indica el modelo de desarrollo del centro CEIGE, estos artefactos fueron:

- ✓ Especificación de requisitos.
- ✓ Descripción de requisitos.
- ✓ Modelo conceptual.
- ✓ Modelo de diseño.
- ✓ Glosario de términos.
- ✓ Manual de usuario.

- ✓ Diseño de caso de prueba.

1.9 Técnicas, lenguajes y herramientas utilizadas.

La forma en que se especifique y elabore la solución de un problema determinado es a lo que se llama técnica de programación. Las técnicas de programación son aplicables cuando se desea trasladar a la computadora el funcionamiento de un proceso, o la solución a un problema determinado, para ello se realiza una abstracción, o sea, un modelo simplificado de la realidad tomando los elementos más significativos y transformándolos en variables, de esta forma la computadora podrá entenderlo obteniendo así el resultado deseado.

En el caso específico de la aplicación no fue necesario el uso de un gestor de base de datos, puesto que la aplicación correrá del lado del cliente y no tendrá necesidad de administrar sus datos a partir de una base de datos; haciéndolo a través de ficheros XML gestionará la actualización de sus datos. Haciendo de esta manera que la aplicación sea más ligera y con un grado menor de complejidad para el usuario.

1.9.1 Programación Orientada a Objetos (POO)

La POO es un paradigma de la programación de computadores; esto hace referencia al conjunto de teorías, estándares, modelos y métodos que permiten organizar el conocimiento, proporcionando un medio bien definido para visualizar el dominio del problema e implementar en un lenguaje de programación la solución a ese problema.(15)

Dicho paradigma es sin lugar a dudas el más utilizado por las empresas de todo el mundo a la hora de encarar desarrollos de aplicaciones de software, ya que permite representar de manera relativamente simple modelos y realidades muy complejas. Esto hace que el software sea más fácil de programar, comprender y mantener.

1.9.2 Lenguajes de Programación.

Un lenguaje de programación es aquel elemento dentro de la informática que nos permite crear programas mediante un conjunto de instrucciones, operadores y reglas de sintaxis; que pone a disposición del programador para que este pueda comunicarse con los dispositivos hardware y software existentes.

Kernighan y Ritchie (28) describen a C como un lenguaje de programación de propósito

CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA

general. Es muy útil y su aprendizaje a un nivel básico es relativamente sencillo y no requiere de mucho tiempo. Tiene un repertorio de instrucciones básicas relativamente pequeño. Aunque incluye numerosas funciones de biblioteca que mejoran las instrucciones básicas. Además los usuarios pueden escribir bibliotecas adicionales para su propio uso. Es uno de los más usados para la programación de algoritmos genéticos.

C++

Es un lenguaje de programación creado por Bjarne Stroustrup en los años 80. Es una variante de C para la manipulación de objetos. Es un lenguaje imperativo orientado a objetos muy ligado al hardware subyacente, manteniendo una considerable potencia para programación a bajo nivel, pero con elementos que le permiten también un estilo de programación con alto nivel de abstracción. “No es un lenguaje orientado a objetos puro (en el sentido en que puede serlo Java por ejemplo), además no nació como un ejercicio académico de diseño. Se trata simplemente del sucesor de un lenguaje de programación hecho por programadores (de alto nivel) para programadores, lo que se traduce en un diseño pragmático al que se le han ido añadiendo todos los elementos que la práctica aconsejaba como necesarios, con independencia de su belleza o purismo conceptual ha experimentado un extraordinario éxito desde su creación. De hecho, muchos sistemas operativos, compiladores e intérpretes han sido escritos en C++. Es un lenguaje de nivel intermedio, pudiéndose utilizar tanto para escribir software de bajo nivel, como drivers y componentes de sistemas operativos, como para el desarrollo rápido de aplicaciones.

C#

Es un lenguaje de programación orientado a objetos creado por Microsoft como parte de la iniciativa .NET Framework y escrito por Andres Hejlsberg en 1999. La última versión conocida es la 3.0 que salió a la luz el 17 de noviembre del 2007, propiedad de la compañía Microsoft. En su constitución, a diferencia de lenguajes como C / C++, que proporcionan un altísimo grado de control de los procesos permitiendo el uso de punteros y muchas otras funciones de bajo nivel, y otros incluidos dentro de este entorno .NET, como Microsoft Visual Basic, que posee un alto nivel y más facilidad a la hora de desarrollar una aplicación, C#, se encuentra en un término intermedio ya que ayuda a desarrollar aplicaciones rápidas pero que también permiten un gran control e integración con el desarrollo de aplicaciones Web, XML, y muchas otras de las tecnologías recientes. Este lenguaje no genera código nativo y para ejecutar los programas que se realicen, la computadora tiene que tener instalado .NET. Tiene que cargar su CLR

(Common Language Infrastructure) y el Framework. NET cada vez que se ejecuta un programa en C # la carga del mismo es considerablemente mayor que la carga de un programa equivalente en C ++.

Java

El lenguaje de programación Java fue desarrollado por Sun Microsystems a principios de los 90's con la idea original de usarlo para la creación de páginas WEB. Entre noviembre de 2006 y mayo de 2007, Sun Microsystems liberó la mayor parte de sus tecnologías Java bajo la licencia GNU GPL, de tal forma que prácticamente todo el Java de Sun es software libre. Según los datos recopilados por el Índice Comunitario de Programación del 2011 (28), Java es el lenguaje más popular dentro de la comunidad internacional. Es orientado a objetos, agrupa en estructuras encapsuladas tanto sus datos como las funciones que manipulan esos datos. Es interpretado, ya que los bytecodes, un formato intermedio indiferente a la arquitectura, se pueden ejecutar directamente sobre cualquier máquina a la cual se hayan portado el intérprete y el sistema de ejecución en tiempo real. Fue diseñado para crear software altamente fiable, proporciona numerosas comprobaciones en compilación y en tiempo de ejecución. Sus características de memoria liberan a los programadores de una gran cantidad de errores. Java soporta aplicaciones que serán ejecutadas en variados entornos de red, diseñado para transportar el código eficientemente a múltiples plataformas de hardware y software. La máquina virtual, al ejecutar el código java, realiza comprobaciones de seguridad, además el propio lenguaje carece de características inseguras. Toma mucha de su sintaxis de C y C++ pero tiene un modelo de objetos más simple eliminando herramientas como la manipulación directa de punteros o memoria.

Se escoge el lenguaje Java porque tiene muchas ventajas. Ha sido diseñado de modo de eliminar las complejidades de otros lenguajes como C y C++. Es gratuito, y no hay que pagar la licencia para ".NET" como C#. Así mismo es un lenguaje muy completo y poderoso, se pueden realizar muchas tareas con él, pues posee una librería y utilidades muy completas que facilitan la programación para el desarrollo de ficheros XML y el consumo de servicio web. La versión de Java a utilizar sería la 7.0.1 la cual contiene importantes mejoras para el rendimiento, estabilidad y seguridad de las aplicaciones Java que se ejecutan en su equipo. La instalación de la actualización gratuita garantiza que sus aplicaciones Java sigan ejecutándose de forma segura y eficaz.

Complemento Java

El complemento Java es un componente del entorno de ejecución de Java (JRE) (**Java Runtime Environment**). El JRE permite ejecutar en diversos navegadores los applets⁶ escritos en el lenguaje de programación Java. El complemento Java no es un programa independiente y no se puede instalar por separado. El entorno de ejecución tiene como principal función brindar la posibilidad de establecer una relación entre el sistema operativo del usuario y la máquina virtual de java, característica que hace al lenguaje ser multiplataforma.

Se decide utilizar Java como lenguaje de programación para el desarrollo de la aplicación ya que el mismo posee gran flexibilidad, portabilidad, robustez y sencillez al mismo tiempo que será fácil de aprender para las personas que continuarán expandiendo la solución propuesta independientemente de todas las características anteriormente mencionadas. Entre las características principales que le dan ventaja a este lenguaje se encuentran las siguientes:

- Orientado a objetos: Soporta las características esenciales del paradigma de la programación orientada a objetos: encapsulamiento⁷, herencia⁸ y polimorfismo⁹.
- Robusto: Elimina el uso de apuntadores para referenciar áreas de memoria, además, libera al desarrollador de la necesidad de liberar la memoria que la aplicación ya no usa. También requiere la declaración explícita tanto de los tipos de datos como de los métodos.
- Multiplataforma: El mismo código Java que funciona en un sistema operativo, funciona en cualquier otro que tenga instalada la máquina virtual de Java.
- Multitareas: Permite la ejecución concurrente de varios procesos ligeros o hilos de ejecución.
- Permite la comunicación con la web a través del consumo de servicios web.

⁶**applet:** es un componente de software que corre en el contexto de otro programa, por ejemplo un navegador web.

⁷**Encapsulamiento:** Mecanismo que consiste en organizar datos y métodos de una estructura, conciliando el modo en que el objeto se implementa.

⁸**Herencia:** Mecanismo basado en clases, por medio del cual una clase se deriva de otra de manera que extiende su funcionalidad.

⁹**Polimorfismo:** se refiere a la posibilidad de enviar un mensaje a un grupo de objetos cuya naturaleza puede ser heterogénea.

- Presenta librerías propias sin necesidad de plugins adicional para el trabajo con ficheros y en especial los ficheros XML.
- Interfaz agradaba tanto para el usuario final como para el desarrollador.

1.10 Herramientas de desarrollo

1.10.1 Visual Paradigm para UML

Visual Paradigm para UML es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Este es un completo sistema que permitirá aumentar la productividad al momento de desarrollar, para los ingenieros y arquitectos de software es excelente ya que muy fácilmente se puede centralizar y planificar las ideas. (19)

Además de mejorar la productividad en el desarrollo y mantenimiento del software, aumentar la calidad del mismo, reducir el tiempo de desarrollo y mantenimiento. Mejora la planificación de un proyecto, realizar una gestión global en todas las fases de desarrollo de software con una misma herramienta. Esta herramienta es de gran ayuda en el ciclo de vida del software permitiendo a las organizaciones visualizar, diseñar, integrar y distribuir una aplicación.

1.10.2 NetBeans IDE 7.2.1

NetBeans IDE es una aplicación de código abierto ("open source") diseñada para el desarrollo de aplicaciones fácilmente portables entre las distintas plataformas, haciendo uso de la tecnología Java. Dispone de soporte para crear interfaces gráficas de forma visual, desarrollo de aplicaciones web, control de versiones, colaboración entre varias personas, creación de aplicaciones compatibles con teléfonos móviles, resaltado de sintaxis. Por sus disímiles características y ventajas fue escogido para el desarrollo de la aplicación; por la perfecta integración que realiza con los elementos necesarios para la implementación de la aplicación como es el caso del lenguaje escogido para el desarrollo, los ficheros que deberá generar la aplicación, y el consumo de los servicios web.

Conclusiones parciales

En este capítulo se expusieron los principales conceptos y términos por los cuales se rige la investigación, en muchos casos se mencionaron otros conceptos definidos y que se consideran válidos. Se dejó plasmado un análisis del problema identificado por la investigación y la necesidad de proponer una solución a dicho problema. Las herramientas para llevar a cabo el

CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA

modelado y la implementación del sistema fueron objeto de estudio también en el presente capítulo.

Durante la investigación se realizaron estudios en búsqueda de soluciones existentes que pudieran aportar de forma general una respuesta o punto de partida para el problema identificado. El autor del presente trabajo concluye en que la investigación actual constituye la primera solución real con vista a resolver el problema identificado y de forma general la primera de su tipo aplicada al problema específico en cuestión.

CAPÍTULO 2 DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

Capitulo II Descripción de la solución propuesta.

2.1 Introducción

En el presente capítulo se muestra una propuesta del sistema a desarrollar, se describen las principales características del sistema, los patrones de diseño utilizados para la solución de la aplicación. Se exponen los requisitos funcionales definidos para el sistema, enumerando en este sentido las técnicas de captura y validación de requerimientos. Se obtienen además los principales artefactos del diseño que guiarán el posterior proceso de implementación de la solución, haciendo énfasis en el modelo de diseño de la solución.

2.2 Modelo Conceptual.

Un modelo conceptual es una representación de conceptos en un dominio del problema. En UML, se ilustra con los diagramas de clase, donde no se definen operaciones solo conceptos, asociaciones entre conceptos y atributos de conceptos. A continuación se muestra la **figura 1.2** el modelo conceptual de presente trabajo

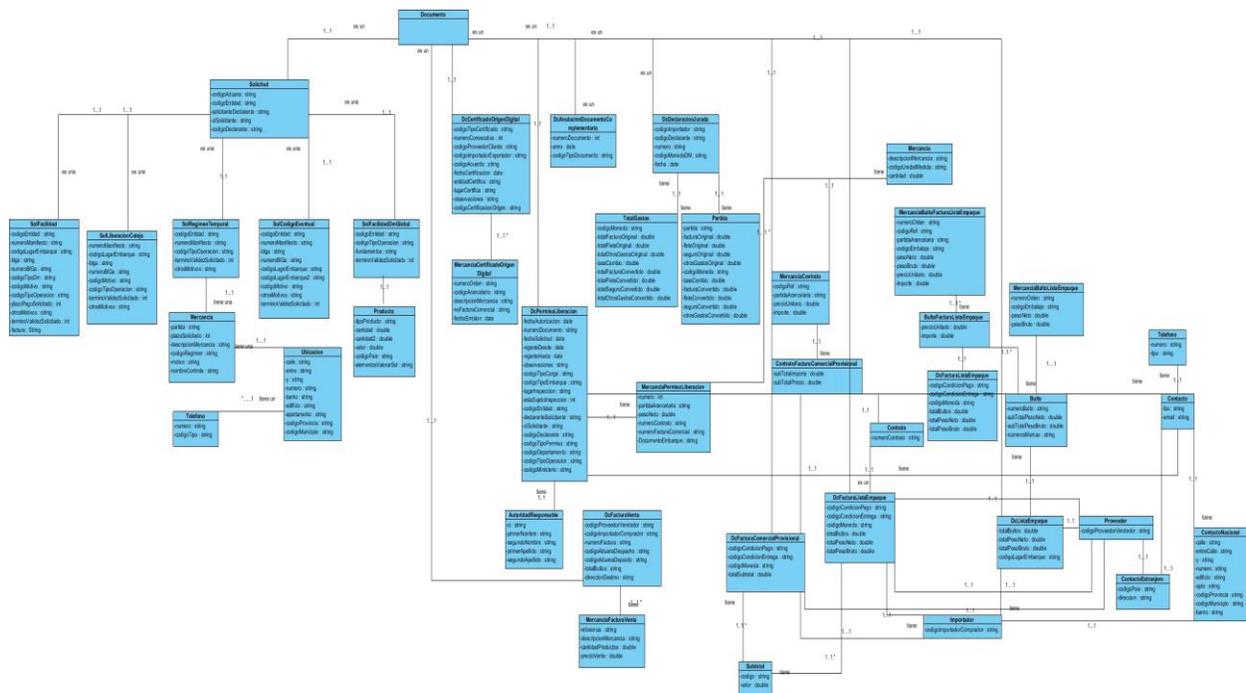


Figura 1.2 Modelo Conceptual.

CAPÍTULO 2 DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

En el modelo conceptual presentado, evidencia la relación que tienen los distintos tipos de solicitudes mediante una solicitud general, igual es el caso de los documentos complementarios Relacionados por una clase documento complementario. Ambos documentos tiene relación entre sí por ser ambos un documento

2.3 Especificación de los requisitos del sistema.

Con la especificación de los requisitos de software se describe cada detalle a tener en cuenta para el desarrollo del sistema lo que ayuda a evitar errores en posteriores etapas. A continuación se muestran las técnicas utilizadas y el listado de los requisitos funcionales definidos para el sistema.

2.3.1. Técnicas para la captura de requisitos.

Para la recopilación y obtención de la información necesaria para la captura de requisitos, se utilizaron las siguientes técnicas: la entrevista, la observación y la tormenta de ideas.

La **entrevista** fue la primera aplicada a la presente investigación, realizándola al jefe de proyecto de la VUCE y conocer sus necesidades. De esta manera se le realizaron preguntas acerca del problema que estaban enfrentando con los documentos que no podían ser generados. La **observación** se llevó a cabo para percibir cómo se desarrollan las operaciones de la VUCE, tomando los principales problemas que están enfrentando en su negocio para darle solución en la medida de lo posible mediante la aplicación. La **tormenta de ideas**, donde en conjunto con la especialista de la aduana que atiende los medios de transporte internacionales y el despacho comercial y el jefe de proyecto de la VUCE, se acumularon ideas para tener una perspectiva general de lo que sería la aplicación.

2.3.2. Requisitos funcionales

Los requisitos funcionales (RF) son declaraciones de los servicios que debe proporcionar el sistema, de la manera en que éste debe reaccionar a entradas particulares y de cómo se debe comportar en situaciones particulares. En algunos casos, los requerimientos funcionales de los sistemas también pueden declarar explícitamente lo que el sistema no debe hacer. (20)

Son capacidades o condiciones que el sistema debe cumplir, describen con detalle la función de éste, sus entradas y salidas, excepciones, entre otros, estos deben estar bien determinados

CAPÍTULO 2 DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

y ser convenientemente comprendidos tanto por los implicados para con el sistema como por los desarrolladores del mismo para que exista un entendimiento común.

A continuación se exponen los **requisitos funcionales** definidos para el sistema modelado:

- ✓ **Cargar documento.** En caso que el usuario desee ver un documento de los que fue creado en la aplicación en otro momento selecciona esta opción y la aplicación le muestra los datos de la misma manera que fue introducida.
- ✓ **Crear documento:** Permite la creación de un documento con formato XML distinto para cada tipo de solicitud o documento complementario a partir de la información introducida por el usuario, el cual será almacenado a preferencia del mismo.

Existen dos tipos de documentos que la aplicación permitirá crear y cargar estos son, documentos complementarios y solicitudes. A continuación se listan los documentos legales que se digitalizaran:

Nº	Agrupación	Funcionalidad	Descripción	Complejidad
RF 1.	Guardar	Solicitud Facilidad.	Se crea un documento con formato XML y extensión .vuce, el cual contiene toda la información de la solicitud.	Media
RF 2.	Guardar	Solicitud Facilidad DM Global.	Se crea un documento con formato XML y extensión .vuce, el cual contiene toda la información de la Facilidad DM Global.	Media
RF 3.	Guardar	Solicitud Régimen Temporal.	Se crea un documento con formato XML y extensión .vuce, el cual contiene toda la información de la Solicitud Régimen Temporal.	Media
RF 4.	Guardar	Solicitud Liberación de Cotejo.	Se crea un documento con formato XML y extensión .vuce, el cual contiene toda la información de la Solicitud Liberación de Cotejo.	Media
RF 5.	Guardar	Solicitud Código Eventual.	Se crea un documento con formato XML y extensión .vuce, el cual contiene toda la información de la Solicitud Código Eventual.	Media
RF 6.	Guardar	Factura Comercial.	Se crea un documento con formato y extensión XML con los datos del documento	Media

CAPÍTULO 2 DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

			llenado.	
RF 7.	Guardar	Factura Provisional.	Se crea un documento con formato y extensión XML con los datos del documento llenado.	Media
RF 8.	Guardar	Factura Lista Empaque.	Se crea un documento con formato y extensión XML con los datos del documento llenado.	Media
RF 9.	Guardar	Lista Empaque.	Se crea un documento con formato y extensión XML con los datos del documento llenado.	Media
RF 10.	Guardar	Factura Venta.	Se crea un documento con formato y extensión XML con los datos del documento llenado.	Media
RF 11.	Guardar	Anulación de Documento.	Se crea un documento con formato y extensión XML con los datos del documento llenado.	Media
RF 12.	Guardar	Permiso y liberaciones.	Se crea un documento con formato y extensión XML con los datos del documento llenado.	Media
RF 13.	Guardar	Certificado Origen	Se crea un documento con formato y extensión XML con los datos del documento llenado.	Media
RF 14.	N/A	Cargar documento	El usuario selecciona el fichero que ya fue creado alguna vez por la aplicación y procede a cargar los datos de está mostrándose en los campos del formulario correspondientes.	Media

A continuación se describe el requisito funcional Guardar solicitud facilidad, donde se evidencia el flujo básico de guardar un documento (*Ver el resto de la descripción de requisitos en los anexos*):

Requisito Funcional Guardar Solicitud Facilidad.

Precondiciones

El usuario selecciona la opción “Solicitud Facilidad” contenida en el menú “Despacho

CAPÍTULO 2 DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

Comercial”.	
Flujo de eventos	
Flujo básico Solicitud Facilidad.	
1	El sistema muestra una pantalla que permite introducir los datos de la solicitud de facilidad.
2	Si el usuario selecciona la opción “Cargar” se ejecuta el <i>Requisito Funcional Extendido Cargar Documento</i>
3	El usuario selecciona la aduana de despacho.
4	El usuario selecciona la opción “Declarante” e introduce su código, nombre y apellido. Si selecciona la opción “Solicitante” se ejecuta el <u><i>Flujo Alternativo 4.a Datos del Solicitante</i></u>
5	El usuario introduce el nombre y apellido del declarante, el código de la entidad y nombre del Importador/Exportador, el número de manifiesto, el número de BL/GA y selecciona el lugar del embarque, el tipo de facilidad, el motivo y el tipo de operación.
6	El usuario selecciona la opción “Adicionar”. Si el usuario selecciona la opción “Modificar” se ejecuta el <u><i>Flujo Alternativo 6.a Modificar factura</i></u> Si el usuario selecciona la opción “Eliminar” se ejecuta el <u><i>6.b Alternativo Eliminar Factura</i></u>
7	El sistema muestra una pantalla que permita introducir el número de factura.
8	Introduce el número de factura y selecciona la opción “Aceptar”.
9	El usuario ejecuta el paso 6 <i>del Flujo Básico de Eventos Solicitud Facilidad</i> tantas veces como desee.
10	Introduce el término de validez solicitado y el plazo solicitado para el pago de facilidad.
11	El sistema realiza las validaciones correspondientes al negocio. Si el sistema detecta errores se ejecuta el <u><i>Flujo Alternativo 11.a Datos Incorrectos.</i></u>
12	Selecciona la opción “Guardar”. Si el usuario selecciona la opción “Cancelar” se ejecuta el <i>Requisito Funcional Extendido Cancelar.</i>
13	El sistema muestra una pantalla que permite guardar la información de la solicitud en un fichero XML con extensión vuce.
14	El usuario selecciona la ubicación y la opción “Guardar”.
15	El sistema salva la información de la solicitud en un fichero en la ubicación especificada.
16	Concluye el requisito.
Pos-condiciones	

CAPÍTULO 2 DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

1 Se guardan los datos en formato XML.

Flujos Alternativos

Flujo alternativo 4.a Datos del Solicitante.

1. El usuario introduce su carnet de identidad, nombre y apellidos.
2. Se ejecuta el paso 5 del *Flujo Básico de Eventos Solicitud Facilidad*.

Pos-condiciones

1. El sistema muestra los datos introducidos

Flujo alternativo 6.a Modificar Factura.

1. El usuario selecciona el número de factura y la opción “Modificar”.
2. El sistema muestra una pantalla con el número de factura seleccionado.
3. El usuario realiza la modificación y selecciona la opción “aceptar”.

Pos-condiciones

1. El sistema muestra los datos introducidos

Flujo alternativo 6.b Eliminar Factura.

1. El usuario selecciona el número de factura y la opción “Eliminar”.

Pos-condiciones

1. El sistema actualiza la tabla de facturas eliminando la factura seleccionada por el usuario.

Flujo alternativo 11.a Datos Incorrectos.

1. El sistema muestra un mensaje notificando que existen datos incorrectos
2. El usuario selecciona la opción aceptar y ejecuta el paso 3 del *Flujo Básico de Eventos Solicitud Facilidad*.

Pos-condiciones

1. El sistema le indica al usuario los campos incorrectos.

Conceptos	SolFacilidad	codigoEntidad
		numeroManifiesto
		codigoLugarEmbarque
		terminoValidez
		plazoPagoSolicitado
		otrosMotivos
		blga
		numeroBIGa

Requisitos especiales Requisito Funcional Extendido Cancelar.
Requisito Funcional Extendido Cargar Documento

Asuntos pendientes N/A.

CAPÍTULO 2 DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

En el caso de los documentos complementarios ocurre el mismo proceso que en las solicitudes. A continuación se muestra la descripción del requisito Guardar factura comercial perteneciente a los complementarios:

✚ Requisito funcional Guardar factura comercial.

Precondiciones	El usuario selecciona la opción “Factura Comercial” contenida en el submenú “Documento Complementario” del menú “Despacho Comercial”.
Flujo de eventos	
Flujo básico Factura Comercial	
1	El sistema muestra una pantalla que permite introducir los datos de la Factura Comercial.
2	Si el usuario selecciona la opción “Cargar” se ejecuta el <i>Requisito Funcional Extendido Cargar Documento</i> .
3	El usuario introduce el código del Proveedor.
4	El usuario selecciona la opción “Adicionar” teléfonos. Si el usuario selecciona la opción la opción “Modificar” se ejecuta el <i>Flujo Alternativo 4.a Modificar Teléfono</i> . Si el usuario selecciona la opción “Eliminar” se ejecuta el <i>Flujo Alternativo 4.b Eliminar Teléfono</i> .
5	El usuario introduce la relación de teléfono.
6	Si el número de teléfono insertado se encuentra repetido ver <i>Flujo Alternativo de Eventos 6.a Teléfono Repetido del Proveedor</i> .
7	El usuario ejecuta el paso 4 del <i>Flujo Básico de Eventos Factura Comercial</i> tantas veces como desee.
8	El usuario selecciona la opción “Adicionar” domicilio. Si el usuario selecciona la opción la opción “Modificar” se ejecuta el <i>Flujo Alternativo 8.a Modificar domicilio</i> . Si el usuario selecciona la opción “Eliminar” se ejecuta el <i>Flujo Alternativo 8.b Eliminar domicilio</i> .
9	El sistema muestra una pantalla para introducir la dirección.
10	El usuario introduce la el país y la dirección, selecciona la opción “Aceptar”
11	El usuario ejecuta el paso 8 del <i>Flujo Básico de Eventos Factura Comercial</i> tantas veces como desee.
12	El usuario selecciona la opción “Adicionar” correo electrónico. Si el usuario selecciona la opción la opción “Modificar” se ejecuta el <i>Flujo Alternativo 12.a Modificar correo electrónico</i>

CAPÍTULO 2 DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

	Si el usuario selecciona la opción “Eliminar” se ejecuta el <u><i>Flujo Alternativo 12.b Eliminar correo electrónico</i></u>
13	El sistema muestra una pantalla para introducir el correo electrónico.
14	El usuario introduce el correo electrónico, selecciona la opción “Aceptar”.
15	El usuario ejecuta el paso 12 del <i>Flujo Básico de Eventos Factura Comercial</i> tantas veces como desee.
16	El usuario selecciona la opción “Adicionar” URL. Si el usuario selecciona la opción la opción “Modificar” se ejecuta el <u><i>Flujo Alternativo 16.a Modificar URL</i></u> Si el usuario selecciona la opción “Eliminar” se ejecuta el <u><i>Flujo Alternativo 16.b Eliminar URL.</i></u>
17	El sistema muestra una pantalla para introducir la URL.
18	El usuario introduce la URL, selecciona la opción “Aceptar”.
19	El usuario ejecuta el paso 12 del <i>Flujo Básico de Eventos Factura Comercial</i> tantas veces como desee.
20	El usuario introduce el código del Importador.
21	El usuario selecciona la opción “Adicionar” teléfonos. Si el usuario selecciona la opción la opción “Modificar” se ejecuta el <u><i>Flujo Alternativo 21.a Modificar Teléfono.</i></u> Si el usuario selecciona la opción “Eliminar” se ejecuta el <u><i>Flujo Alternativo 21.b Eliminar Teléfono.</i></u>
22	El usuario introduce la relación de teléfono.
23	Si el número de teléfono insertado se encuentra repetido ver <u><i>Flujo Alternativo de Eventos 6.a Teléfono Repetido del Importador.</i></u>
24	El usuario ejecuta el paso 21 del <i>Flujo Básico de Eventos Factura Comercial</i> tantas veces como desee.
25	El usuario selecciona la opción “Adicionar” domicilio. Si el usuario selecciona la opción la opción “Modificar” se ejecuta el <u><i>Flujo Alternativo 25.a Modificar domicilio.</i></u> Si el usuario selecciona la opción “Eliminar” se ejecuta el <u><i>Flujo Alternativo 25.b Eliminar domicilio.</i></u>
26	El sistema muestra una pantalla para introducir la dirección.
27	El usuario introduce la el país y la dirección, selecciona la opción “Aceptar”
28	El usuario ejecuta el paso 25 del <i>Flujo Básico de Eventos Factura Comercial</i> tantas veces como desee.
29	El usuario selecciona la opción “Adicionar” correo electrónico. Si el usuario selecciona la opción la opción “Modificar” se ejecuta el <u><i>Flujo Alternativo 29.a Modificar correo electrónico</i></u> Si el usuario selecciona la opción “Eliminar” se ejecuta el <u><i>Flujo Alternativo 29.b Eliminar correo electrónico</i></u>

CAPÍTULO 2 DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

30	El sistema muestra una pantalla para introducir el correo electrónico.
31	El usuario introduce el correo electrónico, selecciona la opción “Aceptar”.
32	El usuario ejecuta el paso 29 <i>del Flujo Básico de Eventos Factura Comercial</i> tantas veces como desee.
33	El usuario selecciona la opción “Adicionar” URL. Si el usuario selecciona la opción la opción “Modificar” se ejecuta el <u><i>Flujo Alternativo 33.a Modificar URL</i></u> Si el usuario selecciona la opción “Eliminar” se ejecuta el <u><i>Flujo Alternativo 33.b Eliminar URL.</i></u>
34	El sistema muestra una pantalla para introducir la URL.
35	El usuario introduce la URL, selecciona la opción “Aceptar”.
36	El usuario ejecuta el paso 33 <i>del Flujo Básico de Eventos Factura Comercial</i> tantas veces como desee.
37	El usuario selecciona la opción “Adicionar” teléfonos. Si el usuario selecciona la opción la opción “Modificar” se ejecuta el <u><i>Flujo Alternativo 37.a Modificar Teléfono.</i></u> Si el usuario selecciona la opción “Eliminar” se ejecuta el <u><i>Flujo Alternativo 37 .b Eliminar Teléfono.</i></u>
38	Si el número de teléfono insertado se encuentra repetido ver <u><i>Flujo Alternativo de Eventos 37.a Teléfono Repetido del Importador.</i></u>
39	El usuario ejecuta el paso 37 <i>del Flujo Básico de Eventos Factura Comercial</i> tantas veces como desee.
40	El usuario introduce la fecha de emisión, la moneda, el número de factura, la condición de pago, la condición de entrega.
41	El usuario selecciona la opción “Adicionar” contrato compra venta. Si el usuario selecciona la opción la opción “Modificar” se ejecuta el <u><i>Flujo Alternativo 41.a Modificar contrato compra venta.</i></u> Si el usuario selecciona la opción “Eliminar” se ejecuta el <u><i>Flujo Alternativo 41.b Modificar contrato compra venta.</i></u>
42	El sistema muestra una ventana al usuario para introducir el número de contrato.
43	El usuario introduce el número de contrato y selecciona la opción aceptar.
44	El sistema comprueba que no esté repetido el contrato. En caso de estar repetido ver <u><i>Flujo Alternativo de Eventos 44.a Contrato Repetido.</i></u>
45	El sistema actualiza la tabla con el listado de contratos compra venta.
46	El usuario ejecuta el paso 41 tantas veces como desee.
47	El usuario selecciona un contrato y la opción “Adicionar” mercancía. Si el usuario selecciona la opción la opción “Modificar” se ejecuta el <u><i>Flujo Alternativo 47.a Modificar Mercancía.</i></u> Si el usuario selecciona la opción “Eliminar” se ejecuta el <u><i>Flujo Alternativo 47.b Modificar Mercancía.</i></u>

CAPÍTULO 2 DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

48	El sistema le muestra una ventana al usuario para que introduzca los datos de la mercancía.
49	El usuario introduce el código de referencia de la mercancía, la partida arancelaria, descripción comercial de la mercancía, unidad de medida, cantidad, precio unitario y el importe.
50	El sistema actualiza la tabla con el listado de mercancías y además el subtotal del importe y el subtotal del precio del listado de bultos.
51	El usuario ejecuta el paso 47 <i>Flujo Básico de Eventos Factura Comercial</i> tantas veces como desee.
52	El usuario selecciona la opción “Adicionar” subtotal. Si el usuario selecciona la opción la opción “Modificar” se ejecuta el <i>Flujo Alternativo 52.a Modificar subtotal</i> . Si el usuario selecciona la opción “Eliminar” se ejecuta el <i>Flujo Alternativo 52.b Modificar subtotal</i> .
53	El sistema muestra una ventana para seleccionar el código del subtotal.
54	El usuario introduce el código y selecciona la opción “Aceptar”. Si el subtotal seleccionado tiene otros subtotales subordinados ver <i>Flujo Alternativo de Alternativo 54.a Subtotales subordinados</i> .
55	El sistema actualiza la tabla con el listado de los subtotales.
56	Introducir la certificación de origen, las observaciones.
57	Selecciona la opción “Guardar”. Si el usuario selecciona la opción “Cancelar” se ejecuta el <i>Requisito Funcional Extendido Cancelar</i> .
58	El sistema realiza las validaciones correspondientes al negocio. Si el sistema detecta errores se ejecuta el <i>Flujo Alternativo 58.a Datos Incorrectos</i> .
59	El sistema muestra una pantalla que permite guardar la información de la solicitud en un fichero XML con extensión vuce.
60	El usuario selecciona la ubicación y la opción “Guardar”.
61	El sistema salva la información de la solicitud en un fichero en la ubicación especificada.
62	Concluye el requisito.

Pos-condiciones

2 Se guarda el documento en un fichero XML.

Flujo alternativo 4.b Eliminar Teléfono del Proveedor.

1. El sistema muestra un mensaje de notificación para saber si esta seguro el usuario de realizar esa acción.
2. El usuario selecciona la opción “Aceptar”.
3. El sistema actualiza la tabla con el listado de los teléfonos eliminando el teléfono seleccionado.
4. Se ejecuta el paso 7 del *Flujo Básico de Eventos Factura Comercial*.

Pos-condiciones

1. Se actualiza la tabla que contiene el listado de teléfonos.

Flujo alternativo 4.a Modificar Teléfono del Proveedor.

CAPÍTULO 2 DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

- 1 El usuario selecciona el teléfono.
- 2 El sistema muestra una ventana con los datos del teléfono seleccionado.
- 3 El usuario realiza las modificaciones y selecciona la opción “Aceptar”.
- 4 El sistema actualiza la tabla de teléfonos actualizando los cambios.
- 5 Se ejecuta el *paso 7 del Flujo Básico de Eventos Factura Comercial*.

Pos-condiciones

- 1 Se actualiza la tabla que contiene el listado de teléfonos.

Flujo alternativo 6.a Teléfono Repetido del Proveedor.

1. El sistema muestra un mensaje notificando que ya existe un teléfono insertado con el mismo número.
2. Seleccionar la opción “Aceptar” del mensaje mostrado.
3. Ejecutar el *paso 4 del Flujo Básico de Eventos Factura Comercial*.

Pos-condiciones

1. Se muestra un mensaje notificando el error detectado.

Flujo alternativo 8.a Modificar domicilio

1. El usuario selecciona el domicilio.
2. El sistema muestra una ventana el domicilio seleccionado.
3. El usuario realiza las modificaciones y selecciona la opción “Aceptar”.
4. El sistema actualiza la tabla de domicilio actualizando los cambios.
5. Se ejecuta el *paso 8 del Flujo Básico de Eventos Factura Comercial*.

Pos-condiciones

1. Se actualiza la tabla que contiene el listado de domicilio.

Flujo alternativo 8.b Eliminar domicilio.

1. El usuario selecciona el domicilio a eliminar y la opción “Eliminar”.
2. El sistema muestra un mensaje de notificación para saber si está seguro el usuario de realizar esa acción.
3. El usuario selecciona la opción “Aceptar”.
4. El sistema actualiza la tabla con el listado de los domicilio eliminando el correo seleccionado.
5. Se ejecuta el *paso 8 del Flujo Básico de Eventos Factura Comercial*.

Pos-condiciones

- 1 Se actualiza la tabla que contiene el listado de domicilio.

Flujo alternativo 12.a Modificar correo electrónico

- 1 El usuario selecciona el correo.
2. El sistema muestra una ventana el correo seleccionado.
3. El usuario realiza las modificaciones y selecciona la opción “Aceptar”.
4. El sistema actualiza la tabla de correos actualizando los cambios.
5. Se ejecuta el *paso 12 del Flujo Básico de Eventos Factura Comercial*.

CAPÍTULO 2 DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

Pos-condiciones

1. Se actualiza la tabla que contiene el listado de correos.

Flujo alternativo 12.b Eliminar correo electrónico.

1. El usuario selecciona el correo a eliminar y la opción "Eliminar".
2. El sistema muestra un mensaje de notificación para saber si está seguro el usuario de realizar esa acción.
3. El usuario selecciona la opción "Aceptar".
4. El sistema actualiza la tabla con el listado de los teléfonos eliminando el correo seleccionado.
5. Se ejecuta el *paso 12 del Flujo Básico de Eventos Factura Comercial*.

Pos-condiciones

1. Se actualiza la tabla que contiene el listado de correos.

Flujo alternativo 16.a Modificar URL.

1. El sistema muestra una ventana con la URL selecciona.
2. El usuario realiza las modificaciones que desea y selecciona la opción "Aceptar".
3. El sistema actualiza el listado de URL
4. Ejecutar el *paso 16 del Flujo Básico de Eventos Factura Comercial*.

Pos-condiciones

1. Se actualiza la tabla que contiene el listado de teléfonos.

Flujo alternativo 16.b Eliminar URL.

1. El usuario selecciona la URL y la opción "Eliminar".
2. El sistema muestra un mensaje de notificación para saber si está seguro el usuario de realizar esa acción.
3. El usuario selecciona la opción "Aceptar".
4. El sistema actualiza la tabla de URL.
5. Ejecutar el *paso 16 del Flujo Básico de Eventos Factura Comercial*

Pos-condiciones

1. Se actualiza la tabla que contiene el listado de URL.

Flujo alternativo 21.a Eliminar Teléfono del Importador.

1. El sistema muestra un mensaje de notificación para saber si está seguro el usuario de realizar esa acción.
2. El usuario selecciona la opción "Aceptar".
3. El sistema actualiza la tabla con el listado de los teléfonos eliminando el teléfono seleccionado.
4. Se ejecuta el *paso 21 del Flujo Básico de Eventos Factura Comercial*.

Pos-condiciones

1. Se actualiza la tabla que contiene el listado de teléfonos.

Flujo alternativo 21.a Modificar Teléfono del Importador.

1. El usuario selecciona el teléfono.

CAPÍTULO 2 DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

2. El sistema muestra una ventana con los datos del teléfono seleccionado.
3. El usuario realiza las modificaciones y selecciona la opción “Aceptar”.
4. El sistema actualiza la tabla de teléfonos actualizando los cambios.
5. Se ejecuta el *paso 21 del Flujo Básico de Eventos Factura Comercial*.

Pos-condiciones

1. Se actualiza la tabla que contiene el listado de teléfonos.

Flujo alternativo 37 Teléfono Repetido del Importador.

1. El sistema muestra un mensaje notificando que ya existe un teléfono insertado con el mismo número.
2. Seleccionar la opción “Aceptar” del mensaje mostrado.
3. Ejecutar el *paso 37 del Flujo Básico de Eventos Factura Comercial*.

Pos-condiciones

1. Se muestra un mensaje notificando el error detectado.

Flujo alternativo 25.a Modificar domicilio

1. El usuario selecciona el domicilio.
2. El sistema muestra una ventana el domicilio seleccionado.
3. El usuario realiza las modificaciones y selecciona la opción “Aceptar”.
4. El sistema actualiza la tabla de domicilio actualizando los cambios.
5. Se ejecuta el *paso 25 del Flujo Básico de Eventos Factura Comercial*.

Pos-condiciones

1. Se actualiza la tabla que contiene el listado de domicilio.

Flujo alternativo 25.b Eliminar domicilio.

1. El usuario selecciona el domicilio a eliminar y la opción “Eliminar”.
2. El sistema muestra un mensaje de notificación para saber si está seguro el usuario de realizar esa acción.
3. El usuario selecciona la opción “Aceptar”.

Pos-condiciones

1. Se actualiza la tabla que contiene los domicilios.

Flujo alternativo 41.a Eliminar Contrato.

1. El usuario selecciona el domicilio y la opción “Eliminar”.
2. El sistema muestra un mensaje de notificación para saber si está seguro el usuario de realizar esa acción.
3. El usuario selecciona la opción “Aceptar”.

Pos-condiciones

1. Se actualiza la tabla que contiene los domicilios.

Flujo alternativo 41.b Modificar Contrato.

1. El usuario selecciona el contrato y la opción “Modificar”.

CAPÍTULO 2 DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

2. El sistema muestra una ventana el contrato seleccionado.
3. El usuario realiza las modificaciones y selecciona la opción "Aceptar".
4. El sistema actualiza la tabla de contrato actualizando los cambios.
5. Se ejecuta el *paso 41 del Flujo Básico de Eventos Factura Comercial*.

Pos-condiciones

1. Se actualiza la tabla que contiene el listado de contrato.

Flujo alternativo 47.a Modificar Mercancía.

1. El usuario selecciona la mercancía y la opción "Modificar".
2. El sistema muestra una ventana la mercancía seleccionada.
3. El usuario realiza las modificaciones y selecciona la opción "Aceptar".
4. Se actualiza la tabla que contiene el listado de mercancía.
5. Se ejecuta el *paso 47 del Flujo Básico de Eventos Factura Comercial*.

Pos-condiciones

1. Se actualiza la tabla que contiene el listado de mercancía.

Flujo alternativo 47.b Eliminar Mercancía

1. El usuario selecciona el mercancía y la opción "Eliminar".
2. El sistema muestra un mensaje de notificación para saber si está seguro el usuario de realizar esa acción.
3. El usuario selecciona la opción "Aceptar".

1. Pos-condiciones

2. Se actualiza la tabla que contiene las mercancías.

Flujo alternativo 52. a Eliminar Subtotal

1. El usuario selecciona el subtotal y la opción "Eliminar".
2. El sistema muestra un mensaje de notificación para saber si está seguro el usuario de realizar esa acción.
3. El usuario selecciona la opción "Aceptar".

Pos-condiciones

1. Se actualiza la tabla que contiene los subtotales y los subtotales subordinados.

Flujo alternativo 52. b Modificar Subtotal

1. El usuario selecciona el subtotal y la opción "Modificar".
2. El sistema muestra una ventana el subtotal seleccionado.
3. El usuario realiza las modificaciones y selecciona la opción "Aceptar".

Pos-condiciones

1. Se actualiza la tabla que contiene los subtotales.

Flujo alternativo 54.a Subtotales subordinados

1. El sistema muestra una ventana con los subordinados del subtotal seleccionado y el valor.
2. El usuario selecciona el subtotal subordinado e introduce el valor. Selecciona la opción "Aceptar".

CAPÍTULO 2 DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

3. El sistema actualiza la tabla con el listado de los subordinados y el valor de subtotal correspondiente al subordinado agregado.
4. Introducir el total del valor monetario de los subordinados
5. Si el usuario selecciona la opción “Eliminar” de la tabla subordinado ver Flujo Alternativo 54.a.5.a.
Si el usuario selecciona la opción “Modificar” de la tabla subordinado ver Flujo Alternativo 54.a.5.b.
6. Ejecutar el *paso 54 del Flujo Básico de Eventos Factura Comercial.*

Pos-condiciones

1. Se actualiza la tabla que contiene el listado de los subordinados del subtotal.

Flujo alternativo 54.a.5.a.Eliminar Subtotal subordinado

1. El usuario selecciona el subtotal y la opción “Eliminar”.
2. El sistema muestra un mensaje de notificación para saber si está seguro el usuario de realizar esa acción.
3. El usuario selecciona la opción “Aceptar”.

Pos-condiciones

1. Se actualiza la tabla que contiene los subtotales y los subtotales subordinados.

Flujo alternativo 54.a.5.b.Modificar Subtotal

1. El usuario selecciona el subtotal y la opción “Modificar”.
2. El sistema muestra una ventana el subtotal seleccionado.
3. El usuario realiza las modificaciones y selecciona la opción “Aceptar”.

Pos-condiciones

1. Se actualiza la tabla que contiene los subtotales.

Flujo alternativo 43.a Datos Incorrectos

1. El sistema muestra un mensaje notificando el/los error(es) detectado(s).
2. Se selecciona la opción “Aceptar” del mensaje mostrado.
3. Se ejecuta el *paso 2 del Flujo Básico de Eventos Factura Comercial.*

Pos-condiciones

1. Se muestra un mensaje notificando el/los error(es) detectado(s).

Conceptos

**DcFacturaComercialProvi
cional**

codigoCondicionPago
codigocondicionEntrega
codigoMoneda
totalSubtotal
contrato

CAPÍTULO 2 DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

Requisitos especiales	Requisito Funcional Extendido Cancelar. Requisito Funcional Extendido Cargar Documento
Asuntos pendientes	N/A

Para ver el resto de los requisitos diríjase a los Anexos de la presente investigación.

2.3.3 Técnicas para la Validación de requisitos.

La validación de los requisitos es el proceso para comprobar que la especificación de los requisitos de software se ajusta a las necesidades de clientes/usuarios y otros implicados, se examinan las especificaciones para asegurar que todos los requisitos del sistema han sido establecidos sin ambigüedad, sin inconsistencias y sin omisiones, que los errores detectados hayan sido corregidos, y que el resultado del trabajo se ajusta a los estándares establecidos para el producto.

Para desarrollar la validación de requisitos se proponen varias técnicas, dentro de ellas las aplicadas son las siguientes:

Revisiones: esta técnica consiste en la lectura y corrección de la documentación o modelado de la definición de requisitos. En el caso de la solución propuesta se le realizaron diversas revisiones al documento de requisitos para lograr una correcta interpretación de la información transmitida, los señalamientos planteados fueron recogidos y corregidos posteriormente.

Prototipos: algunas propuestas, sobre esta técnica señalan que se basa en obtener de la definición de requisitos, prototipos, que a pesar de no contener la totalidad de las funcionalidades del sistema, permitan al usuario hacerse una idea de la estructura de la interfaz de usuario del sistema (*Ver prototipos de interfaz en los anexos*).

2.3.4 Requisitos no funcionales.

Los requerimientos no funcionales sirven para que clientes y usuarios puedan valorar las características no funcionales del producto, pues si se conoce que el mismo cumple con toda la funcionalidad requerida, las propiedades no funcionales, como cuán usable, seguro, conveniente y agradable, pueden marcar la diferencia entre un producto bien aceptado y uno con poca aceptación.

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. (29) Son las características que hacen al producto atractivo, usable, rápido y confiable. Por lo tanto la aplicación debe cumplir con las siguientes propiedades:

Hardware: Las computadoras que utilizarán el software a desarrollar deberán tener la velocidad de la CPU a 1,6 GHz como mínimo, se recomienda uno superior. También deben poseer un procesador Core Duo, Pentium 4 o uno superior. La memoria RAM que deben tener como mínimo es de 1 GB, se recomienda 2 GB o superior. La resolución de la pantalla debe ser de 1024 x 768 o una superior.

Diseño: El sistema debe tener una apariencia profesional y un diseño gráfico sencillo, debe ser de fácil de utilizar para que usuarios inexpertos puedan familiarizarse rápidamente.

Rendimiento: El tiempo de respuesta está dado por la cantidad de información a procesar, entre mayor cantidad de información mayor será el tiempo de procesamiento. Al igual que el tiempo de respuesta, la velocidad de procesamiento de la información, la actualización y la recuperación dependerán de la cantidad de información que tenga que procesar la aplicación.

Software: Las computadoras deben tener instalada la máquina virtual de java en su versión 1.7 o superior o una versión posterior a esta. El sistema que se desarrollará será multiplataforma bajo el lenguaje java.

2.4 Patrones de diseño utilizados.

Un patrón de diseño puede considerarse como el conjunto de reglas que describen cómo afrontar tareas y solucionar problemas que surgen durante el desarrollo de software. Otros autores definen un patrón de diseño como la idea que ha sido útil en la práctica y determinado contexto y que probablemente sea útil a otros. Las clases que participan con sus

CAPÍTULO 2 DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

responsabilidades y colaboraciones entre ellas; en ocasiones muestran fragmentos de códigos que especifican la correcta implementación del patrón.

- Solucionan un problema: los patrones capturan soluciones, no solo principios o estrategias abstractas.
- Son un concepto probado: capturan soluciones demostradas, no teorías o especulaciones.
- La solución no es obvia: los mejores patrones generan una solución a un problema de forma indirecta.
- Describen participantes y relaciones entre ellos: describen módulos, estructuras del sistema y mecanismos complejos. (21)

Los patrones de diseño no son más que soluciones simples y elegantes a problemas específicos y comunes del diseño orientado a objetos. Son soluciones basadas en la experiencia y que se ha demostrado que funcionan, son descripciones de clases cuyas instancias colaboran entre sí y brindan una solución ya probada y documentada a problemas de desarrollo de software que están sujetos a contextos similares.

2.4.1 Patrones GRASP

GRASP es un acrónimo que significa patrones generales de software para asignar responsabilidades (**General Responsibility Assignment Software Patterns**) y representa la descripción de los principios fundamentales de la asignación de responsabilidades expresados como patrones.(22)

El nombre se eligió para indicar la importancia de captar (**grasping**) estos principios, si se quiere diseñar lo más eficazmente el software orientado a objetos. Los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones. Tienen como motivación:

- La asignación de responsabilidades como la habilidad más importante en el análisis y diseño orientado por objetos.
- Respetar los principios fundamentales como uno de los factores críticos, para obtener diseños reutilizables, mantenibles y extendibles.

Los patrones **GRASP** utilizados en el diseño de la aplicación fueron los siguientes:

- **Creador:** en la clase de nombre Principal del proyecto se localizan las acciones definidas para la aplicación, en las cuales se crean los objetos de las clases que representan

CAPÍTULO 2 DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

las interfaces, confirmando que la clase es “creador” de dichas interfaces. Ej.: crea los objetos de todas las solicitudes y los documentos complementarios.

➤ **Controlador:** la aplicación implementa una clase controladora llamada Documento para atender todas las peticiones en el proceso de creación de documentos XML, esta es la encargada de invocar al método que creara el XML correspondiente al documento que se solicite. Es el único punto de entrada de toda la aplicación en un determinado entorno. Cuando recibe una petición, utiliza el sistema de enrutamiento para enviar la acción al controlador responsable de la solución.

➤ **Alta Cohesión:** un ejemplo de ello son las clases correspondientes a cada tipo de documento donde cada una maneja la información propia de ella, siendo responsables de definir las acciones para la creación de XML y colaborar con otras para realizar diferentes operaciones, instanciar objetos y acceder a las propiedades.

➤ **Polimorfismo:** en este caso específico existen distintos tipos de solicitudes pero en dependencia del tipo que sea es que se crearan los ficheros para darle solución a la problemática.

➤ 2.4.2 Patrones GoF.

El grupo de GoF (**Gang of Four**) clasificaron los patrones en 3 grandes categorías basadas en su propósito: creacionales, estructurales y de comportamiento.

- *Creacionales:* los patrones creacionales se relacionan con las formas de crear instancias de objetos. El objetivo de estos patrones es abstraer el proceso de instanciación y ocultar los detalles de cómo son los objetos creados o inicializados.
- *Estructurales:* los patrones estructurales describen como las clases y objetos pueden ser combinados para formar grandes estructuras y proporcionar nuevas funcionalidades.
- *Comportamiento:* los patrones de comportamiento ayudan a definir la comunicación e iteración entre los objetos de un sistema. El propósito de este patrón es reducir el acoplamiento entre los objetos.

Los patrones **GOF** utilizados en el diseño de la aplicación fueron los siguientes:

Cadena de responsabilidad (Chain of Responsibility): Es un patrón del tipo comportamiento el cual especifica cómo repartir las responsabilidades respecto a una petición u objeto, de forma tal que al recibir un objeto éste pasa a través de un conjunto de “manejadores” que tienen la responsabilidad de atender la solicitud. El primer manejador recibe

CAPÍTULO 2 DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

la petición y puede atenderla o bien reenviarla al próximo candidato el cual puede hacer lo mismo. El uso de este patrón en la investigación permitió la creación de asistentes visuales para guiar al usuario paso a paso, donde cada paso representa un elemento o manejador de la cadena y la petición constituye el objetivo que se desea alcanzar.

2.5 Arquitectura de software

Existen diversas definiciones formales referentes al término arquitectura de software como por ejemplo.

- “La arquitectura del software de un programa o sistema de computación es la estructura o estructuras del sistema que comprende los elementos del software, las propiedades externamente visibles de esos elementos, y las relaciones entre ellos”.
- “La arquitectura de software es, a grandes rasgos, una vista del sistema que incluye los componentes principales del mismo, la conducta de esos componentes según se la percibe desde el resto del sistema y las formas en que los componentes interactúan y se coordinan para alcanzar la misión del sistema. La vista arquitectónica es una vista abstracta, aportando el más alto nivel de comprensión y la supresión o diferimiento del detalle inherente a la mayor parte de las abstracciones”.

Esta definición es un poco amplia, por lo que una de las más aplicadas y la que se toma como referencia en este trabajo es la establecida por la IEEE STD 1471-2000 (**Software Engineering Standards Committee of the IEEE Computer Society, 2000**): “La Arquitectura de Software es la organización fundamental de un sistema encarnada en sus componentes, las relaciones entre ellos y el ambiente y los principios que orientan su diseño y evolución.”

Siguiendo este concepto de arquitectura existen un grupo de **estilos arquitectónicos** separando cada tipo de aplicación dentro de uno de estos estilos. Alguno de ellos es:

- La arquitectura en Capas.
- La arquitectura Orientadas a Objetos.
- La arquitectura Basadas en Componentes.

En la aplicación se encuentra dentro de la arquitectura en capas. Para un mayor entendimiento del sistema separamos la interfaz visual de toda lo lógica de negocio dejando al usuario final de la aplicación en interacción solamente con la capa de las vistas; estas capas serán las responsables de interactuar con la lógica de negocio como se muestra en la **figura 1.3**



Figura 1.3 Arquitectura de la aplicación.

2.6 Diseño del sistema.

En el desarrollo o ciclo de vida de un sistema informático, el diseño del sistema constituye un elemento fundamental del mismo. Se centra en proporcionar la funcionalidad del sistema a través de sus diferentes componentes. En esta disciplina se diseña una solución que convierte los requisitos del cliente en un sistema de información real.

2.6.1 Diagrama de clase de la aplicación.

La **figura 1.4** muestra el diagrama de diseño de las clases del sistema, sirve como una aproximación al diseño final de la aplicación en cuanto a la organización de sus clases. Se cuenta con una clase controladora llamada *Documento* la cual está encargada de ejecutar las principales funcionalidades del sistema; repartiendo las tareas en dependencia del tipo de documento que se quiera procesar. Se especializan en tipos de documentos las clases *Solicitud* que consta a su vez con otras clases que heredan de ella y la clase *DocumentoComplementario* de la cual heredan también otras clases como hijas.

CAPÍTULO 2 DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

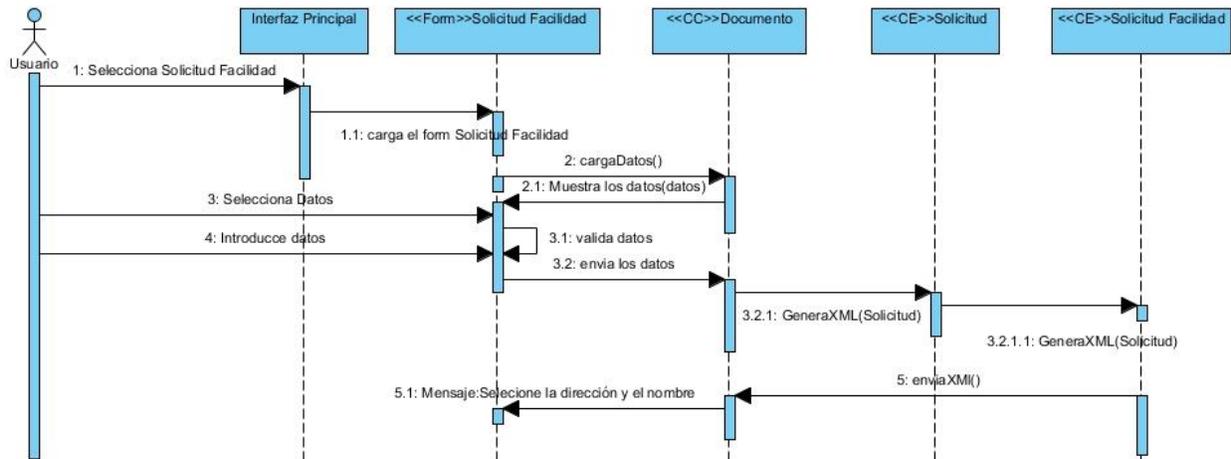


Figura 1.5 Diagrama de secuencia del RF Guardar solicitud facilidad.

En este diagrama se describe el proceso para crear y guardar una solicitud de tipo facilidad; luego de introducir los datos se guarda la información en un documento con un formato XML y extensión .vu en la ubicación y con el nombre que el usuario especifique.

De igual manera sería el flujo para el caso de un tipo de documento complementario como sería el caso del requisito descrito con anterioridad Guardar factura comercial. Donde el usuario introduce de igual manera los datos al sistema, este los valida y le da la opción al mismo de que guarde la información en un archivo. (ver diagrama de secuencia Figura 1.6.)

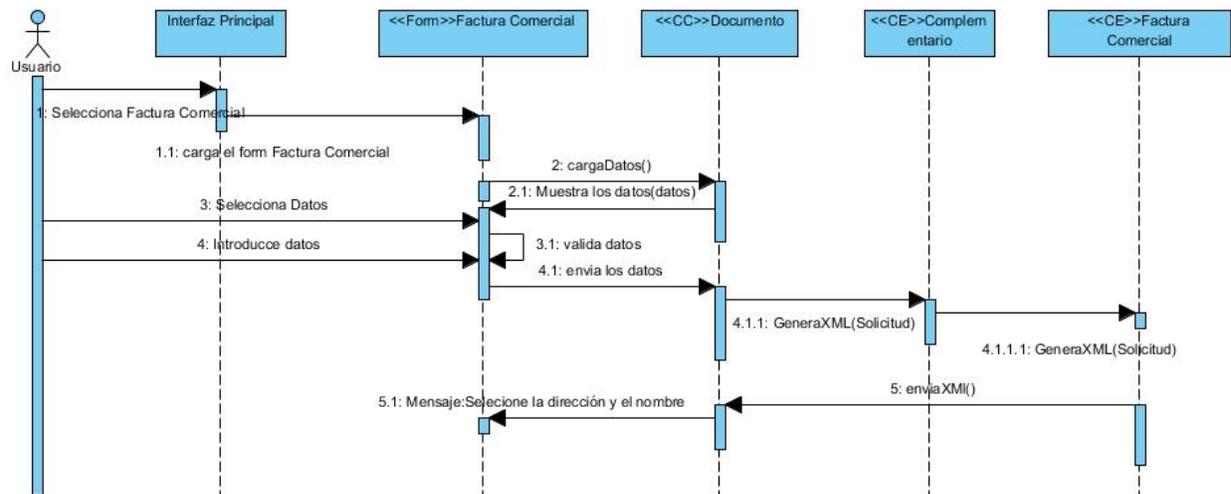


Figura 1.6 Diagrama de secuencia del RF Guardar solicitud factura comercial.

2.6.3 Diagrama de paquetes.

El lenguaje UML ofrece el mecanismo paquete que permite describir los grupos de elementos subsistemas. Un paquete es un conjunto de cualquier tipo de elementos de un modelo: clases, casos de uso, diagramas de colaboración u otros paquetes (los anidados); es un mecanismo utilizado para agrupar elementos de UML que permite organizar los elementos modelados con UML, facilitando de ésta forma el manejo de los modelos de un sistema complejo. (23)

Los diagramas de paquetes se usan para reflejar la organización de los paquetes y sus elementos, y para proveer una visualización de sus componentes.

Entre sus características se tiene:

- Permiten dividir un modelo para agrupar y encapsular sus elementos en unidades lógicas individuales.
- En general, pueden tener una interfaz (métodos de clases e interfaces exportadas) y una realización de éstas interfaces (clases internas que implementan dichas interfaces) Se pueden utilizar para plantear la arquitectura del sistema a nivel macro.

A continuación se muestra la **figura 1.7**, mostrando el diagrama de paquetes de la aplicación donde se encuentran distribuidas las clases, librerías y Swich utilizados en la aplicación y la relación que existe entre ellos. El paquete clases del negocio importa las librerías XML para lograr la generación de los documentos, a su vez este paquete es usado por el de Formularios para crear las instancias de clases y poder manejar los objetos, así como también importa las librerías Swich con el objetivo de darle una vista agradable a las interfaces.

CAPÍTULO 2 DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

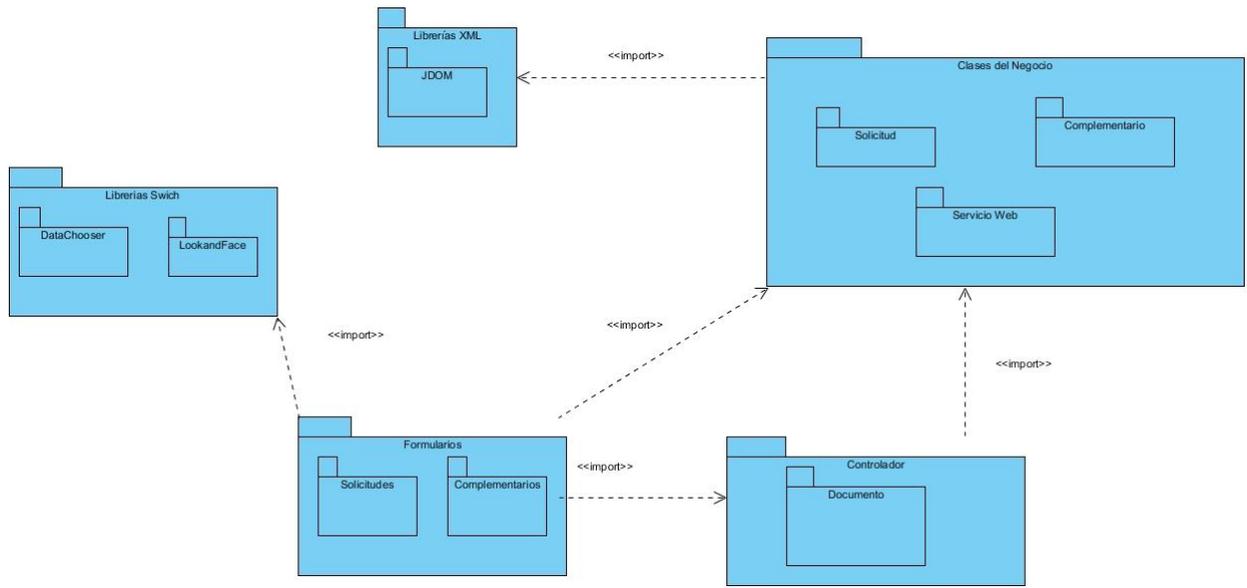


Figura 1.7 Diagrama paquetes de la aplicación.

2.7 Métricas para la evaluación del diseño.

Aunque los términos medida, medición y métricas se utilizan a menudo indistintamente, existe gran confusión a la hora de referirse a ellos. Dentro del contexto de la ingeniería del software, una medida proporciona una indicación cuantitativa de extensión, cantidad, dimensiones, capacidad y tamaño de algunos atributos de un proceso o producto. La medición es el proceso por el cual los números o símbolos son asignados a atributos o entidades en el mundo real tal como son descritos de acuerdo a reglas claramente definidas, también es definida como el acto de determinar una medida. (24)

2.7.1 Métricas propuestas por Lorenz y Kidd.

Lorenz y Kidd dividen las métricas basadas en clases en cuatro categorías: tamaño, herencia, valores internos y valores externos. Las métricas orientadas a tamaños para una clase se centran en cálculos de atributos y de operaciones para una clase individual, para luego promediar los valores para el sistema en su totalidad. Las métricas basadas en herencia se centran en la forma en que se reutilizan las operaciones en la jerarquía de clases. Las métricas para valores internos de clase examinan la cohesión y asuntos relacionados con el código así como las métricas orientadas a valores externos examinan el acoplamiento y la reutilización.(25)

Del conjunto de métricas planteadas por Lorenz y Kidd se aplicó al diseño propuesto:

CAPÍTULO 2 DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

- Tamaño operacional de clase (TOC)

2.7.2 Métrica de Tamaño Operacional de Clase (TOC)

Luego de analizar todas las métricas mencionadas anteriormente, se decidió utilizar para validar el diseño de la solución la de Tamaño Operacional de Clase (TOC). La misma se encarga de medir la calidad de acuerdo a los atributos responsabilidad, complejidad de implementación y reutilización de las clases.

La aplicación de la métrica TOC define los siguientes atributos de calidad:

- Responsabilidad: Consiste en la responsabilidad asignada a una clase en un marco de modelado de un dominio o concepto, de la problemática propuesta.
- Complejidad de implementación: Consiste en el grado de dificultad que tiene implementar un diseño de clases determinado.
- Reutilización: Consiste en el grado de reutilización presente en una clase o estructura de clase, dentro de un diseño de software.

Para evaluar las métricas son necesarios los valores de los umbrales para los parámetros de calidad. Algunos especialistas plantean umbrales para esta métrica basándose en el promedio de operaciones por clases obtenidos, estos valores fueron los aplicados en el diseño del sistema y los mismos son reflejados en la siguiente tabla.

Atributos de calidad	Categoría	Criterio
Responsabilidad	Baja	\leq Promedio
	Media	$>$ Promedio y $\leq 2*$ Promedio
	Alta	$>2*$ Promedio
Complejidad de implementación	Baja	\leq Promedio
	Media	$>$ Promedio y $\leq 2*$ Promedio
	Alta	$>2*$ Promedio
Reutilización	Baja	$>2*$ Promedio
	Media	$>$ Promedio y $\leq 2*$ Promedio
	Alta	\leq Promedio

A continuación se muestra la **figura 1.8** donde se representa el nivel de responsabilidad, complejidad de implementación y reutilización del diseño realizado. Se muestra

CAPÍTULO 2 DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

aproximadamente que en cuanto a responsabilidad el 90% aproximadamente de las clases presentan una responsabilidad baja al igual que la complejidad y con un 90 % aproximado igual de reutilización en los demás casos presentan un aproximado del 4,5%.



Figura 1.8 Evaluación del diseño.

2.8 Aportes de la solución y beneficios esperados

La solución propuesta garantiza el cumplimiento de todas las funcionalidades necesarias para una correcta ejecución del flujo de trabajo en la generación de documentos. Proveerá los diferentes tipos de ficheros XML necesarios que permitirán la realización de los trámites con la ventanilla. Con la presente solución el usuario ya podrá realizar los trámites sin necesidad de interactuar directamente con la VUCE y utilizar la vía de envío de documentos legales por medio de ficheros, funcionalidad que brinda la ventanilla y que se le ve afectada al usuario ya que no necesariamente estará capacitado para crear estos ficheros.

Conclusiones Parciales

- Se expuso de manera clara los requisitos funcionales que debe cumplir el software para que el producto cumpla con las expectativas del cliente, así como su validación mediante técnicas.
- Se mostró los principales artefactos generados en el diseño de la aplicación, evidenciándose los principales conceptos en el modelo conceptual donde se presentan las entidades importantes que se encuentran en el ámbito del problema y sus relaciones.

CAPÍTULO 2 DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

- Se presentaron las técnicas de validación de diseño utilizada en la solución planteada.

Capítulo III: Implementación y pruebas al sistema.

3.1 Introducción

En este capítulo se aborda los aspectos más importantes relacionados con la implementación del sistema y la validación del mismo, se realiza el diagrama de despliegue, se enuncian los estándares de codificación que se siguieron para generar el código. También se valida mediante pruebas, si el software realizado responde a las necesidades del cliente y se corresponde con los requisitos funcionales planteados.

3.2 Estándares de codificación para java

Los estándares de codificación son de un alto valor para los programadores por múltiples razones; por lo general ningún software es mantenido durante toda su vida por su autor original por lo que estos estándares de código mejoran la lectura del software, permitiendo a los programadores entender el nuevo código más rápidamente y mejor.

Fichero fuente Java (.java)

Cada fichero Java debe contener una única clase o interfaz pública en el caso del fichero tiene que coincidir el nombre del fichero con el nombre de la clase

En todo fichero fuente Java distinguimos las siguientes secciones:

- Comentarios de inicio.
- Sentencia de paquete.
- Sentencias de importación.
- Declaraciones de clases e interfaces.

Comentarios de implementación

Comentario a final de línea

Comentario situado al final de una sentencia de código y en la misma línea.

- `int contador = 4 + 10; // Inicialización del contador`

➤ contador++; /* Incrementamos el contador */

Importación de clases

Tras la declaración del paquete se incluirán las sentencias de importación de los paquetes necesarios. Se recomienda minimizar en la medida de lo posible el uso de importaciones del tipo *package*, pues dificultan la comprensión de las dependencias existentes entre las clases utilizadas por la aplicación. A continuación se muestra un ejemplo en la **Figura 1.9**

```
6 package GUI_Solicitudes;
7
8 import ClasesSolicitudes.Mercancia;
9 import ClasesSolicitudes.SolRegimenTemporal;
10 import ClasesSolicitudes.Telefono;
11 import ClasesSolicitudes.Ubicacion;
12 import CC_Controlador.Documento;
13 import java.awt.Toolkit;
14 import java.awt.event.KeyEvent;
15 import java.io.File;
16 import java.io.IOException;
17 import java.util.ArrayList;
18 import java.util.LinkedList;
19 import java.util.List;
20 import javax.swing.*;
21 import javax.swing.table.DefaultTableModel;
22 import javax.swing.table.TableColumn;
23 import javax.swing.table.TableModel;
24 import javax.xml.parsers.DocumentBuilder;
25 import javax.xml.parsers.DocumentBuilderFactory;
26 import javax.xml.parsers.ParserConfigurationException;
```

Figura 1.9 Importación de clases y paquetes en java.

Métodos

Se realizó un estándar a la hora de nombrar los métodos donde el primer nombre sería minúscula y el resto de cada palabra sería mayúscula. En la clase Documento figura 2.0 se puede observar un ejemplo práctico de éste estándar.

```
+ public void guardarDocumento(Solicitud solicitud) {...}
+ public void enviarJson() {...}
+ public void verificarConexion() {...}
```

CAPÍTULO 3 IMPLEMENTACIÓN Y PRUEBA

Figura 2.0 Estandarización de los nombres de los métodos.

A continuación se muestra la **figura 2.1** de una pequeña parte clave en el proceso de creación de los XML. Con el uso de librerías como JDOM la cual consta con una clase llamada Document la cual hace posible la creación de los nodos con el fin de crear estructuras de este tipo:

```
public File generarXML(Solicitud solicitud,File xml) {
    SolRegimenTemporal regimenTemporal=(SolRegimenTemporal)solicitud;
    try {
        DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
        DocumentBuilder builder = factory.newDocumentBuilder();
        DOMImplementation implementation = builder.getDOMImplementation();
        Document document = implementation.createDocument(null, "SolicitudRegimenTemporal", null);
```

Figura 2.1 Generar XML con el uso de la librería JDOM.

En la siguiente **figura 2.2** se observa un segmento de código que muestra como con la ayuda de este **document** creado de la clase Document quedaría conformado un nodo de nuestro XML con el la información de texto según lo maneje la aplicación

```
Element elemento1 = document.createElement("aduanaDespacho");
Text txt1 = document.createTextNode(solicitud.getCodigoAduana());
elemento1.appendChild(txt1);
```

Figura 2.2 Nodo del XML.

Así quedare el nodo en el XML **Figura 2.3** *Nodos de datos*.

```
<SolicitudRegimenTemporal>
  <aduanaDespacho>Aduana General de la República de Cuba</aduanaDespacho>
```

Figura 2.3 Nodo de datos.

Paquetes de clases

Las clases se encuentran agrupadas en paquetes los cuales llevan nombres que corresponden con las respectivas clases que contienen. **Figura 2.4**

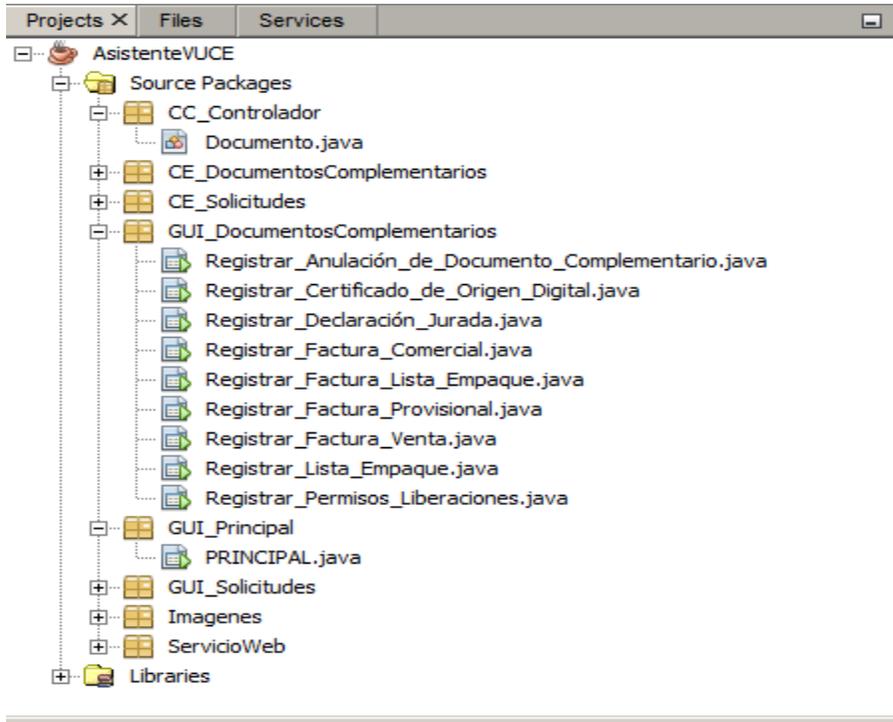


Figura 2.4 Paquetes y Clases.

3.3 Tratamiento de Errores

El tratamiento de los errores en un programa Java puede hacerse mediante las sentencias estructuradas (*if*, *while* y *return*). En la solución propuesta se realiza mediante una técnica de programación denominada gestión de excepciones. Con las excepciones se podrá evitar repetir continuamente código, en busca de un posible error, y avisar a otros objetos de una condición fuera de lo normal de la ejecución del código durante la aplicación.

Para que el sistema de gestión de excepciones funcione, se ha de trabajar en dos partes de los programas:

- Definir qué partes de los programas crean una excepción y bajo qué condiciones. Para ello se utilizan las palabras reservadas *throw* y *throws*.
- Comprobar en ciertas partes de los programas si una excepción se ha producido, y actuar en consecuencia. Para ello se utilizan las palabras reservadas *try*, *catch* y *finally*.

Manejo de excepciones: try - catch

Cuando el programador va a ejecutar una parte del código que pueda provocar una excepción (por ejemplo, una lectura en un fichero XML), debe incluir este fragmento de código dentro de un bloque *try* (Figura 2.5 Excepción *try*):

```
try {  
    DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
```

Figura 2.5 Excepción *try*.

A partir de la posible excepción que se pueda generar en esta parte será el tratamiento que se le dé. Para ello se utilizan las cláusulas *catch*, en las que se especifica que acción realizar: (Figura 2.6 Excepción *catch*):

```
catch (TransformerException ex) {  
    JOptionPane.showMessageDialog(null, "Imposible crear el fichero");
```

Figura 2.6 Excepción *catch*.):

En este caso se le ha mostrado un mensaje al usuario informándole que ha sido imposible crear el fichero deseado.

3.4 Diagrama de despliegue

El diagrama de Despliegue describe la distribución física del sistema en términos de cómo las funcionalidades se distribuyen entre los nodos de computación sobre los que se va a instalar el sistema; mostrando las relaciones entre el hardware y el software en el sistema final. Se representa como un grafo de nodos unidos por conexiones de comunicación.

En la siguiente **figura 2.7** se ilustra el modelo de despliegue para el sistema planteado.

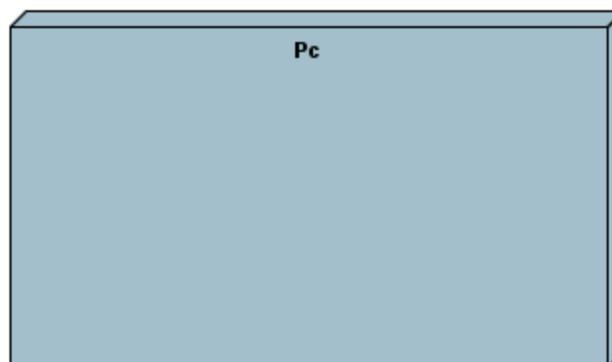


Figura 2.7 Modelo de despliegue.

Para el despliegue de la solución se cuenta únicamente con un nodo de procesamiento el cual tendrá los componentes físicos del sistema. Las especificaciones en cuanto a hardware y software del nodo Pc se describen en los requisitos no funcionales.

3.5 Validación de la solución

Técnicas de evaluación dinámicas o pruebas de software.

Estas técnicas generan entradas al sistema con el objetivo de detectar fallos cuando el sistema ejecuta dichas entradas. Los fallos se observan cuando se detectan incongruencias entre la salida esperada y la salida real. La aplicación de técnicas dinámicas es también conocida como pruebas de software o testing y se aplican generalmente sobre el código, único producto ejecutable del desarrollo.

3.5.1 Técnicos de prueba

Las técnicas de evaluación dinámica o prueba proporcionan distintos criterios para generar casos de prueba que provoquen fallos en los programas. Estas técnicas se agrupan en:

Técnica de caja blanca o estructural: se basan en un minucioso examen de los detalles procedimentales del código a evaluar, por lo que es necesario conocer la lógica del programa. (30)

Técnica de caja negra o funcional: realizan pruebas sobre la interfaz del programa aprobar, entendiendo por interfaz las entradas y salidas de dicho programa. No es necesario conocer la lógica del programa, únicamente la funcionalidad que debe realizar.

Para la validación del sistema desarrollado, se tuvo en cuenta la técnica de caja negra o funcional, la cual se detallará a continuación:

3.5.2 Pruebas de caja negra o funcional.

También conocidas como pruebas de comportamiento, estas pruebas se basan en la especificación del programa o componente a ser probado para elaborar los casos de prueba. El componente se ve como una “Caja Negra” cuyo comportamiento solo puede ser determinado estudiando sus entradas y las salidas obtenidas a partir de ellas. No obstante, como el estudio de todas las posibles entradas y salidas de un programa sería impracticable, se selecciona un conjunto de ellas sobre las que se realizan las pruebas. Para seleccionar el conjunto de entradas y salidas sobre las que trabajar, hay que tener en

cuenta que en todo programa existe un conjunto de entradas que causan un comportamiento erróneo en el sistema y como consecuencia producen una serie de salidas que revelan la presencia de defectos. Dado que el objetivo final de la prueba es encontrar una serie de datos de entrada cuya probabilidad de pertenecer al conjunto de entradas que causan dicho comportamiento erróneo sea lo más alto posible.

3.5.3 Aplicación de pruebas de caja negra

Como se menciona anteriormente una de las pruebas realizadas a la solución propuesta son las pruebas de funcionalidad que se encuentran contenidas dentro de las pruebas de caja negra. Estas pruebas se realizan a través de casos de pruebas insertando valores válidos e inválidos para verificar que:

- ✓ Se aplique apropiadamente cada regla del negocio.
- ✓ Los resultados esperados ocurran cuando se usen datos válidos.
- ✓ Sean desplegados los mensajes apropiados de error y precaución cuando se usan datos inválidos.

Durante la realización de las pruebas al Asistente VUCE se realizaron casos de pruebas (5 para el módulo de **Solicitudes**, 9 para **Documentos Complementarios**) con cada uno de sus Escenarios de Prueba (EP), el caso de prueba referente al requisito descrito con anterioridad *Guardar Solicitud facilidad*.

3.6 Escenario de caso de prueba.

A continuación se muestra un escenario del diseño de caso de prueba realizado al requisito funcional guardar Régimen temporal. En los escenario de prueba del mismo se valora la respuesta del sistema para un posible valor de la variable en cuestión sea este válido o incorrecto. (Ver documento *Diseño de caso de prueba de la aplicación*)

- ✚ Diseño de caso de prueba de Régimen temporal. Escenario “Adicionar mercancía”:

CAPÍTULO 3 IMPLEMENTACIÓN Y PRUEBA

EC	1.2 Se adiciona los	Adicionar Datos de Mercancía.	Datos de Mercancía.	Partida	Plazo	Control	Régimen	Descripción de la mercancía	Motivo de solicitud de la mercancía	Calle	Entre	Y	Número	Barrio	Edificio	Apartamento	Provincia	Municipio	Teléfonos	Tipo de teléfono	
				V	V	V	V	V	V	V	V	V	V	V	N/A	N/A	V	V	V	V	Actualiza el listado de las mercancías.
				80005206	15	AGR	2100	Productos Lácteos	Importar mercancías al país	Carmen	Gloria	Jusú Hidalgo	198	Plaza	vacio	vacio	CIUDAD DE LA HABANA	Regla	2015897	Fijo	
				V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	
				8895206	15	AGR	2100	Productos Lácteos	vacio	vacio	vacio	vacio	vacio	vacio	vacio	189	CIUDAD DE LA HABANA	Regla	vacio	vacio	
				I	I	I	I	I	V	V	V	V	V	V	V	V	I	NP	V	V	Se informa que para adicionar una mercancía es necesario llenar los datos requeridos.
				vacio	vacio	vacio	vacio	vacio	vacio	vacio	vacio	vacio	vacio	vacio	vacio	vacio	vacio	vacio	vacio	vacio	

Luego de ejecutarse las pruebas funcionales divididas en dos iteraciones se identificaron 10 no conformidades en la primera iteración y ninguna en la segunda como se muestra en la siguiente Figura 2.8. Las no conformidades detectadas fueron sobre errores ortográficos en los Diseños de Casos de Prueba en 3 ocasiones y 7 funcionales. Estas inconformidades detectadas fueron resueltas en un cien por ciento, contribuyendo a la mejora del funcionamiento del sistema. Luego de la revisión de la aplicación por parte del equipo de desarrollo del proyecto, hubo una aceptación por parte del cliente con el resultado obtenido.

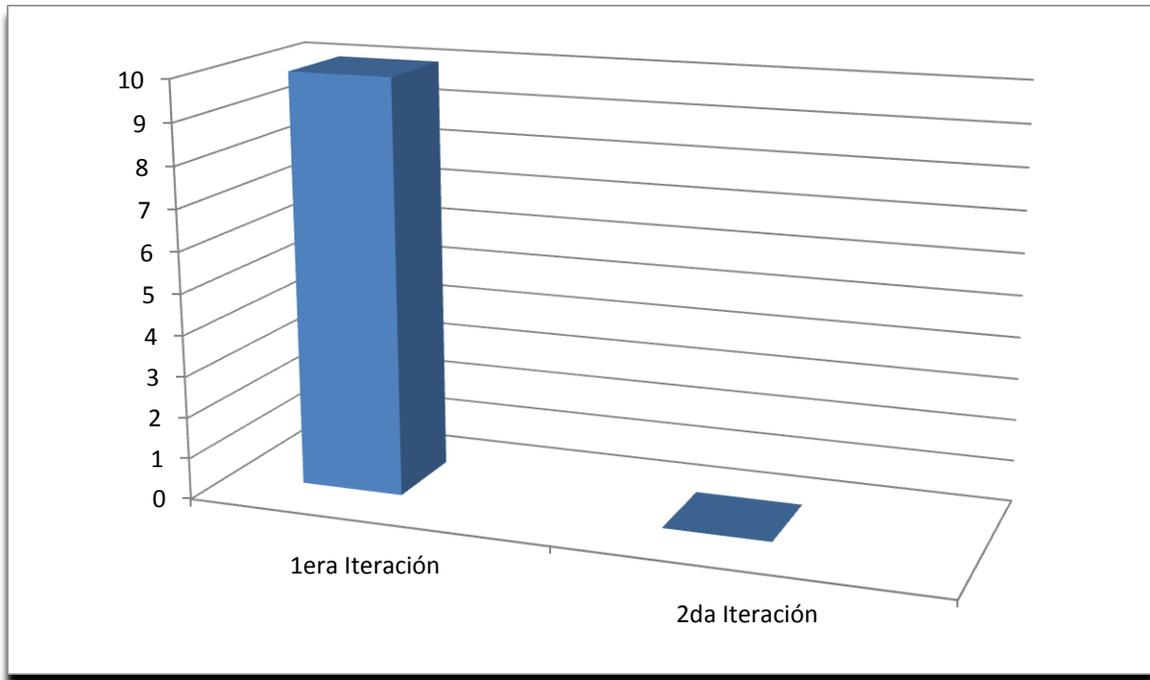


Figura 2.8. Resultado de las pruebas tras haber aplicado los diseños de casos de pruebas.

3.7 Conclusiones parciales.

- La calidad del sistema implementado fue la premisa fundamental en el desarrollo de este capítulo. En ese sentido se obtuvieron los casos de prueba que ayudan a comprobar el correcto funcionamiento del producto.
- Los resultados obtenidos al finalizar las pruebas fueron satisfactorios; con 0 no conformidades, se demostró el correcto funcionamiento de la aplicación.
- Desde el punto de vista funcional todos los requisitos realizan las funcionalidades requeridas y responden a las necesidades del cliente.

Conclusiones

A partir de estudio del arte realizado y todo y trabajo investigativo que estuvo guiado por el modelo de desarrollo del centro CEIGE, se desarrollo una aplicación que permitirá a las entidades crear los documentos XML para la Ventanilla Única de Comercio Exterior de Cuba facilitando la realización de los trámites requeridos.

Con la aplicación de técnicas como la entrevista, la tormenta de ideas y la observación se definieron los requisitos que recogen las funcionalidades necesarias para garantizar que el Asistente VUCE satisfaga las necesidades del cliente.

Mediante métricas definidas de diseño se validó que el realizado en la presente investigación fuera correcta, tal es el caso de la métrica Tamaño Operacional de Clases. Así como se llevó a cabo el modelado de todos los componentes necesarios para su implementación.

La implementación a partir del correcto modelado de la aplicación se desarrolló bajo la idea de software libre utilizando en todo momento herramientas de libre acceso. Finalmente se obtuvo de la implementación el sistema capaz de resolver el problema que enfrentan las entidades externas con los documentos XML.

Con la validación del sistema se comprobó que la propuesta de solución cubre todas las necesidades que enfrenta hoy la Ventanilla de Comercio Exterior de Cuba.

Recomendaciones.

A lo largo de esta investigación fueron identificados algunos elementos que no fueron abordados al no ser parte del alcance inicial, pero se considera que son elementos que incrementarían la utilidad de este trabajo, se recomienda:

- Incrementar la aplicación con nuevos módulos de la recepción de documentos correspondiente a la Ventanilla Única de Comercio Exterior.
- Realizar la integración de la aplicación con la VUCE de Cuba.

REFERENCIA BIBLIOGRÁFICA

Referencias Bibliográficas

1. AGR. SITIO WEB DE LA ADUANA CUBANA. In: [online]. [Accessed 25 febrero 2013]. Available from: <http://www.aduana.co.cu>.
2. Abg. Juan José Gaviria Menéndez. VENTANILLA ÚNICA ECUATORIANA. In: [online]. S.l. [Accessed 22 abril 2013]. Available from: www.asapra.com/.../11.00-3-Apresentacao_Juan%20Jose_Gaviria_M...
3. ARTHA RODRIGUEZ Director. single window FOR FOREIGN TRADE IN COLOMBIA [online]. S.l. [no date]. [Accessed 22 abril 2013]. Available from: www.wto.org.
4. Single Window in Korea. In: UNECE UN/CEFACT [online]. S.l. [Accessed 22 abril 2013]. Available from: http://www.google.com/cu/search?sclient=psy-ab&biw=1024&bih=350&q=single+window+FOR+FOREIGN+TRADE+IN+COLOMBIA&oq=&gs_l=&psj=1.
5. EUGO. In: [online]. [Accessed 22 abril 2013]. Available from: <http://www.eugo.es/>.
6. redgeal_web2.pdf. In: [online]. [Accessed 23 abril 2013]. Available from: redgealc.org.
7. Que es un documento digital. In: [online]. [Accessed 22 enero 2013]. Available from: http://www.tecnologiavirtual.com/index.php?option=com_content&view=article&id=120&Itemid=129.
- 8 CODINA, Luis. Las propiedades de la información digital. En: el profesional de la información. Barcelona, Diciembre, 2001, vol. 10, no. 12. p. 19
- 9- ARCHIVO GENERAL DE LA NACIÓN. Acuerdo 07 del 29 de junio de 1994, REGLAMENTO GENERAL DE ARCHIVOS. (pdf). (Consultado: 3-05-2011). Disponible en: <http://www.archivogeneral.gov.co/index.php?idcategoria=2012#>

REFERENCIA BIBLIOGRÁFICA

10. Convenio de codificación para XML | Marco de Desarrollo de la Junta de Andalucía. In: [online]. [Accessed 22 enero 2013]. Available from: <http://www.juntadeandalucia.es/servicios/madeja/contenido/libro-pautas/10>.
9. Manual XML - Google Drive. In: [online]. [Accessed 22 enero 2013]. Available from: <https://docs.google.com/document/d/1g2svS61-eCPCcq15vvvDGUvqezPnll0-wLxH1RM9r3s/edit?pli=1>.
10. XML y su importancia en la actualidad - Gaiak - Euskonews. In: [online]. [Accessed 22 enero 2013]. Available from: <http://www.euskonews.com/0471zbk/gaia47101es.html>.
11. Guía Breve de Servicios Web. In: [online]. [Accessed 22 enero 2013]. Available from: <http://www.w3c.es/Divulgacion/GuiasBreves/ServiciosWeb>.
12. Servicios Web XML en código administrado. In: [online]. [Accessed 22 enero 2013]. Available from: <http://msdn.microsoft.com/es-es/library/xy59yt45%28v=vs.80%29.aspx>.
13. Visión General de Servicios Web XML. In: [online]. [Accessed 22 enero 2013]. Available from: <http://es.scribd.com/doc/8739651/Vision-General-de-Servicios-Web-XML>.
14. Web o desktop? In: [online]. [Accessed 22 enero 2013]. Available from: <http://software.guisho.com/web-o-desktop>.
15. Fundamentos de la POO. In: [online]. [Accessed 23 enero 2013]. Available from: <http://msdn.microsoft.com/es-es/library/bb972232.aspx>.
16. ¿Qué es Java y por qué lo necesito? - Página de fácil impresión. In: [online]. [Accessed 23 abril 2013]. Available from: http://www.java.com/es/download/faq/whatis_java.xml?printFriendly=true.
17. PROGRAMACIÓN 2.0 | Soluciones Internet. In: [online]. [Accessed 23 abril 2013]. Available from: <http://www.soluciones-internet.es/programacion/>.

REFERENCIA BIBLIOGRÁFICA

18. I.4. Comparativa con otros lenguajes de programación O.O. In: [online]. [Accessed 23 abril 2013]. Available from: http://zarza.usal.es/~fgarcia/doc/tuto2/l_4.htm.
19. Visual Paradigm para UML - Programación - herramienta para desarrollo de aplicaciones utilizando modelado UML* ideal para los que están interesados en construcción de sistemas a gran escala y necesitan confiabilidad | www.targetware.com.ar. In: [online]. [Accessed 23 abril 2013]. Available from: <http://www.software.com.ar/visual-paradigm-para-uml.html>.
20. Requerimientos Funcionales y No Funcionales (RF/RNF). In: [online]. [Accessed 10 abril 2013]. Available from: <http://ingenieriadesoftware.bligoo.com.mx/requerimientos-funcionales-y-no-funcionales-rf-rnf>.
21. Grupo UML. In: [online]. [Accessed 23 abril 2013]. Available from: <http://umlpanamericana.blogspot.com/>.
22. usuarios una herramienta: Topics by WorldWideScience.org. In: [online]. [Accessed 10 abril 2013]. Available from: <http://worldwidescience.org/topicpages/u/usuarios+una+herramienta.html>.
23. UML, BPMN and Enterprise Architecture Tool for Software Development. In: [online]. [Accessed 23 abril 2013]. Available from: <http://www.visual-paradigm.com/>.
24. YUNIERSY PEÑA LEMUS, Yudisleidys and Hernández Díaz. *IMETSE–Sistema de METricas para evaluar el diseño* [online]. S.I. [no date]. [Accessed 23 abril 2013]. Available from: <https://correo.estudiantes.uci.cu/zimbra/#1>.
25. KIDD, J, Lorenz, M. *Object-Oriented Software Metrics* [online]. S.I. 1994. [Accessed 23 abril 2013]. Available from: <https://correo.estudiantes.uci.cu/service/home/~/~RodHarRama00.pdf?auth=co&loc=es&id=61169&part=2>.

REFERENCIA BIBLIOGRÁFICA

26. TECNICAS DE EVALUACION DE SOFTWARE
is.ls.fi.upm.es/docencia/erdsi/Documentacion_Evaluacion_7.pd[17/10/2006]
27. Pruebas de Caja negra y Caja Blanca
<http://pruebasdesoftware.com/cajablancaycajanegra.htm>[2005]
28. Ranking Lenguajes de Programación Marzo 2011. In: [online]. [Accessed 20 May 2013]. Available from: http://www.desarrolloweb.com/de_interes/ranking-lenguajes-programacion-marzo-2011-4994.html.
29. Metodología Gestión de
Requerimientos <https://sites.google.com/site/metodologiareq/capitulo-ii/tecnicas-para-identificar-requisitos-funcionales-y-no-funcionales>
- 30 2nd enero 2007 by Javierpello [Accessed 20 May
<http://www.softqanetwork.com/functional-testing>

GLOSARIO DE TÉRMINO

Anexos

Descripción de requisitos funcionales.

Especificación de requisito Guardar Lista Empaque

Especificación de requisito Guardar Anulación de Documento Complementario

Precondiciones	El usuario selecciona la opción “Anulación de Documento Complementario” contenida en el submenú “Documento Complementario” del menú “Despacho Comercial”.
-----------------------	---

Flujo de eventos

Flujo básico Anulación del Documento Complementario

1. El usuario puede seleccionar la opción cargar documento ver Requisito extendido Cargar Documento. Se selecciona una de las opciones Adicionar o Eliminar.
2. Si se selecciona la opción Eliminar, ver Flujo Alternativo 2.a Eliminar Factura de Venta Anulada.
3. Se introducen los siguientes datos: número de documento, año y se carga el tipo de documento.
4. Se introduce el tipo de documento.
5. Se actualiza la tabla con el listado de los documentos a anular.
6. Se ejecuta el paso 1 tantas veces como desee.
7. Escoger la opción Aceptar.
Si escoge la opción Cancelar, ver Flujo Alternativo 7.a Cancelar Anulación.
8. Realiza las validaciones correspondientes al negocio.
En caso de existir errores ver Flujo Alternativo de Alternativo 8.a Datos Incorrectos.
9. El usuario selecciona la opción de guardar la información en un documento XML o enviarla y el sistema le brinda la posibilidad de guardarlo donde desee. El usuario guarda el

GLOSARIO DE TÉRMINO

documento.

10. Concluye el requisito.

Pos-condiciones

1. Se guarda el fichero.

Flujos alternativos

Flujo alternativo 2.a Eliminar Factura de Venta Anulada.

- 1 Actualiza la tabla con el listado de las facturas eliminando la factura seleccionada.
- 2 Ejecuta el *paso 5 del Flujo Básico de Eventos Anulación de Documento Complementario*.

Pos-condiciones

1 Se actualiza la tabla con el listado de los documentos a anular.

Flujo alternativo 7.a Cancelar Anulación.

1 Concluye el requisito

Pos-condiciones

1 No se registran los datos.

Flujo alternativo 8.a Datos Incorrectos

- 1 Muestra un mensaje notificando el/los error(es) detectado(s).
- 2 Seleccionar la opción "Aceptar" del mensaje mostrado.
- 3 Ejecuta el *paso 1 del Flujo Básico de Eventos Anulación de Documento Complementario*

Pos-condiciones

1 Se muestra un mensaje notificando el/los error(es) detectado(s).

Conceptos

AnulacionDoc

- codigo
- descripción
- fechaInicio

GLOSARIO DE TÉRMINO

- fechaFin

Requisitos Requisito Extendido Cargar Documento.

especiales Requisito Extendido Cancelar Documento.

Asuntos N/A

pendientes

The screenshot shows the 'Sistema de Órganos Aduaneros' interface. At the top, there is a navigation bar with the following tabs: 'Ventanilla Unica', 'Presentación Documentos', 'Liquidación', 'Actualizaciones', 'Alcance', 'Devoluciones', 'Temporalidades', 'Canales', and 'Levante Manual'. The main header includes the 'Sitio Web' logo and the text 'Aduana General de la Republica de Cuba'. The central window is titled 'Registrar Anulación de Documento Complementario' and contains a table with the following structure:

No. Documneto	Año	Tipo de Documneto
		- Select One -

At the top of the window, there are three buttons: 'Adicionar' (with a plus icon), 'Modificar' (with a pencil icon), and 'Eliminar' (with a trash icon). At the bottom, there are two buttons: 'Aceptar' and 'Cancelar'.

GLOSARIO DE TÉRMINO

Glosario de términos

Artefacto: pieza de información tangible que es creada, modificada y usada por los trabajadores al realizar actividades.

Diagrama: representación gráfica de un conjunto de elementos. Visualizan un sistema desde diferentes perspectivas.

JSON: (JavaScript Object Notation) Notación de Objetos JavaScript, es un formato ligero para el intercambio de datos. JSON es un subconjunto de la notación literal de objetos de JavaScript que no requiere el uso de XML.

Mercancías: todos los bienes corporales muebles de comercio o no, con la sola excepción de los Efectos Personales de los viajeros.

Metodología: es un proceso de software detallado que define con precisión los artefactos, roles y actividades involucradas.

Tecnologías de la Información y las Comunicaciones(TIC's): Se denominan Tecnologías de la Información y las Comunicaciones (TIC's) al conjunto de tecnologías que permiten la adquisición, producción, almacenamiento, tratamiento, comunicación, registro y presentación de informaciones, en forma de voz, imágenes y datos contenidos en señales de naturaleza acústica, óptica o electromagnética.

UML: es el Lenguaje de Modelado Unificado para detallar, construir, visualizar y documentar las partes o artefactos de un software.

XMLSchema: es un lenguaje de esquema utilizado para describir la estructura y las restricciones de los contenidos de los documentos XML de una forma muy precisa, más allá de las normas sintácticas impuestas por el propio lenguaje XML.