

Universidad de las Ciencias Informáticas

Facultad 3



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

Título: “Desarrollo de una herramienta para la gestión de la información adelantada establecida por la Organización Marítima Internacional del sistema Despacho de Medios de Transporte Internacional.”

Autora: Isis Morell González

**Tutores: Ing. Manuel Ramón Almaguer
Ing. Gretter Lisbet Rodríguez Valdés
Ing. Leydis Castellanos Cobas**

Ciudad de la Habana, junio del 2014

“Año 56 de la Revolución”

DECLARACIÓN DE AUTORÍA

Declaro que soy la única autora de este trabajo y autorizo a la Facultad 3 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

Isis Morell González

Ing. Gretter Lisbet Rodríguez Ing. Manuel Ramón Almaguer Ing. Leydis Castellanos

DATOS DE CONTACTO

- **Ing. Manuel Ramón Almaguer Ochoa.** Ingeniero en Ciencias Informáticas. Asistente. 6 Años de experiencia en el tema, ha sido tutor y oponente de múltiples tesis.
- **Ing. Gretter Lizbeth Rodríguez Valdés.** Ingeniera en Ciencias Informáticas. Recién Graduado en Adiestramiento. 4 Años de experiencia en el tema.
- **Ing. Leydis Castellanos Cobas.** Ingeniera en Ciencias Informáticas. Especialista General. 4 Años de experiencia en el tema.

AGRADECIMIENTOS

A todos los que de alguna forma hicieron posible la realización de este trabajo.

A mis padres, mi hermano, mi familia y mi novio por su apoyo incondicional, y por estar presente en cada momento en que se ha hecho necesario.

A mis amigos José Daniel y Luismy que siempre estuvieron a mi lado durante todos estos años.

A mis tutores por su entrega, apoyo, consejos y paciencia.

A todos mis compañeros de proyecto y brigada por todos estos años de amistad y ayuda mutua.

A todo aquel que ayudó de una forma u otra que les llegue mi más sincero

AGRADECIMIENTO

DEDICATORIA

A mis padres, por su ejemplo, por enseñarme a luchar por lo que quiero.

A mi mamuchi, por ser la mejor madre del mundo, por su amor, ejemplo y espíritu positivo y de lucha, por dedicarme cada momento, cada acto y cada pensamiento.

A mi papa, por sus consejos y su ejemplo, por el orgullo que siento de ser su hija.

A mi papi, por ser mi ejemplo y mi guía durante toda mi vida, por darme amor, seguridad y apoyo en todo momento.

A mi novio Nilier, por todos estos años juntos, por todo su amor y dedicación, por hacer de mí la mujer más feliz del mundo, por ser lo mejor que me ha pasado en la vida y claro, por darme los mejores suegros que alguien desearía tener.

A toda mi familia, mis primas que aunque esten lejos siempre estan en mi corazón, mi cuñada Marjo, mis suegros, Norlan, Suset por ser un ejemplo de ello, por ser fieles centinelas de mi presente y futuro, por apoyarme en cada momento.

A mis amigos de ahora y de toda la vida, porque las experiencias con ellos compartidas son inolvidables para mí.

¡A todos por ser parte de este momento tan importante en mi vida!



La soberanía del hombre está oculta en la dimensión de sus
conocimientos.

Ernesto Senevara

ÍNDICE

RESUMEN	9
INTRODUCCIÓN	10
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	14
1.1 Introducción.....	14
1.2 Intercambio electrónico de documentos	14
1.3 Organización Marítima Internacional y Documentos OMI.....	15
1.4 Proceso de recepción electrónica de la información adelantada establecida por la OMI	20
1.5 Modelo de Desarrollo de Software	21
1.6 Herramientas y lenguajes para el desarrollo de software	22
1.7 Metodologías y herramientas de diseño	25
1.8 Métricas.....	29
1.9 Conclusiones parciales.....	32
CAPÍTULO 2: PROPUESTA DE SOLUCIÓN	33
2.1 Introducción.....	33
2.2 Modelo Conceptual	33
2.3 Diagrama del proceso de negocio	35
2.4 Descripción de la solución	36
2.5 Requisitos del Sistema	36
2.6 Validación de Requisitos	43
2.7 Patrones de diseño implementados.....	44
2.8 Modelo de diseño	46
2.8.1 Diagrama de clases del diseño.....	46
2.8.2 Diagrama de paquetes	48
2.8.3 Diseño de la Base de Datos	49
2.8.4 Validación del Diseño	51
2.9 Implementación del sistema	55
2.9.1 Diagrama de componentes.....	55
2.9.2 Estándares de codificación.....	56
2.9.3 Tratamiento de errores	57
2.9.4 Comunicación entre las capas.....	57

2.10 Conclusiones Parciales	62
CAPÍTULO 3: VALIDACIÓN DE LA SOLUCIÓN.....	63
3.1 Introducción	63
3.2 Verificación del sistema	63
Pruebas de Rendimiento y Estrés	63
Pruebas de Caja Negra.....	63
Pruebas de Caja Blanca.....	65
3.3 Validación del sistema	71
3.3 Validación de las variables.....	73
3.4 Conclusiones Parciales.....	74
CONCLUSIONES	75
REFERENCIAS BIBLIOGRÁFICAS	76
BIBLIOGRAFÍA	78

RESUMEN

Uno de los procesos que se desarrollan en la Aduana General de la República de Cuba (AGR) es la recepción de información adelantada establecida por la Organización Marítima Internacional (OMI). Actualmente, debido a la gran cantidad de información que se recibe existe demora en el procesamiento de documentos OMI que se deben tener antes del arribo de un buque. El proceso no es ágil, ni eficaz debido a que es realizado utilizando hojas de cálculo con las cuales se hace muy engorroso el trabajo, por lo que el trámite demora más de lo previsto y se cometen numerosos errores, tales como entradas de datos incorrectos, inconsistencias, etc. A su vez los analistas del departamento de Lucha Contra el Fraude (LCF) necesitan mejorar la recepción de la información, porque existe una latencia en el intercambio de la información que se recibe digitalmente, razón por la que en ocasiones se recibe incompleta y tardía.

Teniendo en cuenta los elementos planteados anteriormente se hace necesario desarrollar una herramienta que permita disminuir el tiempo de consulta de la información adelantada establecida por la Organización Marítima Internacional del sistema Despacho de Medios de Transporte Internacional. Con este propósito se realizó una descripción de las herramientas y tecnologías utilizadas, dando paso a la propuesta de solución. En dicha propuesta se muestran los artefactos generados durante el análisis, diseño e implementación, lográndose medir la calidad del trabajo realizado aplicando métricas al diseño y pruebas funcionales a la aplicación obtenida.

Palabras claves: AGR, OMI, recepción de información adelantada.

INTRODUCCIÓN

El creciente desarrollo de las Tecnologías de la Información y las Comunicaciones (TIC) ha brindado nuevas oportunidades en la vida de las personas y el desempeño de muchas empresas. La puesta en práctica de las TIC ha permitido la informatización de los procesos que en un principio se tornaban complejos y de gran envergadura, permitiendo a las empresas mostrar mejorías en la eficiencia y calidad de los servicios que brindan. Cuba se suma a la lista de países que cuentan con un desarrollo progresivo en el campo de la Informática y las Comunicaciones. El proceso de informatización de la sociedad cubana, ha propiciado el aumento del uso de herramientas informáticas en los principales sectores del país. En tal sentido, varios organismos se han dado a la tarea de mejorar los servicios que brindan utilizando las TIC, entre estos se encuentra la Aduana General de la República de Cuba (AGR).

La AGR es un órgano de control que garantiza la seguridad y protección de la sociedad socialista y de la economía nacional, así como la recaudación fiscal y las estadísticas de comercio exterior. Para el cumplimiento de sus objetivos, es creada el área de Lucha Contra el Fraude (LCF) que se encarga de enfrentar las acciones terroristas, de narcotráfico, los contrabandos comerciales o no comerciales y las operaciones que ponen en riesgo el patrimonio cultural y natural del país. (1)

El control aduanero a los Medios de Transporte Internacional (MTI) a su entrada y salida del país es una de las responsabilidades de mayor peso en el área de LCF de la AGR. Para lograr de forma efectiva un control adecuado sobre los medios de transporte es importante para los analistas de LCF contar con información histórica y/o adelantada de los MTI. La información adelantada establecida por la Organización Marítima Internacional (OMI) se compone de 6 documentos OMI que declaran la información relacionada con la carga, provisiones, pasajeros y tripulantes que transporta un buque.

En el año 2005 se estableció el Sistema Único de Aduanas (SUA) donde colaboraron los especialistas del Centro de Automatización para la Dirección y la Información de la AGR (CADI) con trabajadores y estudiantes de la Universidad de las Ciencias Informáticas (UCI). En este proyecto se desarrolló un subsistema de despacho de MTI que se enfoca en el proceso de tramitación ante la Aduana de toda la documentación que declara la carga, provisiones, pasajeros y tripulantes que transporten los MTI.

Actualmente, debido a la gran cantidad de información que se recibe existe demora en el procesamiento de documentos OMI que se deben tener antes del arribo de un buque. La recepción de estos documentos se realiza por vía electrónica. El proceso no es ágil, ni eficaz debido a que es realizado utilizando hojas de cálculo con las cuales se hace muy engorroso el trabajo, por lo que el trámite demora más de lo previsto y se cometen numerosos errores, tales como entradas de datos incorrectos, inconsistencias, etc. A su vez los analistas de LCF necesitan mejorar la recepción de la información, porque existe una latencia en el intercambio de la información que se recibe digitalmente, esto se debe a que esta información es tramitada por muchos intermediarios hasta llegar a su destino, razón por la que en ocasiones se recibe incompleta y tardía.

Para dar solución a la problemática planteada se establece como **problema**: ¿Cómo disminuir el tiempo de consulta de la información adelantada establecida por la Organización Marítima Internacional que recibe la Aduana General de la República por vía electrónica?

En tal sentido el **objeto de estudio** está centrado en el intercambio de documentos tramitados por vía electrónica. Delimitando como **campo de acción**: procesos de recepción, procesamiento, almacenamiento y consulta de documentos OMI tramitados por vía electrónica en la Aduana General de la República.

Se plantea como **objetivo general** para solucionar el problema: desarrollar una herramienta que permita disminuir el tiempo de consulta de la información adelantada establecida por la Organización Marítima Internacional del sistema Despacho de Medios de Transporte Internacional.

Del mismo se derivan los siguientes **objetivos específicos**:

1. Definir el estado actual del proceso de gestión de la información adelantada establecida por la Organización Marítima Internacional del sistema Despacho de Medios de Transporte Internacional para obtener una mejor comprensión del funcionamiento de este proceso.
2. Describir los requisitos que deben estar presentes en la herramienta que se propone desarrollar.
3. Diseñar e implementar la solución modelada que permita la gestión de la información establecida por la Organización Marítima Internacional.
4. Verificar y validar la solución propuesta para comprobar que la misma responde correctamente a las necesidades del cliente.

Se definen como **tareas investigativas**:

1. Elaboración de la fundamentación teórica y la revisión bibliográfica.
2. Identificación y descripción de los requisitos de la herramienta.
3. Validación de los requisitos descritos.
4. Confección del diagrama de clases del diseño.
5. Diseño de la base de datos.
6. Descripción del modelo de datos.
7. Implementación de los requisitos descritos.
8. Diseño de los casos de prueba a aplicar.
9. Aplicación de las pruebas al sistema.
10. Documentación de los resultados de las pruebas realizadas.

Se plantea como **idea a defender** que con el desarrollo de una herramienta que permita la gestión de la información adelantada establecida por la Organización Marítima Internacional del sistema Despacho de Medios de Transporte Internacional disminuirá el tiempo de consulta de la información adelantada establecida por la Organización Marítima Internacional que recibe la Aduana General de la República por vía electrónica.

Durante el transcurso de esta investigación se hará uso de varios **métodos científicos**:

Métodos Teóricos:

Analítico – Sintético: este método permite realizar un análisis de la teoría y documentos facilitando la obtención de los elementos y conocimientos más relevantes que se relacionan con el proceso de recepción electrónica de información adelantada establecida por la Organización Marítima Internacional.

Modelación: como su nombre lo indica este método se encarga de realizar una reproducción simplificada de la realidad, es uno de los más importantes en el campo de la construcción de software y se utiliza para la creación de todos los artefactos necesarios en el diseño como el diagrama de procesos de negocio, diagrama de clases y otros.

Métodos Empíricos:

Entrevistas: se utiliza para comprobar los problemas existentes en la demora en el tiempo de consulta de la información adelantada establecida por la Organización Marítima Internacional que recibe la Aduana General de la República por vía electrónica.

Medición: se utiliza para comprobar el correcto funcionamiento de la herramienta a través de la realización de pruebas y la aplicación de métricas.

Como posible resultado se obtendrá una herramienta que permita disminuir el tiempo de consulta de la información adelantada establecida por la Organización Marítima Internacional del sistema Despacho de Medios de Transporte Internacional.

La presente investigación **se estructura** de la siguiente manera: Introducción, Tres capítulos, Conclusiones, Recomendaciones, Bibliografía y Glosario de Términos.

Capítulo 1: Fundamentación teórica

En este capítulo se realizará un análisis del proceso de recepción de la información adelantada establecida por la Organización Marítima Internacional. Además, se analizan aspectos fundamentales para una mejor comprensión de la problemática. Se describirán los lenguajes de programación, las herramientas y tecnologías a utilizar, entre otros aspectos de vital importancia para el desarrollo de la investigación.

Capítulo 2: Propuesta de solución

Durante este capítulo se abordarán temas correspondientes al desarrollo de la solución. Se describirán las características de la solución que se propone y se expondrán algunos elementos fundamentales de la implementación de la herramienta. Además serán generados los artefactos definidos en el modelo de desarrollo del Departamento de Soluciones para la Aduana correspondientes a las actividades de diseño e implementación.

Capítulo 3. Validación de la Solución

En este capítulo se exponen los resultados de las pruebas realizadas a la solución para determinar posibles no conformidades, así como determinar si se cumplió con los requisitos funcionales y no funcionales que debe cumplir la herramienta a desarrollar.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

En este capítulo se realiza el estudio del estado del arte del proceso de recepción electrónica de la información adelantada establecida por la Organización Marítima Internacional. Se describe como se realiza este proceso para una mejor comprensión de la problemática. Se caracterizan las tecnologías para el desarrollo web que serán utilizadas en el desarrollo de la solución.

1.2 Intercambio electrónico de documentos

El intercambio electrónico de documentos es la transmisión estructurada de datos entre organizaciones por medios electrónicos. Se utiliza para transferir documentos electrónicos o datos de negocios de un sistema computacional a otro.

En la actualidad existen varios sistemas automatizados que se encargan del intercambio de documentos de forma electrónica. Cada uno de estos sistemas se orienta específicamente a las características y necesidades de la entidad donde será utilizado. En años anteriores, debido a la necesidad de recepcionar documentos necesarios para la realización de diferentes módulos del sistema GINA, se han realizado estudios del estado del arte referentes al proceso de recepción y control de documentos. Entre los ejemplos de trabajos de diploma en los que se analizan las soluciones informáticas existentes relacionadas con el proceso de recepción de documentos se encuentran:

- ✓ *Modelado de Negocio y Requerimientos de los procesos de recepción y control de los Documentos Complementarios al Despacho Comercial, de la autora Leydis Castellanos Cobas. (2)*
- ✓ *Módulo de gestión de la información asociada a la Aduana en la Ventanilla Única, de los autores Kamelia Molina Conde y Yorlen Guirado Más. (3)*
- ✓ *Componente para Gestionar Documentos Complementarios en los Procesos aduaneros del sistema GINA, de la autora Yakelin Milagro León López, entre otras. (4)*

Estas investigaciones han revelado que existen varios sistemas encargados de la recepción de documentos como por ejemplo:

Sistema Aduanero Automatizado (SIDUNEA): El SIDUNEA es una herramienta informática para el control y administración de la gestión aduanera. Este sistema actualmente es usado con

éxito en más de 90 países y puede ser configurado de acuerdo a las normativas y requisitos de las Aduanas de cada país.

Sistema Aduanero Automatizado Integral (SAAI M3): Es un sistema aduanero diseñado para la elaboración y validación de pedimentos de importación, exportación, embarque parcial complementario y extracción, entre otros.

Tecnología de Información para el Control Aduanero (TICA): El Proyecto TICA es el resultado de la adaptación en la Aduana de Costa Rica del sistema de información GIA/LUCIA, un sistema desarrollado en Uruguay para apoyar la gestión operativa de la aduana. El mismo se obtuvo como una donación del gobierno uruguayo a este país.

Entre las ventajas ofrecidas por los mismos se encuentran:

- ✓ Optimización de los tiempos y recursos del proceso aduanero.
- ✓ Aplicación de la ley con toda justicia.
- ✓ Detección de errores en los valores de la declaración.
- ✓ Reducción del contrabando.
- ✓ Administración efectiva del proceso de despacho.

A pesar de que varios sistemas brindan una amplia gama de funcionalidades ninguno cumple con las especificaciones del negocio que se necesita en la realización de los procesos de recepción electrónica de información adelantada establecida por la Organización Marítima Internacional en la AGR. Además, tampoco son compatibles con la política que sigue el país de migrar a plataforma libre. Es por ello que se necesita desarrollar una solución donde se pongan de manifiesto las ventajas brindadas por estos sistemas.

1.3 Organización Marítima Internacional y Documentos OMI

Organización Marítima Internacional

La Organización Marítima Internacional es el organismo especializado de las Naciones Unidas encargado de adoptar medidas para mejorar la seguridad del transporte marítimo internacional y prevenir la contaminación del mar por los buques. Se ocupa además de asuntos de carácter jurídico, entre ellos la responsabilidad civil y la indemnización y la facilitación del tráfico marítimo internacional. Se constituyó por medio de un convenio adoptado bajo los auspicios de las Naciones Unidas en Ginebra el 17 de marzo de 1948 y se reunió por primera vez en enero de 1959. En la actualidad tiene 156 Estados Miembros. El órgano rector de la OMI es la Asamblea,

que está integrada por los 156 Estados Miembros y se reúne normalmente una vez cada dos años. Aprueba el presupuesto para el próximo bienio y las resoluciones y recomendaciones de carácter técnico elaboradas por los órganos auxiliares durante los dos años anteriores. El Consejo ejerce las funciones de órgano rector entre los períodos de sesiones de la Asamblea y elabora el presupuesto y el programa de trabajo para la Asamblea. El trabajo técnico principal lo efectúan el Comité de Seguridad Marítima, el Comité de Protección del Medio Marino, el Comité Jurídico, el Comité de Cooperación Técnica y el Comité de Facilitación, así como diversos subcomités. (5)

La existencia de esta organización es de vital importancia debido a que el transporte marítimo es un sector de ámbito internacional. Si cada país elaborase su propia legislación sobre seguridad, el resultado sería un laberinto de leyes nacionales distintas y a menudo contradictorias.

Documentos OMI

La Organización Marítima Internacional (OMI) en virtud de las complejidades que venían presentando los buques cuando arribaban a un puerto recibió la tarea en el año de 1962, de elaborar un convenio internacional para ayudar a la facilitación del tráfico marítimo internacional. Para dicho trabajo fueron convocados un grupo de expertos quienes se dedicaron a la elaboración del proyecto del convenio, el cual se aprobó en el año 1965 y entra en vigencia el 05 de Marzo de 1967.

Los objetivos que contemplaba el mencionado convenio fueron:

- ✓ Facilitar el transporte marítimo mediante la simplificación y reducción al mínimo de los trámites, documentos y formalidades relacionados con la llegada, estancia en puerto y salida de los buques que efectúan viajes internacionales.
- ✓ Uniformar y simplificar las medidas para facilitar el tráfico marítimo internacional reduciendo al mínimo las modificaciones que se estimen necesarias a satisfacer las exigencias de orden interno.
- ✓ Estimular la cooperación entre los Gobiernos.

En su anexo, el Convenio para Facilitar el Tráfico Marítimo Internacional de la Organización Marítima Internacional contiene "Estándares" y "Prácticas Recomendadas" sobre las

formalidades, requisitos documentales y procedimientos que aplican a la llegada, estancia y zarpe de un buque, su tripulación, pasajeros, equipaje y carga.

Los Formularios Estandarizados de la OMI, enumeran una lista de documentos que las autoridades públicas pueden solicitar al buque y recomiendan la cantidad máxima de información y el número de copias que deben solicitarse. (6) La OMI elaboró formularios estandarizados para seis de estos documentos. Ellos son:

- ✓ Declaración General OMI
- ✓ Declaración de las Provisiones del Buque
- ✓ Declaración de los Efectos Personales de la Tripulación
- ✓ Lista de Tripulantes
- ✓ Mercancías Peligrosas
- ✓ Lista de Pasajeros

Existen varios documentos que sirven como guía para el tratamiento de los mismos, pero fundamentalmente se recomienda la utilización de la Resolución No 187-2008, en la que se aprueban normas para el Despacho y Control Aduanero de Buques y Aeronaves, así como los Anexos con sus Metodologías, referidos a documentos y sus formatos a presentar a la Aduana por las vías electrónicas y en papel, los que forman parte integrante de esta Resolución. A continuación se analizan los documentos OMI mencionados anteriormente.

OMI Declaración General (OMI-G)

En este documento se recogen de forma genérica los datos asociados al buque. Se especifican aspectos como el nombre de la Compañía, del Agente del buque, nombre y descripción del buque, número IMO, puerto de llegada o salida según corresponda, fecha y hora de llegada o salida según corresponda, estado de abanderamiento del buque, entre otros. Además se especifican datos referentes al capitán del buque y la cantidad de ejemplares de los documentos que fueron entregados para el despacho. Por último se registra la fecha en que se confecciona la Declaración así como la firma del capitán, agente consignatario u oficial del buque debidamente autorizado. La siguiente imagen muestra la estructura del documento OMI Declaración General, la cual forma parte del Anexo No.6 de la Resolución 187-2008. Para mayor información sobre las indicaciones para llenar este modelo diríjase a la resolución mencionada anteriormente.

OMI – DECLARACIÓN GENERAL

		<input type="checkbox"/> Llegada	<input type="checkbox"/> Salida
1.1 Nombre y tipo del buque 1.2 Número IMO 1.3 Distintivo de llamada		2. Puerto de llegada/salida	3. Fecha y hora de llegada/salida
4. Estado de abanderamiento del buque	5. Nombre del capitán	6. Ultimo puerto de escala/Próximo puerto de escala	
7. Certificado de matricula (puerto; fecha; número)		8. Nombre y datos de contacto del agente del buque	
9. Arqueo bruto	10. Arqueo neto		
11. Situación del buque en el puerto (muelle o puesto de atraque)			
12. Pormenores someros referentes al viaje (puertos donde el buque ha tocado y va a tocar, subrayar puertos en los cuales la mercancía que permanece a bordo será descargada)			
13. Descripción somera de la carga			
14. Número de miembros de la tripulación (incluido el capitán)	15. Número de pasajeros	16. Observaciones	
Documentos anexos (indíquese el número de ejemplares)			
17. Declaración de carga	18. Declaración de provisiones del buque		
19. Lista de la tripulación	20. Lista de pasajeros	21. Necesidades del buque en cuanto a instalaciones de recepción de desechos y residuos	
22. Declaración de efectos de la tripulación*	23. Declaración marítima de sanidad*		

24. Fecha y firma del Capitán o del agente u oficial debidamente autorizados

Fig1. Estructura del documento OMI Declaración General. (7)

OMI Lista de Tripulantes (OMI-T)

El documento OMI Lista de Tripulantes recoge datos asociados al buque, haciendo énfasis en los datos referentes a la tripulación del buque. Entre los datos que se especifican se encuentran el nombre y apellidos de los tripulantes, su nacionalidad, fecha y lugar de nacimiento, entre otros. Por último se registra la fecha en que se confecciona el documento así como la firma del capitán, agente u oficial del buque debidamente autorizado. La imagen (ver anexo 1) muestra la estructura del documento OMI Lista de Tripulantes, la cual forma parte del Anexo No.11 de la Resolución 187-2008. Para mayor información sobre las indicaciones para llenar este modelo diríjase a la resolución mencionada anteriormente.

OMI Lista de Pasajeros (OMI-P)

Este documento recoge los datos asociados al buque haciendo énfasis en la información referente a los pasajeros del buque. De los mismos se especifica nombre y apellidos, nacionalidad, fecha y lugar de nacimiento, puerto de embarco y desembarco, entre otros. Por último se registra la fecha en que se confecciona el documento así como la firma del capitán,

agente u oficial del buque debidamente autorizado. Cabe destacar que la presentación de este documento depende del tipo de buque que envía la documentación. Existen varios tipos de buques, entre ellos se encuentran:

- ✓ Buques Ferrys y Cruceros: Para ellos se cumple la obligatoriedad de presentar de forma anticipada los documentos establecidos para las mercancías y los pasajeros.
- ✓ Buques dedicados al tráfico de Cabotaje: Son los buques que cargan mercancías en un puerto del territorio nacional para ser descargadas en otro puerto del territorio nacional. Estos buques al no trasladar pasajeros, no deben entregar información referente a los mismos.
- ✓ Buques en Arribada Forzosa o en Escalas de Emergencia: Arribada forzosa es la entrada de un buque, por causa de mal tiempo u otras causas a un puerto distinto al de su destino. Los buques que entren al país bajo estas circunstancias también serán sometidos al control aduanero y deben entregar el documento Lista de Pasajeros, si procede.

La imagen (ver anexo 2) muestra la estructura del documento OMI Lista de Pasajeros, la cual forma parte del Anexo No.10 de la Resolución 187-2008. Para mayor información sobre las indicaciones para llenar este modelo diríjase a la resolución mencionada anteriormente.

OMI Declaración Efectos de la Tripulación (OMI-E)

Este documento además de recoger los datos asociados al buque, especifica los nombres y apellidos, grado o clase y firma de los tripulantes que poseen efectos sometidos a régimen de adeudo u objeto de prohibición o limitación cuantitativa, por ejemplo vinos, licores, cigarrillos, tabaco, etc. Por último se registra la fecha en que se confecciona el documento así como la firma del capitán, agente u oficial del buque debidamente autorizado. La imagen (ver anexo 3) muestra la estructura del documento OMI Declaración de Efectos de la Tripulación, la cual forma parte del Anexo No.9 de la Resolución 187-2008. Para mayor información sobre las indicaciones para llenar este modelo diríjase a la resolución mencionada anteriormente.

OMI Declaración Provisiones del Buque (OMI-B)

En el documento OMI Declaración de Provisiones del Buque se recoge la información relacionada con el buque y además las provisiones que el mismo transporta, es decir los artículos y su cantidad. Además se registra el lugar donde son almacenados y su período de permanencia. Por último se registra la fecha en que se confecciona el documento así como la firma del capitán, agente u oficial del buque debidamente autorizado. La imagen (ver anexo 4)

muestra la estructura del documento OMI Declaración de Provisiones del Buque, la cual forma parte del Anexo No.12 de la Resolución 187-2008. Para mayor información sobre las indicaciones para llenar este modelo diríjase a la resolución mencionada anteriormente.

OMI Declaración de Mercancías Peligrosas (OMI-M)

Debido a la importancia y particularidades del control de las mercancías peligrosas se presenta un documento independiente en el que se recoge la información referente a las mismas, como por ejemplo el número y tipo de bultos, la masa en kg, el lugar de estiba a bordo, las marcas de identificación del contenedor, entre otros. Por último se registra la fecha y lugar en que se confecciona el documento así como la firma del capitán, agente u oficial del buque debidamente autorizado. La imagen (ver anexo 5) muestra la estructura del documento OMI Declaración de Mercancías Peligrosas, la cual forma parte del Anexo No.8 de la Resolución 187-2008. Para mayor información sobre las indicaciones para llenar este modelo diríjase a la resolución mencionada anteriormente.

1.4 Proceso de recepción electrónica de la información adelantada establecida por la OMI

El desarrollo de las industrias, el transporte y el comercio, ha influido considerablemente en la evolución y modernización de las Aduanas a nivel mundial convirtiéndolas en una administración económica de notable importancia. Actualmente la función de las Aduanas del mundo es facilitar el comercio, hacerlo más práctico, expeditivo y funcional. El transporte internacional es una operación de tránsito aduanero mediante el cual se efectúa el control de MTI, transporte de mercancías o personas de un país a otro; es decir, el transporte que se efectúa entre varias naciones y que son sometidos a varias actividades materiales y legales. (8)

Un elemento fundamental dentro de los procesos de control aduanero a los MTI es la recepción de la información adelantada establecida por la Organización Marítima Internacional. La misma está representada por los documentos Organización Marítima Internacional (OMI) que se enuncian a continuación:

1. OMI-G Declaración General
2. OMI-T Lista de Tripulantes
3. OMI-P Lista de Pasajeros
4. OMI-E Declaración Efectos de la Tripulación
5. OMI-B Declaración Provisiones del Buque
6. OMI-M Declaración de Mercancías Peligrosas

El flujo actual de los procesos involucrados en el enfrentamiento a las ilegalidades asociadas a los MTI, se inicia con la recepción y gestión de la información adelantada perteneciente a los buques. Para conocer más a fondo el tema se describe seguidamente cómo se recibe la información referente a los documentos OMI.

La AGR recibe y controla un grupo significativo de documentos relativos a las cargas, provisiones, pasajeros y tripulantes que transporta un buque. Entre ellos se encuentran los documentos OMI, los cuales son recepcionados por el sistema Ventanilla Única (VU) y validados por el sistema Gestión Integral de Aduanas (GINA). En el proceso de recepción de los documentos OMI participan varias entidades:

Naviera: Entidad con la que el mando del buque, o su consignatario o representante establecen un contrato de negocio. Es la encargada de enviar a la Consignataria Mambisa todos los documentos OMI.

Agencia Consignataria: Departamento que actúa como intermediario para la recepción y emisión de los documentos OMI entre la Naviera y el SUA. La Consignataria Mambisa es la agencia dedicada al manejo y gestión de toda la documentación de los MTI

Sistema Único de Aduanas (SUA): Sistema establecido en el 2005 producto a la colaboración de especialistas del Centro de Automatización para la Dirección y la Información de la AGR (CADI) y trabajadores y estudiantes de la Universidad de las Ciencias Informáticas (UCI).

Módulo de Despacho MTI: Módulo que representa el despacho de entrada y salida de buques y aeronaves; mediante el cual se tramita ante la AGR la documentación que declara la carga, provisiones, pasajeros y tripulantes que transporten los mismos. El mismo garantiza que los analistas de LCF de la AGR reciban la información en forma de reportes.

VU: Sistema encargado de recepcionar los documentos OMI.

GINA: Sistema encargado de realizar las validaciones correspondiente a los documentos.

1.5 Modelo de Desarrollo de Software

El Departamento de Soluciones para la Aduana tiene como principal responsabilidad desarrollar sistemas informáticos para gestionar los procesos de negocio de las entidades aduaneras en su relación con el comercio exterior. Este departamento se rige por el Modelo de Desarrollo de Software del Centro de Informatización de la Gestión de Entidades (CEIGE) como guía para la especificación de las actividades de cada una de las fases del ciclo de vida de los proyectos del

centro teniendo en cuenta los procesos de CMMI (*Capability Maturity Model Integration*) nivel 2 para la Universidad de las Ciencias Informáticas (UCI). Se detallan por tanto los artefactos a generar en cada momento independientemente de las herramientas o métodos que se utilicen para ello. Además, el mismo se caracteriza por ser:

- ✓ Centrado en la arquitectura
- ✓ Iterativo e Incremental
- ✓ Orientado a componentes
- ✓ Ágil y adaptable al cambio

Este modelo dirige las actividades que se realizan por rol, les da una responsabilidad, y define qué tareas deberán ser desarrolladas. Dentro de los artefactos que se generan a partir del modelo de desarrollo se encuentran el **modelo conceptual** y el **diagrama de procesos de negocio** (escritos en notación BPMN (*Business Process Model Notation*, por sus siglas en inglés) logrando mayor comprensión para los usuarios, tanto del negocio, como para los analistas implicados y los que realizan el diseño de la solución). Los artefactos mencionados anteriormente serán generados durante la disciplina de Modelado de Negocio; por otra parte durante la disciplina Análisis y Diseño serán generados el **diagrama de clases del sistema**, artefacto compuesto por las clases, atributos y métodos; también se reflejan las relaciones como la herencia, la composición, la agregación, la asociación, el uso y la cardinalidad; el **diagrama de paquetes** y el **diagrama de componentes** donde se recogen todas las clases, componentes, interfaces, diagramas, paquetes, entre otros. La relación entre los paquetes puede ser por importación o por uso. La importación ocurre cuando se necesitan todos los elementos de determinado paquete y el uso se utiliza cuando es necesario acceder a los paquetes sin necesidad de modificarlo. Por último se realizará el **diseño de la base de datos** el cual contiene el esquema de la base de datos de donde se generará el **modelo de datos**, con las tablas del sistema y sus relaciones.

1.6 Herramientas y lenguajes para el desarrollo de software

La selección de las tecnologías y herramientas correctas para el desarrollo, es un proceso de gran importancia en el que se deben tener en cuenta una serie de aspectos, fundamentalmente las características y los beneficios que aporta cada una de ser utilizadas. En la UCI, existen varios centros de desarrollo que están distribuidos por las distintas facultades. Cada centro tiene sus políticas, normas y estándares definidos para el desempeño de los distintos proyectos que lo componen. En el Centro CEIGE, específicamente en el Departamento de Soluciones

para la Aduana se desarrolla el producto GINA que constituye la nueva solución integrada para las operaciones aduanales en el cual se encuentra enmarcado el proceso de recepción electrónica de la información adelantada de los MTI. La selección de las herramientas mostradas a continuación, no forma parte de los objetivos de la investigación; pues fueron escogidas por el Departamento de Soluciones para la Aduana del CEIGE para el desarrollo de sus aplicaciones.

Lenguaje de Programación: PHP

PHP, acrónimo de "Hypertext Pre-Processor" se trata de un lenguaje interpretado. Resulta muy útil para diseñar de forma rápida y eficaz aplicaciones Web dinámicas dirigidas a bases de datos. Es multiplataforma (GNU/Linux, Windows, Mac); es libre, convirtiéndose en una alternativa de fácil acceso para todos; tiene capacidad de conexión con la mayoría de los gestores de bases de datos utilizados en la actualidad, siendo ésta una de sus características más potentes; también permite las técnicas de programación orientadas a objetos y tiene soporte para diferentes protocolos. (9)

PHP tiene asequible estructura de programación, la facilidad de llevar a cabo sentencias SQL, además de permitir la posibilidad de ejecutarse en diferentes tipos de servidores, entre ellos Apache. En el desarrollo de la herramienta se utiliza el lenguaje PHP en su versión 5.3

Framework de PHP: Symfony

Actualmente existen varios framework de *PHP* que facilitan el desarrollo de las aplicaciones web que utilizan este lenguaje, entre ellos se encuentra *Symfony*.

*Symfony es un completo framework desarrollado en PHP 5 y es compatible con la mayoría de los gestores de bases de datos existentes en el mundo, como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft. El mismo está diseñado para optimizar el desarrollo de las aplicaciones web y se puede ejecutar tanto en plataformas *nix (Unix, Linux, etc.) como en plataformas Windows. Separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. (10)* En el desarrollo de la solución se utiliza el Framework de PHP *Symfony* en su versión 1.2.8.

Lenguaje de Programación: JavaScript

Es un lenguaje de programación bastante sencillo y pensado para hacer las cosas con rapidez, se utiliza para crear pequeños programas encargados de realizar acciones dentro de una página Web. Es un lenguaje del lado del cliente porque es el navegador el que soporta la carga de procesamiento. Fundamentalmente se basa en la creación de efectos especiales en las páginas, además de definir interactividades con los usuarios. Es uno de los lenguajes más utilizados debido a su compatibilidad con la mayoría de los navegadores de la actualidad. En el desarrollo de la solución se utiliza JavaScript en su versión 1.4.

Framework de JavaScript a utilizar: Ext JS

Ext JS es un Framework de JavaScript que brinda la posibilidad de utilizar un gran número de componentes visuales que mejoran considerablemente la calidad de las aplicaciones. Permite realizar validaciones de formularios de todo tipo, basándose en expresiones regulares y tipos de datos. Trae implícitos componentes como vista en árboles, arrastrado y soltado, cambio de tamaño de imágenes, rejillas, paginado, agrupado de objetos, tabs, asistentes, entre otros. (11)

La utilización de un Framework de JavaScript como Ext JS facilita la separación de las capas de la vista con la del controlador desde el punto de vista productivo pues el código utilizado en la primera es solamente JavaScript y no es necesario utilizar ningún tipo de código PHP, así los desarrolladores pueden centrarse más en el aprendizaje de un solo lenguaje. (12)

Entorno de Desarrollo Integrado (IDE): NetBeans

Un Entorno de Desarrollo Integrado (en inglés Integrated Development Environment o IDE) es un programa compuesto por una serie de herramientas que utilizan los programadores para desarrollar código. Esta herramienta puede estar pensada para su utilización con un único lenguaje de programación o bien puede dar cabida a varios de estos. (13)

Actualmente los entornos de desarrollo proporcionan un marco de trabajo para la mayoría de los lenguajes de programación existentes en el mercado (por ejemplo C, C++, C#, Java, Python y PHP entre otros). Brindando la posibilidad de que un mismo entorno de desarrollo tenga la posibilidad de utilizar varios lenguajes de programación, como es el caso del NetBeans.

NetBeans IDE

NetBeans IDE es una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito en Java pero puede servir para cualquier otro lenguaje de programación. Existe además un número importante de módulos para extender el

NetBeans IDE, además es un producto libre y gratuito sin restricciones de uso. Posee un amplio soporte para el lenguaje *PHP* así como para los framework de trabajo *Symfony* y *Ext JS*. Para el desarrollo de la solución se ha elegido NetBeans IDE, en su versión 7.3, como Entorno de Desarrollo Integrado.

Sistema Gestor de Base de Datos: Oracle

Oracle (Relational Data Base Management System):

En el desarrollo de la solución se utiliza el sistema Gestor de Base de Datos Oracle, en su versión 11G. Es una potente herramienta cliente/servidor para la gestión de Bases de Datos donde se destaca su estabilidad, su escalabilidad, además de ser multiplataforma.

Entre sus principales características se encuentran las siguientes: (14)

- ✓ *Existe para la mayoría de las plataformas de hardware y software modernos garantizando la conectividad y portabilidad entre ellas y entre las distintas versiones de una misma plataforma.*
- ✓ *Ofrece un alto nivel de estabilidad que lo hace viable.*
- ✓ *Está concebido para un trabajo cliente/servidor. La posibilidad de definir procedimientos almacenados permite ejecutar procesos remotos limitando el intercambio a una solicitud de ejecución y envío de una respuesta.*
- ✓ *El acceso a los datos mediante un amplio conjunto de privilegios, roles y perfiles permite un control estricto de los usuarios*
- ✓ *Implementa prácticamente todo el espectro de controles de integridad e integridad referencial para los datos.*
- ✓ *Implementa fuertes mecanismos de resguardo tanto para fallos de instancia (terminación anormal de sesiones de trabajo, procesos, etc.) como para fallos de hardware.*
- ✓ *Ofrece capacidades completas de replicación sincrónica y asincrónica.*
- ✓ *Soporta un gran número de usuarios y cargas de trabajo de alto volumen de transacciones.*

1.7 Metodologías y herramientas de diseño

Metodologías de Diseño

Existen varias metodologías y principios de diseño. A continuación se enuncian algunas características de las metodologías y principios de diseños seleccionados para ser utilizados en el desarrollo de la herramienta. Teniendo en cuenta las características de la aplicación que se

desea desarrollar, se decidió usar en la presente investigación como metodologías de diseño: el Diseño Centrado en el Usuario, el Enfoque Ascendente y el Desarrollo Modular.

- ✓ **Diseño Centrado en el Usuario:** *Es la metodología de diseño más habitual en el mundo de desarrollo de software. Este enfoque coloca todas las necesidades, deseos y limitaciones del usuario como núcleo del proceso de diseño. Así mismo implica que el proyecto debe ser dividido por fases para garantizar los resultados. Dentro del diseño centrado en el usuario hay metodologías específicas, como son KES (Kansei Engineering System) o QFD (Quality Function Deployment). (15)*
- ✓ **Enfoque Ascendente:** *Consiste en partir de un elemento o de las características individuales de los elementos para desarrollar la totalidad del producto. (15)*
- ✓ **Desarrollo Modular:** *Su funcionamiento se basa en una descomposición de la programación en fracciones lógicas y manejables. Este tipo de programación se apega bien al diseño descendente porque enfatiza las interfaces entre los módulos, más que mantenerlas ignoradas hasta el final del desarrollo del sistema. Cada módulo debe ser funcionalmente cohesivo, de manera que satisfaga sólo una función. (15)*

Patrones de Diseño

Los patrones de diseño son soluciones simples y elegantes a problemas específicos y comunes del diseño orientado a objetos. Son soluciones basadas en la experiencia y que se ha demostrado que funcionan. Los patrones de diseño no son fáciles de entender, pero una vez entendido su funcionamiento, los diseños serán mucho más flexibles, modulares y reutilizables. Han revolucionado el diseño orientado a objetos y es muy importante su conocimiento por parte de todos los desarrolladores de software. (16)

Existen varios tipos de patrones de diseño, a continuación se muestran algunos de los empleados en el desarrollo de esta solución; que son implementados por el *framework* utilizado, el que se explicará en epígrafes posteriores:

General Responsibility Assignment Software Patterns (GRASP)

EXPERTO: La responsabilidad de realizar una labor es de la clase que tiene o puede tener los datos involucrados (atributos). Una clase contiene toda la información necesaria para realizar la labor que tiene encomendada. (17)

CREADOR: Este patrón, como su nombre lo indica, es el que crea, él guía la asignación de responsabilidades relacionadas con la creación de objetos, se asigna la responsabilidad de que una clase B cree un Objeto de la clase A solamente cuando:

- ✓ *B contiene A*
- ✓ *B es una agregación (o composición) de A*
- ✓ *B almacena A*
- ✓ *B tiene los datos de inicialización de A (datos que requiere su constructor)*
- ✓ *B usa A. (17)*

BAJO ACOPLAMIENTO: El acoplamiento es una medida de fuerza con que un elemento está a, tiene conocimiento de, confía en, otros elementos. Este patrón es un principio que asigna la responsabilidad de controlar el flujo de eventos del sistema, a clases específicas. Esto facilita la centralización de actividades (validaciones, seguridad, etc.). El controlador no realiza estas actividades, las delega en otras clases con las que mantiene un modelo de alta cohesión. (17)

ALTA COHESION: La cohesión es una medida de la fuerza con la que se relacionan las clases y el grado de focalización de las responsabilidades de un elemento. Cada elemento del diseño debe realizar una labor única dentro del sistema, no desempeñada por el resto de los elementos y auto-identificable. Una clase con baja cohesión hace muchas cosas no relacionadas o hace demasiado trabajo. (17)

CONTROLADOR: Es un evento generado por actores externos. Se asocian con operaciones del sistema como respuestas a los eventos del mismo, tal como se relacionan los mensajes y los métodos. Normalmente un controlador delega en otros objetos el trabajo que se necesita hacer; coordina o controla la actividad. No realiza mucho trabajo por sí mismo. (17)

DECORADOR: Responde a la necesidad de añadir dinámicamente funcionalidad a un objeto. Esto permite no tener que crear sucesivas clases que hereden de la primera incorporando la nueva funcionalidad, sino otras que la implementan y se asocian a la primera. Este patrón puede ser utilizado con varios fines como los que se muestran a continuación:

- ✓ *Añadir objetos individuales de forma dinámica y transparente.*
- ✓ *Responsabilidades de un objeto pueden ser retiradas.*
- ✓ *Cuando la extensión mediante la herencia no es viable.*
- ✓ *Hay una necesidad de extender la funcionalidad de una clase, pero no hay razones para extenderla a través de la herencia.*

- ✓ *Hay la necesidad de extender dinámicamente la funcionalidad de un objeto y quizás quitar la funcionalidad extendida. (17)*

Luego del estudio realizado acerca de los distintos patrones de diseño existentes se pudo comprender cuáles implementa el *framework Symfony*, de ellos se mencionaron anteriormente los que serán utilizados en el desarrollo del presente trabajo, más adelante se explicará cómo los implementa dicho *framework* teniendo en cuenta las particularidades de la presente solución.

Patrón Arquitectónico: Modelo Vista Controlador

Un patrón de arquitectura de software es un esquema genérico probado para solucionar un problema particular, el cual es recurrente dentro de un cierto contexto. Este esquema se especifica describiendo los componentes, con sus responsabilidades y relaciones. (18)

Modelo Vista Controlador (MVC): Es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. Symfony está basado en un patrón clásico del diseño web conocido como arquitectura MVC, donde la vista es la página HTML y el código que provee de datos dinámicos a la página, el controlador es el Sistema de Gestión de Base de Datos. Este patrón divide una aplicación interactiva en tres áreas: procesamiento, salida y entrada. Para esto, utiliza las siguientes abstracciones:

- ✓ *El modelo representa la información con la que trabaja la aplicación, es decir, su lógica de negocio. (10)*
- ✓ *La vista transforma el modelo en una página web que permite al usuario interactuar con ella. (10)*
- ✓ *El controlador se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista. (10)*

En la siguiente figura se muestra la implementación del patrón MVC en Symfony.

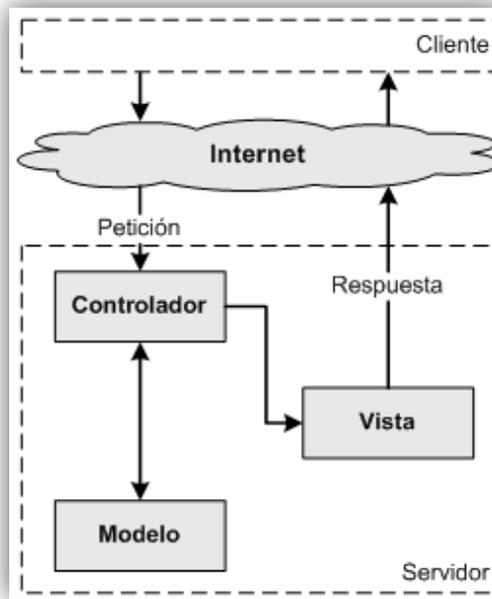


Fig2. Implementación del patrón MVC en Symfony

Herramienta de Diseño: Visual Paradigm

Visual Paradigm utiliza UML como lenguaje de modelado. Soporta el ciclo de vida completo del desarrollo de software. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Es una herramienta que puede ser utilizada en la creación de software libre y es disponible en múltiples plataformas.

Permite realizar ingeniería tanto directa como inversa, ya que a partir de un modelo relacional en Sql Server, MySql, etc., es capaz de desplegar todas las clases asociadas a las tablas. Además, la herramienta es colaborativa, es decir, soporta múltiples usuarios trabajando sobre el mismo proyecto; genera la documentación del proyecto automáticamente en varios formatos como Web o Pdf, y permite control de versiones. (19)

1.8 Métricas

El concepto de métrica es el término que describe muchos y muy variados casos de medición. Siendo una métrica una medida estadística que se aplica a todos los aspectos de calidad de software, los cuales deben ser medidos desde diferentes puntos de vista como el análisis, construcción, funcionalidad, documentación, métodos, proceso, usuario, entre otros. (20)

Métricas para Requisitos

Los requisitos del software son la base de las medidas de la calidad. (21) En la disciplina de Requisitos se analiza el comportamiento de las métricas Estabilidad de Requisitos, Especificidad de Requisitos y Grado de Validación de los Requisitos.

✓ **Estabilidad de los requisitos**

El objetivo de esta métrica es medir la estabilidad de los requisitos para asegurar su adecuación antes de pasar al próximo flujo de trabajo. Se considera que los requisitos son estables cuando no existen adiciones o supresiones en ellos que impliquen modificaciones en las funcionalidades principales de la aplicación. (22)

La estabilidad de los requisitos se calcula como:

$$ETR = \left[\frac{RT - RM}{RT} \right] * 100$$

Donde:

ETR: valor de la estabilidad de los requisitos.

RT: total de requisitos definidos.

RM: número de requisitos modificados, que se obtienen como la sumatoria de los requisitos insertados, modificados y eliminados.

Los valores ofrecidos por esta métrica se encuentran entre 0 y 100. El mejor valor de ETR es el más cercano a 100 porque mostrará que no se están realizando cambios sobre los requisitos, son estables y, por tanto, es confiable trabajar el análisis y diseño sobre ellos.

✓ **Especificidad de los requisitos**

La comprensibilidad de los requisitos depende en gran medida de la ausencia de ambigüedades en su especificación, facilitando el procesamiento de los mismos. El objetivo de esta métrica es cuantificar la especificidad o falta de ambigüedad en la definición de los requisitos.

La especificidad de los requisitos se calcula mediante la fórmula siguiente:

$$Q_1 = \frac{n_{ui}}{n_r}$$

Donde:

Q₁: Especificidad de los requisitos.

n_r: Total de requisitos definidos.

n_{ui}: Total de requisitos para los que los revisores tuvieron interpretaciones idénticas.

Cuanto más cerca de 1 esté el valor de Q_1 menor será la ambigüedad de la especificidad de los requisitos.

✓ **Grado de validación de los requisitos**

Esta técnica se emplea para medir si el proceso de validación de los requisitos identificados fue correcto. El grado de validación de los requisitos mide la corrección en la definición de los requisitos.

El grado de validación de los requisitos se calcula mediante la siguiente fórmula:

$$Q_3 = \frac{n_c}{(n_c + n_{nv})}$$

Donde:

Q₃: Grado de validación de requisitos.

n_c: Total de requisitos validados correctamente.

n_{nv}: Total de requisitos no validados.

El resultado de esta métrica está siempre entre 0 y 1. El valor óptimo, es el más cercano a 1 e indica un alto nivel de corrección en la definición de los requerimientos.

Métricas para la validación del diseño

Las métricas para software como otras métricas no son perfectas, muchos expertos argumentan que se necesita más experimentación hasta que se puedan emplear bien las métricas de diseño, sin embargo el diseño sin medición es una alternativa inaceptable. A continuación se mostrarán algunas de las más comunes. Aunque ninguna es perfecta, pueden proporcionarle al diseñador una mejor visión interna y así el diseño evolucionará a un mejor nivel de calidad.

Existen numerosas métricas para la medición del diseño desarrollado, dentro de las que se encuentran: (23)

- ✓ Tamaño operacional de clase (TOC): métrica que se basa esencialmente en la cantidad de funcionalidades que presenta la clase.
- ✓ Profundidad de herencia (PH): mide el máximo nivel en la jerarquía de herencia. Es la cuenta directa de los niveles en la jerarquía de herencia.
- ✓ Número de descendientes (ND): es el número de subclases subordinadas a una clase en la jerarquía, es decir, el número de subclases que pertenecen a una clase.

- ✓ Número de operaciones redefinidas para una clase hija (NOR): muestra en qué medida las subclases redefinen el comportamiento de sus superclases.
- ✓ Relaciones entre clases (RC): La métrica RC está dada por el número de relaciones de uso de una clase con otras.

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), propuesta por Lorenz y Kid y la de Relaciones entre clases (RC). La métrica TOC se encarga de medir la calidad de acuerdo a los atributos Responsabilidad, Complejidad de Implementación y Reutilización de las clases, mientras que la RC tiene en cuenta los atributos Acoplamiento, Complejidad del Mantenimiento y Cantidad de Pruebas.

1.9 Conclusiones parciales

A partir de la definición del estado actual del proceso de gestión de la información adelantada establecida por la OMI del sistema Despacho de Medios de Transporte Internacional se obtuvo una mejor comprensión del funcionamiento de este proceso. Además se concluye que:

- ✓ Serán utilizadas las tecnologías, herramientas y lenguajes de programación definidas por el Departamento de Soluciones para la Aduana para el desarrollo del sistema GINA puesto que con estas, se podrá realizar la implementación de la herramienta sin dificultades.
- ✓ Se utilizará el Modelo de desarrollo de software del CEIGE como guía para la especificación de las actividades a realizar en cada momento.
- ✓ Se decide utilizar las métricas Estabilidad de Requisitos, Especificidad de Requisitos y Grado de Validación de los Requisitos para validar los requisitos. Por otra parte, para la evaluación del diseño de la solución se utilizarán las métrica de Tamaño Operacional de Clase (TOC) y Relaciones entre Clases (RC).

CAPÍTULO 2: PROPUESTA DE SOLUCIÓN

2.1 Introducción

En este capítulo se describen las características del sistema, se desarrolla el diseño y la implementación que dará solución al problema planteado en el presente trabajo. Durante el diseño se obtendrán los distintos artefactos que guiarán la implementación del sistema. Entre los artefactos que serán generados se encuentran: el modelo conceptual, diagrama de procesos de negocio, diagrama de clases del sistema, diagrama de paquetes, modelo de datos y el diagrama de componentes. Durante la implementación se obtendrá el código fuente de algunas clases, así como también, se muestran algunas pantallas de la aplicación y estándares de codificación utilizados.

2.2 Modelo Conceptual

Para descomponer el dominio del problema hay que identificar los conceptos, los atributos y las asociaciones del dominio que se juzgan importantes. El resultado puede expresarse en un modelo conceptual, el cual se muestra gráficamente en un grupo de diagramas que describen los conceptos. (17)

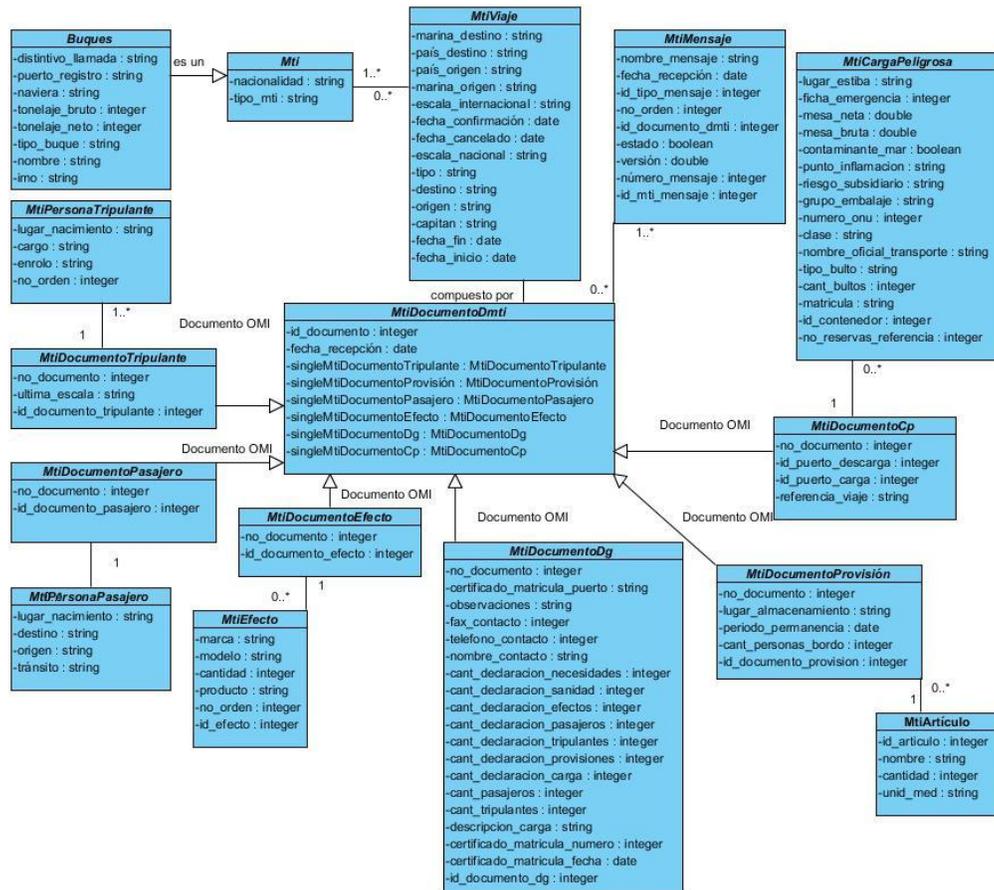


Fig3. Modelo Conceptual

Los principales conceptos y atributos están reflejados en el modelo conceptual descritos en el diccionario de datos que se encuentra en el Anexo 6. A continuación se muestra la descripción de dos conceptos fundamentales, uno de ellos es MTI, el cual representa los medios de transporte internacional; y Buques que especifica el MTI Buque.

MTI

Descripción	Describe los datos de los Medios de Transporte Internacional, encargados del traslado de mercancías y/o personas de un país a otro y que son sometidos a controles aduaneros.					
Atributos						
Nombre	Descripción	Tipo	¿Puede ser nulo?	¿Es único?	Restricciones	
					Clases válidas	Clases no válidas
Tipo_MTI	Nombre del MTI	String	No	No	Caracteres alfabéticos [A..Z, a..z]	Caracteres numéricos y caracteres especiales
Nacionalidad	Nacionalidad que representa el MTI	String	No	No	Caracteres alfabéticos [A..Z, a..z]	Caracteres numéricos y caracteres especiales

Tabla1. Descripción del concepto MTI

BUQUES

Descripción	Describe los datos del tipo de MTI dedicado al comercio o transporte de personas por la vía marítima.					
Atributos						
Nombre	Descripción	Tipo	¿Puede ser nulo?	¿Es único?	Restricciones	
					Clases válidas	Clases no válidas
IMO	Número identificativo del buque	String	No	Sí	Caracteres alfanuméricos [A..Z, a..z, 0..9] Ejemplo: 2W344R67	Caracteres especiales.
Nombre	Indica el nombre del buque.	String	No	No	Caracteres alfanuméricos [A..Z, a..z, 0..9] y los caracteres [(-) y (.)]	Caracteres especiales diferentes de (-) y (.)
Tipo_Buque	Número	String	No	Si	Ferrys	Caracteres

	utilizado por los buques para las comunicaciones				Remorcador Patana Cruceiros Otros	especiales caracteres numéricos.	y
Tonelaje bruto	Describe el peso bruto del buque	Integer	No	No	Caracteres numéricos [0..9]	Caracteres especiales caracteres alfabéticos.	y
Tonelaje neto	Describe el peso neto del buque	Integer	No	No	Caracteres numéricos [0..9]	Caracteres especiales caracteres alfabéticos.	y
Naviera	Describe la naviera propietaria del buque.	String	No	No	Caracteres alfabéticos [A..Z, a..z]	Caracteres especiales caracteres numéricos.	y
Puerto_registro	Indica el nombre del puerto donde se registró al buque.	String	No	Si	Caracteres alfabéticos [A..Z, a..z]	Caracteres especiales caracteres numéricos.	y
Distintivo_llamada	Distintivo que identifica al buque	String	No	Si	Caracteres alfabéticos [A..Z, a..z]	Caracteres especiales caracteres numéricos.	y

Tabla2. Descripción del concepto Buques

2.3 Diagrama del proceso de negocio

“El diagrama de procesos de negocio está conformado por un conjunto de actividades que interactúan entre sí y generan valor para la organización. Son una representación abstracta-gráfica de los procesos de una organización, estos muestran los actores involucrados en el proceso, así como las actividades que estos realizan en secuencia y elementos que se generan”.(30)

A continuación se muestra el diagrama del proceso de negocio, el cual ilustra detalladamente como se lleva a cabo el proceso de recepción electrónica de los documentos OMI.

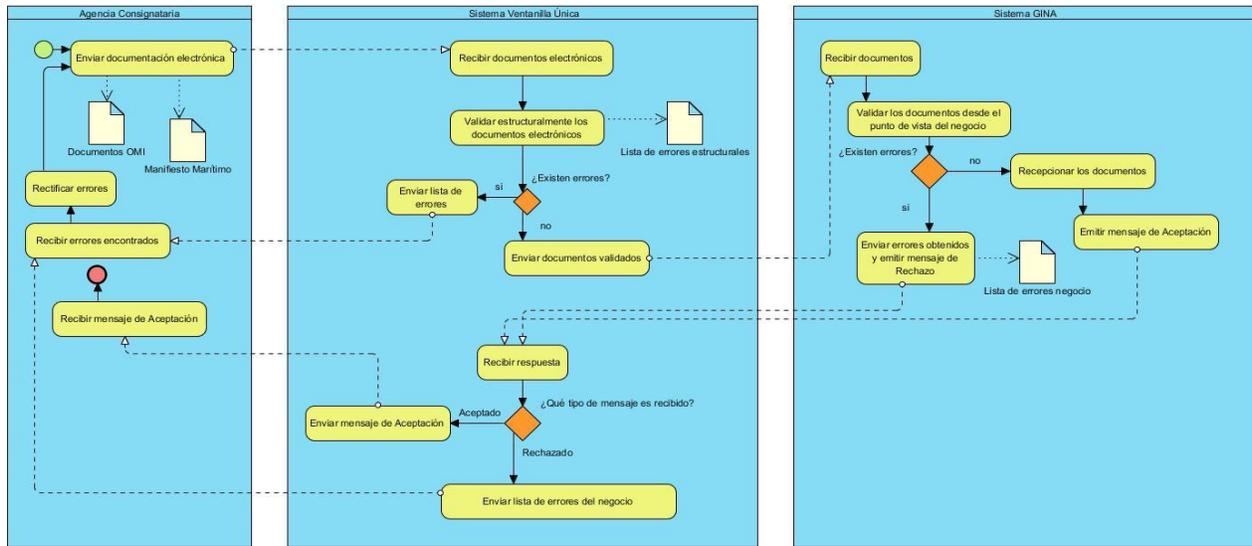


Fig4. Diagrama de Procesos de Negocios: Recepción electrónica de documentos OMI

2.4 Descripción de la solución

De manera general el sistema GINA integra las aplicaciones que informatizan los procesos de la AGR. En trabajos previos, se identificaron los principales problemas que convergen en los procesos aduanales. En este caso, se puede mencionar la creación de una herramienta que permita gestionar el proceso de recepción de la información adelantada establecida por la Organización Marítima Internacional, facilitando a los especialistas del área de lucha contra el fraude u otra área de la AGR mayor accesibilidad y organización cuando se reciba la misma. Como resultado final la herramienta propuesta debe permitir la realización de este proceso de forma ágil y eficaz, garantizando la disminución del tiempo de consulta de la información adelantada de cargas, pasajeros y tripulantes establecida por la OMI que recibe la Aduana General de la República por vía electrónica.

2.5 Requisitos del Sistema

Para la obtención de los requisitos funcionales se hizo necesario emplear la técnica de entrevistas para poder identificarlos. Luego de haber sido realizadas varias entrevistas con el usuario, se identificaron los siguientes requisitos funcionales del software.

Requisitos funcionales:

Requisitos	Descripción
Procesar información adelantada de la Declaración de Efectos de la Tripulación.	Recibida la información adelantada de los documentos enviada por el sistema

Procesar información adelantada de la Declaración General.	Ventanilla Única se procede a validar los datos de los mismos. En caso de existir errores, se emite un mensaje de error y se incluye en el listado de errores detectados. Si no existen errores, se registra la información adelantada del documento correspondiente y se notifica a Ventanilla Única la respuesta positiva del registro.
Procesar información adelantada de las Provisiones del Buque.	
Procesar información adelantada de la Lista de Pasajeros.	
Procesar información adelantada de la Lista de Tripulantes.	
Procesar información adelantada del Manifiesto de Mercancías Peligrosas.	

Tabla3. Requisitos Funcionales

La descripción detallada de cada uno de los requisitos se encuentra en el anexo 7 del presente documento. A continuación se muestra la descripción del requisito **Procesar información adelantada de la Declaración General**.

Precondiciones	Recibida la información adelantada de la Declaración General enviada por el sistema Ventanilla Única.
Flujo de eventos	
Flujo básico Procesar información adelantada de la Declaración General.	
No	Sistema
	Valida la Composición del Nombre del Fichero OMI-G : OMI + '-' + '1/2' + Código Puerto Cuba (3C) + Año (2C) + Escala (4C) + '-' + '1...n (1C)' + G.
1	En caso de existir errores en el nombre del fichero se rechaza el fichero y se emite un mensaje de error. Se incluye en el listado de errores detectados (Nombre del Fichero no válido).
	Valida el Identificador del Tipo de Documento .
2	En caso de que no esté en el formato establecido, incluir en el listado de errores detectados (Identificador del Tipo de Documento no válido).
	Valida el Número del Mensaje XML .
3	En caso de que no se corresponda con el nombre del fichero, incluir en el listado de errores detectados (Número del Mensaje XML no válido).
	Valida la Versión del software emisor del mensaje.
4	En caso de que no esté en el formato establecido, incluir en el listado de errores detectados (Versión no válida).
	Valida Llegada/Salida .
5	En caso de que no esté en el formato establecido, incluir en el listado de errores

	detectados (Llegada/Salida no válida).
	Valida el Nombre del Buque .
6	En caso de que no esté en el formato establecido, incluir en el listado de errores detectados (Nombre del Buque no válido).
	Valida el Tipo de Buque según el tipo de carga.
7	En caso de que el tipo no se corresponda con alguno de los nombrados en las Tablas de Control, incluir en el listado de errores detectados (Tipo de Buque no válido).
	Valida el Número IMO del buque.
8	En caso de que no esté en el formato establecido, incluir en el listado de errores detectados (Número IMO no válido).
	Valida el formato del Distintivo de Llamada (9 caracteres).
9	En caso de existir errores incluir en el listado de errores detectados (Distintivo de llamada no válido).
	Valida la Escala del viaje.
10	En caso de que no esté en el formato establecido, incluir en el listado de errores detectados (Escala no válida).
	Valida el Número de Orden del mensaje.
11	En caso de existir errores incluir en el listado de errores detectados (Número de Orden no válido).
	Valida el Puerto de Llegada/Salida (Puertos Cuba).
12	En caso de que el puerto no se corresponda con uno de los nombrados en las Tablas de Control, incluir en el listado de errores detectados (Puerto de Llegada/Salida no válido).
	Valida la Fecha de Llegada/Salida .
13	En caso de que la fecha no esté en el formato establecido, incluir en el listado de errores detectados (Fecha de Llegada/Salida no válida).
	Valida la Hora de Llegada/Salida .
14	En caso de que la hora no esté en el formato establecido, incluir en el listado de errores detectados (Hora de Llegada/Salida no válida).
	Valida el Estado de Abanderamiento del buque.
15	En caso de que no se corresponda con uno de los nombrados en las Tablas de Control, incluir en el listado de errores detectados (Estado de Abanderamiento no válido).
	Valida el Nombre del capitán .
16	En caso de que no esté en el formato establecido, incluir en el listado de errores detectados (Nombre del capitán no válido).
17	Valida el Último/Próximo Puerto de Escala .

	En caso de que el puerto no se corresponda con uno de los nombrados en las Tablas de Control, incluir en el listado de errores detectados (Último/Próximo Puerto de Escala no válido).
	Valida el Certificado de Matrícula (Puerto) .
18	En caso de que el puerto no se corresponda con alguno nombrado en las Tablas de Control, incluir en el listado de errores detectados (Puerto no válido).
	Valida el Certificado de Matrícula (Fecha) .
19	En caso de que la fecha no esté en el formato establecido, incluir en el listado de errores detectados (Fecha no válida).
	Valida el Certificado de Matrícula (Número) .
20	En caso de que el número no esté en el formato establecido, incluir en el listado de errores detectados (Número no válido).
	Valida los Nombre del Contacto .
21	En caso de que los datos no estén en el formato establecido, incluir en el listado de errores detectados (Nombre del Contacto no válido).
	Valida el Número de Teléfono del contacto del agente del buque.
22	En caso de que el número no esté en el formato establecido, incluir en el listado de errores detectados (Número de Teléfono no válido).
	Valida el Fax del contacto del agente del buque.
23	En caso de que el Fax no esté en el formato establecido, incluir en el listado de errores detectados (Fax no válido).
	Valida la Dirección Email del contacto del agente del buque:
24	En caso de que el email no esté en el formato establecido, incluir en el listado de errores detectados (Dirección Email no válida).
	Valida el Arqueo Bruto (Toneladas Brutas del Registro del Buque):
25	En caso de que el arqueo bruto no esté en Kg, incluir en el listado de errores detectados (Arqueo Bruto no válido).
	Valida el Arqueo Neto (Toneladas Netas del Registro del Buque):
26	En caso de que el arqueo neto no esté en Kg, incluir en el listado de errores detectados (Arqueo Neto no válido).
	Valida la Situación del Buque en el puerto (Muelle o puesto de atraque, 4 caracteres):
27	En caso de que el muelle o puesto de atraque no se corresponda con alguno nombrado en las Tablas de Control, incluir en el listado de errores detectados (Situación del Buque

	no válida).
	Valida la Descripción Somera de la Carga .
28	En caso de que no esté en el formato establecido, incluir en el listado de errores detectados (Descripción no válida).
	Valida el Número de Tripulantes :
29	En caso de que la cantidad sea cero, incluir en el listado de errores detectados (Número de Tripulantes no válido).
	Valida el Número de Pasajeros en caso de que el buque los transporte. Si el valor es cero (0), debe estar la cadena de texto : No
30	En caso de que la cantidad sea cero, incluir en el listado de errores detectados (Número de Pasajeros no válido).
	Valida las Observaciones (200 caracteres).
31	En caso de que no esté en el formato establecido, incluir en el listado de errores detectados (Observaciones no válidas).
	Valida la Cantidad de Ejemplares de la Declaración General . Si el valor es cero (0), debe estar la cadena de texto : No
32	En caso de que la cantidad sea cero, incluir en el listado de errores detectados (Cantidad de Ejemplares de la DG no válida).
	Valida la Cantidad de Ejemplares de la Declaración de Provisiones del Buque . Si el valor es cero (0), debe estar la cadena de texto : No
33	En caso de que la cantidad sea cero, incluir en el listado de errores detectados (Cantidad de Ejemplares de la DPB no válida).
	Valida la Cantidad de Ejemplares de la Lista de Tripulantes . Si el valor es cero (0), debe estar la cadena de texto : No
34	En caso de que la cantidad sea cero, incluir en el listado de errores detectados (Cantidad de Ejemplares de la LT no válida).
	Valida la Cantidad de Ejemplares de la Lista de Pasajeros . Si el valor es cero (0), debe estar la cadena de texto : No
35	En caso de que la cantidad sea cero, incluir en el listado de errores detectados (Cantidad de Ejemplares de la LP no válida).
	Valida la Cantidad de Ejemplares de la Declaración Marítima de Sanidad . Si el valor es cero (0), debe estar la cadena de texto : No
36	En caso de que la cantidad sea cero, incluir en el listado de errores detectados (Cantidad de Ejemplares de la DMS no válida).
	Valida los Pormenores Someros referentes al viaje (puertos donde el buque ha tocado y va a tocar, subrayar puertos en los cuales la mercancía que permanece a bordo será descargada).
37	En caso de que los puertos no se correspondan con los nombrados en las Tablas de

	Control, incluir en el listado de errores detectados (Puertos no válidos).
	Valida los Pormenores Someros referentes al viaje (Descarga Mercancías : Si o No).
38	En caso de que no esté en el formato establecido, incluir en el listado de errores detectados (Descarga no válida).
	Valida las Observaciones de los pormenores, (100 caracteres).
39	En caso de que no esté en el formato establecido, incluir en el listado de errores detectados (Observaciones no válidas).
	Valida el listado de errores:
40	En caso de que se haya detectado alguno, ejecuta el <i>Flujo Alternativo 26.a Datos incorrectos</i> .
41	Envía respuesta positiva del registro a Ventanilla Única.
42	Concluye el requisito.
Pos-condiciones	
1	Se registra la información adelantada de la Declaración General.
2	Se notifica a Ventanilla Única la respuesta positiva del registro.
Flujos alternativos	
Flujo alternativo 40 a Datos incorrectos.	
No	Sistema
1	Registra los errores detectados.
2	Envía respuesta negativa del registro a Ventanilla Única, notificándole el listado de errores detectados.
3	Concluye el requisito.
Pos-condiciones	
1	Se notifica a Ventanilla Única la respuesta negativa del registro.
2	
Validaciones	
1	Se validan los datos según lo establecido en las entidades del Modelo Conceptual (1)
2	Valida que el código del puerto cubano del nombre del fichero del mensaje tenga 3 caracteres y esté nombrado.
3	El identificador del tipo de documento de la Declaración General debe ser OMI-G.
4	Si los OMI enviados están asociados a un viaje de entrada el 1 del número de mensaje correspondiente a la entrada tiene que coincidir con la Llegada (carácter E) y si están asociados a un viaje de salida el 2 del número de mensaje correspondiente a la salida tiene que coincidir con la Salida (carácter S).
5	El tipo del buque es un código de 3 caracteres que tiene que existir y estar nombrado.
6	El número de escala debe tener 4 caracteres numéricos + '/' + Año (4C).
7	El número de orden del mensaje es un consecutivo.
8	El puerto de llegada es un código de 3 caracteres que tiene que existir y estar nombrado.
9	La fecha debe estar en el formato establecido (AAAA-MM-DD) y no puede ser posterior a la fecha actual.
10	La hora debe estar en el formato establecido (99:99).
11	El último puerto de escala o próximo puerto de escala tiene que ser un código de 5 caracteres y estar nombrado.

Conceptos

Relaciones	Requisitos Incluidos	No aplica.
	Requisitos Excluidos	No aplica.
Requisitos especiales	Son los requisitos no funcionales específicos para el requisito.	
Asuntos pendientes	Posibles mejoras al requisito.	

Tabla4. Descripción del requisito Procesar información adelantada de la Declaración General

Requisitos no funcionales:

Entre los requerimientos no funcionales de la herramienta propuesta se encuentran:

✓ **Usabilidad:**

RnF_1 El producto constituirá una herramienta enfocada en las necesidades de la Aduana General de la República.

RNF_2 Todos los mensajes de error de la herramienta deben de tener una descripción textual del error.

✓ **Eficiencia:**

RnF_3 La herramienta debe ser capaz de dar respuestas a las peticiones con un nivel aceptable de desempeño.

✓ **Confiabilidad**

RNF_4 La herramienta debe validar automáticamente la información contenida en los documentos. En el proceso de validación de la información, se deben tener en cuenta aspectos tales como obligatoriedad de campos, longitud de caracteres permitida por campo, manejo de tipos de datos, entre otros.

✓ **Seguridad:**

RNF_5 Autenticación. El sistema concederá acceso a cada usuario autenticado solo a las funciones que le estén permitidas, de acuerdo a la configuración del sistema.

RNF_6 Autorización. Los permisos serán limitados según los roles definido en el sistema, el usuario no debe poder modificar sus permisos. Los permisos de los usuarios deben estar en correspondencia con los niveles de acceso a la información.

✓ **Portabilidad**

RnF_7 Para su Implementación se usaron Herramientas de Programación que son multiplataforma garantizando que pueda ser utilizado en diferentes sistemas operativos como Windows, Linux.

✓ **Mantenibilidad:**

RNF_8 La herramienta debe estar en capacidad de permitir en el futuro su fácil mantenimiento con respecto a los posibles errores que se puedan presentar durante la operación del sistema.

✓ **Escalabilidad:**

RNF_9 La herramienta debe estar en capacidad de permitir en el futuro el desarrollo de nuevas funcionalidades, modificar o eliminar funcionalidades después de su construcción y puesta en marcha inicial.

2.6 Validación de Requisitos

La validación de requisitos se realiza con el objetivo de asegurar que todos los requisitos del sistema han sido establecidos sin ambigüedad, sin inconsistencias, sin omisiones, que los errores detectados hayan sido corregidos, y que el resultado del trabajo se ajusta a los estándares establecidos para el proceso, el proyecto y el producto. (24)

Existen un conjunto de técnicas que se utilizan para llevar a cabo la validación de los requisitos. Entre ellas se encuentra la **revisión técnica formal**, que consiste en la lectura y corrección de la documentación o modelado de la definición de requisitos para validar la correcta interpretación de la información transmitida. Se realiza la validación de los requisitos de la herramienta utilizando esta técnica, donde fueron revisados y aceptados por parte de la dirección del proyecto, obteniendo gran aceptación por parte de los clientes. En el anexo 8 se refleja el acta de aceptación asociada a los requisitos funcionales de la herramienta.

Métricas para Requisitos

Los requisitos del software son la base de las medidas de la calidad. En la disciplina de Requisitos se tuvo en cuenta la métrica para medir su estabilidad.

Aplicación de la métrica Estabilidad de requisitos:

Teniendo en cuenta que se identificaron un total de 6 requisitos funcionales, se calcula:

$$ETR = \left[\frac{RT - RM}{RT} \right] * 100$$

$$\text{ETR} = [(6-0) / 6] * 100 = 100$$

Como resultado se obtuvo un valor de 100. Dicha cifra demuestra que no se han realizado cambios sobre los requisitos, son estables y, por tanto, es confiable el análisis y diseño sobre ellos.

Aplicación de la métrica Especificidad de los requisitos:

Se obtiene como resultado:

$$Q_1 = \frac{n_{ui}}{n_r}$$

$$Q_1 = 6/6 = 1$$

Como el valor de la Q1 es 1, se puede inferir que la especificación de los requisitos no posee ambigüedad. Esta característica asegura mayor calidad en el proceso de especificación.

Aplicación de la métrica Grado de validación de los requisitos:

Se obtiene como resultado:

$$Q_3 = \frac{n_c}{(n_c + n_{nv})}$$

$$Q_3 = 6/(6+0) = 1$$

La aplicación de esta técnica es 1, se concluye que la definición de los requisitos es correcta.

Dichas métricas demuestran que al no haber realizado cambios en los requisitos, estos son estables y es confiable el análisis y diseño sobre ellos; también se puede inferir que la especificación de los mismo no poseen ambigüedad y están correctamente definidos.

2.7 Patrones de diseño implementados

Para la implementación con el *framework* *Symfony* se utilizan varios patrones, situándolos en las capas del modelo y el controlador. Seguidamente se describen dichos patrones.

Patrones GRASP

- ✓ **Alta cohesión:** el *framework* *Symfony* permite la organización del trabajo en cuanto a la estructura del proyecto y la asignación de responsabilidades con una alta cohesión. Esta caracteriza a las clases con responsabilidades estrechamente relacionadas que no

realicen un trabajo enorme. Se evidencia que las clases son reutilizables mediante la agrupación de funcionalidades. La clase *Actions*¹ (*buquesActions*), es formada por varias funcionalidades estrechamente relacionadas, siendo la responsable de definir las acciones para las plantillas y colaborar con otras para realizar diferentes operaciones e instanciar objetos.

- ✓ **Bajo acoplamiento:** asigna la responsabilidad de mantener el control sobre el flujo de eventos del sistema, a clases específicas. Es muy utilizado para mantener organizadas todas las funcionalidades, posibilitando agrupar los procedimientos semejantes, para hacer menos engorroso el proceso de validación y la seguridad. El acoplamiento es una medida de la fuerza con que una clase está conectada a otras clases, con que las conoce y recurre a ellas. La clase *buquesActions* hereda de *sfActions* únicamente para alcanzar un bajo acoplamiento. Las clases que implementan la lógica del negocio y de acceso a datos se encuentran en el modelo, las mismas no tienen asociaciones con las de la vista o el controlador, lo que proporciona que la dependencia en este caso sea baja.
- ✓ **Creador:** guía la asignación de responsabilidades relacionadas con la creación de objetos, tarea muy frecuente en los sistemas orientados a objetos. El propósito fundamental es encontrar un creador que se conecte con el objeto producido en cualquier evento. En la clase *buquesActions* se encuentran las acciones definidas para el sistema y se ejecutan en cada una de ellas. En dichas acciones se crean los objetos de las clases que representan las entidades, lo que evidencia que la clase *buquesActions* es “creador” de dichas entidades.
- ✓ **Experto:** la responsabilidad de realizar una labor es de la clase que tiene o puede tener los atributos. Una clase, contiene toda la información necesaria para realizar la labor que tiene encomendada. Es uno de los patrones más utilizados en el *framework Symfony*. La librería *Prope*², para mapear la Base de Datos y para realizar su capa de abstracción en el modelo, encapsula toda la lógica de los datos y genera las clases con todas las

¹ Es la clase controladora utilizada en el *framework Symfony* para realizar las funcionalidades del sistema.

² *Kit* de mapeo objeto-relacional (ORM) de código abierto escrito en PHP. Es además una parte integral del *framework Symfony*.

funcionalidades comunes de las entidades. Las clases de abstracción de datos poseen un grupo de funcionalidades que están relacionadas directamente con la entidad que representan y contienen la información necesaria de la tabla que representan.

- ✓ **Controlador:** evento generado por actores externos. Se asocian con operaciones del sistema, como respuestas a los eventos del sistema, tal como se relacionan los mensajes y los métodos. Normalmente un controlador delega en otros objetos el trabajo que se necesita hacer, coordina o controla la actividad. Se emplea el controlador “*servicios.php*” que se encarga de procesar todas las peticiones que se realizan a través del mismo para direccionarla al resto de las plantillas. Además todas las peticiones *web* son manipuladas por un solo controlador frontal (*sfActions*), que es el punto de entrada único de toda la aplicación en un entorno determinado. (25)

2.8 Modelo de diseño

En el diseño se modela el sistema y se estructura de acuerdo a la arquitectura definida para que soporte todos los requisitos funcionales y no funcionales. Para poder dar el paso al diseño es necesario previamente realizar un análisis del sistema que se desea obtener, o sea, el modelo de análisis, que proporciona una comprensión detallada de los requisitos.

2.8.1 Diagrama de clases del diseño

En las **figuras 5 y 6** se muestra el diagrama de clases planteado para dar solución al problema. En el mismo se recogen todas las clases, sus relaciones y atributos, involucrados para el desarrollo de la herramienta propuesta. Las clases **MtiDocumentoPasajero**, **MtiDocumentoTripulante**, **MtiDocumentoEfecto**, **MtiDocumentoDg**, **MtiDocumentoCp** y **MtiDocumentoProvision**, contienen los documentos OMI que deben ser entregados de forma anticipada por los buques. Las clases **TcTipoMensaje**, **TcTipoBuque** y **TcDocumentoMti** son entidades nomencladoras y las clases **MtiBuque** y **MtiViaje** son las entidades que contienen el medio de transporte internacional Buque y los viajes efectuados.



Fig6. Parte 2 del Diagrama de Clases del Diseño

2.8.2 Diagrama de paquetes

Los diagramas de paquetes se usan para reflejar la organización de paquetes y sus elementos. Es un mecanismo de agrupamiento que es altamente cohesivo internamente y con las interacciones mínimas con otros paquetes. El diagrama de paquetes que se representa en la **figura 7**, contiene la interacción entre los paquetes y librerías de la herramienta. Se muestra la interacción con el núcleo de Symfony, la base de Datos y el marco de trabajo Propel, utilizado para la generación de las clases del modelo y para la realización de consultas sin la necesidad

de crear ni una única sentencia SQL. Dentro de la misma herramienta se materializa el vínculo con diferentes paquetes, pertenecientes a las diferentes capas que define el patrón arquitectónico MVC. Dentro de los paquetes de mayor importancia se encuentra el Controlador que pertenece a la capa Controladora, presenta las clases y objetos para el control de las peticiones y el flujo de acciones a realizar y el lib que es la capa del Modelo, donde se encuentran las clases del negocio, los formularios y algunas de las encargadas del mapeo y abstracción de la base de datos. La forma de comunicación entre estos componentes se llevará a cabo utilizando servicios que ofrecerán los mismos, de manera que se garantice que no existan violaciones en la seguridad de ninguno de los relacionados en la figura.

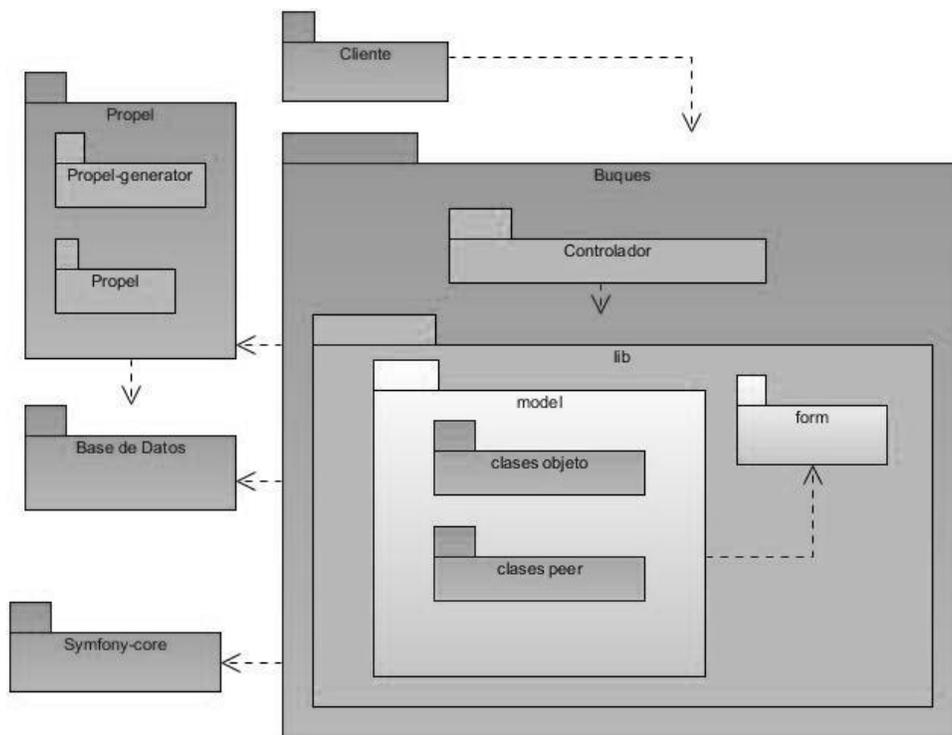


Fig7. Diagrama de Paquetes

2.8.3 Diseño de la Base de Datos

El Diagrama de Entidad-Relación permite que un ingeniero del software especifique los objetos de datos que entran y salen de un sistema, los atributos que definen las propiedades de estos objetos y las relaciones entre los objetos. (24) A continuación se muestra el Diagrama Entidad-Relación correspondiente a la herramienta, sus 24 tablas, las cuales tienen implícitas sus atributos y relaciones entre ellas.

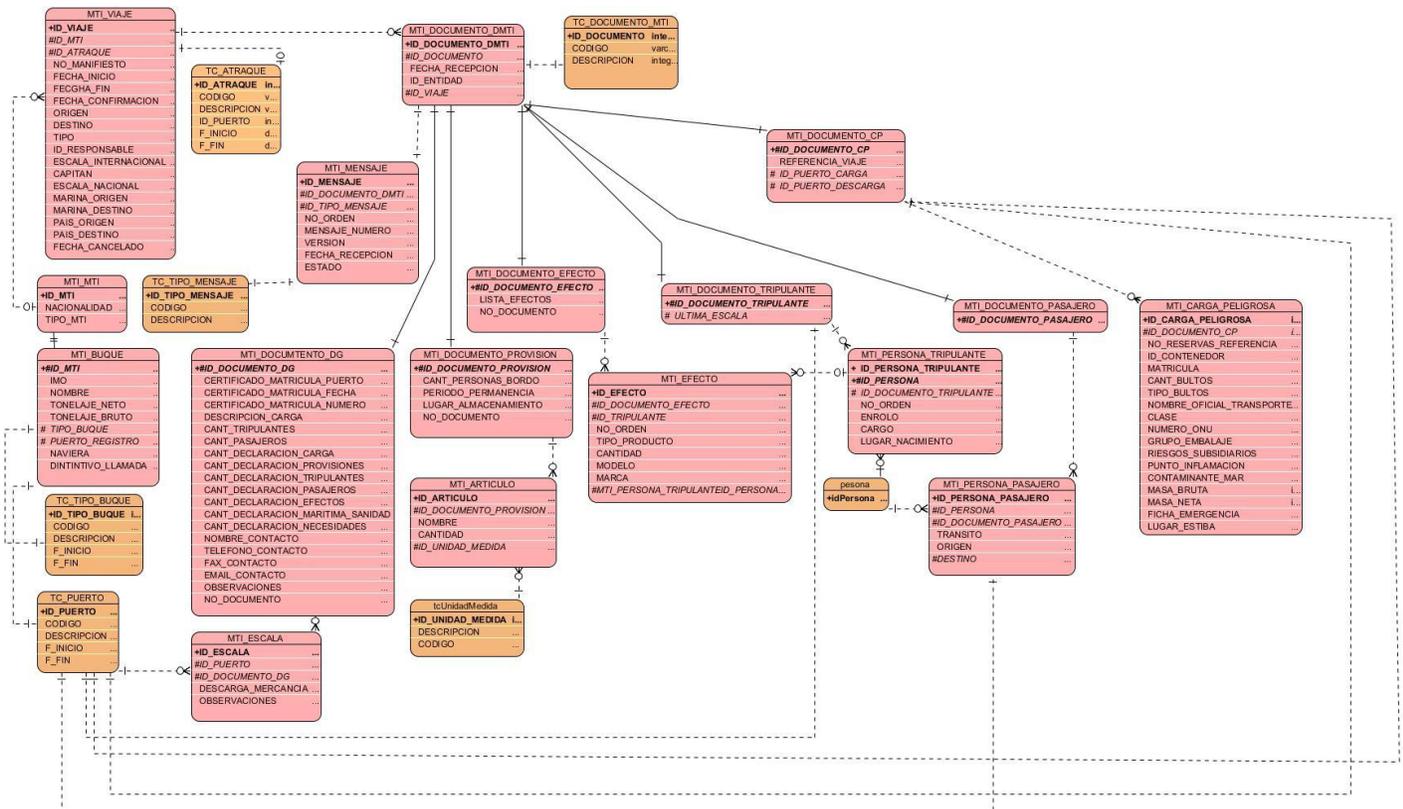


Fig8. Diagrama Entidad-Relación

Se muestra a continuación la descripción de la entidad MTI_DOCUMENTO_DG, la cual está asociada al documento OMI Declaración General.

Nombre: "MTI_DOCUMENTO_DG"		
Tipo: Entidad		
Descripción: Almacena los datos correspondientes al documento OMI Declaración General.		
Atributo	Tipo	Descripción
ID_DOCUMENTO_DG	numeric	Identificador de la tabla.
CERTIFICADO_MATRICULA_FECHA	date	Certificado de la matrícula del buque. Se especifica la fecha de la matrícula.
CERTIFICADO_MATRICULA_NUMERO	varchar	Certificado de la matrícula del buque. Se especifica el número de la matrícula.
DESCRIPCION_CARGA	varchar	Tipo de carga a bordo del buque de forma genérica. Especificar cargas, clasificación IMCO, peligrosas, explosivas u otras que sea de interés señalar.
CANT_TRIPULANTES	numeric	Cantidad de miembros de la tripulación.
CANT_PASAJEROS	numeric	Cantidad de pasajeros, en caso de que el buque

		los transporte.
CANT_DECLARACION_CARGA	numeric	Cantidad de ejemplares de los Manifiestos de Carga entregados para el despacho.
CANT_DECLARACION_PROVISIONES	numeric	Cantidad de ejemplares de la Declaración de Provisiones entregados para el despacho.
CANT_DECLARACION_TRIPULANTES	numeric	Cantidad de ejemplares de la Lista de Tripulación entregados para el despacho.
CANT_DECLARACION_PASAJEROS	numeric	Cantidad de ejemplares de la Lista de Pasajeros entregados para el despacho.
CANT_DECLARACION_EFECTOS	numeric	Cantidad de ejemplares de Efectos de la Tripulación entregados para el despacho.
CANT_DECLARACION_SANIDAD	numeric	Cantidad de ejemplares de la Declaración Marítima de Sanidad entregados para el despacho.
CANT_DECLARACION_NECESIDADES	varchar	Necesidades del buque en cuanto a instalaciones de recepción de desechos y residuos.
NOMBRE_CONTACTO	varchar	Nombre del Agente del Buque.
TELEFONO_CONTACTO	varchar	Teléfono del Agente del Buque.
FAX_CONTACTO	varchar	Fax del Agente del Buque.
OBSERVACIONES	varchar	Señalar la existencia de armas y narcóticos a bordo del buque, así como los números de los Conocimientos de Embarque o cargas peligrosas o explosivos y otros datos de interés.
CERTIFICADO_MATRICULA_PUERTO	numeric	Certificado de la matrícula del buque. Se especifica el puerto de la matrícula.
NO_DOCUMENTO	varchar	Número que se le otorga al documento

Tabla 5. Descripción de la tabla "MTI_DOCUMENTO_DG"

2.8.4 Validación del Diseño

Con el fin de evaluar la calidad del diseño y analizar el tamaño de las clases involucradas en esta funcionalidad, se aplicaron las métricas TOC y RC, explicadas en el capítulo 1. Para la utilización de la Métrica Tamaño Operacional de Clase (TOC), se tuvieron en cuenta los datos mostrados en la siguiente tabla.

No	Clase	Procedimientos
1	MtiDocumentoTripulantePeer	1

2	BaseMtiDocumentoTripulante	28
3	BaseMtiDocumentoTripulanteForm	5
4	MtiMensajeOmiTForm	1
5	BaseMtiPersonaTripulante	47
6	sfMtiIntegracionVU	11
7	sfMtiToolkit	5

Tabla6. Algunas de las clases fundamentales de la herramienta asociada al documento Lista de Tripulantes y sus procedimientos.

La aplicación de la métrica **TOC** define los siguientes atributos de calidad (23):

- ✓ **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.

Atributo	Categoría	Criterio
Responsabilidad	Baja	<=14
	Media	Entre 14 y 28
	Alta	>28
Complejidad de implementación	Baja	<=14
	Media	Entre 14 y 28
	Alta	>28
Reutilización	Baja	>28
	Media	Entre 14 y 28
	Alta	<=14

Tabla7. Rango de valores para los criterios de evaluación de la métrica Tamaño Operacional de Clase (TOC).

No	Clase	Proc.	Responsabilidad	Complejidad	Reutilización
----	-------	-------	-----------------	-------------	---------------

1	MtiDocumentoTripulantePeer	1	Baja	Baja	Alta
2	BaseMtiDocumentoTripulante	28	Media	Media	Media
3	BaseMtiDocumentoTripulanteForm	5	Baja	Baja	Alta
4	MtiMensajeOmiTForm	1	Baja	Baja	Alta
5	BaseMtiPersonaTripulante	47	Alta	Alta	Baja
6	sfMtiIntegracionVU	11	Baja	Baja	Alta
7	sfMtiToolkit	5	Baja	Baja	Alta

Tabla 8 Valores para los criterios de evaluación de la métrica Tamaño Operacional de Clase (TOC).

A continuación se muestra el comportamiento del nivel de procedimientos puesto en práctica en las clases del diseño realizado y se representan los estados de los atributos de calidad: responsabilidad, complejidad de la implementación y reutilización respectivamente.

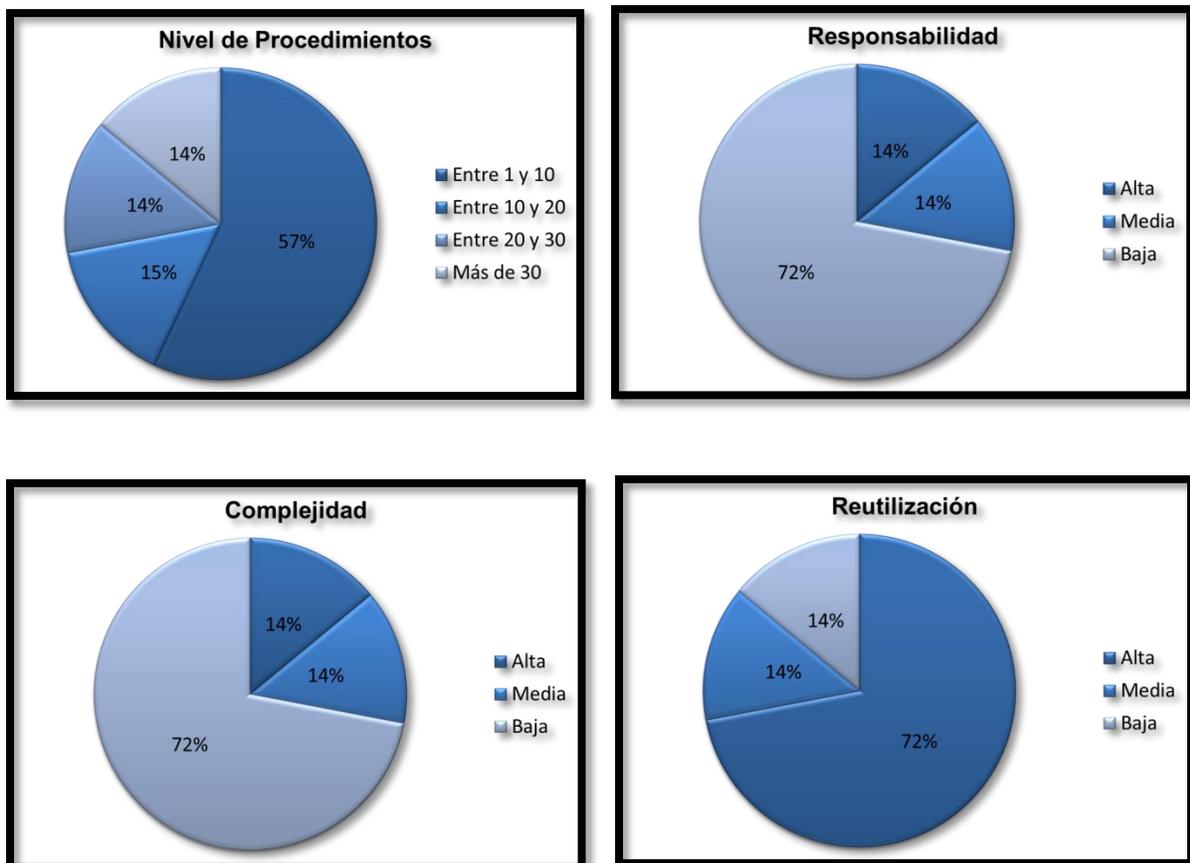


Fig9. Resultados generales de la aplicación de la métrica TOC

Teniendo en cuenta la aplicación de las métricas, se determina que las clases del diseño en la presente solución contienen un promedio de 14 funcionalidades por clases. Encontrándose 3 de ellas en el rango de 1 a 10 funcionalidades, 2 entre 10 y 30, y la restante en el rango de 30 a 50 funcionalidades. Luego de analizar el promedio de funcionalidades por clases, se arriba a la conclusión de que el 72% de las mismas tiene un nivel bajo de responsabilidad y complejidad de implementación, y el resto se dividen en un 14% de un nivel medio y alto en ambos aspectos. Por otra parte, conociendo la inversa proporcionalidad que existe entre la responsabilidad y la complejidad de las clases contra la reutilización, se observa que el 72% de las clases presentan una alta probabilidad de reutilización y el 28% restante un nivel medio y bajo por igual en este parámetro.

Aplicación de la Métrica Relaciones entre clases:

La métrica RC trata de encontrar todas las relaciones que existan entre la clase A con una o varias clases B. Estas relaciones se definen por los atributos o métodos que utilice la clase A de la clase B al menos una vez para darle respuesta a alguna petición. En el proceso de aplicación de esta métrica fue necesario definir de todas las clases persistentes del diseño, las que se apoyan en las operaciones o datos que no se encuentran dentro de sus responsabilidades. Por medio de esta métrica se evalúan los atributos de calidad Acoplamiento, Complejidad de mantenimiento y Cantidad de Pruebas.

Para un total de 29 clases (ver **tabla 9**) que pertenecen al diseño, se obtuvieron tras aplicar la métrica, valores promedio de 1.70 asociaciones de uso por clase.

Total de Clases	Promedio de asociaciones de uso
29	1.7

Tabla9. Total de clases y promedio de asociaciones de uso.

A continuación se muestran los valores de los umbrales para aplicar esta métrica:

Atributo	Categoría	Criterio
Acoplamiento	Ninguno	0
	Bajo	1
	Medio	2
	Alto	Cantidad de relaciones de uso > 2

Complejidad del mantenimiento	Baja	Cantidad de relaciones de uso ≤ 1.7
	Media	$1.7 \leq$ Cantidad de relaciones de uso ≤ 3.4
	Alta	Cantidad de relaciones de uso > 3.4
Cantidad de pruebas	Baja	Cantidad de relaciones de uso ≤ 1.7
	Media	$1.7 \leq$ Cantidad de relaciones de uso ≤ 3.4
	Alta	Cantidad de relaciones de uso > 3.4

Tabla10. Rango de valores para la evaluación de los atributos de calidad relacionados con la métrica.

Luego de ser evaluadas cada una de las clases en correspondencia con los valores de los umbrales definidos para cada uno de los atributos de calidad se obtienen los siguientes resultados:

Cantidad de clases: 29	Baja	Media	Alta
Acoplamiento	57%	31%	12%
Complejidad de mantenimiento	46%	44%	10%
Cantidad de pruebas	46%	44%	10%

Tabla11. Resultados generales de la aplicación de la métrica RC.

Los resultados obtenidos durante la evaluación del instrumento de medición de la métrica RC demuestran que las clases del diseño poseen un bajo acoplamiento, pues para este atributo la categoría bajo tiene un 57% del total. El acoplamiento es una medida de la fuerza con que una clase está conectada a otras clases, con que las conoce y recurre a ellas. Este comportamiento permite la organización de todas las funcionalidades, posibilitando agrupar los procedimientos semejantes, para hacer menos engorroso el proceso de validación y la seguridad. Los atributos complejidad de mantenimiento y cantidad de pruebas, sumaron un 90% en las categorías bajo y medio, lo que demuestra que no es necesario un elevado esfuerzo en el momento de realizar cambios, rectificaciones y pruebas al software.

2.9 Implementación del sistema

Luego de la realización y validación del diseño, se procede a la implementación de la solución propuesta.

2.9.1 Diagrama de componentes

El diagrama de componentes muestra la organización y las dependencias entre un conjunto de componentes. Los mismos constituyen el empaquetamiento físico de los elementos del modelo. En el diagrama que muestra la figura 13 se representa los principales componentes que se utilizan en la herramienta para la recepción de información adelantada de los buques. Se accede al sistema por medio del controlador frontal, que a su vez en dependencia de la petición realizada se comunica con la clase actions; esta clase interactúa con las librerías de Symfony, con las clases del negocio creadas por Propel, y además con los componentes de configuración que están presentes en todas las acciones realizadas en el sistema.

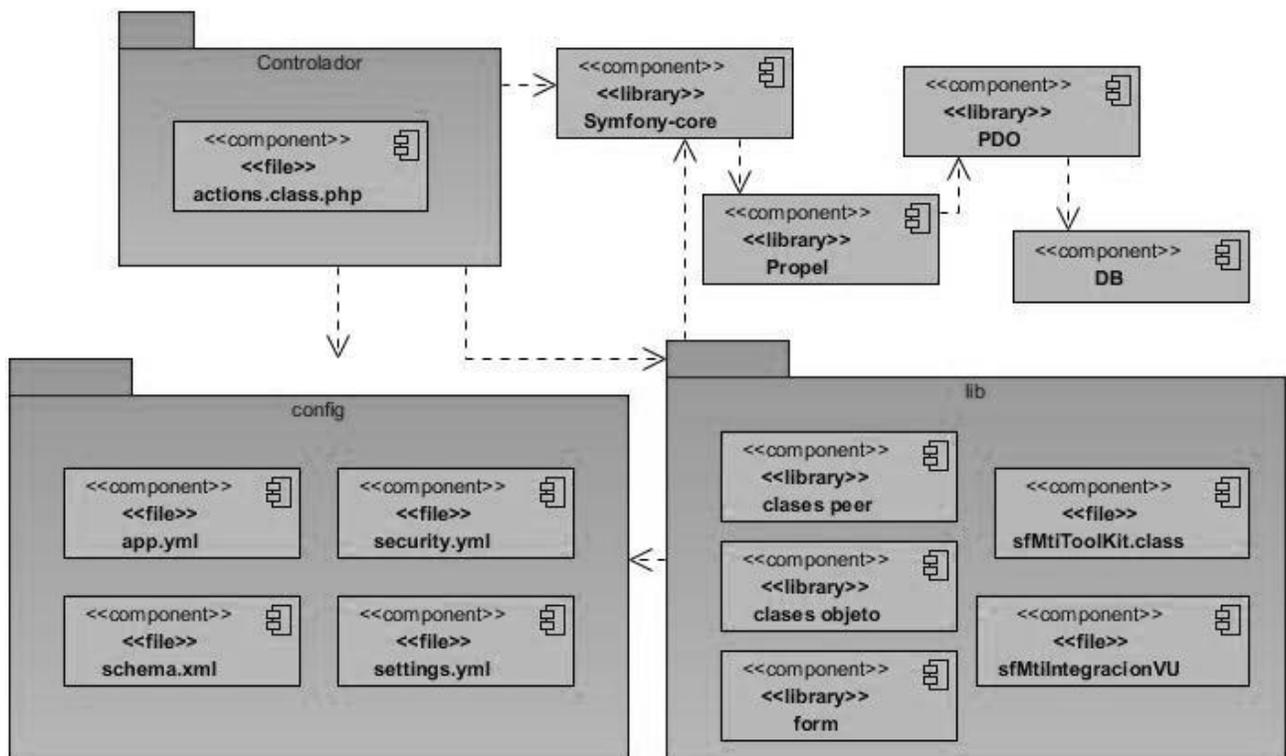


Fig10. Diagrama de Componentes de la Herramienta

2.9.2 Estándares de codificación

Un estándar de codificación no es más que una guía a seguir por los desarrolladores de un proyecto, con el propósito de centrar un estilo propio del equipo a la hora de generar el código fuente de una aplicación. El Departamento de Soluciones para la Aduana define un estándar de codificación que sirve de apoyo al personal del proyecto Aduana para identificar de forma sencilla cuál es el objetivo y las funcionalidades que brinda cada una de las clases, funciones y demás componentes de software dada su nomenclatura. Además debe servir de guía para posteriores implementaciones o modificaciones del sistema. Comprende todo el código

generado bajo la tecnología y el lenguaje PHP y que utilicen la arquitectura regida por la utilización del framework Symfony. (26)

Como la implementación de las aplicaciones está basada en la utilización de un Framework que en su totalidad está codificado en inglés, es imposible desligar la implementación completamente de este idioma. Por lo tanto la implementación se realizará teniendo en cuenta los idiomas de inglés, español, incluso la combinación de ambos al mismo tiempo, sin olvidar la utilización de los sufijos y prefijos “Set”, “Get”, “Is” y “Max”. (26)

Los nombres de las variables deben expresar claramente el contenido de la misma y pueden estar referidas en singular o plural. En caso de que no se le asigne un valor inicial a las variables se deben inicializar con un valor que indique el tipo de dato más general al que debe pertenecer. (26)

Para comentarear una línea se utilizará el o los caracteres que dicta el lenguaje de programación utilizado, al igual que para los comentarios de bloque, es decir, que requiera más de un línea. (26)

2.9.3 Tratamiento de errores

En el desarrollo de una aplicación el tratamiento de errores es fundamental para lograr el correcto funcionamiento del sistema y permite identificar los errores que pueden aparecer cuando el usuario interactúa con el mismo. Para garantizar un correcto tratamiento de errores en el sistema se tratan todos los errores que puedan presentarse durante la comunicación con la base de datos y para lograrlo se realiza la validación que compruebe la corrección de los datos. Uno de los objetivos principales es que el usuario introduzca la menor cantidad de valores posibles para evitar incoherencias de los mismos. Para estos datos se implementan funciones que validen y en caso de errores mostrar mensajes con la información de todos los errores cometidos durante la validación de los datos.

Las validaciones del negocio son las más importantes de todas. Se encargan de controlar el flujo correcto de los datos introducidos, para evitar consecuencias inquietantes. Se lleva a cabo en las diferentes clases por medio del lenguaje PHP. En caso de encontrarse casos de este tipo, se le informa al usuario los datos que están erróneos para que pueda nuevamente arreglarlos.

2.9.4 Comunicación entre las capas

El framework Symfony está basado en el patrón MVC, el cual divide una aplicación en tres capas (ver **epígrafe 1.7**). Mediante este modelo, el framework Symfony interactúa fácilmente con estas capas, facilitando mayor organización e independencia entre las mismas (ver figura 2). Entre el cliente y el controlador frontal y viceversa se utiliza tecnología JSON, por medio de esta los datos pasados por el usuario son serializados³ en objetos PHP, esto permite interactuar con el servidor de la aplicación con cualquier tipo de interfaz y además da seguridad en la comunicación pues los implementadores de interfaz de usuario no van a tener acceso al comportamiento de los objetos que son enviados desde la capa controladora. También se combina esta tecnología con AJAX para aumentar la velocidad y el dinamismo de la interacción con la aplicación siempre que sea posible. Se utiliza AJAX con comunicación basada en XML para cuando no es posible enviar los datos por JSON. (10)

La comunicación entre las capas del Modelo y el Controlador se realiza por el envío de objetos con la información necesaria para el manejo de las peticiones del usuario y de las acciones a realizar, por medio de la tecnología mapeo de objetos a bases de datos (ORM⁴). En la comunicación con la Base de Datos se utiliza PHP Data Objects (PDO⁵), librería que trae el Propel para la abstracción de la Base de Datos. (10)

Para la obtención de la herramienta desarrollada fue necesaria la creación de la clase sfMtiIntegracionVU (ver figura 11). Esta clase permite establecer la comunicación entre los sistemas GINA y Ventanilla Única. Seguidamente se muestran algunos ejemplos de la aplicación donde se evidencia la comunicación entre las capas partiendo de los requisitos funcionales identificados en la fase de análisis.

³ Proceso de convertir el estado de un objeto en un formato que se pueda almacenar o transportar por la red.

⁴ Serie de objetos que permiten acceder a los datos y que contienen en su interior cierta lógica de negocio.

⁵ Proporciona una interfaz entre el código PHP y el código SQL de la base de datos, permitiendo cambiar fácilmente de sistema gestor de bases de datos.

```

class sfMtiIntegracionVU {
    /**...
    ▶ public static function registrarOmi($mensajeOmi) {...}

    /**...
    public static function registrarOmiT($mensajeOmi, $mensajeOmiT) {...}

    /**...
    public static function registrarOmiP($mensajeOmi, $mensajeOmiP) {...}

    /**...
    public static function registrarOmiB($mensajeOmi, $mensajeOmiB) {...}

    /**...
    public static function registrarOmiG($mensajeOmi, $mensajeOmiG) {...}

    /**...
    public static function registrarOmiE($mensajeOmi, $mensajeOmiE) {...}

    /**...
    public static function registrarOmiM($mensajeOmi, $mensajeOmiM) {...}

```

Fig11. Clase sfMtiIntegracionVU de la herramienta

La pantalla principal de la Herramienta (ver figura 12) parte en el momento que el usuario selecciona la opción Subir Fichero.

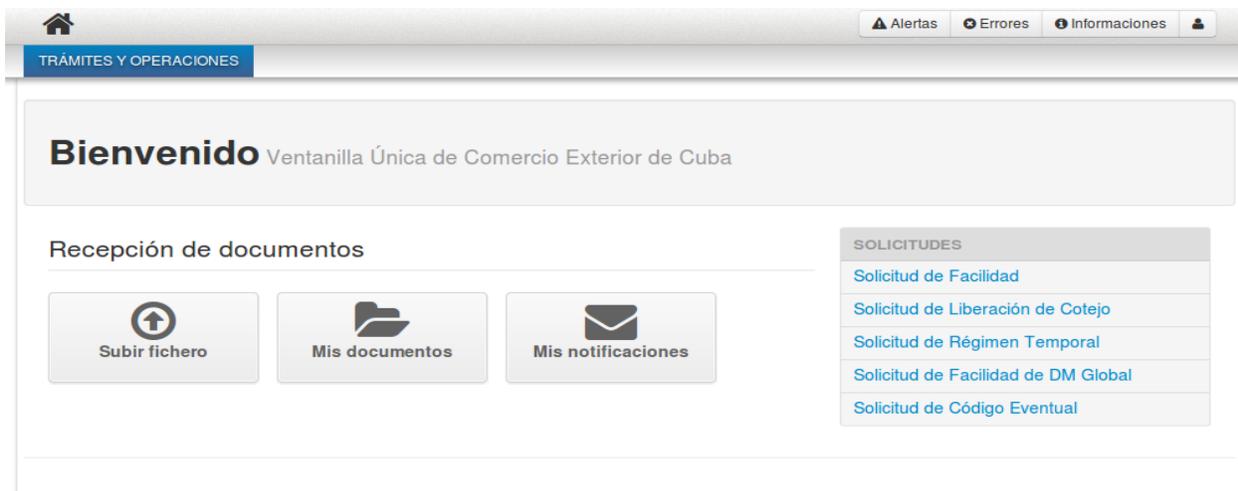


Fig.12 Trámites y Operaciones, espacio para subir el fichero correspondiente

La siguiente figura muestra el mensaje recibido cuando el fichero es subido correctamente, otorgándole al mismo un numero identificativo en la Ventanilla Única.

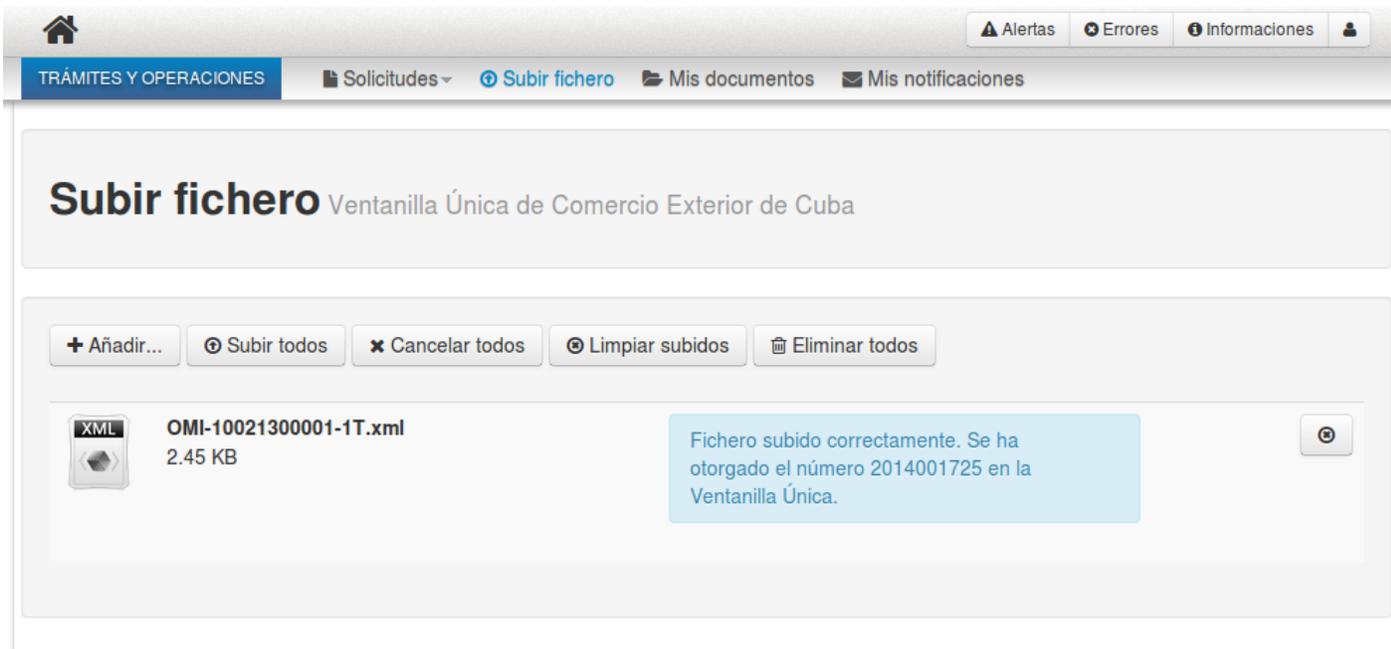


Fig.13 Fichero subido correctamente

La información contenida en el fichero es validada tanto estructuralmente como desde el punto de vista del negocio. Cuando no existen errores, el documento es registrado y se emite un mensaje de Aceptación. La figura siguiente muestra un fragmento de código del método registrarOmiT, que se encarga de validar y registrar el documento correspondiente a la Declaración de Lista de Tripulantes.

```

public static function registrarOmiT($mensajeOmi, $mensajeOmiT) {

    $mensajeOmiT['datosGenerales']['ultimoProximoPuertoEscala'] = array_key_exists('ultimoPuertoEscala', $mensajeOmi['datosDocumento'])['da

    if (empty($mensajeOmi['datosDocumento']['tripulantes'])) {...}

    if (count($mensajeOmi['datosDocumento']['tripulantes']['tripulante'] == 0) {...}
    $tripulantes = $mensajeOmi['datosDocumento']['tripulantes']['tripulante'];
    $mensajeOmiT['tripulantes'] = array();

    if (!array_key_exists(0, $tripulantes)) {

        $mensajeOmiT['tripulantes'][] = array(
            'primerApellido' => array_key_exists('primerApellido', $tripulantes['datosPersonales']) ? $tripulantes['datosPersonales']['pri
            'segundoApellido' => array_key_exists('segundoApellido', $tripulantes['datosPersonales']) ? $tripulantes['datosPersonales']['s
            'primerNombre' => array_key_exists('primerNombre', $tripulantes['datosPersonales']) ? $tripulantes['datosPersonales']['primerN
            'segundoNombre' => array_key_exists('segundoNombre', $tripulantes['datosPersonales']) ? $tripulantes['datosPersonales']['segun
            'nacionalidad' => array_key_exists('nacionalidad', $tripulantes['datosPersonales']) ? $tripulantes['datosPersonales']['naciona
            'fechaNacimiento' => array_key_exists('fechaNacimiento', $tripulantes['datosPersonales']) ? $tripulantes['datosPersonales']['f
            'lugarNacimiento' => array_key_exists('lugarNacimiento', $tripulantes['lugarNacimiento']) ? $tripulantes['datosPersonales']['l
            'pasaporte' => array_key_exists('pasaporte', $tripulantes['datosPersonales']) ? $tripulantes['datosPersonales']['pasaporte'] :
            'noOrden' => array_key_exists('numeroConsecutivo', $tripulantes) ? $tripulantes['numeroConsecutivo'] : null,
            'cargo' => array_key_exists('grado', $tripulantes['datosPersonales']) ? $tripulantes['datosPersonales']['grado'] : null,
        );
    } else {

        foreach ($tripulantes as $tripulante) {

```

Fig.14 Fragmento de código del método registrarOmiT

La validación del mismo se realiza mediante el uso de formularios, utilizando validadores que brinda el propio *Symfony*, y en algunos casos creados por el desarrollador de acuerdo a las

necesidades y validaciones a realizar. En la figura siguiente se muestran por ejemplo validadores como *sfValidatorString*, encargado de especificar que el máximo de caracteres que debe tener el Número de Documento debe ser 30, y *sfValidatorPropelChoice*, que valida que el valor original se encuentre en la lista de registros asociada a un modelo de Propel. Por defecto se comprueba que el valor original sea la clave primaria del registro, pero se puede comprobar cualquier otra columna indicada mediante la opción “column” como es el caso del ejemplo siguiente. (27)

```

public function setup()
{
    $this->setWidgets(array(
        'NO_DOCUMENTO' => new sfWidgetFormInput(),
        'ULTIMA_ESCALA' => new sfWidgetFormPropelChoice(array('model' => 'TcPuerto', 'add_empty' => false)),
        'ID_DOCUMENTO_TRIPULANTE' => new sfWidgetFormInputHidden(),
    ));

    $this->setValidators(array(
        'NO_DOCUMENTO' => new sfValidatorString(array('max_length' => 30)),
        'ULTIMA_ESCALA' => new sfValidatorPropelChoice(array('model' => 'TcPuerto', 'column' => 'ID_PUERTO')),
        'ID_DOCUMENTO_TRIPULANTE' => new sfValidatorPropelChoice(array('model' => 'MtiDocumentoDmti', 'column' => 'ID_DOCUMENTO_DMTI', 'required'
    ));

    $this->widgetSchema->setNameFormat('mti_documento_tripulante[%s]');

    $this->errorSchema = new sfValidatorErrorSchema($this->validatorSchema);

    parent::setup();
}

public function getModelName()
{
    return 'MtiDocumentoTripulante';
}

```

Fig.15 Fragmento de código de la clase BaseMtiDocumentoTripulanteForm

La siguiente figura muestra la interfaz de la herramienta en el caso de no existir errores, cuando el usuario compruebe la opción Mis notificaciones. Esta funcionalidad permite filtrar las notificaciones por diversos aspectos como rango de fechas, tipo de notificación, tipo de documento, entre otras opciones.

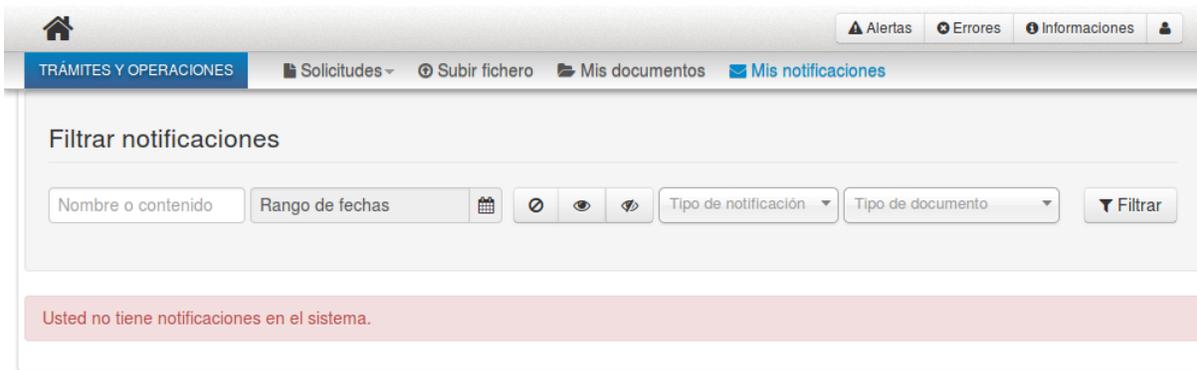


Fig.16 Notificaciones

La imagen siguiente muestra algunos de los errores que pueden ser encontrados cuando es validado un documento.

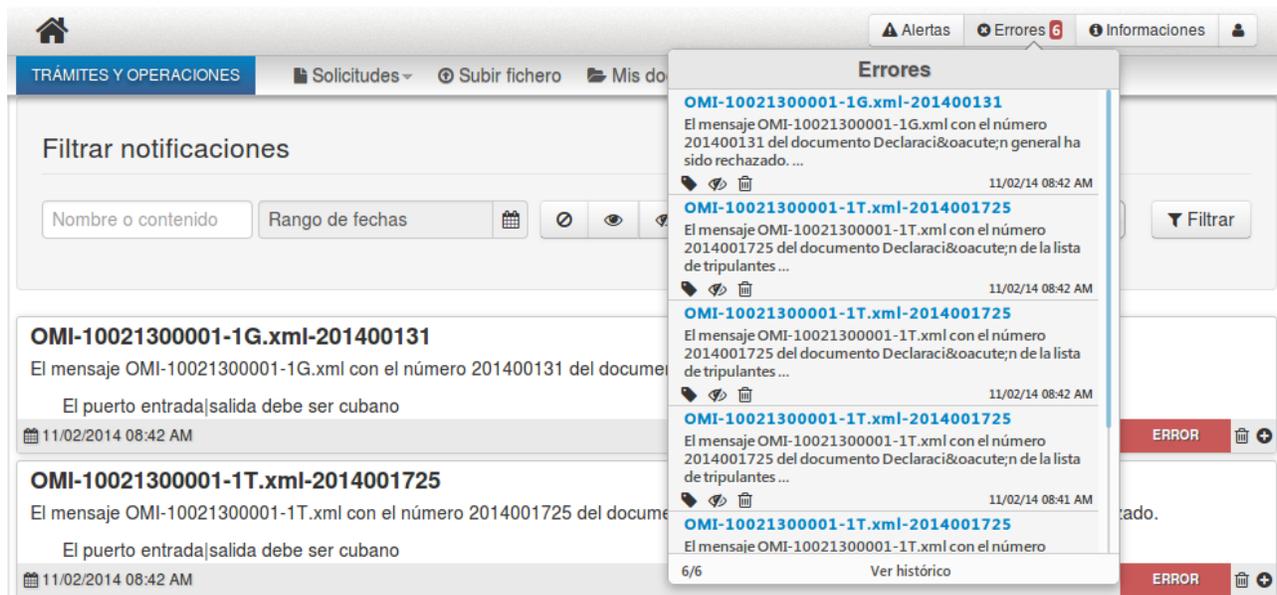


Fig.17 Notificaciones de errores

Por último, la figura 18 muestra un fragmento de código del método registrarOmi en el cual se evidencia como son creadas las notificaciones mostradas anteriormente.

```

$resultado = array(
    'nombreDocumento' => $nombreDocumento,
    'estado' => count($errores) > 0 ? $config['all']['estado_documentos_con_errores'] : $config['all']['estado_documentos_sin_errores'],
    'noVentanilla' => $mensajeOmi['numeroVU'],
    'nombreMensaje' => $mensajeOmi['nombreFichero'],
    'enviarA' => $config['all']['envio_notificacion_a_documento'],
    'errores' => count($errores) > 0 ? $errores : null,
);

sfMtiToolkit::crearNotificacion($resultado, sfMtiToolkit::tipo_error_contenido);

```

Fig.18 Fragmento de código de una notificación

2.10 Conclusiones Parciales

La utilización de patrones durante el diseño de la solución permitió llevar a cabo un buen diseño funcional en esta etapa, adecuado para los requisitos que satisfacen la solución. Con la obtención de cada uno de los artefactos del diseño definidos por el Departamento de Soluciones para la Aduana del CEIGE se logró fundamentar la implementación del sistema. El diseño y la implementación de las clases del negocio permitieron obtener una herramienta capaz de llevar a cabo eficazmente el proceso de recepción de la información adelantada de los MTI y la comunicación entre las diferentes aplicaciones del sistema GINA.

CAPÍTULO 3: VALIDACIÓN DE LA SOLUCIÓN

3.1 Introducción

En este capítulo se describen elementos importantes en las pruebas de la solución, teniendo como objetivo principal comprobar la calidad del resultado de la implementación, mediante los artefactos elaborados durante el flujo de trabajo de pruebas, mostrando los resultados obtenidos al aplicarlas a la herramienta.

3.2 Verificación del sistema

Pruebas de Rendimiento y Estrés

Esta prueba es diseñada para probar el rendimiento del software en tiempo de ejecución dentro del contexto de un sistema integrado. Se realiza en todos los pasos del proceso de prueba. Está incluido al nivel de unidad. Es necesario que estén integrados todos los elementos para poder asegurar el rendimiento del sistema. La prueba de resistencia o de estrés es diseñada para enfrentar los programas con situaciones anormales (Extrema sobrecarga, insuficiente memoria, servicios y hardware no disponible, recursos compartidos no disponibles). El objetivo es ejecutar el sistema para que demande gran cantidad de recursos, frecuencia o volúmenes anormales. Esta prueba busca combinaciones de datos dentro de un rango válido que puedan producir inestabilidad o procesos incorrectos.

Se realizaron las pruebas de rendimiento y estrés, obteniéndose los siguientes resultados: para 500 documentos se obtuvo como tiempo mínimo 120 segundos de respuesta, y como tiempo máximo 140,75 segundos. El rendimiento está en 0,28 segundos por cada documento. Teniendo en cuenta los resultados obtenidos, se considera que la prueba ha sido satisfactoria.

Pruebas de Caja Negra

Las pruebas de caja negra se llevan a cabo sobre la interfaz del software, obviando el comportamiento interno y la estructura del programa, verificando que la entrada se acepta de forma adecuada y que se produce una salida correcta, así como que la integridad de la información externa se mantiene. (28)

Este método consta de varias técnicas, entre ellas se encuentra la Técnica Partición de Equivalencia, la cual divide el campo de entrada de un programa en clases de datos de los que se pueden derivar casos de prueba. La partición equivalente se dirige a una definición de casos de prueba que descubran clases de errores, reduciendo así el número total de casos de prueba que hay que desarrollar.

Técnica partición de equivalencia

La técnica a emplear es la de Partición de Equivalencia, donde una clase de equivalencia representa un conjunto de estados válidos, inválidos o que no aplican, para varias condiciones de entrada. Las condiciones de entrada son valores numéricos específicos, un rango de valores, un conjunto de valores relacionados o una condición lógica.

Se le realizaron las pruebas a los requisitos del sistema aplicando un total de 6 casos de prueba (uno para cada RF) con sus Escenarios de Pruebas (EC) correspondientes. Un ejemplo de ello es el requisito Procesar Información Adelantada de la Lista de Tripulantes, el cual consta de cuatro escenarios, como se muestra a continuación:

- ✓ EC 1.1 Subir fichero xml con datos inválidos.
- ✓ EC 1.2 Subir fichero xml con datos válidos.
- ✓ EC 1.3 Subir fichero inválido.
- ✓ EC 1.4 Añadir fichero con extensión no válida.

Resultado de las pruebas

Durante la puesta en práctica de estos casos de prueba en una primera iteración se reportaron un total de tres no conformidades que se centran fundamentalmente en la validación de los datos teniendo en cuenta el formato establecido. Una vez detectadas las mismas fueron resueltas reduciendo esta cifra a cero. Se realiza una segunda iteración para identificar cualquier otra no conformidad que pudiese existir. En la figura se pueden observar los resultados obtenidos en las iteraciones.

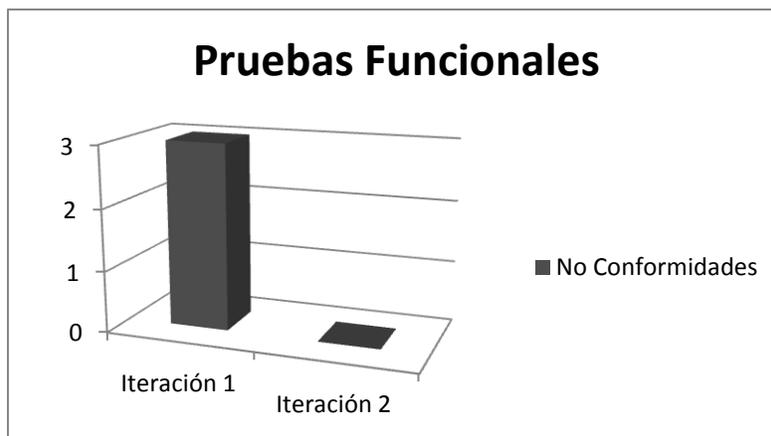


Fig19. Resultados de la aplicación de Pruebas Funcionales.

Dado los resultados obtenidos en la segunda iteración la herramienta quedó en total funcionamiento y acorde a las necesidades funcionales requeridas.

Pruebas de Caja Blanca

Estas pruebas se basan en la especificación del programa o componente a ser probado para elaborar los casos de prueba, en este tipo de prueba se emplea la técnica del camino básico.

Técnica del camino básico

Para aplicar esta prueba se definen una serie de pasos a seguir que a continuación se describen:

1. **Notación del grafo de flujo:** usando el código como base se realiza la representación del grafo de flujo, mediante una sencilla notación. Cada construcción estructurada tiene su correspondiente símbolo.
 - ✓ Nodo: cada círculo, denominado nodo, representa una o más sentencias procedimentales.
 - ✓ Arista: las flechas del grafo de flujo, denominadas aristas, representan flujo de control y son análogas a las flechas del diagrama de flujo.
 - ✓ Región: las áreas delimitadas por aristas y nodos se denominan regiones.
2. **Complejidad ciclomática:** es una métrica que proporciona una medición cuantitativa de la complejidad lógica de un programa. El valor calculado define el número de caminos independientes del conjunto básico de un programa y da un límite superior para el número de pruebas que se deben realizar para asegurar que se ejecuta cada sentencia al menos una vez. La complejidad ciclomática se calcula de tres formas:
 1. El número de regiones del grafo de flujo coincide con la complejidad ciclomática.
 2. La complejidad ciclomática, $V(G)$, de un grafo de flujo G se define como: $V(G) = A - N + 2$, donde A es el número de aristas del grafo de flujo y N es el número de nodos del mismo.
 3. La complejidad ciclomática, $V(G)$, de un grafo de flujo G también se define como $V(G) = P + 1$, donde P es el número de nodos predicado contenidos en el grafo de flujo G .
3. **Determinar un conjunto básico de caminos linealmente independientes:** el valor de $V(G)$ representa el número de caminos linealmente independientes de la estructura de control del programa.
4. **Obtención de casos de prueba:** se realizan los casos de pruebas que forzarán la ejecución de cada camino del conjunto básico.

Aplicación de la prueba

La siguiente figura es el código que se tomó como base para realizar el procedimiento anteriormente descrito para la técnica de camino base, correspondiente al método registrarOMI

```
public static function registrarOmi($mensajeOmi) {
    $erroresFormato = array();
    //1
    if (empty($mensajeOmi)) { //2
        throw new Exception('El documento est&aacute; vac&iacute;o');
        //3
    } else { //4
        if (!array_key_exists('numeroVU', $mensajeOmi) || empty($mensajeOmi['numeroVU'])) { //5
            throw new Exception('El campo numeroVU no existe o es&aacute; vac&iacute;o');
            //6
        }
        if (!array_key_exists('nombreFichero', $mensajeOmi) || empty($mensajeOmi['nombreFichero'])) { //7
            $erroresFormato[] = "El campo nombreFichero no existe o es&aacute; vac&iacute;o";
            //8
        }
        if (!array_key_exists('fecha', $mensajeOmi) || empty($mensajeOmi['fecha'])) { //9
            $erroresFormato[] = "El campo fecha no existe o es&aacute; vac&iacute;o";
            //10
        }
        if (!array_key_exists('datosDocumento', $mensajeOmi) || empty($mensajeOmi['datosDocumento'])) { //11
            $erroresFormato[] = "El campo datosDocumento no existe o es&aacute; vac&iacute;o";
            //12
        } else { //13
            if (!array_key_exists('datosGenerales', $mensajeOmi['datosDocumento']) || empty($mensajeOmi[
                'datosDocumento']['datosGenerales'])) { //14
                $erroresFormato[] = "El campo datosGenerales no existe o est&aacute; vac&iacute;o";
                //15
            }
        }
    } else { //16
        if (!array_key_exists('mensajeNumero', $mensajeOmi['datosDocumento']['datosGenerales']) || empty(
            $mensajeOmi['datosDocumento']['datosGenerales']['mensajeNumero'])) { //17
            $erroresFormato[] = "El campo mensajeNumero no existe o est&aacute; vac&iacute;o.";
            //18
        }
        if (!array_key_exists('numeroImo', $mensajeOmi['datosDocumento']['datosGenerales']) || empty(
            $mensajeOmi['datosDocumento']['datosGenerales']['numeroImo'])) { //19
            $erroresFormato[] = "El campo numeroImo no existe o est&aacute; vac&iacute;o.";
            //20
        }
        if (!array_key_exists('nombreBuque', $mensajeOmi['datosDocumento']['datosGenerales']) || empty(
            $mensajeOmi['datosDocumento']['datosGenerales']['nombreBuque'])) { //21
            $erroresFormato[] = "El campo nombreBuque no existe o est&aacute; vac&iacute;o.";
            //22
        }
        if (!array_key_exists('tipoBuque', $mensajeOmi['datosDocumento']['datosGenerales']) || empty(
            $mensajeOmi['datosDocumento']['datosGenerales']['tipoBuque'])) { //23
            $erroresFormato[] = "El campo tipoBuque no existe o est&aacute; vac&iacute;o.";
            //24
        }
        if (!array_key_exists('numeroImo', $mensajeOmi['datosDocumento']['datosGenerales']) || empty(
            $mensajeOmi['datosDocumento']['datosGenerales']['numeroImo'])) { //25
            $erroresFormato[] = "El campo numeroImo no existe o est&aacute; vac&iacute;o.";
            //26
        }
        if (!array_key_exists('puertoLlegadaSalida', $mensajeOmi['datosDocumento']['datosGenerales']) ||
            empty($mensajeOmi['datosDocumento']['datosGenerales']['puertoLlegadaSalida'])) { //27
            $erroresFormato[] = "El campo puertoLlegadaSalida no existe o est&aacute; vac&iacute;o.";
            //28
        }
    }
}
```



```

    if (array_key_exists('contactos', $mensajeOmi['datosDocumento']['datosGenerales']['agenteBuque'])) { //43
        $mensajeOmiAux['datosGenerales']['telefono'] = array_key_exists('telefonos', $mensajeOmi[
            'datosDocumento']['datosGenerales']['agenteBuque']['contactos']) ? $mensajeOmi['datosDocumento']['
            'datosGenerales']['agenteBuque']['contactos']['telefonos'] : null;
        $mensajeOmiAux['datosGenerales']['fax'] = array_key_exists('fax', $mensajeOmi['datosDocumento']['
            'datosGenerales']['agenteBuque']['contactos']) ? $mensajeOmi['datosDocumento']['datosGenerales']['
            'agenteBuque']['contactos']['fax'] : null;
        $mensajeOmiAux['datosGenerales']['email'] = array_key_exists('email', $mensajeOmi['datosDocumento']['
            'datosGenerales']['agenteBuque']['contactos']) ? $mensajeOmi['datosDocumento']['datosGenerales']['
            'agenteBuque']['contactos']['email'] : null;
        //44
    }
    //45
    $config = sfYaml::load(sfConfig::get('sf_apps_dir') . '/mti/config/app.yml');
    $tipoDocumento = $mensajeOmi['tipoDocumento'];
    $errores = array();
    $nombreDocumento = "";
    switch ($tipoDocumento) { //46
        case 'OMI_ListaTripulantes': { //47
            $errores = sfMtiIntegracionVU::registrarOmiT($mensajeOmi, $mensajeOmiAux);
            $nombreDocumento = "Declaraci&oacute;n de la lista de tripulantes";
        }
        break;
        case 'OMI_ListaPasajeros': { //48
            $errores = sfMtiIntegracionVU::registrarOmiP($mensajeOmi, $mensajeOmiAux);
            $nombreDocumento = "Declaraci&oacute;n de la lista de pasajeros";
        }
        break;
        case 'OMI_DeclaracionProvisionesBuque': { //49
            $errores = sfMtiIntegracionVU::registrarOmiB($mensajeOmi, $mensajeOmiAux);
            $nombreDocumento = "Declaraci&oacute;n de la lista de provisiones del buque";
        }
        break;
        case 'OMI_DeclaracionGeneral': { //50
            $errores = sfMtiIntegracionVU::registrarOmiG($mensajeOmi, $mensajeOmiAux);
            $nombreDocumento = "Declaraci&oacute;n general";
        }
        break;
        case 'OMI_DeclaracionEfectosTripulacion': { //51
            $errores = sfMtiIntegracionVU::registrarOmiE($mensajeOmi, $mensajeOmiAux);
            $nombreDocumento = "Declaraci&oacute;n de la lista de efectos de la tripu;aci&oacte;n";
        }
        break;
        case 'OMI_DeclaracionMercanciasPeligrosas': { //52
            return sfMtiIntegracionVU::registrarOmiM($mensajeOmi, $mensajeOmiAux);
            $nombreDocumento = "Declaraci&oacute;n de mercanc&iacute;as peligrosas";
        }
        break;
        default: //53
            break;
    }
    //54
    $resultado = array(
        'nombreDocumento' => $nombreDocumento,
        'estado' => count($errores) > 0 ? $config['all']['estado_documentos_con_errores'] : $config['all']['
            'estado_documentos_sin_errores'],
        'noVentanilla' => $mensajeOmi['numeroVU'],
        'nombreMensaje' => $mensajeOmi['nombreFichero'],
        'enviar' => $config['all']['envio_notificacion_a_documento'],
        'errores' => count($errores) > 0 ? $errores : null,
    );
    sfMtiToolkit::crearNotificacion($resultado, sfMtiToolkit::tipo_error_contenido);
}

```

Fig20. Función registrarOmi

Se realiza el grafo de flujo del método seleccionado:

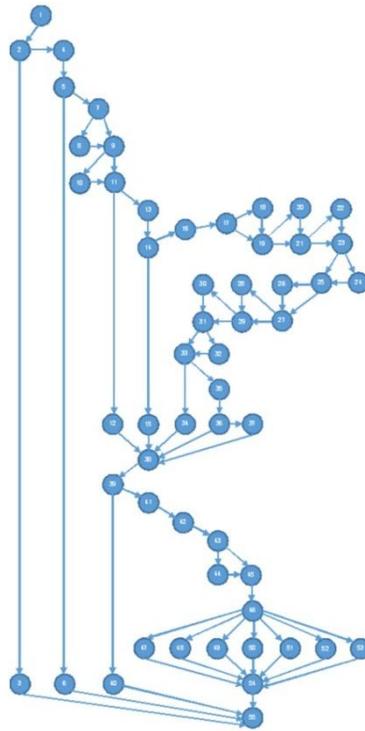


Fig21. Grafo de flujo de la función registrarOmi

Luego se calcula la **complejidad ciclomática**:

$$V(G) = R = 25$$

$$V(G) = A - N + 2 = 78 - 55 + 2 = 25$$

$$V(G) = P + 1 = 24 + 1$$

El valor de $V(G)$ da el número de **caminos linealmente independientes** de la estructura de control del programa, por lo que se definen 25 caminos, entre los que se encuentran:

Camino 1: 1, 2, 3, 55

Camino 2: 1, 2, 4, 5, 6, 55

Camino 3: 1, 2, 4, 5, 7, 9, 11, 12, 38, 39, 40, 55

Camino 4: 1, 2, 4, 5, 7, 9, 11, 13, 14, 15, 38, 39, 40, 55

Camino 5: 1, 2, 4, 5, 7, 9, 11, 13, 14, 16, 17, 19, 21, 23, 25, 27, 29, 31, 33, 34, 38, 39, 40, 55

Cada camino independiente es un **caso de prueba** a realizar, de forma que los datos introducidos provoquen que se visiten las sentencias vinculadas a cada nodo del camino.

A continuación se muestran ejemplos de 5 casos de prueba que se realizaron en el sistema teniendo en cuenta los caminos básicos de esta funcionalidad.

Caso de prueba para el Camino básico #1:

Camino 1: {1-2-3-55}

- ✓ Descripción: Los datos de entrada cumplirán con los siguientes requisitos: el Documento OMI que se desea registrar no debe contener datos.
- ✓ Condición de ejecución: El sistema debe haber recibido el documento que se desea registrar.
- ✓ Entradas: documento OMI.
- ✓ Resultados esperados: mensaje de error "El documento está vacío".

Caso de prueba para el Camino básico #2:

Camino 2: {1-2-4-5-6-55}

- ✓ Descripción: Los datos de entrada cumplirán con los siguientes requisitos: el Documento OMI que se desea registrar no debe contener el campo "número" o tenerlo vacío.
- ✓ Condición de ejecución: El sistema debe haber recibido el documento que se desea registrar.
- ✓ Entradas: documento OMI.
- ✓ Resultados esperados: mensaje de error "El campo número no existe o está vacío".

Caso de prueba para el Camino básico #3:

Camino 3: {1-2-4-5-7-9-11-12-38-39-40-55}

- ✓ Descripción: Los datos de entrada cumplirán con los siguientes requisitos: el Documento OMI que se desea registrar no debe contener errores en el documento, excepto en el campo 'datosDocumento' el cual debe dejarse vacío.
- ✓ Condición de ejecución: El sistema debe haber recibido el documento que se desea registrar.
- ✓ Entradas: documento OMI.
- ✓ Resultados esperados: mensaje de error "El campo datosDocumento no existe o está vacío".

Caso de prueba para el Camino básico #4:

Camino 4: {1-2-4-5-7-9-11-13-14-15-38-39-40-55}

- ✓ Descripción: Los datos de entrada cumplirán con los siguientes requisitos: el Documento OMI que se desea registrar no debe contener errores en el documento excepto en el

campo 'datosGenerales' que se encuentra en 'datosDocumento', este campo debe dejarse vacío.

- ✓ Condición de ejecución: El sistema debe haber recibido el documento que se desea registrar.
- ✓ Entradas: documento OMI.
- ✓ Resultados esperados: mensaje de error "El campo datosGenerales no existe o está vacío".

Caso de prueba para el Camino básico #5:

Camino 5: {1-2-4-5-7-9-11-13-14-16-17-19-21-23-25-27-29-31-33-34-38-39-40-55}

- ✓ Descripción: Los datos de entrada cumplirán con los siguientes requisitos: el Documento OMI que se desea registrar no debe contener errores en el documento excepto en el campo 'agenteBuque' que está en 'datosGenerales' que se encuentra en 'datosDocumento', este campo debe dejarse vacío.
- ✓ Condición de ejecución: El sistema debe haber recibido el documento que se desea registrar.
- ✓ Entradas: documento OMI.
- ✓ Resultados esperados: mensaje de error "El campo agenteBuque no existe o está vacío".

Para llevar a cabo las verificaciones se fueron haciendo modificaciones en un documento OMI para probar lo expuesto en cada uno de los casos de pruebas, arrojando como resultado que la función se encontraba en correcto funcionamiento verificándose que cada camino arrojará los resultados esperados de acuerdo al flujo de datos que debía circular.

3.3 Validación del sistema

La validación se logra cuando el software funciona de acuerdo a las expectativas del cliente. La prueba se concentra en las acciones visibles para el usuario y en la salida del sistema que este puede reconocer. Para comprobar que la aplicación desarrollada muestra los resultados esperados por el cliente, se realizaron los casos de prueba de aceptación. *"Las pruebas de aceptación tienen como fin validar que el sistema cumple los requisitos básicos de funcionamiento esperado y permitir que el usuario determine la aceptación del sistema."* (29)

A continuación se muestra el caso de prueba correspondiente a la funcionalidad Procesar información adelantada de la Lista de Tripulantes:

Caso de prueba de aceptación
Nombre: Procesar información adelantada de la Lista de Tripulantes.
Descripción: Prueba para la funcionalidad de Procesar información adelantada de la Lista de Tripulantes en la aplicación.
Condiciones de Ejecución: <ul style="list-style-type: none"> ✓ El usuario debe seleccionar el documento a subir. ✓ El usuario debe seleccionar la opción Subir.
Resultados esperados: el sistema procesa la información adelantada de la Lista de Tripulantes: <ul style="list-style-type: none"> ✓ Muestra un mensaje indicando que el fichero se subió correctamente y el número otorgado en la Ventanilla Única al documento. ✓ Muestra una notificación de error que contiene el listado de errores detectados en el GINA. ✓ Actualiza el estado del documento.
Evaluación de la prueba: prueba satisfactoria.

Tabla 12 Caso de Prueba para la funcionalidad de Procesar información adelantada de la Lista de Tripulantes en la aplicación.

Para la validación se diseñaron 6 casos de prueba de aceptación. En la primera iteración 5 pruebas mostraron los resultados esperados por el cliente representando aproximadamente un 83,3%, mientras que 1 prueba resultó fallida representando un 16,6% de las pruebas insatisfactorias. En una segunda iteración los resultados fueron satisfactorios pues se obtuvieron un total de 6 pruebas exitosas para un 100% de pruebas satisfactorias y un 0% de pruebas fallidas. A continuación se muestra cómo quedan reflejados los resultados por cada una de las iteraciones de los casos de pruebas de aceptación realizadas al sistema:

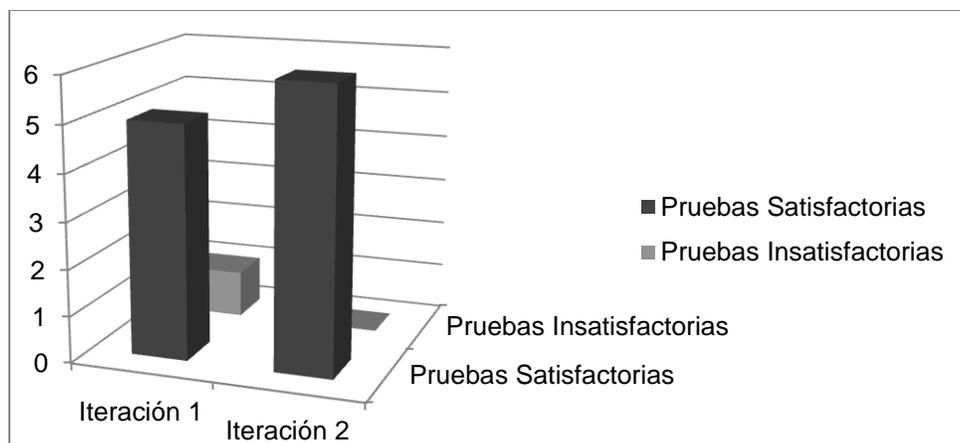


Figura 22 Resultados de la aplicación de Pruebas de Aceptación.

3.3 Validación de la variable

El desarrollo de la investigación estuvo guiado por la siguiente Idea a defender: Con el desarrollo de una herramienta que permita la gestión de la información adelantada establecida por la Organización Marítima Internacional del sistema Despacho de Medios de Transporte Internacional disminuirá el tiempo de consulta de la información adelantada establecida por la Organización Marítima Internacional que recibe la Aduana General de la República por vía electrónica.

Variable	Indicadores
Disminución del tiempo de consulta de la información adelantada establecida por la Organización Marítima Internacional que recibe la Aduana General de la República por vía electrónica.	Tiempo.

Tabla 13 Evaluación de la variable.

Para validar que la aplicación cumplía con estos indicadores se realizó una prueba piloto en el CADi tomando una serie de documentos, arrojando los siguientes resultados:

Indicadores	Antes	Después
Tiempo	Demora en la consulta de los documentos OMI (Tomando un despacho que contaba con un total	Rapidez en la consulta de los documentos OMI (Tomando un despacho que contaba con un total

	de 6 documentos OMI el proceso demoró de 30-40 minutos).	de 6 documentos OMI el proceso demoró menos de 1 minuto).
--	--	---

Tabla 14 Resultados de los indicadores de evaluación.

- ✓ Antes la consulta de los documentos OMI se demoraba mucho, ya que se realizaba mediante hojas de cálculo, luego de la realización de la herramienta se redujo el tiempo para llevar a cabo este proceso, esto se comprobó tomando un despacho que contaba con un total de 6 documentos OMI, el proceso demoró de 30-40 minutos realizándolo de forma manual, mientras que con la aplicación no demoró más de un minuto.

Con los elementos antes expuestos se demuestra el cumplimiento del problema de investigación planteado, constatando que con el desarrollo del sistema realizado, disminuye el tiempo de consulta de la información adelantada establecida por la Organización Marítima Internacional que recibe la Aduana General de la República por vía electrónica.

3.4 Conclusiones Parciales

La calidad de la herramienta implementada fue la premisa fundamental en el desarrollo de este capítulo. Se presentaron los artefactos generados durante la verificación y validación de la herramienta, lo que permitió obtener resultados satisfactorios en las pruebas de rendimiento y estrés, caja blanca y caja negra realizadas a la aplicación. Además se obtuvo una herramienta que permite disminuir el tiempo de consulta de la información adelantada establecida por la Organización Marítima Internacional del sistema Despacho de Medios de Transporte Internacional, la cual responde completamente a los requisitos funcionales y no funcionales definidos, demostrándose la calidad de la misma.

CONCLUSIONES

- ✓ La revisión bibliográfica sobre la recepción de la información adelantada establecida por la OMI permitió identificar y establecer las bases para el desarrollo de la herramienta para la gestión de la información adelantada establecida por la OMII del sistema Despacho de MTI.
- ✓ Se identificaron las principales funcionalidades del producto, así como una serie de artefactos del modelo de diseño que constituyeron la base para dar cumplimiento de manera satisfactoria a la implementación del sistema.
- ✓ Se desarrolló la herramienta propuesta permitiendo dar solución a la problemática planteada.
- ✓ Se verificó y validó la herramienta permitiendo demostrar el correcto funcionamiento de la misma y la disminución del tiempo que deben invertir los usuarios para llevar a cabo la consulta de la información adelantada establecida por la OMI en la AGR.

REFERENCIAS BIBLIOGRÁFICAS

1. **Aduana General de la República.** [En línea] [Citado el: 06 de octubre del 2013.] <http://www.aduana.co.cu/>.
2. **Cobas, Leydis Castellanos.** Modelado de Negocio y Requerimientos de los procesos de recepción y control de los Documentos Complementarios al Despacho Comercial. *Trabajo de Diploma.* La Habana: s.n., 2010.
3. **Más., Kamelia Molina Conde y Yorlen Guirado.** Módulo de gestión de la información asociada a la Aduana en la Ventanilla Única. *Trabajo de Diploma.* La Habana : s.n., 2012.
4. **López, Yakelin Milagro León.** Componente para Gestionar Documentos Complementarios en los Procesos aduaneros del sistema GINA . *Trabajo de Diploma.* 2010.
5. **Internacional, Organización Marítima.** Home Page de la Organización Marítima Internacional. *Organización Marítima Internacional.* [En línea] [Citado el: 6 de noviembre del 2013.] <http://www.imo.org>.
6. **OMI, Documentos Estandarizados.** Guía de implementación de la facilitación del comercio. [En línea] [Citado el: 7 de noviembre del 2013.] <http://tfig.itcilo.org/SP/contents/imo-standardized-documents.htm>.
7. **República, Aduana General de la.** RESOLUCIÓN No. 187-2008. [En línea] 2008. [Citado el: 4 de enero del 2014.] <https://www.aduana.co.cu>.
8. **Arce, Jorge I. Rodríguez.** *La Importancia de las Aduanas en el Comercio Exterior.*2012
9. **Wilfred Ashish, M.G., Kartik Bhatnagar.** *PHP Proyectos Profesionales.* s.l. : Ediciones Anaya, 2002.
10. **Zaninotto., Fabien Potencier y François.** *Symfony la Guía Definitiva.* 2008.
11. **Universidad Carlos III de Madrid.** JAVASCRIPT. [En línea] Universidad Carlos III de Madrid., 26 de Abril del 2012. [Citado el: 22 de Febrero del 2014.] http://perso.wanadoo.es/javascript_12/.
12. **Desarrollo, Comunidad de Desarrollo de Ext JS.Comunidad de.** *Ext JS en Español.*
13. **Oviedo, Escuela de Ingeniería Informática Universidad de.** *Entornos de Desarrollo Integrado.* 2011.
14. **Navarro, J.M.** *Iniciación a Oracle.* 2004.
15. **Reyero, Eusebio.** *Will Web For Food.* 2011. .
16. **García, Joaquín.** IngenieroSoftware. [En línea] 27 de 5 del 2005. [Citado el: 25 de noviembre del 2013.] <http://www.ingenierosoftware.com/analisisydiseno/patrones-diseno.php>.

17. **Larman, Craig.** UML Y PATRONES. *Introducción al análisis orientado a objetos.* PRENTICE HALL, 1999. ISBN 970-17-0261-1.
18. **Ochoa, Sergio.** Introducción a los Patrones . *Diseño y Arquitectura.* 2005.
19. **León, E.** *Visual Paradigm, una herramienta de lo más útil.* 2012.
20. **otros, Hugo Carreon y.** Slideshare. [En línea] [Citado el: 19 de noviembre del 2013.] <http://www.slideshare.net/panchois/métricas-de-software..>
21. **Pressman, Roger S.** *Ingeniería de Software. Un enfoque práctico. 5ta edición.* 2002.
22. **Sánchez Fornaris, Maite y Alcantara Rabí, Dayanis.** Propuesta de una guía de métricas para evaluar el desarrollo de los Sistemas de Información Geográfica. [En línea] 2013. http://vinculando.org/articulos/sociedad_america_latina/propuesta_guía_de_medidas_para_ev.
23. **IEEE.** Metrics. 1991.
24. **Pressman, Roger S.** Ingeniería de Software, un enfoque práctico. 2005.
25. Ecured . Patrones en Symfony. [En línea] [Citado el: 7 de enero del 2014.] http://www.ecured.cu/index.php/Patrones_en_Symfony..
26. **CEIGE, Dpto Aduana.** *Propuesta de un Estándar de Codificación.* La Habana : s.n., 2014.
27. Libros Web. Validadores de Propel. [En línea] [Citado el: 11 de Marzo del 2014.] http://librosweb.es/symfony_formularios/capitulo_13/validadores_de_propel.html.
28. **Pressman, Roger S.** Ingeniería de Software, un enfoque práctico. 2002.
29. Validación y Verificación. . [En línea] <http://www.slideshare.net/josehaar4920/validación-y-verificación-presentation-796097>.
30. **Stephen A. White,** IBM Corporation. Introduction to BPMN. Diciembre 2013.

BIBLIOGRAFÍA

1. **Aduana General de la República.** [En línea] [Citado el: 06 de octubre del 2013.] <http://www.aduana.co.cu/>.
2. **Cobas, Leydis Castellanos.** Modelado de Negocio y Requerimientos de los procesos de recepción y control de los Documentos Complementarios al Despacho Comercial. *Trabajo de Diploma.* La Habana: s.n., 2010.
3. **Más., Kamelia Molina Conde y Yorlen Guirado.** Módulo de gestión de la información asociada a la Aduana en la Ventanilla Única. *Trabajo de Diploma.* La Habana : s.n., 2012.
4. **López, Yakelin Milagro León.** Componente para Gestionar Documentos Complementarios en los Procesos aduaneros del sistema GINA . *Trabajo de Diploma.* 2010.
5. **Internacional, Organización Marítima.** Home Page de la Organización Marítima Internacional. *Organización Marítima Internacional.* [En línea] [Citado el: 6 de noviembre del 2013.] <http://www.imo.org>.
6. **OMI, Documentos Estandarizados.** Guía de implementación de la facilitación del comercio. [En línea] [Citado el: 7 de noviembre del 2013.] <http://tfig.itcilo.org/SP/contents/imo-standardized-documents.htm>.
7. **República, Aduana General de la.** RESOLUCIÓN No. 187-2008. [En línea] 2008. [Citado el: 4 de enero del 2014.] <https://www.aduana.co.cu>.
8. **Arce, Jorge I. Rodríguez.** *La Importancia de las Aduanas en el Comercio Exterior.* 2012
9. **Wilfred Ashish, M.G., Kartik Bhatnagar.** *PHP Proyectos Profesionales.* s.l. : Ediciones Anaya, 2002.
10. **Zaninotto., Fabien Potencier y François.** *Symfony la Guía Definitiva.* 2008.
11. **Universidad Carlos III de Madrid.** JAVASCRIPT. [En línea] Universidad Carlos III de Madrid., 26 de Abril del 2012. [Citado el: 22 de Febrero del 2014.] http://perso.wanadoo.es/javascript_12/.
12. **Desarrollo, Comunidad de Desarrollo de Ext JS.Comunidad de.** *Ext JS en Español.*
13. **Oviedo, Escuela de Ingeniería Informática Universidad de.** *Entornos de Desarrollo Integrado.* 2011.
14. **Navarro, J.M.** *Iniciación a Oracle.* 2004.
15. **Reyero, Eusebio.** *Will Web For Food.* 2011. .
16. **García, Joaquín.** IngenieroSoftware. [En línea] 27 de 5 del 2005. [Citado el: 25 de noviembre del 2013.] <http://www.ingenierosoftware.com/analisydiseno/patrones-diseno.php>.
17. **Larman, Craig.** UML Y PATRONES. *Introducción al análisis orientado a objetos.* PRENTICE HALL, 1999. ISBN 970-17-0261-1.
18. **Ochoa, Sergio.** Introducción a los Patrones . *Diseño y Arquitectura.* 2005.
19. **León, E.** *Visual Paradigm, una herramienta de lo más útil.* 2012.

20. **otros, Hugo Carreon y.** Slideshare. [En línea] [Citado el: 19 de noviembre del 2013.] <http://www.slideshare.net/panchois/métricas-de-software..>
21. **Pressman, Roger S.** *Ingeniería de Software. Un enfoque práctico. 5ta edición.* 2002.
22. **Sánchez Fornaris, Maite y Alcantara Rabí, Dayanis.** Propuesta de una guía de métricas para evaluar el desarrollo de los Sistemas de Información Geográfica. [En línea] 2013. http://vinculando.org/articulos/sociedad_america_latina/propuesta_guía_de_medidas_para_ev.
23. **IEEE.** Metrics. 1991.
24. **Pressman, Roger S.** Ingeniería de Software, un enfoque práctico. 2005.
25. Ecured . Patrones en Symfony. [En línea] [Citado el: 7 de enero del 2014.] http://www.ecured.cu/index.php/Patrones_en_Symfony..
26. **CEIGE, Dpto Aduana.** *Propuesta de un Estándar de Codificación.* La Habana : s.n., 2014.
27. Libros Web. Validadores de Propel. [En línea] [Citado el: 11 de Marzo del 2014.] http://librosweb.es/symfony_formularios/capitulo_13/validadores_de_propel.html.
28. **Pressman, Roger S.** Ingeniería de Software, un enfoque práctico. 2002.
29. Validación y Verificación. . [En línea] <http://www.slideshare.net/josehaar4920/validación-y-verificación-presentation-796097>.
30. **Rodríguez, José Antonio Cobo.** *Línea Base Arquitectónica para el Polo Sistemas Tributarios y de Aduanas.* Ciudad de La Habana : s.n., 2008.
31. **Casella, Lluís-Esteves.** *Estrategias de gestión de los documentos electrónicos en la Administración local: la experiencia del Ajuntament de Girona.* Girona : s.n., 2007.
32. **GSInnova., Grupo soluciones.** Grupo soluciones GSInnova. [En línea] 2009. [Citado el: 24 de 11 de 2013.] <http://www.rational.com.ar/herramientas/rup.html..>
33. **RUMBAUCH, I. J.** *El Proceso Unificado de Desarrollo de Software.* 2000.
34. **Guillot, Carlos Fernández.** ITESO. [En línea] [Citado el: 23 de marzo de 2013.] <http://progra.iteso.mx..>
35. The Home of Groundbreaking Software Quality Management Research. [En línea] www.cigitallabs.com/reso/definiciones/software_testing.html..
36. **Díaz, J. F.** Métodos de Prueba de Programas. . [En línea] http://www.galeon.com/neoprogramadores/met_test.htm.
37. **Juristo, Natalia, Moreno, Ana M. y Vegas, Sira.** *TÉCNICAS DE EVALUACIÓN DE SOFTWARE.* 2006.
38. **Rodríguez, , Gorge I.Arce.** *La Importancia de las Aduanas en el Comercio Exterior.* San Marcos : s.n., 2008.
39. **Guerra, María y de la Fe López, María.** *La automatización de los procesos aduaneros de los países miembros de la CAN en el marco del Proyecto Granadúa (UE-CAN): Caso Venezuela. VIII REUNIÓN DE ECONOMÍA MUNDIAL.* Venezuela : s.n., 2006.

40. **Guerra, B.E.R.** *Tránsito y Transferencia para el Despacho Comercial en la Aduana General de la República de Cuba*. Habana : Universidad de las Ciencias Informáticas., 2011.
41. **Bascón Pantoja, Ernesto.** *El patrón de diseño modelo -vista - controlador (MVC) y su implementación en el Java Swing. Vol. II*. 2004.
42. **Eguíluz Pérez, Javier.** *Introducción a JavaScript*. 2009.
43. **Cobo Rodríguez, José Antonio.** *Línea Base Arquitectónica para el Polo Sistemas Tributarios y de Aduanas*. 2008.
44. **Alfaro, Econ. Félix Murillo.** www.inei.gov.pe . *Herramientas CASE*. [En línea] <http://www.inei.gov.pe/biblioineipub/bancopub/Inf/Lib5103/Libro.pdf>.
45. **Asenjo, Jorge Sánchez.** *Apuntes Completos Sistemas Gestores de Base de Datos*. 2009.
46. **Eguiluz, Javier.** www.symfony.es . [En línea] 2008. <http://www.symfony.es/2008/02/page/2/>.
47. **E.S.Taylor.** *An Interim Report on Engineering Design*. Massachusetts Institute of Technology : s.n., 1959.
48. **BOOCH, Grady, RUMBAUGH, James y JACOBSON, Ivar.** *“El lenguaje unificado de modelado. Manual de referencia”*. s.l. : Addison Wesley, 2000.
49. **Costa, Dolores Costal.** *Introducción al diseño de bases de datos*.
50. **Negro, Ing Pablo Ariel.** *Umbrales para métricas Orientadas a Objeto*. 2008.
51. **Sommerville, Ian.** *Ingeniería del Software (séptima edición)*. Madrid (España) : Pearson Educación, SA., 2005. 84-7829-074-5.
52. **Sahraoui, Houari A., Godin, Robert y Miceli, Thierry.** *Can Metrics Help Bridging the Gap Between the Improvement of OO Design Quality and Its Automation?* Montreal (QC), Canada : s.n.
53. **CEIGE.** *Modelo de Desarrollo de Software*.
54. **Barrios, Johanna Rojas y Emilio.** Grupo ARQUIISOFT. *Grupo ARQUIISOFT*. [En línea] 2007. [Citado el: 26 de abril de 2013.] <http://www.udistrital.edu.co/comunidad/grupos/arquisoft/fileadmin/Estudiantes/Pruebas/HTML%20-%20Pruebas%20de%20software/node28.html>..
55. **Reyero, Eusebio.** Thespacer.net. [En línea] 15 de Enero de 2009. [Citado el: 19 de Febrero de 2014.] <http://wwff.thespacer.net/blog/metodologias-de-diseno>.
56. **Oscar M., F. C., García León, D., & Beltrán Benavides, A.** Un enfoque actual sobre la calidad del software. [En línea] Diciembre de 1995. [Citado el: 25 de Enero de 2014.] http://www.bvs.sld.cu/revistas/aci/vol3_3_95/aci05395.htm.
57. **A. Vizcaino, F. O. García y Caballero, I.** Universidad de Cantabria Fac. de Ciencias. *"Trabajando con visual paragram for uml"*. [En línea] 2008. [Citado el: 25 de febrero de 2014.]
58. **Stephen A. White, . IBM Corporation.** *Introduction to BPMN*. Diciembre 2013.