

**Universidad de las Ciencias Informáticas**  
**Facultad 3**



**Título: Desarrollo del módulo de Configuración y  
Notificación de la Ventanilla Única para el Comercio  
Exterior de Cuba.**

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

**Autores:** Yaimy Navarro Ledea

Boris Atienzar Escobar

**Tutor:** Ing. Julio Cesar Bravo Rodríguez

La Habana, Junio 2012

# *Declaración de Autoría*

---

## **Declaración de autoría**

Declaramos que somos los únicos autores de este trabajo y autorizamos al Centro CEIGE de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmamos la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año 2012.

\_\_\_\_\_  
Yaimy Navarro Ledea

Firma del Autor

\_\_\_\_\_  
Boris Atienzar Escobar

Firma del Autor

\_\_\_\_\_  
Julio César Rodríguez Bravo

Firma del Tutor

\_\_\_\_\_  
Juan Javier Danz Moreno

Firma del Tutor

# *Pensamiento*

---

## **Pensamiento**



*“La inteligencia da bondad, justicia y hermosura;  
Como un ala, levanta el espíritu;  
Como una corona, hace monarca al que la ostenta”*

*José Martí*

# Agradecimientos

---

## Agradecimientos

*A mi mamá, por su paciencia y confianza, porque más que mi madre es mi amiga. Por todos estos años de sacrificio y por enseñarme a luchar por lo que quiero sin importar lo que pase. Por ser la mejor madre del mundo...*

*A mi papá, que ha depositado en mí toda su confianza y que me ha demostrado su amor como a nadie en el mundo. Por su entrega y dedicación, por ser un ejemplo para mí, por enseñarme tantas cosas de la vida. Por ser mi papito lindo...*

*A mi tata, que es mi comedia favorita, que junto a mi mamá y mi papá son las tres personas que más amo en el mundo. Por quererme más que a una hermana y por ese sobrino tan lindo que me ha dado. Por siempre estar ahí para mí...*

*A mi novio Jackson por quererme tanto, y demostrarme que el amor siempre puede hacer maravillas...*

*A mis hermanos Diubel, Venancito, Danny, Ronny y Daimy, por quererme tanto y por todos los momentos lindos que hemos pasado juntos...*

*A mi abuelo Raúl, a mi cuñado Arnel, a mis tíos Luis, Risy, Raida, Raulito, Espe, Nérida, Lumi, Adela y Lidia por ser parte de mi vida, porque juntos formamos una hermosa familia...*

*A mis amigas Tay, Yiya, Baby, Lucre, Lilian, Dailiana, Yusleidis, Leidis, Eylis, Maidelis y Areanne por todas las locuras que hemos hecho juntas, por ayudarme a sentir mejor y por todo lo que representan para mí...*

*A todas las personas que de una forma u otra han aportado un granito de arena en mi vida, a todos, muchas gracias...*

**Yaimy Navarro Ledea**

*A mi madre por ser tan dulce, comprensiva, incansable, intachable, por ser la madre más bella del mundo, por ser incomparable en cualquiera de los sentidos, por ser mi guía y mi bastón, y por guiarme correctamente en la vida...*

*A mi padre por el sacrificio diario de luchar con la familia, por todo el aporte a mi educación y formación, por darme su ejemplo en muchos sentidos para la vida y por su cariño y entrega...*

*A mi hermano, por ser el mejor hermano del mundo y a quien quiero infinitamente, por su forma terca pero afable, por buscar mi apoyo y darme el suyo infinidad de veces, por ser tan incansable y para que siempre logre lo que se proponga...*

*A mis abuelos paternos y maternos sin excepción de ninguno, por luchar conmigo todas las veces que hizo falta y por aportar en mi crianza y educación, a mi tía Leo por dar todo en la vida sin medidas para sus nietos y por ser la viejita más linda de la tierra, y mi abuelo Vera por darme su ejemplo y su amor, por ser trabajador incansable en la vida, por ser incondicional con sus nietos...*

*A mis tíos Tony y Lizy y mis primos Relo y Toño por ser parte indispensable de esta gran familia, por darme su apoyo y cariño invaluable día a día, a quienes voy a estar eterna e incansablemente agradecido...*

*A toda mi familia por creer en lo que hago y aportar en mi vida diariamente*

*A mi hermanito Adriél por ser un excelente amigo, a Rosalina por su incondicionalidad infinita a quien quiero muchísimo, a Vicente y Cristina por entrar en la familia de forma insospechada y por brindarme su hospitalidad y cariño sin medidas, a mi titi Odalis por ser tan linda conmigo y por brindarme su afecto, a Eylis, Leanet y Manuel, por la gratitud que siempre tendré para con ellos...*

**Boris Atienzar Escobar**

# Dedicatoria

---

## Dedicatoria

...A mi mimi linda por siempre estar ahí para mí, por estar con los brazos abiertos cuando los necesito, por toda la dedicación y sacrificio. Por ser mi amiga y confidente, por amarme tanto, por apoyarme y enseñarme a ser yo misma. Porque a tu lado siento que mi vida es perfecta...Te amo mucho.

...A mi papito lindo, por ser un papá perfecto, por ser el mejor del mundo sin miedo a equivocarme y por la dedicación y entrega. Por regalarme la dicha de siempre tenerlo a mi lado, por su confianza y sacrificio, por dejarlo todo atrás por estar conmigo, por todo lo que es y por lo que soy gracias a él...Te amo mucho.

...A mi tata loca, porque este título de ingeniera es para las dos, porque es mi mitad, porque es mi segunda madre y mi hermana perfecta. Por todas esas cosas locas que se le ocurren para hacerme reír. Por ser parte de mi triángulo favorito, por cuidar de mami y papi cuando yo no estoy, por quererme tanto como yo a ella...Te amo mucho.

...A esas dos personas tan importantes para mí y que ya no están conmigo, a mi abuela Virginia y mi tío Paulino, a ellos que me quisieron tanto y que significan mucho en mi vida. Mi corazón se oprime al recordar que ya no están conmigo, pero me enorgullezco de haber tenido dos personas como ellos formando parte de mi educación...Los amo.

*Yaimy Navarro Ledea*

A mi madre, a mi padre, a mi hermano, a mis abuelos, a mis tíos, a mis amigos y compañeros de escuela, a todos ellos por creer en lo que hago y por esperar siempre lo mejor...

*Boris Atienzar Escobar*

## Resumen

Controlar las operaciones comerciales de forma centralizada en la Ventanilla Única del Comercio Exterior es una tarea de prioridad que permite la obtención de información tanto a personas jurídicas como no jurídicas, a petición de las agencias emisoras de permisos de exportación e importación. Como parte del proceso de informatización de la mayor parte de los procesos por parte de la Aduana General de la República surgió la necesidad de concebir un único punto de entrada y salida de información digital estandarizada para las operaciones comerciales bajo regímenes aduaneros. De ahí el sentido de este trabajo de diseñar e implementar el módulo de Configuración para la Notificación y Auditoría de la Ventanilla Única de Comercio Exterior de Cuba.

Con el fin de llevar a cabo dicha acción se realizó un estudio del arte, pertinente para conocer hasta qué punto debía profundizarse en el tema con el fin de cumplimentar la tarea encomendada. De igual modo se valoraron metodologías y herramientas que posibilitaran la obtención de una adecuada solución y estrategia viable para su desarrollo.

Guiados por el Modelo de Desarrollo que utiliza como metodología el Departamento de Soluciones para la Aduana y utilizando las herramientas que este implica, se llevó a cabo el modelado de los procesos del negocio para el manejo de las notificaciones y del mismo modo se diseñó, implementó y probó el módulo resultante, que de manera conjunta con la Aduana General de la República disminuye el intercambio de información en copia dura y una serie de engorrosos trámites ante una transacción comercial.

**Palabras Clave:** operaciones comerciales, centralizada, información, personas jurídicas, agencias, exportación, importación, informatización, procesos, metodología.

# *Tabla de Contenidos*

---

## **Tabla de contenidos**

<b>AGRADECIMIENTOS</b> .....	III
<b>DEDICATORIA</b> .....	IV
<b>RESUMEN</b> .....	V
<b>TABLA DE CONTENIDOS</b> .....	VI
<b>INTRODUCCIÓN</b> .....	8
<b>CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA</b> .....	14
<b>1.1 Introducción del Capítulo</b> .....	14
<b>1.2 La Aduana General de la República de Cuba</b> .....	14
<b>1.3 Ventanilla Única de Comercio Exterior</b> .....	14
<b>1.4 Sistemas semejantes en el mundo</b> .....	15
<b>1.4.1 Aspectos Funcionales de los Sistemas de Ventanilla Única del mundo</b> .....	15
<b>1.4.2 Semejanzas con el concepto de Ventanilla Única para Cuba</b> .....	17
<b>1.5 Herramientas y tecnologías utilizadas</b> .....	17
<b>1.6 Metodología de desarrollo</b> .....	21
<b>1.7 Análisis del Sistema Ventanilla Única para el Comercio Exterior de Cuba</b> .....	24
<b>1.8 Diseño del Sistema de Ventanilla Única para el Comercio Exterior de Cuba</b> .....	28
<b>1.9 Conclusiones del Capítulo</b> .....	31
<b>CAPÍTULO 2: ANÁLISIS Y DISEÑO</b> .....	32
<b>2.1 Introducción del Capítulo</b> .....	32
<b>2.2 Solución Propuesta</b> .....	32
<b>2.3 Análisis de la solución propuesta</b> .....	33
<b>2.3.1 Modelado de los procesos del Negocio del Sistema</b> .....	33
<b>2.3.2 Modelo Conceptual del Sistema</b> .....	36
<b>2.3.3 Técnicas para la captura de los Requisitos</b> .....	37
<b>2.3.4 Requisitos del Sistema</b> .....	38

# *Tabla de Contenidos*

---

2.3.5	Técnicas para la Validación de los Requisitos del Sistema .....	42
2.4	Diseño de la solución propuesta .....	43
2.4.1	Diagrama de Paquetes del Sistema .....	43
2.4.2	Diagrama de Secuencia orientado a Actividades del Sistema .....	44
2.4.3	Diagrama Entidad-Relación del Sistema .....	46
2.4.4	Métricas para la validación del Diseño del Sistema .....	48
2.5	Conclusiones del Capítulo .....	53
<b>CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA .....</b>		<b>54</b>
3.1	Introducción del Capítulo .....	54
3.2	Diagrama de Despliegue .....	54
3.3	Diagrama de Componentes .....	55
3.4	Aspectos relevantes del flujo Guardar Notificación .....	56
3.5	Patrones de Diseño en la Codificación del Sistema .....	61
3.6	Validación de la Solución Propuesta .....	63
3.6.1	Pruebas Unitarias .....	64
3.6.2	Diseño de Casos de Pruebas para Pruebas de caja negra del Sistema .....	65
3.6.3	Ejecución de pruebas de resistencia y rendimiento .....	70
3.7	Conclusiones del Capítulo .....	73
<b>CONCLUSIONES .....</b>		<b>74</b>
<b>RECOMENDACIONES .....</b>		<b>75</b>
<b>REFERENCIAS BIBLIOGRÁFICAS .....</b>		<b>76</b>
<b>GLOSARIO DE TÉRMINOS .....</b>		<b>78</b>



## **Introducción**

La ampliación de mercados e integración comercial entre países tiene un impacto positivo en el actual ritmo de crecimiento del comercio exterior que se desarrolla en todo el mundo. Por ello, asegurar su eficiencia y sostenibilidad futura exige una mayor calidad, transparencia, eficiencia y rapidez en los procesos relacionados con los controles que ejercen las entidades del Estado, con el objetivo de reducir la diversidad y dispersión de los trámites a través de los cuales se permite obtener controles integrados y dar una respuesta rápida y transparente a las personas y entidades implicadas en la importación y exportación en los mercados internacionales; todo esto minimizando costo y tiempo en dichas operaciones.

De igual manera con el fin de cumplimentar el proceso de informatización surge un nuevo concepto, Ventanilla Única de Comercio Exterior (VUCE). Con este concepto se desea establecer un vínculo directo entre las Aduanas y las personas que intervengan en una transacción comercial. El sistema VUCE se implementa enfocado a los beneficios que puede traer consigo para las agencias y comunidades protagonistas.

El país que implemente una VUCE permitirá a los usuarios realizar, por medios electrónicos, los trámites para la obtención de los permisos, certificaciones, licencias y demás autorizaciones exigidas por las autoridades competentes para el ingreso, tránsito o salida de mercancías que lo requieran. Además puede suceder que en los países donde no exista implementado un sistema de VUCE se sobrecargue a la Aduana con el trabajo de recepción y validación de información digital, así como emitir permisos que le corresponden a otras instancias del gobierno.

La Ventanilla Única es “una herramienta que permite la entrega de información estandarizada a una única entidad para el cumplimiento de todos los requerimientos relacionados con la importación, exportación y tránsito. Si la información es electrónica, entonces los datos deben ser enviados una única vez.” (1)

## *Introducción*

---

La producción de aplicaciones informáticas, como una de las principales metas que tiene Cuba ante el inminente desarrollo de las Tecnologías de la Información y las Comunicaciones (TIC), es un paso decisivo en el afán de lograr la independencia tecnológica y contribuir al desarrollo del país. La Universidad de las Ciencias Informáticas (UCI) funge como un pilar importante en dicho proceso, brindando sus servicios a empresas nacionales e internacionales.

Dentro del proceso de informatización, se ve inmersa una entidad de punta en la economía cubana, la Aduana General de la República (AGR), la cual involucró a la UCI con el fin de fortalecer dicho proceso desde el curso 2004-2005. A medida que se fue ganando madurez y fue evolucionando la estructura productiva de la universidad, se crearon estructuras que agruparon y gestionaron la informatización de diversas actividades de desarrollo para la AGR.

La AGR, encargada del control aduanero para trámites, tránsitos, depósitos, clientes, equipajes y regulaciones especiales, fue creada en febrero 5 de 1963 y ejerce sus derechos sobre los productos extranjeros que entran y salen del país. Tan importante entidad, forma parte del sistema de control estatal, de vital importancia para la erradicación del fraude y el contrabando en el sistema en que juega su papel de reguladora. (2)

La Aduana cubana utiliza actualmente el Sistema Único de Aduanas (SUA), sistema que le permite realizar de manera automática la mayor parte de los procedimientos que se realizan diariamente en esta entidad. Este cuenta con un sistema para la recepción de documentos digitales que se encarga de procesarlos mediante la entrega del fichero por parte de un mediador de la AGR al GINA, cuando lo correcto sería que los usuarios externos a la Aduana sean los responsables de la información emitida.

Del mismo modo nuestro país presenta muchos problemas de costo y tiempo en temas comerciales, más aun cuando se trata de importaciones o exportaciones de mercancías. La falta de procesos integrados o simplificados y los excesivos tiempos de demora para realizar los trámites comerciales son algunos de los problemas que se pueden encontrar.

En la actualidad se realiza la recepción de documentos en formato duro, esto implica tener que realizar esta acción tantas veces como entidades involucradas requieran de la información que pudiera ser común entre las mismas. Del mismo modo, es de gran interés el envío de

## *Introducción*

---

notificaciones a cada una de las personas jurídicas que intervengan en las transacciones comerciales de modo automatizado. Hoy en día esta actividad se realiza mediante la comunicación indirecta sin toda la factibilidad que debería tener dicho proceso.

El sistema que se quiere desarrollar debe estar orientado a los documentos digitales que se reciban, lo que implica que la configuración, en cuanto a formato de información, así como el flujo para su validación, debe ser común para todos; de manera tal que no sea necesario programar cada flujo de cada documento de manera independiente y que se pueda modificar en cualquier momento el formato de cada documento a recibir.

De igual manera se debe garantizar el manejo de los documentos que inician la declaración para la importación y exportación de mercancías en la nación. Tales documentos nombrados declaraciones de mercancías, deberán permitir la asociación de otros documentos complementarios con el fin de culminar el proceso de configuración, donde dicha labor es considerado el paso final para poder efectuar la recepción de documentos digitales que transitarán por el flujo de configuración que dará un orden secuencial al manejo de los mismos. Esta investigación surge con el fin de dar solución a la problemática anterior garantizando que a través de un correcto análisis, diseño e implementación se sienten las bases para solventar lo anteriormente planteado.

Una vez planteada la anterior situación problemática se formula el siguiente **problema a resolver**: la recepción de documentos en formato duro dificulta los trámites para realizar la importación y exportación de mercancías en el país así como el mecanismo de aviso sobre las operaciones que se deben efectuar para el procesamiento de los documentos.

Con el fin de darle solución al problema anterior se define como **objeto de estudio**: la gestión de documentos digitales y manejo de notificaciones en sistemas informáticos; centrando su **campo de acción en**: los Procesos de Recepción de documentos digitales y envío de Notificaciones en el sistema de Ventanilla Única para el Comercio Exterior de Cuba.

Con lo anteriormente planteado se puede defender la idea siguiente; con el módulo de Configuración y Notificación se propiciará un viable proceso de recepción de documentos y

# *Introducción*

---

notificación mediante la entrada de información una única vez para cualquier trámite a realizar durante una transacción comercial.

Se define del mismo modo como **objetivo general**: Desarrollar un módulo que permita configurar la recepción de documentos y notificar a los usuarios de la Ventanilla Única para facilitar el comercio en Cuba.

Del objetivo general se derivan los siguientes objetivos específicos:

1. Elaborar el marco teórico relativo a las soluciones de software que gestionan la recepción de documentos digitales y notificación de operaciones comerciales.
2. Identificar, describir y validar los requerimientos asociados a los procesos.
3. Modelar los procesos de Negocio del módulo de Configuración y Notificación de la Ventanilla Única.
4. Modelar el Diseño del módulo de Configuración y Notificación de la Ventanilla Única.
5. Implementar el módulo de Configuración y Notificación de la Ventanilla Única.
6. Realizar pruebas al módulo de Configuración y Notificación de la Ventanilla Única.

Para abarcar con mayor profundidad el cumplimiento de los objetivos específicos se definieron las siguientes **tareas de la investigación**:

1. Realizar un estudio comparativo entre los sistemas de Ventanilla Única para identificar posibles fortalezas para la solución.
2. Realizar diagramas de modelo de negocio.
3. Realizar descripción de los procesos de negocio.
4. Realizar modelo conceptual.
5. Realizar descripción de los requisitos.
6. Describir reglas de negocio.
7. Diseñar modelos de datos.
8. Diseñar la solución descrita.
9. Implementar la solución modelada.
10. Diseño de CU de prueba.

# Introducción

---

11. Manual de usuario.
12. Verificar la solución mediante pruebas funcionales.
13. Validar la solución por el cliente.

Con el fin de proporcionar un mejor desarrollo del presente trabajo, durante la investigación, se utilizarán varios **métodos científicos** que fundamentan la investigación realizada:

## **Teóricos:**

1. **Experimental:** Como rama del método empírico, se utiliza con el objetivo de crear las condiciones necesarias para evaluar las características fundamentales de cada elemento estudiado. (3)
2. **Histórico-lógico:** Con el objetivo de realizar un estudio del estado del arte del tema a investigar. De esta manera se puede conocer acerca de la existencia y características de sistemas de este tipo que hayan sido desarrollados anteriormente. (3)
3. **Sintético:** Con el objetivo de relacionar hechos para formular teorías que unifican elementos dispersos en una nueva teoría. (3)
4. **Hipotético-Deductivo:** Con el objetivo de verificar las hipótesis.
5. **Modelación:** Con el objetivo de desarrollar modelos estrechamente relacionados con el módulo. Este método tiene como objetivo fundamental hacer una reproducción simplificada de la realidad.
6. **Analítico-Sintético:** Con el objetivo de posibilitar el procesamiento de toda la información y poder sintetizar y diferenciar cada una de ellas, enfocada hacia la investigación.
7. **Sistémico:** Con el objetivo de facilitar la integración de cada uno de los elementos desarrollados, en la conformación de los módulos.

## **Empíricos:**

# Introducción

---

1. **Entrevista:** Este método es uno de los más importantes ya que permite una relación directa con el cliente; tiene como objetivo captar los detalles no explícitos en la documentación a consultar, además de precisar cuáles son las necesidades y demandas del cliente en función de la solución al problema que es presentado. Se utiliza con el objetivo de contar con un medio para el conocimiento cualitativo de los procesos del negocio.
2. **Observación (Externo-Incluido):** Con el objetivo de posibilitar la percepción planificada dirigida a un fin y se realiza de forma consciente y orientada a un objetivo determinado.

La estructura adoptada en este trabajo de diploma es la siguiente:

El **capítulo número uno** abarcará lo relacionado a la Fundamentación Teórica de la investigación. En el mismo se realizará un estudio del arte sobre sistemas a nivel nacional e internacional, que sirven de guía para el desarrollo de este trabajo. Por otro lado se tratarán aspectos de interés para la comprensión del sistema que se desea modelar, los conceptos más importantes, tecnologías, técnicas y tendencias actuales, entre otros aspectos de igual connotación.

El **capítulo número dos** tratará acerca del Análisis y el Diseño del sistema. Hará énfasis en el modelado del negocio, en la descripción de los requisitos, en el modelo conceptual, en diagramas de clases, diagramas de paquetes, de entidad de relación para la base de datos concerniente, así como diagramas de componentes y de secuencia generados como artefactos de la solución.

El **capítulo número tres** tratará acerca de la Implementación y la realización de la Validación de la solución propuesta mediante la evaluación del software, con el objetivo de materializar la solución propuesta y detectar posibles no conformidades cumpliendo de manera satisfactoria con los objetivos planteados.

## **Capítulo 1: Fundamentación Teórica**

### **1.1 Introducción del Capítulo**

En el presente capítulo se describirán los elementos principales que sirven de fundamentación para el contenido de la investigación. Se realizará un estudio profundo de las actuales soluciones informáticas para las ventanillas únicas a nivel internacional, así como algunas de las técnicas de captura y validación de requisitos. Se fundamentará además, acerca de aspectos y estilos arquitectónicos, herramientas, y lenguajes y notación de modelado para el desarrollo del producto.

### **1.2 La Aduana General de la República de Cuba**

La Aduana General de la República de Cuba suscrita desde 1995 al Convenio Internacional para la simplificación y armonización de los Regímenes Aduaneros y ratificada en 2009 para el Convenio de Kyoto Revisado, ha adoptado un grupo de medidas para la facilitación y rapidez del despacho mercantil. Este órgano ha sido influenciado por una serie de aspectos en la modernización, como lo son, el desarrollo de las TIC y por consiguiente su amplia red de datos, propiciando así la automatización de la mayoría de los procesos administrativos y de control, haciendo uso del software libre y del ingenio de la propia institución para su desarrollo. (2)

### **1.3 Ventanilla Única de Comercio Exterior**

La Ventanilla Única de Comercio Exterior es, para su buen entendimiento, desde el punto de vista de un ciudadano común, un único punto mediante el cual puede hallarse toda la información necesaria referida al comercio. Técnicamente, está definido como un sistema integrado que permite a las partes involucradas en el comercio exterior y el transporte internacional gestionar a través de medios electrónicos y por un solo punto, los trámites requeridos por las entidades de control pertinentes para el tránsito, ingreso o salida del territorio nacional de mercancías. (4)

Dicho sistema tiene como objetivo primordial, integrar y mejorar los procesos comerciales que naturalmente resultaban tediosos de hacer, esto mediante la simplificación de cada uno de ellos ante las entidades pertinentes. Por otro lado, se propone brindar de manera simple y

## Capítulo 1: Fundamentación Teórica

rápida toda clase de información al cliente, entiéndase como cliente a toda clase de persona jurídica que intervenga en una transacción comercial. Por último y de gran importancia, se propone brindar seguridad jurídica acerca de la información de modo certero y vinculado a las partes involucradas con el comercio exterior. (4)

“Es una herramienta que permite la entrega de información estandarizada a una única entidad para el cumplimiento de todos los requerimientos relacionados con la importación, exportación y tránsito. Si la información es electrónica, entonces los datos deben ser enviados una única vez.” (1)

### 1.4 Sistemas semejantes en el mundo

Actualmente el desarrollo de los procesos en la sociedad evoluciona constantemente, ejemplo de esto es el avance de los sistemas de Ventanilla Única en el mundo. Muchos son los países que se encuentran a la vanguardia en este aspecto. A continuación se exponen algunos de los elementos fundamentales que sirven de guía para el presente trabajo sobre dichos sistemas.

#### 1.4.1 Aspectos Funcionales de los Sistemas de Ventanilla Única del mundo

Aspectos	Servicios	Formato Documentos	Actores	Seguridad
<b>Perú</b>	Recepción de documentos digitales. Obtención de permisos, certificados y licencias. Trámites para transporte de carga.	Documento de texto.	Exportadores. Importadores.	Roles de usuario. Firma Electrónica.
<b>México</b>	Pedimento para trámite de mercancías. Recepción y validación de documentos anexos para el comercio.	Documento de texto.	Agentes navieros y aduanales. Exportadores. Importadores.	Roles de usuario. Firma Electrónica.



## Capítulo 1: Fundamentación Teórica

<b>Corea</b>	<p>Procesamiento de documentos para la liquidación de importación-exportación.</p> <p>Transferencia de mensajes.</p> <p>Manejo de tarifas para los diversos servicios.</p> <p>Proveedor de servicio de aplicaciones<sup>1</sup>.</p>	<p>Documento de texto.</p> <p>XML.</p>	<p>Autoridades de gobierno.</p> <p>Banqueros.</p> <p>Compañías de embarco y de logística.</p>	<p>Roles de usuario.</p> <p>Firma Electrónica.</p>
<b>Suecia</b>	<p>Recepción de documentos para la importación, exportación y el tránsito comercial.</p> <p>Mensajes electrónicos.</p> <p>Tarifa Aduanera.</p>	<p>Documento de texto.</p> <p>EDIFACT<sup>2</sup>.</p> <p>XML.</p>	<p>Exportadores.</p> <p>Importadores.</p> <p>Agencias de impuesto y comercio.</p> <p>Oficina de la Agricultura.</p>	<p>Roles de usuario.</p>
<b>Ecuador</b>	<p>Recepción de documentos digitales para el comercio.</p> <p>Pago de derechos aduaneros.</p> <p>Envío de información de forma instantánea.</p>	<p>Documento de texto.</p> <p>XML.</p>	<p>Exportadores.</p> <p>Importadores.</p>	<p>Roles de usuario.</p> <p>Firma digital.</p> <p>Certificado de autenticación.</p>

<sup>1</sup> **Proveedor de Servicio de Aplicaciones:** ASP, acceso a una aplicación de alta demanda mediante la web

<sup>2</sup> **EDIFACT:** Estándar para intercambio de datos electrónicos por sus siglas, Intercambio de datos electrónicos para la administración, el comercio y el transporte

### **1.4.2 Semejanzas con el concepto de Ventanilla Única para Cuba**

El estudio de los sistemas utilizados en el resto del mundo, ofrece al proyecto VUCE la posibilidad de nutrirse de ideas que han sido ya pensadas y probadas por países con un alto nivel de experiencia en el asunto como Corea, Suecia y Perú, teniendo en cuenta además la medida en que cada uno revela parte de la lógica de su funcionamiento a quienes no interactúan directamente con los mismos. Dentro de los aspectos comunes se encuentra el envío de notificaciones de manera diferenciada a las entidades y personas involucradas en el negocio de forma instantánea mediante la creación de un módulo de notificaciones. En este módulo sería factible manejar el envío de información relacionada con la empresa o las mercancías que son de interés para los clientes, como las actualizaciones sobre los productos que son de interés específico y el estado de los documentos que son procesados en el portal de la Ventanilla. Por otro lado, destaca la definición de documentos utilizando el estándar XML y en algunos casos EDIFACT con el objetivo de usar un lenguaje estándar para el intercambio de información, así como la posibilidad de hacerlo mediante formularios. Por último, se encuentra la consideración de especializar tarifas para el manejo de mensajes dependiendo del uso del mismo, así como del envío de documentos a partir de su tamaño. Para la confección de una Ventanilla Única de Comercio Exterior en Cuba se hace necesaria la implementación de un producto desde cero, considerando las características más convenientes de los sistemas analizados anteriormente. Esta solución tendrá particularidades como el necesario uso del software libre con la idea de apoyar la estrategia del país, de eliminar el software privativo a lo largo del archipiélago.

### **1.5 Herramientas y tecnologías utilizadas**

Para el desarrollo de un sistema de Ventanilla Única es necesario seleccionar las herramientas y tecnologías de desarrollo que cumplan con las necesidades requeridas. A continuación se abordan las que fueron utilizadas.

#### **Herramienta CASE para el modelado: Visual Paradigm for UML**

Visual Paradigm es una herramienta destinada a aumentar la productividad en el desarrollo de software, reduciendo el coste de las mismas en términos de tiempo y de dinero. Dentro de

las ventajas que ofrece se encuentra la de aumentar la calidad del software, partiendo de la posibilidad que brinda efectuar el desarrollo, la documentación, la generación de código y gestión del proyecto de manera automática. Está orientada a reducir el tiempo y el coste del desarrollo, así como mejorar la planificación del proyecto sin necesidad de utilizar más de una herramienta. Además, permite modelar todos los tipos de diagramas de clases, generar código inverso, generar documentación y código desde diagramas.

## **Lenguaje de modelado: UML 2.0**

El lenguaje Unificado de Modelado (UML) es un lenguaje de modelado de sistemas de software y es en la actualidad el más conocido y utilizado. Su función es la de visualizar, especificar, construir y documentar un sistema. Se utiliza para definir un sistema, para detallar los artefactos en el sistema y para documentar y construir, en otras palabras, para describir el modelo.

Es válido decir que UML no está concebido para la programación estructurada puesto que en UML solo se diagrama la realidad a partir de requisitos.

## **Entorno de Desarrollo Integrado: NetBeans**

Un IDE es un entorno de programación resumido a una única aplicación, capaz de fungir como editor de código, compilador, depurador y constructor de interfaz gráfica. Puede ser utilizado para desarrollar en la mayoría de los lenguajes de programación como PHP, Java, C++, C#, etc.

NetBeans, es un proyecto de código abierto que está formado por un conjunto de componentes llamados módulos y está hecho principalmente para Java, aunque ofrece muchas ventajas entre las que se encuentran:

- Creación de proyectos de PHP.
- Integración con sistemas de control de versiones.
- Integración con Bases de Datos como Oracle, PostgreSQL y MySQL.
- Depuración PHP.
- Soporte para Symfony.

# Capítulo 1: Fundamentación Teórica

---

- Soporte para AJAX.

## **Sistema Gestor de Base de Datos: Oracle**

Un sistema gestor de base de datos (SGBD) está destinado a servir de interfaz entre una base de datos, el usuario y las aplicaciones que con ella se comunican. Su principal propósito es el de manejar de manera clara y ordenada un conjunto de datos que pueden ser información relevante para una organización.

Entre los principales objetivos de un SGBD se encuentran:

- Independencia: Brindan la posibilidad de modificar el esquema de datos sin tener que realizar cambios en las aplicaciones.
- Seguridad: Permiten establecer permisos a usuarios y grupos de usuarios asegurando así información sensible y de gran valor.
- Consistencia: Una BD (base de datos), no queda exenta de almacenar información repetida. Un SGBD cuenta con herramientas que impiden que tenga lugar este tipo de error tan común.
- Abstracción: Ahorran a los usuarios detalles acerca del almacenamiento físico de los datos haciendo invisible cualquier aspecto referente a las condiciones en que se encuentran los mismos.

## **BPMN**

Business Process Modeling Notation (BPMN) es una notación gráfica estandarizada que permite el modelado de procesos de negocio, proporciona un lenguaje común para que las partes involucradas puedan comunicar los procesos de forma clara, completa y eficiente. De esta forma BPMN define la notación y semántica de un Diagrama de Procesos de Negocio (Business Process Diagram, BPD). (11)

Inicialmente BPMN fue desarrollada por la organización BPMI<sup>3</sup>, y es actualmente mantenida por el OMG<sup>4</sup>, luego de la fusión de las dos organizaciones en el año 2005. (12)

---

<sup>3</sup> **BPMI**: Iniciativa para el manejo de procesos de negocio, por sus siglas en inglés: Business Process Management Initiative.

<sup>4</sup> **OMG**: Grupo de manejo de objetos, por sus iniciales en inglés: Object Management Group.

# Capítulo 1: Fundamentación Teórica

---

## Lenguaje de Programación: PHP

Por su parte, un lenguaje de programación es un idioma artificial diseñado para expresar computaciones que pueden ser llevadas a cabo por máquinas como las computadoras. Pueden usarse para crear programas que controlen el comportamiento físico y lógico de una máquina, para expresar algoritmos con precisión, o como modo de comunicación humana. Está formado por un conjunto de reglas y símbolos sintácticos y semánticos que definen su estructura y el significado de sus elementos y expresiones. (13)

Un programa escrito en un lenguaje de programación necesita pasar por un proceso de compilación, interpretación o intermedio, es decir, ser traducido al lenguaje de máquina para que pueda ser ejecutado por el ordenador. (14)

PHP fue diseñado originalmente para la creación de páginas web dinámicas. Se usa principalmente para la interpretación del lado del servidor pero en la actualidad gracias a algunas bibliotecas gráficas como QT<sup>5</sup> puede ser utilizado para la creación de aplicaciones con interfaz gráfica. Gran parte de su sintaxis ha sido tomada de C, Java y Perl, además de características propias del lenguaje y el hecho de que sea incrustado dentro de HTML como lenguaje de script lo hace muy útil para la creación dinámica de páginas. (15). El lenguaje utilizado para desarrollar la VUCE es PHP en su versión 5.3.

## Framework de desarrollo: Symfony

Symfony es un Framework de trabajo orientado a servir al MVC<sup>6</sup> desarrollado totalmente en PHP 5. En su composición se separa la lógica del negocio, la lógica del servidor y la presentación web en tres elementos muy bien interrelacionados. Las clases de la capa del modelo se generan automáticamente, en función de la estructura de datos de la aplicación. La librería de Propel se encarga de esta generación automática, ya que crea el esqueleto o estructura básica de las clases y genera automáticamente el código necesario. Ha sido probado en numerosos proyectos reales y se utiliza en sitios web de comercio electrónico de

---

<sup>5</sup> **QT:** Biblioteca multiplataforma para desarrollar interfaces gráficas de usuario.

<sup>6</sup> **MVC:** patrón arquitectónico Modelo Vista Controlador.

# Capítulo 1: Fundamentación Teórica

---

primer nivel. Su carácter libre y multiplataforma lo hacen uno de los Framework de PHP más usados en el mundo. (16)

La utilización de un Framework de PHP garantiza la generación automática de parte del código utilizado durante la implementación, lo cual agiliza el proceso de producción de software y debido a su alto nivel de flexibilidad permite el desarrollo de aplicaciones específicas para cada uno de los sistemas desarrollados por el departamento. (17)

La elección de Symfony viene dada por la necesidad de mantener la línea de desarrollo del departamento y de utilizar además un Framework que se adecue a la necesidad de desarrollo del mismo. Symfony, permite una configuración en cascada que posibilita que cada uno de sus elementos sea adaptable a las necesidades de los desarrolladores lo cual lo hace altamente flexible. Además contiene una alta gama de funcionalidades, pero quizás su mayor poder está en la posibilidad de ampliarlas de forma ilimitada mediante la utilización de Plugins, entre los cuales destaca la publicación de funciones desarrolladas como Servicios Web.

## **1.6 Metodología de desarrollo**

Una metodología de desarrollo es un marco de trabajo usado para estructurar, planificar y controlar el proceso de desarrollo en sistemas de información. (18). Consiste en una filosofía de desarrollo de programas de computación con el enfoque del proceso de desarrollo, que cuenta con modelos y métodos para apoyarlo.

Producto de la complejidad de la solución para el Departamento de Soluciones para la Aduana, se hizo necesaria la utilización de un proceso de desarrollo que guiara el desarrollo del mismo dentro de un mar de soluciones e ideas que buscan converger en una única solución limpia y capaz de resolver cada problema con la calidad necesaria en un tiempo planificado.

## **Proceso Racional Unificado (RUP)**

El Proceso Racional Unificado conocido mayormente por sus siglas RUP, es un proceso de desarrollo de software afamado por ser uno de los más utilizados en el análisis, implementación y documentación de sistemas orientados a objetos. Esto se debe a que RUP no cuenta con una serie de pasos establecidos de modo esquemático, cuando en cambio obedece a las necesidades de cada organización.

# Capítulo 1: Fundamentación Teórica

---

Entre sus principales características se pueden mencionar al desarrollo iterativo, lo cual permite obtener un producto final en cada una de estas iteraciones mientras se va obteniendo su acabado de forma incremental. RUP además, es centrado en la arquitectura y es dirigido por casos de uso, partiendo de la posibilidad que brindan estos últimos de orientar el diseño, la implementación, las pruebas y la estimación de tiempo y esfuerzo, además del tiempo de desarrollo.

Se podrán alcanzar resultados predecibles unificando el equipo con procesos comunes que optimicen la comunicación y creen un entendimiento común para todas las tareas, responsabilidades y artefactos. Desde un único sitio web centralizado de intercambio, el Software Rational, las plataformas, herramientas y expertos de dominios proveen los componentes de proceso necesarios para el éxito. (19)

## **SCRUM**

SCRUM es una metodología para la gestión y el desarrollo basado en un proceso iterativo e incremental, utilizado comúnmente en entornos basados en el desarrollo ágil de software. Su principal objetivo es elevar al máximo la productividad de un equipo reduciendo a niveles muy bajos la burocracia y actividades no orientadas a producir software que funcionen, produciendo resultados en periodos muy breves de tiempos a través de iteraciones. La recogida de requisitos para crear un producto se realiza teniendo en cuenta la visión del cliente y del usuario. Al final el objetivo es entregar algo que funcione para que el usuario pueda probarlo y se puedan introducir los cambios necesarios antes de que sea demasiado tarde. (20)

SCRUM propone lanzar un incremento de software con calidad de entregable cada cierto tiempo con un máximo de 30 días y un mínimo de 15, a lo que llaman sprint y donde no puede haber cambios de requisitos, lo cual indica que los requisitos están congelados en este intervalo de tiempo. Lo que hace a esta metodología particular, es el hecho de que adoptan una convicción sobre que el problema no puede ser del todo entendido o definido de una sola vez, dándose la posibilidad de existir cambios de forma impredecible, y es por eso que el equipo de desarrollo debe tener la capacidad de responder con agilidad al surgimiento de requisitos emergentes. (20)

### XP

El término XP fue por primera vez tratado en el libro *Extreme Programming Explained: Embrace Change* escrito por Kent Becket en 1999 y es una de las metodologías más destacadas y debe sus siglas a Programación Extrema y se diferencia de las tecnologías tradicionales principalmente en que hace más énfasis en la adaptabilidad que en la previsibilidad. Los defensores de esta metodología consideran que los cambios de requisitos sobre la marcha son un aspecto natural, inevitable e incluso deseable del desarrollo de proyectos. Se considera que es capaz de adaptarse a los cambios de requisitos en cualquier punto de la vida del proyecto, lo cual supera a intentar definir todos los requisitos al comienzo del proyecto e invertir esfuerzo en controlar los cambios de requisitos.

### Modelo de Desarrollo para el Departamento de Soluciones de la Aduana

En el Departamento de Soluciones para la Aduana se hace uso de un Modelo de Desarrollo que cumple con las especificaciones y necesidades del departamento, el cual incluye los flujos de trabajo, encuentros y reuniones básicas, los roles y sus responsabilidades, así como la interacción entre ellos (21).

En la fase de análisis, el modelo utilizado genera 8 artefactos, mencionados a continuación:

1. Mapa de Procesos.
2. Modelo Conceptual.
3. Descripción y Modelado de los Procesos del Negocio.
4. Especificación de los Requisitos del Sistema.
5. Diccionario de Datos.
6. Prototipos de IU<sup>7</sup> en AXURE<sup>8</sup>.
7. Manual de usuario.
8. Salidas del Sistema.

---

<sup>7</sup> **IU:** Interfaz de Usuario.

<sup>8</sup> **AXURE:** RP pro, herramienta de software para la construcción de prototipos y especificación de interfaz gráfica orientado tanto a aplicaciones de escritorio como a aplicaciones web.



# Capítulo 1: Fundamentación Teórica

---

Uno de los principales aspectos a resaltar del Modelo dentro de la fase del diseño, lo constituye la particular utilización de un diagrama de Secuencia Orientado a Actividades el cual se nutre de las principales bondades que brinda el diagrama de Actividades y el Diagrama de Secuencia. Este híbrido se realizó con el objetivo de facilitar el trabajo de los programadores que no tienen un alto nivel de involucración en el desarrollo del proyecto, pudiéndose así comprender con facilidad el diseño de cada una de las actividades orientadas por los diseñadores a cualquiera de los programadores del departamento. (21)

## **1.7 Análisis del Sistema Ventanilla Única para el Comercio Exterior de Cuba**

Con ánimo de lograr un producto de excelencia y acotar el riesgo al fracaso mediante el uso de correctas técnicas basadas en la experiencia previa, se toma como buena práctica el empleo de una adecuada Ingeniería de Software, dada por el conjunto de métodos, técnicas y herramientas que se utilizan tanto para desarrollar el producto como para concebirlo. Los métodos y técnicas de la Ingeniería del Software se inscriben dentro del marco delimitado por el ciclo de vida del software.

Una forma de ciclo de vida del software estaría determinada de la siguiente manera: Análisis, Diseño, Programación, Prueba y Mantenimiento. En la etapa primera del ciclo de vida se desarrolla el Análisis de Software donde se determinan los rasgos del sistema, los recursos necesarios para el desarrollo del software, así como la viabilidad técnica, económica y legal del proyecto de desarrollo del software. También es objetivo en esta parte definir las necesidades de información que tendrá que resolver el software, obteniendo una idea precisa de las funciones y de los requisitos funcionales y no funcionales de éste. (22)

### **Técnicas para la captura de requisitos**

La captura de requisitos es la actividad mediante la cual el equipo de desarrollo de un sistema de software extrae de cualquier fuente de información disponible, las necesidades que debe cubrir dicho sistema. El proceso de captura de requisitos puede resultar complejo, principalmente si el entorno de trabajo es desconocido para el equipo de analistas, y depende mucho de las personas que participen en él. Por la complejidad que todo esto puede implicar,

# Capítulo 1: Fundamentación Teórica

---

la ingeniería de requisitos ha desarrollado técnicas que permitan hacer este proceso de una forma más eficiente y precisa (23).

Existen un grupo de técnicas que de forma clásica han sido utilizadas para esta actividad en el proceso de desarrollo de todo tipo de software:

- **Entrevistas:** resultan una técnica muy aceptada dentro de la ingeniería de requisitos y su uso está ampliamente extendido. Las entrevistas le permiten al analista tomar conocimiento del problema y comprender los objetivos de la solución buscada. (23)
- **JAD (Joint Application Development/Desarrollo conjunto de aplicaciones):** esta técnica resulta una alternativa a las entrevistas. Es una práctica de grupo que se desarrolla durante varios días y en la que participan analistas, usuarios, administradores del sistema y clientes (23).
- **Brainstorming (Tormenta de ideas):** una técnica de reuniones en grupo cuyo objetivo es que los participantes muestren sus ideas de forma libre. Consiste en la mera acumulación de ideas y/o información sin evaluar las mismas. El grupo de personas que participa en estas reuniones no debe ser muy numeroso (máximo 10 personas), una de ellas debe asumir el rol de moderador de la sesión, pero sin carácter de controlador (23).
- **Concept Mapping:** Los mapas de conceptos son grafos en los que los vértices representan conceptos y las aristas representan posibles relaciones entre dichos conceptos. Estos grafos de relaciones se desarrollan con el usuario y sirven para aclarar los conceptos relacionados con el sistema a desarrollar. Son muy usados dentro de la ingeniería de requisitos, pues son fáciles de entender por el usuario (23).
- **Sketches y Storyboards:** Está técnica es frecuentemente usada por los diseñadores gráficos de aplicaciones en el entorno web. La misma consiste en representar sobre papel en forma muy esquemática las diferentes interfaces al usuario (sketches). Estos sketches pueden ser agrupados y unidos por enlaces dando idea de la estructura de navegación (storyboard). (23)
- **Casos de Uso:** La ventaja esencial de los casos de uso es que resultan muy fáciles de entender para el usuario o cliente, sin embargo carecen de la precisión necesaria si no

## Capítulo 1: Fundamentación Teórica

---

se acompañan con una información textual o detallada con otra técnica como pueden ser los diagramas de actividades. (23)

- **Cuestionarios y Checklists:** Esta técnica requiere que el analista conozca el ámbito del problema en el que está trabajando. Consiste en redactar un documento con preguntas cuyas respuestas sean cortas y concretas. Este cuestionario será cumplimentado por el grupo de personas entrevistadas o simplemente para recoger información en forma independiente de una entrevista. (23)
- **Comparación de terminología:** Uno de los problemas que surge durante la licitación de requisitos es que usuarios y expertos no llegan a entenderse debido a problemas de terminología. Esta técnica es utilizada en forma complementaria a otras técnicas para obtener consenso respecto de la terminología a ser usada en el proyecto de desarrollo. (23)
- **Sistemas Existentes:** Esta técnica consiste en analizar los sistemas nacionales e internacionales ya desarrollados que estén relacionados con el sistema que va a ser construido. Analizándose las interfaces de usuarios y las entradas y salidas que el mismo produce, con la posibilidad de nutrirse de ideas que han sido ya pensadas y probadas por países con un cierto nivel de experiencia. (24)
- **Observación:** Esta técnica requiere que el analista se sumerja por sí solo en el entorno laboral, observando el trabajo diario y anotando tareas que realizan los involucrados del sistema. Dicha tarea ayuda a los analistas a descubrir requisitos solo perceptibles en la observación de los procesos reales, forma mediante la cual entiende los requisitos sociales y organizacionales del sistema.
- **Talleres:** Los talleres resultan una técnica factible para lograr que el analista y los involucrados en el negocio lleguen a un entendimiento común. Consisten en reuniones formales e informales efectuadas con el fin de encontrar puntos de convergencia en ideas concebidas por los analistas y los stakeholders<sup>9</sup>.

---

<sup>9</sup> **Stakeholders:** Persona involucrada en el negocio, quien puede afectar o ser afectado por las actividades de la empresa.

# Capítulo 1: Fundamentación Teórica

---

Existen más técnicas para la captura de requisitos, incluso es común encontrar alternativas que combinen varias de estas técnicas. Sin embargo, las antes presentadas ofrecen un conjunto representativo de las más utilizadas y resultan suficientes para la captura de los requisitos de cualquier solución de software.

Sin embargo, de las técnicas utilizadas para la captura de los requisitos en el Departamento de Soluciones para la Aduana: entrevistas, observación, tormenta de ideas y talleres, las que se utilizarán para la captura de los requisitos del sistema de Ventanilla Única son: Entrevistas y Tormenta de Ideas, a las cuales se le agregará la técnica Estudio de Sistemas, de vital importancia dado el poco conocimiento que se tiene sobre este tipo de sistemas en la nación.

## **Técnicas para la validación de requisitos**

Los requisitos una vez definidos necesitan ser validados. La validación de requisitos tiene como misión demostrar que los mismos definen realmente el sistema que el cliente necesita.

Existen algunas técnicas que pueden aplicarse para ello:

- **Reviews o Walk-throughs:** Está técnica consiste en la lectura y corrección de la completa documentación o modelado de la definición de requisitos. Con ello solamente se puede validar la correcta interpretación de la información transmitida. Más difícil es verificar consistencia de la documentación o información faltante. (23)
- **Auditorías:** La revisión de la documentación con esta técnica consiste en un chequeo de los resultados contra una lista de chequeo predefinida o definida a comienzos del proceso, es decir sólo una muestra es revisada (23).
- **Matrices de trazabilidad:** Esta técnica consiste en marcar los objetivos del sistema y chequearlos contra los requisitos del mismo. Es necesario ir viendo qué objetivos cubre cada requisito, de esta forma se podrán detectar inconsistencias u objetivos no cubiertos. (23)
- **Prototipos:** Algunas propuestas se basan en obtener a partir de la definición de requisitos, prototipos que sin tener la totalidad de la funcionalidad del sistema, permitan al usuario comprender fácilmente la interfaz del mismo. (23)

Un prototipo es una versión inicial de un sistema de software que se utiliza para demostrar los conceptos, probar las opciones de diseño y entender mejor el problema y su solución. Un prototipo puede revelar errores u omisiones en los requisitos propuestos, favorece la comunicación entre clientes y desarrolladores, da una primera visión del producto.

➤ **Revisión Técnica Formal (RTF):**

Las RTF son actividades de garantía de los sistemas informáticos, son reuniones del personal técnico con el objetivo de descubrir problemas de calidad en el producto a revisar. Su aplicación a la documentación técnica permite detectar deficiencias, ambigüedades, omisiones, errores, tanto de formato como de contenido. Los técnicos deben ser independientes del equipo que ha elaborado el documento.

➤ **Revisión Informal:** Una revisión de requisitos es un proceso manual que involucra a personas tanto de la organización del cliente como de la del contratista. Ellos verifican el documento de requisitos en cuanto a anomalías y omisiones. Las revisiones informales sencillamente implican que los contratistas deben tratar los requisitos con tantos interesados del sistema como sea posible.

(23)

A raíz del análisis realizado a cada una de las técnicas antes mencionadas, se deciden escoger para la validación de requisitos la construcción de Prototipos de Interfaz de Usuario y las Revisiones Informales.

### **1.8 Diseño del Sistema de Ventanilla Única para el Comercio Exterior de Cuba**

Un proceso de desarrollo de software es a grandes rasgos, donde se traducen las necesidades del usuario en requisitos de software para luego transformarlos en diseño y posteriormente en la implementación. Dentro de las principales metas de un software se pueden encontrar la facilidad de uso, una agradable interfaz de usuario, un acertado nivel de funcionalidad alcanzado así como sus contenidos de forma adecuada. Cuando se diseña se

# Capítulo 1: Fundamentación Teórica

---

deben tener en cuenta ciertos estándares, patrones y estilos con el fin de lograr una solución robusta y apropiada.

Producto de la selección del marco de trabajo Symfony prevaleció la implementación del estilo arquitectónico Modelo Vista Controlador el cual divide una aplicación interactiva en 3 componentes. El Modelo, el cual contiene la información y funcionalidad principal; la Vista, encargada de mostrar la información al usuario, y por último el controlador el cual responde a los eventos, usualmente acciones del usuario, e invoca peticiones al modelo, un mecanismo de propagación de cambios que asegura la consistencia entre el modelo y la interfaz de usuario.

## **Patrones de Diseño**

Durante el desarrollo de un software surgen con frecuencia problemas que suelen tener como factor común una misma solución. Tales soluciones cuando son definidas de modo correcto para un problema dado, describiendo sus cualidades, permitiendo su reutilización en posteriores soluciones y evitando la búsqueda reiterativa a problemas ya conocidos y solucionados anteriormente, son conocidas como patrones.

En las palabras de Christopher Alexander, “Cada patrón describe un problema que ocurre infinidad de veces en nuestro entorno, así como la solución al mismo, de tal modo que podemos utilizar esta solución un millón de veces más adelante sin tener que volver a pensarla otra vez. Cada patrón es una regla de tres partes, la cual expresa una relación entre un cierto contexto, un problema y una solución” (27).

### **Patrones de Diseño en Symfony**

Los patrones de diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software. Para que una solución de diseño sea considerada como un patrón debe tener ciertas características. Una de ellas es que debe haber comprobado su efectividad resolviendo problemas similares en ocasiones anteriores. Además debe ser reusable, lo que significa que es aplicable a diferentes problemas de diseño en distintas circunstancias. Existen dos clasificaciones de los patrones de diseño, las cuales se describen a continuación:

**Patrones GoF (Gang of Four) o Cosa de Cuatro:** Su traducción al español significa “Pandilla de Cuatro”, y describen las clases y objetos que se comunican entre sí, y se adaptan

# Capítulo 1: Fundamentación Teórica

---

para resolver un problema general de diseño en un contexto particular. Estos vienen agrupados por los patrones creacionales para abstraer el proceso de instanciación y creación de objetos, los patrones estructurales, que describen como pueden ser combinados las clases y los objetos para formar grandes estructuras; y por último los patrones de comportamiento para definir la comunicación entre los objetos del sistema. (28)

- **Singleton o Instancia Única (Creacional):** Asegura la existencia de una única instancia para una clase, además de un mecanismo de acceso global a ella. Su intención es garantizar que una clase solo tenga una instancia a proporcionar un punto de acceso global a ella.
- **Flyweight o Peso Ligero (Estructural):** Se utiliza con el objetivo de reducir la redundancia al crear una gran cantidad de objetos mediante la creación de uno solo con características globales.
- **Command o Comando (Comportamiento):** Encapsula una cierta cantidad de funcionalidades en una sola estructura haciendo este funcionamiento oculto para el usuario. Brinda la facilidad de solicitar una operación a un objeto aun cuando no se conozca su lógica.
- **Strategy o Estrategia (Comportamiento):** Permite la creación de varios objetos con el mismo objetivo y similares características y elegir cual usar en el momento apropiado.
- **Registry o Registro (Comportamiento):** Este patrón es muy útil para los desarrolladores en la Programación Orientada a Objetos. Este patrón es un medio sencillo y eficiente de compartir datos y objetos en la aplicación sin la necesidad de preocuparse por conservar numerosos parámetros o hacer uso de variables globales.

**Patrones GRASP o Enfoque:** GRASP es un acrónimo de General Responsibility Assignment Software Patterns, en español patrones generales de software para asignar responsabilidades, y su similitud con el término grasp (agarrar) viene dada con toda intención. Estos están basados mayormente en la asignación de responsabilidades a objetos.

# Capítulo 1: Fundamentación Teórica

---

- **Experto:** Indica cómo realizar la asignación de responsabilidades, donde dice que la responsabilidad de un objeto debe recaer sobre la clase que conoce toda la información necesaria para hacerlo.
- **Creador:** Ayuda a identificar quien debe ser el responsable de la creación de nuevos objetos, donde la nueva clase deberá ser creada por la que tiene toda la información necesaria para realizar la acción.
- **Controlador:** Sirve como mediador entre una interfaz y el algoritmo que la implementa, sugiere que la lógica debe estar separada de la presentación.
- **Alta Cohesión:** Orienta en la medida de lo posible un alto nivel de comunicación entre las clases buscando obtener acceso a toda la información de manera simple.

## 1.9 Conclusiones del Capítulo

En este capítulo se abordaron los conceptos fundamentales para la correcta comprensión del presente trabajo entre ellos el estudio de los Sistemas de Ventanilla Única del mundo, el cual arrojó como conclusión la importancia de profundizar en este aspecto dado el pobre conocimiento que se tiene en esta área en la nación. Se concluyó por tal motivo, que la solución a implementar deberá contar con facilidades como el manejo de notificaciones instantáneas y la presentación de documentos en formato digital. Por otra parte las herramientas y tecnologías seleccionadas por el Modelo de Desarrollo permitirán desarrollar el sistema de forma eficiente. Finalmente, estudiar las etapas de Análisis y Diseño provee de un grupo de elementos para efectuar la captura y validación de los requisitos, así como para guiar la estructuración de la implementación a través de los patrones de diseño.



## **Capítulo 2: Análisis y Diseño**

### **2.1 Introducción del Capítulo**

En el presente capítulo se abordará primeramente la etapa del análisis, lo cual incluye la descripción de los requisitos funcionales y no funcionales del sistema, el modelado de los procesos del negocio así como las restricciones de diseño e implementación. Del mismo modo se abundará sobre las clases y el modelo de dicha fase, además de explicar los principales artefactos que le corresponde. Por último se tratará la etapa del diseño lo cual se verá referenciado en el presente documento al igual que el modelo del diseño, el cual aportará elementos importantes como diagramas de paquetes y diagramas de interacción del diseño.

### **2.2 Solución Propuesta**

El módulo para la Configuración y Notificación propicia al resto de los subsistemas de la Ventanilla Única, el mecanismo que da orden estructural y secuencial a los procesos que tienen lugar en la Ventanilla, sin importar en que aplicación o subsistema se encuentren. Esto lo realiza mediante el empleo de un componente adicional o plugin de php conocido como eZ Components, orientado al manejo de actividades desde la perspectiva de un proceso robusto y de buena escalabilidad y adaptabilidad, contemplando así la posibilidad de que existan cambios en el modo en que están intercomunicados.

Con eZ Components los desarrolladores ahorran tiempo de codificación y pueden enfocar sus esfuerzos en el desarrollo de soluciones más específicas para los clientes. Está diseñado técnicamente teniendo en cuenta muchos requisitos comunes para las soluciones actuales de manera que provee a los equipos de desarrollo, de bloques de código y componentes reutilizables en cualquier aplicación. La librería eZ Components no dicta las reglas acerca de cómo se ha de estructurar una aplicación, sino que proporciona la flexibilidad de decidir cómo y cuándo utilizar cada componente. Además, todas las interfaces de programación de aplicación (API) de los componentes están diseñadas de forma similar para hacerlas más consistentes. Como una colección de bloques de construcción independientes de alta calidad, eZ components acelera tanto el desarrollo como la reducción de los riesgos. Una aplicación

puede utilizar uno o más componentes sin esfuerzo, ya que todos se adhieren a las mismas convenciones de nomenclatura y siguen la misma estructura. (29)

Dicho componente permitirá organizar las actividades que se realizan en la Ventanilla en el proceso de recibir documentos digitales y todo lo que implica esto posteriormente, de manera mucho más configurable, lo cual se traduce a modificar el orden de ejecución de dichas actividades y eliminarlas o adicionar nuevas, teniendo en cuenta que en estas intervienen otros subsistemas y que el número de estos puede aumentar con el tiempo así como el orden en que se realizan tales subprocesos.

### **2.3 Análisis de la solución propuesta**

A continuación se describe la solución propuesta en la fase de análisis del sistema. Se realiza el modelado y descripción de los procesos de negocio, así como la descripción de los requisitos.

#### **2.3.1 Modelado de los procesos del Negocio del Sistema**

Con el objetivo de lograr un mayor entendimiento del negocio que posibilite el desarrollo del software con la mayor calidad posible se realiza a continuación la descripción del proceso Ventanilla Única y el subproceso Procesar Información.

#### **Descripción del Proceso Ventanilla Única**

El siguiente diagrama (Ilustración 1) representa el proceso Ventanilla Única a partir del cual se derivan cada uno de los subprocesos que se efectúan en la Ventanilla Única para el Comercio Exterior de Cuba, los cuales son: Generar Workflow, Crear XSD, Generar Auditorías, Generar Notificaciones, Recibir Información, Procesar Información, Enviar Información a Sistema Externo y Procesar Información Final.

El proceso inicia con la interrogante ¿Crear XSD?, si la respuesta es acertada se desarrolla el subproceso Crear XSD y termina el proceso general, de lo contrario el sistema verifica la existencia de este con la interrogante ¿XSD ya creado?, si es acertada la respuesta se desarrolla el subproceso Recibir Información, el cual está precedido por la acción del Usuario que consulta la Ventanilla Única con el objetivo de realizar una operación de importación o exportación de mercancía al país. Si la respuesta es negativa, concluye el proceso general.

## Capítulo 2: Análisis y Diseño

Luego de realizarse satisfactoriamente el subproceso Recibir Información, se ejecuta el subproceso Enviar Información a Sistema Externo y el sistema espera por los datos de la información procesada por el Sistema Externo para realizar el subproceso Procesar Información Final, luego de este subproceso, concluye el proceso general con la actividad Mostrar Información al Declarante, información de la operación de importación o exportación realizada. Paralelo al subproceso Enviar Información se desarrollan los subprocesos Generar Auditorías y Generar Notificaciones, los cuales están presentes durante todo el ciclo restante del proceso del negocio. Independientemente de todos estos subprocesos de la Ventanilla Única se desarrolla el subproceso Generar Workflow, el cual es fundamental para el desarrollo del módulo Configuración y Notificación de la VU, ya que el workflow es el subproceso que funciona como columna vertebral en el proceso general de la ventanilla.

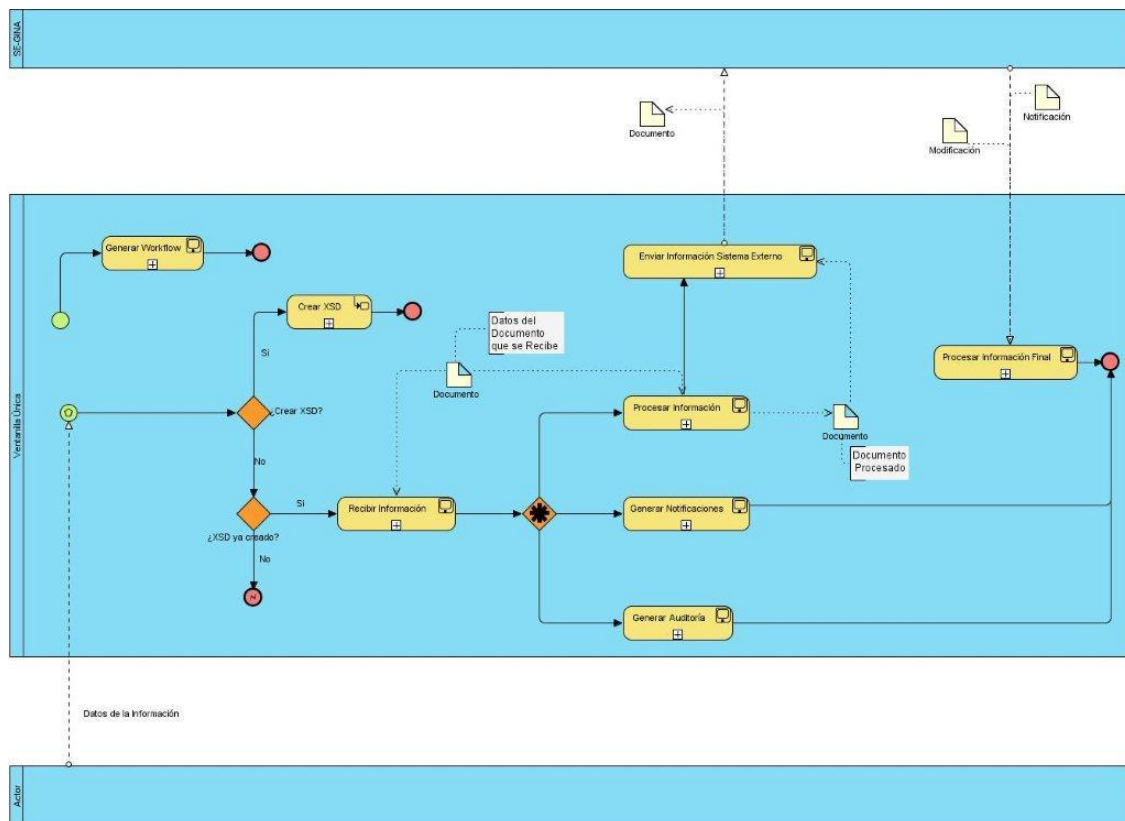


Ilustración 1: Diagrama de procesos del negocio. Proceso Ventanilla Única.

### **Descripción del Subproceso Procesar Información**

El siguiente diagrama (Ilustración 2) representa el subproceso Procesar Información a partir del cual se derivan cada uno de los subprocesos relacionados con el procesamiento de los distintos documentos de la Ventanilla Única, los cuales son: Procesar Documento Remisión de Entrada, Procesar Documento Cancelación, Procesar Documento Informe de Recepción, Procesar Documento Factura de Venta, Procesar Documento Ajuste, Procesar Documento Remisión de Salida, Procesar Documento Solicitudes, Procesar Documentos Complementarios, Procesar Documento Declaración de Mercancías, Procesar Documento Información Adelantada de Carga (ACI) y Procesar Documento Manifiesto de Carga.

El subproceso inicia con la actividad Identificar Documento, la cual realiza el identificado de cada uno de los documentos presentados, los cuales pueden ser: Remisión de Entrada, Informe de Recepción, Factura de Venta, Remisión de Salida, Cancelación, Ajuste, Solicitud, Complementario, Declaración de Mercancía, Información Adelantada de Carga, y Manifiesto de Carga. Después de identificados los documentos según el tipo, se efectúa el subproceso correspondiente a cada uno para su procesamiento y concluye el subproceso.

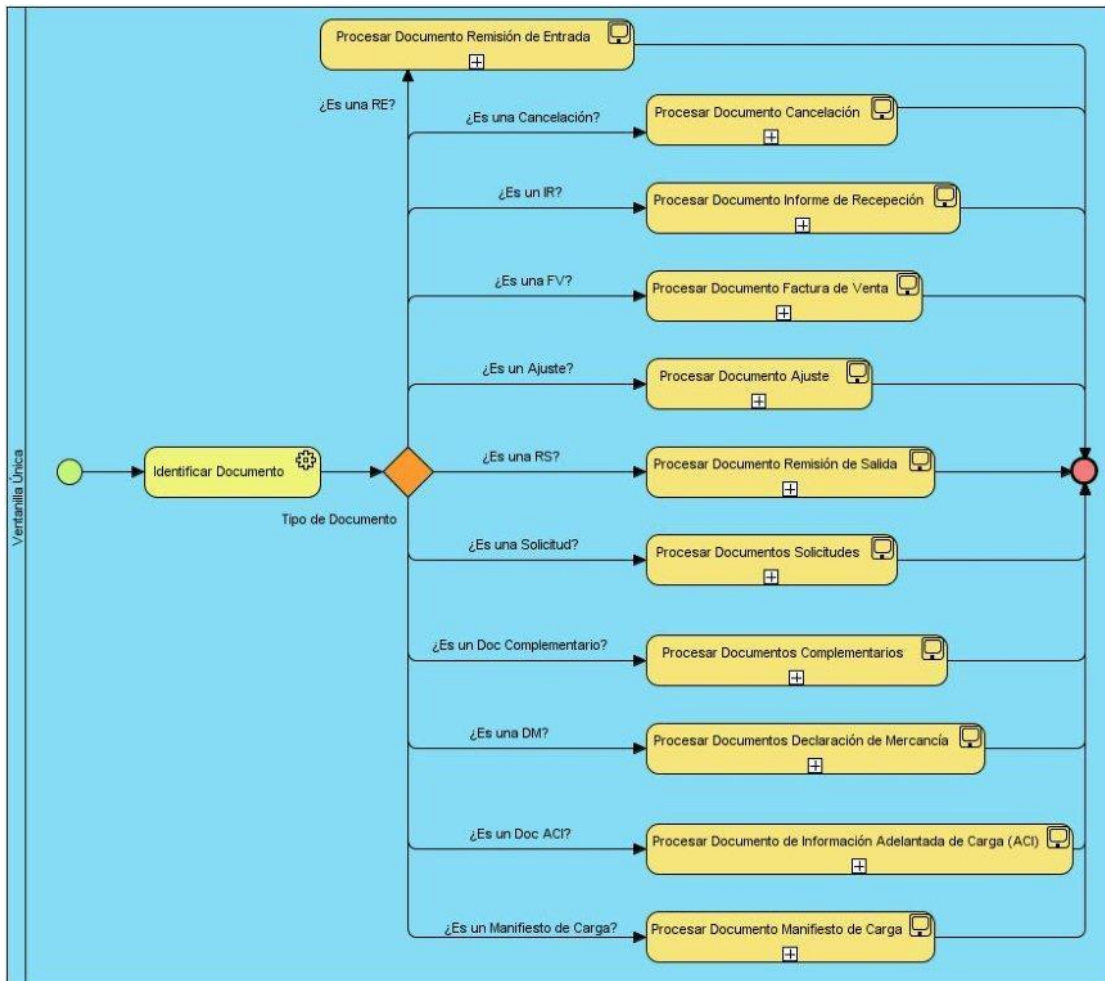


Ilustración 2: Diagrama de Procesos del Negocio. Subproceso Procesar Información.

### 2.3.2 Modelo Conceptual del Sistema

A continuación se muestra (Ilustración 3) el Modelo Conceptual del Negocio, donde se definen las principales estructuras que se manejaron en la conceptualización del módulo, muchos de los cuales serán persistentes en la Base de Datos una vez definido el Diagrama de Clases. Al igual se manejan atributos que pueden pasar a ser clases que servirán como modelos para almacenar objetos de ese tipo.

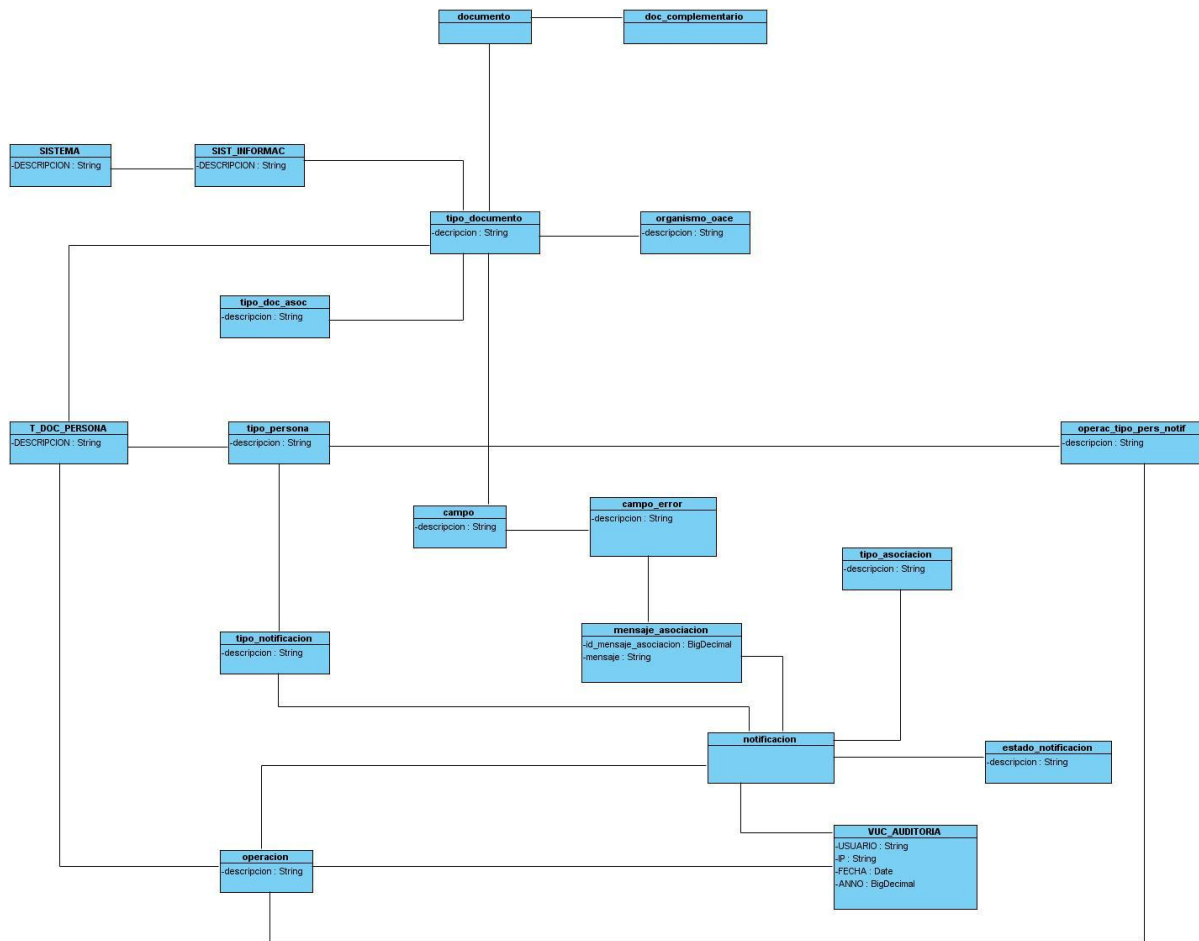


Ilustración 3: Modelo Conceptual.

### 2.3.3 Técnicas para la captura de los Requisitos

Guiados por el procedimiento para la Ingeniería de Requisitos que se desarrolla en el Departamento de Soluciones para la Aduana del CEIGE<sup>10</sup>, se tienen en cuenta las técnicas que son más factibles utilizar para la captura y la validación de los requisitos del sistema, a continuación se exponen las técnicas empleadas además de un listado de los mismos y las técnicas de validación que se utilizaron para asegurar la correcta conceptualización de estos.

De las técnicas utilizadas para la captura de los requisitos del sistema de VUCE, las técnicas de Entrevistas y Tormenta de Ideas fueron combinadas. Se entrevistaron a miembros del

<sup>10</sup> **CEIGE**: Definición de la abreviatura para Centro de Informatización para la Gestión de Entidades.

Departamento de Soluciones para la Aduana que estaban realizando estudios respecto al tema de Ventanillas Únicas en el Mundo, dentro de los cuales se encuentran el Jefe del Departamento de Soluciones para la Aduana y el Jefe del sistema de Ventanilla Única, donde se acumularon ideas para tener una perspectiva general de las necesidades que presentaría el sistema. La técnica Estudio de Sistemas se realizó en el Estudio del Estado del Arte para la elaboración del Marco Teórico, dentro de los sistemas que fueron estudiados se encuentran los de Corea, Perú, Ecuador y Chile.

### 2.3.4 Requisitos del Sistema

#### Requerimientos Funcionales del Sistema

Los requerimientos funcionales (RF) son declaraciones de los servicios que debe proporcionar el sistema, de la manera en que este debe reaccionar a entradas específicas y de cómo se debe comportar en situaciones particulares. (30)

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

**R.F #1: Crear Servicio Validar XSD:** El sistema deberá ser capaz de brindar la estructura para la implementación del servicio para la validación de los ficheros XML mediante el estándar XSD.

**R.F #2: Crear Servicio Validar Negocio:** El sistema deberá ser capaz de brindar la estructura para la implementación del servicio para la validación del negocio.

**R.F #3: Crear Servicio Guardar Incidencia:** El sistema deberá ser capaz de brindar la estructura para la implementación del servicio para guardar las incidencias.

**R.F #4: Crear Servicio Guardar Notificaciones:** El sistema deberá ser capaz de brindar la estructura para la implementación del servicio para guardar las notificaciones.

**R.F#5: Guardar Notificaciones:** El sistema deberá ser capaz de registrar las notificaciones.

**R.F #6: Mostrar Notificación:** El sistema deberá ser capaz de mostrar notificaciones.

**R.F #7: Mostrar Auditoría:** El sistema deberá ser capaz mostrar las auditorías realizadas.

**R.F #8: Definir WorkFlow:** El sistema deberá posibilitar la definición de la estructura del workflow.

## Capítulo 2: Análisis y Diseño

---

**R.F #9: Guardar WorkFlow en Base de Datos:** El sistema deberá ser capaz de registrar la definición del workflow en la Base de datos.

**R.F #10: Cargar WorkFlow de la Base de Datos:** El sistema deberá ser capaz de cargar la definición del workflow que se encuentra en la Base de Datos.

**R.F #11: Guardar WorkFlow en XML:** El sistema deberá ser capaz de registrar la definición del workflow en el fichero XML.

**R.F #12: Cargar WorkFlow de un fichero XML:** El sistema deberá ser capaz de cargar la definición del workflow que se encuentra en un fichero XML.

**R.F #13: Ejecutar WorkFlow:** El sistema deberá ser capaz de activar la ejecución del workflow.

**R.F #14: Asociar Documentos:** El sistema deberá ser capaz de asociar documentos.

**R.F #15: Pago de Facilidad Anticipada:** El sistema deberá ser capaz de permitir que se realice un pago de facilidad anticipada.

**R.F #16: Otorgar Número Manifiesto Externo:** El sistema deberá ser capaz de permitir que se otorgue un número de manifiesto externo.

**R.F #17: Confirmar Exportación:** El sistema deberá ser capaz de permitir que se confirme la exportación.

### Requerimientos No Funcionales del Sistema

Los requisitos no funcionales (RNF) son restricciones de los servicios o funciones ofrecidos por el sistema. Incluyen restricciones de tiempo, sobre el proceso de desarrollo y estándares. A menudo son aplicados en su totalidad al sistema y normalmente apenas se aplican a características o servicios individuales del sistema. (30)

A continuación se exponen los requisitos no funcionales definidos para el sistema:

#### Usabilidad:

**RNF #1: Cumplir con las pautas de diseño de las interfaces:** El sistema deberá contar con un diseño de interfaz estandarizado siguiendo los lineamientos de arquitectura definidos de manera global.



## Capítulo 2: Análisis y Diseño

---

**RNF #2: Agrupar vínculos y botones por grupos funcionales:** El sistema deberá agrupar funcionalidades determinadas de manera global y especializadas de acuerdo a los diferentes niveles de acceso para los usuarios, con el objetivo de propiciar un fácil manejo de la aplicación por parte de los operadores del mismo.

**RNF #3: Los mensajes, títulos y demás textos que aparezcan en la interfaz del sistema deben aparecer en idioma español:** Los mensajes, títulos y demás textos que aparezcan en la interfaz del sistema deben aparecer en idioma español.

**RNF #4: Permitir uso del teclado para realizar operaciones sobre el sistema (Permitir acceso rápido al sistema usando el teclado):** El sistema deberá contar con un grupo de funciones básicas mediante el uso del teclado como, Tab, Enter, ctrl c, ctrl v y ctrl x, para desplazarse por las diferentes regiones de la pantalla, accionar funciones mediante los botones establecidos, así como copiar, pegar y cortar texto.

### **Fiabilidad:**

**RNF #5: Prever contingencias para eventos de caída del sistema:** El sistema deberá resistir contingencias como:

- Exceso de volumen de usuarios.
- Exceso de carga de procesos.
- Exceso de carga de transacciones.
- Exceso de datos en la Base de Datos.

### **Eficiencia:**

**RNF #6: Responder en un tiempo las peticiones que se realicen en el sistema:** El sistema deberá responder con un tiempo menor de 30 segundos a las peticiones efectuadas por los usuarios.

### **Seguridad:**

## Capítulo 2: Análisis y Diseño

---

**RNF #7: Permitir el intercambio de datos entre el cliente y el servidor por canales cifrados:** El sistema deberá manejar de forma segura el envío de información entre clientes y el servidor.

**RNF#8: Definir una jerarquía de usuarios para el manejo centralizado de los mismos en el sistema:** El sistema deberá manejar un nivel de jerarquía para los usuarios con el fin de restringir el acceso al mismo.

**RNF#9: Permitir la gestión de las contraseñas:** El sistema deberá permitir la gestión de las contraseñas por parte de los usuarios teniendo en cuenta exigencias para su correcta construcción.

**RNF#10: Almacenar de manera cifrada las claves de los usuarios en la base de datos:** El sistema deberá permitir almacenar las contraseñas de los usuarios de modo seguro.

**RNF#11: Permitir la gestión y administración de usuarios en el sistema:** El sistema deberá permitir la gestión diferenciada de los usuarios en el sistema.

**RNF#12: Definir niveles de acceso a la información en el sistema:** El sistema deberá restringir el acceso a la información teniendo en cuenta la entidad que lo genera y por consiguiente será responsable de ella.

### Portabilidad:

**RNF#13: Garantizar compatibilidad con navegadores de uso común:** El sistema deberá ser compatible con los navegadores:

- Microsoft Internet Explorer 6.0 o superior.
- Mozilla Firefox 2.0 o superior.

### Capacidad:

**RNF#14: Condiciones técnicas para la ejecución en clientes:** Para que un cliente pueda ejecutar la aplicación correctamente, su PC debe poseer las siguientes características técnicas:

- Procesador 1.2 GHz.
- 512 MB de memoria RAM.
- 8 GB de espacio libre en disco duro.
- Conexión a internet a 56 Kbps.

### 2.3.5 Técnicas para la Validación de los Requisitos del Sistema

#### Prototipos

Para la validación de los requisitos se realizaron tantos prototipos de interfaz de usuario como requisitos funcionales lo requerían.

El prototipo Otorgar Número de Manifiesto Externo (Ilustración 4), muestra cómo debe ser implementada esta funcionalidad. Una vez que el consignatario interactúa con la vista principal de la funcionalidad y al insertar el IMO que es un número único que le ha sido otorgado al buque por la Organización Marítima Internacional, el sistema muestra al usuario los datos del buque, los cuales son el Nombre del buque, Tipo de Buque, Nacionalidad y Puerto de Registro. Luego de que el sistema muestre los datos pertenecientes al buque el consignatario escoge la entidad de tipo Agencia Consignataria e inserta el Destino del buque, luego al ser pulsado el botón Aceptar el sistema envía la información para realizar la operación de Otorgar número de Manifiesto Externo.



La imagen muestra un prototipo de interfaz de usuario para un sistema web. En la parte superior, hay un banner con el logo de 'Ventanilla Única' y el lema 'La puerta del comercio cubano', acompañado de una imagen de un avión. El título principal de la ventana es 'Otorgar No. de Manifiesto Externo'. El formulario contiene los siguientes campos:

- IMO:
- Nombre del Buque:
- Nacionalidad:
- Tipo de Buque:
- Puerto de Registro:
- Agencia Consignataria:
- Destino:

En la esquina inferior derecha del formulario hay un botón etiquetado como 'Aceptar'.

**Ilustración 4: Prototipo de Interfaz de Usuario Otorgar número de Manifiesto Externo.**

### 2.4 Diseño de la solución propuesta

En la etapa de Diseño de Software, el Modelo de Desarrollo para el Departamento de la Aduana, define una serie de tareas que se hacen necesarias para una correcta implementación de la solución que se pretende construir. Dicho esto, cabe mencionar la importancia de hacer énfasis en esta fase, la cual tiene un impacto grande en el producto final ya que sirve como abstracción de la implementación del sistema y por consiguiente se pone de manifiesto en esta última etapa del software de manera significativa.

#### 2.4.1 Diagrama de Paquetes del Sistema

El diagrama de paquetes, (Ilustración 5) muestra cómo está dividido el sistema en grupos lógicos, así como la dependencia existente entre ellos. El mismo se realiza con el objetivo de saber cómo sería la asignación de elementos entre los propios paquetes.

## Capítulo 2: Análisis y Diseño

Este diagrama se encuentra compuesto por un paquete principal denominado Notificación y Configuración el cual se encarga de gestionar todo lo referente a la gestión de las notificaciones y la configuración del sistema de Ventanilla Única. Dicho paquete está compuesto por cuatro paquetes que se describen a continuación y se relaciona con el paquete GINA, los cuales se describen a continuación:

**GINA:** Sistema Externo con el cual se comunica la Ventanilla Única con el objetivo de complementar el procesamiento de documentos digitales.

**Configuración:** Contiene los archivos de configuración de la aplicación.

**Abstracción datos Symfony:** Contiene todo lo referente al modelo de la aplicación.

**Formularios:** Contiene los formularios para la realización de validaciones.

**Interfaces:** Contiene todo lo referente a la vista.

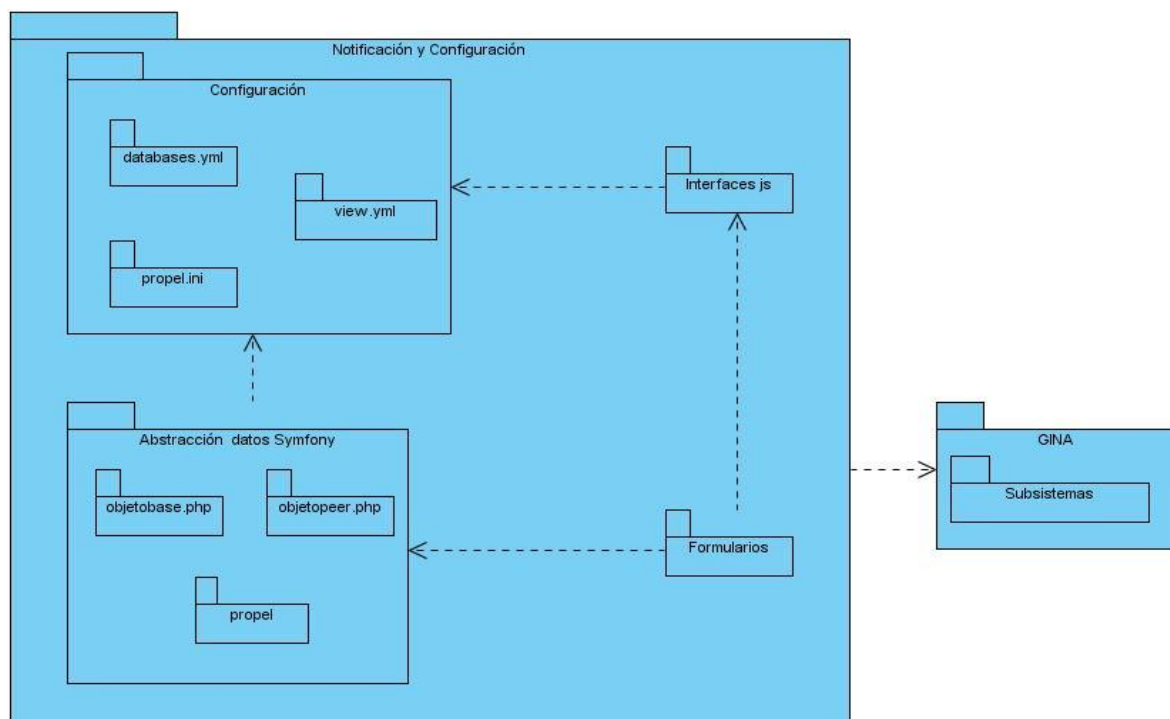


Ilustración 5: Diagrama de Paquetes del Sistema.

### 2.4.2 Diagrama de Secuencia orientado a Actividades del Sistema

## *Capítulo 2: Análisis y Diseño*

---

El diagrama de secuencia orientado a actividades (Ilustración 6) tiene como principal meta definir con mayor exactitud la relación existente entre las funcionalidades que se deben implementar y las clases correspondientes donde deben ser declaradas. Como se explica en el capítulo uno, esto trae consigo un mayor entendimiento de las funcionalidades, aún sin que la persona encargada de la implementación tenga conocimiento sobre cómo está distribuido el modelo del proyecto y cuáles son las políticas de codificación existentes para el mismo, ya que estas, están de cierto modo plasmadas en el diagrama. Éste describe entre sus principales características, cómo declarar funcionalidades y variables, así como la especificación de los parámetros establecidos para las funciones, que a su vez muestran su visibilidad y estructura interna.

El siguiente diagrama muestra la interacción entre las clases y los objetos que intervienen en la creación de la funcionalidad Guardar Notificación.

# Capítulo 2: Análisis y Diseño

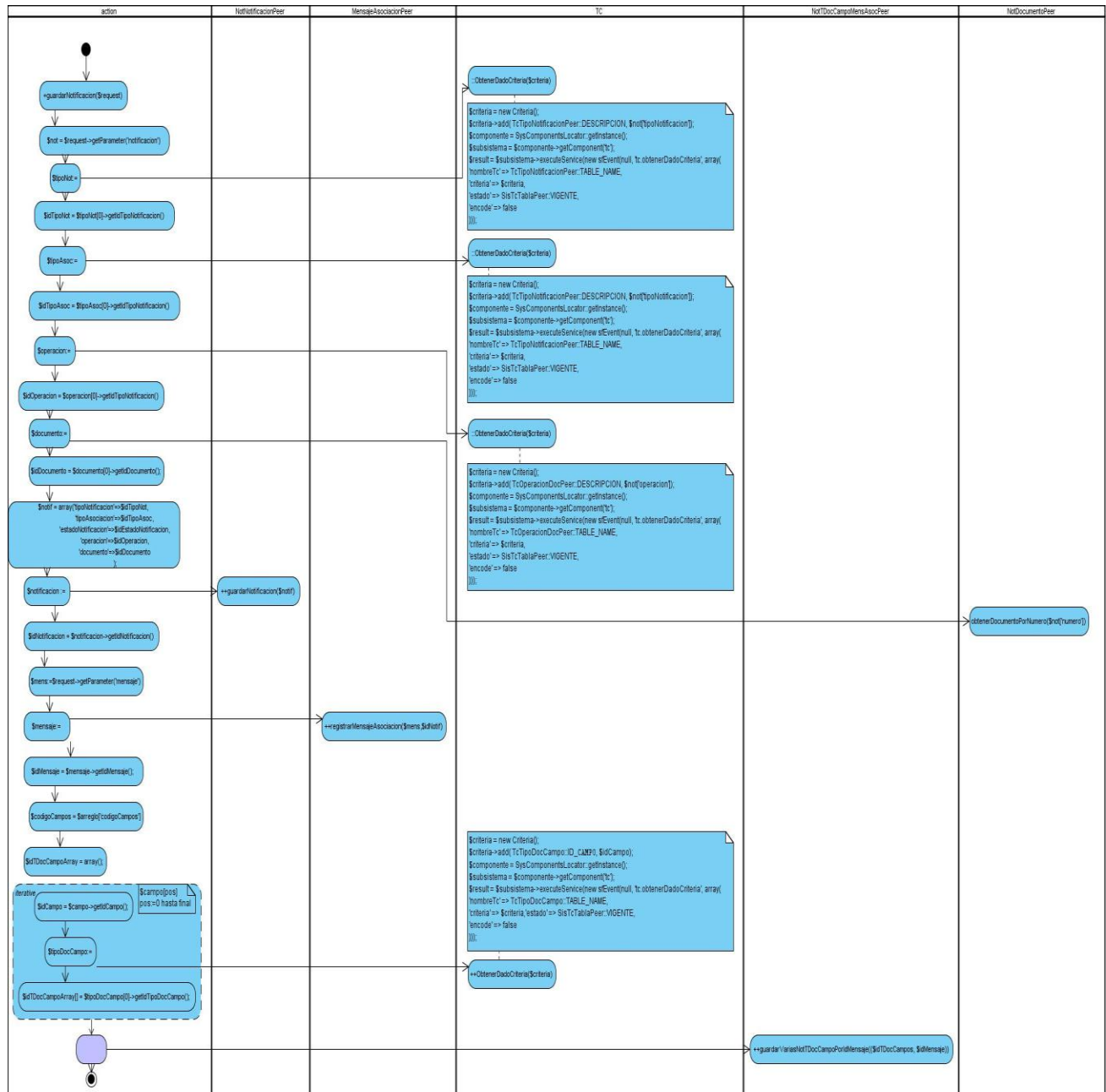


Ilustración 6: Diagrama de Secuencia Orientado a Actividades. Guardar Notificación.

## 2.4.3 Diagrama Entidad-Relación del Sistema

# Capítulo 2: Análisis y Diseño

El modelo Entidad-Relación, (Ilustración 7) compuesto por entidades, atributos, relaciones y conjuntos de relaciones, está orientado a tratar cualquier objeto, concepto o parte del mundo de lo cual se desea almacenar información persistente. El siguiente Diagrama Entidad Relación muestra las tablas, atributos y relaciones, que servirán como estructura de almacenamiento para la información correspondiente a las notificaciones para el módulo de Notificación y Configuración.

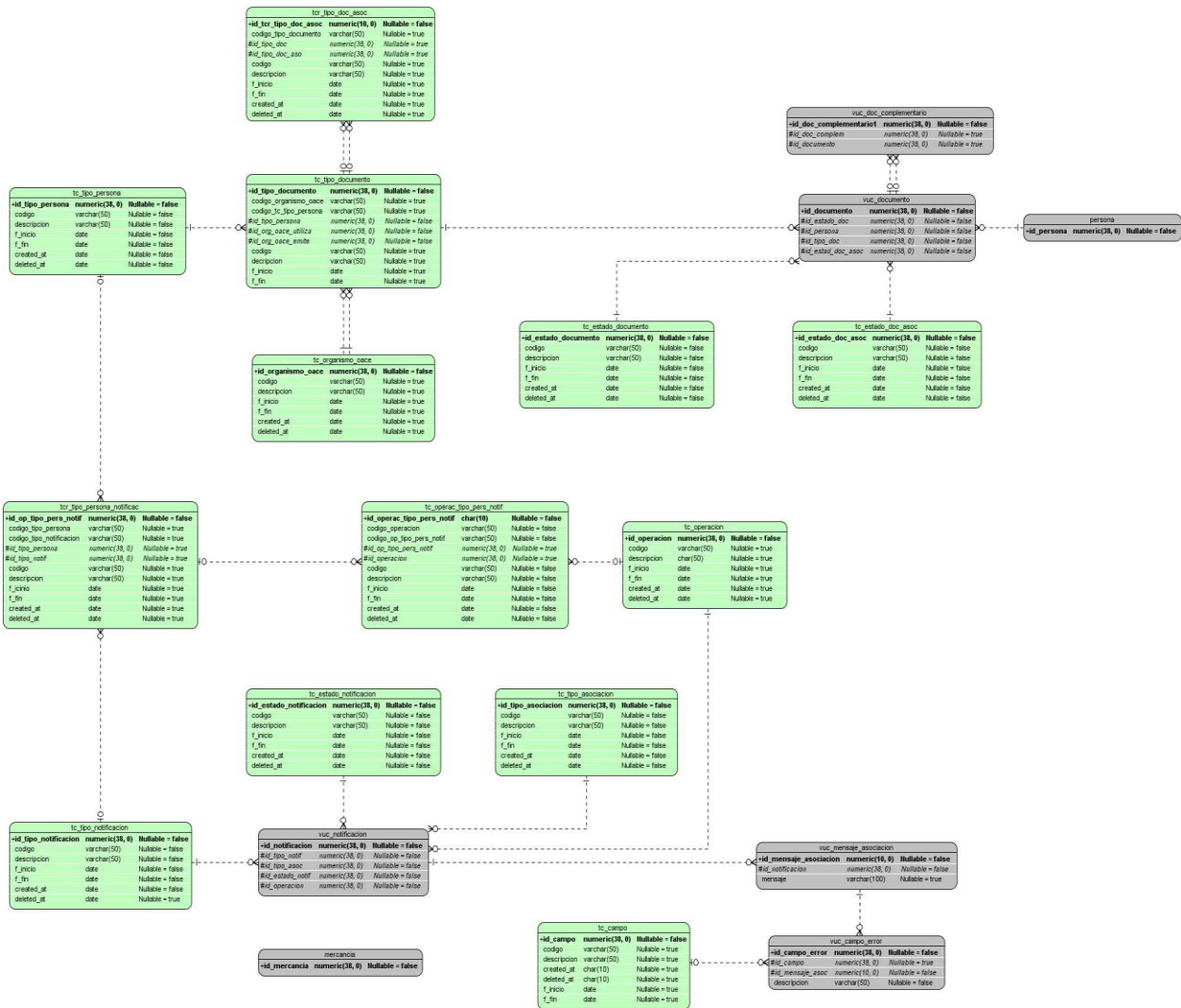


Ilustración 7: Diagrama de Entidad Relación. Esquema Notificación.



### 2.4.4 Métricas para la validación del Diseño del Sistema

Las métricas suelen calificarse como instrumentos que cuantifican un criterio determinado. Estas son utilizadas para indicar la calidad de un producto de software, evaluar los beneficios en términos de productividad y de calidad, así como para justificar el uso de determinadas herramientas, entre otras.

#### Tamaño Operacional de clase (TOC) :(31)

El tamaño general de una clase puede medirse determinando las siguientes medidas:

- El total de operaciones (tanto heredadas como privadas de la instancia), que se encapsulan dentro de la clase.
- El número de atributos (tanto heredados como privados de la instancia), encapsulados por la clase.

Valores grandes de TC representan gran responsabilidad de la clase. Esto implica la reducción de la reutilización de la clase y complica la implementación y las pruebas. De forma general, operaciones y atributos deben ser ponderados al determinar el tamaño de la clase. Para valores pequeños de TC para una clase existe mayor posibilidad de que la clase pueda ser reutilizada.

Parámetros de Calidad	Valores Grandes de TC
Reutilización	Reduce la reutilización de la clase
Implementación	Complica la implementación
Complejidad de las pruebas	Hace compleja las pruebas del sistema
Responsabilidad	La clase debe tener bastante responsabilidad

Tabla 1: Métrica Tamaño operacional de las clases.

#### Medidas o umbrales aplicados:

TC	Umbral
Pequeño	$\leq 20$
Medio	$> 20$ y $\leq 30$

## Capítulo 2: Análisis y Diseño

Grande >30

Tabla 2: Número de operaciones y/o atributos.

### Medidas para las principales clases de la solución:

No.	Clases	Atributos	Operaciones	Tamaño
1	UtilSW	0	2	Pequeño
2	NotNotificacion	6	2	Pequeño
3	TcEstadoNotificacion	7	1	Pequeño
4	TcTipoAsociacion	7	1	Pequeño
5	TcOperacionDoc	7	1	Pequeño
6	NotMensajeAsociacion	3	2	Pequeño
7	NotTDocCampoMensAsoc	2	3	Pequeño
8	NotDocumento	9	3	Pequeño
9	TcTipoNotificacion	7	1	Pequeño
10	TcrTipoTrabExtNotif	11	1	Pequeño
11	TcrTrabDocOper	11	1	Pequeño
12	TcrTipoTrabTipoDoc	11	1	Pequeño
13	Action(Módulo Notificación)	0	3	Pequeño
		Total 81	Total 22	

Tabla 3: Aplicación de la métrica TOC.

### Resultados obtenidos de la aplicación de la métrica al sistema:

Al ser aplicada la métrica de TC a un total de 13 clases, 81 atributos (con promedio de 6.23 atributos) y 22 operaciones (promediando 1.69 operaciones) se llega a la conclusión de que las clases analizadas tienen tamaño pequeño.

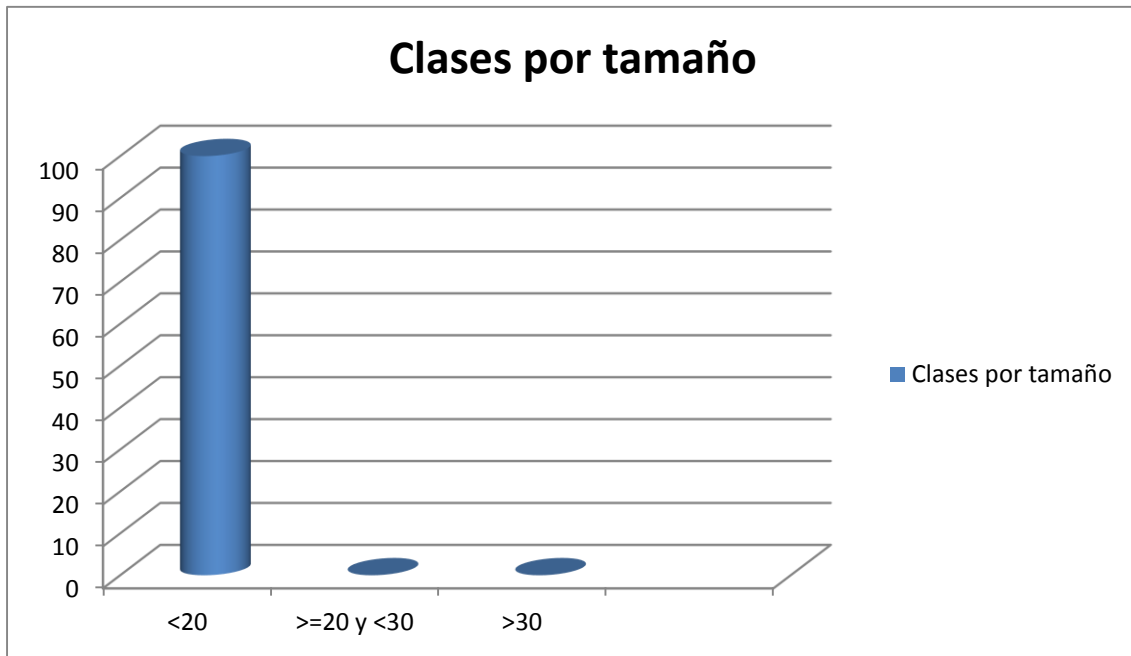


Ilustración 8: Gráfico de resultados de la métrica TOC aplicada al diseño de la solución.

El gráfico anterior muestra que todas las clases tienen tamaño Pequeño, lo que implica un resultado positivo según los parámetros de calidad propuestos para esta métrica, los cuales arrojan como resultado que el diseño realizado:

Aumenta la reutilización de la clase
Facilita la implementación
Facilita las pruebas del sistema
La clase debe tener menor responsabilidad

Tabla 4: Conclusiones de la métrica TOC aplicada al diseño de la solución.

### Métrica RC: Relaciones entre clases

Está dado por el número de relaciones de uso entre clases y evalúa los siguientes atributos de calidad:

## Capítulo 2: Análisis y Diseño

---

- **Acoplamiento:** Un aumento del RC implica un aumento del Acoplamiento de la clase.
- **Complejidad de mantenimiento:** Un aumento del RC implica un aumento de la complejidad del mantenimiento de la clase.
- **Reutilización:** Un aumento del RC implica una disminución en el grado de reutilización de la clase.
- **Cantidad de pruebas:** Un aumento del RC implica un aumento de la Cantidad de pruebas de unidad necesarias para probar una clase.

Representación porcentual de los resultados obtenidos en el instrumento agrupados en los intervalos definidos:

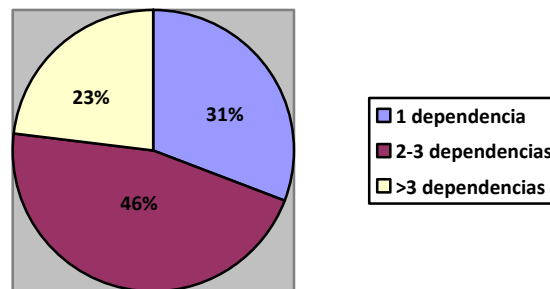


Tabla 5: Representación porcentual de las asociaciones de uso por cantidad a de clases.

- Representación porcentual de la incidencia de los resultados obtenidos en el atributo **Acoplamiento**:

## Capítulo 2: Análisis y Diseño

---

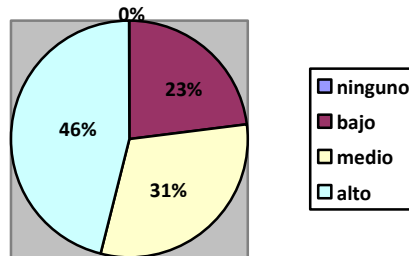


Tabla 6: Representación porcentual del valor del atributo Acoplamiento.

- Representación porcentual de la incidencia de los resultados obtenidos en el atributo **Complejidad de Mantenimiento**.

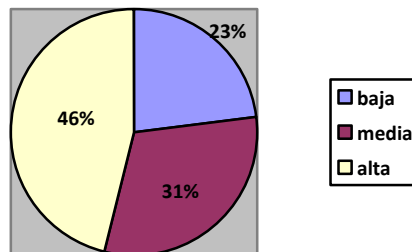


Tabla 7: Representación porcentual del valor del atributo Complejidad de Mantenimiento.

- Representación porcentual de la incidencia de los resultados obtenidos en el atributo **Cantidad de Pruebas**.

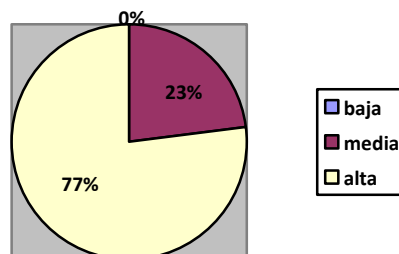


Tabla 8: Representación porcentual del valor del atributo Cantidad de Pruebas.

- Representación porcentual de la incidencia de los resultados obtenidos en el atributo **Reutilización**.

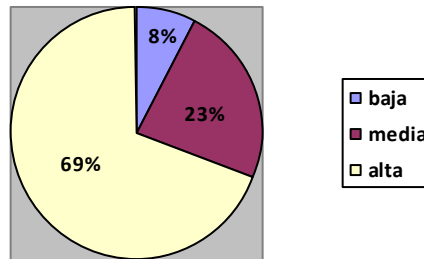


Tabla 9: Representación porcentual del valor del atributo Reutilización.

Considerando que la media de asociaciones para las clases es 5 y que un alto por ciento tiene menos de 2 asociaciones se puede afirmar que el modelo presenta un bajo nivel de colaboraciones por clase. De igual forma presenta un bajo nivel de complejidad de mantenimiento, una alta cantidad de pruebas y de reutilización debido a la cantidad de comprobaciones sobre el modelo y la cantidad de funcionalidades a reutilizar.

### 2.5 Conclusiones del Capítulo

En este capítulo se abordaron las principales actividades desarrolladas en los flujos de análisis y de diseño. Efectuar la etapa del Análisis mostró la importancia que tiene obtener y validar los requisitos del sistema de forma correcta, ya que estos se deben tener presente durante todo el proceso de desarrollo del mismo. Por último, en la etapa de Diseño se comprobó que para lograr una correcta representación de la solución es necesario el uso de patrones que guíen de manera satisfactoria su implementación. Además se realizó la evaluación del diseño utilizando la métrica Tamaño operacional de clases (TOC) obteniendo como resultado una calidad aceptable del diseño realizado teniendo en cuenta la valoración de la complejidad de la implementación de la solución propuesta en la presente investigación.

## Capítulo 3: Implementación y Prueba

### 3.1 Introducción del Capítulo

En el presente capítulo se abordará todo lo referente a la implementación del módulo, evidenciándose la utilización de patrones de diseño en el código fuente del mismo. Esto se logrará mediante la traducción del diseño en la solución y para ello se hará uso de los diagramas de interacción como base para la codificación. Además, en el proceso de desarrollo de esta etapa se garantizará la calidad de la solución mediante la aplicación de pruebas de software, lo cual permitirá revelar la calidad del producto dado diferentes indicadores como la usabilidad y calidad.

### 3.2 Diagrama de Despliegue

El diagrama de despliegue (Ilustración 9) se utiliza para modelar la configuración de los elementos de procesado en tiempo de ejecución y de los componentes, procesos y objetos de software que viven en ellos. Se modelan los nodos físicos y las asociaciones de comunicación que existe entre ellos. (32)

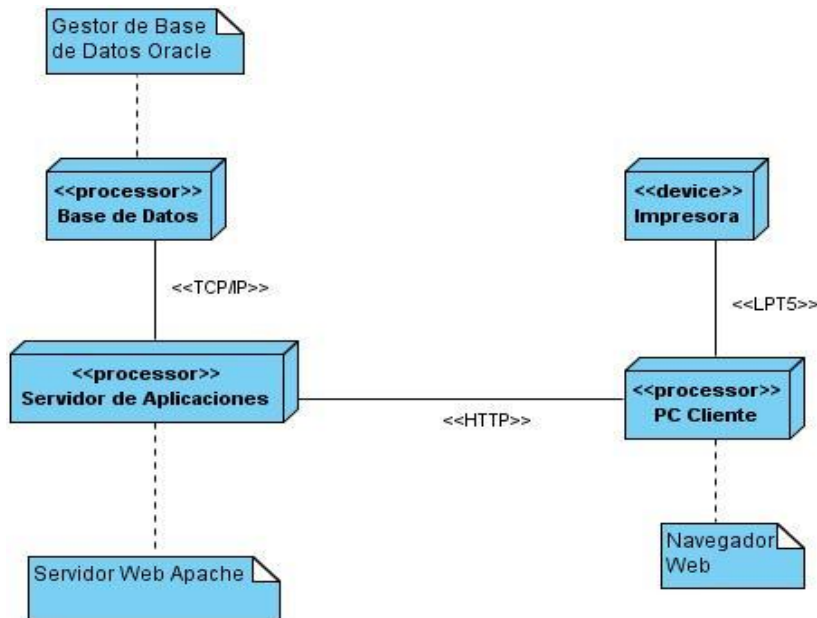


Ilustración 9: Diagrama de Despliegue del Sistema.

- **Nodo PC Cliente:** Representa las computadoras que utilizarán los usuarios para interactuar con la aplicación. Establece comunicación con el servidor de aplicaciones a través del protocolo HTTP.
- **Nodo Impresora:** Representa las impresoras, pues se requiere del uso de las mismas con el objetivo de imprimir los documentos que se necesiten.
- **Nodo Servidor de Aplicaciones:** En este nodo se encuentran los scripts de la aplicación.
- **Nodo Servidor de Base Datos:** En este nodo se encuentra el Servidor de Base de datos de la Ventanilla Única.

### 3.3 Diagrama de Componentes

Los diagramas de componentes (Ilustración 10) describen los elementos físicos del sistema y su relación, mostrando las dependencias lógicas entre componentes software. El diagrama de componente hace parte de la vista física de un sistema, la cual modela la estructura de implementación de la aplicación por sí misma, su organización en componentes y su despliegue en nodos de ejecución. La vista de implementación se representa con los diagramas de componentes. (32)



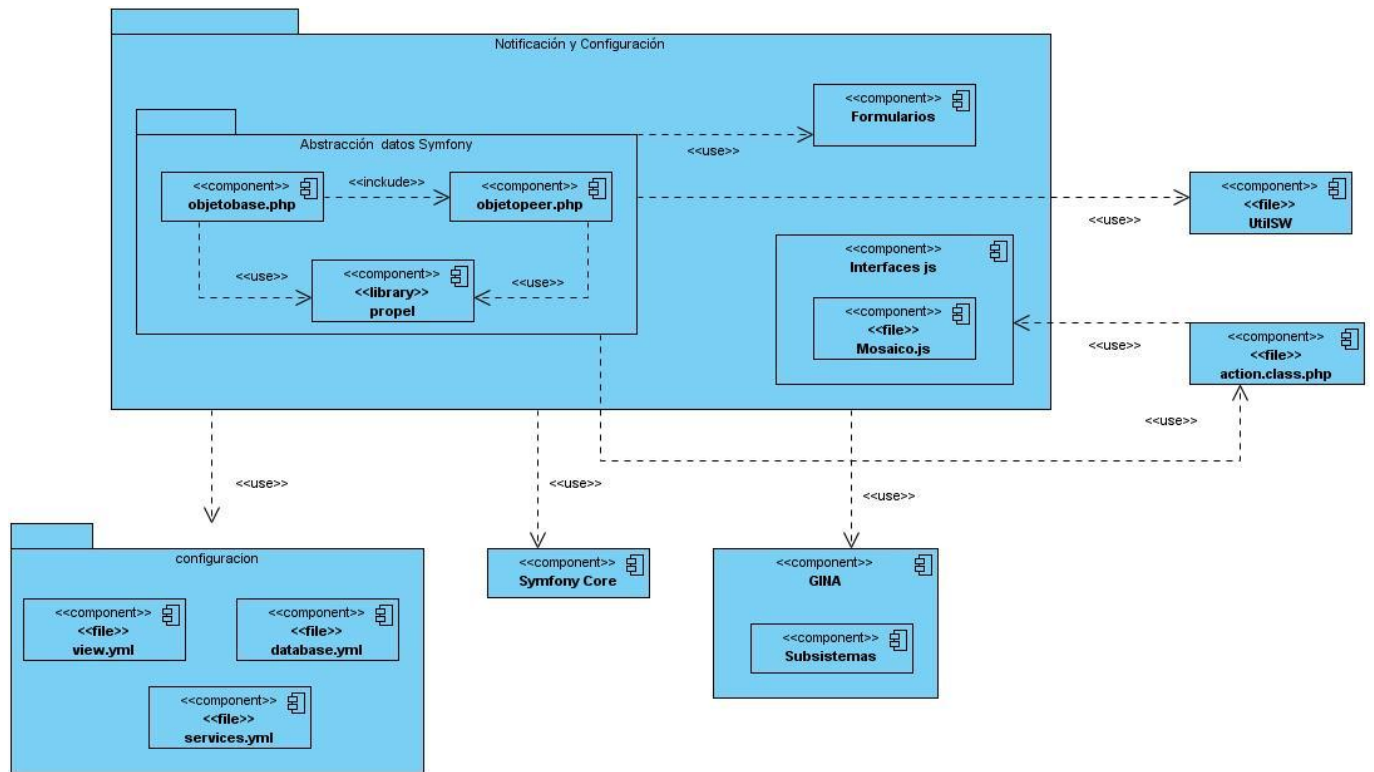


Ilustración 10: Diagrama de Componentes del Sistema.

## Estándares de codificación:

El Departamento de Soluciones para la Aduana utiliza el documento Estándar de Codificación con el objetivo de establecer como se debe codificar. Este procura crear un estándar que sirva para que el personal de proyecto identifique de forma sencilla cual es el objetivo y las funcionalidades que brinda cada una de las clases, funciones y demás componentes dada su nomenclatura.

### 3.4 Aspectos relevantes del flujo Guardar Notificación

El flujo Guardar Notificación responde al requisito funcional Guardar Notificación descrito con anterioridad. Dicho flujo cuenta con una funcionalidad principal y varias funcionalidades complementarias, a continuación se describen los principales elementos a considerar para su correcto entendimiento.

### Clases persistentes y del modelo en el flujo Guardar Notificación

## Capítulo 3: Implementación y Prueba

---

### 1. **NotNotificación:**

Es utilizada para registrar la notificación en cuestión.

### 2. **TcTipoAsociación:**

Es consultada para conocer el identificador del tipo de asociación perteneciente a la notificación en cuestión a partir del código del mismo.

### 3. **TcOperaciónDoc:**

Es consultada para conocer el identificador de la operación asociada a la notificación en cuestión a partir del código del mismo.

### 4. **TcTipoNotificación:**

Es consultada para conocer el identificador del tipo de notificación de la notificación en cuestión a partir del código de la misma.

### 5. **NotMensajeAsociación:**

Es utilizada para registrar el mensaje asociado a la notificación en cuestión.

### 6. **NotTDocCampoMensAsoc:**

Es utilizada para registrar la relación entre los campos nombrados asociados al mensaje asociado a la notificación en cuestión.

### 7. **NotDocumento:**

Es consultada para conocer el documento a quien pertenece la notificación que se desea registrar.

### 8. **TcrTipoDocCampo:**

Es consultada para conocer el identificador de los campos asociados al mensaje asociado a la notificación en cuestión.

Del mismo modo son utilizadas las clases del modelo referentes a las clases persistentes, en donde se llevan a cabo las implementaciones pertinentes para cumplimentar la funcionalidad que se desea desarrollar, así como la clase UtilSW donde se codificó el método principal de dicha funcionalidad.

#### 1. **UtilSW.**

#### 2. **TcTipoNotificacionPeer.**

3. TcTipoAsociacionPeer.
4. NotNotificacionPeer.
5. TcOperacionDocPeer.
6. NotDocumentoPeer.
7. NotMensajeAsociacionPeer.
8. TcrTipoDocCampoPeer.
9. NotTDocCampoMensAsocPeer.

### Codificación de la funcionalidad Guardar Notificación

En el siguiente fragmento de código se declara la funcionalidad Guardar Notificación y a continuación se utiliza el servicio ObtenerDadoCriteria con el fin de realizar una consulta a una clase del modelo que no se encuentra en la misma aplicación. De la misma forma se realiza con las consultas que se desean realizar a otras clases TC como: TcTipoAsociacion y TcOperacionDoc, implicadas en la funcionalidad.

```
public static function guardarNotificacion($arreglo)
{
    $not = $arreglo['notificacion'];
    //OBTENER TIPO NOTIFICACION
    $criteriatipoNot = new Criteria();
    $criteriatipoNot->add(TcTipoNotificacionPeer::CODIGO,
    $not['codigoTipoNot']);
    $syscomponent = SysComponentsLocator::getInstance();
    $componente = $syscomponent->getComponent('tc');
    $array = array();
    $array['nombreTc'] = TcTipoNotificacionPeer::TABLE_NAME;
    $array['estado'] = SisTcTablaPeer::VIGENTE;
    $array['criteria'] = $criteriatipoNot;
    $tipoNot = $componente->executeService(new sfEvent(null,
    'tc.obtenerDadoCriteria', $array));
    $idTipoNot = $tipoNot[0]->getIdTipoNotificacion();
    //-----
```

**Ilustración 11:** Fragmento de código. Funcionalidad Guardar Notificación.

En el siguiente fragmento de código se obtiene el identificador del documento relacionado a la notificación mediante la funcionalidad ObtenerDocumentoPorNumero definida en la clase del modelo NotDocumentoPeer.

## Capítulo 3: Implementación y Prueba

---

```
//OBTENER EL ID DEL DOCUMENTO A PARTIR DEL NUMERO
$documento = NotDocumentoPeer::
obtenerDocumentoPorNumero($not['numero']);
$idDocumento = $documento[0]->getIdDocumento();
//-----
```

Ilustración 12: Fragmento de código. Funcionalidad Obtener ID del documento.

En el siguiente fragmento de código se registra la notificación así como el mensaje asociado a la misma mediante los servicios correspondientes en las clases responsables del modelo.

```
//GUARDO LA NOTIFICACION
$notificacion = NotNotificacionPeer::
guardarNotificacion($notif);
$idNotificacion = $notificacion->getIdNotificacion();
//GUARDO EL MENSAJE
$mensaje = NotMensajeAsociacionPeer::
registrarMensajeAsociacion($arreglo['mensaje'],
$idNotificacion);
```

Ilustración 13: Fragmento de código. Funcionalidad Guardar Mensaje de Asociación a la notificación.

A continuación se obtienen los códigos de los campos asociados al mensaje y los identificadores asociados, mediante el servicio ObtenerDadoCriteria en un ciclo y se almacenan en un arreglo para registrarlos en la tabla NotTDocCampoMensAsocPeer.

```
foreach($codigoCampos as $codigoCampo)
{
    //OBTENER ID TIPO DOC CAMPO'S
    $criteriaCampo = new Criteria();
    $criteriaCampo->add(TcrTipoDocCampoPeer::
CODIGO_CAMPO,$codigoCampo);
    $syscomponent = SysComponentsLocator::getInstance();
    $componente = $syscomponent->getComponent('tc');
    $array = array();
    $array['nombreTc'] = TcrTipoDocCampoPeer::TABLE_NAME;
    $array['estado'] = SisTcTablaPeer::VIGENTE;
    $array['criteria'] = $criteriaCampo;
    $tipoDocCampo=$componente->executeService(new sfEvent(null,
'tc.obtenerDadoCriteria', $array));
    //-----
    $idTDocCampoArray[] = $tipoDocCampo[0]->getIdTipoDocCampo();
}
}
```

Ilustración 14: Fragmento de código. Funcionalidad Obtener ID del tipo de documento.

## Capítulo 3: Implementación y Prueba

---

En el siguiente fragmento de código se registran los campos asociados al mensaje de la notificación en cuestión.

```
public static function guardarNotTDocCampoMensaje($idTDocCampo, $idMensaje)
{
    $form = new NotTDocCampoMensAsocForm();
    $objeto = array('idTipoDocCampo'=>$idTDocCampo,
                  'idMensajeAsociacion'=>$idMensaje);
    $temp = UtilSW::procesarFormulario($objeto, $form);
    return $temp;
}
```

Ilustración 15: Fragmento de código. Funcionalidad Guardar Mensaje.

En el siguiente fragmento de código se obtienen los documentos con notificaciones asociadas, mediante el número del mismo, para su posterior utilización en la funcionalidad Guardar Notificación.

```
public static function obtenerDocumentoPorNumero($numero)
{
    $documento = NotDocumentoQuery::create()
        ->filterByNumero($numero)
        ->find();
    return $documento;
}
```

Ilustración 16: Fragmento de código. Funcionalidad Obtener documento por el número.

En el siguiente fragmento de código se hacen persistentes las notificaciones una vez procesadas en la tabla NotNotificacion mediante el servicio procesar Formulario de la clase UtilSW.

```
public static function guardarNotificacion($notificacion)
{
    $form = new NotNotificacionForm();
    $notificacion = UtilSW::
        procesarFormulario($notificacion, $form);
    return $notificacion;
}
```

Ilustración 17: Fragmento de código. Funcionalidad Guardar Notificación dada la notificación.

En el siguiente fragmento de código se implementa una funcionalidad para encapsular el proceso de validación de objetos y manejo de errores mediante formularios, de modo que sea más factible para la codificación.

## Capítulo 3: Implementación y Prueba

```
public static function procesarFormulario($array, sfForm $form)

    $result;
    $form->bind($array);
    if ($form->isValid()) {
        $result = $form->save();
    }
    else{
        $result = $form->getErrorSchema()->getMessage();
    }
    return $result;
```

Ilustración 18: Fragmento de código. Funcionalidad Procesar formulario.

En el siguiente fragmento de código se registra el mensaje asociado a la notificación una vez extraída del arreglo de la notificación en la clase UtilSW. En este se crea al formulario correspondiente a la notificación y posteriormente se procesa el objeto mensaje mediante el servicio procesarFormulario.

```
public static function registrarMensajeAsociacion($men, $idNotificacion)
{
    $form = new NotMensajeAsociacionForm();
    $mensa = array('mensaje'=>$men, 'idNotificacion'=>$idNotificacion);
    $mensaje = UtilSW::procesarFormulario($mensa, $form);
    return $mensaje;
}
```

Ilustración 19: Fragmento de código. Funcionalidad Registrar Mensaje de Asociación.

### 3.5 Patrones de Diseño en la Codificación del Sistema

#### Patrón Alta Cohesión y Patrón Experto

En el siguiente fragmento de código se evidencia cómo el patrón Alta Cohesión propicia la definición de las acciones en la clase Actions para instanciar objetos y acceder a sus propiedades. Esto último también pone en práctica al patrón Experto, puesto que este propicia que la clase Actions sea responsable de la creación de las instancias de las diferentes entidades para el acceso posterior a las funcionalidades de estas.

```
27
28 public function executeRegistrarIncidencia(sfWebRequest $request)
29 {
30     $this->form = new PRegIncPersonaForm();
31     PRegIncPersona::registrarIncidencia($request);
32 }
```

Ilustración 20: Fragmento de código donde se evidencia el patrón Alta Cohesión y Experto.

### Patrón Controlador

En el siguiente fragmento de código se evidencia el uso del patrón Controlador, donde la clase Index.php se encarga de manipular todas las peticiones web.

```
require_once(dirname(__FILE__).'/../config/ProjectConfiguration.class.php');

$configuración = ProjectConfiguration::getApplicationConfiguration('configuración', 'prod', false);
sfContext::createInstance($configuración)->dispatch();
```

Ilustración 21: Fragmento de código donde se evidencia el patrón Controlador.

### Patrón Instancia Única

El patrón Instancia Única se evidencia mediante la utilización de las funciones clearRoutes y hasRoutes para la gestión manual de las rutas en la clase sfRouting.

```
133 abstract public function getRoutes();
134 /*=...
142 abstract public function setRoutes($routes);
143 /*=...
149 abstract public function hasRoutes();
150 /*=...
154 abstract public function clearRoutes();
```

Ilustración 22: Fragmento de código donde se evidencia el patrón Instancia Única.

### Patrón Comando

El patrón comando se evidencia en la siguiente imagen, donde se muestra el encapsulamiento y tratamiento de funcionalidades de manera automática de forma transparente al usuario. La funcionalidad procesarFormulario se encarga de utilizar los mecanismos necesarios para la validación y el tratamiento de errores con un formulario y el objeto que se desea procesar facilitando este proceso.

```
public static function procesarFormulario(sfWebRequest $request, sfForm $form)
{
    $form->bind($request->getParameter($form->getName()));

    if ($form->isValid()) {
        $form->save();
        print_r('insertó');
    }
    else{
        print_r($form->getErrorSchema()->getMessage());
    }
}
```

Ilustración 23: Fragmento de código donde se evidencia el patrón Comando.

### 3.6 Validación de la Solución Propuesta

Como parte del proceso de desarrollo del software la fase de pruebas añade valor al producto que se maneja: todos los programas tienen errores y la fase de pruebas tiene como objetivo específico encontrarlos. Las pruebas de software son los procesos que verifican y revelan la calidad de un producto de software. Son utilizadas para identificar los posibles fallos de la implementación, calidad o la usabilidad de una aplicación informática. Entre las principales pruebas se encuentran las pruebas de Caja Blanca y de Caja Negra:

**Pruebas de Caja Blanca:** Consiste en realizar pruebas para verificar que líneas específicas de código funcionan tal como está definido y se comprueban los componentes internos. Al mismo tiempo prueba los caminos lógicos del software proponiendo casos de prueba que ejerciten conjuntos específicos de condiciones. Se puede examinar el estado del programa en varios puntos para determinar si el estado real coinciden con el esperado o mencionado.

**Pruebas de Caja Negra:** Se verifica la funcionalidad que se está probando, cuando se dan las entradas apropiadas produce los resultados esperados es decir se comprueban las funcionalidades sin tener en cuenta la estructura interna. Se refiere a las pruebas que se llevan a cabo sobre la interfaz del software los cuales demuestran a través de los casos de prueba que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce un resultado correcto, así como que la integridad de la información externa se mantiene.

Este tipo de pruebas permite encontrar:



## Capítulo 3: Implementación y Prueba

---

- Funciones incorrectas o ausentes.
- Errores de interfaz.
- Errores en estructuras de datos o en accesos a las bases de datos externas.
- Errores de rendimiento.
- Errores de inicialización y terminación.

Dentro de la prueba de caja negra se incluyen varias Técnicas de Pruebas tales como:

- Partición de Equivalencia: Divide el campo de entrada en clases de datos que tienden a ejercitar determinadas funciones del software.
- Análisis de Valores Límites: Prueba la habilidad del programa para manejar datos que se encuentran en los límites aceptables.
- Grafos de Causa-Efecto: Permite al encargado de la prueba validar complejos conjuntos de acciones y condiciones.

### 3.6.1 Pruebas Unitarias

Las pruebas Unitarias son enfocadas a los elementos verificables más pequeños del software. Es aplicable a componentes representados en el modelo de implementación para verificar que los flujos de control y de datos están cubiertos y que estos funcionen como se espera. (33)

Antes de iniciar cualquier otra prueba es preciso probar el flujo de datos de la interfaz del componente. Si los datos no entran correctamente, todas las demás pruebas no tienen sentido. El diseño de casos de prueba de una unidad comienza una vez que se ha desarrollado, revisado y verificado en su sintaxis el código a nivel fuente. (34)

Las pruebas unitarias aseguran que un único componente de la aplicación produzca una salida correcta para una determinada entrada. Este tipo de pruebas validan la forma en la que las funciones y métodos trabajan en cada caso particular. Se encargan de un único caso cada vez, lo que significa que un único método puede necesitar varias pruebas unitarias si su funcionamiento varía en función del contexto. (35)

### Pruebas Unitarias en Symfony

## Capítulo 3: Implementación y Prueba

---

El Framework Symfony se ha convertido en uno de los Frameworks de PHP más populares gracias a sus características avanzadas y su gran documentación. Además es el Framework utilizado por el Departamento de Soluciones para la Aduana. (36)

Symfony introduce herramientas y utilidades para automatizar las pruebas, las cuales aseguran la calidad de la aplicación incluso cuando el desarrollo de nuevas versiones es muy activo. Las pruebas unitarias de Symfony son archivos PHP cuyo nombre termina en Test.php y que se encuentran en el directorio test/unit/ de la aplicación. (37) Su sintaxis es sencilla y fácil de leer. Las pruebas unitarias se encargan de un único caso cada vez, lo que significa que un único método puede necesitar varias pruebas unitarias si su funcionamiento varía en función del contexto. (38)

### Pruebas Unitarias realizadas al Sistema

La evaluación de las funcionalidades del sistema se realizó mediante pruebas unitarias. Se generaron cuatro pruebas unitarias en total, dos a las auditorías y dos a las notificaciones. A continuación se muestran los resultados de estas pruebas cargados desde la consola.

```
c:\xampp\htdocs\UU>symfony test:unit notificacion
1..2
ok 1 - UtilSW::guardarNotificacion, debe guardar la notificacion
ok 2 - UtilSW::mostrarNotificaciones, debe mostrar las notificaciones
# Looks like everything went fine.
```

```
c:\xampp\htdocs\UU>symfony test:unit auditoria
1..2
ok 1 - UtilSW::guardarAuditoria, debe guardar la auditoria
ok 2 - UtilSW::mostrarAuditorias, debe mostrar las auditorias
# Looks like everything went fine.
```

Ilustración 24: Pruebas unitarias desde la consola.

### 3.6.2 Diseño de Casos de Pruebas para Pruebas de caja negra del Sistema

La evaluación del sistema fue ejecutada mediante pruebas funcionales aplicando el método de caja negra, y específicamente la técnica de partición de equivalencia (39). Esta última divide el dominio de entrada de un programa en un número finito de variables de equivalencia. Las variables de equivalencia representan un conjunto de estados válidos y no válidos para las condiciones de entrada de un programa. Se definen dos tipos de variables de equivalencia, las

## *Capítulo 3: Implementación y Prueba*

válidas, que representan entradas válidas al programa, y las no válidas, que representan valores de entrada erróneos, aunque pueden existir algunos de estos no relevantes a los que no sea necesario proporcionar un valor real de dato (40). A continuación se describe uno de los casos de prueba utilizado:

Requisito a probar: **Asociar Documento**

Escenario	Descripción	No. DM de VU	Código DC	Tipo de Solicitud	Respuesta del sistema	Flujo central
EC 1.1 Introducir Número de DM otorgado por VU.	Se introduce el número de DM otorgado por Ventanilla Única.	I	N/A	N/A	El sistema debe mostrar un mensaje notificando el error detectado.	1-El usuario introduce el No. de DM dado por la Ventanilla Única. 2-El sistema muestra un mensaje notificando el error detectado.
		V	N/A	N/A	Muestra los siguientes datos asociados a la DM: aduana de Despacho, el año de presentación, la operación, código y nombre de la entidad, código y nombre del declarante, tipo de DM, fecha de registro, no. de Manifiesto, BL/GA, de cada uno de los documentos complementarios su código y número y de cada una de las solicitudes su tipo y número.	1-El usuario introduce el No. de DM dado por la Ventanilla Única. 2-El sistema muestra los siguientes datos asociados a la DM: aduana de Despacho, el año de presentación, la operación, código y nombre de la entidad, código y nombre del declarante, tipo de DM, fecha de registro, no. de Manifiesto, BL/GA, de cada uno de los documentos complementarios su código y número y de cada una de las solicitudes su tipo y número.
EC 1.2	Se introduce	V	V	N/A	Actualiza la tabla que	1-El usuario introduce el

## Capítulo 3: Implementación y Prueba

Adicionar Documentos Complementarios	correctamente el no. de DM de la VU y se escoge la opción adicionar documento complementario .				contiene el listado de documentos complementarios adicionando el documento insertado.	No. de DM dado por la Ventanilla Única. 2-El sistema muestra los siguientes datos asociados a la DM: aduana de Despacho, el año de presentación, la operación, código y nombre de la entidad, código y nombre del declarante, tipo de DM, fecha de registro, no. de Manifiesto, BL/GA, de cada uno de los documentos complementarios su código y número y de cada una de las solicitudes su tipo y número. 3-El usuario escoge la opción adicionar Documento Complementario. 4-El usuario introduce el código del documento y escoge el número de aduana. 5- El sistema adiciona el documento complementario.
EC 1.3 Eliminar Documentos Complemen	Se introduce correctamente el no. de DM de la VU y se	V	V	N/A	Actualiza la tabla que contiene el listado de documentos complementarios	1-El usuario introduce el No. de DM dado por la Ventanilla Única. 2-El sistema muestra los

## Capítulo 3: Implementación y Prueba

tarios	escoge la opción eliminar documento complementario .				eliminando el documento seleccionado.	siguientes datos asociados a la DM: aduana de Despacho, el año de presentación, la operación, código y nombre de la entidad, código y nombre del declarante, tipo de DM, fecha de registro, no. de Manifiesto, BL/GA, de cada uno de los documentos complementarios su código y número y de cada una de las solicitudes su tipo y número. 3-El usuario escoge la opción eliminar Documento Complementario. 4-El usuario introduce el código del documento y escoge el número de aduana. 5- El sistema elimina el documento complementario escogido.
EC 1.2 Adicionar Solicitud	Se introduce correctamente el no. de DM de la VU y se escoge la opción adicionar solicitud.	V	N/A	V	Actualiza la tabla que contiene el listado de las solicitudes adicionando la solicitud insertada.	1-El usuario introduce el No. de DM dado por la Ventanilla Única. 2-El sistema muestra los siguientes datos asociados a la DM: aduana de Despacho, el año de presentación, la operación, código y nombre de la entidad, código y nombre del declarante, tipo de DM, fecha de registro, no. de Manifiesto, BL/GA, de cada uno de los documentos complementarios su código y número y de cada una de las solicitudes su tipo y

## Capítulo 3: Implementación y Prueba

						<p>número.</p> <p>3-El usuario escoge la opción adicionar solicitud.</p> <p>4-El usuario introduce el tipo de solicitud y escoge el número de aduana.</p> <p>5- El sistema adiciona la solicitud.</p>
EC 1.2 Eliminar Solicitud	Se introduce correctamente el no. de DM de la VU y se escoge la opción eliminar solicitud.	V	N/A	V	Actualiza la tabla que contiene el listado de las solicitudes eliminando la solicitud escogida.	<p>1-El usuario introduce el No. de DM dado por la Ventanilla Única.</p> <p>2-El sistema muestra los siguientes datos asociados a la DM: aduana de Despacho, el año de presentación, la operación, código y nombre de la entidad, código y nombre del declarante, tipo de DM, fecha de registro, no. de Manifiesto, BL/GA, de cada uno de los documentos complementarios su código y número y de cada una de las solicitudes su tipo y número.</p> <p>3-El usuario escoge la opción eliminar solicitud.</p> <p>4-El usuario introduce el tipo de solicitud y escoge el número de aduana.</p> <p>5- El sistema elimina la solicitud escogida.</p>

Tabla 10: Diseño de Casos de prueba. Requisito Asociar Documento.

## Capítulo 3: Implementación y Prueba

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	No. DM de VU	Campo texto	No	Número que le otorga la Aduana a la DM. Es un Entero consecutivo.
2	Código DC	Campo de selección múltiple.	No	Código de 3 caracteres.
3	Tipo de Solicitud	Campo de selección múltiple.	No	Combobox con los tipos de solicitudes que se asocian a la DM.

Tabla 11: Descripción de las Variables.

### Resultados de las pruebas realizadas:

Luego de aplicados los métodos de prueba a los módulos implementados, es válido señalar que los resultados obtenidos han sido satisfactorios desde el punto de vista interno y funcional de los componentes. Cada una de las no conformidades detectadas fue debidamente atendida logrando un correcto comportamiento de las funcionalidades ante diferentes situaciones (entradas válidas y no válidas).

A continuación se presenta el número de no conformidades detectadas tras cada iteración de pruebas realizadas:

- **En la primera iteración** se detectaron 20 No Conformidades (NC), de las cuales 14 fueron por error de aplicación y 6 en la documentación.
- **En la segunda iteración** se detectaron 6 NC, entre ellas 1 por error de aplicación y 5 en la documentación.
- **En la tercera iteración** se detectaron 2 NC en la documentación.

Todas las pruebas realizadas y las métricas de calidad aplicadas al diseño del componente han arrojado resultados satisfactorios y se puede afirmar que el sistema cumple con los estándares de calidad de diseño requeridos y que los requisitos funcionales han sido correctamente implementados.

### 3.6.3 Ejecución de pruebas de resistencia y rendimiento

Por las características que posee la solución implementada y al recibir una notable cantidad de peticiones por parte de los usuarios o sistemas que usen los servicios de la misma se hace

## Capítulo 3: Implementación y Prueba

---

necesario realizar un estudio de la resistencia y rendimiento de las diferentes funcionalidades del sistema. En aras de reconocer las debilidades de la solución ante un gran flujo de peticiones, se decidió utilizar para las pruebas de carga y estrés la herramienta JMeter por las funcionalidades que presenta.

### **JMeter:**

Es una herramienta Java que permite realizar Pruebas de Rendimiento y Estrés sobre Aplicaciones Web. Entre las principales facilidades que permite se destacan, permite realizar test mediante los protocolos FTP y HTTP, así como mediante JDBC<sup>11</sup>, LDAP<sup>12</sup> y SOAP/XML<sup>13</sup>. Permite la ejecución de pruebas distribuidas entre distintos ordenadores, para realizar pruebas de rendimiento. Además es posible activar o desactivar una parte del test, lo que es muy útil cuando se está desarrollando un test largo, y se desea deshabilitar ciertas partes iniciales que sean muy tediosas o largas. Este permite además almacenar los resultados de la prueba y generar gráficos que representan los aspectos que se han probado. Permite integración con el navegador Web y es multiplataforma. (41)

A continuación se describe la ejecución de las pruebas utilizando la herramienta JMeter, para las cuales se seleccionaron 4 pruebas con alto nivel de importancia, estas son:

- Cargar Configuración del workflow de la Base de Datos.
- Cargar Configuración del workflow de un fichero XML.
- Guardar la configuración del workflow en la Base de Datos.
- Guardar la configuración del workflow en un XML.

Las pruebas se realizaron en una estación de trabajo con las siguientes características:

- Sistema Operativo Debian 6.0.
- Servidor de Aplicaciones Apache 2.2.
- Marco de trabajo Symfony 1.4.16.
- Gestor y cliente de Base de Datos Oracle 11g.

---

<sup>11</sup> **JDBC:** Interfaz de programación que permite la ejecución de operaciones sobre la Base de Datos desde el lenguaje Java.

<sup>12</sup> **LDAP:** Protocolo que permite el acceso a un servicio de directorio ordenado para buscar información en la red.

<sup>13</sup> **SOAP-XML:** Protocolo para el intercambio de datos mediante el formato XML.



## Capítulo 3: Implementación y Prueba

- Microprocesador Intel Core 2 Quad.
- Memoria de sistema 18 Gb.

### Elección de las variables adecuadas:

Para llevar a cabo las pruebas de rendimiento y estrés se utilizarán 300 hilos de ejecución con un tiempo de espera entre cada una de 1 segundo, dada la alta concurrencia que podrá tener la aplicación. Las ejecuciones se efectuaron en todos los casos de forma reiterativa en dos ocasiones debido a la posibilidad latente de que esto ocurra en el entorno real de la aplicación. Para los 4 casos se analizarán las variables de salida, verificando los valores obtenidos y comprobando la velocidad de respuesta del servidor de aplicaciones.

### Resultados de las pruebas aplicadas:

Funcionalidad	Muestras <sup>14</sup>	Media <sup>15</sup>	Mínimo <sup>16</sup>	Máximo <sup>17</sup>
Cargar Workflow de un fichero XML	300	48669	27378	70204
Cargar Workflow de la BD	300	60759	45463	77192
Guardar Workflow en un fichero XML	300	2091	1443	2941
Guardar Workflow en la BD	300	2642	1715	3725
<b>TOTAL</b>	<b>1200</b>	<b>28540</b>	<b>1443</b>	<b>77192</b>

Tabla 12: Resultados de las pruebas de carga y estrés con 30 peticiones.

Las pruebas efectuadas tuvieron lugar con un 0% de errores, arrojando como resultado que la funcionalidad Cargar Workflow de un fichero XML, obtuvo una media de ejecución por debajo de la media de Cargar Workflow de la Base de Datos, lo cual evidencia lo inconveniente de cargar dicha configuración de un servidor en el caso actual en que la configuración del workflow

<sup>14</sup> **Muestras:** Cantidad de peticiones realizadas a la aplicación.

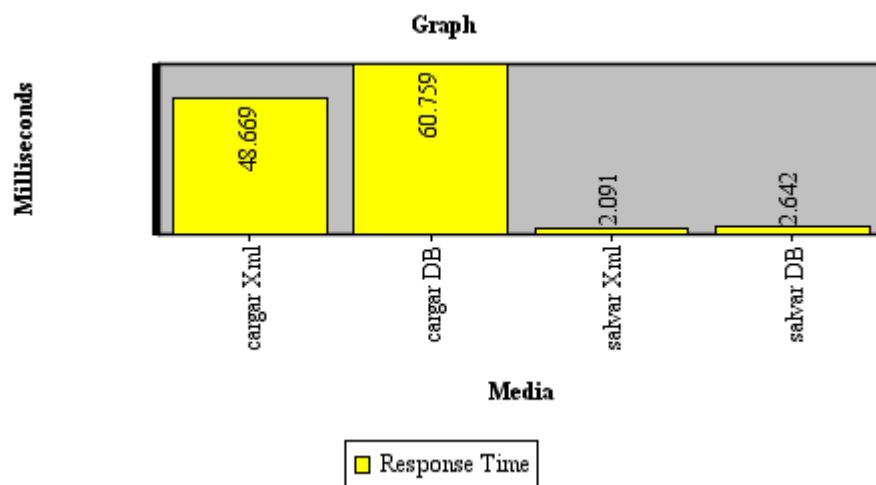
<sup>15</sup> **Media:** Media del tiempo de respuesta para las peticiones en milisegundos

<sup>16</sup> **Mínimo:** Tiempo mínimo en dar respuesta a las peticiones en milisegundos

<sup>17</sup> **Máximo:** Tiempo máximo en dar respuesta a las peticiones en milisegundos

## Capítulo 3: Implementación y Prueba

no es muy extensa. Por otro lado las funcionalidades Cargar y Guardar Workflow en la Base de Datos arrojaron resultados negativos en cuanto a rendimiento y estrés, lo cual entra en contradicción con el RNF #6 y avala la idea anteriormente mencionada, de cargar y guardar dichas configuraciones de un fichero XML. En ambos casos sus medias estuvieron por encima de las medias de sus equivalentes, dígase cargar la configuración y guardarla mediante un fichero XML, lo cual se evidencia mediante el siguiente gráfico.



**Ilustración 25 :** Gráfico del resultado de las pruebas de carga y estrés con 30 peticiones.

### 3.7 Conclusiones del Capítulo

En este capítulo se realizó la implementación de la solución propuesta donde se realizaron los diagramas de despliegue y de componentes de la aplicación, así como la descripción de las clases persistentes y los patrones de diseño utilizados en la codificación. Además se efectuaron pruebas de caja negra al módulo Configuración y Notificación lo que permitió detectar tras 3 iteraciones de realización de pruebas funcionales un total de 28 no conformidades las cuales fueron resueltas permitiendo cumplir con los requisitos capturados.

## **Conclusiones**

Como resultado de la investigación realizada en el presente trabajo, se logró una mejor comprensión acerca de los procesos de negocios asociados a la Configuración y el envío de Notificaciones en distintas aplicaciones informáticas, así como la valoración de algunas soluciones existentes a nivel internacional lo que permitió reconocer la no existencia de una solución informática que responda cabalmente a lo que en realidad requiere este proceso en las entidades cubanas.

Se obtuvo el modelo de análisis y diseño del módulo Configuración y Notificación, así como una propuesta del modelo de datos que sustentará las clases persistentes del Módulo, mostrando los datos, sus relaciones y las restricciones que deben cumplirse entre ellos facilitando de esta manera la implementación de la solución propuesta.

Mediante la aplicación de métodos y métricas usados en la validación de la solución, se obtuvieron resultados satisfactorios en el análisis y diseño propuesto, lo que garantizó la calidad del mismo.

Se destaca con la solución propuesta, la incorporación de principios por los que se mide la factibilidad del diseño de una aplicación informática. La utilización de patrones y métricas es un ejemplo fehaciente de ello.

Se demostró que la solución que se propone proporciona un soporte confiable para el almacenamiento y la integridad de la información que se genera durante la ejecución del proceso de Configuración y envío de Notificaciones, asegurando la organización, compartimentación y entendimiento común de la información procesada.

## **Recomendaciones**

Utilizar la especificación de requisitos de software obtenida para continuar el desarrollo de los futuros procesos de configuración en la Ventanilla Única.

Liberar la aplicación por el equipo de calidad de software del centro.

## Referencias Bibliográficas

1. Aduana de Cuba. [En línea] 2008. [Citado el: 8 de 12 de 2011.] [www.aduana.co.cu](http://www.aduana.co.cu).
2. VUCE Perú. [En línea] 2010. [Citado el: 8 de 12 de 2010.] [www.vuce.gob.pe](http://www.vuce.gob.pe).
3. *Recomendacion No. 33 UN/CEFACT Naciones Unidas para la Facilitación del Comercio y el Negocio Electrónico*. UN/CEFACT. 2004.
4. VUCEM. [En línea] 2011. [Citado el: 7 de 12 de 2011.] [www.ventanillaunica.gob.mx/vucem](http://www.ventanillaunica.gob.mx/vucem).
5. **Klimovsky, Gregorio**. Las desventuras del conocimiento científico. s.l. : A-Z editora, 1997.
6. Ventanilla Única de Suecia. [En línea] 2011. [www.tullverket.se](http://www.tullverket.se).
7. ITDS Ventanilla Única EEUU. [En línea] 2011. [www.itds.gov](http://www.itds.gov).
8. **Garlan, D y Shaw, M**. *Software Architecture: Perspectives on an Emerging Discipline*. 1996.
9. **Perry, D y Wolf, A**. *Foundations for the Study of Software Architecture*. 1992.
10. **Cobo, Jose A**. *Línea Base Arquitectónica para el Polo Sistemas Tributarios y de Aduanas*. 2008.
11. **A., CHRISTOPHER**. *A Pattern Language: Towns, Buildings, Construction*. 1977.
12. **Buschmann, F**. *Pattern-Oriented Software Architecture: A System of Patterns*. 1996.
13. **Lutz, Mark**. *Learning Python, 4th Edition*. 2010.
14. **Tucumán, Universidad Tecnológica Nacional-Facultad Regional**. Lenguajes de Programación. [En línea] 2000. [Citado el: 15 de Enero de 2011.]
15. PHP.net. [En línea] 2001. [Citado el: 15 de 1 de 2011.] [php.net/manual/en/faq.general.php](http://php.net/manual/en/faq.general.php).
16. **Potencier, Fabien**. *Symfony la guía definitiva*. 2008.
17. **Burker, Coyner B &**. *Extreme Programming*. 2005.
18. **Matraspa, Verdecia D I &**. *Procedimiento para el modelado de los procesos del Negocio y la Captura de Requisitos del Software a la medida desarrollados en la UCI*. 2008.
19. Rational. [En línea] 2007. [www.rational.com.ar](http://www.rational.com.ar).
20. Buenas Tareas. [En línea] [www.buenastareas.com/Methodologias-de-Desarrollo-de-Software](http://www.buenastareas.com/Methodologias-de-Desarrollo-de-Software).
21. UN/CEFACT. *II Encuentro Regional Latinoamericano y del Caribe sobre las VUCE*. 2010.
22. **Falgueras, Benet Campderrich**. *Ingeniería del Software*. 2003.
23. **Escalona, MJ. y Koch, N**. *Ingeniería de Requisitos en Aplicaciones para la Web: Un estudio comparativo*. . España y Alemania : s.n., 2002.
24. **Menendez, Abg Juan Jose Gaviria**. *Ventanilla Única Ecuatoriana*. s.l. : Federacion Ecuatoriana de Agentes de Aduana.
25. The Korea International Trade Association. [En línea] 2000. [Citado el: 14 de 2 de 2012.] <http://www.kita.net>.
26. Unipass-Korea Custom Services. [En línea] 2009. [Citado el: 14 de 2 de 2012.] <http://www.unipass.or.kr>.
27. **Bizagi**. bizagi. [En línea] 2011. [Citado el: 2 de 3 de 2012.] <http://www.bizagi.com/docs/BPMNbyExampleSPA.pdf>.
28. **Group, Object Management**. bpmn. [En línea] omg, 1997. [Citado el: 2 de 3 de 2012.] <http://www.bpmn.org/>.
29. **Beatriz Ayala, Claudia Marcela Ramírez, Lina María Ocampo**. *La Ingeniería de Requerimientos aplicada al desarrollo de sistemas de información*. Manizales, Colombia : s.n., 2011.

## Referencias Bibliográficas

---

30. **Sommerville, Ian.** *Ingeniería del Software*. Enero, 2005.
31. **Driggs, Idalberto Carlos Alonso.** *Modelo de Desarrollo del Departamento de Soluciones para la Aduana*. Universidad de las Ciencias Informáticas : s.n., 2012.
32. eZ Component. [En línea] 2007. eZComponents.org.
33. ECURED. [En línea] 16 de Abril de 2012. www.ecured.cu.
34. **Osmar Leyet Fernandez, Ismel Rodriguez Lorenzo.** *Desarrollo de una herramienta generadora de ficheros de mapeo para la persistencia de esquemas de objetos relacionales basada en NHibernate*. Universidad de las Ciencias Informáticas, La Habana : s.n., 2008.
35. **Laurencio Pérez, Daniuska y Diéguez García, Yunieski.** *Sistema de Gestión de la Trayectoria Productiva de cada estudiante en la facultad I*. Universidad de las Ciencias Informáticas La Habana : s.n., 2009.
36. **Sánchez, Elizabeth Quintas.** *Procedimiento para la realización de pruebas de unidad dentro del proyecto Sistema*. Universidad de las Ciencias Informáticas, La Habana : s.n., 2010.
37. **Fabien Potencier, François Zaninotto.** *Guía Definitiva de Symfony 1.2*. 2008.
38. **Potencier, Fabien.** *La Guía Definitiva de Symfony 1.4*. 2009.
39. **Ramírez, Jaime.** *Unidad de Programación. Métodos de pruebas del Software*. .
40. —. *Unidad de Programación. Métodos de pruebas del Software*.
41. **Anay Caridad Barban Frómata, Paula Zenaida Hernández Figueredo.** *Procedimiento para pruebas de rendimiento de Carga y Estrés al Sistema Único de Aduanas*. Universidad de las Ciencias Informáticas, La Habana : s.n., 2009.

## **Glosario de Términos**

**Proceso:** Define “quién” está haciendo “qué”, “cuándo” y “cómo” para alcanzar un determinado objetivo.

**Proceso de negocio:** Es un conjunto estructurado de actividades, diseñado para producir una salida determinada o lograr un objetivo. Los procesos describen cómo es realizado el trabajo en la empresa y se caracterizan por ser observables, medibles, mejorables y repetitivos.

**Declarante:** Toda persona natural o jurídica que hace una declaración en la Aduana o en nombre de la cual esta declaración es hecha.

**EDIFACT:** Intercambio electrónico de datos para la Administración, el Comercio y el Transporte.

**XML:** Lenguaje de marcado extensible.

**UNI-PASS:** Nombre del sistema desarrollado por Corea. Convenio con Ecuador.

**ECU-PASS:** Nombre del sistema desarrollado por Corea para Ecuador.

**E-Docs:** Documentos Electrónicos.

**BPMI:** Business Process Management Initiative.

**OMG:** Object Management Group.

**QT:** Biblioteca multiplataforma para desarrollar interfaces gráficas de usuario.

**MVC:** Patrón arquitectónico Modelo Vista Controlador.

**Interfaz:** La interfaz es lo que “media”, lo que facilita la comunicación, la interacción entre dos sistemas de diferente naturaleza, típicamente el ser humano y una máquina como el computador. Esto implica, además que se trata de un sistema de traducción, ya que los dos “hablan” lenguajes diferentes.

**Metodología:** Métodos de investigación que se siguen para alcanzar una gama de objetivos en una ciencia.

**Prototipos de interfaz usuario:** Presentación de la interfaz del producto que representa la funcionalidad contenida en los requisitos; de manera que permita que el usuario verifique que el sistema va a satisfacer sus necesidades.