

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

Facultad 3



Título: Tránsito y Transferencia para el Despacho
Comercial en la Aduana General
de la República de Cuba

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

Autor: Boris Enrique Ruiz Guerra

Tutores: Ing. Julio Cesar Bravo Rodríguez

Ing. Ernesto D. Vargas Allegue

Ciudad de La Habana, junio 16 de 2011

“Año 53 de la Revolución”

Pensamiento

“Para lograr grandes cosas
primero se deben tener grandes
sueños”

Conrad Hilton

Declaración de Autoría

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Boris Enrique Ruiz Guerra

Julio Cesar Bravo Rodríguez

Firma del Autor

Firma del Tutor

Datos de Contacto

Ing. Julio Cesar Bravo Rodríguez:

Graduado de Ingeniero en Ciencias Informáticas por la UCI. Posee dos años de experiencia. Sin categoría docente. Jefe de proyecto del subsistema Despacho Comercial y diseñador principal del proyecto GINA.

Agradecimientos

Agradesco primeramente a mi familia, y cuando hablo de familia no solo me refiero a la que lleva mi sangre por las venas también hablo de todos aquellos que pertenecen a ella, atados solo por los lazos del Amor. A mi familia por darme la oportunidad de la vida, por estar conmigo en todo momento, buenas malas. Por guiarme siempre hacia el buen camino, por enseñarme valores que separan al niño del hombre, por todo el amor y el cariño que han sembrado en mis entrañas. Por quererme así como soy, con mis virtudes y defectos por ser yo su nieto, su hijo, su hermano, su sobrino, su primo. Por cuidar de mí, por tantos dolores de cabeza y tantos desvelos gracias a mis travesuras y enfermedades. Estoy orgulloso de formar parte de ustedes.

Agradesco a mis amigos, a todos los que conocí en mis años de universidad y a todos los que deje detrás de un teléfono durante todo este tiempo, por ser complemento en mi vida, por brindarme tanta sabiduría, tanto cariño y amistad. Por darme el apoyo que en su momento necesité, por su preocupación sobre mi bienestar, por las risas los llantos lamentos y logros que hemos disfrutado juntos. Por ayudarme a ser una mejor persona, por darme la oportunidad de ser útil, y la oportunidad del perdón.

Agradecimientos

Agradesco a mi tutor por la paciencia que tuvo al tratar conmigo, siempre brindándome su apoyo y confianza.

Agradesco a todos mis compañeros de trabajo, a ese gran equipo que es el Proyecto Aduana, allí donde crecí como estudiante, como persona y como profesional, allí donde nace la humildad y la sencillez, por su total e incondicional apoyo para que pudiera salir adelante con mi trabajo diploma. Incontables las tantas personas que se sentaron conmigo para brindarme de su ayuda y conocimiento, sembrando responsabilidad y amistad.

Agradesco a la Revolución Cubana por darme la oportunidad de tener una educación y poder adquirir conocimientos para mi futura y pronta vida como profesional.

Dedicatoria

Quiero brindar mi dedicatoria principalmente y ante todo en esta vida, a mi Familia por su apoyo incondicional, no solo en mis años de universitario, sino por la constancia de toda una vida a mi lado, brindándome lo mejor de sí, para poder ser yo, una persona correcta y preparada.

A todas aquellas personas que han estado a mi lado durante los cinco años de estancia en la Universidad de las Ciencias Informáticas y que han sido mi punto de apoyo para seguir adelante.

A aquellos(as) que han creído en mí como persona, como amigo, como hijo, como hermano, como hombre, como amante.

Resumen

A la par del desarrollo del mundo, lo ha hecho también el comercio y la forma de llevarse a cabo. Es objetivo de la mayoría de los países, tratar de simplificar los trámites que se llevan a cabo por las partes implicadas en un intercambio de mercancías, sin dejar de mantener el control y administración. Con el surgimiento de las aduanas, se da cumplimiento a los objetivos anteriormente dichos, debido a que son el órgano responsable dentro de cada país de llevar el control, en materia de comercio. La Aduana General de la República de Cuba (AGR), que se encarga de poner en práctica los procedimientos aduaneros en las diferentes fronteras del país al cual representa. Dentro de los procedimientos que se llevan a cabo en la AGR se encuentran el despacho sobre los regímenes del Tránsito y la Transferencia, los cuales no son más que una serie de formalidades, que se tienen en cuenta para fomentar la administración sobre las mercancías que se comercializan. Estos regímenes, presentan soluciones en el Sistema Único de Aduana (SUA), que es actualmente el que está en pleno funcionamiento en la AGR, sistema que presenta dificultades de rendimiento, mantenimiento y funcionamiento, siendo las causas principales para que los especialistas del Centro de Informatización para la Dirección e Informatización de la Aduana (CADI), en conjunto con la Universidad de las Ciencias Informáticas (UCI), se centraran en la creación del Sistema de Gestión Integral de Aduana (GINA), específicamente un módulo que gestione los regímenes de Tránsito y Transferencia, eliminando los problemas antes dichos.

Como entrada para la solución del objetivo, se plantearon 7 requisitos funcionales, que fueron el punto de partida para el posterior diseño desarrollado. Se utilizaron diversas herramientas y metodologías, teniendo en cuenta las buenas prácticas utilizadas mundialmente, dentro de las que destaca como metodología de desarrollo RUP y Visual Paradigm para el modelado, así como la creación de un entorno Cliente-Servidor por medio de tecnología WEB, sobre la base del patrón arquitectónico Modelo Vista Controlador (MVC) y del lenguaje PHP, sin dejar de mencionar el importante uso de los frameworks Ext JS para el trabajo con las interfaces de usuario y Symfony como rector para el trabajo a nivel de servidor.

Luego de concluir el desarrollo de la solución se realizaron pruebas funcionales, así como la implantación del mismo en diferentes aduanas del país, brindando resultados satisfactorios.

Palabras claves: Tránsito, Transferencia, GINA, Aduana, Control

Tabla de Contenido

Introducción	1
Capítulo 1.Fundamentación Teórica	5
1.1 Introducción.....	5
1.2 Las aduanas	5
1.2.1 Otros conceptos.....	6
1.3 Principios	7
1.3.1 Sistemas informáticos en el mundo.....	8
1.3.2 Sistemas informáticos en Cuba.....	13
1.4 Descripción de la solución	14
1.5 Tendencias actuales y tecnologías empleadas.....	15
Metodología de desarrollo	15
Metodologías de diseño	17
1.5.1 Patrones de diseño	18
1.5.2 Patrones Arquitectónicos	20
1.5.3 Lenguajes de programación	21
1.5.4 Frameworks	26
1.5.4.1 Framework Ext JS.....	27
1.5.4.2 Framework Symfony	28
1.5.5 Entornos de Desarrollo Integrado (IDE)	30
1.5.6 Herramientas de Diseño	30
1.6 Gestor de Base de Datos.....	31
1.7 Conclusiones parciales	33
Capítulo 2.Diseño e Implementación	34
2.1 Introducción.....	34
2.2 Modelo del Diseño	34
2.2.1 Requerimientos.....	35
2.2.2 Diagrama de Paquetes	35

Tabla de Contenido

2.2.3 Diagrama de Clases del Sistema.....	37
2.2.4 Diagrama de Interacción del Diseño	39
2.2.5 Diseño de la Base de Datos	42
2.2.6 Métricas para la validación del diseño	45
2.3 Modelo de Implementación.....	51
2.3.1 Estándar de Codificación.....	51
2.3.2 Tratamiento de Errores	53
2.3.3 Comunicación entre Capas	53
2.3.4 Diagrama de Componentes	58
2.4 Conclusiones Parciales.....	59
Capítulo 3. Constatación de la Solución.....	60
3.1 Introducción.....	60
3.2 Pruebas de Software	60
3.2.1 Pruebas Funcionales o de Caja Negra	61
3.3 Impacto de la Solución Obtenida.....	65
3.4 Conclusiones Parciales.....	66
Conclusiones	67
Recomendaciones	68
Bibliografía.....	69
Glosario de Términos	73

Introducción

Las exportaciones e importaciones realizadas principalmente entre países, implican el flujo de bienes y servicios. En la medida que se ha desarrollado el mundo se ha ido incrementado el comercio, lo cual ha conllevado a crear procesos para la realización y mejora de ese tipo de procedimientos, así como la toma de medidas para regular el tráfico y brindar protección a las partes implicadas. Aparecen las aduanas como punto de control las cuales cumplen funciones claves para el desarrollo del país, ya que tiene un rol preponderante en materia de comercio exterior, especialmente, en la facilitación y agilización de las operaciones de importación y exportación, a través de la simplificación de trámites y procesos.

Dentro de los procesos importantes que se llevan a cabo en las Aduanas se encuentran el Tránsito y la Transferencia, donde el conjunto de actos y formalidades que presentan los mismos es tratado por cada aduana de manera independiente, según las necesidades y políticas de los gobiernos, para así tener un control regulador y eficiente de la entrada y salida de las mercancías.

La Aduana General de la República es la encargada de regular el control aduanero aplicable a la entrada, el tránsito, el cabotaje, el trasbordo, el depósito y la salida del territorio nacional de mercancías, viajeros y sus equipajes, bienes y valores sujetos a regulaciones especiales, incluidas la flora y la fauna protegidas, así como los medios en que se transporten, además forma parte de la Administración Central del Estado y se subordina al Consejo de Ministros. Cuenta con regímenes para el tratamiento del Tránsito y la Transferencia, actividades que han sido mejoradas debido al desarrollo que ha ido alcanzando la informática y las comunicaciones en el país. La Universidad de Ciencias Informáticas lleva el liderazgo en la producción de software para la informatización de todo el territorio cubano, actualmente la AGR está en un proceso de desarrollo directamente vinculado con la UCI realizando la automatización de todos sus sectores, creando el Sistema de Gestión Integral de Aduanas, al cual todavía se le están incluyendo módulos para la gestión del negocio aduanero. La AGR trabaja sobre el sistema SUA, el cual realiza los procesos de Tránsito y Transferencia. El bajo rendimiento es una de las principales dificultades que presenta este sistema, y se pone de manifiesto debido a que en el momento de realizar la solicitud de despacho de una mercancía en tránsito, la aduana solicita información que no se conoce, sobre el destino de la carga, trayendo como consecuencia que al momento de la extracción e incluso después de la misma, sea necesaria la modificación de la solicitud de despacho, provocando el atraso y pérdida de

Introducción

tiempo. El sistema realiza las operaciones por pantallas poco orientadas al uso, así como validaciones a nivel de base de datos por medio de procedimientos de almacenado, trayendo como consecuencia que en caso de mantenimiento, sea engorroso y difícil. Toda la información que se utiliza para el despacho de la mercancía, es introducida manualmente, no se cuenta con la facilidad de procesarla mediante ficheros.

Luego de haber realizado un análisis detallado de los problemas existentes y la captura de requisitos, surge el siguiente **Problema a resolver**:

¿Cómo gestionar los procesos asociados al módulo Tránsito y Transferencia contribuyendo a la mejora del rendimiento, el mantenimiento y funcionamiento?

Como **Objeto de estudio** se define las etapas de diseño e implementación.

Presenta un **Campo de acción** que se enmarca en el desarrollo de las etapas de diseño e implementación del módulo Tránsito y Transferencia del sistema GINA de la Aduana General de la República de Cuba.

Como **Objetivo General**: Diseñar e implementar el módulo de Tránsito y Transferencia del sistema GINA de la Aduana General de la República de Cuba contribuyendo a la mejora del rendimiento, mantenimiento y funcionamiento.

Derivándose algunos **objetivos específicos**:

1. Realizar el estudio del estado del arte.
2. Diseñar e implementar el módulo teniendo en cuenta las herramientas definidas por el centro.
3. Constatar la solución obtenida.

Para dar cumplimiento a los objetivos antes planteados se desglosan una serie de **tareas de investigación**:

- Evaluación de las herramientas a utilizar.
- Evaluación de las técnicas y patrones de diseño de aplicaciones.

Introducción

- Evaluación de los estándares de codificación definidos en el proyecto.
- Elaboración del modelo de dato del módulo Tránsito y Transferencia.
- Elaboración del diagrama de clases del diseño del módulo Tránsito y Transferencia.
- Elaboración de los diagramas de diseño de la solución del módulo Tránsito y Transferencia.
- Implementación de los diagramas de diseño realizado para el módulo Tránsito y Transferencia.
- Elaboración del diagrama de componentes para el módulo Tránsito y Transferencia.
- Realización de pruebas funcionales al sistema orientadas a la lista chequeo.
- Documentación de las pruebas realizadas.

Como **idea a defender** se plantea que con el desarrollo del módulo de Tránsito y Transferencia del sistema GINA de la Aduana General de la República de Cuba se contribuirá a la mejora del rendimiento, mantenimiento y funcionamiento del módulo actual, perteneciente al sistema SUA.

A continuación se plantean un conjunto de **métodos investigativos** que se utilizarán para brindar solución a los objetivos dichos:

- Análisis de la Literatura según el libro Thesis Projects del año 2008, creado por M. Berndtsson y colaboradores: se refiere a un examen sistemático de un problema determinado, por medio de un análisis de fuentes publicadas, llevado a cabo con un propósito específico en mente. Se utilizará este método para realizar el estudio a profundidad de las herramientas de trabajo que se usarán; en la realización del estudio del estado del arte; también para conocer a plenitud los artefactos que servirán para el diseño incluyendo las técnicas y patrones; la profundización en el lenguaje de programación y los estándares de codificación definidos y en la búsqueda de otros sistemas que realicen funciones de la misma índole.
- Modelado: como su nombre lo indica este método se encarga de realizar una reproducción simplificada de la realidad, es uno de los más importantes en el campo de la construcción de

Introducción

software y se utiliza para la creación de todos los artefactos necesarios del diseño; el modelo de datos, los diagramas de clases, diagrama de componentes, el diseño de la solución del módulo.

- Implementación según el libro Thesis Projects del año 2008, creado por M. Berndtsson y colaboradores: El objetivo de este método, es demostrar que la solución tiene ciertas propiedades, o que (bajo ciertas condiciones) se comporta de una manera específica. Se emplea en la realización de la implementación de todo el diseño que se modeló anteriormente y así dejar plasmado las ventajas obtenidas.
- Histórico – Lógico: combinación del método histórico y el lógico, donde el método lógico descubre las leyes fundamentales de los fenómenos basándose sobre los datos que le proporciona el método histórico. Se pone en práctica en la realización del estudio del estado del arte, vinculándose con la evaluación de las herramientas a utilizar, técnicas y patrones de diseño y de otros softwares que realizan los procesos del tránsito y la transferencia.

El presente trabajo está estructurado de la siguiente manera:

Capítulo 1. Fundamentación Teórica: este capítulo se introducirá en el estudio del estado del arte, se presentarán todos los conceptos esenciales vinculados con el Despacho Comercial a nivel mundial y nacional, específicamente el Tránsito y la Transferencia, las herramientas, tendencias y tecnologías actuales que se emplearán en la construcción del módulo y aspectos esenciales que sirven de soporte del mismo.

Capítulo 2. Diseño e Implementación del sistema: ya en este capítulo se penetra más en el mundo de la informática, se presentan los diferentes artefactos que se obtienen del diseño realizado, también se detallará la implementación desarrollada, el diagrama de clases, el modelo de datos, diagramas de secuencia, en general toda una descripción del diseño e implementación del módulo Tránsito y Transferencia.

Capítulo 3: Constatación de la solución: este capítulo contendrá todos los resultados obtenidos de las diferentes comprobaciones que se le realizaron a la solución que se obtuvo.

Capítulo 1. Fundamentación Teórica

1.1 Introducción

En el capítulo se realizará un estudio del estado del arte sobre los sistemas que se utilizan para realizar el Despacho Comercial principalmente los regímenes de Tránsito y Transferencia. Se plasmará los resultados de la investigación ejecutada sobre las tecnologías y herramientas a utilizar para el desarrollo de la solución planteada, incluyendo diferentes patrones y metodologías de diseño.

1.2 Las aduanas

La Aduana es el organismo responsable de la aplicación de la Legislación Aduanera y del control de la recaudación de los derechos de Aduana y demás tributos; encargados de aplicar en lo que concierne la legislación sobre comercio exterior, generar las estadísticas que ese tráfico produce y ejercer las demás funciones que las leyes le encomiendan.(1)

En el contexto mundial las aduanas son entidades, muy antiguas, que fueron creadas para el control del intercambio comercial con el exterior, las naciones modernas desarrollan sus actividades a través de las aduanas, en el sentido de regular la entrada y salida de mercancías, en /o su territorio, para alcanzar principalmente sus objetivos económicos, es decir, son como las llaves que abren o cierran el comercio exterior; en consecuencia es innegable la gran importancia que ellas tienen en la actividad económica de cada país.(2)

No es función de la aduana ni de la administración que gira alrededor de ella, el establecer trabas al Comercio Internacional, más bien todo lo contrario. En las diferentes fases que abarca, las operaciones, como embarque de productos, documentación, liquidación de derechos e impuestos, valoración, clasificación, entre otros, su objetivo es facilitar el comercio, hacerlo más práctico, expeditivo y funcional.(2)

Capítulo 1. Fundamentación Teórica

1.2.1 Otros conceptos

Despacho

El Despacho en las aduanas significa Cumplimiento de las formalidades aduaneras necesarias para importar y exportar las mercancías o someterlas a otros regímenes, operaciones o destinos aduaneros.(1)

Comercio

El término comercio proviene del concepto latino *commercium* y se refiere a la negociación que se entabla al comprar o vender géneros y mercancías. También se denomina comercio a la tienda, almacén o establecimiento comercial, y al conjunto o clase de comerciantes.(3)

Despacho Comercial

Por tanto el despacho comercial son las formalidades aduaneras que se realizan cuando se comercializan mercancías, tanto en importaciones como exportaciones.

Régimen Aduanero

Se le llama régimen aduanero al tratamiento aplicable a las mercancías sometidas al control de la aduana, de acuerdo con la normativa aduanera, según la naturaleza y objetivos de la operación. A través de los regímenes aduaneros se determina el destino de las mercancías y mediante su utilización; el tratamiento fiscal aplicable a las mismas.(1)

Dentro de los regímenes que existen en la Aduana General de la República de Cuba se encuentra el Tránsito Aduanero. Se entiende por tránsito de mercancías el régimen bajo el cual son transportadas las mercancías de una aduana a otra del territorio nacional bajo control aduanero.

Se consideran en tránsito aduanero internacional las mercancías que circulan de una aduana de entrada a una aduana de salida; de una aduana de entrada a una aduana interior; de una aduana interior a una aduana de salida; y como tránsito aduanero nacional las mercancías que circulan de una aduana interior a una aduana interior.

Capítulo 1. Fundamentación Teórica

Se entiende por Transferencia de mercancías el régimen bajo el cual se transportan las mercancías de un depósito de una aduana hacia el depósito de otra aduana.

1.3 Principios

En las primitivas organizaciones sociales se encuentran ya los impuestos (directos), tanto en su forma personal, como en el servicio militar, tanto en su forma real (parte del botín que se adjudica al jefe de la tribu). Posteriormente aparecen los tributos en especie, como la capitación, y los tributos sobre los rendimientos de la agricultura y de la ganadería (diezmos). Mucho después, cuando crecen las necesidades del Estado, el impuesto adopta la forma indirecta. Entre otros impuestos indirectos, el de aduana parece haber sido conocido en la India, así como en Persia y Egipto. En Grecia existieron, junto con el de capitación (sobre los extranjeros), el de consumo, sobre las ventas.(4)

Las aduanas existieron en todos los pueblos de la antigüedad, según la importancia del comercio de cada ciudad, con la finalidad de controlar la entrada y salida de mercancías, y como forma de recaudar fondos.(5)

En Cuba, la Aduana constituye un órgano de control en la frontera y de fiscalización en la actividad vinculada al comercio exterior.(6)

Entre sus misiones está garantizar la seguridad y protección de la sociedad socialista y de la economía nacional, así como la recaudación fiscal y la emisión de las estadísticas del comercio exterior, a través del cumplimiento de las políticas estatales de competencia aduanera para el tráfico internacional de viajeros, mercancías y medios de transporte.(6)

La Aduana cubana suscribió en 1995 el Convenio Internacional para la Simplificación y Armonización de los Regímenes Aduaneros (Convenio de Kyoto) y en 2009, ratificó el Convenio de Kyoto Revisado, siendo Cuba el único país signatario de Latinoamérica.(6)

Paralelamente y como parte del proceso de modernización emprendido, se han adoptado un grupo de medidas para la facilitación y rapidez del despacho mercantil tales como la autovaloración y autoliquidación por el propio declarante, inspecciones de origen y destino, otorgamiento de facilidades al

Capítulo 1. Fundamentación Teórica

despacho y pago mediante convenios, sistemas de selectividad automatizada y la calificación y actualización en esa materia.(6)

1.3.1 Sistemas informáticos en el mundo

En la actualidad mundial se evidenció la presencia de innumerables aduanas que realizan la gestión de sus procesos mediante sistemas informáticos logrando así la automatización de los mismos. A continuación se presentaran varios ejemplos de estos sistemas.

Sistemas Informáticos DDS



Los Sistemas Informáticos DDS son una empresa que se especializa en prestar servicio. Aparece desde el 2003, ofreciendo soluciones empresariales y profesionales a cualquier tipo de clientes, no importa si se trata de un estudio contable, un comercio minorista, una pyme o una gran empresa. A diferencia de otras empresas, ofrecen soluciones que no son mágicas, en otras palabras no intentan vender lo que quieren, sino lo que al cliente le sirve, venden un sistema adaptado a los deseos del cliente.

Dentro de las soluciones que presentan se encuentra una línea que provee soluciones para el comercio exterior. Estos son sistemas de gestión que permiten llevar el control de trámites y mercaderías en tránsito, tanto para la importación como para la exportación.(7)

Dentro de los módulos que poseen se encuentran:

Módulo de sistema de gestión administrativa para despachantes de aduana: facturación de ventas y compras, manejo de cuentas corrientes, tesorería (caja y bancos), contabilidad integrada, administración

Capítulo 1. Fundamentación Teórica

de anticipos, gastos por carpeta, informes para el análisis comercial, pagos por cuenta y orden, importación de información desde el sistema María, y muchas utilidades más.(7)

Módulo de gestión de despachos pre-María: este software permite ingresar despachos en forma veloz y transferirlos a cualquier terminal María. Almacena mercaderías con su clasificación; emite formularios DGI 7366, sin necesidad de volver a cargar la información, y posibilita el ingreso de ítems de acuerdo al orden de facturación con el agrupamiento automático de los mismos. También permite realizar el alta automática del despacho con información de las facturas remitidas por mail. Emite hojas de valor por impresoras láser.(7)

Módulo de sistema de presupuestos: genera rápidamente presupuestos para actividades de importación y exportación, y remítalos por correo electrónico.(7)

Módulo de trazabilidad de operaciones: monitorea el estado de las operaciones, y envía dicha información a sus clientes por email de manera automática.(7)

Módulo de consulta de despachos oficializados: consulta los despachos oficializados, con un amplio rango de filtros por los diversos campos. Las destinaciones oficializadas se ingresan automáticamente desde el sistema María.(7)

Aunque Sistemas Informáticos DDS presenta buenas soluciones referentes al comercio exterior, no cumple con las funcionalidades que se quieren desarrollar para resolver el problema del Tránsito y la Transferencia del Despacho Comercial de la Aduana General de la República de Cuba. Solo es posible lograr adquirir el sistema por medio de abono mensual, siendo el software privativo, alejándose de los intereses del país de migrar a plataforma libre para la adquisición de la independencia tecnológica.

SIDUNEA



SIDUNEA es un sistema automatizado de gestión de las operaciones aduaneras que cubre la mayor parte de trámites de comercio exterior. El sistema cubre el proceso de manifiestos, declaraciones de aduana,

Capítulo 1. Fundamentación Teórica

pagos, contabilidad, inventarios y regímenes suspensivos. Genera información detallada sobre transacciones de comercio exterior que pueden utilizarse para planificación y análisis económicos. (8)

El sistema SIDUNEA está escrito totalmente en Java, funciona sobre diferentes tipos de plataformas informáticas Linux, Solaris, AIX, HP/UX, Windows XP/NT, Maco cualquier lugar donde haya una máquina virtual Java y presenta un entorno cliente/servidor. Los datos de transacciones y controles se almacenan en un sistema de gestión de bases de datos relacionales (Oracle, Informix, Sybase, Interbase, SQL Server). Presenta capacidad para soportar una amplia gama de dispositivos, en cualquier momento y en cualquier lugar dentro de los mismos encontramos (PCs, PDAs, teléfonos móviles, Tablet PC) ultra-fino (para dispositivos móviles), los clientes ligeros (con los principales navegadores web) y gruesos (clientes independientes) Linux, Windows, Apple. Tiene en cuenta los códigos y normas internacionales relacionados con los trámites de aduanas establecidos por la ISO, OMA (Organización Mundial de Aduanas) y Naciones Unidas. SIDUNEA puede configurarse para poder adaptarse a las distintas necesidades nacionales como los regímenes individuales de aduanas, aranceles nacionales, regulación y legislación aduaneras. Se basa en el Intercambio Electrónico de Datos (EDI) entre usuarios externos y administraciones de aduanas utilizando las normas UN/EDIFACT.(8)

El software SIDUNEA presenta una amplia gama de soluciones persistentes que cubren la mayor parte de los trámites de comercio exterior. Fue por las características que presenta que se llegó a instalar en la Aduana General de la República de Cuba pero luego de realizárseles pruebas prácticas, se llegó a la conclusión que las funcionalidades que poseía no cumplían con todos los requerimientos que se necesitaban. También en la Aduana Nacional de Bolivia presentó numerosas fallas de seguridad y operatividad. Tanto desde el interior y exterior de la Aduana manipularon los datos: eliminado información, creando y borrando usuarios, y sustituyendo documentos de importación y exportación, todo esto se debe a que permite acceder al sistema a través de dispositivos inalámbricos, donde quiera que se encuentre por lo que tiene mayor rango de ataque. Además, hace poco la empresa de software estadounidense **Oracle** ha comprado a Sun Microsystems **JAVA** por lo que pasa a ser privativo y rompe con las políticas del país de migrar a software libre.

Capítulo 1. Fundamentación Teórica

KCS (Korea Custom Service)



El Sistema de Servicios Aduanales de Korea trabaja sobre la base de los sistemas **EDI (Intercambio Electrónico de Datos)**, logrando la transferencia por medios electrónicos de información comercial o de negocios. Presenta un eficaz Servicio de Aduanas mediante el establecimiento de un sistema de Liquidación preciso y rápido. Mejora la competitividad nacional al reducir el tiempo y costo requerido para el despacho de aduana, y la reducción de los costos logísticos. Mejora también los servicios públicos, ofreciendo la información relativa a la Administración de Aduanas de manera precisa y rápida.(9)

Dentro de los Subsistemas de Automatización de Aduanas que presenta se destacan:

Sistema de Liquidación de Exportación EDI

Es un sistema de tipo EDI que permite a las empresas de exportación, las empresas de corretaje de aduanas, presentar declaración de exportación en la aduana de una manera sin soporte de papel y recibir el resultado de la declaración electrónicamente.(9)

Sistema de Exportación de Carga EDI

Es un sistema tipo EDI que permite a las compañías navieras, los trazadores de líneas aéreas, agentes de presentar informes de manifiesto, su salida del puerto de una manera sin papeles y recibirlos resultados de la presentación, la autorización y el permiso de salida por vía electrónica.(9)

Sistema de Importación de Carga EDI

Es un sistema tipo EDI que permite a las compañías navieras, los trazadores de líneas aéreas, transportistas, almacenes generales de depósito a presentar los documentos necesarios para la manipulación de la carga como la llegada, descarga y transporte en condiciones de servidumbre, y recibir el resultado de dicha presentación como la aprobación y la información sobre la carga en un momento adecuado por vía electrónica.(9)

Capítulo 1. Fundamentación Teórica

Desarrollo del Sistema de Declaración de Importación sin papeles

Permite archivar en formato no duro copias de documentos necesarios para las declaraciones de importación a través de una red electrónica EDI gracias a la simplificación y normalización, así como la verificación electrónica de los requisitos de importación exigidos por otras agencias del gobierno en cuestión.(9)

El sistema KCS se destaca por la rapidez y calidad con que realiza todos los trámites de aduanas, principalmente en la importación y exportación de mercancías. Aproximadamente un 95% de las declaraciones de exportación e importación se realizan mediante un tratamiento informatizado que tiene como característica la no inspección física por parte de los funcionarios de aduanas; lo cual puede traer consigo a Cuba, huecos de seguridad en las mercancías comercializadas. Las formalidades llevadas a cabo por el KCS para la importación y exportación de mercancías no son compatibles en toda su magnitud con las llevadas a cabo por la Aduana General de la República de Cuba.

Sistema HM Revenue & Customs



El Sistema HM Revenue & Customs(CHIEF) se encarga de la gestión de los procesos aduanales del Reino Unido. La arquitectura de este sistema permite la conexión con seis sistemas comerciales independientes que sirven directamente a cientos de compañías aéreas, hangares de tránsito y los transportistas para registrar y rastrear el movimiento de mercancías en los puertos y aeropuertos. Usando el EDI como sistema de mensajes internos, CHIEF controla que los datos incluidos dentro de la declaración aduanera coincidan con el inventario mantenido en cada sistema de comercio. CHIEF corre sobre los servicios de plataforma de Fujitsu Super Nova usando el sistema operativo propietario Open VME. El sistema de base de datos, IDMSX, es jerárquico, facilitando el mantenimiento del sistema en el futuro. Presenta una interfaz de alto nivel entre el código de la aplicación y el entorno de la máquina.(10)

CHIEF soporta Interfaz Hombre-Computadora (HCI) y la interfaz de tráfico EDI, con enlaces dedicados de comunicaciones de alta velocidad a Proveedores de Sistema Comunitario (DEP) y las oficinas de aduanas en todo el Reino Unido.(10)

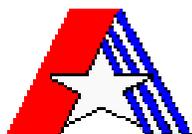
Capítulo 1. Fundamentación Teórica

La arquitectura del CHIEF contiene un sofisticado sistema de seguridad y características de resistencia para evitar el sabotaje o el acceso no autorizado y garantizar la continuidad de negocio eficaz.(10)

El sistema CHIEF presenta numerosas características que brindan ventajas referentes al comercio exterior. Facilita que los importadores, exportadores y transportistas completen las formalidades aduaneras necesarias con el mínimo de intervención manual, también permite a los principales puertos y aeropuertos realizar las operaciones de manera más eficiente y eficaz; pero presenta la peculiaridad que estas ventajas no son aplicadas a países fuera del Reino Unido, o sea, que no pueden ser tomadas en cuenta para la Aduana General de la República de Cuba. Trabaja el concepto de Tránsito de manera diferente, por lo que sería inservible. El sistema está montado sobre plataforma privativa, característica que va en contra de los principios seguidos por Cuba de migrar a software libre.

1.3.2 Sistemas informáticos en Cuba

Sistema Único de Aduana (SUA)



Un gran peso en la modernización de la Aduana cubana lo ha tenido el desarrollo de la informática y las comunicaciones, que unido a la consolidación de una amplia red propia de datos, ha permitido la automatización de la mayoría de los procesos administrativos y de control, todos con programas desarrollados completamente sobre software libre y por personal de la propia institución(6). El SUA es el sistema que presenta la AGR actualmente, automatiza los cinco procesos aduaneros (despacho comercial, no comercial, de viajeros, de medios de transporte y de postal y envío), posee una base de datos única y centralizada, a la que acceden en línea todas las aduanas del país. El módulo de Despacho Comercial que está desarrollado en dicho sistema cumple con la mayoría de las formalidades necesarias

Capítulo 1. Fundamentación Teórica

que se exigen en la AGR, pero también presenta los problemas dichos anteriormente sobre rendimiento, funcionamiento y mantenimiento.

Ninguno de los Sistemas anteriormente explicados cumple con las formalidades de negocio o de software que se necesitan en la realización de los procesos de Tránsito y Transferencia para el Despacho Comercial en la AGR y en muchas ocasiones tampoco son compatibles con las políticas que sigue el país de migrar a plataforma libre. Es por lo anterior expuesto que se necesita realizar un estudio y análisis para desarrollar una solución donde se ponga de manifiesto también las ventajas brindadas por estos sistemas. El uso de aplicaciones WEB es una característica importante que se aprecia y se pone en práctica en el GINA, para así tener un solo sistema en todo el país que brinde servicios a las distintas aduanas. Otro de los aportes obtenidos es el envío de información a través de medios electrónicos, para agilizar los procedimientos.

1.4 Descripción de la solución

La Universidad de las Ciencias Informáticas (UCI), en específico el Centro de Informatización de Gestión de Entidades (CEIGE), en conjunto con los especialistas de la Aduana General de la República de Cuba (AGR), están interesados en el desarrollo de un módulo que permita administrar y controlar las formalidades de los regímenes Tránsito y Transferencia pertenecientes al Despacho Comercial en todas las aduanas del país. Como parte de estas formalidades se encuentran el almacenamiento y validación de los datos de las Declaraciones de Mercancías (DM), así como el mantenimiento de los mismos una vez dentro del sistema. La información será introducida por medio de ficheros estandarizados en formato XML o por interfaz gráfica orientada al uso, garantizando el agrado y conformidad de los usuarios. El sistema tendrá interacción con otros subsistemas pertenecientes al GINA, brindará servicios a los mismos y permitirá separar la lógica del negocio del gestor de base de datos, garantizando el mantenimiento.

1.5 Tendencias actuales y tecnologías empleadas.

En la actualidad el uso de aplicaciones web¹ es un elemento que ha contribuido considerablemente al desarrollo de numerosas empresas. Este tipo de aplicaciones presenta ventajas significativas en el ahorro de tiempo, compatibilidad y disponibilidad.

En este apartado se especifican las principales tecnologías empleadas en el desarrollo de la solución y sus principales características. No se adentrará en comparaciones que justifiquen por qué fueron escogidas, debido a que el Centro delegó esa responsabilidad anteriormente, recogiendo los resultados en el Documento Base de la Arquitectura.

Metodología de desarrollo

El Proceso Unificado Racional (Rational Unified Process en inglés, habitualmente resumido como RUP) es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos, debido a su principal característica es la organización que brinda y la documentación que genera. Es un marco de trabajo genérico que puede ser especializado para una gran variedad de software para distintas áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyectos

El RUP no es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización.

En el Centro para la Informatización de la Gestión de Entidades (CEIGE), se utiliza RUP con adaptaciones especiales para lograr aligerar el procedimiento y así convertirlo en una metodología ágil. Las plantillas están regidas por el programa de mejoras para alcanzar el nivel 2 en CMMI². El modelado se realiza por Procesos, escritos en notación BPMN para así lograr una comprensión fácil para todos los usuarios, tanto del negocio, como para los analistas implicados y los que realizan el diseño de la solución. BPMN crea un

¹**Aplicación web:** son aquellas aplicaciones que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una intranet mediante un navegador web.

²**CMMI:** modelo de calidad del software que clasifica las empresas en niveles de madurez

Capítulo 1. Fundamentación Teórica

puente estandarizado entre los procesos de negocio, diseño e implementación. Los artefactos a obtener o muchos han variado u otros se han eliminado.

Esta metodología constituye una guía rectora que decide quién hace qué, cuándo y cómo lo hace. RUP dirige las actividades que se realizan por rol, les da un orden, define qué artefactos deberían ser desarrollados y puede además monitorear y medir los productos y actividades de un proyecto, teniendo en cuenta la calidad del producto final. RUP se caracteriza por dividir el ciclo de vida de la producción del software en 4 fases:

1. **Inicio o Conceptualización:** es donde se determina la visión del proyecto, o sea se comprende el entorno y se determina el alcance del producto.
2. **Elaboración:** en esta etapa se determinan los cimientos de la arquitectura y se analiza el dominio del problema.
3. **Construcción:** en esta fase se obtiene la capacidad operacional inicial del producto.
4. **Transición:** Se obtiene el release o liberación del producto y se pone en manos de los usuarios finales.

Además de esto cuenta con 9 flujos de trabajo, 6 de ingeniería y 3 de soporte los cuales son: Modelado del negocio, Requerimientos, Análisis y Diseño, Implementación, Prueba, Despliegue, Gestión de configuración y cambios, Gestión de proyectos y por último Entorno, donde los 3 últimos constituyen los flujos de soporte. Esta metodología puede especializarse para una gran variedad de sistemas software, para diferentes áreas de aplicaciones, diferentes tipos de organizaciones, diferentes niveles de amplitud y diferentes tamaños de proyectos.

Los elementos característicos del RUP son:

- **Actividades:** son los procesos que se llegan a realizar en cada iteración.
- **Trabajadores:** son las personas o entidades involucradas en cada proceso.
- **Artefactos:** un artefacto puede ser un modelo, o un elemento de modelo, un documento, en fin todo lo que puede ser generado en el proceso.
- **Flujo de Actividades:** secuencia de actividades realizadas por trabajadores que producen un resultado de valor observable.

Capítulo 1. Fundamentación Teórica

Para más información sobre los cambios realizados por el centro con la metodología RUP dirigirse a la tesis que lleva por título: “Procedimiento para la Ingeniería de Requisitos en el Departamento de Desarrollo de Soluciones para la Aduana del CEIGE.”(11)

Metodologías de diseño

Diseño Axiomático: Cuando se emplean axiomas para tomar decisiones. Este enfoque se basa principalmente en las necesidades funcionales del cliente. Se analizan sistemáticamente la transformación de las necesidades de cliente en requisitos, parámetros de diseño, y variables funcionales del proceso.

Diseño Orientado al Uso: Cuando el diseño se centra en las tareas asociadas al uso y sus objetivos. La usabilidad se refiere a la capacidad de un software de ser comprendido, aprendido, usado y ser atractivo para el usuario, en condiciones específicas de uso.

Diseño Centrado en el Usuario:(UCD User Center Design) es la metodología de diseño más habitual en el mundo. Este enfoque coloca todas las necesidades, deseos y limitaciones del usuario como núcleo del proceso de diseño. Por lo cual esta metodología conlleva a la investigación y análisis del usuario.(12)

Enfoque Ascendente: este enfoque se denomina ascendente, refiriéndose a que la computarización se implanta desde un nivel más bajo. El diseño ascendente se refiere a la identificación de aquellos procesos que necesitan computarizarse conforme vayan apareciendo, su análisis como sistemas y su codificación; o bien, la adquisición de paquetes de software para satisfacer el problema inmediato.

Enfoque descendente: Se refiere a ver una gran imagen del sistema y luego de explotarla en partes o subsistemas pequeños, tratando de lograr integrar a todos para que funcionen al unísono.

Desarrollo Modular: Se trata de un enfoque de concepción modular, se basa en una descomposición de la programación en fracciones lógicas y manejables. Enfatiza en las interfaces entre los módulos, no las ignoradas hasta el final del desarrollo del sistema. Cada módulo debe ser funcionalmente cohesivo, de manera que satisfaga sólo una función.

Capítulo 1. Fundamentación Teórica

De los ejemplos de metodologías de diseño presentadas anteriormente se evidencian con mayor énfasis el Diseño Orientado al Uso, el Diseño Centrado en el Usuario, Enfoque Ascendente, Descendente y Modular debido a que todo el trabajo que se desarrolla en el GINA por parte del CEIGE está basado en las necesidades del cliente, que es la AGR y para ello se trabaja en busca de la satisfacción completa, tratando no solo que el software cumpla con los requisitos necesarios sino también que sea agradable y fácil utilizar. Se realiza una planificación desde una visión global para la definición de la estructura del sistema que incluye subsistemas y módulos, así como la comunicación entre ellos, comenzando luego el desarrollo desde la raíz hasta la culminación del software completo.

1.5.1 Patrones de diseño

Un patrón de diseño es una solución estándar para un problema común de programación, una técnica para flexibilizar el código haciéndolo satisfacer ciertos criterios, una estructura de implementación que logra una finalidad determinada, manera práctica de describir aspectos de la organización de un programa.(13)

Los patrones brindan la facilidad de reutilizar el conocimiento de desarrolladores, clasificando y describiendo problemas y soluciones a problemas que surgen con frecuencia durante el desarrollo, por lo que se convierten en un historial que no ayuda a no cometer errores ya descritos.(13)

El sistema GINA emplea numerosos patrones de diseño, algunos de los más usados serán descritos a continuación:

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

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

1. B contiene A
2. B es una agregación (o composición) de A
3. B almacena A

Capítulo 1. Fundamentación Teórica

4. B tiene los datos de inicialización de A (datos que requiere su constructor)
5. B usa A.

A la hora de crear objetos se deben tener en cuenta las características de la clase.

Bajo Acoplamiento: El acoplamiento es una medida de la fuerza con que una clase está conectada a otras clases, con que las conoce y con que recurre a ellas. Una clase con bajo o débil acoplamiento no depende de muchas otras.

Una clase con alto o fuerte acoplamiento recurre a muchas otras. Este tipo de clases no es conveniente: presenta los siguientes problemas:

- Los cambios de las clases afines, provocan cambios locales.
- Son más difíciles de entender cuando están aisladas.
- Son más difíciles de reutilizar porque se requiere la presencia de otras clases de las que dependen.(14)

Controlador: Es un evento generado por actores externos. Se asocian con operaciones del sistema, como respuestas a los eventos del sistema, tal como se relacionan los mensajes y los métodos. Normalmente un controlador delega en otros objetos el trabajo que se necesita hacer; coordina o controla la actividad. No realiza mucho trabajo por sí mismo. (13)

Alta Cohesión: La cohesión es una medida de la fuerza con la que se relacionan las clases y el grado de focalización de las responsabilidades de un elemento. Cada elemento del diseño debe realizar una labor única dentro del sistema, no desempeñada por el resto de los elementos y auto-identificable, una clase con baja cohesión hace muchas cosas no relacionadas o hace demasiado trabajo.(14)

Decorador: Responde a la necesidad de añadir dinámicamente funcionalidad a un Objeto. Permitiendo no tener que crear sucesivas clases que hereden de la primera incorporando la nueva funcionalidad, sino otras que la implementan y se asocian a la primera.(15)

Su aplicación se basa principalmente en los siguientes casos:

- Añadir objetos individuales de forma dinámica y transparente
- Responsabilidades de un objeto pueden ser retiradas

Capítulo 1. Fundamentación Teórica

- Cuando la extensión mediante la herencia no es viable
- Hay una necesidad de extender la funcionalidad de una clase, pero no hay razones para extenderlo a través de la herencia.



Figura 1: Implementación del patrón Decorador por Symfony

1.5.2 Patrones Arquitectónicos

Son los que definen la estructura de un sistema software, los cuales a su vez se componen de subsistemas con sus responsabilidades, también tienen una serie de directivas para organizar los componentes del mismo sistema, con el objetivo de facilitar la tarea del diseño de tal sistema.

Modelo Vista Controlador (MVC), es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. Symfony está basado en un patrón clásico del diseño web conocido como arquitectura MVC. Este patrón divide una aplicación interactiva en 3 áreas: procesamiento, salida y entrada. Para esto, utiliza las siguientes abstracciones:

- El modelo representa la información con la que trabaja la aplicación, es decir, su lógica de negocio.
- La vista transforma el modelo en una página web que permite al usuario interactuar con ella.
- El controlador se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista.(16)

En la siguiente figura se muestra la implementación del patrón MVC en Symfony, un framework de desarrollo que se estará explicando más adelante.

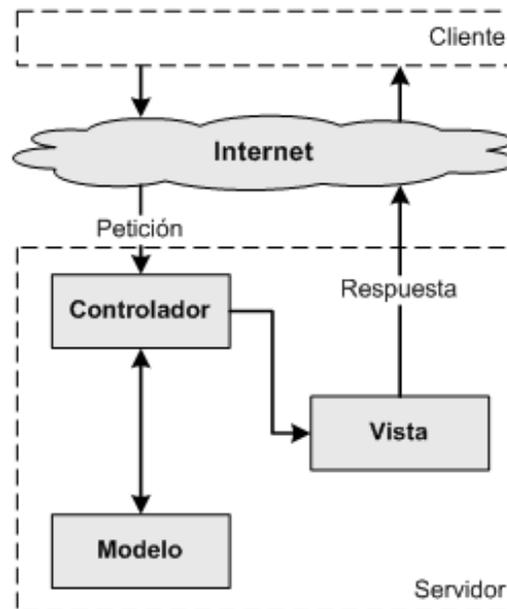


Figura 2: Implementación del patrón MVC por Symfony

Los patrones de diseño mencionados anteriormente son los más visualizados y utilizados dentro del desarrollo de la solución. Todos están evidenciados principalmente en los framework escogidos y en el diseño realizado para darle cumplimiento a los requerimientos.

1.5.3 Lenguajes de programación

Un lenguaje de programación no es más que un conjunto de sintaxis y reglas semánticas que definen los programas del computador. Es una técnica estándar de comunicación para entregarle instrucciones al computador. Un lenguaje le da la capacidad al programador de especificarle al computador, qué tipo de datos actúan y que acciones tomar bajo una variada gama de circunstancias, utilizando un lenguaje relativamente próximo al lenguaje humano.

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.(17)

Capítulo 1. Fundamentación Teórica

1.5.3.1 Lenguajes utilizados en el lado del cliente

Para realizar la programación en el lado del cliente³ fueron elegidos por el Proyecto de Soluciones Aduaneras los siguientes lenguajes:

XHTML:

El lenguaje HTML es un estándar reconocido en todo el mundo y cuyas normas define un organismo sin ánimo de lucro llamado W3C (World Wide Web Consortium). Como se trata de un estándar reconocido por todas las empresas relacionadas con el mundo de Internet, una misma página HTML se visualiza de la misma manera en cualquier navegador de cualquier sistema operativo.(18)

El propio W3C define el lenguaje HTML como “un lenguaje reconocido universalmente y que permite publicar información de forma global”. Desde su creación, el lenguaje HTML ha pasado de ser un lenguaje utilizado exclusivamente para crear documentos electrónicos a ser un lenguaje que se utiliza en muchas aplicaciones electrónicas.

El lenguaje XHTML es muy similar al lenguaje HTML. De hecho, XHTML no es más que una adaptación de HTML al lenguaje XML. Técnicamente, HTML es descendiente directo del lenguaje SGML⁴ (Standard Generalized Markup Language), mientras que XHTML lo es del XML (que a su vez, también es descendiente de SGML).(18)

³**Lenguajes en el lado del cliente:** Son los lenguajes que basan su procesamiento en el cliente web, es decir que se ejecutan en el navegador del usuario.

⁴**SGML:** Consiste en un sistema para la organización y etiquetado de documentos

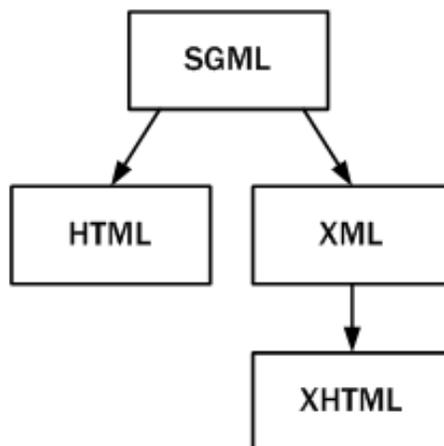


Figura 3: Esquema de la evolución de HTML y XHTML

En el sistema GINA es utilizada la versión 1.0 del estándar XHTML. Toda la interfaz de usuario desarrollada en el sistema tiene como base este lenguaje.

CSS:

CSS es un lenguaje de hojas de estilos creado para controlar la presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos y su presentación y es imprescindible para la creación de páginas web complejas.

La separación de los contenidos y su presentación presenta numerosas ventajas, ya que obliga a crear documentos HTML/XHTML bien definidos y con significado completo (también llamados “documentos semánticos”). Además, mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferentes.(19)

A través del CSS se logra una apariencia agradable a las vistas diseñadas e implementadas para el GINA. Describe cómo se va a mostrar un documento en la pantalla, o cómo se va a imprimir, o incluso cómo va a ser pronunciada la información presente en ese documento a través de un dispositivo de lectura. Ofrece a los desarrolladores el control total sobre estilo y formato de sus documentos.

Capítulo 1. Fundamentación Teórica

JavaScript:

Lenguaje de programación del lado del cliente inventado por Brendan Eich en la empresa Netscape Communications. Usado en las páginas web para agregar mayor funcionalidad, interacción o animaciones.(20)

Entre sus principales características podemos citar:

- Es interpretado por el cliente.
- Está basado en objetos.
- Su código se integra en las páginas XHTML, incluido en las propias páginas.
- Las referencias a objetos se comprueban en tiempo de ejecución, por lo tanto no se compila.
- Es independiente de la plataforma, por lo que se ejecuta en cualquier sistema operativo.(21)

En el sistema GINA se utiliza, principalmente, para manejar objetos dentro de las páginas web. Dichos objetos facilitan la programación de páginas interactivas, a la vez que se evita la posibilidad de ejecutar comandos que puedan ser peligrosos para la máquina del usuario, tales como formateo de unidades y modificación de archivos.

1.5.3.2 Lenguajes utilizados en el lado del servidor

El sistema GINA se ha implementado sobre la base del PHP como lenguaje de programación del lado del servidor⁵.

El **PHP** es un lenguaje de script incrustado dentro del HTML. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas de sí mismo. La meta del lenguaje es permitir rápidamente a los desarrolladores la generación dinámica de páginas.(22)

Cuenta con cuatro grandes características que hacen que este lenguaje sea distinguido entre los demás existentes en el mundo:

⁵**Lenguajes en el lado del servidor:** Son los lenguajes que se procesan en un servidor remoto y que generan la página web antes de enviarla al cliente.

Capítulo 1. Fundamentación Teórica

- **Velocidad:** No solo la velocidad de ejecución, la cual es importante, sino además no crear demoras en la máquina. Por esta razón no debe requerir demasiados recursos de sistema. PHP se integra muy bien junto a otro software, especialmente bajo ambientes Unix, generalmente es utilizado como módulo de Apache, lo que lo hace extremadamente veloz.
- **Estabilidad:** Ninguna aplicación es 100% libre de errores, pero PHP goza de la ayuda de un gran grupo de programadores, permitiendo que los fallos de funcionamiento se encuentren y se reparan rápidamente. PHP utiliza su propio sistema de administración de recursos y dispone de un sofisticado método de manejo de variables, conformando un sistema robusto y estable.
- **Seguridad:** PHP provee diferentes niveles de seguridad, estos pueden ser configurados de archivos de configuración que se encuentran en el servidor.
- **Simplicidad:** Es un lenguaje de programación simple de implementar por lo que usuarios con experiencia en C y C++ podrán utilizar PHP rápidamente.(23)

Otra característica que cabe citar además es la conectividad. PHP dispone de una amplia gama de librerías, y agregarle extensiones para extender su alcance es muy fácil. Esto le permite al PHP ser utilizado en muchas áreas diferentes, tales como encriptado, gráficos y XML.

Ventajas adicionales de PHP

- PHP se ejecuta en (casi) cualquier plataforma utilizando el mismo código fuente, pudiendo ser compilado y ejecutado en alrededor de 25 plataformas, incluyendo diferentes versiones de Unix, Windows y Macs. Como en todos los sistemas se utiliza el mismo código base, los scripts pueden ser ejecutados de manera independiente al sistema operativo.
- PHP es completamente expandible. Está compuesto de un sistema principal (escrito por Zend), un conjunto de módulos y una variedad de extensiones de código.
- Muchas interfaces distintas para cada tipo de servidor. PHP actualmente se puede ejecutar bajo Apache, IIS, AOLServer, Roxen y THTTPD. Otra alternativa es configurarlo como módulo CGI.
- Puede interactuar con muchos motores de bases de datos tales como MySQL, MS SQL, Oracle, Informix, PostgreSQL, y otros muchos.

Capítulo 1. Fundamentación Teórica

- PHP es Open Source, lo cual significa que el usuario no depende de una compañía específica para arreglar cosas que no funcionan, además de no estar forzado a pagar las actualizaciones del software.(23)

1.5.3.3 Otros Lenguajes

XML es un Lenguaje de Etiquetado Extensible muy simple, pero estricto que juega un papel fundamental en el intercambio de una gran variedad de datos, sirve para estructurar, almacenar e intercambiar información.(24)

Representa una manera distinta de hacer las cosas, más avanzada, cuya principal novedad consiste en permitir compartir los datos con los que se trabaja a todos los niveles, por todas las aplicaciones y soportes. Así pues, el XML juega un papel importantísimo en este mundo actual, que tiende a la globalización y la compatibilidad entre los sistemas, ya que es la tecnología que permitirá compartir la información de una manera segura, fiable, fácil.(25)

Un documento XML puede contener muchos tipos de información. En el proyecto GINA se utiliza para la definición del schema.xml, donde fundió entidades y atributos de la base de datos. También se maneja información entrada por ficheros de las DM destinadas al Despacho Comercial.

1.5.4 Frameworks

Se han realizado importantes progresos orientados a la reusabilidad del software a través de la aplicación del paradigma de la programación orientada a objetos y mediante el uso de los componentes tecnológicos. Sin embargo, estas tecnologías solo proveen el re-uso en el nivel individual, frecuentemente a menor escala. Con la aparición de los patrones de diseño se ha demostrado la reutilización de soluciones para resolver problemas de escala mayor enfocados en la solución de problemas sencillos. Sin embargo, el más complejo problema de la reutilización de soluciones es en el nivel de los grandes componentes, para que estos se puedan adaptar para las solicitudes individuales.(26)

Un framework, en el desarrollo de software es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas,

Capítulo 1. Fundamentación Teórica

bibliotecas y un lenguaje interpretado entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto.(27)

Un framework simplifica el desarrollo de una aplicación mediante la automatización de algunos de los patrones utilizados para resolver las tareas comunes. Además, un framework proporciona estructura al código fuente, forzando al desarrollador a crear código más legible y más fácil de mantener. Facilita la programación de aplicaciones, ya que encapsula operaciones complejas en instrucciones sencillas.(16)

Realmente no existe una definición oficial de Framework, pero todos los autores coinciden en la utilización de un tema común: la reutilización. Una definición dada por R. E. Johnson, and B. Foote en 1988 en su publicación "Designing Reusable Classes: "Un Framework es un conjunto de clases que personifican un diseño abstracto para soluciones de una familia de problemas relacionados..."(28)

1.5.4.1 Framework Ext JS

Ext JS es una librería JavaScript ligera y de alto rendimiento, compatible con la mayoría de navegadores que permite crear páginas e interfaces web dinámicas(29). Permite realizar aplicaciones Web enriquecidas basándose en tecnología AJAX, JSON, DHTML y DOM. Ext JS 3.0, está patentado bajo licencia LGPL, lo que posibilita su uso para aplicaciones empresariales privativas de código cerrado.

Con Ext JS, se pueden desarrollar aplicaciones web multiplataforma con facilidad. El modelo de componente de Ext JS mantiene su código bien estructurado por lo que incluso las aplicaciones más grandes pueden ser de fácil mantenimiento.

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

La utilización de un Framework de JavaScript como Ext JS facilita la separación de las capas de la vista con la del controlador desde el punto de vista productivo ya que el código utilizado en la primera es solamente JavaScript y no es necesario utilizar ningún tipo de código PHP, así los desarrolladores pueden

Capítulo 1. Fundamentación Teórica

centrarse más en el aprendizaje de un solo lenguaje. Además al soportar serialización de objetos mediante tecnología JSON permite que los datos enviados desde el controlador como respuesta a la vista contengan solo las propiedades de dichos objetos, pero no el comportamiento, minimizando los posibles errores de programación y los accidentes de que los objetos sean modificados erróneamente desde la vista.(29)

El framework Ext JS es utilizado en el GINA para la realización de las pantallas de interfaz de usuario. Permite la comunicación con otras capas del proyecto de manera sencilla. Brinda facilidades y dinamismos en las vistas.

1.5.4.2 Framework Symfony

Symfony es un completo framework diseñado para optimizar, gracias a sus características, el desarrollo de las aplicaciones web. Para empezar, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. El resultado de todas estas ventajas es que no se debe reinventar la rueda cada vez que se crea una nueva aplicación web.

Symfony está desarrollado completamente con PHP 5. Ha sido probado en numerosos proyectos reales y se utiliza en sitios web de comercio electrónico de primer nivel. Symfony es compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft. Se puede ejecutar tanto en plataformas *nix (Unix, Linux) como en plataformas Windows.(16)

Symfony se diseñó para que se ajustara a los siguientes requisitos:

- Fácil de instalar y configurar en la mayoría de plataformas (y con la garantía de que funciona correctamente en los sistemas Windows y *nix estándares)
- Independiente del sistema gestor de bases de datos
- Sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos

Capítulo 1. Fundamentación Teórica

- Basado en la premisa de "convenir en vez de configurar", en la que el desarrollador solo debe configurar aquello que no es convencional
- Sigue la mayoría de mejores prácticas y patrones de diseño para la web
- Preparado para aplicaciones empresariales y adaptable a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo
- Código fácil de leer que incluye comentarios de phpDocumentor y que permite un mantenimiento muy sencillo
- Fácil de extender, lo que permite su integración con librerías desarrolladas por terceros (16)

Symfony en su versión 1.2.8, es el framework utilizado para la creación del GINA. Brinda facilidades para el desarrollo con el lenguaje PHP. Utiliza metodologías y patrones de diseño que brindan la posibilidad de reutilizar el código. Presenta buena seguridad, es multiplataforma y sencillo de configurar. Todo el soporte del sistema GINA está sobre la base de Symfony. Incluye la premisa de no inventar la rueda y es por eso que utiliza lo mejor de diferentes framework para proporcionar un trabajo limpio sobre los proyectos a realizar. Permite por medio de Propel el mapeo de las base de datos, transformando las tablas en clases del negocio. Crea formularios seguros para la entrada de información, soporta internacionalización, incluye lo mejor de los frameworks más usados en el mundo.



Figura 4. Frameworks utilizados por Symfony

Capítulo 1. Fundamentación Teórica

1.5.5 Entornos de Desarrollo Integrado (IDE)

Un Entorno de Desarrollo Integrado (en inglés Integrated Development Environment o IDE) es un programa compuesto por una serie de herramientas que utilizan los programadores para desarrollar código. Esta herramienta puede estar pensada para su utilización con un único lenguaje de programación o bien puede dar cabida a varios de estos.(30)

Las herramientas que normalmente componen un entorno de desarrollo integrado son las siguientes: un editor de texto, un compilador, un intérprete, herramientas para la automatización, un depurador, un sistema de ayuda para la construcción de interfaces gráficas de usuario y, opcionalmente, un sistema de control de versiones.

NetBeans IDE

NetBeans IDE es una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito en Java - pero puede servir para cualquier otro lenguaje de programación. Existe además un número importante de módulos para extender el NetBeans IDE, además es un producto libre y gratuito sin restricciones de uso.(31) Posee un amplio soporte para el lenguaje PHP así como para los framework de trabajo Symfony y Ext JS.

Para el desarrollo de la solución se ha elegido NetBeans IDE, en su versión 6.9, como Entorno de Desarrollo Integrado.

1.5.6 Herramientas de Diseño

Hoy en día, muchas empresas se han extendido a la adquisición de herramientas CASE⁶, con el fin de automatizar los aspectos clave de todo el proceso de desarrollo de un sistema, desde el principio, hasta el final y así incrementar su posición en el mercado competitivo.

Algunas de estas herramientas tienen un valor económico muy alto y requieren costos elevados de entrenamiento del personal, además de las dificultades de adaptación de la herramienta a las que se

⁶**CASE**: Siglas en inglés que se utilizan para referirse a Ingeniería de Software Asistida por Computadora

Capítulo 1. Fundamentación Teórica

enfrentan los usuarios que la utilizarán, así como a la arquitectura de la información en la que está compuesta y a las metodologías de desarrollo utilizadas por la organización.

Visual Paradigm for UML (VP-UML) es un modelador UML que permite el diseño de sistemas con todo tipo de tipos de diagramas UML. También es utilizado para diseñar diagramas de casos de uso, diagramas de requerimiento y diseño de bases de datos relacionales. Con VP-UML, el equipo de desarrollo de software puede realizar el análisis y diseño de sistemas con eficacia.(32)

Es una herramienta multiplataforma y se integra con algunas herramientas realizadas en Java entre las cuales se puede citar el NetBeans IDE.

Debido a las características y facilidades el Visual Paradigm for UML en su versión 3.4 es el utilizado para la realización del diseño de la solución.

1.6 Gestor de Base de Datos

Un Sistema Gestor de Bases de Datos (SGBD) o DBMS (Data Base Management System) es una colección de programas cuyo objetivo es servir de interfaz entre la base de datos, el usuario y las aplicaciones. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. Un SGBD permite definir los datos a distintos niveles de abstracción y manipular dichos datos, garantizando la seguridad e integridad de los mismos.(33)

En las soluciones implementadas para la AGR el gestor de base de datos que se utiliza es Oracle 11g: sistema de gestión de base de datos relacional desarrollado por Oracle Corporation.

Con Oracle 11g se reducen los costes informáticos y es posible ofrecer una calidad de servicio superior:

1. Consolidando las aplicaciones de negocio en redes de bases de datos rápidas, fiables y ampliables.
2. Maximizando la disponibilidad y eliminando la redundancia del centro de datos inactivo.
3. Comprimiendo datos en particiones de almacenamiento de bajo coste para un rendimiento más rápido.

Capítulo 1. Fundamentación Teórica

4. Protegiendo con seguridad la información y permitiendo el cumplimiento.
5. Duplicando la productividad de la DBA y reduciendo el riesgo de cambios.(34)

Principales ventajas de Oracle

- Las entidades complejas del mundo real y la lógica se pueden modelar fácilmente, lo que permite reutilizar objetos para el desarrollo de base de datos de una forma más rápida y con mayor eficiencia.
- Los programadores de aplicaciones pueden acceder directamente a tipos de objetos Oracle, sin necesidad de ninguna capa adicional entre la base de datos y la capa cliente.
- Las aplicaciones que utilizan objetos de Oracle son fáciles de entender y mantener porque soportan las características del paradigma orientado a objetos.
- Tiene buen rendimiento y hace buen uso de los recursos.
- Posee un rico diccionario de datos.
- Brinda soporte a la mayoría de los lenguajes de programación.
- Es un sistema multiplataforma, disponible en Windows, Linux y Unix.
- Permite tener copias de la base de datos productiva en lugares lejanos a la ubicación principal. Las copias de la Base de Datos productiva pueden estar en modo de lectura solamente. (34)

Desventajas

- Es un producto de elevado precio por lo que por lo general se utiliza en empresas muy grandes y multinacionales
- Los costos de soporte técnico y mantenimiento son elevados
- Vulnerabilidades en la seguridad de la plataforma, se hace necesario aplicar parches de seguridad.(34)

Es la única herramienta privativa que se ha ganado un lugar en la AGR debido a sus insustituibles características. Ha estado presente desde 1997 en la aduana siempre brindando buenos resultados.

1.7 Conclusiones parciales

En el capítulo se explicaron todas las herramientas que están definidas para la realización de los objetivos, herramientas que se caracterizan por las facilidades, ventajas y optimización que le brindan a los desarrolladores, así como las mejores prácticas que existen mundialmente sobre patrones y metodologías para la creación de páginas web. También se evidenció la necesidad de realizar un módulo para la AGR que realice todas las formalidades que se tienen en cuenta con los regímenes de Tránsito y Transferencia, teniendo en cuenta que no existe sistema que sea completamente compatible con esas necesidades.

Capítulo 2. Diseño e Implementación

2.1 Introducción

En el presente capítulo se plasmarán los flujos de diseño e implementación. Se obtendrán una serie de artefactos⁷ de diseño que contienen la solución detallada sobre las clases a utilizar, la comunicación del módulo con otros subsistemas, el flujo que debe seguir las diferentes peticiones del usuario, la composición que presenta la base de datos y los pasos a seguir para poder realizar la elaboración de los diferentes requerimientos. El objetivo de todos estos resultados es definir la vía correcta y el orden en que se van a realizar las actividades facilitando la comprensión y rapidez a los implementadores logrando un mejor desarrollo de la solución.

2.2 Modelo del Diseño

El Diseño de Sistemas se define como el proceso de aplicar ciertas técnicas y principios con el propósito de definir un dispositivo, un proceso o un Sistema, con suficientes detalles como para permitir su interpretación y realización física.

La importancia del Diseño del Software se puede definir en una sola palabra **Calidad**, dentro del diseño es donde se fomenta la calidad del Proyecto. El Diseño es la única manera de materializar con precisión los requerimientos del cliente.

El Diseño del Software es un proceso y un modelado a la vez. El proceso de Diseño es un conjunto de pasos repetitivos que permiten al diseñador describir todos los aspectos del Sistema a construir. A lo largo del diseño se evalúa la calidad del desarrollo del proyecto con un conjunto de revisiones técnicas:

- El diseño debe implementar todos los requisitos explícitos contenidos en el modelo de análisis y debe acumular todos los requisitos implícitos que desea el cliente.

⁷ **artefactos**: un artefacto puede ser un modelo, o un elemento de modelo, un documento, en fin todo lo que puede ser generado en el proceso.

Capítulo 2. Diseño e Implementación

- Debe ser una guía que puedan leer y entender los que construyan el código y los que prueban y mantienen el Software.
- El Diseño debe proporcionar una completa idea de lo que es el Software, enfocando los dominios de datos, funcional y comportamiento desde el punto de vista de la Implementación. (35)

2.2.1 Requerimientos

La mayoría de proyectos de software son complejos, y la estrategia primaria para superar la complejidad, es la descomposición (divide y vencerás). La estrategia es dividir el problema en unidades más pequeñas que sean manejables. Un enfoque de este tipo de acción lo representa el análisis y diseño, donde se trata de descomponer el problema en funciones o procesos. Cada uno de estos procesos se deriva en diferentes requisitos funcionales que no es más que la descripción de los deseos del cliente y luego cada uno de los requerimientos es desglosado aún más para definir bien como debe ser el flujo de ejecución.

El módulo Tránsito y Transferencia perteneciente al Despacho Comercial presenta siete requerimientos funcionales.

- Despachar Transferencia
- Despachar Tránsito
- Modificar Transferencia
- Modificar Tránsito
- Anular Transferencia
- Anular Tránsito
- Pagar Facilidad Tránsito Anticipado

2.2.2 Diagrama de Paquetes

Los diagramas de paquetes se usan para reflejar la organización de paquetes y sus elementos. Es un mecanismo de agrupamiento que es altamente cohesivo internamente y con las interacciones mínimas con otros paquetes.

Capítulo 2. Diseño e Implementación

El diagrama de paquetes que se representa en la **figura 5**, contiene la interacción del módulo Tránsito y Transferencia con los diferentes subsistemas que presenta el GINA, así como la interacción con los paquetes y librerías de sí mismo.

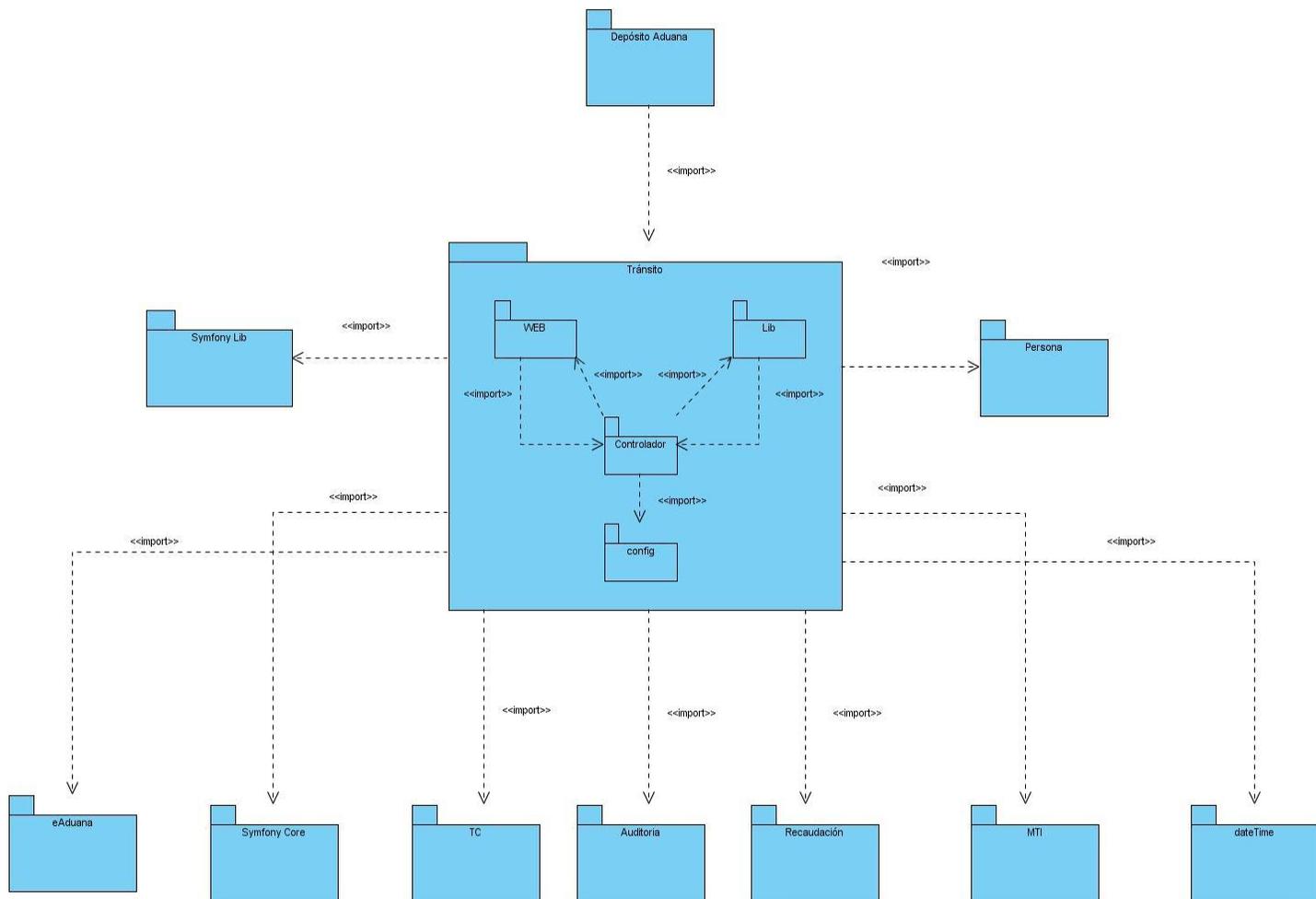


Figura 5 Diagrama de Paquetes

El diagrama de paquetes se encuentra en el documento Modelo de Diseño.(36)

Capítulo 2. Diseño e Implementación

Dentro de los subsistemas que se relacionan con el módulo Tránsito y Transferencia se encuentra TC que es el encargado de administrar las clases que son nomencladoras y comunes para todos los subsistemas del GINA. También se halla MTI brindando servicios especializados sobre confirmación de información referente al Cotejo de Manifiesto. El subsistema Persona brinda todo tipo de información referente al personal de la AGR y el historial de incidencias que presenten. Se utilizan los subsistemas Auditoria y Recaudación para guardar todas las acciones realizadas sobre el módulo y referentes al pago de las operaciones respectivamente, eAduana para el trabajo con los archivos y DateTime para las soluciones de edición de fechas, así como la interacción con el núcleo de Symfony y la librería. Dentro del mismo módulo se materializa el vínculo con diferentes paquetes, pertenecientes a las diferentes capas que define el patrón arquitectónico MVC. Dentro de los paquetes de mayor importancia se encuentra, el WEB encargado de la capa de presentación o Vista, en el se encuentran todas las interfaces de usuarios desarrolladas, el CONTROLADOR que pertenece a la capa Controladora, presenta las clases y objetos para el control de las peticiones y el flujo de acciones a realizar y el LIB que es la capa del Modelo, donde se encuentran las clases del negocio, los formularios y algunas de las encargadas del mapeo y abstracción de la base de datos. También el paquete CONFIG que es el que contiene todos los archivos de configuración del módulo.

2.2.3 Diagrama de Clases del Sistema

Un diagrama de clases sirve para visualizar las relaciones entre las clases que involucran el sistema. Las clases están compuestas por los atributos pasivos y los activos, mientras que las relaciones pueden ser de diferentes tipos como herencia, Composición, Agregación, Asociación y Uso.

El diagrama de clases que se representa a continuación (**Figura 6**) es el usado en el módulo Tránsito y Transferencia del Despacho Comercial y forma parte de la capa de Modelo. Presenta un total de 27 clases. De ellas 17 son nomencladores de las cuales 15 pertenecen al subsistema TC, donde se obtiene información a través de servicios que se brindan. De las clases más significativas se encuentran la DcDm que es contenedora de todas las funcionalidades y atributos comunes de las DcDmTransferencia y DcDmTránsito y estas a su vez contienen sus especificaciones, y en su conjunto, las tres, contienen toda la implementación principal que se necesita para la realización de los regímenes de Tránsito y Transferencia, la relación que existe entre estas clases y debido a la utilización del Object-Relational

2.2.4 Diagrama de Interacción del Diseño

El diagrama de interacción, representa la forma en cómo un Cliente (Actor) u Objetos (Clases) se comunican entre sí, en dependencia del evento ejecutado. Esto implica recorrer toda la secuencia de llamadas, de donde se obtienen las responsabilidades claramente.(37)

Hay dos tipos de diagrama de interacción, ambos basados en la misma información, pero cada uno enfatizando un aspecto particular: Diagramas de Secuencia y Diagramas de Colaboración. (38)

2.2.4.1 Diagrama de Secuencia

Un diagrama de Secuencia muestra una interacción ordenada según la secuencia temporal de eventos. En particular, muestra los objetos participantes en la interacción y los mensajes que intercambian ordenados según su secuencia en el tiempo. El eje vertical representa el tiempo, y en el eje horizontal se colocan los objetos y actores participantes en la interacción, sin un orden prefijado. Cada objeto o actor tiene una línea vertical, y los mensajes se representan mediante flechas entre los distintos objetos. El tiempo fluye de arriba abajo. Se pueden colocar etiquetas (como restricciones de tiempo, descripciones de acciones) bien en el margen izquierdo o bien junto a las transiciones o activaciones a las que se refieren.

2.2.4.2 Diagrama de Secuencia Orientado a Actividades del Negocio

En el CEIGE se realiza un diagrama de secuencia orientado a actividades. El objetivo principal de este diagrama es definir la interacción entre las clases y los usuarios así como el flujo a seguir de las actividades a realizar, experimentando un nivel de especificación tal, que permita a los programadores lograr la implementación de la solución en menor tiempo y con mejor calidad. Pueden existir uno o varios diagramas asociados a cada requisito funcional. Se representan por calles, actividades y la relación entre estas actividades incluyendo también comentarios para mayor comprensión.

En la **figura 7** se presenta un ejemplo del diagrama de secuencia orientado a actividades del requisito funcional Despachar Tránsito. Presenta un total de 6 calles que representan los objetos participantes, incluye también actividades que identifican llamadas a funcionalidades y pasos a seguir para la implementación, así como bifurcaciones y ciclos. El objetivo de este diagrama es lograr realizar todas las

Capítulo 2. Diseño e Implementación

validaciones exigidas para realizar el despacho de los Tránsitos y luego el guardado de los datos en la base de datos.



Figura 7. Diagrama de Secuencia Despachar Tránsito

Los otros diagramas de secuencia que presenta el sistema, se encuentran definidos en el documento de diseño del módulo Tránsito y Transferencia.(36)

Capítulo 2. Diseño e Implementación

En la **Figura 8** se muestra de manera más específica, un flujo de actividades a seguir y la relación que existe entre estas, incluyendo algunas llamadas a funcionalidades, lanzamiento de errores y condicionales.

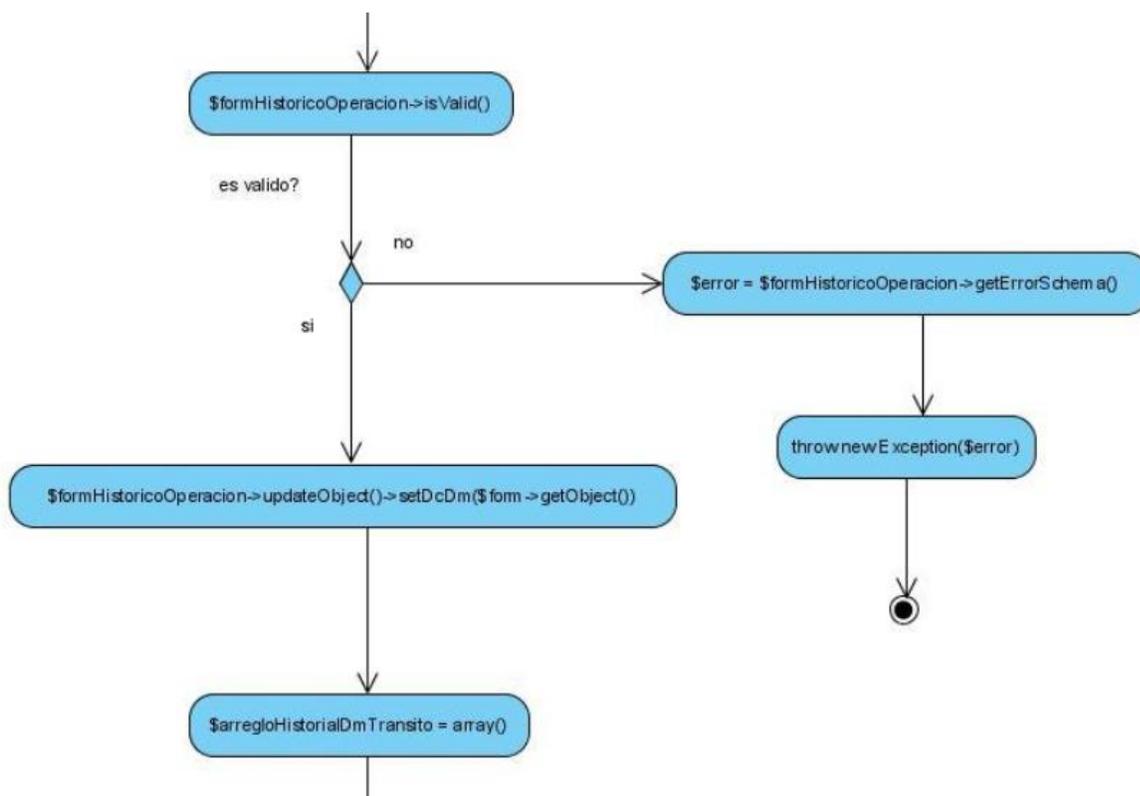


Figura 8. Flujo de actividades

Capítulo 2. Diseño e Implementación

2.2.4.3 Diagrama de Secuencia Orientado a Actividades de Componentes Visuales

Este diagrama presenta las mismas políticas del orientado a negocio. Está dirigido principalmente a los programadores de las interfaces de usuario para que tengan una base por donde guiarse a la hora de brindar las funcionalidades y la forma de comportamiento de las pantallas. Se evidencian en ellos una serie de comentarios y acciones que incluyen nombre de las funcionalidades a llamar, así como de los valores a enviar y mensajes que mostrar. En la **figura 9** se presenta un ejemplo de la interfaz Despachar Tránsito.

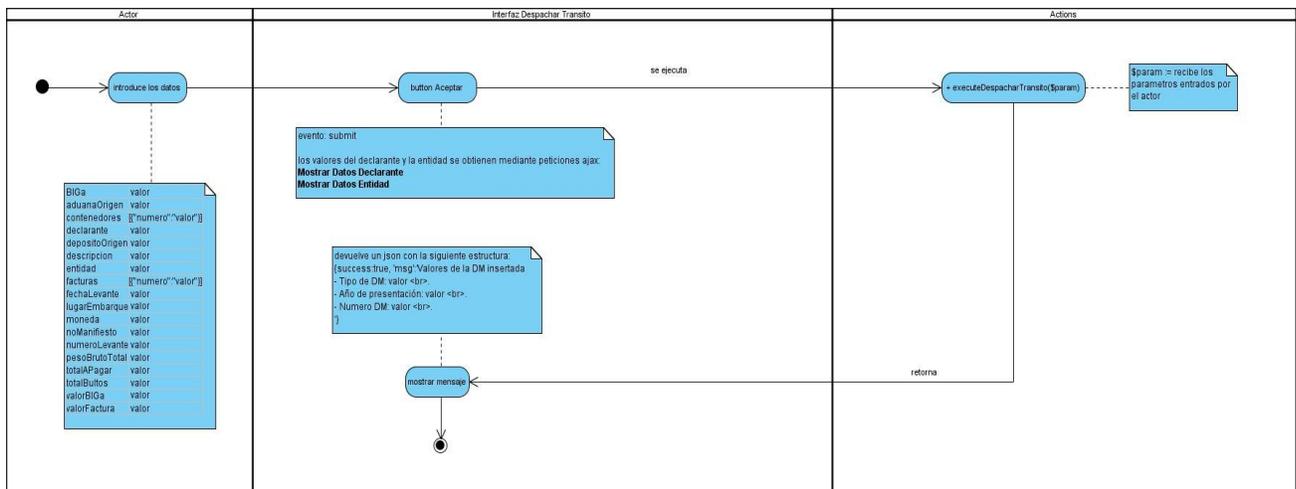


Figura 9. Diagrama de Secuencia de Componentes Visuales Despachar Tránsito

2.2.5 Diseño de la Base de Datos

El diseño de la base de datos es el hito más importante que se realiza en el desarrollo de un sistema que necesite trabajar con información persistente. En general, el objetivo del diseño de una base de datos relacional es generar un conjunto de esquemas de relaciones que permitan almacenar la información con un mínimo de redundancia, pero que a la vez faciliten la recuperación de la información.

Capítulo 2. Diseño e Implementación

2.2.5.1 Modelo Entidad-Relación

El modelo entidad-relación **figura 10** generado para el sistema contiene las tablas (y las relaciones entre ellas) que guardarán la información de las formalidades de los regímenes de Tránsito y Transferencia.

Las tablas de color verde son las más importantes arquitectónicamente debido a que en ellas se almacenan los datos principales de los Tránsitos y las Transferencias.

Las tablas de color gris son nomencladoras, contribuyen al completamiento de la información de las tablas dichas anteriormente. De ellas las grises claras son pertenecientes al esquema TC de la base de datos, permitiendo el acceso de todos los subsistemas que conforman el GINA; las grises oscuras pertenecen a el esquema de TRANSITO.

Las tablas azules son de vital importancia debido a que en ellas se lleva un control del historial de cada uno de los regímenes que se realizan.

Todas en su conjunto conforman un modelo de entidad-relación, normalizado en Tercera Forma Normal (3FN), según la propuesta original de Codd (1972), que plantea que una relación está en tercera forma normal si todos los atributos de la relación dependen funcionalmente sólo de la clave, y no de ningún otro atributo.

La *normalización de los datos* puede considerarse como un proceso durante el cual los esquemas de relación que no cumplen las condiciones se descomponen repartiendo sus atributos entre esquemas de relación más pequeños que cumplen las condiciones establecidas. Un objetivo del proceso de normalización realizado para el modelo antes presentado, es garantizar que no ocurran anomalías de actualización.

Capítulo 2. Diseño e Implementación

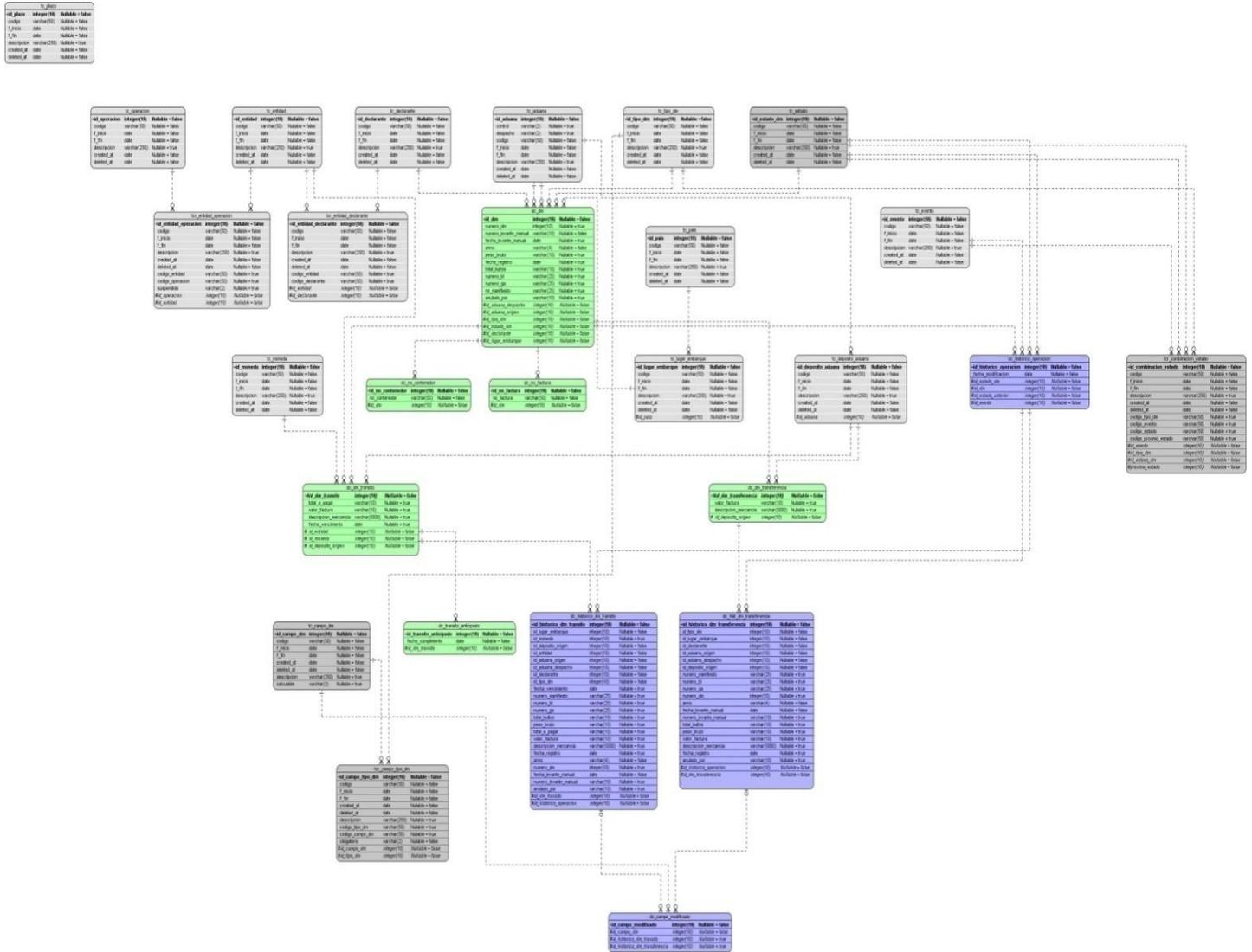


Figura 10. Modelo Entidad-Relación del módulo Tránsito y Transferencia

La descripción de las tablas de la Base de Datos, empleadas en el desarrollo del sistema se encuentra reflejada en el documento de Descripción de las Tablas.(39)

2.2.6 Métricas para la validación del diseño

La *medición* “es el proceso por el cual los números o símbolos son asignados a atributos o entidades en el mundo real tal como son descritos de acuerdo a reglas claramente definidas” (40). Una *medida* “proporciona una indicación cuantitativa de extensión, cantidad, dimensiones, capacidad y tamaño de algunos atributos de un proceso o producto” (41).

El IEEE “*Standard Glossary of Software Engineering Terms*” define *métrica* como “una medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo dado” (42).

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

- Tamaño operacional de clase (TOC): métrica que se basa esencialmente en la cantidad de funcionalidades que presenta la clase.
- Relaciones entre Clases (RC): métrica que se basa en la cantidad de relaciones que presenta cada clase con el resto. Se establece relación entre dos clases cuando se tiene instancias, atributos y llamadas a funcionalidades de una clase en otra.
- Profundidad de herencia (PH): mide el máximo nivel en la jerarquía de herencia. Es la cuenta directa de los niveles en la jerarquía de herencia.
- Número de descendientes (ND): es el número de subclases subordinadas a una clase en la jerarquía, es decir, el número de subclases que pertenecen a una clase.
- Número de operaciones redefinidas para una clase hija (NOR): muestra en qué medida las subclases redefinen el comportamiento de sus superclases.

Para evaluar el diseño realizado se utilizaron las métricas enfocadas a la calidad, definidas por el CEIGE, empleándose la métrica Tamaño operacional de clase (TOC), la cual mide la calidad de acuerdo a los atributos Responsabilidad, Complejidad de implementación y Reutilización, teniendo en cuenta cálculos sobre la base de las operaciones para cada clase individual. También se realizó la métrica de Relaciones entre Clases (RC), para definir el nivel de dependencia existente. Con la utilización de las dos métricas dichas anteriormente, se realiza una valoración, sobre el nivel que presentan las propiedades de

Capítulo 2. Diseño e Implementación

acoplamiento y cohesión. Debido que el modelo de clases del sistema no presenta herencias, no se desarrollaron las otras métricas definidas.

Atributos de calidad que se abarcan en la métrica TOC:

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

Atributos de calidad que se abarcan en la métrica RC:

- **Acoplamiento.** Consiste en el nivel de dependencia existente entre las clases.
- **Complejidad de Mantenimiento.** Consiste en el grado de dificultad que se presenta a la hora de realizar el mantenimiento del sistema.

La asignación de categorías se lleva a cabo según los siguientes umbrales. Donde la variable **Prom** se refiere al promedio calculado entre las clases, teniendo en cuenta la cantidad de operaciones y de relaciones que se presentan en cada una de ellas.

Para el cálculo de la Responsabilidad:

Responsabilidad	Categoría	Criterio
	Baja	$\leq \text{Prom.}$
	Media	Entre Prom. y $2 * \text{Prom.}$
	Alta	$> 2 * \text{Prom.}$

Para el cálculo de la Complejidad de Implementación:

Complejidad implementación	Categoría	Criterio
	Baja	$\leq \text{Prom.}$
	Media	Entre Prom. y $2 * \text{Prom.}$
	Alta	$> 2 * \text{Prom.}$

Capítulo 2. Diseño e Implementación

Para el cálculo de la Reutilización:

Reutilización	Categoría	Criterio
	Baja	$> 2 * \text{Prom.}$
	Media	Entre Prom. y $2 * \text{Prom.}$
	Alta	$\leq \text{Prom.}$

Para el cálculo del Acoplamiento:

Acoplamiento	Categoría	Criterio
	Ninguno	0
	Baja	1
	Media	2
	Alta	> 2

Para el cálculo de la Complejidad de Mantenimiento:

Reutilización	Categoría	Criterio
	Baja	$\leq \text{Prom.}$
	Media	Entre Prom. y $2 * \text{Prom.}$
	Alta	$> 2 * \text{Prom.}$

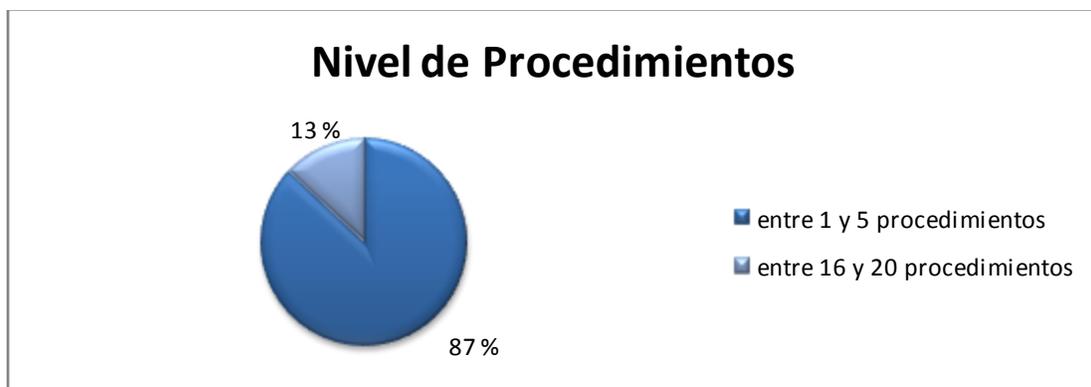


Figura 11. Representación del Nivel de Procedimientos por Clases



Figura 12. Representación de la Responsabilidad por Clases



Figura 13. Representación de Complejidad por Clases



Figura 14. Representación de la Reutilización por Clases



Figura 15. Representación de las Dependencias por Clases



Figura 16. Nivel de Acoplamiento por Clases

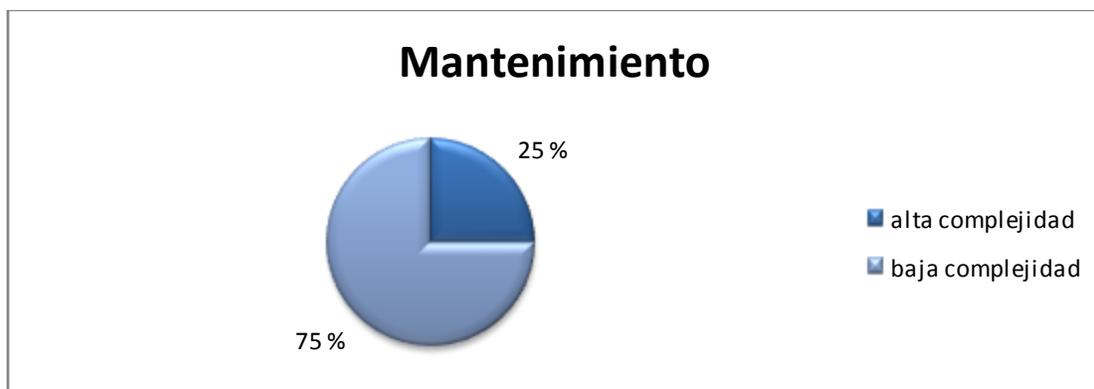


Figura 17. Nivel de la Complejidad de Mantenimiento por Clases

2.2.6.1 Resultado de la evaluación de las métricas.

Las clases de la solución, presentan un nivel bajo de funcionalidades, al igual que de dependencias entre ellas, contribuyendo al bajo acoplamiento. La máxima responsabilidad se deroga en unas pocas clases del diseño, sin dejar de contar que las demás no dejan de ser de gran importancia para el sistema. La gran

Capítulo 2. Diseño e Implementación

mayoría de las clases son poco complejas de implementar y de realizar posterior mantenimiento, debido a que la mayor cantidad de algoritmos se concentra en las más significativas arquitectónicamente. Se cuenta con un alto por ciento de reutilización.

2.3 Modelo de Implementación

La implementación es el proceso que comienza a partir que se complete el diseño y que tiene como objetivo principal el desarrollo de la arquitectura y el sistema como un todo. En el modelo de implementación se recogen mediante diferentes artefactos, todas las acciones que se llevaran a cabo en este proceso. Se realiza en una sucesión de pasos pequeños y manejables de manera incremental distribuyendo por todo el sistema los diferentes componentes y organizándolos de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de implementación y en el lenguaje o lenguajes de programación utilizados.

2.3.1 Estándar de Codificación

Las convenciones o estándares de codificación son pautas de programación que no están enfocadas a la lógica del programa, sino a su estructura y apariencia física para facilitar la lectura, comprensión y mantenimiento del código.(43)

El objetivo de crear un estándar de codificación es que los programadores se rijan por el mismo a la hora de implementar. Este estándar debe servir para el personal del proyecto para identificar de forma sencilla cual es el objetivo y las funcionalidades que brinda cada una de las clases, funciones y demás componentes de Software dada su nomenclatura, es necesario que esto se pueda identificar a simple vista. Además debe servir de guía para los implementadores que tienen que continuar el desarrollo de las aplicaciones luego.(44)

La implementación llevada a cabo para el desarrollo del módulo está regida por el documento “Propuesta de un estándar de codificación”. Comprende todas las aplicaciones desarrolladas bajo el Departamento de Soluciones para la Aduana del Centro de Informatización y Gestión de Entidades y todo el código

Capítulo 2. Diseño e Implementación

generado bajo la tecnología y el lenguaje PHP y que utilicen la arquitectura regida por la utilización del Framework Arquitectónico Symfony.(44)

Como la implementación de las aplicaciones está basada en la utilización de un Framework que en su totalidad está codificado en inglés, es imposible desligar la implementación completamente de este idioma. Por lo tanto la implementación se realizará teniendo en cuenta los idiomas de inglés, español, incluso la combinación de ambos al mismo tiempo, sin olvidar la utilización de los sufijos y prefijos “Set”, “Get”, “Is” y “Max”.(44)

Teniendo en cuenta los componentes de Symfony, las aplicaciones deben tener nombres que dejen reflejado bien claramente cuál es el propósito de la misma, ya en una palabra o siglas. Todos los nombres de acciones deben estar en la nomenclatura “CamelCase⁸” comenzando por la palabra **execute**. Los nombres de las clases deben estar expresados en notación UpperCamelCase⁹ y deben expresar con claridad cuál es el alcance y la responsabilidad de la clase. Se mantiene el uso de los sufijos “**Peer**” para las clases que se encargan de estas funciones en el modelo.(44)

Los nombres de las variables deben expresar claramente el contenido de la misma y pueden estar referidas en singular o plural. En caso de que no se le asigne un valor inicial a las variables se deben inicializar con un valor que indique el tipo de dato más general al que debe pertenecer. Los arreglos comenzaran con la palabra **arreglo** y luego el resto en CamelCase.(44)

Teniendo en cuenta las políticas de las llaves se utilizará la tradicional de Unix de colocar la primera llave en la misma línea del comienzo del bloque separada por un espacio y la llave de cierre del bloque en la última línea.(44)

Cuando se declara un arreglo inicializado con valores que contienen diversos índices y se extiende más allá del largo de una línea de código se abre el paréntesis y se continúa en la próxima línea utilizando una sangría.(44)

⁸**CamelCase**: Consiste en escribir frases o palabras compuestas eliminando los espacios intermedios y poniendo en minúscula la primera letra y en mayúscula las demás primeras letras de cada palabra contigua.

⁹**UpperCamelCase**: Consiste en escribir frases o palabras compuestas eliminando los espacios intermedios y poniendo en mayúscula la primera letra de cada palabra incluyendo la primera letra de la frase.

Capítulo 2. Diseño e Implementación

Para comentar una línea se utilizará el o los caracteres que dicta el lenguaje de programación utilizado, al igual que para los comentarios de bloque, es decir, que requiera más de un línea. (44)

2.3.2 Tratamiento de Errores

El tratamiento de errores es un aspecto importante para toda aplicación. Muchos son los errores cometidos por los usuarios, unos por accidentes y otros no tan accidentales. En el módulo Tránsito y Transferencia se controlan información importante que influye en la toma de valiosas decisiones para el Despacho Comercial realizado por la AGR, es por eso que se realizan varios tipos de validaciones de errores.

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

Las demás validaciones también juegan un papel importante y son las realizadas en las vistas por medio de expresiones regulares y en el negocio por medio de los formularios. Todas tienen como función principal evitar que introduzcan tipos de datos incorrectos en los diferentes campos, logrando así evitar ataques contra el sistema por esta vía, incluyendo las inyecciones SQL.

2.3.3 Comunicación entre Capas

El marco de trabajo que se utiliza es el Symfony y este utiliza el patrón arquitectónico MVC que separa el sistema en tres capas diferentes, por lo que se hace necesario realizar la comunicación entre las mismas.

La capa de la Vista es la encargada de recoger la información deseada y enviarla a través de sus formularios hacia la capa del controlador, por medio en algunos casos de peticiones AJAX para lograr mayor rapidez y a través de objetos JSON y en otras ocasiones por XML logrando el intercambio de información y objetos completos entre las capas de presentación y la del controlador.

Capítulo 2. Diseño e Implementación

La capa del controlador, que contiene el código que liga la lógica de negocio con la presentación, está dividida en varios componentes que se utilizan para diversos propósitos y se encargan de obtener los datos y según la petición, realizar la acción que le corresponde, la cual devolverá una respuesta determinada:

- El controlador frontal es el único punto de entrada a la aplicación. Carga la configuración y determina la acción a ejecutarse.
- Las acciones contienen la lógica de la aplicación. Verifican la integridad de las peticiones y preparan los datos requeridos por la capa de presentación.
- Los objetos request, response y session dan acceso a los parámetros de la petición, las cabeceras de las respuestas y a los datos persistentes del usuario. Se utilizan muy a menudo en la capa del controlador.
- Los filtros son trozos de código ejecutados para cada petición, antes o después de una acción. Por ejemplo, los filtros de seguridad y validación son comúnmente utilizados en aplicaciones web. Puedes extender el framework creando tus propios filtros.

Symfony utiliza Propel que es un ORM para PHP, transformando las tablas de la base de datos en objetos y realizando el tratamiento de la información mediante ellos, con los que se puede recuperar, insertar y modificar datos, proporcionando la posibilidad que no sea necesario preocuparse por las conexiones, ni de escribir SQL.

2.3.3.1 Ejemplo de Comunicación entre Capas

En este ejemplo se plasmará la solución del Requisito Funcional “Despachar Tránsito”.

Primeramente el usuario se encuentra en la ventana principal que presenta un menú que contiene todas las funcionalidades que se llevan a cabo para la realización de los regímenes de Tránsito y Transferencia. Selecciona la opción Tránsito XML para realizar el despacho del Tránsito por medio de fichero.

Capítulo 2. Diseño e Implementación

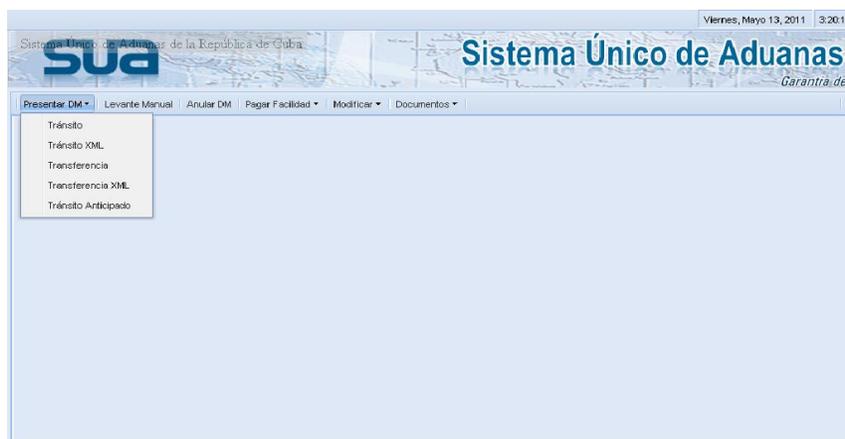


Figura 18. Página Principal

Levantando la pantalla que permite buscar y subir el fichero.



Figura 19. Pantalla lista para subir el fichero

Al presionar en el botón Adicionar se levanta una ventana para realizar la búsqueda del fichero.

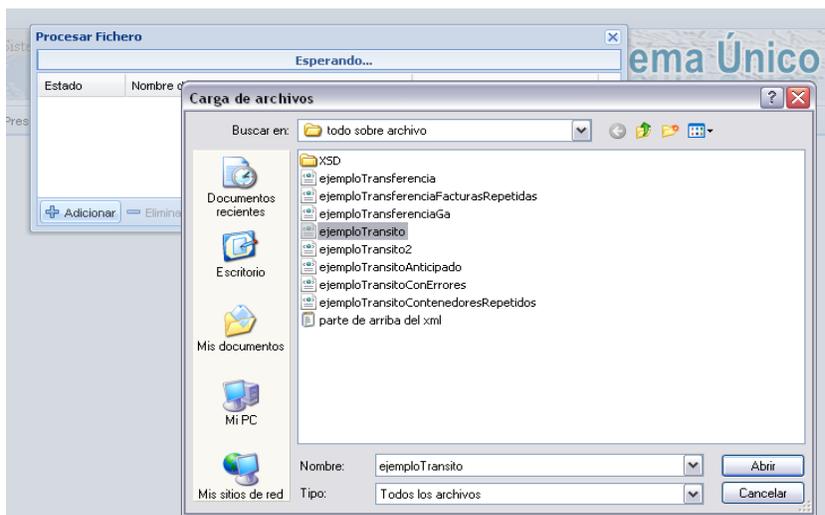


Figura 20. Pantalla para la búsqueda del fichero.

Luego pueden ser escogidas la cantidad de ficheros que se deseen procesar al mismo tiempo.



Figura 21. Ventana con ficheros para procesar

Se selecciona en el botón llamado Subir para enviar toda la información contenida dentro del XML para que sea validada y sea enviada al servicio que la va a recibir, el XML es convertido durante este proceso en un arreglo para su mejor funcionamiento. Ya aquí se hace uso del controlador frontal que es el encargado de definir según la URL hacia que módulo y acción debe llevar la información. El servicio que no es más que una funcionalidad se encarga de realizar todas las validaciones de negocio y formularios, es el responsable de insertar el fichero o modificarlo según lo deseado.

En caso de haber algún tipo de error, este es procesado y enviado al usuario para que presente conocimiento del mismo y así pueda corregirlo.

Si se envían errores se le muestran al usuario.



Figura 24. Pantalla con notificación de errores.

Y al seleccionar el botón Errores se muestra la siguiente ventana.

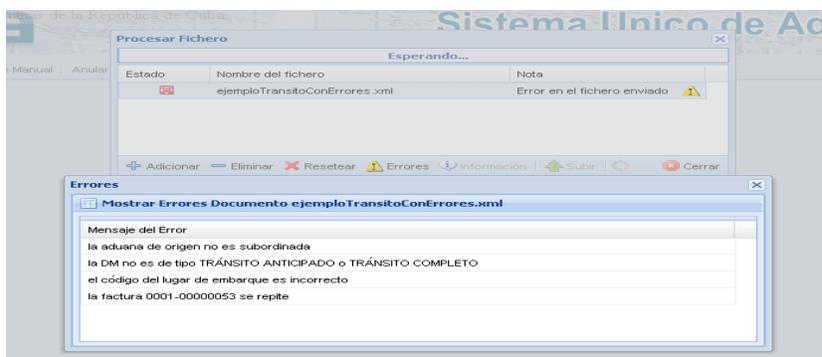


Figura 25. Pantalla que muestra los errores.

En caso que todo haya fluido correctamente también se le notifica al usuario.



Figura26. Pantalla para mostrar culminación satisfactoria

Al seleccionar el botón Información se muestran datos generados con la inserción del fichero procesado.

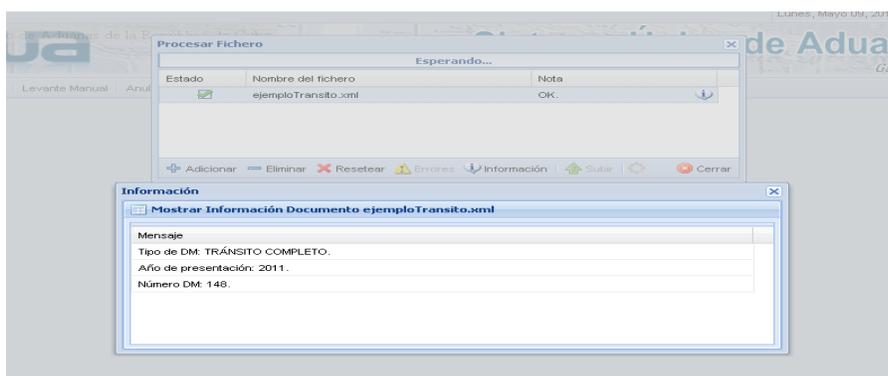


Figura27. Pantalla que muestra datos relevantes.

2.4 Conclusiones Parciales

Se obtuvieron como resultado, una serie de diagramas que fueron la base utilizada para dar cumplimiento de manera satisfactoria a la implementación del sistema. Se logró incorporar a la solución obtenida todos los requerimientos previstos con un bajo nivel de funcionalidades, minimizando la complejidad de implementación de las clases y atribuyendo la responsabilidad equitativamente, logrando también un alto por ciento de reutilización. Se aprovechó al máximo las posibilidades brindadas por los frameworks utilizados, logrando un correcto tratamiento de errores y comunicación entre las capas previstas por el patrón arquitectónico MVC.

Capítulo 3. Constatación de la Solución

3.1 Introducción

En el presente capítulo se realizarán una serie de pruebas para la validación de la solución obtenida. A través de una serie de casos de pruebas realizados, se comprobará si todos los requerimientos sobre los procesos de tránsito y transferencia obtenidos en el análisis, fueron cumplidos satisfactoriamente.

3.2 Pruebas de Software

Las Pruebas son la actividad en la cual se somete a un sistema o uno de sus componentes a una evaluación de los resultados que arroja en base a la ejecución de éste en condiciones específicas.

Los Casos de Pruebas son un conjunto de entradas y condiciones que arrojan resultados esperados desarrollados con un objetivo en particular.

Se conoce como error a la acción humana que produce un resultado incorrecto.

El Defecto es la manifestación de un error en el software. Es encontrado porque causa una falla, la cual es una desviación del servicio o resultado esperado.

Las pruebas de software, son los procesos que permiten verificar y revelar la calidad de un producto software. Son utilizadas para identificar posibles fallos de implementación, calidad, o usabilidad. Básicamente es una fase en el desarrollo de software consistente en probar las aplicaciones construidas.

¿Por qué probar?

Las fallas de software ocasionan graves pérdidas económicas y son más costosas de encontrar y reparar después de la construcción. También para evitar plazos y presupuestos incumplidos, insatisfacción del usuario, escasa productividad y mala calidad en el software producido y finalmente la pérdida de clientes.

3.2.1 Pruebas Funcionales o de Caja Negra

Se denominan pruebas funcionales o Functional Testing, a las pruebas de software que tienen por objetivo probar que los sistemas desarrollados, cumplan con las funciones específicas para los cuales han sido creados, es común que este tipo de pruebas sean desarrolladas por analistas de pruebas con apoyo de algunos usuarios finales, esta etapa suele ser la última etapa de pruebas y al dar conformidad sobre esta el paso siguiente es el pase a producción.(45)

A este tipo de pruebas se les denomina también pruebas de comportamiento o pruebas de caja negra, ya que los testers o analistas de pruebas, no enfocan su atención a como se generan las respuestas del sistema, básicamente el enfoque de este tipo de prueba se basa en el análisis de los datos de entrada y en los de salida, esto generalmente se define en los casos de prueba preparados antes del inicio de las pruebas.(45)

Las pruebas funcionales llevadas a cabo en el sistema, fueron realizadas manualmente por los analistas de pruebas del proyecto que desarrolla el GINA. Basados principalmente en los requerimientos definidos por el cliente (AGR), se planificaron una serie de casos de pruebas que incluye todos los escenarios previstos para los procesos de Tránsito y Transferencia.

A continuación se muestra ejemplo de un caso de pruebas realizado para el requisito funcional Despachar Tránsito. Durante la puesta en práctica de los casos de pruebas realizados para cada requisito funcional, incluyendo el ejemplo a presentar, se identificaron varias no conformidades, la mayoría sobre validación de campos en las interfaces. También se evidenciaron en menor grado, inconformidades con el envío de peticiones AJAX, recomendaciones sobre notificaciones que se necesitaban realizar cuando se insertaban ficheros y orden de realización de validaciones a nivel de servidor. Todas las inconformidades detectadas fueron resueltas en un cien por ciento, contribuyendo a la mejora del funcionamiento.

Capítulo 3. Constatación de la Solución

1.1 Requisitos a probar

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
2: Despachar Tránsito	Permitir realizar el Despacho de Tránsito con la información exigida por la aduana.	EP 2.1: Despachar Tránsito.	1: Seleccionar opción Despachar Tránsito. 2: Muestra una ventana para captar los datos de la Tránsito.
		EP 2.2: Registrar los datos de la Declaración de Mercancías en Tránsito: Código Declarante Nombre Declarante Apellidos Declarante Código Entidad Nombre Entidad No. Manifiesto BL GA Lugar de Embarque Aduana Origen Total de Bultos Total a Pagar Valor de Factura Deposito Origen Descripción Moneda de Pago No. Factura Precio Bruto Total No Contenedores	Valida los datos contenidos en la declaración de mercancías en Tránsito.

Capítulo 3. Constatación de la Solución

Juego de datos a probar para el requisito Despachar Tránsito

1.1.2 Juegos de datos a probar (Despachar Tránsito)

Id del escenario	Escenario	Código Declarante	Nombre Declarante	Apellidos Declarante	Código Entidad	Nombre Entidad	Aduana Origen	Deposito Origen	No. Manifiesto	Factura	Descripción	BL	GA	Total de Bultos	Total a Pagar	Moneda de Pago	Lugar de Embarque	Peso Bruto Total	Valor Factura	Contenedores	Respuesta del sistema	Resultado de la prueba	
EC 2.1	Despachar Tránsito	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	Muestra la ventana de Transferencia	Muestra la ventana de Transito correctamente con todos los datos que lleva.
EC 2.2	Registrar los datos de la Declaración de Mercancía en Tránsito	V	NA	NA	V	NA	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	Notificar el número que se le asigna a la Declaración de Mercancía en tránsito este debe ser consecutivo.	Satisfactoriamente muestra el Valor de la DM insertada que es Tipo de DM, Año de presentación: 2011, Numero Dm que este es consecutivo

Los Casos de Pruebas que se realizaron se encuentran especificados en el documento Diseño de Casos de Pruebas del módulo Tránsito y Transferencia.(46)

Capítulo 3. Constatación de la Solución

Se presentaron además, otra serie de casos de pruebas para los demás requisitos. A continuación se enumeran cada una de los mismos:

1. Requisito Despachar Transferencia:
 - EC 1.1 Despachar Transferencia.
 - EC 1.2 Registrar los datos de la Declaración de Mercancías en Transferencia.
2. Requisito Despachar Tránsito:
 - EC 2.1 Despachar Tránsito.
 - EC 1.2 Registrar los datos de la Declaración de Mercancías en Tránsito.
3. Modificar Tránsito:
 - EC 3.1 Modificar DM de Tránsito.
 - EC 3.2 Registrar el número de la DM y año de presentación de la declaración de mercancías que se quiere modificar.
 - EC 3.3 Registrar los datos que se quiere modificar de la declaración de mercancías en tránsito.
4. Modificar Transferencia:
 - EC 4.1 Modificar Transferencia.
 - EC 4.2 Registrar el número de la DM y año de presentación de la declaración de mercancías que se quiere modificar.
 - EC 4.3 Registrar los datos que se quiere modificar de la declaración de mercancías en transferencia.
5. Anular Transferencia y Tránsito:
 - EC 5.1 Anular Transferencia
 - EC 5.2 Registrar el número de la DM y año de presentación de la declaración de mercancías que se quiere Anular.
 - EC 5.3 Registrar los datos de la declaración de mercancías de Transferencia que se quiere anular.
6. Despachar Tránsito Anticipado:
 - EC 6.1 Despachar Tránsito Anticipado.
 - EC 6.2 Registrar los datos de la Declaración de Mercancías en Tránsito Anticipado.
7. Pagar Facilidad de Tránsito Anticipado:

Capítulo 3. Constatación de la Solución

- EC 7.1 Pagar Facilidad Tránsito Anticipado.
- EC 7.2 Registrar el número de la DM y año de presentación.
- EC 7.3 Registrar los datos de la declaración de mercancías de Tránsito Anticipado que se quiere pagar.

3.3 Impacto de la Solución Obtenida

Luego de haber culminado la implementación y realizado las pruebas funcionales, se procedió a la implantación del sistema en la Aduana General de la República de Cuba para detectar inconformidades y medir el funcionamiento en el ámbito real. Se utilizó en el Puerto de la Habana y el Aeropuerto Internacional José Martí, para así poder realizar las prácticas tanto en la especialización del tráfico marítimo como aéreo.

Transcurrido el mes de implantado el sistema, arrojó una serie de información que permitió realizar comparaciones entre la operaciones que se realizaban y las nuevas propuestas.

Dentro de los resultados de envergadura, se pone de manifiesto, la derogación de responsabilidad, sobre el control de la información de destino, hacia los depósitos, posibilitando con este cambio, la reducción de las modificaciones en las solicitudes de despacho, que traían consigo la pérdida de tiempo y de dinero al país. El sistema que está implantado actualmente, y debido a que los depósitos no tienen la autoridad para realizar modificaciones al despacho de una mercancía en Tránsito o Transferencia, trae consigo que los atrasos sean de días completos, porque solo el declarante y únicamente a través de las aduanas de despacho, puede realizar las modificaciones pertinentes, además que se tiene que realizar todo el procedimiento nuevamente.

Otro de los resultados importantes se basa en la introducción del fichero para el despacho de los regímenes de Tránsito y Transferencia, lo que trae consigo numerosas posibilidades, como son:

- La rapidez con que se realiza un despacho es equivalente a 3 segundos, cifra que supera el tiempo que se demora un trabajador de la aduana en introducir manualmente los valores.
- Ahorro de tiempo, debido a que el declarante a la hora de realizar la solicitud de despacho, ya no necesita moverse de su puesto de trabajo.

3.4 Conclusiones Parciales

En el capítulo se realizó la constatación del sistema construido mediante pruebas funcionales, incluyendo valoraciones obtenidas por la Aduana General de la República de Cuba. Se construyeron Casos de Pruebas, y sus escenarios correspondientes, que brindaron resultados insatisfactorios, dando paso a la vacunación de los problemas, contribuyendo de esta manera a la mejora del módulo desarrollado y llevándolo en conjunto con las valoraciones propuestas por los especialistas que le dieron seguimiento en las aduanas Puerto de la Habana y el Aeropuerto Internacional José Martí, a un cien por ciento de su funcionamiento.

Conclusiones

En el presente trabajo se recogen una serie de problemas que presentaba la Aduana General de la República de Cuba, referente a los procedimientos de Tránsito y Transferencia que se llevan a cabo por el sistema SUA, definiendo de esta manera un problema a resolver y objetivos para dar cumplimiento a la situación.

Luego de realizado un estudio del estado del arte y una profunda investigación sobre las buenas prácticas llevadas a cabo en el mundo, se logró dar cumplimiento a los objetivos planteados, logrando contribuir a la mejora del rendimiento, mantenimiento y funcionamiento de los regímenes de Tránsito y Transferencia.

Como parte de la solución obtenida se lograron generar una serie de artefactos, que fueron la base para la posterior implementación del sistema, destacando el diagrama de secuencia orientado a actividades, solución obtenida por el Proyecto, para agilizar la implementación del módulo.

Se utilizó, en la creación del sistema, el patrón arquitectónico MVC, trayendo como resultado, que quedara separada la arquitectura en tres niveles distintos e independientes, logrando facilitar el mantenimiento en caso que sea necesario. También se incorporó a la solución, el trabajo con ficheros estructurados sobre la base del lenguaje XML, fomentando de esta manera la compatibilidad con la mayoría de los sistemas y la agilización de los procedimientos de Tránsito y Transferencia, dando paso, en conjunto con la derogación de la responsabilidad del procesamiento de la información de destino de la carga a los depósitos, a la mejora del rendimiento y funcionamiento, según pruebas y valoraciones que se llevaron a cabo por la AGR.

Se logró realizar pruebas funcionales al sistema, a través de casos de pruebas realizados por los especialistas del proyecto, brindando inconformidades que propiciaron la modificación de actividades, que aportaron como resultado la culminación de un producto cien por ciento utilizable.

Recomendaciones

Una vez cumplido con los objetivos de la investigación, teniendo en cuenta las experiencias obtenidas en la misma, se recomienda:

- La liberación de la aplicación por el equipo de calidad de software de la universidad.
- La implementación de reportes con el objetivo de favorecer el control y las estadísticas de los procedimientos de Tránsito y Transferencia realizados.
- La implementación del resto de los subsistemas con que se relaciona el módulo, para tener un completo funcionamiento del mismo.
- Extender el uso del sistema a todas las aduanas de despacho del país.

Bibliografía

1. **Agenda Magna.***Agenda Magna*. [En línea] 27 de Febrero de 2009. [Citado el: 24 de Febrero de 2011.] <http://agendamagna.wordpress.com/2009/02/26/glosario-de-terminos-aduaneros/>.
2. *La Importancia de las Aduanas en el Comercio Exterior.* **Rodriguez, Gorge I.Arce**. San Marcos : s.n., 2008.
3. **Definicion.de.***Definicion.de*. [En línea] [Citado el: 22 de Febrero de 2011.] <http://definicion.de/comercio/>.
4. **Gestiopolis.com.***Gestiopolis.com*. [En línea] 2008. [Citado el: 11 de Febrero de 2011.] <http://www.gestiopolis.com/recursos/documentos/fulldocs/eco1/anteduanas.htm>.
5. **Monografias.com.***Monografias.com*. [En línea] [Citado el: 11 de Febrero de 2011.] <http://www.monografias.com/trabajos24/legislacion-aduanera/legislacion-aduanera.shtml>.
6. *Aduana General de la República de Cuba*. [En línea] 2008. [Citado el: 8 de Febrero de 2011.] <http://www.aduana.co.cu/>.
7. **Sistemas Informaticos DDS.***Sistemas Informaticos DDS*. [En línea] 2010. [Citado el: 3 de Marzo de 2011.] <http://www.ddsoftware.com.ar/software-comercio-exterior.html>.
8. **UNCTAD.***SIDUNEA*. [En línea] [Citado el: 3 de Marzo de 2011.] <http://www.asycuda.org/spanish/>.
9. **Korea Customs Service.***Korea Customs Service*. [En línea] 2006. [Citado el: 2 de Marzo de 2011.] <http://english.customs.go.kr/>.
10. **HM Revenue & Customs.***HM Revenue & Customs*. [En línea] [Citado el: 4 de Marzo de 2011.] http://customs.hmrc.gov.uk/channelsPortalWebApp/channelsPortalWebApp.portal?_nfpb=true&_pageLabel=pageImport_Home.
11. **Martínez, Jenni Manso.***Procedimiento para la Ingeniería de Requisitos en el Departamento de Desarrollo de Soluciones para la Aduana del CEIGE*. 2010.
12. **Reyero, Eusebio.** Will Web For Food. *Metodologías de diseño*. [En línea] [Citado el: 15 de Febrero de 2011.]
13. **Larman, Craig.***UML y Patrones*. 1999. ISBN 970-1 7-0261-1.
14. **Larman, Craic.***UML y Patrones. Introducción al Analisis y Diseño OO*. México : Prentice Hall, 1999.
15. **Fowler, Martin.***Patterns of Enterprise Application Architecture*. ISBN: 0-32112-742-0.

Bibliografía

16. **Fabien Potencier, François Zaninotto.** *Symfony la guía definitiva*. 2008.
17. **Universidad Tecnológica Nacional - Facultad Regional Tucumán** . *Lenguajes de Programación*. [En línea] 2000. [Citado el: 22 de Febrero de 2011.] <http://frt.utn.edu.ar/sistemas/paradigmas/lenguajes.htm>.
18. **Pérez, Javier Eguíluz.** *Introducción a XHTML*. 2007.
19. —. *Introducción a CSS*. 2007.
20. **Pixelkit.** *Diccionario Web - Estudio de Contenidos Digitales*. [En línea] 2009. [Citado el: 2 de Febrero de 2011.] <http://pixelkit.cl/diccionario-web/>.
21. **Universidad Carlos III de Madrid.** *JavaScript*. [En línea] 2006. [Citado el: 2 de Febrero de 2011.] http://perso.wanadoo.es/javascript_12/.
22. **PHP.net.** PHP: Hypertext Preprocessor. *PHP: General Information*. [En línea] 2001. [Citado el: 20 de Febrero de 2011.] <http://www.php.net/manual/en/faq.general.php>.
23. Programación en Castellano. *¿Por qué elegir PHP?* [En línea] 1998. [Citado el: 22 de Febrero de 2011.] http://www.programacion.com/articulo/por_que_elegir_php_143.
24. **World Wide Web Consortiun.** *Guía Breve de Tecnologías XML*. [En línea] 09 de Enero de 2008. [Citado el: 04 de Marzo de 2011.] <http://www.w3c.es/divulgacion/guiasbreves/tecnologiasxml>.
25. **DesarrolloWeb.com.** *DesarrolloWeb.com*. [En línea] [Citado el: 04 de Marzo de 2011.] <http://www.desarrolloweb.com/articulos/449.php>.
26. **Rodríguez, Jose Antonio Cobo.** *Línea Base Arquitectónica para el Polo Sistemas Tributarios y de Aduanas*. 2008.
27. **CodeBox.** *Glosario*. [En línea] 2008. [Citado el: 4 de Febrero de 2011.] <http://www.codebox.es/glosario>.
28. **Johnson, R. E. y Foote, B.** Designing Reusable Classes. *Journal of Object-Oriented Programming*. 1988.
29. **Comunidad de Desarrollo de ExtJs** . *ExtJs en Español*. [En línea] [Citado el: 4 de Febrero de 2011.] <http://extjs.es/>.
30. **Escuela de Ingeniería Informática Universidad de Oviedo.** *Entornos de Desarrollo Integrado*. [En línea] 2008. [Citado el: 6 de Febrero de 2011.] <http://petra.euitio.uniovi.es/~i1667065/HD/documentos/Entornos%20de%20Desarrollo%20Integrado.pdf>.
31. **Oracle Corporation.** *Portal del IDE Java de Código Abierto*. [En línea] 2010. [Citado el: 6 de Febrero de 2011.] http://netbeans.org/index_es.html.

Bibliografía

32. **Visual Paradigm International.** *UML, BPMN and Database Tool for Software Development*. [En línea] [Citado el: 8 de Febrero de 2011.] <http://www.visual-paradigm.com/>.
33. **CAVSI.com.** *¿Qué es un Sistema Gestor de Bases de Datos o SGBD?* [En línea] 2007. [Citado el: 6 de Febrero de 2011.] <http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sgbd/>.
34. **Oracle.** *Oracle*. [En línea] [Citado el: 02 de Febrero de 2011.] <http://www.oracle.com/us/index.html>.
35. **monografias.com.** *monografias.com*. [En línea] [Citado el: 11 de Marzo de 2011.] <http://www.monografias.com/trabajos/anaydisis/anaydisis.shtml>.
36. **Centro de Informatización de la Gestión de Entidades (CEIGE).** *Modelo de Diseño del GINA para el Tránsito y Transferencia*. 2011.
37. **Departamento de Ciencias de Computación Universidad de Chile.** *DCC*. [En línea] [Citado el: 10 de Marzo de 2011.] <http://www.dcc.uchile.cl/~psalinas/uml/interaccion.html>.
38. **Pozo, Pedro.** *Cliekar.com*. [En línea] [Citado el: 11 de Marzo de 2011.] <http://www.cliekar.com/manuales/uml/diagramasinteraccion.aspx>.
39. **Centro de Informatización de la Gestión de Entidades (CEIGE).** *Descripción de las tablas del módulo Tránsito y Transferencia*. 2011.
40. **Norman, Fenton E.** *Software Metrics A Rigorous Approach*. s.l. : Chapman & Hall, 1991.
41. **Roger, Pressman S.** *Ingeniería de Software Un enfoque Práctico*. México : Mc Graw Hill, 1998. Cuarta Edición.
42. **Ejiogo O., Len.** *Software Engineering whit Formal Metrics*. s.l. : QED Technical, 1991.
43. **Guillot, Carlos Fernández.** *ITESO*. [En línea] [Citado el: 11 de Marzo de 2011.] <http://progra.iteso.mx>.
44. **CEIGE, Dpto Aduana.** *Propuesta de un Estándar de Codificación*. 2011.
45. **B, Ing. Alexander Oré.** *CalidadySoftware.com*. [En línea] 2009. [Citado el: 20 de Mayo de 2011.] http://www.calidadysotware.com/testing/pruebas_funcionales.php.
46. **Centro de Informatización de la Gestión de Entidades (CEIGE).** *Diseño de Casos de Pruebas para el módulo Tránsito y Transferencia*. 2011.
47. **Ramírez, Arturo.** *Las Aduanas*. [En línea] Enero de 2005. [Citado el: 10 de Febrero de 2011.] http://www.aduanas.com.ve/boletines/boletin_12/aduana.htm.

Bibliografía

48. **Real Academia Española.** *Diccionario*. [En línea] [Citado el: 8 de Febrero de 2011.] http://buscon.rae.es/drae/SrvltGUIBusUsual?TIPO_HTML=2&TIPO_BUS=3&LEMA=aduana.
49. *ADUANA SIGLO XXI*. D, Lic. **LUIS ML. SÁNCHEZ**. Santo Domingo, D.N : s.n., 2007.
50. **Real Academia Española.** *Diccionario*. [En línea] [Citado el: 8 de Febrero de 2011.] http://buscon.rae.es/drae/SrvltGUIBusUsual?TIPO_HTML=2&TIPO_BUS=3&LEMA=historial.
51. **Arellano, Ana Alejandra Febres.** RRPP/ UNCTAD-SIDUNEA. *Conociendo al SIDUNEA - Sistema Aduanero Automatizado*. [En línea] Marzo de 2004. [Citado el: 16 de Febrero de 2011.] <http://www.gestiopolis.com/canales2/economia/sidunea.htm>.

Glosario de Términos

- **Ajax:** (Asynchronous JavaScript And XML) JavaScript asíncrono y XML, es una técnica de desarrollo web para crear aplicaciones interactivas. Esta se ejecuta en el cliente y mantiene comunicación asíncrona con el servidor en segundo plano. Esto significa aumentar la interactividad, velocidad y usabilidad en la misma.
- **BPMN:** Es una notación gráfica estandarizada que permite el modelado de procesos de negocio.
- **DHTML:** (Dynamic HTML) Designa el conjunto de técnicas que permiten crear sitios web interactivos utilizando una combinación de lenguaje HTML estático, un lenguaje interpretado en el lado del cliente (como JavaScript), el lenguaje de hojas de estilo en cascada (CSS) y la jerarquía de objetos de un DOM.
- **DOM:** (DocumentObjectModel) Modelo en Objetos para la representación de Documentos. Es un modelo computacional a través de la cual los programas y scripts pueden acceder y modificar dinámicamente el contenido, estructura y estilo de los documentos HTML y XML.
- **GINA:** Sistema de Gestión Integral de Aduana. Sistema desarrollado por el centro CEIGE, en la Universidad de las Ciencias Informáticas, para la gestión de los procesos asociados a la Aduana General de la República de Cuba.
- **HTML:** Siglas de Hyper Text Markup Language (Lenguaje de Marcado de Hipertexto), es el lenguaje de marcado predominante para la elaboración de páginas web.
- **Java:** Lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++.
- **Javascript:** Es un lenguaje de programación interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas web, con una sintaxis semejante a la del lenguaje Java y el lenguaje C.
- **JSON:**(JavaScript Object Notation) Notación de Objetos JavaScript, es un formato ligero para el intercambio de datos. JSON es un subconjunto de la notación literal de objetos de JavaScript que no requiere el uso de XML.
- **Linux:** Es una implementación de libre distribución UNIX para computadoras personales (PC), servidores, y estaciones de trabajo.

Glosario de Términos

- **ORM:** (Object-Relational mapping) Es una técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional, utilizando un motor de persistencia. En la práctica esto crea una base de datos orientada a objetos virtual, sobre la base de datos relacional.
- **PHP:** Es un lenguaje de script incrustado dentro del HTML. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas de sí mismo. La meta del lenguaje es permitir rápidamente a los desarrolladores la generación dinámica de páginas.
- **POST:** Es un método de petición que se refiere normalmente a la invocación de procesos que generan datos que serán devueltos como respuesta a la petición. Además se utiliza para aportar datos de entrada a esos programas.
- **Unix:** Es un sistema operativo portable, multitarea y multiusuario; desarrollado, en principio, en 1969 por un grupo de empleados de los laboratorios Bell de AT&T, entre los que figuran Ken Thompson, Dennis Ritchie y Douglas McIlroy.
- **URL:** Acrónimo de Uniform Resource Locator, es decir, localizador uniforme de recurso. Es una secuencia de caracteres, de acuerdo a un formato estándar, que se usa para nombrar recursos.
- **WEB:** Es un medio de comunicación de texto, gráficos y otros objetos multimedia a través de Internet, es decir, la web es un sistema de hipertexto que utiliza Internet como su mecanismo de transporte o desde otro punto de vista, una forma gráfica de explorar Internet.
- **Windows:** Es el nombre de una serie de sistemas operativos desarrollados por Microsoft desde 1981.
- **XML:** Siglas en inglés de Extensible Markup Language (lenguaje de marcas extensibles), es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C).