

**Universidad de las Ciencias Informáticas**

**Facultad 3**



**Título: Implementación del módulo Configuración de  
Tablas de Control versión 1. 1 del Sistema de Gestión  
Integral Aduanera**

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

**Autor:** Jorge Ariel Mancha Alvarez

**Tutor (es):** 1. Ing. Raymond Wedeen Gamboa

2. Ing. Liliam López García

Habana – junio 2015

## DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo al <nombre área> de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

Jorge Ariel Mancha Alvares

---

Ing. Raymond Weeden Gamboa

---

Ing. Liliam López García

---

## DATOS DE CONTACTO

Ing. Raymond Weeden Gamboa. Ingeniero en Ciencias Informáticas. Graduado en 2009. 4 Años de experiencia en el tema. Profesor Instructor.

Ing. Liliam López García. Ingeniera en Ciencias Informáticas. Graduada en 2013. 1 Año de experiencia en el tema. RGA

## AGRADECIMIENTOS

A mis padres gracias por todo su apoyo y por todos los consejos, por siempre creer en mí, por siempre estar presentes cuando los necesite, por todos los bien merecidos alones de oreja que me daban por estar bobiando y ser unas de las personas más importantes den mi vida

A mis dos hermanos mayores Heidy y Jorge Alberto gracias por estar siempre ahí para mi, son los mejores hermanos que alguien pudiera desear por alentarme siempre a seguir adelante y a no rendirme nunca, y por todo el cariño que me an dado

A toda mi familia po siempre creer en mi, en que podía llegar a graduarme, y todos sus consejos que me sirvieron de mucho

A mi nueva y gran amiga Eliane, que a pesar de conocerme hace muy poco me brindo toda su ayuda incindicional, te has convertido en alguien indispensable para mi

A mi amiga y colega, Lilian por todas las horas que dedico a ayudarme a culminar este trabajo y todos sus consejos y por creer en mí

A mis amigos, colegas y hermanos Felix y Ernesto por todo el apoyo que me brindaron, por las largas noches que pasamos programando para culminar muchos trabajos, les deseo grandes éxitos en su vida profesional

A mis amigos Arai, los jimaguas por soportarme todos estos años, se que aveces puedo ser un poco insoportable, me alegra mucho haber compartido estos años con ustedes

A mis tutores Liliam, Raymond por todo lo que me enseñaron durante este curso, las largas noches en los laboratorios programando

A Adrian Naranjo que a pesar de no ser mi tutor, me brindo su ayuda en todo momento

## DEDICATORIA

Dedico este trabajo de diploma a mis padres, los cuales son responsables de que me encuentre hoy aquí, por todo su amor y comprensión, me siento orgulloso de ser su hijo, este logro esta dedicado especialmente para ustedes, también a mi querida hermana Heidy y mi hermano Jorge Alberto

## RESUMEN

En la actualidad, el Sistema para la Gestión Integral Aduanera de la Aduana General de la República de Cuba no cuenta con una interfaz única que permita gestionar los nuevos nomencladores que se quieran introducir al sistema. Esta situación implica que para crear un nomenclador, los técnicos aduaneros deban tener conocimientos sobre el Sistema Gestor de Base de Datos y Symfony. Por lo que se torna el trabajo más complicado y se presentan errores por la mala manipulación del código. El objetivo de esta tesis de diploma es agregar nuevas funcionalidades al módulo: Tablas de Control, que pertenece al Sistema para la Gestión Integral Aduanera, con el propósito de que este será capaz de gestionar nomencladores dinámicamente a través de una única interfaz que resulta en una mejor organización de la información.

Con este propósito, se hace un estudio de los diferentes sistemas que existen en Cuba y el mundo que resuelven la gestión de nomencladores, pero debido a las deficiencias que presentan, se hizo necesario implementar una nueva versión de dicho módulo. Finalmente, con las funcionalidades desarrolladas, es posible mejorar las condiciones de trabajo de los usuarios finales a través de una única interfaz, ya que logra optimizar el trabajo de los funcionarios y minimizar los errores que puedan existir debido a: mal manejo de los códigos o el desconocimiento de ellos.

## PALABRAS CLAVE

Interfaz, Nomencladores, Tablas de Control

## **Abstract**

In the present day, the Integral Management System for Customs of the General Customs of Republic of Cuba doesn't have a single interface to manage new nomenclators which will want to enter to the system. This situation means that to create a nomenclator, customs technicians must have knowledge about the Database Manager and System Symphony. So the work becomes more complicated and errors occur by mishandling of the code. The objective of this diploma thesis is to add new functionalities to the module: Control Tables, which belongs to the Integrated Management System for Customs, with the purpose that this will be able to manage nomenclators dynamically through a single interface resulting in a better organization of information. With this purpose, it makes a study of the different systems which exist in Cuba and the world that solve the nomenclators management, but due to deficiencies in them, it was necessary to implement a new version of the module. Finally, with the developed functionalities, is possible to improve the working conditions of the end users through a single interface, because it achieves to optimize the work of the officers and minimize errors that may exist because to: wrong handling codes or the unknowledge of them.

## **KEYWORDS**

Interface, Nomenclators, Control Tables

## TABLA DE CONTENIDOS

AGRADECIMIENTOS.....	I
DEDICATORIA .....	II
RESUMEN.....	III
INTRODUCCIÓN.....	8
<b>CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA .....</b>	<b>13</b>
1.1 Introducción .....	13
1.2 Sistemas similares .....	13
1.3 Lenguajes, metodología y herramientas.....	20
1.4 Conclusiones del capítulo .....	30
<b>CAPÍTULO 2: ANÁLISIS Y DISEÑO DE LA PROPUESTA DE SOLUCIÓN.....</b>	<b>31</b>
2.1 Introducción. ....	31
2.2 Propuesta de solución.....	31
2.3 Modelo Conceptual. ....	31
2.4 Técnicas para la Captura de Requisitos .....	32
2.5 Requisitos.....	33
2.5.1 Requisitos funcionales .....	33
2.5.2 Requisitos no funcionales .....	34
2.5.3 Descripción de los requisitos del sistema .....	35
2.5.4 Técnicas para la validación de los requisitos .....	41
2.6 Patrones de diseño utilizados en la solución.....	42
2.7 Diagrama de Clases del Diseño con estereotipos Web. ....	45
2.8 Diagrama de Secuencia. ....	46
2.9 Métricas para la validación del diseño .....	47
2.10 Modelo de Datos. ....	53
2.11 Conclusiones del Capítulo. ....	54
<b>CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA.....</b>	<b>55</b>
3.1 Introducción .....	55
3.2 Modelo de implementación del sistema.....	55
3.2.1 Estándares de codificación.....	55
3.2.2 Tratamiento de errores .....	57
3.2.3 Diagrama de despliegue .....	58
3.3 Pruebas de validación de la solución .....	59
3.3.1 Aplicación de pruebas de caja negra o funcional.....	59
3.3.2 Aplicación de pruebas de caja blanca .....	62
3.4 Validación de la investigación .....	65



3.5 Conclusiones .....	70
CONCLUSIONES.....	71
RECOMENDACIONES .....	72
BIBLIOGRAFÍA.....	73
ANEXOS.....	75
GLOSARIO.....	76

## ÍNDICE DE FIGURAS

Figura 1 Fases e Iteraciones .....	20
Figura 2 Modelo Conceptual .....	32
Figura 3 Prototipo Crear Tabla.....	37
Figura 4 Diagrama de clases del diseño "Eliminar TC" .....	46
Figura 5 Diagrama de secuencia "Registrar TC" .....	47
Figura 6 Cantidad de procedimientos por clase para la métrica TOC.....	49
Figura 7 Resultados de la métrica Tamaño de Clase .....	50
Figura 8 Resultados de la métrica RC .....	52
Figura 9 Diagrama entidad-relación .....	54
Figura 10 Aplicación de los estándares de codificación.....	57
Figura 11 Tratamiento de errores .....	58
Figura 12 Diagrama de despliegue .....	58
Figura 13 Resultados de las pruebas de caja negra.....	60
Figura 14 Método EliminarTabla.....	63
Figura 15 Grafo del flujo correspondiente a la función EliminarTabla.....	63
Figura 16 Definición de las pruebas de Symfony .....	65
Figura 17 Resultado de la prueba de Symfony .....	65

## ÍNDICE DE TABLAS

Tabla 1: Comparación entre los sistemas estudiados .....	19
Tabla 2 Especificación de requisitos .....	33
Tabla 3 Descripción del requisito "Registrar TC" .....	35
Tabla 4 Descripción del requisito "Modificar TC" .....	37
Tabla 5 Descripción del requisito "Eliminar TC" .....	39
Tabla 6 Descripción del requisito "ListarTC" .....	40
Tabla 7 Umbrales de la métrica TOC.....	48
Tabla 8 Resultados por clase luego de la aplicación de la métrica TOC.....	49
Tabla 9 Umbrales de la métrica RC .....	51
Tabla 10 Resultados por clase luego de la aplicación de la métrica RC .....	51
Tabla 11 Descripción de las variables del requisito "Eliminar TC" .....	60
Tabla 12 Diseño del caso de prueba .....	61
Tabla 13 Resultados de la encuesta.....	68
Tabla 14 Antes y después de la implementación de la nueva versión del módulo de TC.....	69

## INTRODUCCIÓN

El creciente desarrollo de las Tecnologías de la Información y las Comunicaciones (TIC) ha marcado un cambio significativo en la sociedad con respecto a la información. El manejo de información es cada vez más dependiente de las TIC, pues genera grandes volúmenes de información, por lo que la necesidad de utilizar y acceder a estas de manera instantánea es de vital importancia a nivel mundial.

En Cuba el uso de las TIC está jugando un papel fundamental en las empresas logrando grandes mejoras en sus operaciones y beneficios para la economía. La informatización de las empresas cubanas es hoy una realidad, donde la Aduana General de la República de Cuba (AGR) está interesada en que se realice la informatización de sus procesos. La AGR es el órgano encargado del control en la frontera y fiscalización en la actividad vinculada al comercio exterior, garantizando de esta manera la seguridad y protección de la sociedad y de la economía del país (1).

A diario entran y salen personas o mercancías por los puertos y aeropuertos del país, generando grandes volúmenes de información, esta es almacenada en sistemas computacionales con bases de datos relacionales. Con el objetivo de mejorar los servicios y el control de la información se inicia la automatización de los procesos que tienen lugar en las distintas sucursales aduanales del país.

En la primera mitad de la década del 90 se implanta en Cuba el sistema SIDUNEA (Sistema Aduanero Automatizado) creado en Conferencia de Naciones Unidas para el Comercio y Desarrollo (UNCTAD). SIDUNEA presentaba un conjunto de módulos que resolvían en gran medida las necesidades que presentaban las aduanas del país en cuanto a la automatización de sus procesos, sustituyendo la mayoría de los procedimientos que se hacían de forma manual, pero no todos se pudieron implantar por incongruencias en su concepción con las características específicas del despacho de mercancías en Cuba y por el elevado costo de licencia y mantenimiento del sistema (2).

A raíz de estas dificultades en el Centro de Automatización para la Dirección y la Información (CADI) de la AGR se desarrolló el Sistema Automatizado de Despacho Mercantil (SADEM). Para la explotación de este sistema (SADEM) se hizo necesario el uso de nomencladores, facilitando la flexibilidad del sistema, además de tener organizada la información y así asegurar la consistencia de los datos.

Sin embargo al explorar la utilización de este sistema, se pudo detectar diferentes problemas que dificultaban el trabajo de los técnicos aduaneros, restando flexibilidad a las operaciones que realizaban, estos problemas fueron (3):

- Mayoritario y dependiente uso del teclado ante el ratón.
- Uso de un sistema multiusuario el cual trabaja directamente en la Base de Datos lo cual implica un riesgo de seguridad, una recarga del trabajo del servidor y no permite la conexión de muchos usuarios.
- Dificultad a la hora de actualizar las versiones a todos los clientes.

Por estas dificultades se decide cambiar el sistema a plataforma Web, surgiendo así el Sistema Único de Aduanas (SUA). Creado en el curso 2003-2004 entre la Universidad de las Ciencias Informáticas (UCI) y el CADI. Teniendo como objetivo principal desarrollar nuevas soluciones que cumplan con las necesidades de los trabajadores de la Aduana. SUA consta del módulo Tablas de Control (TC) que tiene como objetivo (3):

- Actualizar la información manteniendo el referente histórico y la consistencia de los datos.
- Posibilitar que cada acción sea auditable.
- Permitir la obtención de reportes para consultas.
- Replicar la información en todas las unidades y publicarla para terceros.

En la actualidad estos sistemas aún no responden a las necesidades que van surgiendo. A diario gran número de datos son generados por estos sistemas creando dificultad al acceder a estas. Tampoco comparte una base de datos única con los demás sistemas provocando redundancia de la información. No se ha logrado mediante una interfaz única gestionar los nomencladores existentes, lo que implica que cada vez que se crea un nomenclador sea necesario re-implementar parte del sistema. Asimismo, no permiten mantener un historial, lo que hace muy complicado realizar los procesos ordenados que dependen mucho de los nomencladores.

Por todo lo anteriormente descrito, el departamento de Soluciones Financieras y Aduanales del Centro de Informatización de Entidades (CEIGE) perteneciente a la UCI, en colaboración con el CADI, desarrollan el Sistema para la Gestión Integral Aduanera (GINA). Dicho sistema presenta como principal objetivo automatizar todos los procesos que se realizan en la aduana y está compuesto por varios módulos, siendo uno de estos el módulo de TC. Indispensable para el funcionamiento de los diferentes módulos del sistema que necesitan acceder a estos datos.

Con la automatización de las TC, se logra agilizar el trámite aduanero y optimizar la labor de los funcionarios. Además, permite conocer los diferentes nomencladores con la información que estos poseen y garantizar el histórico de los datos.

En la primera versión del sistema GINA se logró dar solución a todos los requisitos funcionales correspondientes al módulo de TC, pero en su uso diario surgieron nuevos requisitos, entre los que se encuentran: registrar una TC, modificarla o eliminarla a través de una interfaz única.

Actualmente para registrar una TC, la técnica aduanera encargada del trabajo con este módulo, debe poseer conocimientos sobre el Sistema Gestor de Base de Datos (SGBD), ya que debe registrarse de forma manual. Luego de creada la TC, se deben generar las clases del modelos correspondientes a la nueva TC, por lo que la técnica aduanera debe poseer también conocimientos sobre symfony 1.2.8. Estas dificultades traen consigo la ocurrencia de errores, ya sea por mala manipulación de los códigos o desconocimiento de los mismos, además de que la nueva TC pudiera no tener la estructura que fue definida para ella. Por esta razón el sistema se encuentra en el desarrollo de su versión 1.1.

Ante esta situación surge el siguiente **problema a resolver**: ¿Cómo lograr que el módulo de TC del sistema GINA permita un mejor control de los nomencladores?

Para dar solución a este problema se define como **objeto de estudio**: el proceso de estandarización de nomencladores en las TC.

El **campo de acción** abarca el proceso de gestión de las TC de la AGR.

Para ello se plantea como **objetivo general del trabajo**: Desarrollar las nuevas funcionalidades del módulo de TC del sistema GINA que permita lograr un mejor control de los nomencladores. Para dar cumplimiento a dicho objetivo general se plantean los siguientes **objetivos específicos**:

- Elaborar el marco teórico relativo a la investigación para obtener un estado del arte que permita tener un conocimiento amplio sobre el tema.
- Modelar la ingeniería de requisitos para garantizar el diseño del sistema que se desea desarrollar.
- Diseñar e implementar la solución modelada de manera que cumpla con las necesidades del cliente.
- Validar la solución mediante pruebas unitarias aplicadas al sistema para garantizar la calidad del código.

Para lograr el cumplimiento de los objetivos planteado anteriormente se definen como principales **tareas de la investigación**:

- Elaboración de la fundamentación teórica y la revisión bibliográfica.
- Especificación de los requisitos.
- Validación de los requisitos.
- Diseño de la base de datos.
- Describir el modelo de datos.
- Diseñar e implementar las interfaces de usuario.
- Diseñar el diagrama de clases del diseño.
- Diseñar los diagramas de secuencia.
- Implementar los requisitos.
- Diseñar los casos de prueba.
- Aplicar las pruebas al sistema.
- Documentar los resultados de las pruebas al sistema.

La presente investigación tiene como **pregunta científica**: ¿Se tendrá un mejor control de los nomencladores si se desarrolla una nueva versión del módulo de TC?

Los métodos de investigación utilizados fueron:

### **Métodos Teóricos**

- **Histórico lógico:** Se utiliza para realizar una revisión histórica del desarrollo de los sistemas aduaneros o que realicen la gestión de nomencladores y constatar teóricamente cómo ha evolucionado.
- **Analítico-sintético:** Permitió el procesamiento de la información y arribar a las conclusiones prácticas y teóricas de la investigación.

### **Métodos Empíricos**

- **Entrevistas:** Se realizan con el objetivo de adquirir una comprensión detallada de lo que necesitan los clientes y para poder conocer cómo podrían interactuar con el sistema y las dificultades a las que se enfrentan en la actualidad.
- **Observación:** Se utiliza para realizar una evaluación de la situación problemática en cuestión.

### **Estructuración del trabajo**

El trabajo consta de tres capítulos que cubren la fundamentación teórica, análisis y diseño de la solución propuesta e implementación y validación de la solución. Además de las conclusiones, recomendaciones, bibliografía y el glosario de términos.

**Capítulo 1 “Fundamentación teórica”:** En este capítulo se muestra la fundamentación teórica de la investigación. Se enuncian los principales conceptos que están asociados al dominio del problema. Se realiza el estudio crítico de las soluciones anteriores relacionadas con el problema planteado. Se relacionan las tendencias tecnológicas, herramientas, lenguajes empleados en el desarrollo y la metodología que guiara el desarrollo de la solución.

**Capítulo 2 “Análisis y diseño de la propuesta de solución”:** Este capítulo abarca el análisis y diseño del sistema, haciendo una descripción de los productos de trabajo que son generados en el desarrollo de estas fases. Se describen y validan los requisitos del sistema, se elaboran los prototipos de interfaz de usuario, se realiza el diseño haciendo uso de patrones y se aplican métricas para comprobar su validez.

**Capítulo 3 “Implementación y validación”:** En este capítulo se lleva a cabo la implementación del sistema a partir de los productos de trabajo concebidos en las etapas de análisis y diseño. Se realiza la validación de la solución propuesta mediante el uso de las distintas técnicas de evaluación de software, con el objetivo de determinar las posibles inconformidades; así como determinar si se cumplieron los objetivos propuestos inicialmente.

## CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

### 1.1 Introducción

En el presente capítulo se detallan los elementos fundamentales asociados al dominio del problema. Se realiza una investigación de los sistemas aduaneros o de los que realizan la gestión de los nomencladores<sup>1</sup>, utilizados a nivel mundial y en Cuba. Del mismo modo, se puntualizan aspectos importantes como: herramientas, metodología y lenguaje a utilizar para el desarrollo de la solución.

### 1.2 Sistemas similares

Se seleccionaron sistemas que realizan la gestión de nomencladores o sistemas aduaneros, como: GIA, SIDUNEA, GESIO, SADEM, SUA, GINA (versión 1.0) y el Sistema para la Gestión de Nomencladores. Teniéndose en cuenta para su estudio los siguientes indicadores: herramientas utilizadas, si son web o de escritorio, lenguajes, *framework*<sup>2</sup>, además del tipo de licencia que poseen y si permiten o no la creación de nuevos nomencladores.

#### GIA

El Sistema de Gestión Integral Aduanera (GIA) provee una completa solución para la gestión aduanera cubriendo todas las áreas de operación: importaciones, exportaciones y tránsitos. Además, facilita el control de inventarios de los recintos aduaneros, manifiestos marítimos, aéreos y terrestres. El sistema está basado en un código editable y configurable, que permite la gestión aduanera y mantiene el dinamismo en los controles requeridos. El proyecto GIA tiene como objetivo que las aduanas logren su autonomía, es decir, que garantice su propia libertad de operar y evolucionar. Este sistema cuenta con el módulo de Registro que es el encargado de realizar todo el manejo de lo referente a datos básicos (nomencladores) del sistema, en el mismo se mantienen Tasas, Aranceles, Acuerdos, Partidas y Certificados(4).

Características (4):

---

<sup>1</sup> Es un sistema de clasificación y de codificación, aplicado para designar conceptos fundamentales tales como países, aranceles, tarifas, monedas etc.

<sup>2</sup> Simplifica el desarrollo de una aplicación mediante la automatización de algunos de los patrones utilizados para resolver las tareas comunes. 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.



- Computación en la nube.
- Sistema web y privativo.
- Desarrollado en java y .Net.
- Gestor de Base de Datos Oracle y SQLServer.

## **SIDUNEA**

El Sistema computarizado de Aduanas SIDUNEA es una herramienta informática que se ha introducido en varios países con el objeto de mejorar el comercio internacional, a través de medidas que incluyen reformas a la práctica administrativa ya existente (2).

Los módulos principales del sistema SIDUNEA incorporan tareas de administración de sistemas, configuración nacional (regulaciones, tarifas y códigos de un país en particular), procesamiento de la declaración aduanera, evaluación de riesgos (selectividad), contabilidad, ingreso de información por parte de interesados, empleo de un documento administrativo y despacho electrónico de bienes (2). Este sistema emplea códigos internacionales y estándares desarrollados por la Organización Internacional para la Estandarización (ISO), la Organización Mundial de Aduanas (OMA) y la Organización de Naciones Unidas (ONU). El sistema SIDUNEA puede ser configurado para adaptarse en los requerimientos operacionales de cualquier administración aduanera, pues su sistema de configuración permite a la aduana definir la información opcional, condicional y obligatoria que considere necesaria (2).

A principios de los años 1990 se introdujo en Cuba la versión 2.51 y a partir de 1996 se actualizó a la versión 2.66. Sin embargo, se comprobó que al no ser propio, suprimía las posibilidades de desarrollo al comercio (2).

El módulo MODSDI (TC) del SIDUNEA, utilizado por los funcionarios de aduana, permite consultar la información sobre las tablas de referencia del sistema como son: países, códigos de embalaje, clasificación, arancelaria, aduanas, acuerdos y empresas. Incluyendo también una serie de reportes que permiten hacer un seguimiento a las declaraciones.

Deficiencias del SIDUNEA (2):

- Elevados costos de capacitación, licencia, mantenimiento, soporte y actualización del sistema.
- No implantación de todos los módulos por incongruencias en su concepción con las características específicas del despacho de mercancías en Cuba. Por lo que se hizo necesario la elaboración de un conjunto de programas o aplicaciones complementarias al mismo que dieran solución a estas especificidades, aunque existen aún, un gran número de registros y operaciones que se realizaban manualmente.
- Existencia en Cuba de la dualidad monetaria en las operaciones comerciales.

Características:

- SGBD Oracle.
- Sistema de escritorio y privativo.
- Lenguaje Java.

## **GESIO**

El Sistema de Gestión de la Información para Ópticas (GESIO) es un software que se creó con el propósito de gestionar la información que tributa al adecuado funcionamiento de las ópticas de Cuba, teniendo control sobre la información relacionada con las solicitudes, los recursos y los reportes que se generan diariamente. Dicho sistema cuenta con el módulo Nomencladores, cuya función es la de gestión de los nomencladores como los tipos de armaduras, cristales, graduaciones y medios básicos, además de la gestión de los municipios, provincias y el color para las armaduras de los espejuelos. Sin embargo, no brinda la posibilidad de crear nuevos nomencladores (5).

Características (5):

- Implementado en Java y Groovy.
- Apache Tomcat como servidor Web.
- Framework de desarrollo Grails.
- Sistema web.
- PostgreSQL v9.1 como gestor de Base de Datos.

## **Sistema para la Gestión de los Nomencladores**

Este sistema se creó con el propósito de sustituir el módulo de Nomencladores del Sistema de Balance y Planificación de Insumos Médicos (alás Bap). Este permite nombrar la información común y poco variable en el tiempo, además de poseer características como la configuración y flexibilidad ante posibles cambios, por lo que permite (6):

- Gestionar los campos de los elementos nombrados.
- Gestionar el código de cada nomenclador.
- Gestionar los grupos de los elementos a nombrar.
- Asociar los campos a los nomencladores.
- Brindar servicios web que permitan el consumo de estos nomencladores por otras aplicaciones.
- Gestionar la información relacionada con cada nomenclador.
- Generar reportes de la información de un nomenclador específico o de varios de ellos.
- Gestionar los usuarios que interactúan con el sistema.

Características (6):

- Desarrollado en Symfony 1.4 y ExtJs 3.1.

- Mapeador de Objetos Relacional Doctrine.
- SGBD PostgreSql 9.1.
- Software libre

## **SADEM**

A partir del estudio realizado al proceso de control de las importaciones y exportaciones comerciales en 1996, se decidió que era necesario y de vital importancia, desarrollar el SADEM (Sistema Automatizado de Despacho Mercantil). Este sistema se realizaría basado en las características del comercio cubano, respondiendo a los intereses de la Aduana cubana (7).

El SADEM se implantó el primero de enero del 2001 en todas las unidades del Sistema de Órganos Aduaneros (SOA), con un mínimo de módulos a los que se les fueron sumando el resto, paulatinamente. Para la explotación del SADEM fue necesario el uso de nomencladores (7).

Este sistema está formado por catorce módulos, uno de ellos es, la TC, que tiene como propósito relacionar las diferentes informaciones de carácter constante que serían usadas en el sistema, organizar estas informaciones a través de nomencladores asignándole un código para facilitar su tratamiento automatizado y así aseguran la consistencia de los datos (7).

Las TC facilitan la flexibilidad del sistema. La estructura y contenido de los mismos estará en correspondencia con las necesidades de análisis y agrupamiento de la información que se requiera durante el procesamiento (7).

El módulo de las TC del SADEM es el encargado de (7):

- Crear, validar y actualizar toda la información referente a las TC del sistema.
- Brindar listados del contenido de las tablas tanto por pantalla como por impresora.
- Facilitar el establecimiento de un esquema de auditoría y control sobre los resultados.
- Controlar la historia de las actualizaciones de cada artículo con el uso de la fecha de inicio y fecha de fin para conocer en qué momento estuvo vigente.
- Posibilitar que los encargados de la actualización de las tablas en las unidades puedan ser cualquier inspector con conocimientos mínimos de computación, al tener un ambiente amigable que facilita la actualización por medio de pantallas de captación de datos.

Deficiencias del SADEM (7):

- Debido al gran volumen de información que manejan en entradas de datos, el uso del teclado es mayoritario ante el ratón, existencia de opciones que aún se encuentran sin automatizar.
- El módulo TC se encontraba implantado sobre el sistema operativo Unix Santa Cruz Operation, por lo que fue necesario su reprogramación, por presentar dificultades en la instalación del mismo en los servidores actuales con los que cuenta la aduana.

- Se decidió desarrollar un sistema automatizado único, que contemplara todos los procesos de la Aduana para implantarse en todas las aduanas del país, este sistema debía responder a los intereses de las aduanas y en el cual todo funcionario tuviera sus requerimientos satisfechos.

Características (7):

- Sistema de escritorio.
- Desarrollado en consola.
- SGBD Oracle.
- Software libre.

## **SUA**

El Sistema Único de Aduana (SUA) tiene como objetivo, automatizar el procesamiento informativo referente a todas las operaciones que conforman los diferentes procesos, ya sea de Medios de Transporte Internacional, Importaciones y Exportaciones con y sin carácter comercial, Bultos Postales y Viajeros y las TC en ambiente WEB, por las facilidades que brinda a los usuarios .SUA es un sistema en el cual todos los módulos validan y controlan las entradas de datos contra los nomencladores y clasificadores (3).

Los principales objetivos del módulo de TC de SUA son:

- Crear, validar y actualizar toda la información referente al control de países, documentos, productos, regímenes, medios de transporte, atraques, muelles, personas, aduanas, entre otras (ver listado de tablas).
- Brindar listados del contenido de las tablas tanto por pantalla como por impresora.
- Facilitar un esquema de auditoría y control sobre los resultados.
- Controlar la historia de las actualizaciones de cada artículo con el uso de la fecha de inicio y fecha de fin para conocer en qué momento estuvo vigente.
- Brindar la posibilidad de que los encargados de la actualización de las tablas, con conocimientos mínimos de computación, tengan un ambiente amigable para realizar la actualización.

Las tablas son actualizadas centralmente en la AGR por los técnicos aduaneros de las diferentes direcciones y además son publicadas en el sitio en Internet con el objetivo de que usuarios no aduaneros se informen acerca del estado de la información que se guarda en dichas tablas (6).

Dificultades que presentaba el módulo de TC del SUA (3):

- Programación estructurada.
- Base de Datos desordenada (no existía una Base de Datos de TC fijada).
- Las TC, la mayoría sueltas sin relación.

- Mal diseño de la interfaz visual.
- Existía un formulario para cada una de las TC existentes.

Características (3):

- SGBD Oracle.
- Lenguaje PHP.
- Software libre.
- Entorno web.

## **GINA**

El sistema GINA es desarrollado debido a los problemas presentados en el SUA, es desarrollado en la UCI en conjunto con el CADI. A diferencia del sistema SUA, este emplea una metodología de desarrollo más robusta, programación orientada a objetos y patrones de diseño. El SUA se dejará de desarrollar por parte de la UCI y de usar por parte de la AGR, para dar paso a GINA que pasaría hacer una versión mejorada del SUA (7).

Esta solución es un sistema Web que se desarrolla sobre el sistema operativo GNU/Linux. Debido a las características que presenta dicho sistema informático, tales como, el lenguaje utilizado, los marcos de trabajo o framework y el gestor de base de datos, posee la gran ventaja de ser multiplataforma, o sea, no depende de un sistema operativo en específico para ser utilizado.

El sistema en cuestión es uno de los pasos más importantes en la actualización y modernización de la AGR, con una configuración y uso más amigable, tecnológicamente más moderno, más robusto y mejor desarrollado. Con un conjunto de subsistemas y módulos para todas las áreas de la aduana en desarrollo, por lo cual sería el primer sistema en cubrir todas las necesidades de la aduana.

En TC los datos son introducidos con posibilidad de ser modificados, es válido aclarar que en el caso de las modificaciones y eliminaciones se realizan de forma lógica actualizando sólo la fecha de vencimiento del artículo (tomado como experiencia del SIDUNEA y que se aplicó en el SUA posteriormente), esto ocurre en todas las tablas ya que por requerimientos del sistema, la información se guarda durante cinco años y para garantizar la consistencia, se realizan las actualizaciones de esta manera.

Con la automatización de las tablas, se logra agilizar el trámite aduanero y optimizar la labor de los funcionarios. Así como disminuir al mínimo los errores que puedan existir por mala manipulación de los códigos o desconocimiento de los mismos. Además se puede conocer los diferentes nomencladores con la información que estos poseen. TC debe garantizar el histórico de los datos tomando como experiencia los Sistemas ya estudiados, permitiendo que los mismos sean auditables.

A continuación se presenta un resumen de las características de los sistemas estudiados.

Tabla 1: Comparación entre los sistemas estudiados

	Gestión de nuevos nomencladores	SO	Licencia	Tecnología en que se desarrolló	SGBD
<b>GIA</b>	No	Window, Linux.	Propietario	Java . Net	Oracle SQLServer
<b>SIDUNEA</b>	No	Windows.	Propietario	Java	Oracle
<b>GESIO</b>	No	Windows, Linux.	Libre	Java Groovy	PostgreSql
<b>Sistema para la Gestión de los Nomencladores</b>	Si	Windows, Linux.	Libre	Symfony v1.4 ExtJs v3.1	PostgreSql
<b>SADEM</b>	No	Windows, Linux.	Libre	Consola	Oracle
<b>SUA</b>	No	Windows, Linux.	Libre	Php ExtJs	Oracle
<b>GINA</b>	No	Windows, Linux.	Libre	Symfony v1.2.8 ExtJs v3.0	Oracle

Al concluir el estudio se comprobó que los sistemas citados con anterioridad no pueden ser utilizados pues, con excepción del Sistema para la Gestión de los Nomencladores, no permiten la gestión de nuevos nomencladores a través de una interfaz única. De igual manera estos sistemas no pueden ser integrados al GINA, debido a que las herramientas y tecnologías con las que fueron desarrollados no son compatibles con las empleadas para la realización de este sistema. El Sistema para la Gestión de Nomencladores, por su parte, a pesar de utilizar las herramientas con las que fue desarrollado el GINA, no garantiza que se respete la estructura que deben poseer las TC. A pesar de esto, dichos sistemas permitieron la reutilización del diseño de sus interfaces y como realizan el proceso de gestión de los nuevos nomencladores. Teniendo en cuenta lo

anterior se propone realizar una nueva versión del módulo de TC en el cual se le dé solución al problema a resolver.

### 1.3 Lenguajes, metodología y herramientas.

El módulo TC forma parte del sistema GINA, el cual tiene ya definido la tecnología que se va a usar para su desarrollo, por ello, la metodología, el lenguaje y las herramientas a usar se encuentran sujetas a las políticas especificadas por el proyecto. A continuación se realizará una fundamentación de las mismas.

#### Metodología

La metodología de software surgen para guiar a las personas implicadas en este proceso, brindando un conjunto de técnicas, procedimientos y herramientas, de manera que sepan que hacer en cada momento y como alcanzar un producto de alta calidad.

Para el desarrollo de esta solución se aplicó la variación realizada por la UCI a la metodología del Proceso Unificado Ágil de Scott Ambler o Agile Inified Process (AUP), la cual se adapta al ciclo de vida productivo de la UCI. Esta variación cuenta con tres fases: Inicio, Ejecución y Cierre, además de ocho disciplinas: Modelado de Negocio, Requisitos, Análisis y Diseño, Implementación, Pruebas Internas, Pruebas de Liberación, Pruebas de Aceptación y Despliegue (8).

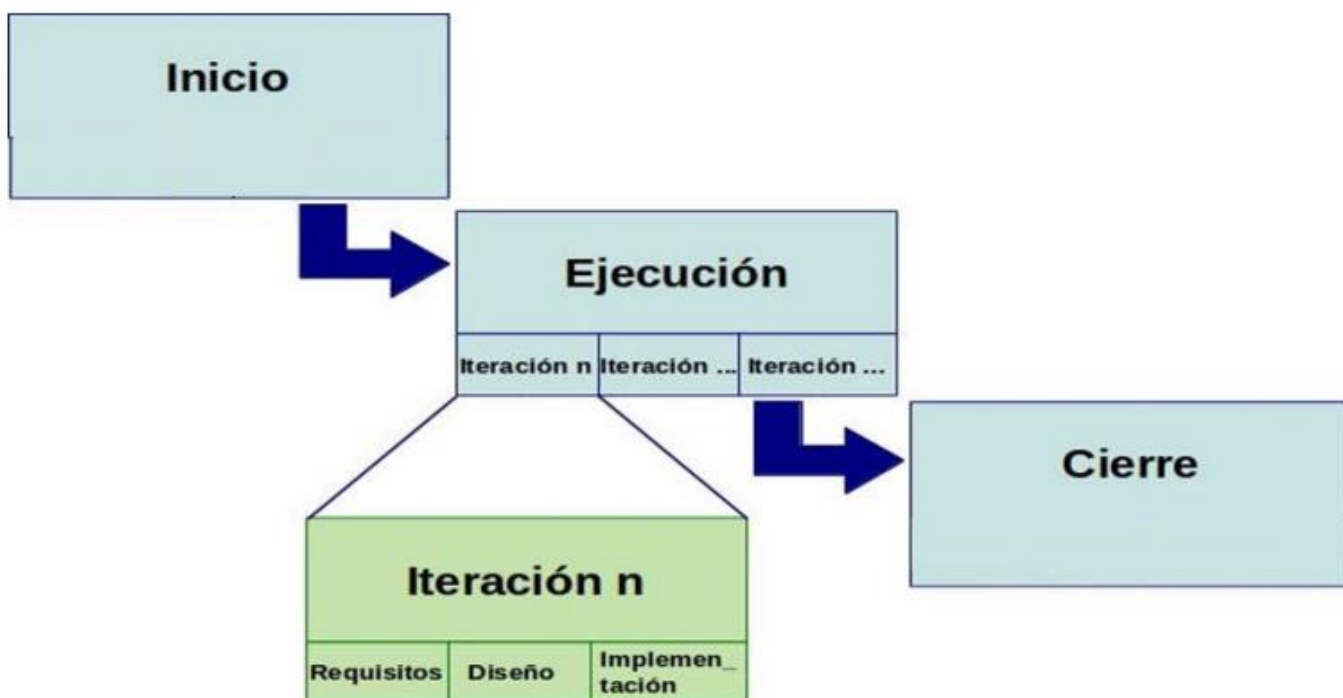


Figura 1 Fases e Iteraciones

## Lenguajes

### PHP 5.3

PHP es el acrónimo de Hypertext Preprocessor, es un lenguaje interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor pues funciona en un servidor remoto que procesa la página Web antes de que sea abierta por el navegador del usuario. PHP ha sido especialmente creado para el desarrollo de páginas Web dinámicas y puede ser incluido con facilidad dentro del código HTML. Ha alcanzado gran popularidad y existe una amplia comunidad de desarrolladores y programadores que continuamente implementan mejoras en su código. Se caracteriza por una sencilla integración con múltiples bases de datos y, entre las que se encuentran MySQL, PostgreSQL, Oracle entre otras. Está dotado de un gran número de funciones predefinidas que simplifican enormemente tareas habituales como descargar documentos, generar imágenes GIF, enviar correos electrónicos, trabajar con cookies y sesiones, establecer conexiones a otros servicios de red y generar documentos PDF. La sintaxis de PHP se basa en otros lenguajes de programación, principalmente en C y Perl, o un lenguaje de tipo C como C++ o Java, se distingue por su facilidad de aprendizaje y uso. Entre los competidores principales de PHP se puede citar a Perl, Microsoft Active Server Pages (ASP), Java Server Pages (JSP) y Allaire ColdFusion. PHP también soporta el uso de servicios que usen protocolos como IMAP, SNMP, NNTP, POP3, HTTP y derivados; además de poder interactuar con otros protocolos (9).

PHP es un lenguaje multiplataforma, permite leer y manipular datos desde diversas fuentes, incluyendo datos que pueden ser ingresados por los usuarios desde formularios HTML, es libre, por lo que es una alternativa de fácil acceso para todos. Es orientado a objetos y posee una arquitectura extensible. Dentro de las opciones que brinda PHP se encuentra: la simplicidad, la estabilidad y la compatibilidad (9).

### JavaScript

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. Estas páginas son aquellas que incorporan efectos como textos que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario. Es un lenguaje que no requiere compilación, lo que se conoce como lenguaje de programación interpretado. Su sintaxis es muy parecida a la del lenguaje Java y el lenguaje C. Hoy por hoy, todos los navegadores interpretan el código JavaScript integrado dentro de las páginas Web. Por tradición, el lenguaje JavaScript se utilizaba para realizar tareas y operaciones en el marco de la aplicación únicamente del lado del cliente, sin acceso a funciones del servidor (10).

Actualmente, existen aplicaciones en JavaScript para el servidor. Aunque JavaScript de cliente y de servidor comparten el mismo conjunto base de funciones y características, en algunos casos se



utilizan de distinta forma. De manera general, JavaScript permite crear aplicaciones específicamente orientadas a su funcionamiento en la red Internet. Usando JavaScript se pueden crear páginas HTML dinámicas que procesen la entrada del usuario y que sean capaces de gestionar datos persistentes usando objetos especiales, archivos y bases de datos relacionales. Además, se pueden construir aplicaciones que varían desde la gestión de la información corporativa interna y su publicación en Intranets hasta la gestión masiva de transacciones de comercio electrónico (10).

## **CSS**

El principio de las hojas de estilo consiste en la utilización de un solo documento para almacenar las características de presentación de las páginas asociadas a grupos de elementos. Esto implica nombrar un conjunto de definiciones y características de presentación de las páginas, y activar esos nombres para aplicarlos a una parte del texto. Las hojas de estilo se desarrollaron para compensar los defectos de HTML con respecto a la presentación y al diseño de las páginas (11).

HTML tiene varias etiquetas para modificar la presentación y definir los estilos del texto, pero cada elemento tiene su propio estilo, independientemente de los elementos que lo rodean. Al utilizar hojas de estilo, cuando se necesite cambiar la apariencia de un sitio que tiene cientos de páginas Web todo lo que hay que hacer es editar las definiciones de la hoja de estilo en un solo lugar para cambiar la apariencia del sitio completo. Se denominan "hojas de estilo en cascada" porque se pueden definir múltiples hojas y los estilos pueden aplicarse a todas las páginas (con un sistema predefinido para resolver conflictos) (11).

Las hojas de estilo pueden utilizarse para (11):

- Lograr una apariencia uniforme de todo el sitio al activar una sola definición de estilo en cada página.
- Cambiar un aspecto en todo el sitio Web con tan solo editar unas pocas líneas.
- Facilitar la lectura de los códigos HTML ya que los estilos se definen por separado.
- Permitir que las páginas se carguen más rápido ya que hay menos cantidad de código HTML en cada página.
- Posicionar los elementos de la página de una manera más uniforme.

## **UML 2.0**

El Lenguaje Unificado de Desarrollo (UML) representa un número de modelos de desarrollo basados en componentes que han sido propuestos en la industria. El proceso unificado define los componentes que se utilizarán para construir el sistema y las interfaces que conectarán los componentes. Utilizando una combinación del desarrollo incremental e iterativo, el proceso unificado define la función del sistema aplicando un enfoque basado en escenarios (desde el punto de vista

del usuario) y acopla la función con un marco de trabajo arquitectónico que identifica la forma que tomará el software (12).

## **JSON**

JSON<sup>3</sup> es un formato ligero de intercambio de datos, un subconjunto de la notación literal de objetos de JavaScript que no requiere el uso de XML. Es un formato de texto que es completamente independiente del lenguaje, estas propiedades hacen que JSON sea un lenguaje ideal para el intercambio de datos (13).

## **Marco de Trabajo**

### **Symfony 1.2**

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 (14).

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, etc.) como en plataformas Windows (14).

Características y Ventajas (14):

- Fácil de instalar y configurar en la mayoría de las plataformas, 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 los casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos. 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 adaptables a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo.

---

<sup>3</sup> JavaScript ObjectNotation

- 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 bibliotecas desarrolladas por terceros.

### **ExtJs 3.0**

Ext JS es una biblioteca JavaScript ligera y de alto rendimiento, que nos permite crear páginas e interfaces web dinámicas. Provee interfaces gráficas de usuario que brindan experiencias parecidas o iguales a las que se tienen con aplicaciones de escritorio. Es extensible para la gran mayoría de los navegadores, evitando el tedioso problema de validar el código para cada uno de estos. Entre sus principales ventajas se encuentra el balance entre Cliente-Servidor, la carga de procesamiento se distribuye permitiendo que el servidor, al tener menor carga, pueda manejar los clientes de manera más eficiente. La comunicación asíncrona permite el intercambio de información con el servidor sin necesidad de estar sujeta a una acción del usuario, dando la libertad de cargar la información sin que este lo note (15) (16).

### **Propel**

Propel es un servicio de objeto persistente y de consulta, lo que significa que Propel provee un sistema para almacenar objetos en una base de datos y un sistema para la búsqueda y restauración de objetos desde una base de datos. Propel le permite realizar consultas complejas y manipulación de bases de datos sin escribir una sola cláusula SQL. Propel hace más fácil la escritura de aplicaciones, más fácil de desplegar, y mucho más fácil para migrar si alguna vez la situación lo amerita (17).

Propel puede ser descrito como un mapeado objeto-relacional, una capa DAO, o una capa objeto persistente. Propel es un puerto de Apache torque basado en acercamientos probados, desarrollado por el proyecto Torque y optimizado para PHP. Propel espera proporcionar un inteligente y comprensivo servicio de manejo de datos con un mínimo costo de realización para su aplicación en PHP (17).

Sin embargo, Propel también genera las clases para cada tabla que exhibe algunas de las propiedades de la tabla del patrón datos de entrada (17):

- Una tabla de entrada de datos almacena todo el SQL para acceder a una sola tabla o vista: selecciones, inserciones, actualizaciones, y eliminaciones. Otro código llama los métodos para todas las interacciones con la base de datos.
- En Propel las clases de tabla de entrada de datos son llamadas clasesPeer, mientras las clases de filas de entrada de datos son llamadas entidad o clases objeto.

Como una aplicación, Propel tiene dos componentes principales (y ahora formalmente separados) (17):

- Un motor generador para construir sus clases y archivos SQL (generador-propel).
- Un ambiente de ejecución que proporciona herramientas para construir consultas SQL, ejecutando consultas compiladas, y herramientas para el manejo de conexiones para múltiples bases de datos simultáneamente (propel).

El ambiente de ejecución proporciona una capa de abstracciones y encapsulación de bases de datos, reglas y lógicas de negocios. Las clases Propel representan la capa modelo del tradicional MVC, diseñado para encapsular cualquier nivel de validación de dato necesitado por su aplicación (17).

## **Herramienta de modelado**

### **Visual Paradigm 8**

Visual Paradigm es una herramienta CASE que utiliza como lenguajes, UML para el modelado del sistema y la notación BPMN para el negocio. Soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue (18).

El lenguaje de modelado UML ayuda a una construcción más rápida de aplicaciones de calidad, y a un menor costo. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación (18).

Principales características (18):

- Disponibilidad en múltiples plataformas (Windows, Linux).
- Diseño centrado en casos de uso y enfocado al negocio que generan un software de mayor calidad.
- Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.
- Soporta aplicaciones web.
- Fácil de instalar y actualizar.
- Generación de bases de datos.

## **Herramienta para el control de versiones**

### **Subversion**

Subversion es un sistema de control de versiones libre y de código fuente abierto. Es decir, Subversion maneja ficheros y directorios a través del tiempo. Hay un árbol de ficheros en un repositorio central. El repositorio es como un servidor de ficheros ordinario, excepto porque recuerda todos los cambios hechos a sus ficheros y directorios. Esto le permite recuperar versiones antiguas de sus datos, o examinar el historial de cambios de los mismos. En este aspecto, mucha gente piensa en los sistemas de versiones como en una especie de “máquina del tiempo” (19).

Subversion puede acceder al repositorio a través de redes, lo que le permite ser usado por personas que se encuentran en distintos ordenadores. A cierto nivel, la capacidad para que varias personas puedan modificar y administrar el mismo conjunto de datos desde sus respectivas ubicaciones fomenta la colaboración. Se puede progresar más rápidamente sin un único conducto por el cual deban pasar todas las modificaciones. Y puesto que el trabajo se encuentra bajo el control de versiones, no hay razón para temer porque la calidad del mismo vaya a verse afectada por la pérdida de ese conducto único—si se ha hecho un cambio incorrecto a los datos, simplemente deshaga ese cambio (19).

Algunos sistemas de control de versiones son también sistemas de administración de configuración de software. Estos sistemas son diseñados específicamente para la administración de árboles de código fuente, y tienen muchas características que son específicas del desarrollo de software—tales como el entendimiento nativo de lenguajes de programación, o el suministro de herramientas para la construcción de software. Sin embargo, Subversion no es uno de estos sistemas. Subversion es un sistema general que puede ser usado para administrar cualquier conjunto de ficheros. Para usted, esos ficheros pueden ser código fuente—para otros, cualquier cosa desde la lista de la compra de comestibles hasta combinaciones de vídeo digital y más allá (19).

### **Características (19):**

- **Versionado de directorios:** Implementa un sistema de ficheros versionado “virtual” que sigue los cambios sobre árboles de directorios completos a través del tiempo. Ambos, ficheros y directorios, se encuentran bajo el control de versiones.
- **Verdadero historial de versiones:** Permite añadir, borrar, copiar, y renombrar ficheros y directorios. Y cada fichero nuevo añadido comienza con un historial nuevo, limpio y completamente suyo.
- **Envíos atómicos:** Una colección cualquiera de modificaciones o bien entra por completo al repositorio, o bien no lo hace en absoluto. Esto permite a los desarrolladores construir y enviar los cambios como fragmentos lógicos e impide que ocurran problemas cuando sólo una parte de los cambios enviados lo hace con éxito.

- **Versionado de metadatos:** Cada fichero y directorio tiene un conjunto de propiedades — claves y sus valores —asociado a él. Usted puede crear y almacenar cualquier par arbitrario de clave/valor que desee. Las propiedades son versionadas a través del tiempo, al igual que el contenido de los ficheros.
- **Elección de las capas de red:** Subversion tiene una noción abstracta del acceso al repositorio, facilitando a las personas implementar nuevos mecanismos de red. Subversion puede conectarse al servidor HTTP Apache como un módulo de extensión. Esto proporciona a Subversion una gran ventaja en estabilidad e interoperabilidad, y acceso instantáneo a las características existentes que ofrece este servidor—autenticación, autorización, compresión de la conexión, etcétera. También tiene disponible un servidor de Subversion independiente, y más ligero. Este servidor utiliza un protocolo propio, el cual puede ser encaminado fácilmente a través de un túnel SSH.
- **Manipulación consistente de datos:** Subversion expresa las diferencias del fichero usando un algoritmo de diferenciación binario, que funciona idénticamente con ficheros de texto (legibles para humanos) y ficheros binarios (ilegibles para humanos). Ambos tipos de ficheros son almacenados igualmente comprimidos en el repositorio, y las diferencias son transmitidas en ambas direcciones a través de la red.
- **Ramificación y etiquetado eficientes:** El coste de ramificación y etiquetado no necesita ser proporcional al tamaño del proyecto. Subversion crea ramas y etiquetas simplemente copiando el proyecto, usando un mecanismo similar al enlace duro. De este modo estas operaciones toman solamente una cantidad de tiempo pequeña y constante.
- **Hackability:** Subversion no tiene un equipaje histórico; está implementado como una colección de bibliotecas compartidas en C con APIs bien definidas. Esto hace a Subversion extremadamente fácil de mantener y reutilizable por otras aplicaciones y lenguajes.

## **Herramienta de desarrollo**

### **Netbeans IDE 7.4**

NetBeans IDE (Entorno de Desarrollo Integrado) 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. NetBeans IDE es un producto libre y gratuito sin restricciones de uso, contiene las herramientas para que los desarrolladores de software puedan crear aplicaciones desktop, enterprise, web, y aplicaciones móviles, con el lenguaje Java, así como también C/C++, PHP, JavaScript, Groovy, y Ruby (20).

NetBeans permite crear aplicaciones Web con PHP 5, un potente debugger integrado y además viene con soporte para Symfony. Tiene también soporte para AJAX. NetBeans IDE funciona en sistemas operativos compatibles con la máquina virtual Java: Window, MacOs, Linux (20).

## **Servidor de aplicaciones**

### **Apache 2.2**

Apache es un servidor web de código libre para la transferencia de hipertextos (Hypertext Transfer Protocol, HTTP por sus siglas en inglés) para plataformas Unix, Windows y Macintosh. Su implementación se realiza de forma colaborativa, con prestaciones y funcionalidades equivalentes a las de los servidores comerciales. El proyecto está dirigido y controlado por un grupo de voluntarios de todo el mundo que, usando Internet y la web para comunicarse, planifican y desarrollan el servidor y la documentación relacionada (21).

Principales características (21):

- Es un servidor de web conforme al protocolo HTTP/1.1.
- Soporta tanto host basados en IP como host virtuales.
- Apache soporta autenticación básica basada en la Web.
- Modular: puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo que proporciona, y con la API de programación de módulos, para el desarrollo de módulos específicos.
- Extensible: gracias a ser modular se han desarrollado diversas extensiones entre las que destaca PHP, un lenguaje de programación del lado del servidor.
- Personalización de las respuestas ante los posibles errores que se puedan dar en el servidor.

## **Sistemas Gestores de Base de Datos (SGDB)**

### **Oracle 11g**

Oracle Database 11g Enterprise Edition ofrece confiabilidad, escalabilidad y desempeño de primer nivel para configuraciones en cluster y en un solo servidor. Ofrece las más completas características para soportar el procesamiento de transacciones más exigente, inteligencia de negocios, y aplicaciones para la administración de contenido. Protección ante las fallas del servidor, fallas del sitio, errores humanos y reducción del tiempo de baja programado. Protección de datos con seguridad única en el nivel de filas, auditorías detalladas, y encriptación transparente de datos. Incluye data warehousing de alto desempeño, procesamiento analítico online, y características de extracción de datos. Constituye un sistema gestor de bases de datos con características objeto-relacionales (22).

Sus principales características son las siguientes (22):

- Entorno cliente/servidor.
- Gestión de grandes bases de datos.
- Usuarios concurrentes.
- Alto rendimiento en transacciones.
- Sistemas de alta disponibilidad.
- Disponibilidad controlada de los datos de las aplicaciones.
- Adaptación a estándares de la industria, como SQL-92.
- Gestión de la seguridad.
- Autogestión de la integridad de los datos.
- Opción distribuida.
- Portabilidad.
- Compatibilidad.
- Replicación de entornos.

Provee un control de accesos discrecional, es decir, acceso restringido a la información basado en privilegios (22).

Gestiona la seguridad de la base de datos usando (22):

- Usuarios y esquemas de la base de datos.
- Privilegios.
- Roles.
- Ajustes de rendimiento y cuotas.
- Límites sobre los recursos.
- Auditoría.

Cada usuario posee un dominio de seguridad, que determina (22):

- Acciones (privilegios y roles) disponibles para el usuario.
- Cuotas sobre tablespaces.
- Límites en los recursos del sistema.

Posee varias estructuras y mecanismos de software para proveer (22):

- Recuperación de la base de datos ante distintos tipos de fallos.
- Operaciones de recuperación flexibles.
- Disponibilidad de los datos durante las operaciones de backup y recovery.

Utiliza varias estructuras para proveer la recuperación completa de la instancia (22):

- Redo Log.
- Segmentos de rollback.



- Fichero de control.
- Copias necesarias de la base de datos.

#### **1.4 Conclusiones del capítulo**

En este capítulo se realizó un estudio de los sistemas aduaneros o de los que realizan la gestión de los nomencladores, utilizados a nivel mundial y en Cuba, concluyendo:

- Que ninguno de los existentes puede integrarse al sistema GINA para lograr una mejor gestión de los nomencladores.
- Se estableció que el desarrollo de la solución será guiado por la metodología AUP en su variación realizada por la UCI.
- Como herramienta de modelado, se utilizará el Visual Paradigm.
- Se determinó utilizar el framework Ext JS en su versión 3.0 para el trabajo en el lado del cliente, Symfony en su versión 1.2.8 como framework de trabajo en la lado del servidor y el SGBD Oracle en su versión 11g.

## CAPÍTULO 2: ANÁLISIS Y DISEÑO DE LA PROPUESTA DE SOLUCIÓN

### **2.1 Introducción.**

En este capítulo se expone la solución propuesta, para ello, primeramente se realiza la fase de análisis teniendo en cuenta los conceptos significativos relacionados en el modelo conceptual. Posteriormente se define la forma en que serán capturados los requisitos, su descripción y validación, lo que da paso al diseño. En el cual se plantean los patrones que se utilizan para su realización y una vez terminado se aplican métricas para evaluar su validez.

### **2.2 Propuesta de solución.**

Para darle solución al problema planteado, se propone el desarrollo de una nueva versión del módulo de TC que logre una mejora en la gestión de estas, a través de una interfaz única y de forma dinámica. Esta nueva versión permitirá a los técnicos aduaneros introducir en el sistema nuevas TC, modificarlas o eliminarlas en dependencia de cual sea su necesidad.

### **2.3 Modelo Conceptual.**

Un modelo conceptual explica los conceptos significativos en un dominio del problema real, siendo este uno de los productos de trabajo más importantes generados en la fase del análisis (23). El modelo conceptual que se presenta a continuación tiene como objetivo, describir los conceptos relacionados con el módulo de TC, representando así las principales asociaciones y atributos. Este documento se encuentra en el repositorio de la Aduana (CIG-ADU-N-CFGa-i1301 Modelo Conceptual. doc).

Campo: Son los campos que posee una TC

TC: Representa la estructura que deben tener todas las TC.

Aplicación: Define que aplicación estará utilizando cada una de las TC.

Usuario Sistema: Es quien tiene los permisos para gestionar las TC del sistema.

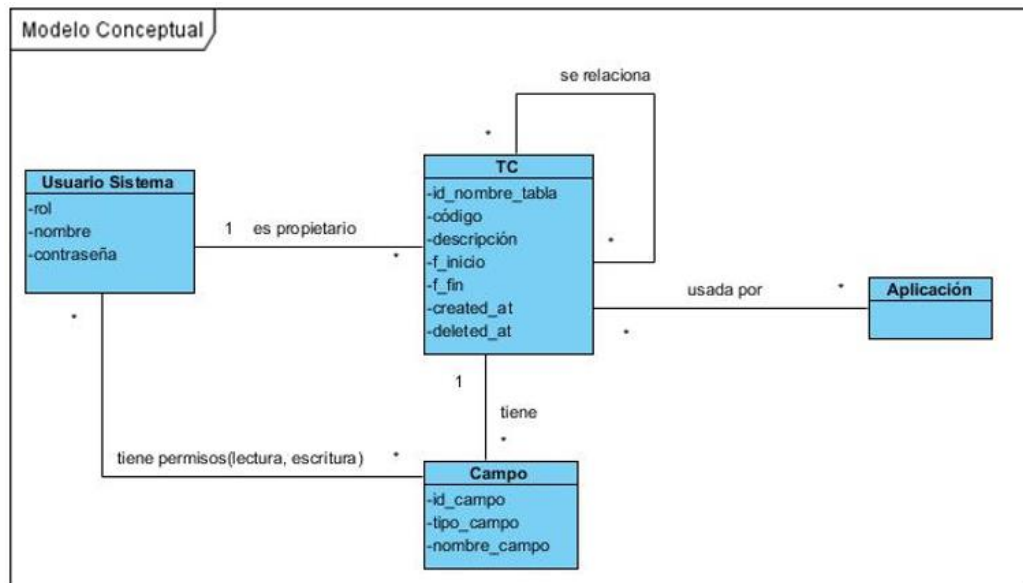


Figura 2 Modelo Conceptual

Además, este modelo tiene una estrecha relación con los requisitos funcionales debido a que estos dependen de las restricciones de negocio que pueden reflejarse a través de las relaciones y los conceptos que se representan en el mismo. Es decir, estos conceptos poseen relaciones entre ellos que tributan a que las funcionalidades reflejadas en los requisitos, se comporten de esa manera teniendo en cuenta los atributos que se manejan en el negocio.

## 2.4 Técnicas para la Captura de Requisitos

Siguiendo el procedimiento para la Ingeniería de Requisitos en el Departamento de Desarrollo de Soluciones para la Aduana del centro CEIGE y teniendo en cuenta las características del cliente, se decidió utilizar como técnicas que permitirán a los analistas la recopilación y obtención de la información necesaria para la captura de requisitos, la entrevista y la observación.

Se entrevistó a la técnica aduanera encargada del trabajo con el módulo de TC, en visitas realizadas a las instalaciones, ya que las entrevistas permiten un intercambio más abierto, mediante preguntas que esclarecen con precisión el funcionamiento de todo el trabajo con el módulo de TC. La observación se llevó a cabo para percibir cómo se desarrollan las operaciones de registro, modificación o eliminación de una TC en el sistema, pudiendo captar directamente las particularidades de estos procesos en las visitas realizadas.

## 2.5 Requisitos

Un requisito es una descripción de necesidades o aspiraciones respecto a un producto. El objetivo principal de la actividad de definición de requisitos, consiste en identificar qué es lo que realmente se necesita. Esto se realiza de un modo que sirva de comunicación entre el cliente y el equipo de desarrollo. Los requisitos se pueden clasificar en: funcionales y no funcionales. Los primeros son capacidades o condiciones que el sistema debe cumplir; y los segundos, son propiedades o cualidades que el producto debe tener (23).

### 2.5.1 Requisitos funcionales

Como resultado de la aplicación de las técnicas de captura de requisitos descritas, se obtuvieron 4 requisitos funcionales los cuales se especifican en la tabla 2; de estos, 3 poseen una complejidad alta, esto se debe a que en el momento de registrar, modificar o eliminar se realizan cambios en la estructura de la base de datos, luego se agregan, modifican o eliminan registros en algunas tablas y por último se debe agregar, modificar o eliminar algunos archivos.

Tabla 2 Especificación de requisitos

Nº	Nombre	Descripción	Prioridad para el cliente	Complejidad
RF 1	Registrar Tabla de Control	Permite registrar en el sistema los datos de una Tabla de Control	<i>Alta</i>	<i>Alta</i>
RF 2	Modificar Tabla de Control	Permite al técnico aduanero modificar la Tabla de Control	<i>Alta</i>	<i>Alta</i>
RF 3	Listar Tablas de Control	Permite listar las Tablas de Control que se encuentran registradas	<i>Alta</i>	<i>Media</i>
RF 4	Eliminar Tabla de Control	Permite eliminar una Tabla de Control	<i>Alta</i>	<i>Alta</i>

## **2.5.2 Requisitos no funcionales**

### **Usabilidad**

Deben contar con un menú que les permita a los usuarios acceder a las principales funciones que son de su interés. La resolución de la página se adaptará a la pantalla del cliente. Podrá ser usado por cualquier persona que acceda a él, que tenga algún conocimiento básico de computación y trabajo en la Web.

### **Soporte**

Las aplicaciones clientes deben ser capaz de correr sobre cualquier plataforma, para el caso de Windows se recomienda XP por la experiencia acumulada por los usuarios. Para la parte servidora se recomienda que arranque sobre plataforma Linux. Ser programado en PHP 5. 3 y con un SGBD Oracle 11g.

### **Portabilidad**

Multiplataforma. El sistema se podrá montar sobre Unix, Linux, Windows. Así mismo podrá usar una serie de SGBD, como PostgreSQL, MySQL, Oracle, aunque preferiblemente se desea la portabilidad sobre software libre.

### **Hardware**

#### **Cliente:**

Las aplicaciones son desarrolladas para que las PC clientes de los usuarios puedan utilizarlas haciendo uso de la menor cantidad de requisitos de hardware posible.

- Procesador Intel Pentium III de 1. 4GHz de velocidad de procesamiento y 512 Mb de memoria RAM y 10Gb libres de disco duro.
- Tarjeta de red 10/100Mbps.

#### **Servidor:**

- Procesador Intel Core 2 Duo a 2. 6 GHz de velocidad de procesamiento y 2Gb de memoria RAM.
- 60Gb de espacio libre en disco.
- Tarjeta de red 10/100Mbps.

### 2.5.3 Descripción de los requisitos del sistema

En la tabla 3 se muestra la descripción del requisito Registrar TC.

Tabla 3 Descripción del requisito "Registrar TC"

<b>Precondiciones</b>	El Especialista se ha autenticado en el sistema. El Especialista tiene permisos para registrar una TC.
<b>Flujo de eventos</b>	
<b>Flujo básico Registrar una TC</b>	
1	Selecciona el menú "Gestionar Tablas".
2	El sistema muestra las opciones a realizar.
3	Accede a la opción "Registrar Tabla de Control".
4	El sistema muestra el prototipo de interfaz "Crear Tabla" que se muestra en la figura 1.
5	Introduce los datos requeridos para adicionar una TC y oprime el botón "Guardar".  En caso que no introduzca todos los datos y seleccione la opción "Guardar" ver <u>Flujo alternativo "5a. Campos vacíos"</u> .
6	El sistema valida los datos introducidos.  En caso de datos incorrectos ver <u>Flujo alternativo "6a. Datos incorrectos"</u> .
7	El sistema valida si la TC no está registrada en el sistema. En caso que la TC esté registrada en el sistema ver <u>Flujo alternativo "7a. TC Registrada"</u> .
8	Guarda todos los datos de la nueva TC en el sistema.
9	El sistema muestra el prototipo de interfaz "Agregar restricción" que se muestra en la figura 2.
10	Selecciona los datos requeridos para adicionar una restricción y oprime el botón "Guardar". En caso que no seleccione todos los datos y seleccione la opción "Guardar" ver <u>Flujo alternativo "10a. Campos vacíos"</u> .
11	El sistema valida los datos introducidos. En caso de datos incorrectos ver <u>Flujo alternativo 11a. Tipo datos"</u> .

12	Guarda todos los datos de la nueva restricción en el sistema.
13	Termina el requisito.
<b>Pos-condiciones</b>	
1	Se registró en el sistema un cobro de documentos.
<b>Flujos alternativos</b>	
<b>Flujo alternativo 5. a Campos vacíos</b>	
1	Muestra el mensaje “Los campos obligatorios no pueden estar vacíos”.
2	Se ejecuta el paso 5 del Flujo básico Registrar una TC
<b>Pos-condiciones</b>	
1	Se muestra el mensaje de error
<b>Flujo alternativo 6. a Datos incorrectos</b>	
1	Muestra el mensaje “Revise que los campos no contengan caracteres extraños”.
2	Se ejecuta el paso 5 del Flujo básico Registrar una TC
<b>Pos-condiciones</b>	
1	Se muestran el mensaje de error.
<b>Flujo alternativo 8. a TC registrada</b>	
1	Muestra el mensaje “La Tabla de Control ya se encuentra registrada”.
2	Se ejecuta el paso 5 del Flujo básico Registrar una TC.
<b>Pos-condiciones</b>	
1	Se muestran el mensaje de error.
<b>Flujo alternativo 10. a Campos vacíos</b>	
1	Muestra el mensaje “Los campos obligatorios no pueden estar vacíos”.
2	Se ejecuta el paso 10 del Flujo básico Registrar una TC.
<b>Pos-condiciones</b>	
1	Se muestran el mensaje de error.
<b>Flujo alternativo 11. a Tipo dato</b>	
1	Muestra el mensaje “Los tipos datos de las columnas no son iguales”.
2	Se ejecuta el paso 10 del Flujo básico Registrar una TC.
<b>Pos-condiciones</b>	
2	Se muestran el mensaje de error.
<b>Validaciones</b>	
1	

<b>Relaciones</b>	<b>Requisitos Incluidos</b>	N/A
	<b>Extensiones</b>	N/A
<b>Concepto</b>		N/A
<b>Requisitos especiales</b>	N/A	
<b>Asuntos pendientes</b>	N/A	

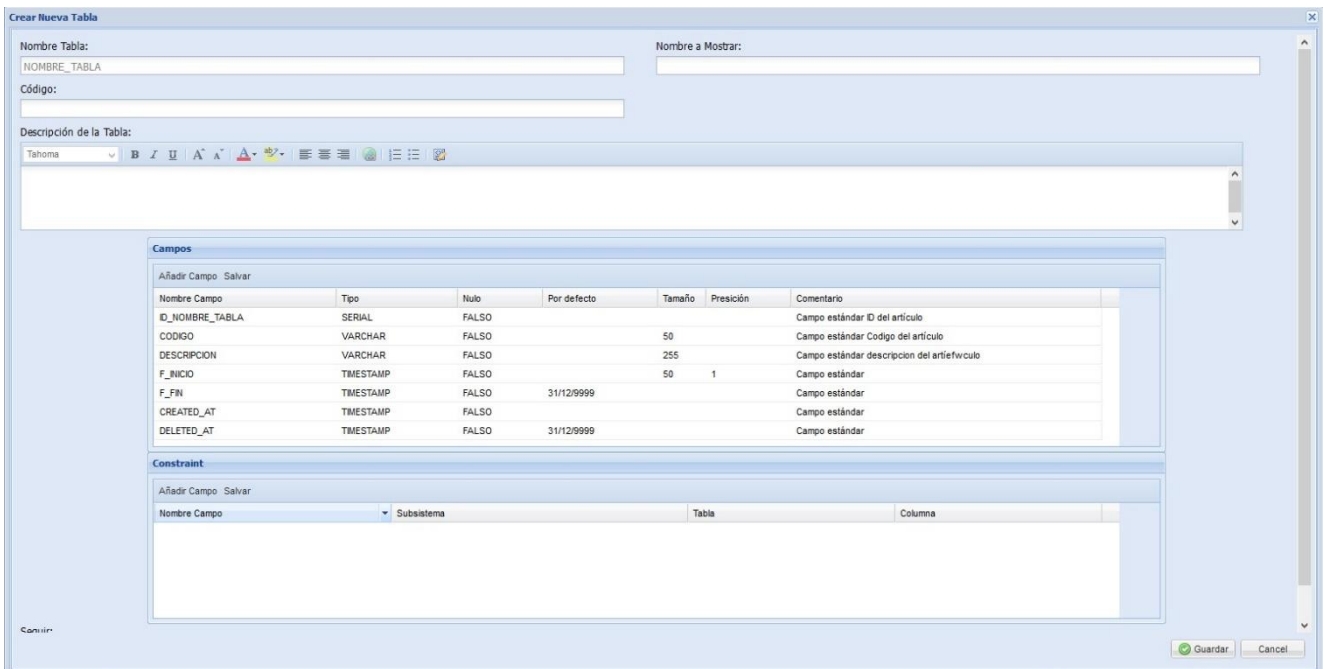


Figura 3 Prototipo Crear Tabla

En la tabla 4 se muestra la descripción del requisito Modificar TC.

Tabla 4 Descripción del requisito "Modificar TC"

<b>Precondiciones</b>	El Especialista se ha autenticado en el sistema. El Especialista tiene permisos para modificar una TC.
<b>Flujo de eventos</b>	
<b>Flujo básico Modificar TC</b>	



1	Selecciona una de las tablas que se muestra en el grid “Tablas”.
2	Muestra la estructura referente a esa tabla.
3	Se selecciona una de las columnas de la tabla que se muestran en el grid “Columnas”
4	Se selecciona el menú “Gestionar Estructura de Tabla”
5	Muestra las modificaciones que se pueden realizar
6	Selecciona la modificación que desea realizar En caso de no haber seleccionado uno de los elementos del grid “Columnas” ver flujo alternativo <u>“6a. Selección de columnas”</u>
7	Muestra el prototipo de interfaz perteneciente a la opción seleccionada
8	Introduce los nuevos valores que va a tomar la columna seleccionada En caso que no introduzca los datos y seleccione la opción “Modificar” ver <u>Flujo alternativo “8a. Campos vacíos”</u> .
9	Valida los datos introducidos. En caso de datos incorrectos ver <u>Flujo alternativo “9a. Datos incorrectos”</u> .
10	Modifica la columna seleccionada
11	Termina el requisito.

#### **Pos-condiciones**

1	Se registra la TC en el sistema
---	---------------------------------

#### **Flujos alternativos**

##### **Flujo alternativo 6a. Selección de columnas**

1	Muestra el mensaje “Debe seleccionar la columna que desea modificar”.
2	Se ejecuta el paso 3 del Flujo básico Modificar una TC

#### **Pos-condiciones**

1	Mostrado el mensaje de error.
---	-------------------------------

##### **Flujo alternativo 8a. Campos Vacíos**

1	Muestra el mensaje “Los campos obligatorios no pueden estar vacíos”.
2	Se ejecuta el paso 8 del Flujo básico Modificar una TC.

#### **Pos-condiciones**

1	Mostrado mensaje de error.
---	----------------------------

##### **Flujo alternativo 9a. Datos incorrectos**

1	Muestra el mensaje “Los datos ingresados no son correctos”.
2	Se ejecuta el paso 8 del Flujo básico Modificar una TC.

<b>Pos-condiciones</b>		
1	Mostrado mensaje de error.	
<b>Validaciones</b>		
1	N/A	
<b>Relaciones</b>	<b>Requisitos</b>	N/A
	<b>Incluidos</b>	
	<b>Extensiones</b>	N/A
<b>Conceptos</b>	N/A	
<b>Requisitos especiales</b>	N/A	
<b>Asuntos pendientes</b>	N/A	

En la tabla 5 se muestra la descripción del requisito Eliminar TC.

*Tabla 5 Descripción del requisito "Eliminar TC"*

<b>Precondiciones</b>	El Especialista se ha autenticado en el sistema. El Especialista tiene permisos para eliminar una TC.
<b>Flujo de eventos</b>	
<b>Flujo básico Cobrar cheque devuelto</b>	
1	Selecciona una de las tablas que se muestra en el grid "Tablas".
2	Se selecciona el menú "Gestionar Tablas"
3	El sistema muestra las opciones a realizar.
4	Accede a la opción "Eliminar Tabla de Control". En caso de no haber seleccionado uno de los elementos del grid "Tablas" ver flujo alternativo " <u>4a. Selección de tablas</u> "
5	Marca las opciones a realizar y da click en el botón "Eliminar" En caso de no haber seleccionado ninguna de las opciones ver flujo alternativo " <u>5a. Selección de opciones</u> "
6	El sistema elimina la tabla
7	El sistema muestra un mensaje confirmando que la tabla fue eliminada
8	Concluye el requisito.

<b>Pos-condiciones</b>		
1	Se eliminó la tabla.	
<b>Flujos alternativos</b>		
<b>Flujo alternativo 4. a Selección de tablas</b>		
1	El sistema muestra el mensaje “Debe seleccionar una tabla”.	
2	Aceptar el mensaje.	
3	Vuelve al paso 1 del flujo básico eliminar una TC.	
<b>Flujo alternativo 5. a Selección de opciones</b>		
1	El sistema muestra un mensaje indicando marcar como mínimo una opción	
2	Acepta el mensaje.	
3	Vuelve al paso 5 del flujo básico eliminar una TC.	
<b>Validaciones</b>		
	N/A	
<b>Relaciones</b>	<b>Requisitos</b>	N/A
	<b>Incluidos</b>	
	<b>Extensiones</b>	N/A
<b>Conceptos</b>		N/A
<b>Requisitos especiales</b>	N/A	
<b>Asuntos pendientes</b>	N/A	

En la tabla 6 se muestra la descripción del requisito Registrar TC.

*Tabla 6 Descripción del requisito "ListarTC"*

<b>Precondiciones</b>	El Especialista se ha autenticado en el sistema.
<b>Flujo de eventos</b>	

<b>Flujo básico Cobrar cheque devuelto</b>		
1	El sistema muestra las tablas	
<b>Pos-condiciones</b>		
1	Se listaron las tablas.	
<b>Flujos alternativos</b>		
N/A		
<b>Validaciones</b>		
N/A		
<b>Relaciones</b>	<b>Requisitos</b>	N/A
	<b>Incluidos</b>	
	<b>Extensiones</b>	N/A
<b>Conceptos</b>		N/A
<b>Requisitos especiales</b>	N/A	
<b>Asuntos pendientes</b>	N/A	

#### 2.5.4 Técnicas para la validación de los requisitos

Siguiendo el procedimiento para la Ingeniería de Requisitos en el Departamento de Desarrollo de Soluciones para la Aduana del CEIGE, el cual plantea sobre las técnicas para la validación de los requisitos lo siguiente (24):

En el procedimiento propuesto se recomienda el uso de varias técnicas para la correcta validación de los requisitos en el Departamento de Soluciones para la Aduana las cuales son: revisiones del documento de requerimientos, construcción de prototipos, matrices de trazabilidad y generación de casos de pruebas.

La validación de los requisitos se realizó con la combinación de las siguientes técnicas: revisiones del documento de requerimientos y construcción de prototipos. Se realizaron diversas revisiones al documento de requerimientos con la participación de los analistas del proyecto, jefe del subsistema, jefe del proyecto y especialistas del CADI para lograr una correcta interpretación de la información transmitida, los señalamientos planteados fueron recogidos y rectificados posteriormente. Además se efectuó la construcción de prototipos, los cuales fueron presentados por cada requisito de software, aclarando con

la técnica aduanera encargada del trabajo con el módulo de TC, si las necesidades del módulo de TC fueron cubiertas por el sistema de control.

## **2.6 Patrones de diseño utilizados en la solución.**

Un Patrón de Diseño es una solución repetible a un problema recurrente en el diseño de software, existen dos tipos de patrones, los GOF (en inglés Gang of Four) y los GRASP (en inglés General Responsibility Assignment Software Patterns) (25).

Los patrones de diseño GOF son aquellos que expresan esquemas para definir estructuras de diseño (o sus relaciones) con las que construir sistemas software (25).

Estos están clasificados en tres grupos (25):

- Patrones Creacionales: Abstraen el proceso de instanciación de objetos, ayudando a que el sistema sea independiente de cómo se crean, componen y representan sus objetos.
- Patrones Estructurales: Se encargan de cómo se combinan clases y objetos para formar estructuras más grandes.
- Patrones de Comportamiento: Tienen que ver con algoritmos y asignación de responsabilidades.

Los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en formas de patrones (25).

La utilización del framework Symfony proporciona ventajas significativas para los desarrolladores de software. Este marco de trabajo es capaz de fusionar buenas prácticas de trabajo por sí mismo, de forma que los desarrolladores no tengan que preocuparse por implementar varios de los patrones de diseño y arquitectónicos más utilizados en la actualidad, ya que el mismo framework los utiliza. Se describen a continuación los patrones usados en el desarrollo de este trabajo:

### **Model-View-Controller(MVC)**

El MVC es un patrón arquitectónico aportado por SmallTalk y hoy en día muy difundido en uso en aplicaciones de entorno web. La evolución de lo que se conoce como modelo 2 de aplicaciones web (separación de responsabilidades de presentación, negocio y navegación) avanza un poco más en el reparto de tareas en la aplicación web. Pese a que hay distintos puntos de vista acerca de la forma de aplicar e implementar este patrón, en esencia las ideas principales sobre su estructura y funcionalidad son las mismas. El MVC tiene tres piezas claves que se reparten la responsabilidad de la aplicación (25):

### **El modelo (Model)**

La lógica de negocio de las aplicaciones web depende casi siempre en su modelo de datos. El componente que se encarga por defecto de gestionar el modelo en Symfony es una capa de tipo ORM (object/relational mapping) realizada mediante el proyecto Propel. En las aplicaciones Symfony, el acceso y la modificación de los datos almacenados en la base de datos se realiza mediante objetos; de esta forma nunca se accede de forma explícita a la base de datos. Este comportamiento permite un alto nivel de abstracción y permite una fácil portabilidad.

La principal ventaja que aporta el ORM es la reutilización, permitiendo llamar a los métodos de un objeto de datos desde varias partes de la aplicación e incluso desde diferentes aplicaciones, también encapsula la lógica de los datos.

La utilización de objetos en vez de registros y de clases en vez de tablas, tiene otra ventaja: permite añadir métodos accesorios en los objetos que no tienen relación directa con una tabla.

### **La vista (View)**

La vista se encarga de producir las páginas que se muestran como resultado de las acciones. La vista en Symfony está compuesta por diversas partes, estando cada una de ellas especialmente preparada para que pueda ser fácilmente modificable por la persona que normalmente trabaja con cada aspecto del diseño de las aplicaciones.

Los diseñadores web normalmente trabajan con las plantillas y con el layout. Estas partes están formadas por código HTML que contiene pequeños trozos de código PHP, que normalmente son llamadas a los diversos helpers disponibles.

Para mejorar la reutilización de código, los programadores suelen extraer trozos de las plantillas y los transforman en componentes y elementos parciales. De esta forma, el layout se modifica para definir zonas en las que se insertan componentes externos. Los diseñadores web también pueden trabajar fácilmente con estos trozos de plantillas.

Los programadores normalmente centran su trabajo relativo a la vista en los archivos de configuración YAML (que permiten establecer opciones para las propiedades de la respuesta y para otros elementos de la interfaz) y en el objeto respuesta. Cuando se trabaja con variables en las plantillas, deben considerarse los posibles riesgos de seguridad de XSS (cross-site scripting) por lo que es necesario conocer las técnicas de escape de los caracteres introducidos por los usuarios. Independientemente del tipo de trabajo, existen herramientas y utilidades para simplificar y acelerar el trabajo (normalmente tedioso) de presentar los resultados de las acciones. En este capítulo se detallan todas estas herramientas.

### **El controlador (Controller)**

Responsable del flujo de control, la navegabilidad y el estado de la entrada del usuario. Habitualmente implementado por medio de un servlet (en proyectos java, lógicamente) es el corazón del funcionamiento del patrón. Es responsable de:

1. Interceptar y recoger las peticiones http del cliente. Así, el cliente no invocará directamente ninguna página jsp o html, sino que será redireccionado adecuadamente por el controlador.
2. Traducir la petición en una operación de negocio específica.
3. Invocar la operación o bien delegar en un manejador.
4. Determinar la siguiente vista a mostrarle al cliente
5. Retornar el control al cliente.
6. El hecho de que todas las peticiones http pasen por el controlador facilita el mantenimiento de la aplicación, sobre todo en lo referente al control de la navegabilidad, sustitución de páginas, etc.

## **Patrones GOF**

### **Adapter**

Problema: ¿Cómo tener una única interfaz de acceso a múltiples bases de datos? (26)

Propósito: Convertir la interfaz de una clase para que se adapte a lo que el cliente que la usa necesita, permitiendo así que trabajen juntas clases cuyas interfaces son incompatibles (26).

Solución: Symfony da la posibilidad de cambiar a otro sistema de base de datos completamente diferente a mitad de desarrollo, sólo basta con configurar un archivo YAML. La capa de abstracción utilizada encapsula toda la lógica de los datos. El resto de la aplicación no tiene que preocuparse por las consultas SQL y el código SQL que se encarga del acceso a la base de datos. Un ejemplo utilizado en la aplicación es:

```
Propel::getConnection('tc');
```

### **Singleton**

Problema: Obtener un punto de acceso global a una clase (26).

Propósito: Asegurar que sólo exista una instancia de una clase específica en un sistema a desarrollar (26).

Solución: Este patrón se evidencia en el controlador frontal, donde hay una llamada a la función `sfContext::getInstance()` que garantiza que siempre se acceda a la misma instancia.

### **Command**

Este patrón se observa en la clase `sfWebFrontController`, en el método `dispatch()`. Esta clase está por defecto y es la encargada de establecer el módulo y la acción que se va a usar según la petición del

usuario. Este patrón se aplica además en la clase sfRouting, que está desactivada por defecto y procede según las necesidades del administrador del sistema donde se aplique el framework, la cual se puede activar o desactivar. En este método es parseada la URL con el objetivo de precisar los parámetros de la misma y de esta forma saber el Actions que debe responder a la petición (26).

## **Patrones GRASP**

### **Bajo acoplamiento**

Problema: ¿Cómo dar soporte a una dependencia escasa y a un aumento de la reutilización? (26)

Propósito: Aumentar la reutilización y eliminar las dependencias entre las clases para propiciar un fácil mantenimiento y entendimiento (26).

Solución: Symfony asigna a cada clase una responsabilidad para mantener pocas dependencias entre las mismas. La aplicación desarrollada mantiene el sistema de clases generadas por symfony y sus dependencias.

### **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 (26). 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. Cada módulo tiene una clase controladora llamada action.class.php que es la encargada de atender todas las peticiones y pasar los datos de la misma a las clases del modelo para su procesamiento, y es la que a su vez se encarga de enviar las respuestas a la vista.

## **2.7 Diagrama de Clases del Diseño con estereotipos Web.**

El diagrama de clases del diseño describe gráficamente las especificaciones de las clases de software y de las interfaces en una aplicación. En la figura 4 se muestra el diagrama de clases del diseño con estereotipos web del requisito funcional Eliminar TC, donde se representan las principales clases, operaciones y relaciones que se necesitan para darle cumplimiento al requisito.



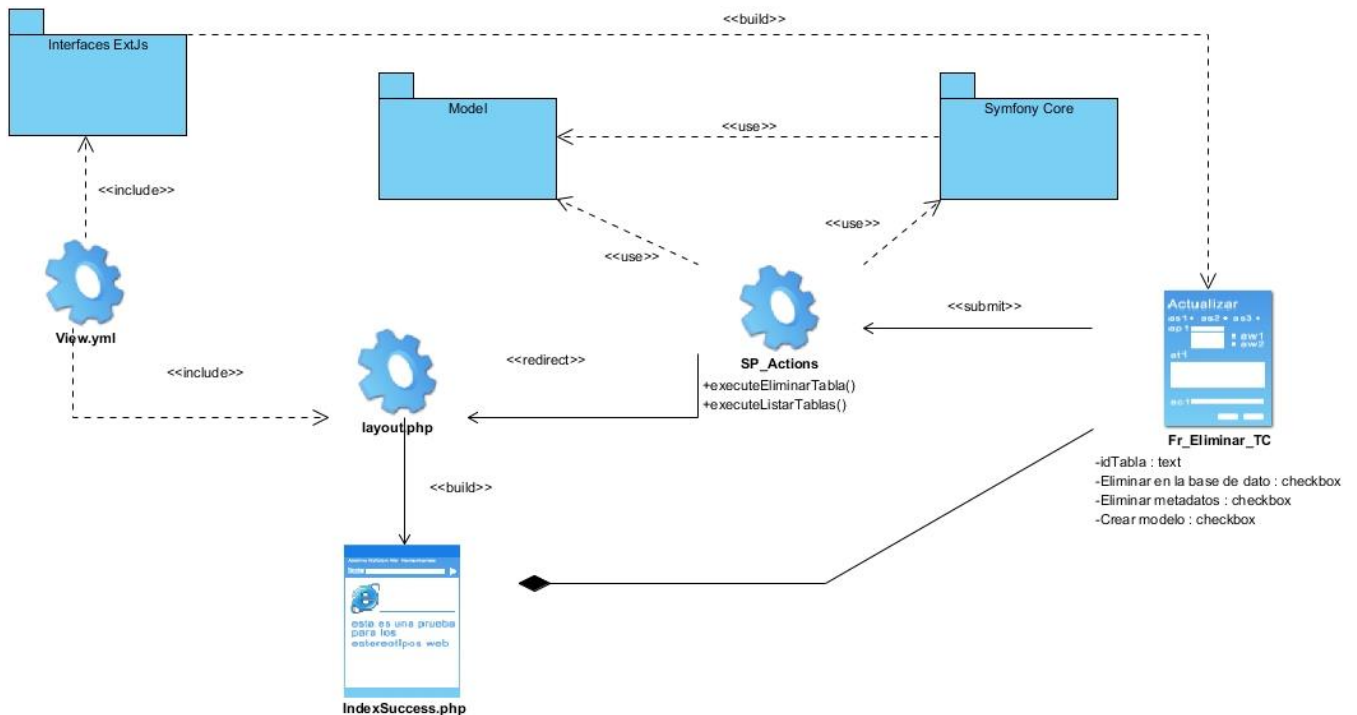


Figura 4 Diagrama de clases del diseño "Eliminar TC"

## 2.8 Diagrama de Secuencia.

Un diagrama de secuencia muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela para cada caso de uso. Este contiene detalles de implementación del escenario, incluyendo los objetos y clases que se usan para implementar el escenario, y mensajes intercambiados entre los objetos. Además, describen la secuencia de acciones en un caso de uso que comienza cuando el actor invoca el caso de uso mediante el envío de algún tipo de mensaje al sistema

En la figura 5 se muestra el diagrama de secuencia del requisito funcional "RegistrarTC", donde el actor "Técnica Aduanera" comienza la interacción al elegir la opción "Registrar tabla de control". También se representa la clase controladora Actino.class.php que llama a la clase Tabla.class.php que contiene el método para eliminar una TC y es la que se comunica con las clases del modelo SisTcTablaPeer.php, SisTcCampoPeer.php, SisCampoForaneoPeer.php, SisTcDescendientePeer.php y RelTablaSubsistemaPeer.php, que permiten modificar los registros de estas tablas en la base de datos. Esta secuencia de eventos permite mostrar al usuario como se realiza la ejecución de este requisito en el sistema.

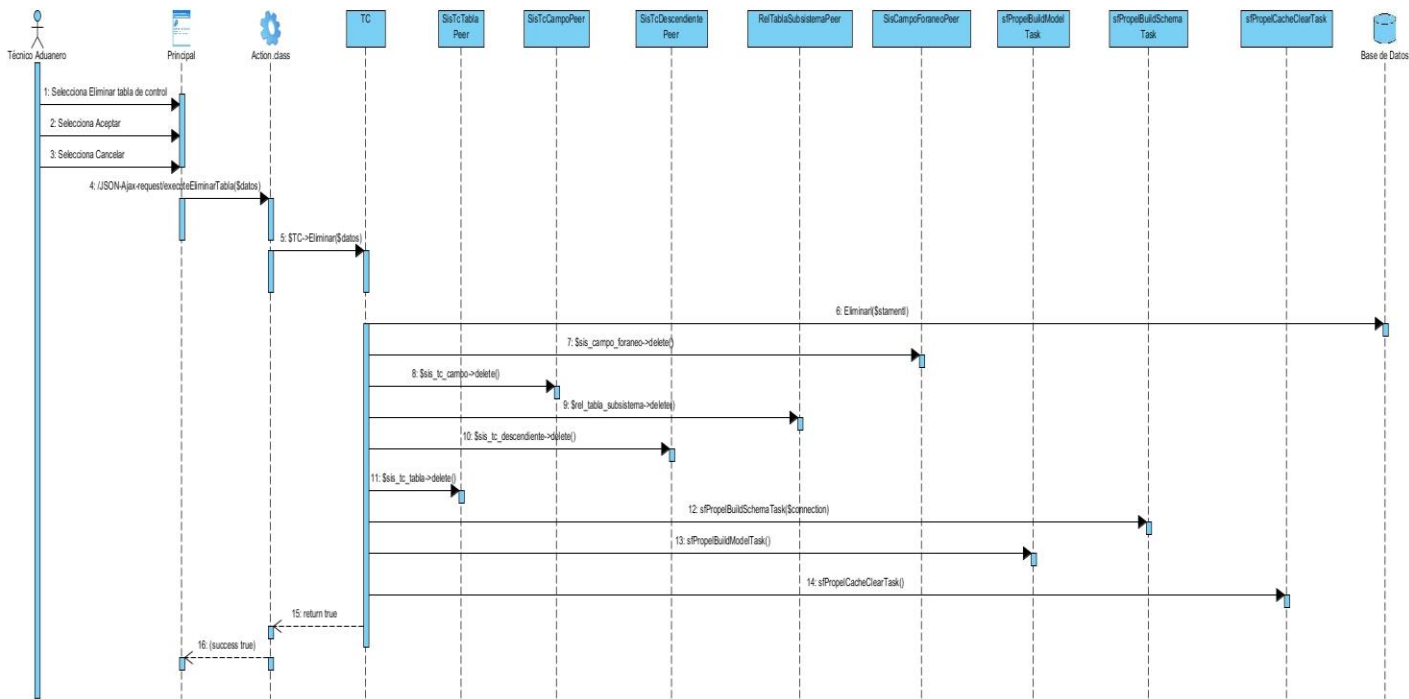


Figura 5 Diagrama de secuencia "Registrar TC"

## 2.9 Métricas para la validación del diseño

Deben usarse un conjunto de métricas de validación del diseño, para corroborar si el diseño que se ha propuesto cumple con los de requisitos de calidad que debe tener para considerarse adecuado y que no representa ningún tipo de limitante para realizar la implementación basada en esta propuesta.

### Métricas Orientadas a Objeto

Las métricas Orientadas a Objetos (OO) deben de ajustarse a las características que distinguen el software Orientado a Objetos del software convencional. Estas métricas hacen hincapié en el encapsulamiento, la herencia, complejidad de clases y polimorfismo (27).

Los objetivos principales de las métricas orientadas a objetos son los mismos que los existentes para las métricas surgidas para el software estructurado (27):

- Evaluar mejor la calidad del producto.
- Estimar la efectividad del proceso.
- Mejorar la calidad del trabajo realizado en el nivel del proyecto.

### Métricas propuestas por Lorenz y Kidd

Existe un grupo de métricas muy conocidas que fueron propuestas por estos ingenieros, también basadas en las clases como la base para la obtención de métricas especializadas. La gran diferencia consiste en las definiciones hacia las cuales fueron dirigidas en su concepción (27):

- Tamaño
- Herencia
- Valores Internos
- Valores Externos

### Métrica Tamaño Operacional de Clase

La métrica Tamaño Operacional de Clase (TOC) se emplea con el propósito de mejorar la comprensión de la calidad de la solución, estimar la efectividad del proceso y mejorar la calidad del trabajo. Está diseñada para evaluar los siguientes atributos de calidad:

- Complejidad de las pruebas: Se refiere al grado de dificultad que tiene implementado un diseño de clases determinado, lo que complejiza la aplicación de pruebas al sistema.
- Responsabilidad: Consiste en la responsabilidad asignada a una clase de un dominio.
- Reutilización: Consiste en el grado de reutilización presente en una clase dentro de un diseño de software.

Dicha métrica consiste en medir el tamaño general de una clase tomando el valor de la cantidad de operaciones y el número de atributos que están encapsulados dentro de dicha clase. Si el resultado obtenido indica valores grandes, significa que la clase posee un alto grado de responsabilidad, debido a esto se reducirá la reutilización, se hará mucho más difícil la implementación y la realización de pruebas a dicha clase (27).

En la tabla 7 se muestran las medidas utilizadas para evaluar cada uno de estos parámetros de calidad.

*Tabla 7 Umbrales de la métrica TOC*

	Categoría	Criterio
Responsabilidad	Baja	$\leq 11.85$ .
	Media	Entre 11.85. y 23.7
	Alta	$> 23.7$ .

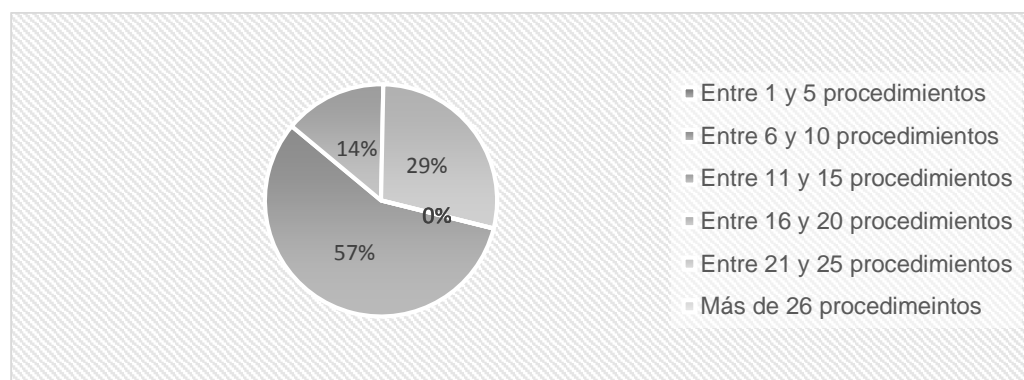
Complejidad implementación	Baja	$\leq 11.85$ .
	Media	Entre 11.85. y 23.7
	Alta	$> 23.7$ .

Reutilización	Baja	$> 23.7$ .
	Media	Entre 11.85. y 23.7
	Alta	$\leq 11.85$ .

En la tabla 8 muestra el resultado de la aplicación de la métrica TOC.

*Tabla 8 Resultados por clase luego de la aplicación de la métrica TOC*

Clase	Cantidad de Procedimientos	Responsabilidad	Complejidad	Reutilización
TC	10	Baja	Baja	Alta
SisTcTabla	18	Media	Media	Media
SisTcCampo	20	Media	Media	Media
SisCampoForaneo	8	Baja	Baja	Alta
SisTcDescendiente	6	Baja	Baja	Alta
RelTablaSubsistema	8	Baja	Baja	Alta
Action. class	13	Media	Media	Media



*Figura 6 Cantidad de procedimientos por clase para la métrica TOC*

En la figura 7 se muestran los resultados de la evaluación de la métrica TOC:

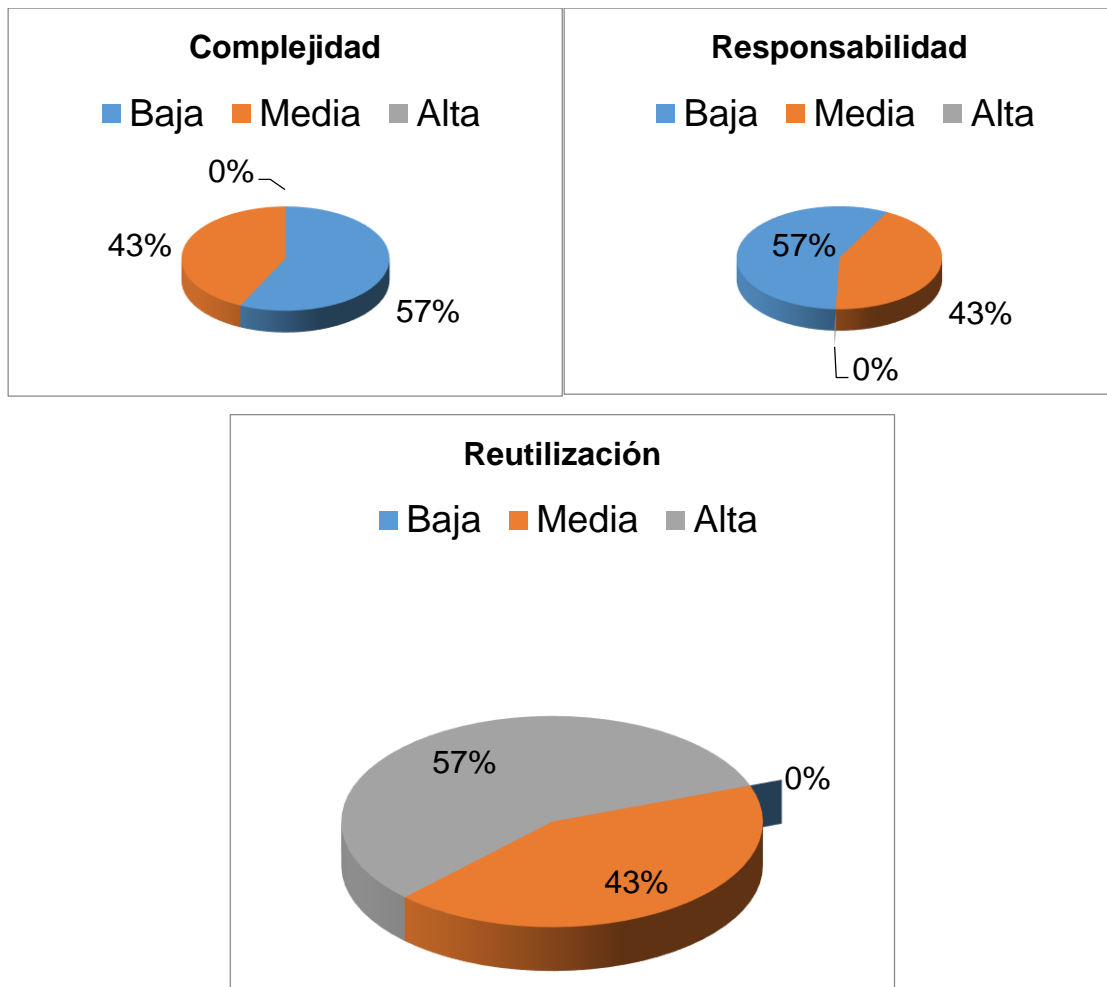


Figura 7 Resultados de la métrica Tamaño de Clase

Después de mostrados los datos en las tablas anteriores se puede apreciar que las clases se clasifican en pequeñas. Además, al observar los parámetros de calidad podemos comprobar que el 57% de las clases tienen responsabilidad y complejidad bajas, el 43% de las clases poseen responsabilidad y complejidad media y un 0% alta. Respecto a la reutilización el 57% de las clases presentan elevados índices de reutilización, el 43% tiene una reutilización media y un 0% presenta una reutilización baja. Finalmente, se demuestra la presencia de niveles bajos de complejidad y responsabilidad asociados con altos índices de reutilización, lo que facilitará la implementación de la solución. Estos valores demuestran que los indicadores de reutilización, complejidad y responsabilidad no se ven afectados.

### Relaciones ente clases (RC)

Esta métrica está dada por el número de relaciones de uso de una clase con otra y evalúa los siguientes atributos de calidad:

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

En la tabla 9 se muestran las medidas utilizadas para evaluar cada uno de estos parámetros de calidad.

*Tabla 9 Umbrales de la métrica RC*

	Categoría	Criterio
Acoplamiento	Ninguno	$\leq 0$
	Bajo	$\leq 3$
	Medio	3
	Alto	$>3$

	Categoría	Criterio
Complejidad de mantenimiento.	Baja	$\leq 1.3.$
	Media	Entre 1.3. y 2.6.
	Alta	$> 2.6.$

	Categoría	Criterio
Reutilización	Baja	$>2.6.$
	Media	Entre 1.3. y 2.6.
	Alta	$\leq 1.3.$

	Categoría	Criterio
Cantidad de Pruebas	Baja	$\leq 1.3.$
	Media	Entre 1.3. y 2.6.
	Alta	$> 2.6.$

En la tabla 10 muestra el resultado de la aplicación de la métrica RC.

*Tabla 10 Resultados por clase luego de la aplicación de la métrica RC*

Clase	Cantidad de Relaciones de Uso	Acoplamiento	Complejidad de Mantenimiento	Reutilización	Cantidad de Pruebas
Action. Class	4	Alto	Media	Media	Media
TC	5	Alto	Media	Media	Media
SisTcTabla	0	Ninguno	Baja	Alta	Baja
SisTcCampo	0	Ninguno	Baja	Alta	Baja
SisTcDescendiente	0	Ninguno	Baja	Alta	Baja
SisCampoForaneo	0	Ninguno	Baja	Alta	Baja
RelTablaSubsistema	0	Ninguno	Baja	Alta	Baja

En la figura 10 se muestran los resultados de la evaluación de la métrica RC

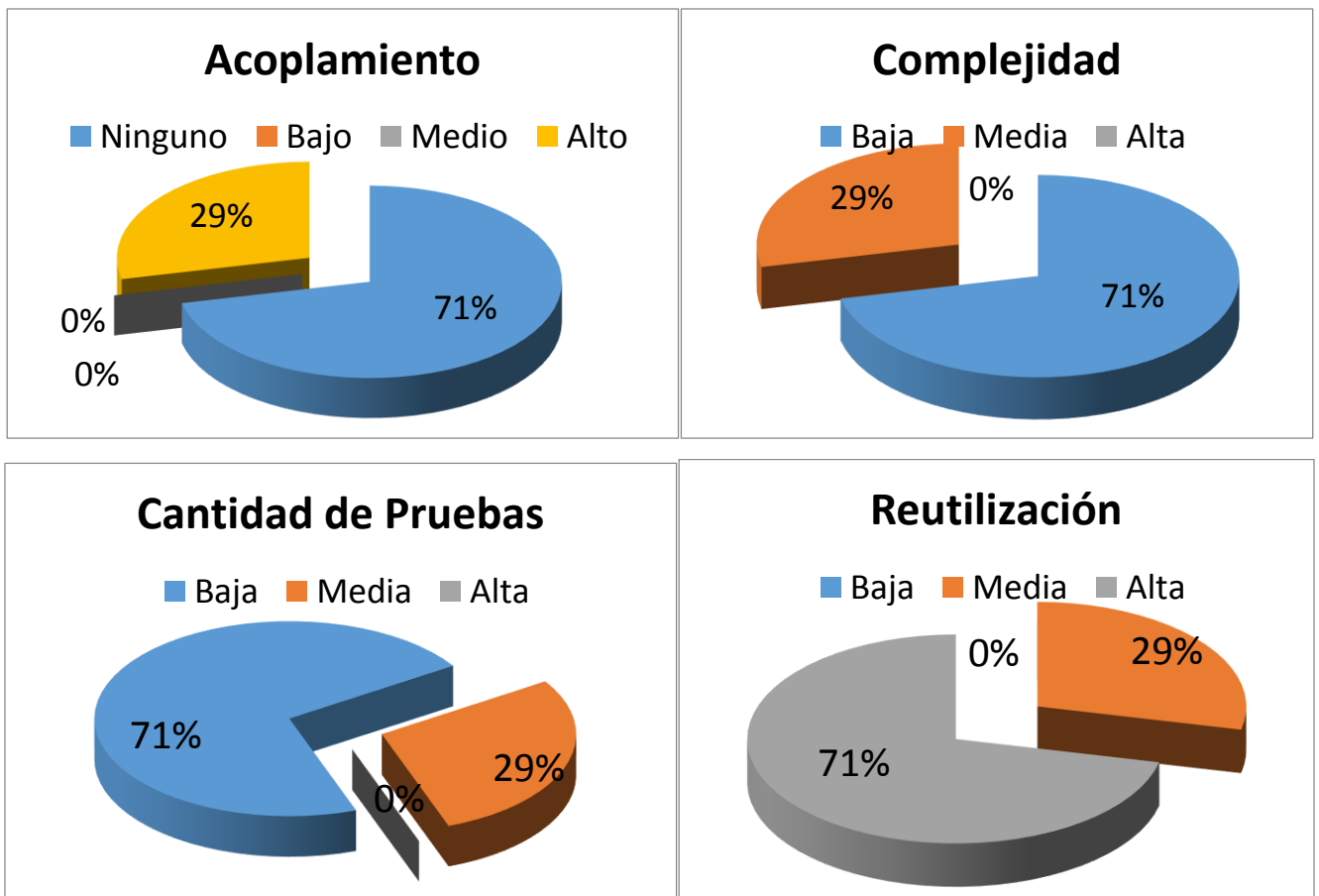


Figura 8 Resultados de la métrica RC

Al analizar los resultados obtenidos para cada uno de los atributos de calidad que mide esta métrica se evidencia que el 71 % de las clases no posee acoplamiento y un 29 % lo tiene alto. Por otra parte el 71% de las clases presentan una complejidad de mantenimiento baja, el 29% media y un 0% alta, el 71 %

presenta una cantidad de pruebas baja, el 29 % media y el 0 % alta, también el 71% de las clases presenta una alta reutilización, el 29 % media y el 0% baja. Estos datos demuestran que las clases están bien diseñadas ya que al presentar un acoplamiento relativamente bajo junto con una complejidad de mantenimiento baja y una reutilización alta facilita la implementación de estas clases. El hecho de presentar una cantidad de pruebas baja implica que se necesitará menos esfuerzo a la hora de realizar pruebas unitarias a estas clases.

## **2.10 Modelo de Datos.**

Una base de datos sirve para almacenar la información que se utiliza en un sistema de información determinado. Las necesidades y los requisitos de los futuros usuarios del sistema de información se deben tener en cuenta para poder tomar adecuadamente las decisiones anteriores.

Una forma de agrupar los datos es el Modelo de Entidad-Relación que no es más que un modelo de datos basado en una percepción del mundo real que consiste en un conjunto de objetos básicos llamados entidades y relaciones entre estos objetos, implementándose en forma gráfica (29)

El siguiente diagrama entidad-relación, se encuentra normalizado en Tercera Forma Normal (3FN), según la propuesta original de Codd (1972), que plantea que una relación está en tercera forma normal si todas las relaciones poseen atributos atómicos (1FN), además si todo atributo no primo de la relación depende funcionalmente o de manera total de la clave primaria de la misma (2FN) y por último, si todos los atributos de la relación dependen funcionalmente sólo de la clave, y no de ningún otro atributo. A continuación se presenta el diagrama de entidad-relación correspondiente al módulo de TC.



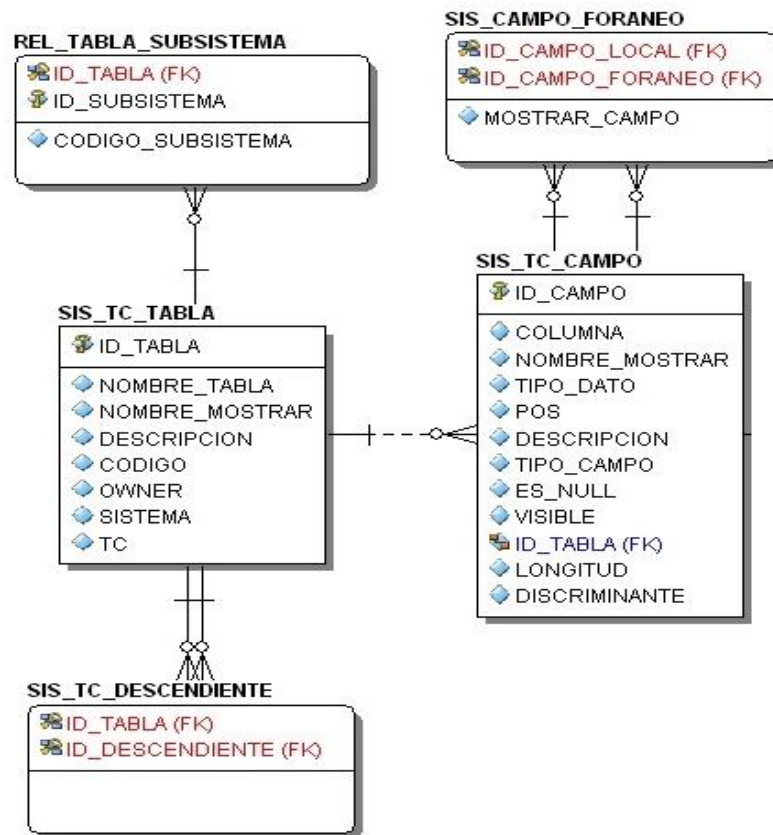


Figura 9 Diagrama entidad-relación

## 2.11 Conclusiones del Capítulo.

El estudio realizado en este capítulo permitió determinar lo siguiente:

- La identificación de los principales conceptos que rigen el negocio a través del diagrama conceptual facilitó el entendimiento del entorno en el cual se desarrolla el problema a resolver.
- El análisis y especificación de los requisitos funcionales y no funcionales, así como la elaboración de prototipos, tributó a un mejor trabajo en las próximas fases del desarrollo y posibilitó a los desarrolladores tener una mejor visión de las características que debe tener el sistema.
- El desarrollo de los diagramas de clases del diseño, secuencia y el modelo de datos, posibilitó establecer una guía para lograr la correcta implementación del módulo TC, teniendo siempre en cuenta los patrones de diseño definidos. Además, la validación del diseño permitió valorar positivamente el mismo y no arrastrar errores a etapas posteriores.

## CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA

### 3.1 Introducción

En este capítulo se describe cómo los elementos del modelo de diseño son implementados en términos de componentes y cómo estos se organizan de acuerdo a los nodos específicos en el modelo de despliegue. Se define el estándar de codificación que será utilizado durante la implementación y el tratamiento de los errores dentro del sistema. Así como las pruebas que se llevaran a cabo para la validación del sistema.

### 3.2 Modelo de implementación del sistema

A partir del resultado del diseño se comienza la implementación del sistema en términos de componentes, es decir, ficheros de código fuente, scripts, ficheros de código binario, ejecutables y similares. Los diagramas de despliegue y componentes, son productos de trabajo generados en este flujo de trabajo, conforman lo que se conoce como un modelo de implementación al describir los componentes a construir, su organización y dependencia entre los nodos físicos en los que funcionará la aplicación.

#### 3.2.1 Estándares de codificación

Los estándares de codificación se utilizan para mantener un estilo de programación igual para todos los desarrolladores del proyecto, permitiendo que el personal del proyecto identifique de forma sencilla cuales son los objetivos de cada uno de las funcionalidades que brindan las clases, funciones y demás componentes de software dada su nomenclatura, sirviendo así de guía para posteriores desarrolladores. Se utilizó el estándar definido por el Departamento de Soluciones para la Aduana, al cual pertenece el desarrollo de la solución de este trabajo.

A continuación se definen algunas de las reglas más importantes a tener en cuenta a la hora de la implementación de la solución (30):

- Las aplicaciones deben tener nombres que dejen reflejado bien claramente cuál es el propósito de la misma, ya sea en una palabra o siglas (En caso de ser mediante siglas se pondrán todas en mayúsculas).

- Los nombres de las funciones deben estar en la nomenclatura CamelCase<sup>4</sup> y las acciones además tienen que comenzar por la palabra “execute”.
- Los nombres de las funciones deben dejar reflejado claramente cual es la acción que realiza el mismo.
- Symfony especifica que la notación de los plugins siempre debe estar atada al sufijo Plugin. El nombre de un plugin debe indicarse con la menor cantidad de palabras cual es el objetivo del plugin e iniciarse con el prefijo sf.
- Las clases del modelo generadas por Propel deben cumplir con la nomenclatura “CamelCase” y sustituir los guiones bajos de los nombres de las tablas por palabras compuestas. Además el uso de los sufijos “Peer” al final de la palabra, para las clases que se encargan de estas funciones en el modelo.
- Los formularios tendrán el sufijo Form para identificarlos.
- Los nombres de los archivos de las clases deben estar compuestos por el nombre de la clase seguido de un punto y la palabra “class” y la extensión del archivo “. php”. (Ej. : actions. class. php)
- 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.
- Para la documentación de las variables y las funciones se utilizarán notas de bloque donde se especifique un comentario de la variable o función con el objetivo de la misma, el o los tipos de parámetros y el tipo de retorno si es necesario para la función.

En la figura 10 se muestra un ejemplo donde se aplican estos estándares de codificación.

---

<sup>4</sup> CamelCase: Consisten escribir frases o palabras compuestas eliminando los espacios intermedios y poniendo en mayúscula la primera letra de cada palabra, incluyendo la primera.

```

154
155 public function executeCrearTabla(sfWebRequest $request) {
156
157     $nomb = $this->getRequestParameter('nomT');
158     $columnas = $this->getRequestParameter('columnas');
159     $tipo = $this->getRequestParameter('tipo_tabla');
160     $modificar = $this->getRequestParameter('modificar');
161     $modelo = $this->getRequestParameter('modelo');
162     $metadatos = $this->getRequestParameter('metadatos');
163     $con = $this->getRequestParameter('subsistema');
164     try {
165         $tabla = new Tabla();
166         $confirmacion = $tabla->RegistrarTabla($con, $nomb, $columnas,$tipo,'on','on','on');
167         return $this->renderText(json_encode(array('success' => true, $confirmacion)));
168
169     } catch (Exception $exc) {
170         return $this->renderText(json_encode(array('success' => false, 'msg' =>$exc->getMessage())));
171     }
172
173 }
174

```

Figura 10 Aplicación de los estándares de codificación

### 3.2.2 Tratamiento de errores

El correcto tratamiento de los errores en un sistema, influye notablemente en el buen funcionamiento de este, a la vez que se garantiza la integridad de la información. Para lograr esto, fueron previstos los posibles errores que pudiera generar el sistema a partir de la interacción del usuario con este sistema. En la capa de la presentación, se realizan validaciones para asegurar el formato y tipo de los datos introducidos por el usuario, como son el uso de las expresiones regulares para validar los campos con mayor complejidad. En la capa de negocio, las validaciones son realizadas a través de los validadores de formularios que provee el *framework* Symfony. Estos formularios son utilizados para las inserciones o modificaciones, ya que garantizan la limpieza de los datos obtenidos de la capa de la vista. Ante estos errores se les muestran a los usuarios un mensaje de notificación mostrando el error cometido, en el momento que ocurra. En la figura 11 se muestra un ejemplo de como se realiza el tratamiento de errores

```

154
155 public function executeCrearTabla(sfWebRequest $request) {
156
157     $nomb = $this->getRequestParameter('nomT');
158     $columnas = $this->getRequestParameter('columnas');
159     $tipo = $this->getRequestParameter('tipo_tabla');
160     $modificar = $this->getRequestParameter('modificar');
161     $modelo = $this->getRequestParameter('modelo');
162     $metadatos = $this->getRequestParameter('metadatos');
163     $scon = $this->getRequestParameter('subsistema');
164     try {
165         $tabla = new Tabla();
166         $confirmacion = $tabla->RegistrarTabla($scon, $nomb, $columnas,$tipo,'on','on','on');
167         return $this->renderText(json_encode(array('success' => true, $confirmacion)));
168
169     } catch (Exception $exc) {
170         return $this->renderText(json_encode(array('success' => false, 'msg' =>$exc->getMessage())));
171     }
172
173 }
174

```

Figura 11 Tratamiento de errores

### 3.2.3 Diagrama de despliegue

El Diagrama de Despliegue muestra las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos, donde cada nodo representa un recurso de cómputo que poseen relaciones que representan medios de comunicación entre ellos. Representa una correspondencia entre la arquitectura software y la arquitectura del sistema (hardware). En la figura 12 se muestra el diagrama de despliegue.

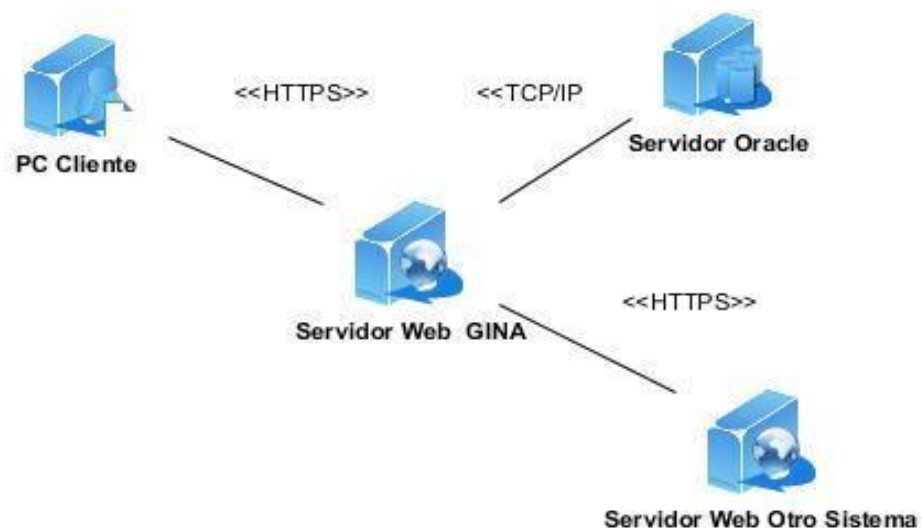


Figura 12 Diagrama de despliegue

**Servidor Oracle:**

Representa el servidor de bases de datos Oracle, en el cual se encuentran almacenados los datos del sistema GINA.

**Nodo PC Cliente:**

Representa los ordenadores personales con los cuales los usuarios del sistema harán acceso a la aplicación del sistema GINA.

**Nodo Servidor Apache GINA:**

Representa el servidor web donde se encontrará la aplicación del sistema GINA en la cual se encuentra la solución abordada en el presente trabajo.

**Nodo Servidor Web Otro Sistema:**

Representa otro servidor web de un sistema con el cual el sistema GINA puede tener intercambio de información.

### **3.3 Pruebas de validación de la solución**

Las pruebas constituyen una actividad en la cual un sistema o componente de este, es ejecutado bajo ciertas condiciones o requerimientos específicos, en el que los resultados obtenidos son observados y registrados, para la realización posterior de alguna evaluación de dicho componente o sistema. Las técnicas de evaluación dinámica o prueba proporcionan distintos criterios para generar casos de prueba que provoquen fallos en los programas. Estas técnicas se agrupan en (31):

**Técnicas de caja blanca:** Se basan en un minucioso examen de los detalles procedimentales del código a evaluar, por lo que es necesario conocer la lógica del programa (32).

**Técnicas de caja negra o funcional:** Realizan pruebas sobre la interfaz del programa a probar, entendiendo por interfaz las entradas y salidas de dicho programa. No es necesario conocer la lógica del programa, únicamente la funcionalidad que debe realizar. También conocidas como Pruebas de Comportamiento, estas pruebas se basan en la especificación del programa o componente a ser probado para elaborar los casos de prueba. El componente se ve como una “Caja Negra” cuyo comportamiento sólo puede ser determinado estudiando sus entradas y las salidas obtenidas a partir de ellas (32).

#### **3.3.1 Aplicación de pruebas de caja negra o funcional**

Para probar la solución propuesta se realizaron pruebas funcionales, utilizando el método de pruebas de caja negra mediante la técnica de particiones equivalentes. Se realizaron 3 iteraciones donde se utilizaron los casos de prueba diseñados para examinar los valores válidos e inválidos de las entradas existentes en el software.

Durante la primera iteración se encontraron un total de 8 no conformidades, sobre validaciones que debían realizarse para determinados campos. En la segunda iteración se detectaron 4 no conformidades sobre errores ortográficos, pero estas fueron corregidas en su totalidad, ya que al efectuar la tercera iteración no se detectó ninguna no conformidad. En la figura 13 se muestran los resultados obtenidos por iteración.

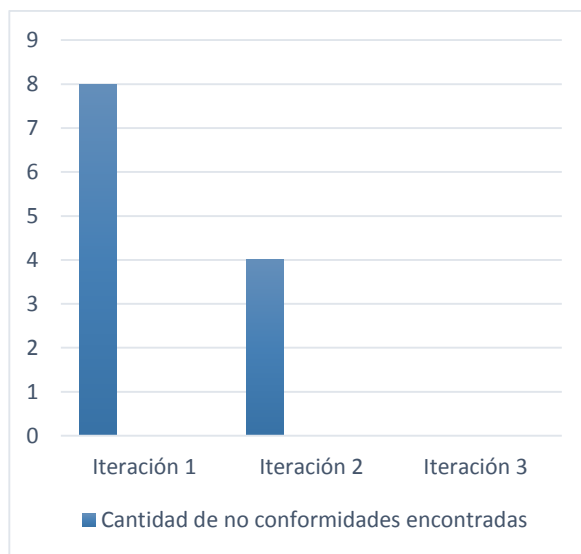


Figura 13 Resultados de las pruebas de caja negra

En la tabla 11 se muestra la descripción de las variables para validar el requisito funcional Eliminar TC.

Tabla 11 Descripción de las variables del requisito "Eliminar TC"

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Eliminar en la base de datos	Campo de selección	Si	Puede estar desmarcado, si al menos uno de los otros Campo de selección no están desmarcados
2	Eliminar Metadatos	Campo de selección	Si	Puede estar desmarcado, si al menos uno de los otros Campo de selección no están desmarcados
3	Eliminar Modelo	Campo de selección	Si	Puede estar desmarcado, si al menos uno de los otros Campo de selección no están desmarcados

En la tabla 12 se muestra los escenarios que fueron diseñados para la prueba del requisito funcional "Eliminar TC"

Tabla 12 Diseño del caso de prueba

Escenario	Descripción	Eliminar base de datos	Eliminar metadatos	Modificar modelo	Respuesta del sistema	Flujo central
EC 1.1 Eliminar TC	Se debe Eliminar la Tabla de Control	V Eliminar	V Metadatos	V Modelo	Se muestra un mensaje de confirmación "La tabla fue eliminada sin problema"	Seleccionar una de las tablas que se muestran en el grid "Tablas", Dar click en el menú "Gestionar Tabla", luego escoger la opción "Eliminar Tabla" de Control, despues seleccionar las opciones que se brindan y dar click el botón "Eliminar"
		N/A	V Metadatos	N/A		
		V Eliminar	N/A	N/A		
		N/A	N/A	V Modelo		
		N/A	N/A	N/A		
EC 1.2 Eliminar TC sin seleccionar una tabla	No se elimina la tabla, porque no se marca una de las tablas mostradas				Se muestra un mensaje indicando que se debe seleccionar una tabla antes de eliminarla	Dar click en el menú "Gestionar Tabla", luego escoger la opción "Eliminar Tabla de Control".
EC 1.3 Eliminar TC sin seleccionar las opciones	No se elimina la tabla, porque no se marca ninguna de las opciones	I	I	I	Se muestra un mensaje indicando que se debe seleccionar al menos una de las opciones antes de eliminar una tabla	Seleccionar una de las tablas que se muestran en el grid "Tablas", Dar click en el menú "Gestionar Tabla", luego escoger la opción "Eliminar Tabla" de Control y dar





```

public function EliminarTabla($id, $nomb, $con, $modificar, $metadatos, $modelo) {
    $c = new Criteria();//1
    $c->add(SisTcCampoPeer::ID_TABLA, $id);//1
    $sis_tc_campo = SisTcCampoPeer::doSelect($c);//1
    foreach ($sis_tc_campo as $value) {//2
        $this->EliminarColumna($value->getIdCampo(), $id, $con, '', $metadatos, '')://3
    }
    if ($modificar == "on") {//4
        $connection = Propel::getConnection($con);//5
        $sql = "DROP TABLE " . $nomb . " CASCADE CONSTRAINT";//5
        $statement = $connection->prepare($sql);//5
        $statement->execute();//5
        $stm = "DROP SEQUENCE " . $nomb . "_SEQ";//5
        $statement = $connection->prepare($stm);//5
        $statement->execute();//5
    }
    if ($metadatos == "on") {//6
        $sis_tc_tabla = SisTcTablaPeer::retrieveByPK($id);//7
        $sis_tc_tabla->delete();//7
    }
    if ($modelo == "on") {//8
        ini_set('max_execution_time', 1000000);//9
        $file = getcwd();//9
        chdir(sfConfig::get('sf_root_dir'));//9
        $cc = new sfPropelBuildSchemaTask(sfContext::getInstance()->getEventDispatcher(), new sfFormatter());//9
        $error = $cc->run(array(), array();//9
            '--application=' . sfConfig::get('sf_app'),//9
            '--connection=tc',//9
            '--env=dev',//9
        );
        chdir($file);//9
        $dir = sfConfig::get('sf_config_dir') . '\schema.yml'://9
        $arreglo = array();//9
        $arreglo = sfYaml::load($dir);//9
        $temp = array();//9
        foreach ($arreglo as $value) {//9
            $value["_attributes"] = array("package" => "lib.model.tc", "defaultIdMethod" => "native");
            $temp = array('tc' => $value);
        }
        file_put_contents($dir, sfYaml::dump($temp));//9
        $file = getcwd();//9
        chdir(sfConfig::get('sf_root_dir'));//9
        $cc = new sfPropelBuildModelTask(sfContext::getInstance()->getEventDispatcher(), new sfFormatter());//9
        $error = $cc->run(array(), array());//9
        chdir($file);//9
        $file = getcwd();//9
        chdir(sfConfig::get('sf_root_dir'));//9
        $cc = new sfCacheClearTask(sfContext::getInstance()->getEventDispatcher(), new sfFormatter());//9
        $cc->run();//9
        chdir($file);//9
    }
    return true;//10
}

```

Figura 14 Método EliminarTabla

El primer paso para aplicar el método del camino básico es obtener el grafo del flujo, a partir del código de la función anterior. En la figura 15 se muestra el grafo correspondiente al código anterior

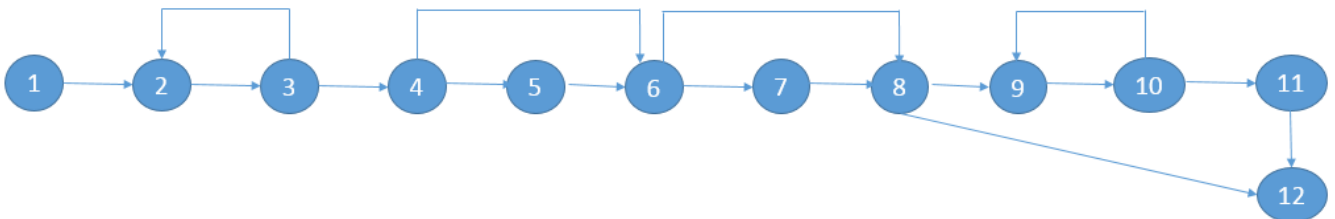


Figura 15 Grafo del flujo correspondiente a la función EliminarTabla

### Cálculo de la complejidad ciclomática.

La complejidad ciclomática es una métrica de software extremadamente útil pues proporciona una edición cuantitativa de la complejidad lógica de un programa. El valor calculado como complejidad ciclomática define el número de caminos independientes del conjunto básico de un programa y da un límite superior para el número de pruebas que se deben realizar para asegurar que se ejecute cada sentencia al menos una vez (12).

Una vez construido el grafo se determina la complejidad ciclomática, el cálculo es necesario efectuarlo mediante tres fórmulas, de manera tal que sea el mismo resultado en cada caso:

La complejidad ciclomática  $V(G)$  de un grafo de flujo  $G$  se define como:

- $V(G) = \text{cantidad de aristas}(A) - \text{cantidad de nodos}(N) + 2$   
 $V(G) = 16 - 12 + 2 = 6$
- $V(G) = \text{cantidad de nodos predicados}(P) + 1$   
 $V(G) = 5 + 1 = 6$
- $V(G) = \text{número de regiones del grafo de flujo}(R)$   
 $V(G) = 6$

Luego de efectuar el cálculo se obtuvo una complejidad ciclomática de valor 6, de manera que existen 6 posibles caminos por donde el flujo puede circular, este valor representa el número mínimo de casos de pruebas para el procedimiento analizado. A continuación se especifican los caminos básicos que puede tomar el algoritmo durante su ejecución.

- Camino básico #1: 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 10 – 11 – 12
- Camino básico #2: 1 – 2 – 3 – 4 – 6 – 7 – 8 – 9 – 10 – 11 – 12
- Camino básico #3: 1 – 2 – 3 – 4 – 6 – 8 – 9 – 10 – 11 – 12
- Camino básico #4: 1 – 2 – 3 – 4 – 6 – 7 – 8 – 12
- Camino básico #5: 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 12
- Camino básico #6: 1 – 2 – 3 – 4 – 5 – 6 – 8 – 12

Posteriormente se procede a ejecutar un caso de prueba.

- Caso de prueba para el camino básico #1
    - Descripción: Se quiere eliminar una tabla con las opciones de eliminar en la base de datos y eliminar los metadatos.
    - Condición de ejecución: Se selecciona un tabla y al menos una de las opciones a realizar.
    - Entrada: \$idTabla, \$nombreTabla, \$conexion, \$modificar, \$metadatos, \$modelo
    - Resultados esperados: Teniendo en cuenta los datos pasados por parámetro se espera que la tabla se elimine y se envíe un mensaje de confirmación.
- El resultado fue el esperado.

Para corroborar que con la ejecución del caso de prueba anterior se logró el resultado esperado, se aplicaron las pruebas unitarias de *Symfony*. Estas son archivos PHP normales cuyo nombre termina en Test.php y se encuentran en el directorio test/unit/ de la aplicación. Su sintaxis es sencilla y fácil de implementar además, son hechas por los propios desarrolladores. Para ejecutar las pruebas, se utiliza la tarea test:unit desde la línea de comandos. El resultado de esta tarea en la línea de comandos permite localizar fácilmente las pruebas que han fallado y las que se han ejecutado correctamente. Las pruebas unitarias del framework *Symfony* para el caso anterior se definieron de la siguiente forma:

```
<?php

require_once dirname(__FILE__) . '/../../bootstrap/unit.php';
$configuration = ProjectConfiguration::getApplicationConfiguration('tc', 'test', true);
$t = new lime_test(1, new lime_output_color());
$tabla = new Tabla();
$t->isa_ok($tabla->EliminarTabla(3456, "TC_PROVINCIA", "tc", 'on', 'on', 'on'),
    true, 'La tabla fue eliminada correctamente');
?>
```

Figura 16 Definición de las pruebas de *Symfony*

El ejecutar estas pruebas el resultado arrojado es el siguiente:

```
ok 1 - La tabla fue eliminada correctamente
Looks like everything went fine.
```

Figura 17 Resultado de la prueba de *Symfony*

El resultado anterior demuestra que la prueba realizada se ejecuta correctamente.

### 3.4 Validación de la investigación

Para validar el resultado final de la investigación en su aporte práctico, se aplica la técnica de validación de expertos tomando a aquellas personas especializadas en el trabajo con la gestión de los nomencladores. Dichos expertos se encuentran en estrecha relación con el negocio, ya sea en la Aduana General de la República o integrados al proyecto GINA.

El primer paso para aplicar la técnica de validación a través del criterio de expertos es hacer la selección y priorizar el grado de responsabilidad de los involucrados. El grupo está compuesto por cuatro especialistas en el trabajo con el módulo de TC. Estos cuentan con un grupo de competencias que les permiten poseer las bases conceptuales para conocer cómo debe funcionar el sistema en el entorno en el que se va a desplegar y cómo se lleva a cabo actualmente el procedimiento para la gestión de las nuevas TC.

## Competencias de los expertos

Para realizar el proceso de selección de los expertos mencionados seguidamente, se tuvo en cuenta que reunieran los siguientes aspectos:

- Concepción global de la solución propuesta.
- Experiencia en el trabajo con el módulo.
- Conocimiento sobre el framework Symfony 1.2 y el SGBD Oracle 11g.

Código	Nombre	Especialidad	Grado de responsabilidad
EXP-01			
EXP-02			
EXP-03			
EXP-04			

Una vez determinado el conjunto de expertos para validar, es preciso definir la herramienta de medición que se va a utilizar, en este caso la encuesta que se relaciona a continuación. Las preguntas de la misma están enfocadas a contrastar la versión anterior con las nuevas funcionalidades desarrolladas en esta investigación.

### Encuesta

La presente encuesta se realiza con la intención de conocer su grado de satisfacción con la nueva versión del módulo de Tablas de Control (**TC**) del Sistema de Gestión Integrado de Aduana (**GINA**) es de **carácter anónimo** por lo que se solicita sean respondidas las preguntas con la mayor sinceridad y objetividad posible. **Por su tiempo y colaboración, gracias.**

En cada caso, marque con una X la respuesta que considere más acertada.

1. En qué medida facilita la nueva versión el proceso de gestión de las TC en cuanto a:

	Mucho	Poco	Nada
El usuario no necesita de conocimientos previos para manejar la aplicación.			
Rapidez en la gestión de las tablas			
Gestión de las TC			

2. Durante el proceso de trabajo con la aplicación, ¿varía la estructura de las TC?

\_\_\_ Sí \_\_\_ No

3. ¿En qué medida influye la nueva versión en el rendimiento de los trabajadores?

\_\_\_ Aumenta \_\_\_ Se mantiene \_\_\_ Disminuye

4. Considera usted que la implementación de la nueva versión del sistema es una decisión

\_\_\_ Favorable \_\_\_ Desfavorable

5. Según su criterio, la interfaz visual de la nueva versión es (puede señalar más de una opción)

Agradable	
No agradable	
Simple	
Cargada de elementos innecesarios	
Difícil de entender	
Fácil de entender	

Luego de elaborada la encuesta se procede a su aplicación y los encuestados manifestaron su opinión sobre los diferentes criterios que se enuncian a continuación, seleccionando las posibles opciones en cada caso:

- Medida en que facilita la nueva versión el proceso de gestión de las TC en cuanto a tres criterios:
  - Casos en blanco (existía desconocimiento), se registró con el número 4.
  - Mucho, se registró con el número 3.
  - Poco, se registró con el número 2.
  - Nada, se registró con el número 1.
- Existencia de variedad en la estructura de las TC:
  - Casos en blanco (existía desconocimiento), se registró con el número 3.
  - Sí, se registró con el número 2.
  - No, se registró con el número 1.
- Rendimiento de los trabajadores con la nueva versión:
  - Casos en blanco (existía desconocimiento), se registró con el número 4.
  - Aumenta, se registró con el número 3.

- Se mantiene, se registró con el número 2.
  - Disminuye, se registró con el número 1.
- Consideración sobre la implementación de la nueva versión del sistema:
- Casos en blanco (existía desconocimiento), se registró con el número 3.
  - Favorable, se registró con el número 2.
  - Desfavorable, se registró con el número 1.
- Opinión sobre la interfaz de la nueva versión:
- Casos en blanco (existía desconocimiento), se registró con el número 7.
  - Agradable, se registró con el número 6.
  - No agradable, se registró con el número 5.
  - Simple, se registró con el número 4.
  - Cargada de elementos innecesarios, se registró con el número 3.
  - Difícil de entender, se registró con el número 2.
  - Fácil de entender, se registró con el número 1.

A partir de la numeración establecida anteriormente para las posibles respuestas, se muestran en la siguiente tabla los resultados de la encuesta por preguntas para cada experto.

*Tabla 13 Resultados de la encuesta*

Expertos	Pregunta 1			Pregunta 2	Pregunta 3	Pregunta 4	Pregunta 5
	Criterio 1	Criterio 2	Criterio 3	2	3	4	5
EXP-01	1	3	3	1	3	2	1
EXP-02	1	3	3	1	3	2	6,4
EXP-03	1	3	3	1	3	2	1
EXP-04	1	3	3	1	3	2	6

Los resultados de esta encuesta arrojaron que todos los expertos consideraron que la utilización de la nueva versión “Aumenta” el rendimiento de los trabajadores, dado que la tarea de gestionar los nuevos nomencladores se realiza en un solo paso. Además, todos consideraron que se necesita de muy “Poco” tiempo para realizar la gestión de los nuevos nomencladores, que el proceso puede clasificarse como “Nada” complejo y el usuario no necesita tener conocimiento sobre las herramientas de desarrollo del sistema con el uso de la nueva versión. Asimismo se apreció, que su empleo garantiza que se respete la estructura de cada TC que se desee gestionar.

Dichos resultados también demuestran las ventajas que supone la utilización de la nueva versión del módulo TC basada en los antecedentes de cómo se realiza el trabajo actualmente. Para ello se tienen en cuenta cuatro variables de medición. En la tabla 14 se muestra la comparación realizada.

Tabla 14 Antes y después de la implementación de la nueva versión del módulo de TC.

<b>Variables de medición</b>	<b>Antes</b>	<b>Propuesta</b>
<b>Conocimiento de las herramientas</b>	La técnica aduanera debía poseer conocimientos básicos sobre el <i>framework</i> Symfony y del sistema gestor de base de datos Oracle.	La técnica aduanera no necesitará poseer ningún conocimiento sobre el <i>framework</i> Symfony y del sistema gestor de base de datos Oracle puesto que el sistema se encarga de realizar todos los procesos.
<b>Estructura de las TC</b>	En el momento de registrar o modificar una TC podía no respetarse la estructura que debían tener.	En el momento en que se registre o modifique una TC siempre se respetará la estructura que esta definida para esta.
<b>Dificultad del proceso</b>	Registrar, modificar o eliminar una TC se hace un proceso engorroso.	El registro, modificación o eliminación será un proceso simple, pues la acción se realiza en un solo paso.
<b>Tiempo</b>	Se necesitaba de mucho tiempo para registrar, modificar o eliminar una TC.	Solo se necesitarán como máximo 3 minutos para realizar el registro, modificación o eliminación de una TC.

Por lo antes expuesto se puede evidenciar que el uso del nuevo módulo de TC supone una mejora con respecto a la manera en que se realiza actualmente el proceso. Teniendo en cuenta que para el trabajo diario el conocimiento de las herramientas no será un requisito necesario, que será respetada la estructura de las TC, que el procedimiento de gestión de las tablas no se tornará difícil y el tiempo necesario para hacerlo será reducido a pocos minutos.



### **3.5 Conclusiones**

El estudio realizado en este capítulo permitió determinar lo siguiente:

- El estándar de codificación que se utilizó en la implementación y el tratamiento que se le daba a los errores obtenidos en el desarrollo de la solución.
- Se muestra el diagrama de despliegue, ilustrando los nodos que serán usados para la implantación de la aplicación.
- Se aplicaron pruebas funcionales y unitarias que ayudaron a comprobar el correcto funcionamiento de la solución.
- Los resultados de instrumentos de encuesta aplicados a un grupo de expertos en el trabajo con el módulo de TC permitió demostrar que el uso de la nueva versión de este contribuirá a la mejora de la gestión de los nomencladores.

## CONCLUSIONES

Al concluir el presente trabajo se arriban a las siguientes conclusiones:

- El estudio de los sistemas aduanales o que realizan la gestión de los nomencladores tanto a nivel mundial como nacional, permitió comprender y demostrar la necesidad de desarrollar nuevas funcionalidades para mejorar la gestión de los nomencladores asociados a las TC del sistema GINA.
- La realización del análisis sobre la metodología de desarrollo de software, los lenguajes de programación y las herramientas, permitió justificar la utilización de las mismas y sentar las bases desde el punto de vista técnico, para iniciar el desarrollo.
- Con la definición de los requisitos se lograron identificar las actividades a informatizar. Además, su validación satisfactoria posibilitó la construcción del modelo de diseño.
- La aplicación de métricas al diseño permitió valorar positivamente el mismo para dar paso a la implementación, con la cual se lograron satisfacer las necesidades del cliente y resolver los problemas por los que se decidió comenzar el desarrollo de las nuevas funcionalidades.
- La aplicación de pruebas de software permitió evaluar en buena medida el nivel de calidad de la implementación de la solución.

## RECOMENDACIONES

Considerando que se han cumplido los objetivos trazados, se recomienda, que el sistema permita exportar los modelos de cada una de las TC.

## BIBLIOGRAFÍA

1. Sitio Oficial de la Aduana de la Republica. [En línea] [Citado el: 27 de 2 de 15.] [www.aduana.co.cu](http://www.aduana.co.cu).
2. [En línea] [Citado el: 3 de 12 de 2014.] [http://www.sidunea.aduana.gob.bo/capacitacion/ManualesUsuario/man\\_mod\\_sdi.pdf](http://www.sidunea.aduana.gob.bo/capacitacion/ManualesUsuario/man_mod_sdi.pdf) ..
3. Tamayo, Islandy Antonio Guevara. *Trabajo de Diploma, Analisis y Diseño del Modulo Tablas de Control del SUA*. .
4. Concepto. [En línea] [Citado el: 10 de 3 de 2015.] <http://www.concepto.uy/gia/gia--gestion-integral-aduanera..>
5. [En línea] [Citado el: 6 de 12 de 2014.] <Http://biomundi.idict.cu/sites/default/files/ponencias/SISTEMA WEB DE GESTIÓN DE LA INFORMACIÓN PARA ÓPTICAS.PDF..>
6. *Revista Cubana de Ciencias Informáticas (RCCI)*. Informáticas, Universidad de las Ciencias. 2, Cuba : Ediciones Futuro, 2012, Vol. 6.
7. Francesena, Gilberto Pérez. *Trabajo de Diploma, Diseño e Implementación del modulo Tablas de Control*. 2011.
8. Sánchez, Tamara Rodríguez. *Metodología de desarrollo para la Actividad productiva de la UCI*. La Habana : Universidad de las Ciencias Informáticas, 2014.
9. Php Documentation. *Group Hypertext Pre-processor*. [En línea] 2 de 12 de 2011. [Citado el: 5 de 1 de 2015.] <http://www.php.net/manual/es/index.php>.
10. Pérez, Javier Eguíluz. *Introducción a Javascript*. 2009.
11. Introducción a CSS. *Libros Web*. [En línea] 2008. [Citado el: 7 de 1 de 2015.] <http://www.librosweb.es/css>.
12. Pressman, Roger S. *Ingenieria de Software. Un enfoque Práctico 5ta Edición*. Madrid : Concepción Fernández, 2002.
13. json. *The JSON Specification*. [En línea] [Citado el: 20 de 1 de 2015.] <http://json.org>.
14. Zaninotto, Fabien Potencier y Francois. *Symfony la guía definitiva*. 2008.
15. Corzo, Giancarlo. *Desarrollo en Web. Blog sobre el desarrollo de aplicaciones web en Java, Python y JavaScript*. [En línea] 22 de 10 de 2008. [Citado el: 15 de 1 de 2015.] <http://blogs.antartec.com/desarrolloweb/2008/10/extjs-lo-bueno-lo-malo-y-lo-feo/>.
16. [En línea] [Citado el: 15 de 1 de 2015.] <http://extjs.com/>.
17. Propelorm. *Guía de usuario de Propel*. [En línea] [Citado el: 16 de 1 de 2015.] [http://svn.propelorm.org/tags/1.2.1/docs/es/user\\_guide/book/chapters/Introduction.html](http://svn.propelorm.org/tags/1.2.1/docs/es/user_guide/book/chapters/Introduction.html).
18. Sitio oficial del Visual Paradigm. [En línea] [Citado el: 20 de 1 de 2015.] <http://www.visual-paradigm.com>.
19. Collins-Sussmann, Ben, W. Fitzpatrick, Brian y C. Michael Pilato. *Control de versiones con Subversion: Revision 4980*. s.l. : O'Reilly Media.
20. NetBeans. *Oracle Corporation*. [En línea] [Citado el: 25 de 1 de 2015.] [http://netbeans.org/index\\_es.html](http://netbeans.org/index_es.html).
21. Ford, A. *Apache 2 pocket reference*. s.l. : O'Reilly Media.
22. Sitio oficial de Oracle. [En línea] [Citado el: 29 de 1 de 2015.] <http://www.oracle.com..>
23. Pressman. *Modelado del Análisis Parte 1*. 2008.
24. Martínez, Ing. Jenni Mason. *Procedimiento para la Ingeniería de Requisitos en el Departamento de Desarrollo de soluciones para Aduana del CEIGE*. La Habana : Universidad de las Ciencias Informáticas, 2010.

25. Mestras, Juan Pavón. Dep. Ingeniería del Software e Inteligencia Artificial. [En línea] Universidad Complutense Madrid, 2008-09. [Citado el: 10 de 2 de 2015.] <https://www.fdi.ucm.es/profesor/jpavon/poo/2.14.MVC.pdf>.
26. Larman, Craig. *UML y patrones*. Mexico : Prentice-Hall Hispanoamericana, 1999.
27. Negro, Ing. Pablo Ariel. *Umbral para métricas Orientadas a Objeto*. 2008.
28. Lorenz, M y Kidd, J. *Object-Oriented Software Metrics*. 1994.
29. Costa, Dolores Costal. *Introducción al diseño de bases de datos*.
30. Aduana, Dpto. *Propuesta de un estandar de codificación*. s.l. : Universidad de las Ciencias Informáticas.
31. James Jacobson, Ivar, Booch, Grady y Rumbaugh. *El Proceso Unificado de Desarrollo de Software*. s.l. : Addison Wesley Reading, 2000.
32. Natalia Juristo, Ana M. Moreno, Sira Vegas. *TÉCNICAS DE EVALUACIÓN DE SOFTWARE*. 2006.
33. Alvarez, J.V. *Técnicas de Prueba de Software Tema 7*. s.l. : Departamento de Informática de la Universidad de Valladolid.

# ANEXOS

## Anexo 1: Carta de aceptacion

### CENTRO DE INFORMATIZACIÓN DE GESTIÓN DE ENTIDADES

Acta de aceptación del proyecto

En la Habana a los 9 días del mes de junio de 2015

De una parte, el Departamento de Desarrollo del CADI, representado por la **Ing. Zenia Herrera O' Farrill**, Especialista al frente del Sistema de Tablas de Control, quien a los fines y efectos derivados del presente documento se denominará como **PARTE CLIENTE**, y de otra Parte el equipo de desarrollo de la versión 1.1 del módulo Configuración del Sistema de Tablas de Control, representado en este acto por **Jorge Ariel Mancha Alvarez**, que a los fines y efectos derivados del presente documento se denominará como **PARTE COMERCIALIZADORA**.

Primero: Que en cumplimiento de los acuerdos, han sido efectuadas las actividades que se describen, **Las partes DECLARAN**:

CONSIDERANDO: Que se han efectuado las actividades siguientes:

1. Modelado del negocio del componente: **Configuración de Tablas de Control**.
2. Levantamiento de requisitos.
3. Análisis y diseño.
4. Implementación.
5. Pruebas aceptación.

CONSIDERANDO: Que las actividades realizadas han sido desarrolladas con la calidad requerida y bajo las condiciones pactadas y aprobadas por **Las partes**.

CONSIDERANDO: Que las actividades que se han ejecutado cumplen con los requisitos de la **PARTE CLIENTE**.

CONSIDERANDO: que la **PARTE COMERCIALIZADORA** ha entregado la documentación que avala la ejecución de este acto a la **PARTE CLIENTE**.

POR TANTO: **Las partes** acuerdan formalizar mediante el Acta, Aceptadas las actividades que han sido ejecutadas en esta fecha.

Y para que así conste, se extiende la presente Acta en dos (2) ejemplares, rubricados por **Las partes**.

Por la PARTE CLIENTE

  
Ing. Zenia Herrera O' Farrill  
Especialista de Sistemas de Tablas de Control  
CADI

Por la PARTE COMERCIALIZADORA

  
Jorge Ariel Mancha Alvarez



## GLOSARIO

**Framework:** Estructura de soporte definida mediante la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

**Aduana:** Es la institución de un país encargada de aplicar la legislación aduanera y de recaudar los derechos e impuestos que se aplican a la importación, a la exportación, al movimiento o al almacenaje de mercancías.

**Interfaz de usuario:** Es la parte de una aplicación que se encarga de interactuar con el usuario.

**JSON:** Acrónimo de JavaScript Object Notation, 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.

**Módulo:** Es un componente autocontrolado de un sistema, el cual posee una interfaz bien definida hacia otros componentes; algo es modular si es construido de manera tal que se facilite su ensamblaje, acomodamiento flexible y reparación de sus componentes.

**AJAX:** Acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (Rich Internet Applications). Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones.

**ORM:(ObjectRelationalMapping)** Mapeo de Objetos Relacional es una técnica de programación para convertir tipos de datos incompatibles entre sistemas de bases de datos y lenguajes orientados a ob

