

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



**Título: Diseño e implementación de un núcleo de gestión de operaciones aduaneras.**

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

**Autor(es):** Juan Ernesto Borggiano Mikulenko  
Wilber Ernesto Ramirez Verdecia

**Tutor:** Ing. Alain Eduardo Rodríguez Arias

Junio de 2011

## DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores de este trabajo y autorizamos al Departamento de Soluciones para Aduana del CEIGE de la Facultad 3 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

Juan Ernesto Borggiano Mikulenko

Wilber Ernesto Ramirez Verdecia

\_\_\_\_\_

\_\_\_\_\_

Ing. Alain Eduardo Rodríguez Arias

\_\_\_\_\_

## **DATOS DE CONTACTO**

Tutor: Ing. Alain Eduardo Rodríguez Arias  
Ingeniero Informático.  
Graduado en el 2004 en la Instituto Superior Politécnico “José Antonio Echeverría”.  
Asistente de la Facultad 3 en la UCI.  
Jefe de Departamento de Soluciones para la Aduana, CEIGE.  
Correo Electrónico: [arod@uci.cu](mailto:arod@uci.cu)

## **AGRADECIMIENTOS**

Quiero agradecer primeramente a mis padres por su apoyo y esfuerzo para que tuviera lo mejor y aprendiera a superarme y a lograr mis metas en la vida, a ellos por la formación que me han brindado y por dejarme ser yo.

A mi hermano por su apoyo y preocupación, por ser a su forma un ejemplo para mí aunque más nadie lo crea así, por ser aparte de hermano amigo.

A mi familia en general por el apoyo en los momentos necesarios y por siempre preocuparse por mí.

A mis amigos más cercanos (mykore) de la universidad: Lian (mi segundo hermano), Claudia (mi hermana menor), Elizabeth, Lauren, Naydeline (Kira), Miguel (kuko) y Daniela (el apoyo que me faltaba y que por suerte me encontró en 5to año, mi vasito de agua), que me han apoyado en los buenos y malos momentos y que a su forma me han ayudado a resistir esta maratón de 5 años de carrera y ser mi otra familia.

A mi amigo Mario Merino, por ser como un profesor para mí y con su ayuda ampliar mis conocimientos y logros en la carrera.

A mi tutor y amigo Alain Eduardo, por ser la persona que es, por ser tan buen profesor, por prepararme y por el apoyo y confianza que depositó en mí.

A todos mis compañeros de aula, por formar parte de las vivencias en la universidad, que ha sido de alguna forma como mi segunda casa.

A mis profesores, sobre todo los de la primaria que fueron los que más influyeron y más valores formaron en mí.

Juan Ernesto Borggiano Mikulenko

Primeramente agradecerles a mis padres por su apoyo, que aunque estuvieron tan lejos todo este tiempo siempre los sentí bien cerca de mí, por la educación que me dieron y haberme enseñado desde pequeño el camino correcto que me permiten ser como soy.

Agradecerle a toda mi familia, sin excepciones, que siempre me apoyaron y se preocuparon por mí.

A mi novia agradecerle por cómo se ha portado conmigo y por haberme acompañado todo este tiempo en los buenos y malos momentos, y a su familia que los quiero y que son mi familia también.

Agradecerles mucho a los amigos que son hermanos para mí, ya que no tengo ninguno de sangre. A los que son hermanitos míos desde pequeño y que lo seguirán siendo para toda la vida, (Alexander, Isidro, Orestes, Mayo, Merconchini), A todos los compañeros de mi grupo del IPVCE, que fuimos en esos 3 años una familia y lo seguimos siendo. Por último, a todos los amigos nuevos que hice en la universidad, pero principalmente a los que me acompañaron bien de cerca estos 5 años en todos los momentos, que se ganaron todos mis respetos y cariño, no los quiero mencionar para que no se me quede ninguno porque son muchos, pero ellos saben quiénes son, y somos conocidos como (La Banda).

Agradecer a mi tutor y profesor Alain el cual para mí es el mejor profesor de esta universidad, tanto por sus conocimientos como por su trato y su forma de ser con los estudiantes, por lo que se ha ganado el respeto y aprecio de todos.

Agradecer, ya que tuve la suerte de contar con ellos, a todos los profesores y maestros que influyeron en mi formación desde la primaria, secundaria, y el IPVCE de Las Tunas que cuenta con un claustro de profesores impresionante.

Wilber Ernesto Ramírez Verdecia

## **DEDICATORIA**

A mi familia completa, tanto los que están como los que no, a los más cercanos y los más lejanos, sobre todo, especialmente a mis padres por mi formación y además por haber cargado conmigo todos estos años, así como a mí hermano. A mis amigos más cercanos (mykore) que también de alguna forma me han apoyado.

Juan Ernesto Borggiano Mikulenko

A toda mi familia, especialmente a mi abuelo Dago, a mis padres, que este título es lo que más esperaban de mí, a mi novia Liudmi y a todos mis amigos.

Wilber Ernesto Ramírez Verdecia

## RESUMEN

La Aduana General de la República (AGR) es el órgano encargado de dirigir, ejecutar y controlar la aplicación de la política del Estado en materia aduanera. Dada la rapidez de los cambios en el comercio internacional, en la tecnología de la información y la globalización, las instituciones aduaneras se han visto obligadas a modificar sus prácticas. En respuesta a esta evolución la Aduana Cubana está modernizándose para prestar un servicio eficaz y eficiente, aprovechando los avances tecnológicos en la informática y en las comunicaciones. Se desarrolló el Sistema Único de Aduanas (SUA) por especialistas del propio organismo, y actualmente está siendo desarrollado el producto Gestión Integral de Aduanas (GINA) en colaboración con la UCI. Este sistema en desarrollo ha presentado varias dificultades relativas sobre todo a la capacidad de responder a los constantes cambios en los procesos a automatizar. En el presente trabajo se brinda una solución alternativa para aminorar esta situación. Consiste en la creación de un núcleo de aplicación para la gestión de las operaciones aduaneras. El mismo está desarrollado en Python 2.6.6 con base de datos sobre PostgreSQL 9.0.1, cumpliendo con las tendencias de los sistemas más representativos existentes y con las políticas del país del desarrollo de software libre. Esta solución proporciona una mejora en las condiciones de desarrollo y configuración de los sistemas, brindándole al usuario una escalabilidad y acoplamiento de los módulos más fluido y amigable.

Palabras claves: Aduana, Núcleo, Operaciones Aduaneras

# ÍNDICE

<b>INTRODUCCIÓN</b> .....	1
<b>CAPÍTULO 1: ESTADO DEL ARTE</b> .....	4
1.1 INTRODUCCIÓN.....	4
1.2 SOLUCIONES INFORMÁTICAS Y SU CAPACIDAD DE ADAPTACIÓN AL CAMBIO ORGANIZACIONAL.....	4
1.2.1 Cambio organizacional.....	4
1.2.2 Soluciones informáticas adaptables al cambio.....	4
1.2.2.1 Arquitectura Orientada a Servicios.....	4
1.2.2.2 Software AG webMethods.....	5
1.2.2.3 Oracle SOA Suite.....	6
1.2.3 Características relevantes de las soluciones adaptables al cambio.....	6
1.3 SISTEMAS DE GESTIÓN ADUANERA.....	7
1.3.1 Ámbito Internacional.....	7
1.3.1.1 SIDUNEA WORLD.....	7
1.3.1.2 Korean Custom System.....	9
1.3.1.3 US Customs Automated Systems.....	9
1.3.1.4 UK HM Revenue & Custom CHIEF System.....	13
1.3.1.5 ATLAS German System.....	14
1.3.2 Ámbito Nacional.....	14
1.3.2.1 SUA.....	14
1.3.2.2 GINA.....	15
1.3.3 Características relevantes de las soluciones de gestión aduanera.....	15
1.4 TECNOLOGÍA.....	16
1.4.1 Diseño de Software.....	16
1.4.1.1 Modelo de desarrollo a utilizar.....	16
1.4.1.2 Diagrama de Interacción y Actividades.....	16
1.4.2 Patrones de Diseño de Software.....	17
1.4.3 Modelo de Datos Entidad-Atributo-Valor.....	20
1.4.4 Lenguajes de Programación.....	20
1.4.4.1 C++.....	20
1.4.4.2 Java.....	21
1.4.4.3 Python.....	22
1.4.5 Sistema Gestores de Base de Datos.....	22
1.4.5.1 MySQL.....	22
1.4.5.2 Oracle.....	23
1.4.5.3 PostgreSQL.....	24
1.4.6 Visual Paradigm. Herramienta CASE.....	24
1.4.7 ER\Studio. Herramienta para modelado de Base de Datos.....	25
1.5 CONCLUSIONES PARCIALES.....	25
<b>CAPÍTULO 2: DISEÑO DE LA SOLUCIÓN</b> .....	26
2.1 INTRODUCCIÓN.....	26
2.2 PROPUESTA DE SOLUCIÓN.....	26
2.3 REQUISITOS DEL ANÁLISIS.....	27
2.4 DIAGRAMAS DE DISEÑO.....	29
2.4.1 Diagrama de Clases.....	29
2.4.2 Modelo de Datos.....	36

2.4.3	Diagramas de Interacción y Actividades .....	37
2.5	DIAGRAMA DE PAQUETES .....	39
2.6	DIAGRAMA DE DESPLIEGUE .....	40
2.7	CONVENCIONES DE CODIFICACIÓN .....	40
2.8	CONCLUSIONES PARCIALES .....	42
<b>CAPÍTULO 3: VALIDACIÓN DE LA SOLUCIÓN .....</b>		<b>43</b>
3.1	INTRODUCCIÓN .....	43
3.2	MÉTRICAS PARA EL MODELO DE DISEÑO ORIENTADO A OBJETOS .....	43
3.2.1	Métricas orientadas a clases. ....	43
3.2.2	Métricas propuestas por Chidamber y Kemerer. ....	43
3.2.2.1	Aplicación de las métricas. ....	44
3.2.3	Métricas propuestas por Lorenz y Kidd .....	45
3.2.3.1	Aplicación de las métricas. ....	45
3.3	DESCRIPCIÓN DE LAS PRUEBAS .....	48
3.3.1	Aplicación de las pruebas de caja negra .....	48
3.4	CONCLUSIONES PARCIALES .....	48
<b>CONCLUSIONES GENERALES .....</b>		<b>50</b>
<b>RECOMENDACIONES .....</b>		<b>51</b>
<b>BIBLIOGRAFÍA .....</b>		<b>52</b>
<b>REFERENCIAS BIBLIOGRÁFICAS .....</b>		<b>53</b>
<b>ANEXOS .....</b>		<b>54</b>
ANEXO I. DESCRIPCIÓN REQUISITOS FUNCIONALES .....		54
ANEXO II. TABLAS DE DESCRIPCIÓN DE LAS TABLAS DEL MODELO DE DATOS. ....		87
ANEXO III. TABLAS DE DISEÑOS CASOS DE PRUEBAS DE CAJA NEGRA .....		90
ANEXO IV. AVALES .....		98
<b>GLOSARIO .....</b>		<b>100</b>

## ÍNDICE DE TABLAS Y FIGURAS

IMAGEN 1. EJEMPLO DE DIAGRAMA DE INTERACCIÓN Y ACTIVIDADES.....	17
IMAGEN 2. REPRESENTACIÓN DEL PATRÓN FACADE.....	18
IMAGEN 3. REPRESENTACIÓN DEL PATRÓN BUILDER.....	19
IMAGEN 4. ESTRUCTURA GENERAL DE LA SOLUCIÓN.....	27
IMAGEN 5. DIAGRAMA DE CLASES DE DISEÑO.....	29
IMAGEN 6. DIAGRAMA ENTIDAD RELACIÓN DEL ESQUEMA DE CONFIGURACIÓN DE LA BASE DE DATOS.....	36
IMAGEN 7. DIAGRAMA DE INTERACCIÓN Y ACTIVIDADES GENERAL DE LA CONFIGURACIÓN DE UN PROCESO.....	37
IMAGEN 8. DIAGRAMA DE INTERACCIÓN Y ACTIVIDADES GENERAL DE LA EJECUCIÓN DE UN PROCESO.....	38
IMAGEN 9. DIAGRAMA DE PAQUETES.....	39
IMAGEN 10. DIAGRAMA DE DESPLIEGUE DE LA SOLUCIÓN.....	40
IMAGEN 11. GRAFICA DISTRIBUCIÓN DE MÉTODOS POR CLASES.....	47
IMAGEN 12. GRÁFICA PORCIENTOS DE CLASES CON CANTIDAD DE MÉTODOS.....	47
IMAGEN 13. GRÁFICA AFECTACIÓN DE LA MÉTRICA TOC EN LOS ATRIBUTOS QUE SE MUESTRAN.....	48
AVAL I.....	98
AVAL II.....	99
TABLA 1. DESCRIPCIÓN CLASE FACADE.....	30
TABLA 2. DESCRIPCIÓN CLASE CONNECTION.....	30
TABLA 3. DESCRIPCIÓN CLASE BUILDER.....	31
TABLA 4. DESCRIPCIÓN CLASE DESCRIPTOR.....	32
TABLA 5. DESCRIPCIÓN CLASE PROCESS.....	32
TABLA 6. DESCRIPCIÓN CLASE ACTION.....	33
TABLA 7. DESCRIPCIÓN CLASE ENTITY.....	34
TABLA 8. DESCRIPCIÓN CLASE RELATION.....	34
TABLA 9. DESCRIPCIÓN CLASE ATTRIBUTE.....	35
TABLA 10. RESULTADO DE LAS MÉTRICAS CHIDAMBER Y KEREMER.....	44
TABLA 11. RESULTADO DE LAS MÉTRICAS LORENZ Y KIDD.....	46
TABLA 12. DATOS MÉTRICA TOC.....	46
TABLA DE LISTADO DE LOS REQUISITOS FUNCIONALES.....	54
ANEXO II .TABLA 1.....	87
ANEXO II .TABLA 2.....	87
ANEXO II .TABLA 3.....	87
ANEXO II .TABLA 4.....	87
ANEXO II .TABLA 5.....	88
ANEXO II .TABLA 6.....	88
ANEXO II .TABLA 7.....	88
ANEXO II .TABLA 8.....	89
ANEXO II .TABLA 9.....	89
ANEXO II .TABLA 10.....	89
TABLA DESCRIPCIÓN DE LAS SECCIONES Y ESCENARIOS DEL CASO DE PRUEBA.....	90
TABLA DESCRIPCIÓN DE LAS VARIABLES.....	90
TABLA MATRIZ DE DATOS: CASO DE PRUEBA DEL REQUISITO GESTIONAR OBJETO.....	91

## INTRODUCCIÓN

En nuestro país la Aduana General de la República (AGR) es la institución rectora a la que pertenecen todas las aduanas de nuestro territorio, ya sean de vía marítima, aérea o postal. Sus funciones se enmarcan en:

- Dirigir, ejecutar y controlar la aplicación de la política del Estado en materia aduanera.
- Recaudar los aranceles de aduana.
- Responder dentro de su jurisdicción y competencia a los hechos que incidan en el tráfico internacional postal, de mercancías, de viajeros, y de los medios que los transportan, previniendo, detectando y enfrentando el fraude y el contrabando.

Dada la rapidez de los cambios ocurridos en el comercio internacional, en la tecnología de la información y la globalización, las instituciones aduaneras se han visto obligadas a modificar sus regímenes y prácticas para lograr estar a la par de los nuevos adelantos, y la cubana no se queda exenta a esta realidad. En respuesta a esta evolución la AGR se ha planteado ser una aduana moderna a tono con los desafíos actuales del comercio internacional y lograr la prestación de un servicio eficaz y eficiente. Debido a esta situación desde finales del año 90 se ha planteado una mayor utilización de los avances de la informática y de las comunicaciones y otros adelantos científico-técnicos.

En 1994 se comenzó a emplear el Sistema Aduanero Automatizado (SIDUNEA) creado por la Conferencia de Naciones Unidas para el Comercio y Desarrollo (UNCTAD por sus siglas en inglés). El uso de dicho sistema trajo consigo la creación de una cultura informática en la institución, además de la automatización de los procesos internos de la aduana mejorando su eficiencia.

En 1996 a partir de estudios realizados se decide desarrollar el Sistema Automatizado de Despacho Mercantil (SADEM) en base al sistema que se empleaba pero ajustado a las características del comercio cubano.

El sistema SADEM una vez desarrollado comenzó a presentar un grupo de dificultades. Algunas se detallan a continuación:

- Inflexibilidad a causa del mayoritario y dependiente uso del teclado ante el ratón,
- Existencia de opciones sin automatizar.
- Era un sistema multiusuario que trabajaba directamente sobre la base de datos, lo que implicaba riesgos de seguridad, recarga del trabajo del servidor y no permitía la conexión de muchos usuarios.

Por estas razones se decidió que era necesario desarrollar un sistema automatizado, que concretamente, integrara en un sistema y base de datos única a todos los procesos que se llevan a cabo, y que se implantase en todas las aduanas del país. Este sistema debía responder a los intereses de la aduana cubana y en el cual todo funcionario tuviera sus requerimientos satisfechos surgiendo así el Sistema Único de Aduanas (SUA).

El SUA es el sistema que actualmente se encuentra en explotación en todas las aduanas del país. Su desarrollo en PHP estructurado le impone ciertas limitaciones en cuanto a escalabilidad y mantenimiento.

En esta coyuntura surge la Universidad de las Ciencias Informáticas (UCI) con el consecuente impulso al desarrollo de software en todo el país. La AGR fue de las primeras entidades que se asoció a la UCI en el objetivo de informatizar la sociedad cubana. A principio del nuevo siglo por acuerdo de ambas partes se decide desarrollar un nuevo producto conocido como Gestión Integral de Aduanas (GINA) que paulatinamente sustituirá en su totalidad al SUA.

En base a las experiencias alcanzadas hasta el momento con el desarrollo del producto GINA del departamento de desarrollo de Soluciones Aduaneras y en el despliegue de algunos de sus módulos integrados al SUA o como sistemas independientes, se han detectado diferentes fenómenos que traen dificultades con respecto a la integración, despliegue y mantenimiento de los módulos del sistema que tienen gran relación con los procesos y entidades principales del negocio, así como al lograr una fluida configuración al ser desplegado, los mismos son los siguientes:

- Dificultades en el proceso de desplegar los módulos, debido al gran nivel de interdependencia entre ellos.
- La existencia de entidades de negocio que interactúan con más de un módulo o subsistema lo que hace difícil identificar a cual pertenecen de manera natural.
- Existencia de varias funcionalidades implementadas en subsistemas diferentes con objetivos similares haciendo de cierta forma redundante e innecesaria algunas de ellas.
- Bajo rendimiento relativo a la existencia de un exceso de servicios internos y de intercomunicación entre los módulos y el sistema.
- Se complica el diseño de los modelos de datos al momento de decidir en cual esquema de la base de datos ubicar una entidad que es utilizada por más de un módulo.

Una complicación adicional viene dada por la estabilidad en la definición de los procesos que rigen el funcionamiento de la AGR. En su afán por ser cada día más eficiente y minimizar el tiempo en las operaciones sin perder en la calidad del control, la dirección de la aduana cubana emite nuevas reglamentaciones y decretos que afectan la secuencia de los procesos, crean nuevos requerimientos de información a entregar, cambian los formatos de los documentos rectores y complementarios. Esta inestabilidad hace que sea prácticamente imposible tener un sistema desplegado el tiempo suficiente como para estabilizarlo y poder asegurar que resuelve todas las necesidades de informatización de los órganos aduaneros de nuestro país.

Por todo lo anteriormente expresado se define el siguiente **problema a resolver**:

El dinamismo en la definición de los procesos aduaneros cubanos así como la interdependencia entre los mismos ha impedido la estabilización de una solución informática para la gestión de la AGR.

Se puede definir como **objeto de estudio** de esta investigación:

Soluciones informáticas para entornos dinámicos y con interdependencia entre sus procesos.

Específicamente la investigación centrará su **campo de acción** en:

Soluciones para la gestión de aduanas en entornos dinámicos.

El **objetivo general** que persigue el presente trabajo es:

Desarrollar una solución informática que sea capaz de modelar dinámicamente los procesos de gestión de la AGR.

Para facilitar el exitoso cumplimiento del objetivo planteado se realiza el siguiente desglose en **objetivos específicos**:

- Realizar un estudio que permita definir qué tipo de solución informática ayudará mejor a la solución del problema así como las tecnologías a emplear en el desarrollo de la misma.
- Diseñar e implementar la solución informática definida.
- Validar la solución desarrollada.

#### **Tareas de investigación:**

1. Obtención de información de las principales funcionalidades de los sistemas de gestión aduanera existentes tanto nacional como internacionalmente.
2. Estudio del estado del arte de las soluciones informáticas diseñadas para asimilar cambios drásticos en las organizaciones.
3. Estudio y selección de las tecnologías posibles a emplear en el desarrollo de la solución.
4. Formalización del tipo de solución a desarrollar y las características que debe tener.
5. Obtención del diseño de la solución informática formalizándolo a través de los artefactos y diagramas definidos en el Departamento de Soluciones para Aduana del Centro para la Informatización de las Gestión de Entidades (CEIGE).
6. Validación del diseño propuesto.
7. Implementación de la solución diseñada utilizando las tecnologías definidas.
8. Diseño de las pruebas de caja negra que validen la solución implementada.
9. Aplicación de las pruebas diseñadas en la validación de la solución implementada.

En el **Capítulo 1** se ofrecerá una panorámica sobre las tendencias actuales del diseño de soluciones informáticas con capacidad de adaptación al cambio organizacional. Además se identificarán las características deseables en un sistema de gestión aduanero. Se definirá el tipo de solución a desarrollar así como los artefactos a emplear en su diseño. Se identificarán patrones de diseño que ayuden y faciliten la implementación exitosa del sistema así como también las tecnologías más adecuadas para el desarrollo de la aplicación.

En el **Capítulo 2** se presentarán, haciendo uso de los diagramas y artefactos establecidos, las características específicas del sistema a desarrollar, así como una descripción del proceso de implementación del mismo.

En el **Capítulo 3** se realizará la validación del diseño de solución propuesto mediante la aplicación de métricas para validar diseños basadas en características de las clases. Además se realizará la validación funcional del sistema con pruebas de caja negra aplicadas a las funcionalidades.

## **1.1 Introducción.**

El presente capítulo ofrece una panorámica y breve descripción de los principales conceptos asociados al dominio del problema existente en el actual sistema en desarrollo para la aduana. Son identificadas las principales tendencias de las soluciones aduaneras, que fundamentan la propuesta de solución. Así como las tecnologías, herramientas y conceptos de desarrollo de software actuales y determinando las seleccionadas a emplear para la implementación del sistema. Se formulan las ventajas y desventajas de cada elemento analizado.

## **1.2 Soluciones informáticas y su capacidad de adaptación al cambio organizacional.**

### **1.2.1 Cambio organizacional**

Un cambio organizacional es cualquier conjunto de acciones que resulten en un re-direccionamiento de los procesos de manera que afecten la forma de trabajo de una organización. Estos cambios pueden ser deliberadamente planificados por los líderes de la organización o pueden producirse fuera de la misma y no existe control sobre ellos.

Los cambios pueden afectar las estrategias de una organización para cumplir su misión, los procesos que implementan estas estrategias, las tareas y funciones llevadas a cabo por las personas de la organización y las relaciones entre esas personas. Algunos cambios son relativamente pequeños mientras resultan en un cambio radical de la organización. (1)

### **1.2.2 Soluciones informáticas adaptables al cambio**

Si para una organización es complicada la gestión de cualquier cambio por pequeño que este sea, es mucho más complicado para una empresa mantener sus sistemas informáticos adecuados a todas las modificaciones organizacionales que puedan ocurrir.

El ideal de una solución informática que se utiliza en un entorno cambiante sería un alto grado de configuración de los procesos que simula y soporta. No solamente a nivel de nomencladores sino que también sea capaz de almacenar las descripciones de los procesos y la forma en que se orquestan unos con otros. En este aspecto en los últimos años se ha desarrollado el concepto de Arquitectura Orientada a Servicios (SOA por sus siglas en inglés).

#### **1.2.2.1 Arquitectura Orientada a Servicios**

La Arquitectura Orientada a Servicios es un concepto de arquitectura de software que define la utilización de servicios para dar soporte a los requisitos del negocio.

Permite la creación de sistemas altamente escalables que reflejan el negocio de la organización, a su vez brinda una forma bien definida de exposición e invocación de servicios (comúnmente pero no exclusivamente servicios web), lo cual facilita la interacción entre diferentes sistemas propios o de terceros.

SOA define las siguientes capas de software:

- **Aplicaciones básicas** - Sistemas desarrollados bajo cualquier arquitectura o tecnología, geográficamente dispersos y bajo cualquier figura de propiedad;
- **De exposición de funcionalidades** - Donde las funcionalidades de la capa aplicativa son expuestas en forma de servicios (generalmente como servicios web);
- **De integración de servicios** - Facilitan el intercambio de datos entre elementos de la capa aplicativa orientada a procesos empresariales internos o en colaboración;
- **De composición de procesos** - Que define el proceso en términos del negocio y sus necesidades, y que varía en función del negocio;
- **De entrega** - donde los servicios son desplegados a los usuarios finales. (2)

SOA proporciona una metodología y un marco de trabajo para documentar las capacidades de negocio y puede dar soporte a las actividades de integración y consolidación.

Los beneficios que puede obtener una organización que adopte SOA son:

- Mejora en los tiempos de realización de cambios en procesos.
- Facilidad para evolucionar a modelos de negocios basados en tercerización.
- Facilidad para abordar modelos de negocios basados en colaboración con otros entes.
- Poder para reemplazar elementos de la capa aplicativa SOA sin interrupción en el proceso de negocio
- Facilidad para la integración de tecnologías disímiles.

### 1.2.2.2 Software AG webMethods

webMethods es un conjunto de productos ofrecidos por Software AG para la mejora de procesos, la habilitación SOA, la modernización de TI y la integración de negocio y socios. Brinda una amplia gama de funcionalidades y facilidades para la administración, control y configuración de los procesos del negocio y de los servicios para el consumo y publicación de estos. Dicha suite de productos brinda un conjunto de facilidades tales como:

- alinea las operaciones de TI con los imperativos de negocio, gracias a la interoperabilidad del modelo de procesos;
- da respuesta rápida a los clientes y socios con la arquitectura orientada a eventos de webMethods;
- simplifica el despliegue y actualización capacidades de solución avanzada al despliegue;
- simplifica la gestión de cambios para acelerar los proyectos para un retorno de la inversión más rápido;
- explota la arquitectura orientada a web (WOA por sus siglas en inglés) usando servicios para construir rápidamente mashups y aplicaciones web y móviles. (3)

A nivel de infraestructura SOA presenta un grupo de funcionalidades y facilidades como:

- construir servicios de acuerdo a las necesidades del negocio a través de la administración de la cartera SOA, gracias a la integración de CentraSite y ARIS IT Architect ambos productos de la misma empresa incluidos en la suite;
- monitorear las transacciones de servicios en webMethods Broker.

### 1.2.2.3 Oracle SOA Suite

Oracle SOA Suite es un conjunto de software completo y con funcionamiento permanente para la creación, implementación y administración de una arquitectura orientada a servicios. Esto incluye el desarrollo de aplicaciones orientadas a servicios, la integración de sistemas de Tecnologías de la Información (TI) y aplicaciones orientadas a servicios y la administración de procesos de negocio orientados a servicios. Se conecta a las infraestructuras de TI heterogéneas y permite a las empresas adoptar SOA de manera gradual.

Los componentes de la suite se benefician con capacidades en común, con inclusión de un solo modelo de administración e implementación, herramientas consistentes, seguridad integral y administración de metadatos unificados.

Oracle SOA Suite mejora la capacidad de la empresa tanto para predecir los cambios como para responder a esos cambios, permitiendo a las empresas desarrollar y optimizar los procesos de negocio rápidamente y mejorando la visibilidad de lo que ocurre en el entorno de los negocios en tiempo real. Simplifica el entorno de TI al ser abastecido, implementado, monitoreado y administrado como una sola infraestructura cohesiva. Aprovecha las inversiones existentes al ser modular, abierto y extensible.

Oracle SOA Suite consta de los siguientes componentes:

- Un administrador de procesos basado en Lenguaje de Ejecución de Proceso de Negocio (BPEL por sus siglas en inglés) para componer servicios en los procesos de negocio.
- Una solución para el monitoreo de la actividad de los negocios a fin de obtener visibilidad en tiempo real de las operaciones y el desempeño de los servicios y procesos de negocio.
- Un motor de reglas de negocio para capturar y automatizar las políticas de negocios.
- Conectividad a prácticamente todas las fuentes de datos, con inclusión de las aplicaciones, bases de datos, colas, dispositivos de Identificación por Radiofrecuencia (RFID por sus siglas en inglés) y otros dispositivos físicos, así como la integración de datos de gran volumen y de alto desempeño.
- Oracle JDeveloper, un Entorno de Desarrollo Integrado (IDE por sus siglas en inglés) para administrar, depurar, elaborar perfiles e implementar servicios.
- Una solución de seguridad y administración de servicios web para hacer cumplir las políticas de autenticación y autorización en torno a los servicios.
- Un registro de servicios para detectar y administrar el ciclo de vida de los servicios. (4)

### 1.2.3 Características relevantes de las soluciones adaptables al cambio

Los enfoques y sistemas revisados arrojan que las soluciones existentes en el mercado que son capaces de soportar cambios en los proceso de negocio tienen un grupo de características comunes más allá de la definición tecnológica y los elementos de software que las componen. Estos elementos se resumen a continuación y además son los deseados para la solución que se pretende implementar:

- Procesos como servicios administrados: Este tipo de solución identifica los procesos de negocio como entidades que pueden cambiar en el tiempo, que es necesario administrar y que además tienen la capacidad de ser publicados para que sus resultados puedan ser explotados por otros sistemas.
- Independencia de la fuente de datos: Las soluciones deben ser capaces de interactuar con diferentes suministradores de información sin importar el formato o la vía de obtención.

- Punto común de acceso a los servicios: Un elemento de software que sirve como punto de conexión a donde todos los usuarios de los servicios deben conectarse a solicitar la ejecución de uno de los procesos publicados.
- Independencia de los clientes: Las soluciones de este tipo no se responsabilizan por la forma en que los clientes acceden a la información que los servicios brindan ni como la muestran a los usuarios finales.

### **1.3 Sistemas de Gestión Aduanera.**

#### **1.3.1 Ámbito Internacional.**

En la actualidad, en el ámbito internacional, existen una variedad de sistemas informáticos de gestión de los procesos de las aduanas. Entre ellos se destacan los 6 siguientes:

1. Sistema Aduanero Automatizado (SIDUNEA WORLD) - Conferencia de Naciones Unidas para el Comercio y Desarrollo.
2. Sistema Aduana de Corea (KCS por sus siglas en inglés) – Corea.
3. Sistemas Automatizados de Aduanas (US Customs Automated Systems) – Estados Unidos de América
4. UK HM Revenue & Custom (HMRC) CHIEF System – Reino Unido
5. Sistema de Automatizado de Aranceles y Despacho de Aduanas Locales (ATLAS por sus siglas en alemán) – Alemania

Los sistemas mencionados se destacan por ser utilizados en los países con las aduanas más avanzadas y desarrolladas del mundo, además de ser sistemas aduaneros con grandes resultados y aceptación.

##### **1.3.1.1 SIDUNEA WORLD.**

El sistema SIDUNEA WORLD fue desarrollado por la Conferencia de Naciones Unidas para el Comercio y Desarrollo (UNCTAD por sus siglas en inglés) para la administración de aduanas. Es un sistema web, que permite vía Internet la administración de aduanas y el manejo de la mayoría de las transacciones con los comerciantes.

Las funcionalidades principales del sistema están diseñadas para:

- Facilitar y mejorar el cálculo, recaudación y contabilización de los aranceles aduanales y otros impuestos relacionados con las operaciones aduaneras.
- Acelerar el despacho de las mercancías y evitar el contrabando.
- Proveer la administración de las aduanas con información precisa y oportuna.

Esto es consumado por:

- Verificaciones automáticas, validación y contabilidad de las declaraciones de aduanas.
- Aplicación uniforme de la ley y de las tasas de aranceles correctas.
- Habilidad de monitorear y seguir los pagos de los aranceles e impuestos.
- Control de valores y descripción de la corrección de bienes a través del uso de valuación en línea y base de datos de tarifas.
- Selección de transacción de riesgos para propósitos de control.
- Procesamiento rápido de documentos y minimización del manejo manual de documentos.

- Implementación de métodos de trabajos sistemáticos y una óptima distribución de las tareas y recursos.

Características generales:

- **Facilitación del comercio**  
Completamente soportada, incluyendo las recomendaciones para Comercio Libre y Seguro de la Organización Mundial de Aduanas (WCO FAST por sus siglas en inglés). Incrementando los servicios para la comunidad de comercio, al emplear el acceso a internet, procedimientos simples y documentación.
- **Aplicación de la Ley**  
Capacidades de control aduanero incrementadas (evaluación de riesgos, acceso en línea a base de datos externas, control de imágenes, funciones de selectividad, etc.)
- **Ingresos y estadísticas**  
Capacidades estadísticas y contables. Desarrollo de capacidades de auditoría después de despachos de aduanas. Administración de estadísticas e información, fácil agregación de datos a niveles regionales e internacionales.
- **Estandarización y normalización**  
Basado en estándares de ISO, ONU, OMA y OMC, base de datos diseñada en cumplimiento de modelo de datos de aduanas de la OMA.
- **E-Gobierno**  
Soporte completo del gobierno electrónico y capacidad para interoperar en línea con sistemas y base de datos externos/gubernamentales. (5)

Características funcionales:

- **Operaciones aduaneras**  
Manejo automatizado de todos los pasos del proceso de despacho de aduanas para todo tipo de procedimientos y regímenes (control de carga, monitoreo de tránsito) aduaneros, así como contabilidad. El sistema asegura un cubrimiento completo del proceso de despacho de aduanas e incluye la capacidad incorporada de soportar requerimientos nacionales específicos y/o cambios de regulación frecuentes. Es capaz de adaptarse a las diferentes estructuras de organización, generando economías financieras de escala.
- **Administración de riesgos**  
Completa evaluación de riesgo y capacidad de selectividad. Nuevas funciones disponibles, tales como: la integración de imágenes o el uso de códigos de barras y el acceso en línea a base de datos externas, todo esto en conexión con WCO FAST e iniciativas de control de contenedores.
- **Valuación**  
Sistema experto de valuación compatible con el Acuerdo General sobre Aranceles Aduaneros y Comercio (GATT por sus siglas en inglés) en su artículo 7. Incluye una declaración de valor de aduana electrónica. Mediante el empleo de dicho sistema se reduce la pérdida de ingresos al introducir el artículo 7 y se aumentan los ingresos generales de la aduana (fácil detección de devaluaciones).
- **Administración de flujo de trabajo**  
Uso de documentos electrónicos y administración del flujo de trabajo de la trayectoria de procesamiento de documento. Oculta la complejidad del desarrollo de aplicaciones de clase gubernamental.

- **Sistema de inspección no intrusivo**

Administración de Firma de imagen de bienes electrónicos (EGIS por sus siglas en inglés) cierra la brecha entre los bienes y documentos de apoyo. Facilita la identificación, inspección y seguimiento de las mercancías. (5)

### **1.3.1.2 Korean Custom System.**

KCS actualmente está operando un total de 58 sistemas de información relacionados con la administración de aduanas. El sistema de despacho electrónico que este emplea hace referencia a 7 sistemas, entre los que están los que realizan funciones principales de despacho de exportación/importación en una base consistente. El sistema de despacho electrónico, el producto más avanzado de la informatización de aduanas, ha sido desarrollado a través de una fuerte cooperación con los clientes de aduanas para responder activamente a los cambios en el ámbito del comercio mundial aprovechando las últimas tecnologías de la información. (6)

UNI-PASS, es el nombre comercial del sistema de despacho aduanero electrónico, el cual posee varios significados. Este proporciona un servicio **unificado** para el despacho de las exportaciones/importaciones, pago de aranceles, reintegro de impuestos y confirmaciones de requerimientos, todo esto en una sola parada. Ha adoptado estándares globales para un uso **universal** por parte de todos los países del mundo. Tiene una característica **única** de servicio de seguimiento de mercancías, donde el cliente puede acceder en tiempo real a la información del procesamiento de la mercancía, un servicio que no es brindado en ninguna otra parte del mundo. "PASS" expresa la rapidez del despacho aduanero, también encierra el significado de conexión por parte de los clientes desde cualquier parte, flujo de mercancías continuo, sistema adaptable y la importancia de la participación del cliente en el desarrollo y mejora de dicho sistema.

El sistema UNI-PASS para un trabajo más eficiente, rápido e integral se relaciona estrechamente con las aerolíneas, las agencias del gobierno, las compañías de comercio, agentes de aduanas, bancos, agencias de aprobación, oficinas de inspección/cuarentena, compañías de embarque, cargadores, intermediarios, transportistas y depósitos. Esto lo logra a través del portal del sistema de despacho en internet, haciendo uso de Intercambio Electrónicos de Datos (EDI por sus siglas en inglés) con el sistema automatizado EDI de despacho aduanero. A través de este portal también se posee una interconexión con las redes de logística nacional, las redes de finanzas y las compañías de logística, importación y exportación.

Con el uso del sistema de seguimiento en tiempo real de la mercancía se tiene acceso a toda la información del proceso (notificación de arribo, entrada, descarga, cadena de transportación, almacenamiento, despacho de importación y liberación) desde cualquier parte del mundo vía internet, accesible desde cualquier dispositivo, asegurando una mayor comunicación con el cliente y una mejor planificación.

### **1.3.1.3 US Customs Automated Systems.**

Las aduanas de los Estados Unidos de América funcionan bajo los Sistemas Automatizados de Aduanas los cuales son: Sistema de Exportación Automatizado (AES por sus siglas en inglés), Interfaz de Agente Comercial Automatizado (ABI por sus siglas en inglés), Sistema de Manifiesto Automatizado (AMS por sus siglas en inglés) y Sistema Comercial Automatizado (ACS por sus siglas en inglés), entre otros.

- **AES**

Este sistema es una empresa conjunta entre el Departamento de Protección de Aduanas y Fronteras (CBP por sus siglas en inglés), División de Comercio Exterior de la Oficina del Censo, la Oficina de Industria y Seguridad, la Dirección de Defensa y Control Comercial, otras agencias federales y la comunidad comercial de exportación. Es el punto central a través del cual los datos de embarques de exportación requeridos por agencias múltiples son archivados electrónicamente por la Aduana, usando las eficiencias de EDI. El sistema proporciona una alternativa a la presentación en papel de las Declaraciones de Exportación del Transportista (SED por sus siglas en inglés). La información de exportación es recogida electrónicamente y editada inmediatamente, y los errores son detectados y corregidos al momento de ser llenados. El AES es un sistema usado en toda la nación, operacional en todos los puertos y para todos los medios de transporte. Fue diseñado para asegurar el cumplimiento y aplicación de las leyes relativas a la exportación, mejorar las estadísticas de comercio, reducir los reportes duplicados para las múltiples agencias y mejorar el servicio aduanero.

- **ABI**

Es el componente de Sistema de Servicio Comercial Automatizado de las aduanas norteamericanas que permite al personal calificado archivar electrónicamente los datos de importación requeridos en las aduanas. ABI es un programa voluntario disponible para corredores, importadores, transportistas, autoridades portuarias y centros de servicio independientes.

Características:

- Liberación de mercancías

Esta funcionalidad está disponible para los archivadores de resumen de entrada operacionales cuando estos datos no están disponibles o esa información no se desea. Esta información puede ser transmitida previa a la llegada de la mercancía. Los errores de los datos son corregidos a través del sistema antes de que el resumen de entradas haya sido archivado en las aduanas.

- One Stop Quota

Los solicitantes con la capacidad para transmitir ciertas entradas a través de ABI son elegidos para One Stop Quota. En los lugares donde esta opción está disponible, se le permitirá al solicitante archivar dichos documentos directamente con un equipo asignado de especialistas en importaciones. Cada equipo ha sido especialmente entrenado en el proceso de entrada de resúmenes. La One Stop Quota reduce grandemente el tiempo de respuesta desde que presentan los documentos de entradas para el tiempo en que la mercancía ha sido liberada.

- Interfaz de Otras Agencias del Gobierno (OGA Interface por sus siglas en inglés)

La interfaz de ABI para la conexión con otras agencias del gobierno permite la transferencia de datos sobre transacciones de importación. Dicha interfaz elimina la necesidad de hacer copias de los formularios de las otras agencias del gobierno, que de alguna forma son necesarios para las aduanas. Esta interfaz permite a los aplicantes cumplir con los requisitos del Departamento de Transportación, la Oficina del Censo, Comisión Federal de Comunicaciones, Administración de Drogas y Alimentos, Servicio de Pesca y Vida Salvaje de los EE.UU.

- Procesamiento de declaraciones

Esto permite a los aplicantes de ABI pagar múltiples entradas de resúmenes con un cheque o transacción de pago. La aduana genera una declaración preliminar de los dichos resúmenes y transmite la declaración al aplicante a través de ABI. El aplicante imprime y revisa la declaración preliminar, actualiza la declaración con adiciones o supresiones, y envía la declaración final con el pago. Luego que la

declaración ha sido pagada, el aplicante recibe la declaración final de la aduana, a través de ABI, donde se listan todos los resúmenes pagados. La última declaración sirve como comprobante de pago.

El procesamiento de declaraciones reduce el tiempo de procesamiento para la recolección y aceptación de cualquier resumen de entrada. También reduce los costos y riesgos de manipular cheques en papel y ofrece más control sobre el archivado de los resúmenes y el pago de aranceles.

Los aplicantes pueden tener sus resúmenes de entradas agrupados por dos tipos de declaraciones: Declaración de Corredor y Declaración de Importador. Además los aplicantes que usen esta vía son elegibles para participar en la Casa de Pago Automatizada (ACH por sus siglas en inglés).

- Pago Electrónico a través de ACH

ACH es una opción de pago electrónico que permite a los aplicantes de ABI pagar las tarifas aduaneras, aranceles e impuestos con una sola transacción electrónica. Los aplicantes transmiten la autorización de pago a las aduanas a través de ABI. La precisión y velocidad de ACH da como resultado un gran volumen de transacciones completadas, ahorro de tiempo y dinero para las aduanas y los aplicantes.

- Registro y Actualizaciones del Sistema de Armonizado de Tarifas

El registro de tarifas es un elemento esencial de ABI. Empleando este registro se tiene una reducción significativa de errores y rechazos en el procesamiento de resúmenes. El registro es emitido en Diciembre de cada año y contiene los nuevos números de las tarifas programadas y las tasas de impuestos para el año siguiente. Las actualizaciones de los archivos de tarifas están disponibles electrónicamente durante el año mientras estos aparecen. Los usuarios de ABI son notificados a través de mensajes administrativos cuando hay actualizaciones disponibles.

- Mensajes administrativos

Las aduanas pueden comunicarse con los usuarios de ABI de forma individual, por puerto específico o proveer información general a todos los solicitantes a través de mensajes administrativos

- Consulta de Ficheros
  - ✓ Consultar Datos del Importador y depósito.
  - ✓ Consultar archivo de tarifas.
  - ✓ Consultar estadísticas de errores.
  - ✓ Consultar información de Cuota y Visa.
  - ✓ Consultar información de Protesta. (7)
- Extraer archivos de referencia

Los solicitantes de ABI pueden consultar los archivos del Centro de Datos de las Aduanas y descargar los datos siguientes para su sistema:

- ✓ Códigos de transportistas.
- ✓ Códigos de los países (en formato ISO).
- ✓ Códigos de conversión de monedas.
- ✓ Tasas de cambio de monedas.
- ✓ Estadísticas de errores de avisos.
- ✓ Códigos de sistema de administración de recursos e información de instalaciones.
- ✓ Códigos de puertos extranjeros.
- ✓ Asignaciones de equipo de especialistas de importaciones.
- ✓ Información de otras agencias del gobierno.
- ✓ Códigos de puertos.
- ✓ Códigos postales y de estados. (7)

- **AMS**

Este componente es el empleado para el control de Manifiesto y se desglosa en 3 subcomponentes AMS para mar, AMS para aire y AMS para ferrocarriles.

- AMS para mar

Está disponible para transportistas, autoridades portuarias y centros de servicios. Las autoridades portuarias y los centros de servicios pueden archivar y recibir los datos de manifiesto y mercancías de múltiples transportistas llamando a varios puertos.

- AMS para aire

Se emplea para la transmisión de información electrónica de la carga aérea antes de la llegada del vuelo, la cual es recibida por las aduanas y por el transportista aéreo; y de ser posible la información de la carga que trae aprobada por otras fuentes. Para los cargamentos no consolidados, el transportista debe transmitir toda la información para el registro de la hoja de ruta. Para los cargamentos consolidados, el transportista debe transmitir la información de la hoja de ruta principal y este debe transmitir además toda la información referente a la mercancía para todas las instalaciones asociadas a la hoja de ruta, a no ser que la otra parte transmita electrónicamente dicha información.

Los aplicantes de AMS para aire reciben una notificación electrónica del estado de la mercancía tan pronto como CBP (Aduanas y Protección de la Frontera) es notificado de la partida del último aeropuerto extranjero o el vuelo ha sido recibido por CBP.

- AMS para ferrocarriles

Este subsistema permite a los transportistas ferroviarios transmitir electrónicamente la información de factura a las aduanas. Cuando todas las facturas en un tren están asignadas, el transportista transmite un “compuesto” de las facturas y contenedores en el orden en que se encuentran en los vagones. Esto le da a la aduana la oportunidad de revisar la documentación enviada y determinar previamente que mercancía debe examinar o liberar.

Con este sistema se facilita los reportes al hacerse en un proceso sin papeles, también se puede emplear para mandar mensajes de que el tren está en camino o para pedir liberación de la línea y para la petición de permisos para movimientos de tránsito variado por la frontera, todo esto de forma electrónica.

- **ACS**

El sistema ACS para la selección de mercancías se usa para diferenciar las mercancías de alto riesgo de las de bajo riesgo y determinar el tipo de examen requerido. Este sistema acepta datos transmitidos por el sistema ABI y los compara contra un criterio establecido. Esto facilita un proceso de transportación de mercancías más rápido.

Para los resúmenes de entrada el sistema automatiza la revisión de los datos y usando la información transmitida a través de ABI se comprueba contra los criterios de selección nacional y local para evaluar los riesgos por importador, número de tarifa, país de origen, manufacturador y valor.

El sistema de cuotas de ACS sigue los controles de cantidad de la mercancía importada, también las visas de otros países las visas determinan la cantidad de exportaciones permitidas por ciertos países). El sistema comprueba las cantidades contra las visas y transmite esta información a los países de origen, simplificando así las reconciliaciones de las importaciones y exportaciones.

#### 1.3.1.4 UK HM Revenue & Custom CHIEF System.

Las aduanas del Reino Unido funcionan bajo el departamento HMRC, estas emplean el Sistema de Manejo Aduanero de Importación y Exportación de Cargas (CHIEF por sus siglas en inglés) el cual provee los siguientes servicios:

- Acceso directo del comerciante al procesamiento electrónico de las importaciones y exportaciones (incluyendo el cálculo de los aranceles, la moneda y las conversiones de cantidad) y el despacho automático de envíos.
- Identificación de bienes que requieran documentación o examen físico haciendo uso de un sistema de perfiles de riesgo altamente sofisticado.
- Información para la producción de estadísticas de comercio exterior del Reino Unido.
- Un medio de comunicación electrónica entre las aduanas y los negocios de los usuarios.
- Validación de la exactitud de la entrada de datos donde el sistema informa de cualquier error al usuario.
- Almacenamiento, monitoreo y contabilización de los aranceles e impuestos incurridos por importadores individuales.

Como parte de la arquitectura del sistema se puede resaltar que este se conecta con 6 sistemas independientes de comercio que sirven directamente a cientos de transportistas, almacenadores transitorios e intermediarios de mercancías para registrar y seguir el movimiento de los bienes entre los puertos y aeropuertos. Empleando mensajes EDI internamente en el sistema, CHIEF comprueba que los datos en las declaraciones de aduanas coincidan con el inventario mantenido en cada sistema de comercio. Todos estos sistemas notifican a CHIEF tan pronto como los bienes llegan, después de lo cual estos son rápidamente validados, descartados o seleccionados para examen. A pesar del hecho de que todos los sistemas independientes de comercio están basados en diferentes tecnologías, CHIEF maneja cada interfaz sin problemas.

El sistema CHIEF se ejecuta sobre la plataforma Fujitsu Services Super Nova usando el sistema operativo abierto Entorno de Máquina Virtual (VME por sus siglas en inglés). El sistema de base de datos, Sistema Extendido para la Gestión Integrada de Datos (IDMSX por sus siglas en inglés), el cual es jerárquico y para que el sistema de procesamiento de transacciones que abarca la funcionalidad de negocio pueda ser fácilmente mejorado en el futuro, British Telecommunications desarrollo una arquitectura intermedia que provee una interfaz de alto nivel entre el código de la aplicación y el entorno de la máquina. (8)

CHIEF soporta tráfico de Interfaz Humano-Computadora (HCI por sus siglas en inglés) y la interfaz EDI, con conexiones dedicadas de alta velocidad al Proveedor de Servicios de Comunicación (CSP por sus siglas en inglés) y oficinas de aduanas por todo el Reino Unido. Además posee un sistema de seguridad sofisticado y características de resistencia han sido construidas dentro de la arquitectura del sistema para prevenir el sabotaje o acceso no autorizado y proveer continuidad efectiva del negocio.

CHIEF es adaptable a los cambios futuros en los requerimientos de las aduanas y en el ambiente de comercio, tanto en términos de volúmenes o patrones de comercio.

### **1.3.1.5 ATLAS German System.**

Las aduanas alemanas usan ATLAS el cual forma parte del proceso interno de las ciencias de la computación de las aduanas alemanas. Con este sistema las declaraciones y medidas administrativas escritas (ejemplo: notificaciones de impuestos de importación) son reemplazadas por mensajes electrónicos, cumpliendo así con el deber administrativo del despacho aduanero de una forma automatizada, simplificada y acelerada.

Todos los departamentos de las autoridades aduaneras alemanas poseen las tareas necesarias para su funcionamiento en el ATLAS equipado con procedimientos especializados.

Las credenciales son centralmente archivadas y en conformidad con las políticas de privacidad del Centro de Análisis de Riesgos, la Oficina Federal de Estadísticas, la Oficina Federal de Economía y Control de Exportaciones, la Agencia Federal de Agricultura y Alimentación, los servicios de inspección, las oficinas de investigación de aduanas y la administración de impuestos estatales que se han hecho. (9)

El Ministerio Federal de Finanzas provee para la administración de aduanas alemanas con ATLAS las condiciones para el manejo y monitoreo automatizado de los movimientos entre fronteras de los bienes disponibles.

Por parte del aplicante los datos son guardados, transmitidos a la oficina y procesados electrónicamente en consecuencia. El aplicante recibe la decisión de la oficina y la decisión en aranceles de importación y el establecimiento/reconocimiento de las bases también de forma electrónica. El uso de este rápido sistema de procesamiento de aduanas requiere cierto hardware y software, un proveedor de software certificado o un proveedor de servicios que ofrezca diferentes posibilidades de acceso.

El uso óptimo de ATLAS resulta en usar transferencias automatizadas de datos desde el entorno operacional existente, eliminando la reentrada manual de los datos, acelerando el proceso de asentamientos, reduciendo los gastos de mano de obra y los costos, la responsabilidad se incrementará y habrá un aumento en la seguridad de los datos aportados.

### **1.3.2 Ámbito Nacional.**

En el ámbito nacional a partir del sistema informático de gestión aduanera SIDUNEA fueron creados con sus mejoras dos nuevos sistemas, adoptados a las condiciones de Cuba:

1. Sistema Único de Aduanas (SUA)
2. Sistema de Gestión Integral de Aduanas (GINA)

#### **1.3.2.1 SUA**

El Sistema Único de Aduanas es la solución de software cubana para la gestión de los procesos aduanales. Es desarrollado por la Universidad de las Ciencias Informáticas (UCI) en trabajo conjunto con los especialistas del Centro de Automatización de la Dirección y la Información de la AGR (CADI). Este trabajo se inició en el año 2004. El sistema se estableció en el año 2005 como una solución integrada de todos los procesos de la Aduana; dicha solución cuenta con un subsistema Control de Personas, para mejorar el trabajo que se realiza hoy en el flujo de viajeros, otro subsistema para el Despacho MTI, el cual incluye el proceso de tramitación ante la AGR de toda la documentación que declara la carga, provisiones, pasajeros y tripulantes que transporten los MTI.

El sistema puesto en marcha se modeló basado en la metodología estructurada orientada a procesos METRICA3, al igual que su implementación, la cual se realizó en PHP estructurado, sin el empleo de

buenas prácticas de programación o patrones de diseño, tanto arquitectónicos como de aplicación, presentando problemas de mantenimiento, usabilidad y rendimiento. Además de que la información se encuentra de forma fragmentada y comprendiendo muchas interfaces de usuario, por lo que hace el trabajo de los analistas de la AGR un poco engorroso. Tampoco recoge todas las necesidades del departamento de Inspecciones e Investigaciones del área de Lucha Contra el Fraude (LCF).

El sistema es una mejora significativa para la AGR, dado que fue un sistema más completo que el anterior empleado (SADEM) y suplía más prestaciones, además de estar configurado completamente para las necesidades del país. No obstante, le falta madurez y robustez en lo referente al desarrollo e implementación del mismo al no emplear buenas prácticas y técnicas de programación y emplear tecnología que actualmente ya está en desuso, provocando esto problemas de mantenimiento y configuración haciéndolo un sistema arcaico para las necesidades actuales.

### **1.3.2.2 GINA**

Las decisiones que se toman en el seno de la AGR provienen de todas las operaciones aduanales y de los fenómenos que se producen en su entorno organizacional. Estas operaciones y fenómenos se expresan en términos de información. Para gestionar esta información se está desarrollando el Sistema para la Gestión Integral de Aduanas (GINA), el cual constituye un sistema con diversos módulos que se encargan de automatizar y gestionar todo lo relacionado con los procesos y fenómenos aduanales.

El sistema GINA es desarrollado en base a solucionar los problemas presentados en el sistema SUA, se desarrolla también en la UCI en conjunto con el CADI y a diferencia de SUA, GINA 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 pasará a ser una versión mejorada del SUA.

Esta solución es un sistema web que se desarrolla sobre el sistema operativo GNU/Linux. Esta solución es multiplataforma por lo que su funcionamiento puede ser tanto en GNU/Linux como Windows. A su vez son multiplataforma también el lenguaje utilizado PHP, los marcos de trabajo de desarrollo Symfony y ExtJS y el gestor de base de datos Oracle.

El sistema en cuestión sería uno de los pasos más importantes para 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 que cubriría todas las necesidades de la aduana. A pesar de esto está presentando algunos problemas referentes al despliegue y configuración de algunos de sus módulos, principalmente por no presentar un gran desacople entre estos.

### **1.3.3 Características relevantes de las soluciones de gestión aduanera**

Independientemente que las soluciones de gestión aduaneras estudiadas tengan capacidad de adaptación al cambio organizacional o no, se hace necesario hacer un resumen de los elementos más importantes que debe tener toda solución para este tipo de negocio. Se resumen a continuación los más relevantes:

- Recepción de información digital: Todos los sistemas estudiados del ámbito internacional tienen la capacidad de importar y exportar información en formato digital. En algunos países incluso es un requisito indispensable que las empresas realicen las gestiones usando solamente documentos digitales.

- Gestión de importación y exportación de mercancías: Es imprescindible que un sistema de gestión aduanera sea capaz de controlar el flujo de la entrada y salida de mercancías al país y que en su concepción esté incluido el cumplimiento de las normativas de cada país para estos procesos.
- Análisis de riesgo: Aunque no es un requisito obligatorio si se ha hecho común incluir elementos de análisis de riesgo para lograr que los órganos aduaneros estén mejor preparados para enfrentar fraude e ilegalidades.

## **1.4 Tecnología.**

### **1.4.1 Diseño de Software.**

#### **1.4.1.1 Modelo de desarrollo a utilizar.**

Para el desarrollo de esta solución se aplicará el modelo de desarrollo definido por el Centro de Informatización para la Gestión de Entidades (CEIGE) tomando en consideración las adecuaciones realizadas en el departamento de desarrollo de Soluciones Aduaneras específicamente para la fase de diseño. Al aplicar dicho modelo se generarán un conjunto de artefactos para el diseño de la solución, los cuales serán: Diagrama de Clase, Modelo de Datos, Diagrama de Interacción y Actividades, Diagrama de Componente y Diagrama de Despliegue. Por razones asociadas al tiempo disponible para el desarrollo de esta investigación se ha decidido prestar especial atención a los procesos de diseño e implementación de la solución por lo que se obviarán los diagramas y artefactos del proceso de análisis. Se realizará la identificación y descripción de los requisitos funcionales y se saltará directamente hacia el diseño del sistema.

#### **1.4.1.2 Diagrama de Interacción y Actividades.**

El diagrama de Interacción y Actividades surge por la necesidad existente en el Departamento de Soluciones para Aduana del CEIGE de presentar a los desarrolladores una explicación lo más detallada posible de la implementación que les corresponde realizar. Este nivel de claridad no se lograba con los diagramas de secuencia y no se consideró pertinente someter a los desarrolladores a la interpretación de la descripción de los requisitos descritos.

Este diagrama consiste principalmente en la utilización de algunos estereotipos del Diagrama de Actividad definido por el Lenguaje de Modelado Unificado (UML por sus siglas en inglés) donde las calles serían las clases que se involucran en la secuencia que sigue la acción a describir por el diagrama en cuestión. También se incluye la posibilidad de agregar comentarios para una explicación más específica y detallada, aceptando la inclusión de pseudocódigos. A continuación un pequeño ejemplo de muestra:

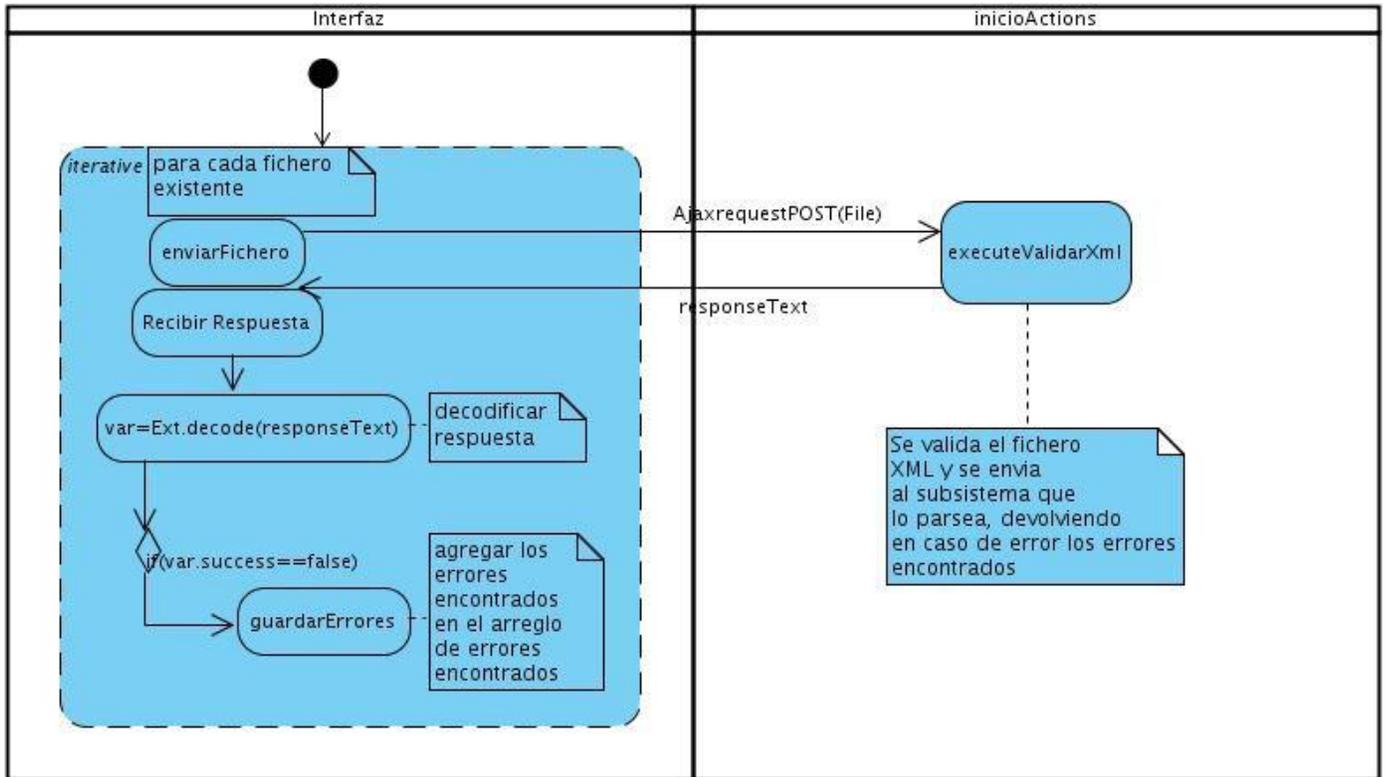


Imagen 1. Ejemplo de Diagrama de Interacción y Actividades.

### 1.4.2 Patrones de Diseño de Software.

En la ingeniería de software, un patrón de diseño es una solución general reusable a un problema que ocurre comúnmente en el diseño de software. Un patrón de diseño no es un diseño terminado que puede transformarse directamente en código. Es una descripción o plantilla de cómo resolver un problema que puede ser usada en muchas situaciones diferentes.

Los patrones de diseño de software que se emplearán en la concepción e implementación de la solución son mayormente patrones Gang of Four (GoF por sus siglas en inglés), los cuales se dividirán por las subcategorías a las que pertenecen.

#### Patrones GoF:

##### Patrones estructurales

- Facade

El patrón Facade es un patrón de diseño de ingeniería de software usado en la programación orientada a objeto. Su nombre se debe a la analogía de fachada arquitectónica.

Facade es un objeto que provee una interfaz simplificada a un bloque de código más grande, como a una librería de clases. (10)

Facade puede:

- hacer una librería de software más fácil de usar, entender y probar, dado que Facade tiene métodos convenientes para tareas comunes;
- hacer el código que usan las librerías más legible, por la misma razón;
- reducir las dependencias de código externo en los trabajos internos de las librerías, dado que mientras más código use Facade permitirá más flexibilidad en el desarrollo del sistema.

La estructura del patrón sería la siguiente:

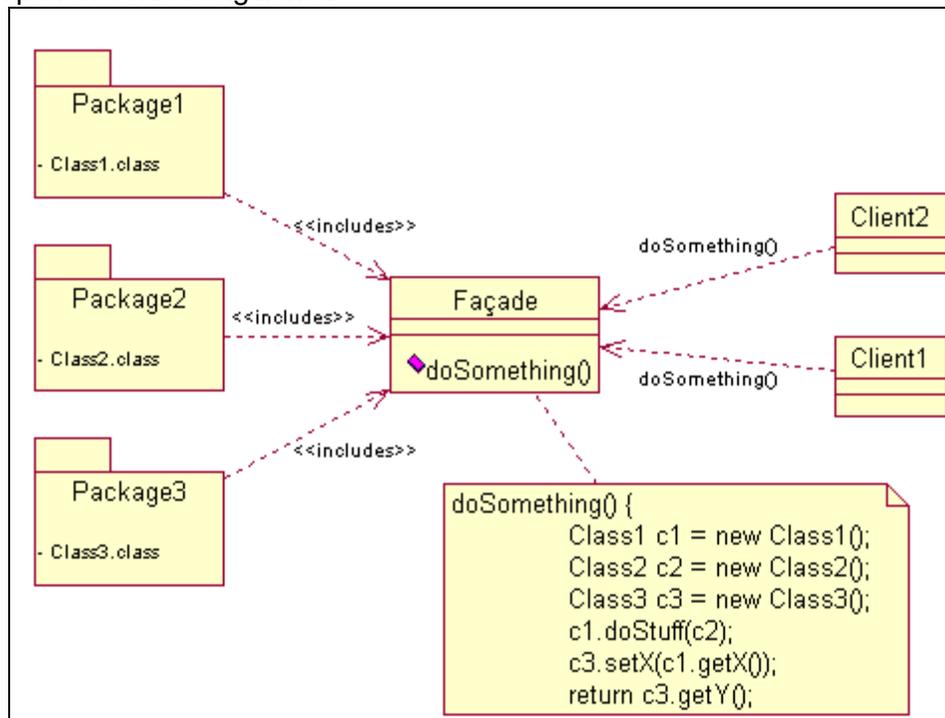


Imagen 2. Representación del patrón Facade.

Donde **Facade** sería la clase Facade que abstrae el paquete 1, 2 y 3 del resto de la aplicación y **Client1** y **Client2** sería los objetos que usan el patrón Facade para acceder a los recursos de los paquetes.

Patrones creacionales

- Builder

El patrón Builder es un patrón de diseño de software, su intención es abstraer los pasos de construcción de objetos, de tal forma que diferentes implementaciones de estos pasos puedan construir diferentes representaciones de los objetos.

Builder se centra en construir objetos complejos paso por paso. Builder puede usar otros patrones a la hora de implementar algún componente en específico en el momento de construirlo, como por ejemplo Singleton. (10)

La estructura del patrón sería la siguiente:

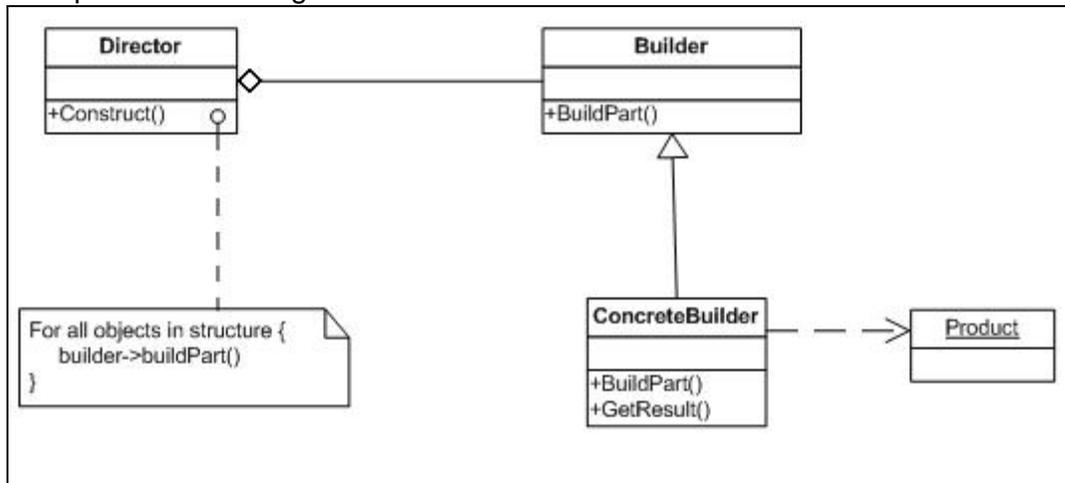


Imagen 3. Representación del patrón Builder.

Dónde:

- La clase Builder hace la función de interfaz abstracta para crear los objetos.
- La clase ConcreteBuilder provee la implementación para Builder. Es un objeto capaz de construir y ensamblar las partes para construir otros objetos.
- La clase Director es la responsable de administrar la secuencia correcta de creación de los objetos. Recibe un ConcreteBuilder como parámetro y ejecuta las operaciones necesarias en él.
- La clase Producto es el objeto final que se creara por Director usando Builder.

#### Otros patrones de diseño utilizados

- Descriptor

Este patrón es empleado y descrito por algunas empresas de desarrollo de software, que en el transcurso de su desarrollo han visto la necesidad de emplearlo, aunque no sea un estándar oficial y para el desarrollo de la aplicación es de utilidad. Ejemplo de dichas empresas son OpenOffice.org y Hudson, ambas pertenecientes actualmente a Oracle. (11) (12)

El descriptor es un tipo de objeto especial que refleja la estructura del objeto que debe ser añadido a un objeto contenedor. Esto significa que el descriptor, una vez creado puede ser añadido más de una vez a la estructura con solo pequeños cambios.

Este patrón es útil a la hora de manejar objetos complejos que poseen atributos o campos simples, debido a que estos "Descriptor" contendrán la información necesaria para construir las instancias complejas y los objetos "Builder" se encargarán de su creación. Esto traería consigo el empleo del patrón Builder el cual se trató previamente en el documento.

### 1.4.3 Modelo de Datos Entidad-Atributo-Valor.

El modelo Entidad-Atributo-Valor (EAV) es un modelo de datos para describir entidades, donde el número de atributos (propiedades, parámetros) que pueden ser empleados para describirlas es potencialmente vasto, pero el número que se aplicará a una entidad dada será relativamente modesto.

Estructura de una tabla EAV

En un modelo de datos EAV cada par atributo-valor es un hecho describiendo una entidad y una fila en una tabla EAV almacena un solo hecho. Las tablas EAV a menudo se describen como “largas y flacas”, donde “largas” se refiere al número de filas y “flacas” a las pocas columnas.

Los datos son almacenados en 3 columnas principalmente:

- La entidad: el objeto a ser descrito.
- El atributo o parámetro: una llave foránea dentro de una tabla de definición de atributos. Al menos, las tablas de definiciones de atributos podrán contener las siguientes columnas: un identificador (id) de atributo, un nombre de atributo, una descripción, un tipo de dato, y columnas de ayuda a la validación de entradas (tamaño máximo de cadenas y expresiones regulares, conjunto de valores permitidos, etc.)
- El valor del atributo. (13)

### 1.4.4 Lenguajes de Programación.

#### 1.4.4.1 C++.

C++ es un lenguaje de programación diseñado a mediados de los años 1980 por Bjarne Stroustrup. La intención de su creación fue el extender al exitoso lenguaje de programación C con mecanismos que permitan la manipulación de objetos. En ese sentido, desde el punto de vista de los lenguajes orientados a objetos, el C++ es un lenguaje híbrido. Es uno de los lenguajes más populares y usados en el mundo. (14) C++ tiene ciertas características con respecto a los otros lenguajes de programación, las más notables son:

- Programación Orientada a Objetos

La posibilidad de orientar la programación a objetos permite a los programadores diseñar aplicaciones desde un punto de vista más parecido a una comunicación entre objetos que una secuencia de código estructurado. En adición este permite una mayor reusabilidad del código en una forma más productiva y lógica.

- Portabilidad

Se puede prácticamente compilar el mismo código C++ en casi cualquier tipo de computadora y sistema operativo sin hacer ningún cambio

- Brevedad

El código escrito en C++ es bastante pequeño en comparación con otros lenguajes, dado que el uso de caracteres especiales es preferido para palabras claves, reduciendo el esfuerzo del programador.

- Programación Modular

El cuerpo de una aplicación en C++ puede estar hecho por varios ficheros de código fuente que son compilados separadamente y luego conectados entre sí, ahorrando tiempo, dado que no es necesario recompilar la aplicación completa cuando se hace un cambio pequeño, sino solo el fichero que lo contiene.

Además, esta característica permite relacionar código C++ con código producido en otros lenguajes, como Ensamblador y C.

- Compatibilidad con C

C++ es compatible hacia atrás con el lenguaje C. Cualquier código escrito en C puede ser incluido fácilmente en un programa en C++ sin hacer ningún cambio.

- Velocidad

El código resultante de una compilación de C++ es eficiente, en medida a su dualidad como lenguaje de alto y bajo nivel y al tamaño reducido del lenguaje en sí.

Según lo que se planeta anteriormente este lenguaje es el más empleado en el mundo, el de mejor tiempo de ejecución y es robusto, pero su código es algo más complicado de leer y presenta una desventaja muy grande, la cual es que presenta grandes problemas por errores de acceso directo a las direcciones de memorias o interrupciones del hardware, la estructuración de sus proyectos son algo complicadas a la hora de distribuir los ficheros para la compilación, no presenta variantes o facilidades para la integración o uso web.

#### 1.4.4.2 Java.

Java es un lenguaje de programación orientado a objetos, desarrollado por Sun Microsystems a principios de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria.

Java fue diseñado con ciertos principios claves en mente:

**Fácil de Usar:** Las bases de Java vienen del lenguaje de programación C++. Un lenguaje poderoso pero que sin embargo se sentía muy complicado en la sintaxis, e inadecuado para todos los requerimientos de Java. Java construyó y mejoró las ideas de C++, para proveer un lenguaje que fuera poderoso pero simple de usar a la vez.

**Fiabilidad:** Java necesitaba reducir la probabilidad de errores fatales por errores de los programadores. Con esto en mente, la programación orientada a objeto fue introducida. Una vez que los datos y su manipulación fueron empacados juntos en un solo lugar, la robustez de Java se incrementó.

**Seguro:** Como Java fue originalmente orientado a dispositivos móviles que intercambiaría datos a través de redes, fue construido para incluir un alto nivel de seguridad.

**Plataforma Independiente:** dado la necesidad de programas para trabajar sin importar la computadora donde se fuera a ejecutar. Java fue escrito para ser un lenguaje portable que no se preocupara acerca del sistema operativo o hardware de la computadora.

El equipo de Sun Microsystems fue exitoso al combinar estos principios claves, y la popularidad de Java ha seguido a este por ser un lenguaje robusto, seguro, fácil de usar y portable. (15)

Por lo anteriormente descrito se puede concluir que Java es uno de los lenguajes más empleados actualmente en varias plataformas y tecnologías, presenta buenos IDEs de software libre, se integra con las web y es uno de los más modernos y de los primeros en fomentar la programación orientada a objetos, pero uno de sus principales problemas es que es muy pesado para su ejecución, por lo que consume muchos recursos de las PC y sus tiempos de ejecución son de los más lentos, lo cual en algunos entornos puede ser muy perjudicial.

#### **1.4.4.3 Python.**

Python es un lenguaje de programación creado por Guido van Rossum a principios de los años 90 cuyo nombre está inspirado en el grupo de comics ingleses “Monty Python”. Es un lenguaje similar a Perl, pero con una sintaxis muy limpia (simple, clara y sencilla), muy cercana al lenguaje natural que favorece un código legible que llega a parecer pseudocódigo. Se trata de un lenguaje semi-interpretado o de script, multiplataforma y orientado a objetos, con tipo de datos dinámico y fuerte, con un buen gestor de memoria, gran cantidad de librerías disponibles y gran potencia, haciendo que desarrollar una aplicación en Python sea sencillo, muy rápido y divertido. (16)

Python tiene muchas de las características de los lenguajes compilados, por lo que se podría decir que es semi-interpretado (híbrido). Siendo más flexible y más portable. Algunos casos de éxito en el uso de Python son Google, Yahoo!, la NASA, Industrias Light & Magic, y todas las distribuciones Linux, en las que Python cada vez representa un tanto por ciento mayor de los programas disponibles.

Ha ganado el premio TIOBE Programming Language del 2007 y 2010, siendo el lenguaje de programación del año. Este premio es otorgado al lenguaje de programación que ha logrado ganar mayor cantidad de mercado durante un año.

El lenguaje es el más nuevo de los analizados en este trabajo, pero es el que más ventajas brinda. Este es semi-interpretado. Presenta una gran y variada gama de librerías para diversos usos y necesidades de desarrollo, cuenta con un gran apoyo por parte de los seguidores del software libre. Ha tenido una gran aceptación y buenos resultados recientemente, principalmente en el uso de grandes aplicaciones de compañías muy importantes. Gracias a sus facilidades como lenguaje con tipado dinámico y algunas otras funcionalidades da la posibilidad de desarrollar software que se basan en un desarrollo y concepto genérico, logrando una gran abstracción, lo cual esto es de gran utilidad para el presente trabajo. Para la implementación se hará uso de la versión Python 2.6.6 con el empleo del IDE Eclipse 3.6.0 incluyendo el plugin Python Development Environment (PyDev) 1.6.2.

#### **1.4.5 Sistema Gestores de Base de Datos.**

##### **1.4.5.1 MySQL.**

El software MySQL proporciona un servidor de base de datos de Lenguaje Estructurado de Consultas (SQL por sus siglas en inglés) de tipo relacional muy rápido, multihilo, multiusuario y robusto. El servidor MySQL está diseñado para entornos de producción críticos, con alta carga de trabajo, así como para integrarse en software para ser distribuido.

El software MySQL tiene una doble licencia. Los usuarios pueden elegir entre usar el software MySQL como un producto Open Source bajo los términos de la licencia GNU General Public License o pueden adquirir una licencia comercial estándar de MySQL AB.

MySQL se ha convertido en el gestor de base de datos de código abierto más popular del mundo debido a su alto rendimiento, alta fiabilidad y facilidad de uso. Es también el gestor escogido para la nueva generación de aplicaciones desarrolladas en el paquete LAMP (Linux, Apache, MySQL, PHP / Perl / Python) Muchas de las organizaciones más grandes y de más rápido crecimiento del mundo, incluyendo Facebook, Google, Adobe, Wikipedia, Alcatel Lucent and Zappos, confían en MySQL para ahorrar tiempo

y dinero en el sustento de sus sitios webs de alto volumen, sistemas de negocios críticos y software empaquetado. Además, proyectos de software libre que requieren un sistema de administración de base de datos completo, a menudo usan MySQL. Algunos ejemplos son: Joomla, WordPress, phpBB, Drupal entre otros. (17)

MySQL se ejecuta en más de 20 plataformas incluyendo Linux, Windows, Mac OS, Solaris, HP-UX, IBM AIX, brindando el tipo de flexibilidad que pone al usuario en control.

Se conoce que está escrito en C y C++ y dispone con interfaces de integración para C, C++, Java, Perl, PHP, Python y Ruby.

Como análisis de la información descrita anteriormente se concluye que este gestor es uno de los más empleados actualmente por su facilidad de uso y configuración, además de que posee todas las funcionalidades básicas y vitales para la gestión de base de datos, no obstante presenta problemas para algunas situación no posee toda la robustez necesaria además de los problemas que podría presentar por tener doble licencia por lo que puede ser de pago o gratuito.

#### **1.4.5.2 Oracle.**

Oracle es un gestor de base de datos relacional para entornos cliente/servidor. Es un producto vendido a nivel mundial, aunque la gran potencia que tiene y su elevado precio hacen que sólo se vea en empresas muy grandes y multinacionales.

Este Sistema Manejador de Base de Dato Relacional (RDBMS por sus siglas en inglés) es el mayor y más usado en el mundo. La corporación Oracle ofrece este RDBMS como un producto incorporado a la línea de producción. Además incluye cuatro generaciones de desarrollo de aplicación, herramientas de reportes y utilitarios. (18)

Se divide en dos capas, una física y otra lógica, posee los datos y la estructura de las bases de datos por separado y en diferentes ficheros, esto gracias a su arquitectura robusta. Además posee un buen sistema de trazas, control de cambios y de recuperación.

Algunas de sus características principales son:

- Ofrece capacidades completas de replicación sincrónica y asincrónica,
- Cifrar la información de la clave de ingreso al momento de ingresar al sistema de la red cliente-servidor.
- En conjunto con Oracle SQL Net garantizan la autenticidad apropiada de los usuarios y la privacidad e integridad de los datos, y permiten manejar la asignación de privilegios y monitorear las operaciones de la base de datos a lo largo de toda la empresa.
- Proporciona un rendimiento excepcional DatawareHouse a través de técnicas avanzadas de optimización.
- Su arquitectura ofrece aplicaciones para procesamiento de transacciones en línea (OLTP por sus siglas en inglés) con la escalabilidad para soportar un gran número de usuarios y cargas de trabajo de alto volumen de transacciones.

Oracle es uno de los sistemas gestores de bases de datos más robustos y que más funcionalidades provee. Además posee asociado un excelente servicio de soporte técnico, de configuración, de despliegue y personalización. Solo presenta una inconveniencia es privativo y sus licencias de uso son bastante caras

estando alrededor de € 21.100 para un máximo de 25 usuarios en un servidor con la versión 11g Enterprise Edition.

### **1.4.5.3 PostgreSQL.**

El Sistema Gestor de Bases de Datos Relacionales Orientadas a Objetos conocido como PostgreSQL está derivado del paquete Postgres escrito en Berkeley. Con cerca de una década de desarrollo tras él, PostgreSQL es el gestor de bases de datos de código abierto más avanzado hoy en día, ofreciendo control de concurrencia multiversión, soportando casi toda la sintaxis SQL, incluyendo subconsultas, transacciones, tipos y funciones definidas por el usuario, contando también con un amplio conjunto de enlaces con lenguajes de programación como: C, C++, Java, Perl y Python.

PostgreSQL ofrece una potencia adicional sustancial al incorporar los siguientes cuatro conceptos adicionales básicos (Clases, Herencia, Tipos y Funciones) en una vía en la que los usuarios pueden extender fácilmente el sistema.

PostgreSQL tiene algunas características que son propias del mundo de las bases de datos orientadas a objetos. De hecho, algunas bases de datos comerciales han incorporado recientemente características en las que PostgreSQL fue pionera. (19)

Del análisis de la información se concluye que el presente gestor es el de mayor auge recientemente debido a sus grandes prestaciones, tanto básicas como especializadas, y al ser software libre de adquisición gratuita amplía su aceptación, es casi como el Oracle para el software libre, en gran medida por las nuevas funcionalidades de alto rendimiento que añade en cada una de sus nuevas versiones.

### **1.4.6 Visual Paradigm. Herramienta CASE.**

Visual Paradigm es una herramienta profesional para el modelado que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Permite dibujar los diferentes tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación.

Esta herramienta soporta hasta la fecha UML 2.1 completo y Notación para Modelado de Procesos de Negocio (BPMN por sus siglas en inglés), permite realizar ingeniería tanto directa como inversa a partir de un modelo relacional en SQL Server, MySQL, etc. Es capaz de desplegar todas las clases asociadas a las tablas con el patrón de diseño Una Clase-Una Tabla. Además, la herramienta es colaborativa, por lo que soporta múltiples usuarios trabajando sobre el mismo proyecto; genera la documentación del proyecto automáticamente en varios formatos web y permite control de versiones, además de ser multiplataforma. Posee una amplia vinculación con las tecnologías propietarias y de software libre. Es el que está definido por el Departamento de Soluciones Aduaneras del centro CEIGE como herramienta de modelado para el desarrollo de software.

Esta herramienta con el paso de los años se ha hecho bastante robusta y ha madurado mucho, presenta muchas funcionalidades y facilidades, además de una gran integración con las tecnologías de software libre, no obstante es privativa y sus licencias son algo caras.

#### **1.4.7 ER\Studio. Herramienta para modelado de Base de Datos.**

Es una herramienta de modelado de datos fácil de usar y multinivel, de gran utilidad para el diseño y construcción de bases de datos a nivel físico y lógico. Direcciona las necesidades diarias de los administradores de bases de datos, desarrolladores y arquitectos de datos que construyen y mantienen aplicaciones complejas y de bases de datos grandes. ER/Studio está equipado para crear y manejar diseños de bases de datos funcionales y confiables. Ofrece fuertes capacidades de diseño lógico, sincronización bidireccional de los diseños físicos y lógicos, construcción automática de bases de datos, documentación y fácil creación de reportes.

Funcionalidades:

- Capacidad fuerte en el diseño lógico.
- Sincronización bidireccional de los diseños lógico y físico.
- Construcción automática de base de datos.
- Reingeniería inversa de base de datos.
- Documentación basada en HTML.
- Un repositorio para el modelado.

A partir de la versión 7.5 ofrece soporte XML para la generación de esquemas, que permite a los modeladores de datos y desarrolladores de aplicaciones colaborar en iniciativas de SOA. (20)

Es la herramienta de modelado de base de datos definido por el Departamento de Soluciones Aduaneras del centro CEIGE como herramienta de modelado de datos.

#### **1.5 Conclusiones parciales.**

En este capítulo se abordó información relativa a los sistemas aduaneros líderes internacionalmente y de los existentes en el país. De este análisis se puede resumir que las características que debe tener una solución de gestión aduanera deben incluir:

- Soporte de recepción electrónico de información.
- Gestión de la información concerniente a la importación y exportación de mercancías.
- Análisis de posibles riesgos asociados a la entrada o salida de las mercancías.

Otro elemento importante identificado se refiere al tipo de solución a desarrollar. Es deseable que tenga algunos elementos conceptuales de SOA principalmente en la identificación del concepto de proceso como un elemento configurable y que puede variar en el tiempo y que debe poder ser modificado si es necesario. Cualquier modificación debe ser transparente a los usuarios finales de la aplicación. Igualmente se adoptará el concepto de centralización de los conceptos de negocio así como permitir flexibilidad en la descripción de sus atributos particulares.

Mediante el estudio de los fundamentos teóricos de las herramientas, tecnologías y lenguajes que se han abordado en este capítulo, se llega a la conclusión que el sistema propuesto será modelado con la herramienta Visual Paradigm la cual brinda una amplia gama de funcionalidades para el proceso de desarrollo de software; la programación se llevará a cabo empleando el lenguaje Python, el cual es ligero pero robusto, de fácil lectura y comprensión, libre, multiplataforma, con un gran soporte de librerías y de gran auge en las fechas recientes; el gestor de base de datos será PostgreSQL, el cual es el más completo y robusto de los SGBD de software libre; y para el modelado de la datos se utilizará Embarcadero ER\Studio por sus capacidades fuertes de diseño y fácil empleo. Para el diseño se hará uso de los patrones estudiados (FACADE, BUILDER, DESCRIPTOR).

## **2.1 Introducción.**

Este capítulo trata una panorámica del funcionamiento del sistema a proponer, y de los procesos que se llevan a cabo, así como la modelación del negocio y de la necesidad de su realización, donde se presentarán características generales y específicas de dicha solución, y además se detallarán aspectos relativos a la solución del sistema a diseñar e implementar. También se abordará la arquitectura del mismo y al conjunto de clases a desarrollar, así como la confección de los artefactos necesarios para un mejor entendimiento del problema y un acercamiento más avanzado a la solución del mismo. Se dará paso a la fase de implementación realizando para el mismo los correspondientes diagramas de clases de diseño, de interacción y modelo de datos que servirán de apoyo para lograr el funcionamiento del sistema solución.

## **2.2 Propuesta de solución.**

Como propuesta de solución para el desarrollo de la aplicación, se presenta un sistema que gestione las operaciones aduaneras, este será diseñado para un funcionamiento como núcleo de aplicación. Estará compuesto por 4 módulos: Entidades y Relaciones del Negocio, Procesos del Negocio, Documentos Digitales e Interfaz de Usuario, siendo los 3 primeros los fundamentales y centro del núcleo. Dichos módulos serán accesibles al usuario a través de una fachada que contendrá las funcionalidades del negocio y que son de vital importancia para el usuario, dicha fachada se brindará a través de la clase Facade, la cual aplica el patrón del mismo nombre. El último módulo se dedicará a definir los componentes necesarios para la interacción con el usuario a través de las Interfaces de Usuarios.

También mediante la clase Builder y Descriptor se aplican los patrones del mismo nombre respectivamente, donde el Builder construirá los objetos necesarios para el negocio empleando la descripción suministrada por el Descriptor.

Toda la aplicación será implementada en Python 2.6.6 con el empleo del IDE Eclipse 3.6.0 incluyendo el plugin PyDev 1.6.2 siguiendo una arquitectura de capas, donde habrá una capa de datos, en la cual la base de datos (PostgreSQL 9.0.1), estará estructurada en 2 esquemas (Configuración, Información). El primero contendrá lo referente a los nomencladores o tablas de control para la configuración y creación de las entidades, relaciones y procesos, este se modelará empleando alguna de las características que brinda el modelo de datos EAV por las prestaciones y facilidades que proporciona este modelo. El segundo contendrá la información de las entidades y sus relaciones en las tablas, las cuales se generarán de forma dinámica según sean necesarias para el funcionamiento de la aplicación.

También habrá una capa controladora, la cual encierra las clases donde se aplican los patrones Facade, Builder y Descriptor. Esta aplicación además será configurable en el proceso de despliegue e instalación, permitiendo al usuario la personalización del sistema según sus necesidades.

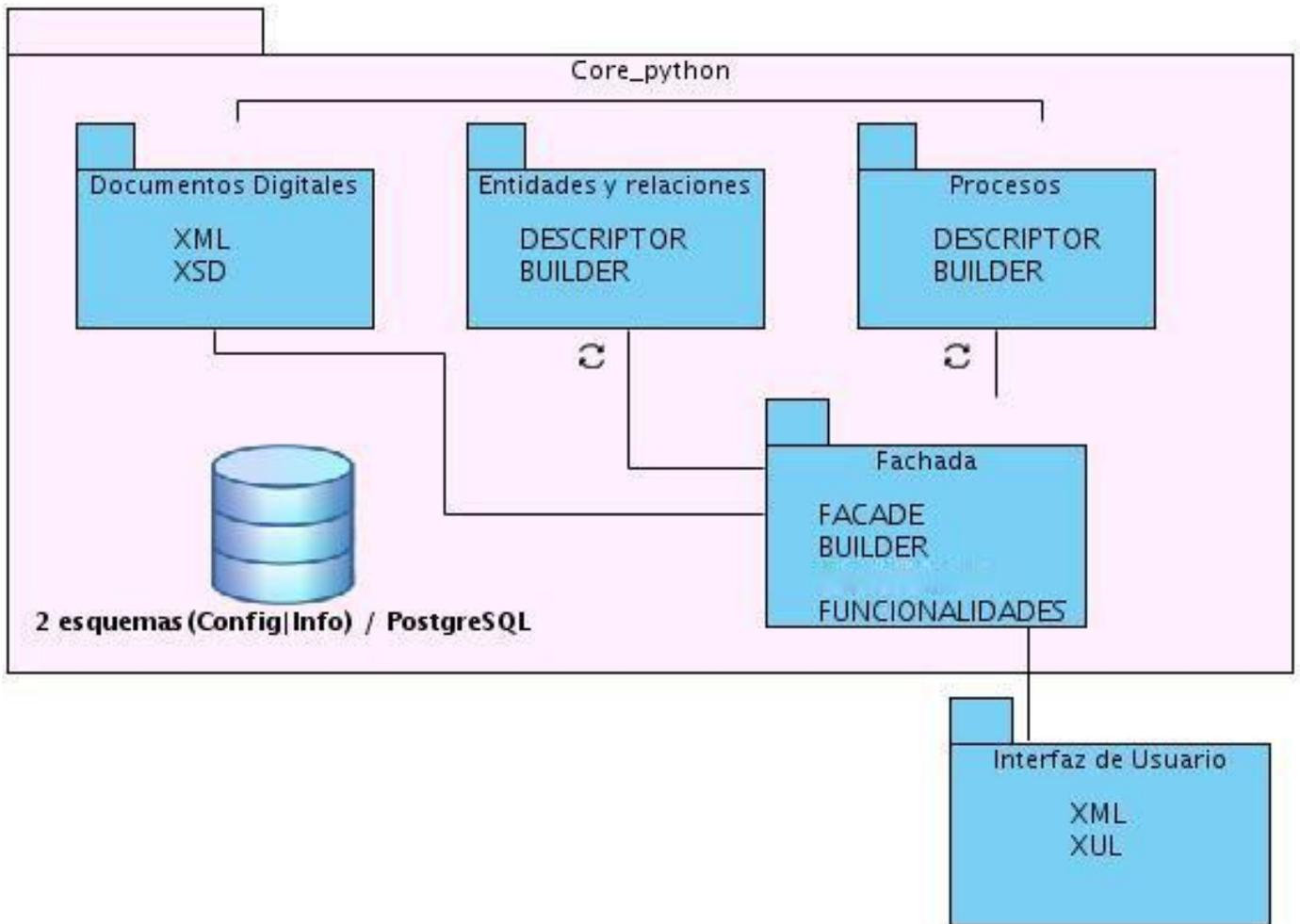


Imagen 4. Estructura general de la solución.

### 2.3 Requisitos del Análisis.

Listado de los requisitos funcionales capturados en el análisis para el desarrollo de la propuesta solución, los cuales se tomarán como base para proceso a implementar en la solución actual.

1. Gestionar Objeto
  - 1.1. Crear Objeto
  - 1.2. Modificar Objeto
  - 1.3. Eliminar Objeto
2. Gestionar Atributo
  - 2.1. Crear Atributo
  - 2.2. Modificar Atributo
  - 2.3. Eliminar Atributo

3. Gestionar Entidad
  - 3.1. Crear Entidad
  - 3.2. Modificar Entidad
  - 3.3. Eliminar Entidad
4. Gestionar Relación
  - 4.1. Crear Relación
  - 4.2. Modificar Relación
  - 4.3. Eliminar Relación
5. Gestionar Acción
  - 5.1. Crear Acción
  - 5.2. Modificar Acción
  - 5.3. Eliminar Acción
6. Gestionar Proceso
  - 6.1. Crear Proceso
  - 6.2. Modificar Proceso
  - 6.3. Configurar Proceso
  - 6.4. Eliminar Proceso
7. Publicar Proceso
8. Ejecutar Proceso

Descripción de los requisitos funcionales ver Anexo I.

## 2.4 Diagramas de Diseño.

### 2.4.1 Diagrama de Clases.

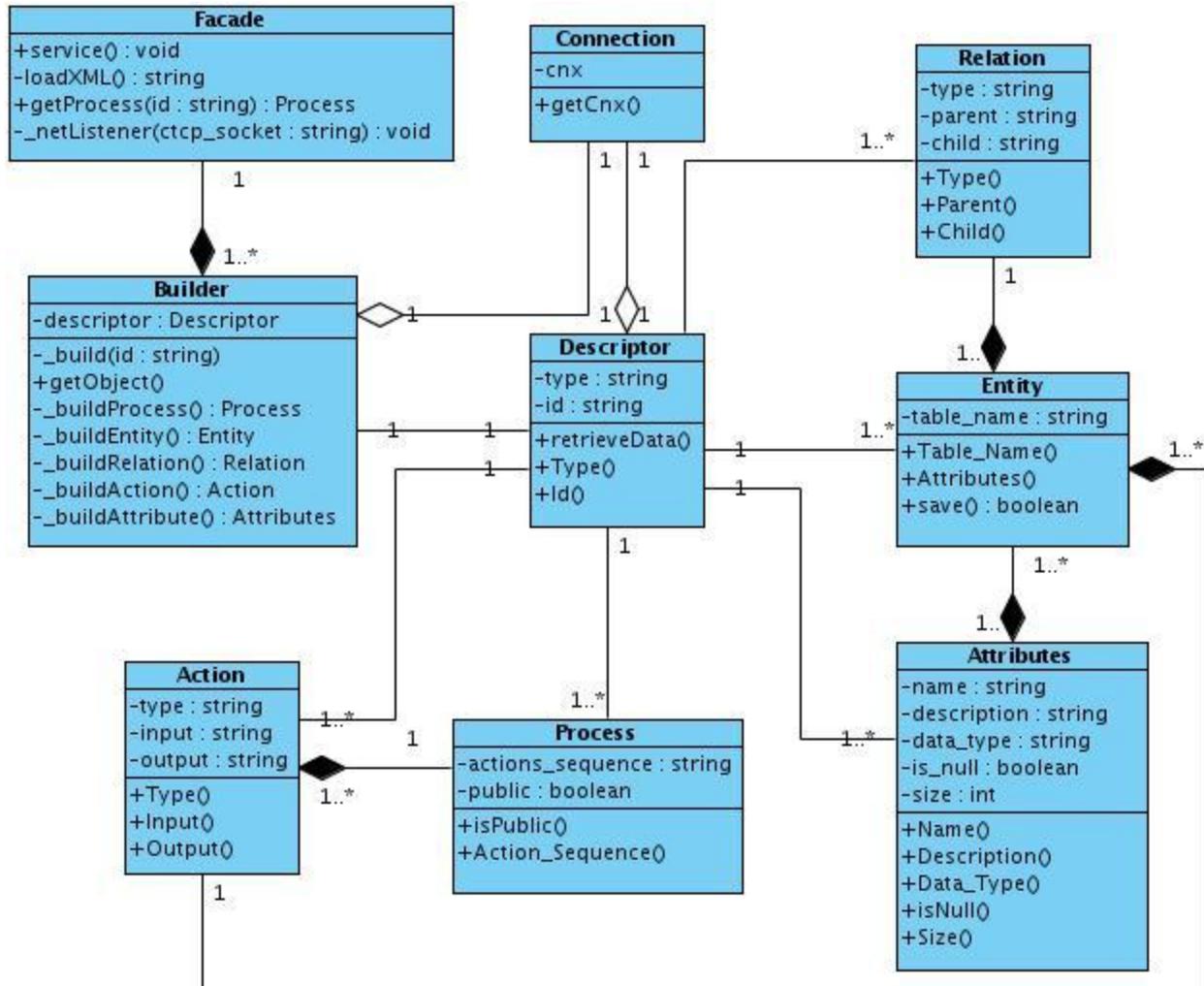


Imagen 5. Diagrama de clases de diseño.

Tablas de descripción de las clases de diseño.

<b>Nombre: Facade</b>	
<b>Tipo de clase controladora</b>	
<b>Atributo</b>	<b>Tipo</b>
<b>Para cada responsabilidad:</b>	
Nombre:	service()
Descripción:	Brinda el conjunto de servicios de la aplicación, los cuales serían los correspondientes a las funcionalidades involucradas en los procesos que son públicos dentro de la aplicación.
Nombre:	loadXML()
Descripción:	Carga un XML con los procesos públicos
Nombre:	getProcess(id:String)
Descripción:	Brinda el proceso seleccionado, el cual tiene como parámetro el id del proceso requerido
Nombre:	__netListener(ctcp_socket:String)
Descripción:	Se encarga de recibir y responder las peticiones que son aceptadas por la red a través de un puerto específico y por medio del protocolo TCP.

Tabla 1. Descripción clase Facade.

<b>Nombre: Connection</b>	
<b>Tipo de clase controladora</b>	
<b>Atributo</b>	<b>Tipo</b>
cnx	
<b>Para cada responsabilidad:</b>	
Nombre:	getCnx()
Descripción:	Retorna un objeto de tipo conexión al realizar una llamada al método connect del paquete de conexión a la base de datos, psycopg2

Tabla 2. Descripción clase Connection.

<b>Nombre: Builder</b>	
<b>Tipo de clase controladora</b>	
<b>Atributo</b>	<b>Tipo</b>
descriptor	Descriptor
<b>Para cada responsabilidad:</b>	
Nombre:	__build(id:String)
Descripción:	Según el id pasado como parámetro invoca uno de los métodos build en cuestión para la creación de un objeto específico.
Nombre:	getObject()
Descripción:	Retorna el objeto creado en cuestión, ya sea un proceso, entidad, relación, acción o atributo
Nombre:	__buildProcess()
Descripción:	Crea un proceso dado
Nombre:	__buildEntity()
Descripción:	Crea una entidad dada
Nombre:	__buildRelation()
Descripción:	Crea una relación dada
Nombre:	__buildAction()
Descripción:	Crea una acción dada
Nombre:	__buildAttribute()
Descripción:	Crea un atributo dado

Tabla 3. Descripción clase Builder.

<b>Nombre: Descriptor</b>	
<b>Tipo de clase controladora</b>	
<b>Atributo</b>	<b>Tipo</b>
type	String
id	String

<b>Para cada responsabilidad:</b>	
Nombre:	retrieveData()
Descripción:	Consulta el esquema de configuración de la base de datos, para la obtención de la información referente al objeto en cuestión.
Nombre:	Type()
Descripción:	Propiedad que encapsula las funciones get y set del atributo type, que identifica el tipo de objeto a describir
Nombre:	Id()
Descripción:	Propiedad que encapsula las funciones get y set del atributo id, que identifica el objeto en cuestión

Tabla 4. Descripción clase Descriptor.

<b>Nombre: Process</b>	
<b>Tipo de clase controladora</b>	
<b>Atributo</b>	<b>Tipo</b>
actions_sequence	String
public	Boolean
<b>Para cada responsabilidad:</b>	
Nombre:	isPublic()
Descripción:	Propiedad que encapsula las funciones get y set del atributo public, que identifica si el proceso es público o no
Nombre:	Action_Sequence()
Descripción:	Propiedad que encapsula las funciones get y set del atributo actions_sequence, que lista la secuencia de acciones del proceso en cuestión

Tabla 5. Descripción clase Process.

<b>Nombre: Action</b>	
<b>Tipo de clase controladora</b>	
<b>Atributo</b>	<b>Tipo</b>
type	String
input	String
output	String
<b>Para cada responsabilidad:</b>	
Nombre:	Type()
Descripción:	Propiedad que encapsula las funciones get y set del atributo type, que identifica el tipo de acción
Nombre:	Input()
Descripción:	Propiedad que encapsula las funciones get y set del atributo input, que identifica la entrada que posee la acción
Nombre:	Output()
Descripción:	Propiedad que encapsulas las funciones get y set del atributo output, que identifica la entrada que posee la acción

Tabla 6. Descripción clase Action.

<b>Nombre: Entity</b>	
<b>Tipo de clase entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
table_name	String
<b>Para cada responsabilidad:</b>	
Nombre:	Table_Name()
Descripción:	Propiedad que encapsula las funciones get y set del atributo table_name, que identifica el nombre de la tabla que le corresponderá a la entidad en cuestión
Nombre:	Attributes()
Descripción:	Propiedad que encapsula las funciones get y set del atributo attributes, que lista el

	conjunto de atributos relacionados a la entidad
Nombre:	save()
Descripción:	Almacena la entidad en cuestión en una tabla correspondiente a esta en la base de datos, específicamente en el esquema de información.

Tabla 7. Descripción clase Entity.

<b>Nombre: Relation</b>	
<b>Tipo de clase controladora</b>	
<b>Atributo</b>	<b>Tipo</b>
type	String
parent	String
child	String
<b>Para cada responsabilidad:</b>	
Nombre:	Type()
Descripción:	Propiedad que encapsula las funciones get y set del atributo type, que identifica el tipo de relación
Nombre:	Parent()
Descripción:	Propiedad que encapsula las funciones get y set del atributo parent, que identifica el padre de la relación
Nombre:	Child()
Descripción:	Propiedad que encapsula las funciones get y set del atributo child, que identifica el hijo de la relación

Tabla 8. Descripción clase Relation.

<b>Nombre: Attribute</b>	
<b>Tipo de clase entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
name	String
description	String

data_type	String
is_null	boolean
size	Integer
<b>Para cada responsabilidad:</b>	
Nombre:	Name()
Descripción:	Propiedad que encapsula las funciones get y set del atributo name, que identifica el nombre del atributo como se representara en la tabla entidad correspondiente en el esquema de información de la base de datos
Nombre:	Description()
Descripción:	Propiedad que encapsula las funciones get y set del atributo description, que identifica la descripción del atributo que se representara en la tabla entidad correspondiente en el esquema de información de la base de datos
Nombre:	Data_Type()
Descripción:	Propiedad que encapsula las funciones get y set del atributo data_type, que identifica el tipo de dato del atributo que se representara en la tabla entidad correspondiente en el esquema de información de la base de datos
Nombre:	Is_Null()
Descripción:	Propiedad que encapsula las funciones get y set del atributo is_null, que identifica si el atributo que se representara en la tabla entidad correspondiente en el esquema de información de la base de datos es nulo o no
Nombre:	Size()
Descripción:	Propiedad que encapsula las funciones get y set del atributo size, que identifica el tamaño del atributo que se representara en la tabla entidad correspondiente en el esquema de información de la base de datos

Tabla 9. Descripción clase Attribute.

## 2.4.2 Modelo de Datos.

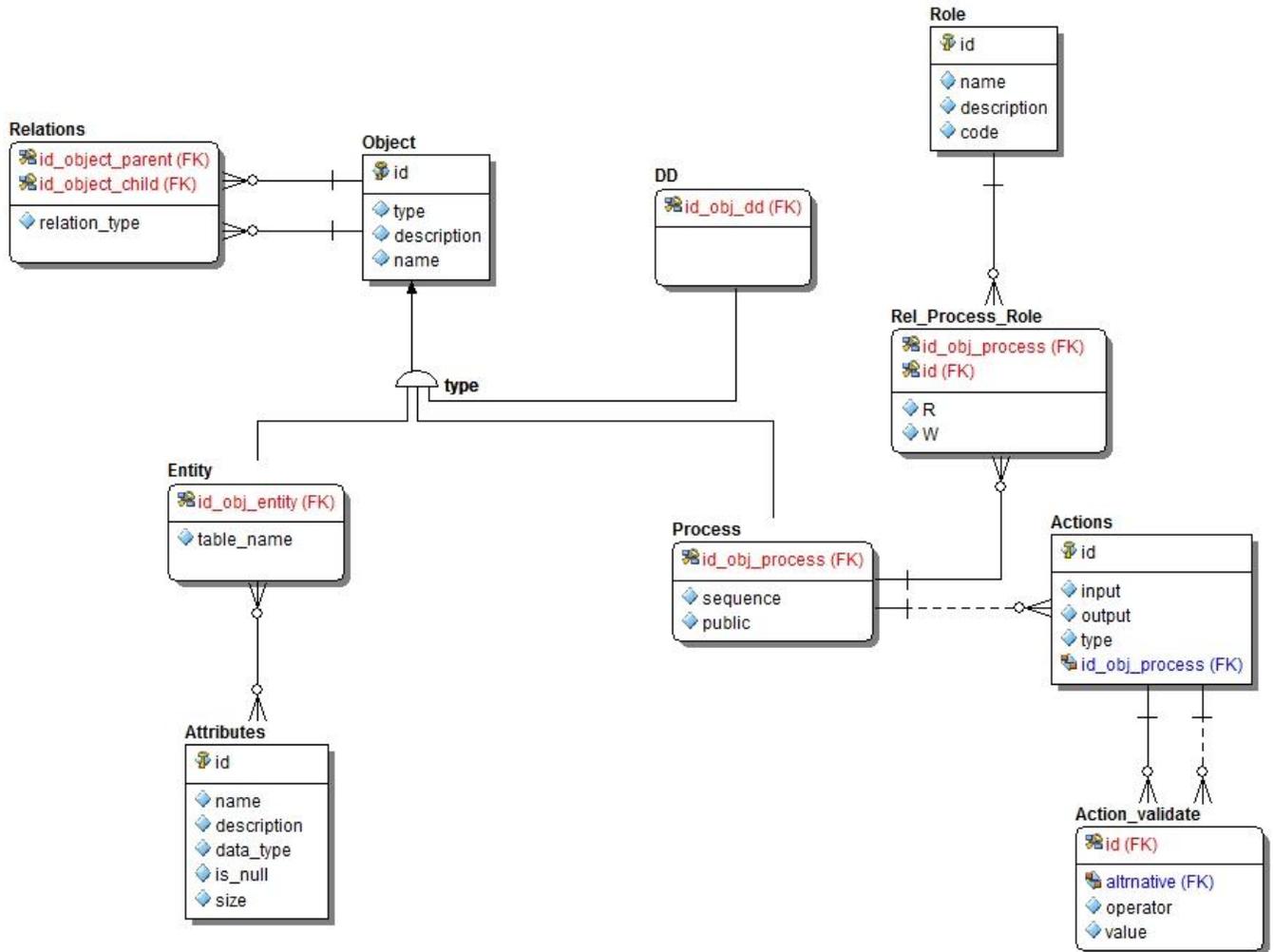


Imagen 6. Diagrama Entidad Relación del esquema de configuración de la base de datos.

Descripciones de las tablas ver en las tablas del Anexo II.

### 2.4.3 Diagramas de Interacción y Actividades.

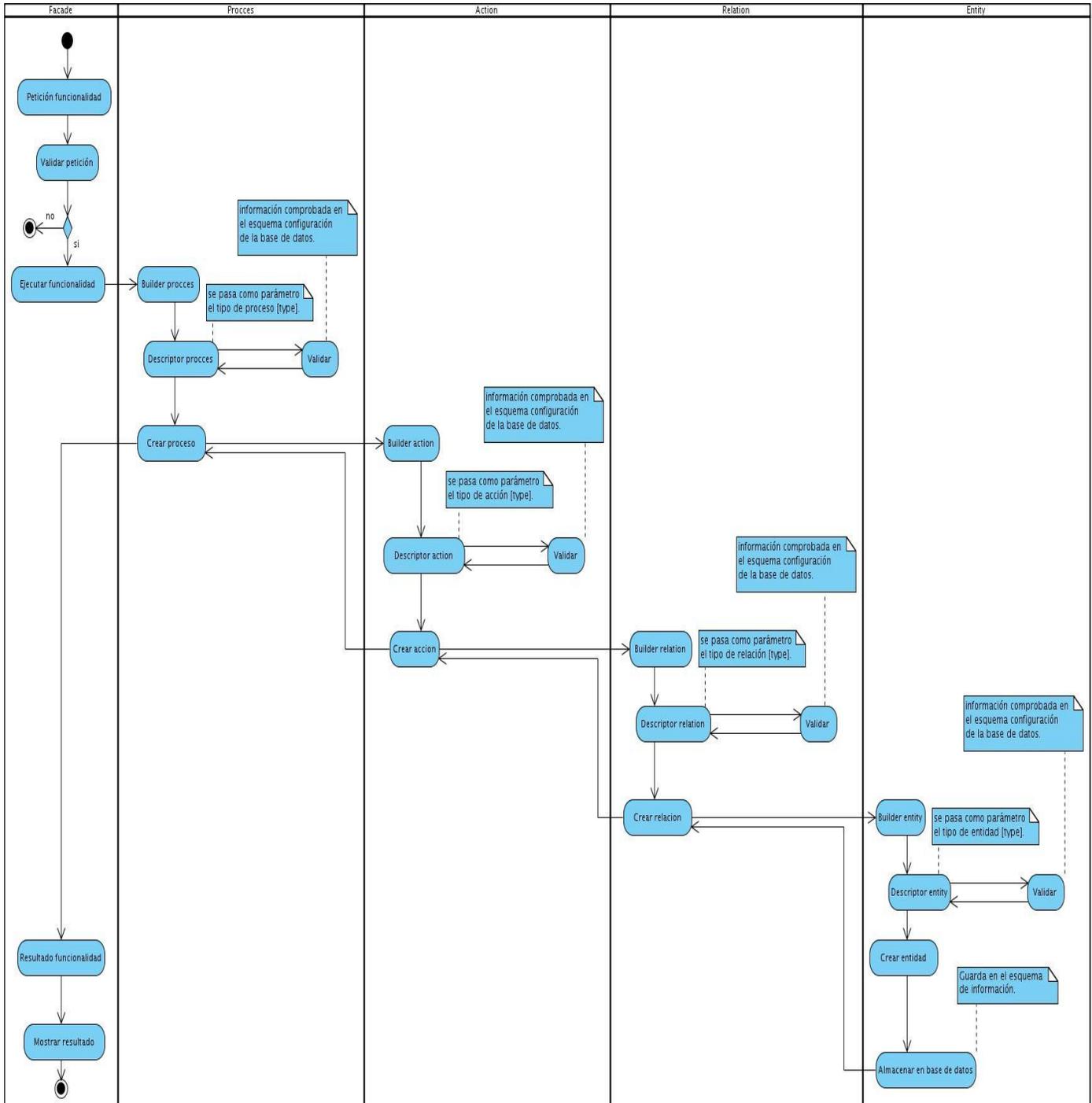


Imagen 7. Diagrama de Interacción y Actividades General de la configuración de un proceso.

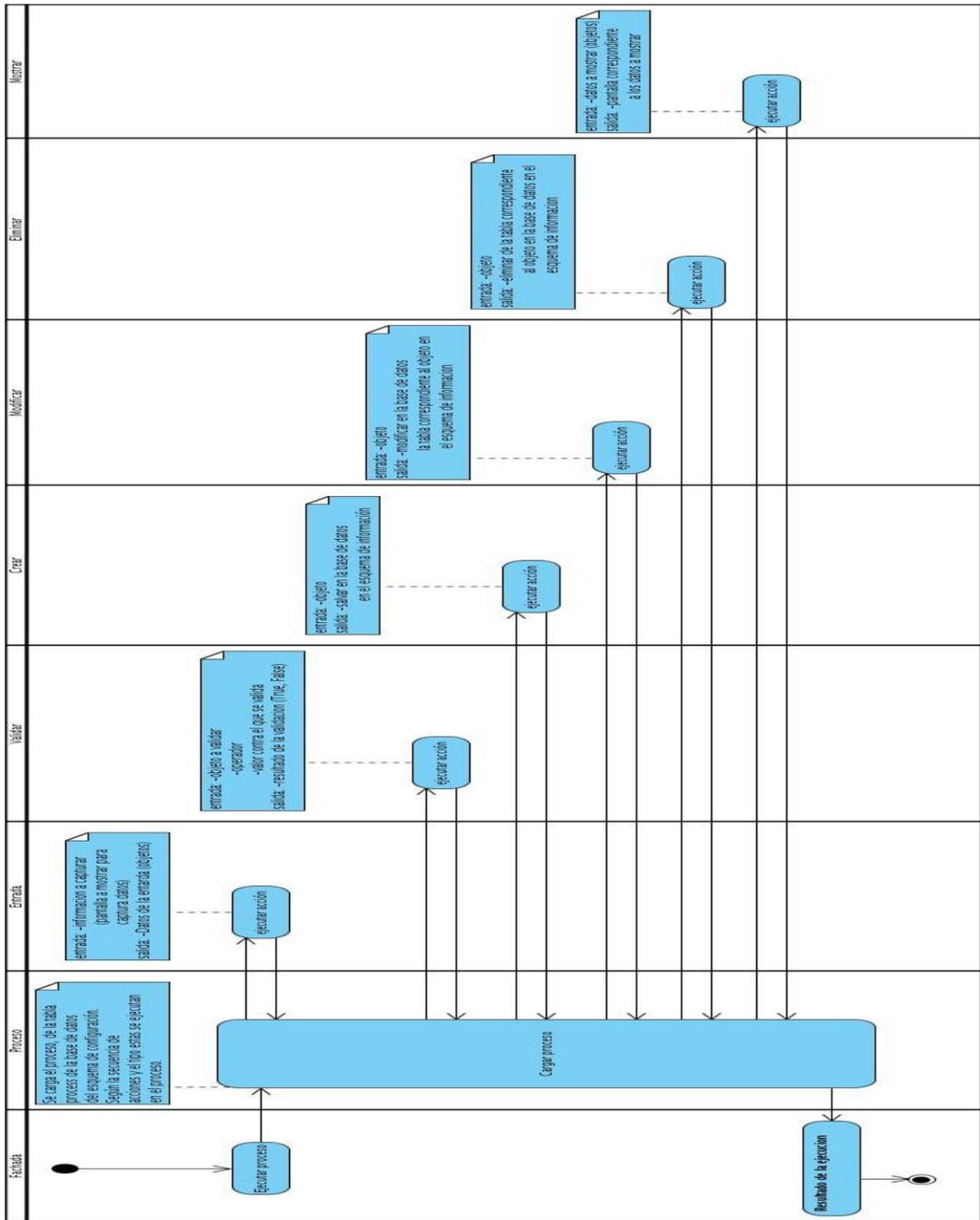


Imagen 8. Diagrama de Interacción y Actividades General de la ejecución de un proceso.

## 2.5 Diagrama de Paquetes.

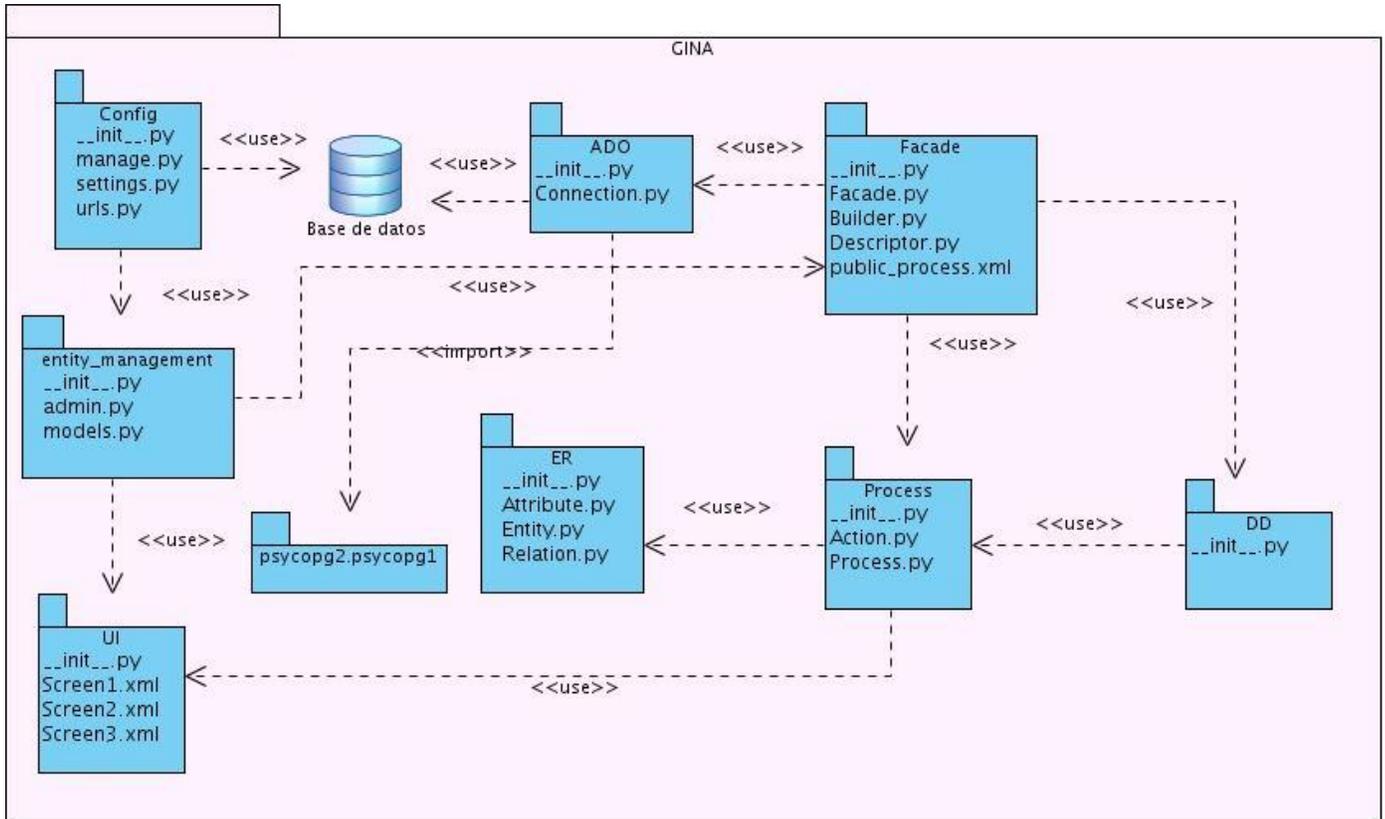


Imagen 9. Diagrama de Paquetes

## 2.6 Diagrama de Despliegue.

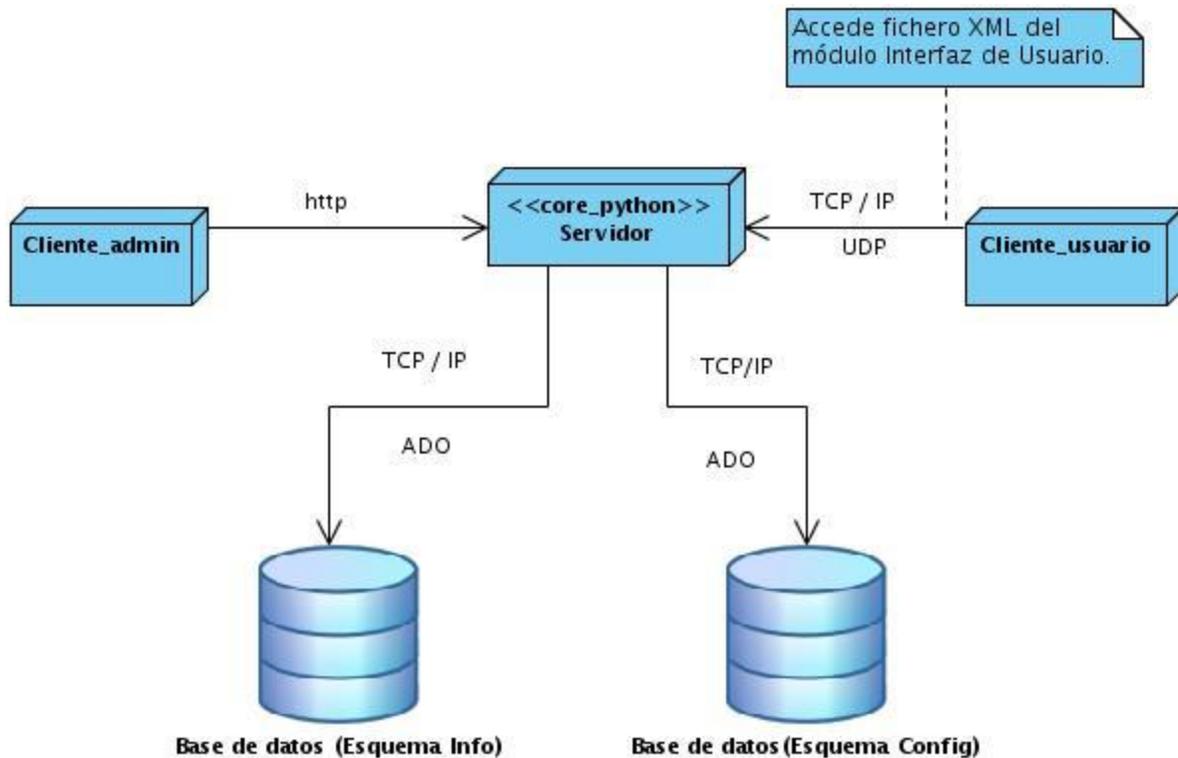


Imagen 10. Diagrama de despliegue de la solución

## 2.7 Convenciones de codificación.

Estas convenciones de código son un extracto principalmente de la guía de estilo oficial para código Python (21), empleado para la lograr buenas prácticas de programación y hacer el código mucho más legible.

### Disposición del código

- Sangría  
Se usaran 4 espacios por nivel de sangría, a través del empleo de *Tabulaciones* y nunca se mezclarán con *Espacios*.
- Líneas en blanco  
Se separan las funciones de alto nivel y las definiciones de clases con dos líneas en blanco. Las definiciones de métodos dentro de una clase se separan por una simple línea en blanco. Se empleará líneas en blanco económicamente dentro de funciones para indicar secciones lógicas.

### Importaciones

- Las importaciones estarán siempre al comienzo de los ficheros, justo después de cualquier comentario del módulo.
- Las importaciones estarán en líneas separadas.

- Las importaciones se agruparán en el siguiente orden:
  - ✓ Importaciones de librerías estándar
  - ✓ Importaciones relacionadas de terceros
  - ✓ Importaciones de locales de la aplicación/librerías específicas
- Se empleará una línea en blanco entre cada grupo de importaciones

#### Espacios en blancos en las expresiones y declaraciones

- Se evitará el uso de espacios en blanco en las siguientes situaciones:
  - ✓ Inmediatamente dentro de paréntesis, corchetes o llaves.
  - ✓ Inmediatamente antes de una coma, dos puntos o punto y coma.
  - ✓ Inmediatamente antes de abrir el paréntesis que comienza la lista de argumentos de la llamada de una función.
  - ✓ Inmediatamente antes del paréntesis o corchete de un indexado
  - ✓ Más de un espacio alrededor de una asignación o cualquier otro operador para alinearlos con otro.
- Siempre rodear los operadores binarios y aritméticos con un espacio en blanco a cada lado: asignación (=), asignación aumentada (+=, -= etc.), comparaciones (==, <, >, !=, <>, <=, >=, in, not in, is, is not), booleanos (and, or, not).

#### Convenciones de nomenclatura

- Descriptivo: Estilos de nomenclatura
  - ✓ lowercase
  - ✓ lower\_case\_with\_underscores
  - ✓ UPPERCASE
  - ✓ UPPER\_CASE\_WITH\_UNDERSCORES
  - ✓ CapitalizedWords (o CapWords, o CamelCase, o StudlyCaps)  
Nota: Cuando se usa abreviaciones en CapWords, capitalizar todas las letras de la abreviación. HTTPServerErrores mejor que HttpServerError.
  - ✓ mixedCase (difiere de CapitalizedWords por un carácter inicial lowercase)
  - ✓ Capitalized\_Words\_With\_Underscores
- Prescriptivo: Convenciones de nomenclatura
  - ✓ Nombres de paquetes y módulos  
Se empleará nombres cortos e informativos, con el empleo del estilo CamelCase y underscores para una mayor legibilidad.
  - ✓ Nombres de clases  
Sin ninguna excepción las clases emplearán la convención CamelCase.
  - ✓ Nombres de funciones o métodos  
Se empleará la convención mixedCase.  
Para funciones o métodos no públicos se empleará un underscore al principio.
  - ✓ Variables o atributos de clases  
Se empleará lowercase para las variables o atributos, con el empleo de underscore para una mejor lectura y además para los que no son públicos un doble underscore al principio.

## **2.8 Conclusiones parciales.**

En este capítulo se comenzó haciendo una descripción general de las características técnicas de la solución a implementar. Dicha descripción fue luego complementada con los correspondientes diagramas y artefactos que ilustran como estará estructurado el sistema.

Se presentó la vista a nivel de clases, de datos, de componentes y despliegue. Se ilustró la secuencia e interacción de las clases apoyado en los diagramas de interacción y actividades.

Finalmente se describió la convención de codificación utilizada en la implementación del sistema.

### 3.1 Introducción.

En este capítulo se hace referencia a las métricas que son aplicadas para validar el diseño de software en cuanto a su calidad y a las pruebas que se realizarán, siendo esta una actividad en la cual un sistema es ejecutado bajo condiciones o requerimientos específicos. Se definen las métricas que se aplicaron a la solución y se realiza una valoración de los resultados, al igual que las pruebas de caja negra que se ejecutarán.

### 3.2 Métricas para el Modelo de diseño orientado a objetos.

Las métricas para el diseño proporcionan una mejor visión interna, permiten estimar la efectividad del proceso y ayudan a que el diseño evolucione a un nivel superior de calidad. A medida que los modelos de diseño orientado a objetos van creciendo en tamaño y complejidad, es aconsejable y beneficioso tener una visión más objetiva de las características del diseño. Esto se puede lograr a través del uso de las métricas para validar diseños basadas en características de las clases.

#### 3.2.1 Métricas orientadas a clases.

Una clase es la unidad principal de todo sistema orientado a objetos, por lo que las medidas y métricas para una clase individual, la jerarquía de clases y las colaboraciones de clases resultan sumamente valiosas para un ingeniero de software que necesite estimar la calidad de un diseño.

Las características mencionadas anteriormente han sido utilizadas como bases para el conjunto de métricas Chidamber y Kemerer (CK).

#### 3.2.2 Métricas propuestas por Chidamber y Kemerer.

Las métricas de Chidamber y Kemerer consisten originalmente en seis métricas calculadas por cada clase:

- Métodos ponderados por clases (MPC)
- Árbol de profundidad de herencia (APH)
- Número de descendientes (ND)
- Acoplamiento entre clases objeto (ACO)
- Relaciones entre clases (RC)
- Carencia de cohesión de los métodos (CCM)

En el presente trabajo se hace uso de algunas de ellas para la validación de las clases del diseño.

### Árbol de profundidad de herencia (APH)

El objetivo de esta métrica es medir la distancia desde una clase a la clase raíz del árbol de herencia. Una clase con un número de herencia muy alto tiende a incrementar los errores, por lo que se recomienda un APH de cinco o menos clases herederas.

### Número de descendientes (ND)

Esta métrica cuenta el número de hijos inmediatos del árbol de herencia. Un alto grado de NDD indica una alta reutilización de los métodos, ya que la herencia es una forma de reutilización; no obstante, la abstracción representada por la clase predecesora puede diluirse. Esto significa que existe una posibilidad de que algunos descendientes no sean miembros, realmente apropiados, de la clase predecesora. Una clase con muchos hijos requiere además, de mayor tiempo para las pruebas.

### Carencia de cohesión de los métodos (CCM)

Cada método dentro de una clase, accede a uno o más atributos (también llamados variables de instancia). CCM es el número de métodos que accede a uno o más de los mismos atributos. Si no existen métodos que accedan a los mismos atributos, entonces  $CCM = 0$ .

### Relaciones entre Clases (RC)

Está dado por el número de relaciones de uso de una clase con otras.

#### 3.2.2.1 Aplicación de las métricas.

Métricas CK	Resultado del análisis
Árbol de profundidad de herencia	La longitud máxima desde la raíz tiene un valor de 0, por lo que el diseño de las clases se puede clasificar como sencillo y fácil de implementar.
Número de descendientes	El número mayor de clases hijas es 0, lo cual refleja un buen manejo de la abstracción en clases predecesoras.
Carencia de cohesión de los métodos	Luego de revisar detenidamente cada diagrama de diseño se pudo constatar que en la mayoría de las clases, a lo sumo existe un método que accede a un mismo atributo. Esto arroja un valor $CCM = 1$ . Exceptuando la clase Builder donde existen 6, para un $CCM = 6$ Dado que es una sola clase de las 9 existentes, la que posee un valor medio y el resto posee un valor bastante pequeño, se llega a la conclusión de que las clases del diseño poseen un alto grado de cohesión.
Relaciones entre Clases	El modelo posee una calidad buena pudiéndose observar que el 89% de las clases posee cuando más 3 dependencias de otras clases.

Tabla 10. Resultado de las métricas Chidamber y Keremer.

Al nivel orientado a clases, la colección de métricas CK puede complementarse con las métricas propuestas por Lorenz y Kidd.

### 3.2.3 Métricas propuestas por Lorenz y Kidd.

Lorenz y Kidd separan las métricas basadas en clases en cuatro amplias categorías: tamaño, herencia, valores internos y valores externos.

- Tamaño operacional de clase (TOC).
- Número de operaciones redefinidas para una subclase (NOR).
- Número de operaciones añadidas por una subclase (NOA).
- Índice de especialización (IES).

En el presente trabajo se hace uso de algunas de ellas para la validación de las clases del diseño.

#### Número de operaciones redefinidas para una subclase.(NOR)

Existen casos en que una subclase reemplaza una operación heredada de su superclase por una versión especializada para su propio uso, a lo cual se le llama redefinición. Los valores grandes para el NOR generalmente indican un problema de diseño.

#### Índice de especialización (IES)

El índice de especialización proporciona una indicación aproximada del grado de especialización para cada una de las subclases en un sistema orientado a objetos. La especialización se puede alcanzar añadiendo o eliminando operaciones, pero también redefiniendo.  $IE = [NOR * nivel] / Mtotal$  donde nivel corresponde al nivel en la jerarquía de clases en que reside la clase, y **Mtotal** es el número total de métodos de la clase. Cuanto más elevado sea el valor de IE, más probable será que la jerarquía de clases tenga clases que no se ajusten a la abstracción de la superclase.

#### Tamaño operacional de clase (TOC).

Está dado por el número de métodos asignados una clase.

#### 3.2.3.1 Aplicación de las métricas.

Métricas Lorenz y Kidd	Resultado del análisis
Número de operaciones redefinidas para una subclase	Las clases del diseño no redefinen los métodos que heredan de las superclases. Por lo que ha sido manejada adecuadamente la abstracción representada por las superclases. Dado de que no hay herencia.
Índice de especialización	$IE = [0 * 0] / 4 = 0$ (1) $IE = [0 * 0] / 3 = 0$ (2) En (1) se utilizó la clase Facade y en (2) la clase Entity. En el resto de las clases del diseño este valor también es cero lo cual prueba que las clases están ajustadas a la abstracción de las superclases.

Tamaño operacional de clase	Todas las clases poseen un tamaño pequeño, dado que la que más posee solo tiene 7 operaciones, siendo mucho menor que 20 que es valor que se establece para definir una clase pequeña. El 67% (6 clases) de las clases presenta una responsabilidad y complejidad baja y solo un 13% (3 clases) posee una responsabilidad y complejidad media, de un total de 9 clases. Por lo que estas no poseen una gran responsabilidad, lo cual aumenta la reutilización de estas y facilitara su implementación y prueba.
-----------------------------	---

Tabla 11. Resultado de las métricas Lorenz y Kidd.

CTDAD. PROCEDIMIENTOS	RESPONSABILIDAD	COMPLEJIDAD	REUTILIZACIÓN
4	M	M	M
1	B	B	A
3	B	B	A
7	M	M	M
3	B	B	A
3	B	B	A
3	B	B	A
2	B	B	A
5	M	M	M
31			

Tabla 12. Datos métrica TOC.

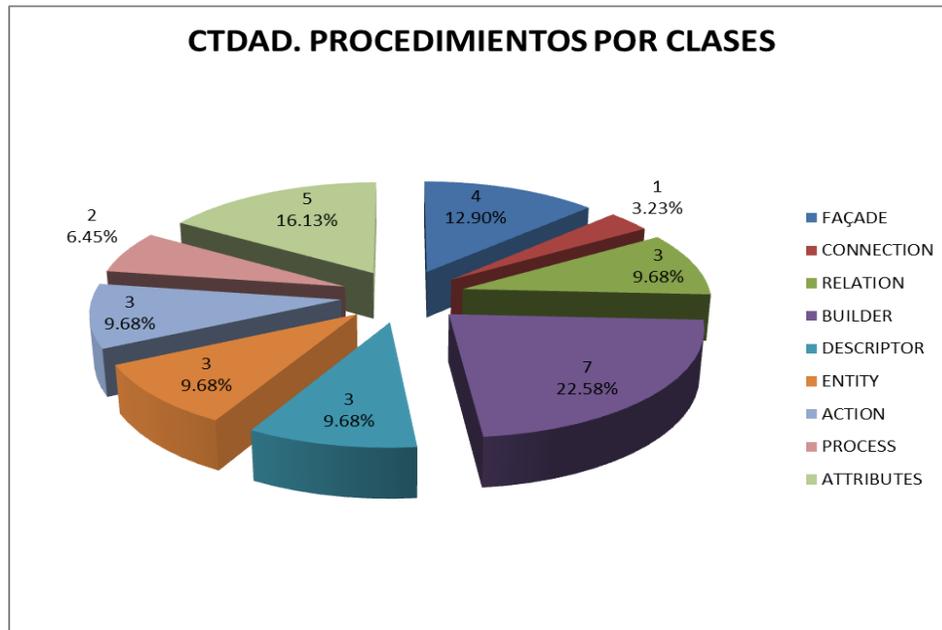


Imagen 11. Grafica distribución de métodos por clases.

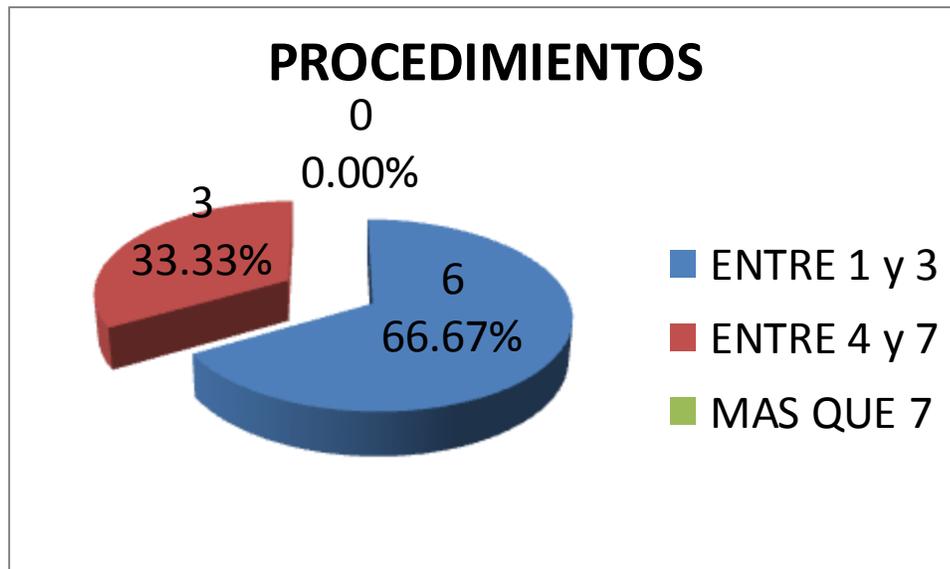


Imagen 12. Gráfica Porcientos de clases con cantidad de métodos.

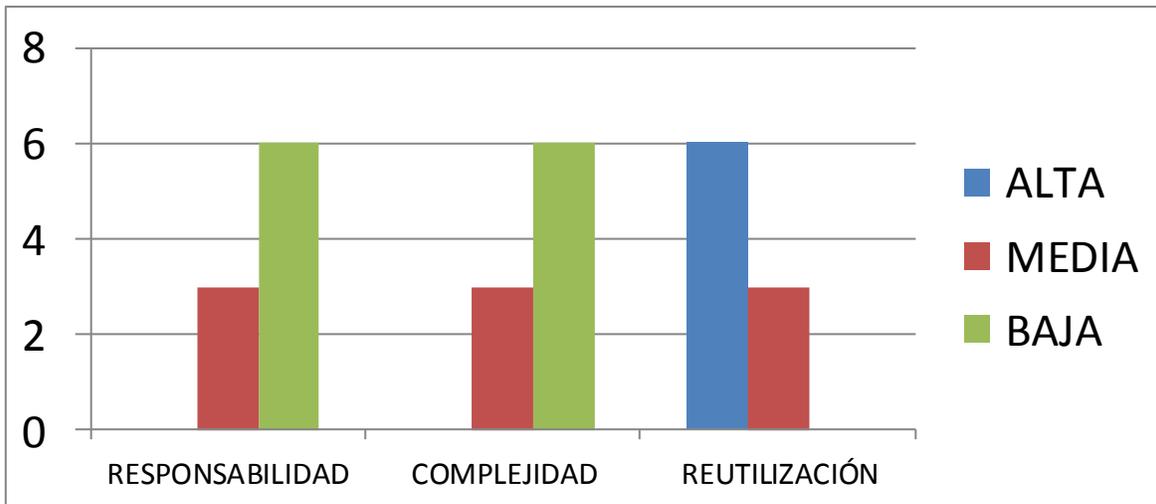


Imagen 13. Gráfica afectación de la métrica TOC en los atributos que se muestran.

### 3.3 Descripción de las pruebas.

Las pruebas del software son un elemento crítico para la garantía de la calidad del software y representa una revisión final de las especificaciones, del diseño y de la codificación. Las pruebas son realizadas con el objetivo de detectar errores en el sistema, por lo que se llevan a cabo durante todo el ciclo de vida del producto, su mayor punto de desarrollo se puede encontrar en la culminación de la fase de implementación. Dentro de las pruebas se destacan dos tipos de pruebas: caja blanca y caja negra.

**Caja blanca:** son realizadas al código obtenido, con el objetivo de comprobar todos los posibles caminos lógicos del software que ponen a prueba las funcionalidades del sistema a través de un conjunto de condiciones. Esto permite que se pueda analizar el sistema en varios puntos para determinar si el estado real coinciden con el esperado o mencionado.

**Caja negra:** son las pruebas que se llevan a cabo sobre la interfaz del software. O sea, a través de los casos de prueba se demuestra que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce un resultado correcto, así como que la integridad de la información externa se mantiene.

Esta fue precisamente de las dos la que se aplicó al sistema desarrollado permitiendo obtener las entradas necesarias para probar los requisitos funcionales del sistema. Fueron realizadas a través de la confección de los casos de pruebas que se muestran a continuación.

#### 3.3.1 Aplicación de las pruebas de caja negra.

Para la aplicación de este tipo de prueba se usará el requisito Gestionar Objeto. El objetivo de este requisito es la creación, modificación y eliminación de un objeto de negocio.

Casos de pruebas ver Anexo III.

### 3.4 Conclusiones parciales.

La validación del diseño efectuado como se ha mostrado, arrojó resultados satisfactorios en la evaluación de la calidad en la concepción de las clases y sus relaciones. Se ilustraron estos resultados a través de gráficas y tablas que pretenden hacer más claro el análisis de los mismos.

Al mismo tiempo se presentó la forma en que se realizó la validación funcional con el detalle de los casos de pruebas realizados y sus correspondientes resultados.

Al examinar los datos arrojados por ambas validaciones se puede concluir, que no se logró una solución perfecta, pero que tiene un nivel medio de calidad en su diseño y cumple cabalmente con las funcionalidades que se desean para que cumpla el objetivo que tiene destinado.

## **CONCLUSIONES GENERALES**

Luego de culminada la presente investigación se puede arribar a las siguientes conclusiones:

- La revisión de los sistemas aduaneros y las soluciones informáticas con capacidad de adaptación al cambio, permitió concebir el desarrollo de un núcleo de aplicación, que permite la configuración de los procesos y entidades asociadas al negocio aduanero de manera dinámica.
- El diseño realizado para la solución, cubre las funcionalidades identificadas para poder satisfacer las necesidades existentes, y sirvió de base a la implementación del sistema.
- La validación realizada tanto al diseño como a la aplicación, permitió catalogar el sistema como adecuado para una primera fase de solución, señalándose que tiene un grupo de oportunidades de mejora, algunas de las cuales se incluyen en las recomendaciones del presente trabajo.

## RECOMENDACIONES

La solución desarrollada aunque es un acercamiento completo a la solución del problema planteado es mejorable en muchos sentidos. A continuación se detallan algunos elementos que se recomienda abordar:

- Implementar un sistema cliente que se pueda conectar al núcleo desarrollado para poder realizar una validación completa del ciclo de funcionamiento propuesto.
- Refinar la concepción del módulo de Documentos Digitales, que en este momento solo admite definir que documentos interactúan con que procesos, sin que se puedan detallar y modificar el formato de dichos documentos.
- Desarrollar un editor de interfaces visuales, que facilite el diseño de los ficheros de Lenguaje de Interface de Usuario basado en XML (XUL) que soportan la descripción de las interfaces de usuario de cada acción de los procesos.

## BIBLIOGRAFÍA

1. Alvarado, P., & Laitano, R. *Modelo de Bases de Datos con ER\Studio*.
2. Downey, A., Elkne, J., & Meyers, C. (2002). *Aprenda a Pensar Como un Programador con Python*. Wellesley, Massachusetts: Green Tea Press.
3. González Duque, R. *Python para Todos*.
4. UNCTAD. (n.d.). *ASYCUDA*. Retrieved from <http://www.asycuda.org/spanish/asyversions.asp>
5. Korea Custom System. (n.d.). *Korea Custom System*. Retrieved from <http://english.customs.go.kr>
6. U.S Custom Border Protection. (n.d.). *CBP.gov*. Retrieved from <http://www.cbp.gov>
7. HM Revenue & Customs. (n.d.). *HM Revenue & Customs*. Retrieved from [http://customs.hmrc.gov.uk/channelsPortalWebApp/channelsPortalWebApp.portal?\\_nfpb=true&pageLabel=pagelImport\\_ShowContent&propertyType=document&featurearticle=true&id=HMCE\\_CL\\_001614#P4\\_73](http://customs.hmrc.gov.uk/channelsPortalWebApp/channelsPortalWebApp.portal?_nfpb=true&pageLabel=pagelImport_ShowContent&propertyType=document&featurearticle=true&id=HMCE_CL_001614#P4_73)
8. ZOLL. (n.d.). *ZOLL*. Retrieved from [http://www.zoll.de/b0\\_zoll\\_und\\_steuern/a0\\_zoelle/c0\\_zollanmeldung/d10\\_atlas/index.html](http://www.zoll.de/b0_zoll_und_steuern/a0_zoelle/c0_zollanmeldung/d10_atlas/index.html)
9. Portland Pattern Repository. (n.d.). *Portland Pattern Repository*. Retrieved from <http://c2.com/cgi/wiki?FacadePattern>
10. Portland Pattern Repository. (n.d.). *Portland Pattern Repository*. Retrieved from <http://c2.com/cgi/wiki?DecoratorPattern>
11. Hudson Continuous Integration. (n.d.). *Hudson Continuous Integration*. Retrieved from <http://javadoc.jenkins-ci.org/hudson/model/Descriptor.html>
12. OpenOffice.org. (n.d.). *OpenOffice.org*. Retrieved from [http://wiki.services.openoffice.org/wiki/Documentation/DevGuide/Database/The\\_Descriptor\\_Pattern](http://wiki.services.openoffice.org/wiki/Documentation/DevGuide/Database/The_Descriptor_Pattern)
13. JAVAWORLD. (n.d.). *JAVAWORLD*. Retrieved from <http://www.javaworld.com/javaworld/jw-01-2004/jw-0102-toolbox.html>
14. Oracle. (n.d.). Retrieved from <http://java.about.com/od/gettingstarted/a/whatisjava.htm>
15. Nadkarni, P. (2009). *The EAV/CR Model of Data Representation*.
16. MySQL. (n.d.). *MySQL*. Retrieved from <http://dev.mysql.com/doc/refman/5.0/es/introduction.html>
17. Price, J. (2004). *Oracle Database 10g SQL*. McGraw-Hill/Osborne.
18. Lockhart, T. *Tutorial de PostgreSQL*.

## REFERENCIAS BIBLIOGRÁFICAS

1. **Management Decision and Research Center.** Organizational Change Premier. [En línea] <http://www.hsr.d.research.va.gov>.
2. **Hurwitz, Judith.** *Service Oriented Architecture for Dummies*. Indianapolis, Indiana : Wiley Publishing, Inc, 2007.
3. **Software AG.** Software AG webMethods. [En línea] [http://www.soa-know-how.de/fileadmin/articles/715\\_SAG\\_webMethods\\_Suite.pdf](http://www.soa-know-how.de/fileadmin/articles/715_SAG_webMethods_Suite.pdf).
4. **Oracle.** Oracle SOA Suite. [En línea] <http://www.oracle.com/technetwork/es/documentation/317519-esa.pdf>.
5. **UNCTAD.** ASYCUDA. [En línea] <http://www.asycuda.org/spanish/asyversions.asp>.
6. **Korea Custom System.** Korea Custom System. [En línea] <http://english.customs.go.kr>.
7. **U.S Custom Border Protection.** CBP.gov. [En línea] <http://www.cbp.gov>.
8. **HM Revenue & Customs.** HM Revenue & Customs. [En línea] [http://customs.hmrc.gov.uk/channelsPortalWebApp/channelsPortalWebApp.portal?\\_nfpb=true&pageLabel=pagelImport\\_ShowContent&propertyType=document&featurearticle=true&id=HMCE\\_CL\\_001614#P4\\_73](http://customs.hmrc.gov.uk/channelsPortalWebApp/channelsPortalWebApp.portal?_nfpb=true&pageLabel=pagelImport_ShowContent&propertyType=document&featurearticle=true&id=HMCE_CL_001614#P4_73).
9. **ZOLL.** ZOLL. [En línea] [http://www.zoll.de/b0\\_zoll\\_und\\_steuern/a0\\_zoelle/c0\\_zollanmeldung/d10\\_atlas/index.html](http://www.zoll.de/b0_zoll_und_steuern/a0_zoelle/c0_zollanmeldung/d10_atlas/index.html).
10. **Portland Pattern Repository.** Portland Pattern Repository. [En línea] <http://c2.com/cgi/wiki?FacadePattern>.
11. **Hudson Continuous Integration.** Hudson Continuous Integration. [En línea] <http://javadoc.jenkins-ci.org/hudson/model/Descriptor.html>.
12. **OpenOffice.org.** OpenOffice.org. [En línea] [http://wiki.services.openoffice.org/wiki/Documentation/DevGuide/Database/The\\_Descriptor\\_Pattern](http://wiki.services.openoffice.org/wiki/Documentation/DevGuide/Database/The_Descriptor_Pattern)
13. **Nadkarni, Prakash.** *The EAV/CR Model of Data Representation*. 2009.
14. **cplusplus.com.** cplusplus.com. [En línea] <http://www.cplusplus.com/info/description>.
15. **Oracle.** Oracle. [En línea] <http://java.about.com/od/gettingstarted/a/whatisjava.htm>.
16. **González Duque, Raúl.** *Python para Todos*.
17. **MySQL.** MySQL. [En línea] <http://dev.mysql.com/doc/refman/5.0/es/introduction.html>.
18. **Price, J.** *Oracle Database 10g SQL*. s.l. : McGraw-Hill/Osborne, 2004.
19. **Lockhart, Thomas.** *Tutorial de PostgreSQL*.
20. **Alvarado, Pedro y Laitano, Rommel.** *Modelo de Bases de Datos con ER\Studio*.
21. **van Rossum, Guido y Warsaw, Barry.** Style Guide for Python Code. [En línea] <http://www.python.org/dev/peps/pep-0008>

## ANEXOS

### Anexo I. Descripción requisitos funcionales.

Tabla de listado de los requisitos funcionales

No	Funcionalidad	Descripción	Complejidad	Prioridad
1	Gestionar Objeto	Se gestiona la información básica de todos los objetos que maneja la solución. Se permite crear objetos nuevos, eliminar objetos y modificar la información referente a un objeto existente.	Baja	Requerido
3	Gestionar Atributo	Se gestiona la información básica de todos los atributos que maneja la solución. Se permite crear atributos nuevos, eliminar atributos y modificar la información referente a un atributo existente.	Baja	Requerido
3	Gestionar Entidad	Se gestiona la información específica de todas las entidades que maneja la solución. Se permite crear entidades nuevas, eliminar entidades y modificar la información referente a una entidad existente.	Baja	Requerido
4	Gestionar Relación	Se gestiona la información específica de todas las relaciones entre entidades que maneja la solución. Se permite crear relaciones nuevas, eliminar relaciones y modificar la información referente a una relación existente.	Media	Requerido
5	Gestionar Acción	Se gestionan los tipos de acciones que se pueden desarrollar a lo largo de un proceso de los que se pretende informatizar. Además si la acción es de tipo Validación se registran los datos específicos para este tipo de acción.	Media	Requerido.
6	Gestionar Proceso	Se gestiona la información básica de todos los procesos que maneja la solución. Se permite crear	Alta	Requerido

		procesos nuevos, eliminar objetos y modificar la información referente a un objeto existente. De cada proceso se registran las acciones que contiene y la secuencia de ejecución de las mismas.		
7	Publicar proceso	Permite definir alguno de los procesos definidos dentro de la solución como público. Luego de que esté público el proceso se debe permitir su ejecución desde una estación remota.	Media	Requerido
8	Ejecutar proceso	Permite que un proceso que haya sido solicitado para su ejecución desde una estación remota responda a la secuencia que tiene definida e interactúe con los datos que requiere cada una de las acciones de la misma forma que envía hacia el cliente la respuesta correspondiente a cada acción en función de conseguir el resultado final que espera el usuario.	Alta	Requerido

**Requisito 1.1: Crear objeto**  
**Descripción textual del requisito**

<b>Precondiciones</b>		
<b>Flujo de eventos</b>		
<b>Flujo básico Crear objeto</b>		
<b>No.</b>	<b>Actor</b>	<b>Sistema</b>
1.	Seleccionar la opción Crear objeto en la pantalla de Gestión de objetos	
2.		Muestra la pantalla para entrar los datos del nuevo objeto (nombre, descripción y tipo)
3.	Introduce los datos solicitados en el formulario y selecciona la opción Save.	
4.		Valida los datos proporcionados por el usuario para el nuevo objeto a crear En caso de identificarse errores en la validación ver Flujo alternativo 1: Error en los datos suministrados
5.		Registra la información en el esquema de

		configuración de la BD del sistema generando un identificador para el nuevo objeto
6.		Muestra un mensaje de operación satisfactoria
<b>Pos-condiciones</b>		
1.	En el sistema queda registrado un nuevo objeto	
<b>Flujos alternativos</b>		
<b>Flujo alternativo 1: Error en los datos suministrados</b>		
<b>No.</b>	<b>Actor</b>	<b>Sistema</b>
1.		Muestra la pantalla de entrada de datos nuevamente señalando los datos con errores
2.	Corrige los errores cometidos y selecciona la opción Save.	
3.		Continúa en el paso 4 del Flujo básico Crear objeto
<b>Pos-condiciones</b>		
1.	En el sistema queda registrado un nuevo objeto	
<b>Validaciones</b>		
1.		
<b>Conceptos</b>		No aplica
		No aplica
<b>Relaciones</b>	<b>Requisitos Incluidos</b>	Ninguno
	<b>Requisitos Excluidos</b>	Ninguno
<b>Requisitos especiales</b>	No aplica	
<b>Asuntos pendientes</b>	No aplica	

## Prototipo elemental de interfaz gráfica de usuario

GINA v2.0 administration
Welcome, **Juan Ernesto**. [Change password](#) / [Log out](#)

[Home](#) > [Entity\\_management](#) > [Objetos](#) > [Add Objeto](#)

### Add Objeto

Tipo:

Nombre:

Descripcion:

### Formatos de Entrada

No aplica

### Formatos de Salida

No aplica

**Requisito 1.2: Eliminar objeto**  
**Descripción textual del requisito**

<b>Precondiciones</b>		El objeto que se desea eliminar debe existir en la BD del sistema
<b>Flujo de eventos</b>		
<b>Flujo básico Eliminar objeto</b>		
No.	Actor	Sistema
1.	Selecciona un objeto del listado de objetos que se muestra en la pantalla de Gestión de objetos y luego selecciona la opción Delete	
2.		Elimina la información del objeto seleccionado del esquema de configuración de la BD
3.		Muestra un mensaje de operación satisfactoria
<b>Pos-condiciones</b>		
1.	En el sistema no existe la información relativa al objeto eliminado	
<b>Validaciones</b>		
1.		
<b>Conceptos</b>		No aplica
		No aplica
<b>Relaciones</b>	<b>Requisitos Incluidos</b>	Ninguno
	<b>Requisitos Excluidos</b>	Ninguno
<b>Requisitos especiales</b>	No aplica	
<b>Asuntos pendientes</b>	No aplica	

**Prototipo elemental de interfaz gráfica de usuario**

The screenshot shows the 'Change Objeto' form in the GINA v2.0 administration system. The form has the following fields and controls:

- Tipo:** A dropdown menu with 'Proceso' selected.
- Nombre:** A text input field containing 'MTI'.
- Descripcion:** A text input field containing 'Medios de Transporte Internacional'.
- Buttons:** At the bottom, there is a 'Delete' button (circled in red), a 'Save and add another' button, a 'Save and continue editing' button, and a 'Save' button.

**Formatos de Entrada**

No aplica

**Formatos de Salida**

No aplica

### Requisito 1.3: Modificar datos de objeto

#### Descripción textual del requisito

<b>Precondiciones</b>		El objeto que se desea modificar debe existir en la BD del sistema
<b>Flujo de eventos</b>		
<b>Flujo básico Modificar datos de objeto</b>		
<b>No.</b>	<b>Actor</b>	<b>Sistema</b>
1.	Selecciona un objeto del listado de objetos que se muestra en la pantalla de Gestión de objetos y luego selecciona la opción Modificar	
2.		Muestra una pantalla donde aparecen los datos actuales del objeto a modificar (nombre, descripción y tipo)
3.	Modifica los datos que desea en el formulario y selecciona la opción Save	
4.		Valida los datos proporcionados por el usuario para la modificación del objeto En caso de identificarse errores en la validación ver Flujo alternativo 1: Error en los datos suministrados
5.		Registra la información modificada en el esquema de configuración de la BD del sistema
6.		Muestra un mensaje de operación satisfactoria
<b>Pos-condiciones</b>		
1.	En el sistema la información del objeto está actualizada con las modificaciones	
<b>Flujos alternativos</b>		
<b>Flujo alternativo 1: Error en los datos suministrados</b>		
<b>No.</b>	<b>Actor</b>	<b>Sistema</b>
1.		Muestra la pantalla de entrada de datos nuevamente señalando los datos con errores
2.	Corrige los errores cometidos y selecciona la opción Save	
3.		Continúa en el paso 4 del Flujo básico Modificar datos de objeto
<b>Pos-condiciones</b>		
1.	En el sistema la información del objeto está actualizada con las modificaciones	
<b>Validaciones</b>		
1.		
<b>Conceptos</b>		No aplica
		No aplica
<b>Relaciones</b>	<b>Requisitos Incluidos</b>	Ninguno
	<b>Requisitos Excluidos</b>	Ninguno

<b>Requisitos especiales</b>	No aplica
<b>Asuntos pendientes</b>	No aplica

## Prototipo elemental de interfaz gráfica de usuario

GINA v2.0 administration
Welcome, **Juan Ernesto**. [Change password](#) / [Log out](#)

Home > Entity\_management > Objetos > MTI - Proceso

### Change Objeto History

Tipo: Proceso

Nombre: MTI

Descripción: Medios de Transporte Internacional

✖ Delete
Save and add another
Save and continue editing
Save

### Formatos de Entrada

No aplica

### Formatos de Salida

No aplica

### Requisito 2.1: Crear atributo

#### Descripción textual del requisito

Precondiciones		
Flujo de eventos		
Flujo básico Crear atributo		
No.	Actor	Sistema
7.	Seleccionar la opción Crear atributo en la pantalla de Gestión de atributos	
8.		Muestra la pantalla para entrar los datos del nuevo atributo (nombre, descripción, tipo de dato, si es nulo o no y tamaño)
9.	Introduce los datos solicitados en el formulario y selecciona la opción Save.	
10.		Valida los datos proporcionados por el usuario para el nuevo atributo a crear En caso de identificarse errores en la validación ver Flujo alternativo 1: Error en los datos suministrados
11.		Registra la información en el esquema de configuración de la BD del sistema generando un identificador para el nuevo atributo
12.		Muestra un mensaje de operación satisfactoria
Pos-condiciones		

2.	En el sistema queda registrado un nuevo atributo	
<b>Flujos alternativos</b>		
<b>Flujo alternativo 1: Error en los datos suministrados</b>		
<b>No.</b>	<b>Actor</b>	<b>Sistema</b>
4.		Muestra la pantalla de entrada de datos nuevamente señalando los datos con errores
5.	Corrige los errores cometidos y selecciona la opción Save.	
6.		Continúa en el paso 4 del Flujo básico Crear atributo
<b>Pos-condiciones</b>		
2.	En el sistema queda registrado un nuevo atributo	
<b>Validaciones</b>		
2.		
<b>Conceptos</b>		No aplica
		No aplica
<b>Relaciones</b>	<b>Requisitos Incluidos</b>	Ninguno
	<b>Requisitos Excluidos</b>	Ninguno
<b>Requisitos especiales</b>	No aplica	
<b>Asuntos pendientes</b>	No aplica	

## Prototipo elemental de interfaz gráfica de usuario

GINA v2.0 administration
Welcome, **Juan Ernesto**. [Change password](#) / [Log out](#)

[Home](#) > [Entity\\_management](#) > [Atributos](#) > [Add Atributo](#)

### Add Atributo

Nombre:

Descripcion:

Tipo de dato:

Tamaño:

Nulo?

### Formatos de Entrada

No aplica

### Formatos de Salida

No aplica

**Requisito 2.2: Eliminar atributo**  
**Descripción textual del requisito**

<b>Precondiciones</b>		El atributo que se desea eliminar debe existir en la BD del sistema
<b>Flujo de eventos</b>		
<b>Flujo básico Eliminar atributo</b>		
No.	Actor	Sistema
4.	Selecciona un atributo del listado de atributos que se muestra en la pantalla de Gestión de atributos y luego selecciona la opción Delete	
5.		Elimina la información del atributo seleccionado del esquema de configuración de la BD
6.		Muestra un mensaje de operación satisfactoria
<b>Pos-condiciones</b>		
2.	En el sistema no existe la información relativa al atributo eliminado	
<b>Validaciones</b>		
2.		
<b>Conceptos</b>		No aplica
		No aplica
<b>Relaciones</b>	<b>Requisitos Incluidos</b>	Ninguno
	<b>Requisitos Excluidos</b>	Ninguno
<b>Requisitos especiales</b>	No aplica	
<b>Asuntos pendientes</b>	No aplica	

**Prototipo elemental de interfaz gráfica de usuario**

The screenshot shows a web interface for 'GINA v2.0 administration'. The user is logged in as 'Juan Ernesto'. The breadcrumb trail is 'Home > Entity\_management > Atributos > nombre'. The main heading is 'Change Atributo' with a 'History' button. The form contains the following fields:
 

- Nombre:** nombre
- Descripcion:** nombre de la mercancia
- Tipo de dato:** Varchar
- Tamaño:** 50
- Nulo?

 At the bottom left, there is a red 'Delete' button. At the bottom right, there are three buttons: 'Save and add another', 'Save and continue editing', and 'Save'.

**Formatos de Entrada**

No aplica

**Formatos de Salida**

No aplica

### Requisito 2.3: Modificar datos de atributo

#### Descripción textual del requisito

<b>Precondiciones</b>		El atributo que se desea modificar debe existir en la BD del sistema
<b>Flujo de eventos</b>		
<b>Flujo básico Modificar datos de atributo</b>		
No.	Actor	Sistema
7.	Selecciona un atributo del listado de atributos que se muestra en la pantalla de Gestión de atributos y luego selecciona la opción Modificar	
8.		Muestra una pantalla donde aparecen los datos actuales del atributo a modificar (nombre, descripción, tipo de dato, si es nulo o no y tamaño)
9.	Modifica los datos que desea en el formulario y selecciona la opción Save	
10.		Valida los datos proporcionados por el usuario para la modificación del atributo En caso de identificarse errores en la validación ver Flujo alternativo 1: Error en los datos suministrados
11.		Registra la información modificada en el esquema de configuración de la BD del sistema
12.		Muestra un mensaje de operación satisfactoria
<b>Pos-condiciones</b>		
2.	En el sistema la información del atributo está actualizada con las modificaciones	
<b>Flujos alternativos</b>		
<b>Flujo alternativo 1: Error en los datos suministrados</b>		
No.	Actor	Sistema
4.		Muestra la pantalla de entrada de datos nuevamente señalando los datos con errores
5.	Corrige los errores cometidos y selecciona la opción Save	
6.		Continúa en el paso 4 del Flujo básico Modificar datos de atributo
<b>Pos-condiciones</b>		
2.	En el sistema la información del atributo está actualizada con las modificaciones	
<b>Validaciones</b>		
2.		
<b>Conceptos</b>		No aplica
		No aplica
<b>Relaciones</b>	<b>Requisitos Incluidos</b>	Ninguno

	<b>Requisitos Excluidos</b>	Ninguno
<b>Requisitos especiales</b>	No aplica	
<b>Asuntos pendientes</b>	No aplica	

## Prototipo elemental de interfaz gráfica de usuario

GINA v2.0 administration
Welcome, **Juan Ernesto**. [Change password](#) / [Log out](#)

[Home](#) > [Entity\\_management](#) > [Atributos](#) > nombre

**Change Atributo**
History

**Nombre:**

**Descripción:**

**Tipo de dato:** Varchar

**Tamaño:**

Nulo?

✖ Delete
Save and add another
Save and continue editing
Save

### Formatos de Entrada

No aplica

### Formatos de Salida

No aplica

### Requisito 3.1: Crear entidad

#### Descripción textual del requisito

<b>Precondiciones</b>	Debe existir en la BD del sistema objetos de tipo entidad.	
<b>Flujo de eventos</b>		
<b>Flujo básico Crear entidad</b>		
No.	Actor	Sistema
13.	Seleccionar la opción Crear entidad en la pantalla de Gestión de entidades	
14.		Muestra la pantalla para entrar los datos de la nueva entidad (nombre objeto y nombre tabla)
15.	Introduce los datos solicitados en el formulario y selecciona la opción Save.	
16.		Valida los datos proporcionados por el usuario para la nueva entidad a crear En caso de identificarse errores en la validación ver Flujo alternativo 1: Error en los datos suministrados
17.		Registra la información en el esquema de

		configuración de la BD del sistema.
18.		Muestra un mensaje de operación satisfactoria
<b>Pos-condiciones</b>		
3.	En el sistema queda registrado una nueva entidad	
<b>Flujos alternativos</b>		
<b>Flujo alternativo 1: Error en los datos suministrados</b>		
<b>No.</b>	<b>Actor</b>	<b>Sistema</b>
7.		Muestra la pantalla de entrada de datos nuevamente señalando los datos con errores
8.	Corrige los errores cometidos y selecciona la opción Save.	
9.		Continúa en el paso 4 del Flujo básico Crear entidad
<b>Pos-condiciones</b>		
3.	En el sistema queda registrado una nueva entidad	
<b>Validaciones</b>		
3.		
<b>Conceptos</b>		No aplica
		No aplica
<b>Relaciones</b>	<b>Requisitos Incluidos</b>	Ninguno
	<b>Requisitos Excluidos</b>	Ninguno
<b>Requisitos especiales</b>	No aplica	
<b>Asuntos pendientes</b>	No aplica	

## Prototipo elemental de interfaz gráfica de usuario

### Add Entidad

Objeto:  +

Nombre  
Tabla:

Atributos:

**Available Atributos**

Q

nombre  
descripcion  
id\_mercancia  
UM

+  
+

Choose all

**Chosen Atributos** +

Select your choice(s) and click +

Clear all

### Formatos de Entrada

No aplica

### Formatos de Salida

No aplica

### Requisito 3.2: Eliminar entidad

#### Descripción textual del requisito

<b>Precondiciones</b>	La entidad que se desea eliminar debe existir en la BD del sistema	
<b>Flujo de eventos</b>		
<b>Flujo básico Eliminar entidad</b>		
<b>No.</b>	<b>Actor</b>	<b>Sistema</b>
7.	Selecciona una entidad del listado de entidades que se muestra en la pantalla de Gestión de entidades y luego selecciona la opción Delete	
8.		Elimina la información de la entidad seleccionada del esquema de configuración de la BD
9.		Muestra un mensaje de operación satisfactoria
<b>Pos-condiciones</b>		
3.	En el sistema no existe la información relativa a la entidad eliminada	
<b>Validaciones</b>		
3.		
<b>Conceptos</b>	No aplica	

		No aplica
<b>Relaciones</b>	<b>Requisitos Incluidos</b>	Ninguno
	<b>Requisitos Excluidos</b>	Ninguno
<b>Requisitos especiales</b>	No aplica	
<b>Asuntos pendientes</b>	No aplica	

## Prototipo elemental de interfaz gráfica de usuario

GINA v2.0 administration

Welcome, **Juan Ernesto**. Change password / Log out

Home > Entity\_management > Entidades > DM - Proceso

### Change Entidad

History

Objeto: DM - Proceso +

Nombre Tabla:

Atributos:

**Available Atributos**

descripcion  
UM

+ -

Choose all

**Chosen Atributos** +

Select your choice(s) and click +

nombre  
id\_mercancia

+ -

Clear all

Delete Save and add another Save and continue editing Save

### Formatos de Entrada

No aplica

### Formatos de Salida

No aplica

### Requisito 3.3: Modificar datos de entidad

#### Descripción textual del requisito

<b>Precondiciones</b>	La entidad que se desea modificar debe existir en la BD del sistema	
<b>Flujo de eventos</b>		
<b>Flujo básico Modificar datos de entidad</b>		
<b>No.</b>	<b>Actor</b>	<b>Sistema</b>
13.	Selecciona una entidad del listado de entidades que se muestra en la pantalla de Gestión de entidades y luego selecciona la opción Modificar	
14.		Muestra una pantalla donde aparecen los datos

		actuales de la relación a modificar (nombre objeto y nombre tabla)
15.	Modifica los datos que desea en el formulario y selecciona la opción Save	
16.		Valida los datos proporcionados por el usuario para la modificación de la entidad En caso de identificarse errores en la validación ver Flujo alternativo 1: Error en los datos suministrados
17.		Registra la información modificada en el esquema de configuración de la BD del sistema
18.		Muestra un mensaje de operación satisfactoria
<b>Pos-condiciones</b>		
3.	En el sistema la información de la entidad está actualizada con las modificaciones	
<b>Flujos alternativos</b>		
<b>Flujo alternativo 1: Error en los datos suministrados</b>		
<b>No.</b>	<b>Actor</b>	<b>Sistema</b>
7.		Muestra la pantalla de entrada de datos nuevamente señalando los datos con errores
8.	Corrige los errores cometidos y selecciona la opción Save	
9.		Continúa en el paso 4 del Flujo básico Modificar datos de la entidad
<b>Pos-condiciones</b>		
3.	En el sistema la información de la entidad está actualizada con las modificaciones	
<b>Validaciones</b>		
3.		
<b>Conceptos</b>		No aplica
		No aplica
<b>Relaciones</b>	<b>Requisitos Incluidos</b>	Ninguno
	<b>Requisitos Excluidos</b>	Ninguno
<b>Requisitos especiales</b>	No aplica	
<b>Asuntos pendientes</b>	No aplica	

## Prototipo elemental de interfaz gráfica de usuario

### Change Entidad

History

Objeto: DM - Proceso +

Nombre Tabla:

Atributos:

**Available Atributos**

descripcion  
UM

+  
+

Choose all

**Chosen Atributos** +

Select your choice(s) and click +

nombre  
id\_mercancia

Clear all

Delete Save and add another Save and continue editing Save

### Formatos de Entrada

No aplica

### Formatos de Salida

No aplica

### Requisito 4.1: Crear relación

#### Descripción textual del requisito

<b>Precondiciones</b>	Debe existir en la BD del sistema objetos.	
<b>Flujo de eventos</b>		
<b>Flujo básico Crear relación</b>		
<b>No.</b>	<b>Actor</b>	<b>Sistema</b>
19.	Seleccionar la opción Crear relación en la pantalla de Gestión de relaciones	
20.		Muestra la pantalla para entrar los datos de la nueva relación (nombre objeto padre, nombre objeto hijo y tipo de relación)
21.	Introduce los datos solicitados en el formulario y selecciona la opción Save.	
22.		Valida los datos proporcionados por el usuario para la nueva relación a crear En caso de identificarse errores en la validación ver Flujo alternativo 1: Error en los datos suministrados

23.		Registra la información en el esquema de configuración de la BD del sistema.
24.		Muestra un mensaje de operación satisfactoria
<b>Pos-condiciones</b>		
4.	En el sistema queda registrado una nueva relación	
<b>Flujos alternativos</b>		
<b>Flujo alternativo 1: Error en los datos suministrados</b>		
<b>No.</b>	<b>Actor</b>	<b>Sistema</b>
10.		Muestra la pantalla de entrada de datos nuevamente señalando los datos con errores
11.	Corrige los errores cometidos y selecciona la opción Save.	
12.		Continúa en el paso 4 del Flujo básico Crear relación
<b>Pos-condiciones</b>		
4.	En el sistema queda registrado una nueva relación	
<b>Validaciones</b>		
4.		
<b>Conceptos</b>		No aplica
		No aplica
<b>Relaciones</b>	<b>Requisitos Incluidos</b>	Ninguno
	<b>Requisitos Excluidos</b>	Ninguno
<b>Requisitos especiales</b>	No aplica	
<b>Asuntos pendientes</b>	No aplica	

## Prototipo elemental de interfaz gráfica de usuario

GINA v2.0 administration

Welcome, Juan Ernesto. Change password / Log out

Home > Entity\_management > Relaciones > Add relacion

### Add relacion

Objeto Padre:	<input type="text"/>	<input style="color: green;" type="button" value="+"/>
Objeto Hijo:	<input type="text"/>	<input style="color: green;" type="button" value="+"/>
Tipo Relacion:	<input type="text"/>	

### Formatos de Entrada

No aplica

### Formatos de Salida

No aplica

**Requisito 4.2: Eliminar relación**  
**Descripción textual del requisito**

<b>Precondiciones</b>		La relación que se desea eliminar debe existir en la BD del sistema
<b>Flujo de eventos</b>		
<b>Flujo básico Eliminar relación</b>		
<b>No.</b>	<b>Actor</b>	<b>Sistema</b>
10.	Selecciona una relación del listado de relaciones que se muestra en la pantalla de Gestión de relaciones y luego selecciona la opción Delete	
11.		Elimina la información de la relación seleccionada del esquema de configuración de la BD
12.		Muestra un mensaje de operación satisfactoria
<b>Pos-condiciones</b>		
4.	En el sistema no existe la información relativa a la relación eliminada	
<b>Validaciones</b>		
4.		
<b>Conceptos</b>		No aplica
		No aplica
<b>Relaciones</b>	<b>Requisitos Incluidos</b>	Ninguno
	<b>Requisitos Excluidos</b>	Ninguno
<b>Requisitos especiales</b>	No aplica	
<b>Asuntos pendientes</b>	No aplica	

**Prototipo elemental de interfaz gráfica de usuario**

The screenshot shows the 'Change relacion' form in the GINA v2.0 administration system. The form has three main sections: 'Objeto Padre' with a dropdown menu set to 'MTI - Proceso', 'Objeto Hijo' with a dropdown menu set to 'DM - Proceso', and 'Tipo Relacion' with a dropdown menu set to 'Uno a Uno'. At the bottom of the form, there is a 'Delete' button circled in red, and three other buttons: 'Save and add another', 'Save and continue editing', and 'Save'. The top of the page shows the user is logged in as 'Juan Ernesto' and has options to 'Change password' or 'Log out'.

**Formatos de Entrada**

No aplica

**Formatos de Salida**

No aplica

### Requisito 4.3: Modificar datos de relación

#### Descripción textual del requisito

<b>Precondiciones</b>		La relación que se desea modificar debe existir en la BD del sistema
<b>Flujo de eventos</b>		
<b>Flujo básico Modificar datos de relación</b>		
No.	Actor	Sistema
19.	Selecciona una relación del listado de relaciones que se muestra en la pantalla de Gestión de relaciones y luego selecciona la opción Modificar	
20.		Muestra una pantalla donde aparecen los datos actuales de la relación a modificar (nombre objeto padre, nombre objeto hijo y tipo de relación)
21.	Modifica los datos que desea en el formulario y selecciona la opción Save	
22.		Valida los datos proporcionados por el usuario para la modificación de la relación En caso de identificarse errores en la validación ver Flujo alternativo 1: Error en los datos suministrados
23.		Registra la información modificada en el esquema de configuración de la BD del sistema
24.		Muestra un mensaje de operación satisfactoria
<b>Pos-condiciones</b>		
4.	En el sistema la información de la relación está actualizada con las modificaciones	
<b>Flujos alternativos</b>		
<b>Flujo alternativo 1: Error en los datos suministrados</b>		
No.	Actor	Sistema
10.		Muestra la pantalla de entrada de datos nuevamente señalando los datos con errores
11.	Corrige los errores cometidos y selecciona la opción Save	
12.		Continúa en el paso 4 del Flujo básico Modificar datos de la relación
<b>Pos-condiciones</b>		
4.	En el sistema la información de la relación está actualizada con las modificaciones	
<b>Validaciones</b>		
4.		
<b>Conceptos</b>		No aplica
		No aplica
<b>Relaciones</b>	<b>Requisitos Incluidos</b>	Ninguno

	<b>Requisitos Excluidos</b>	Ninguno
<b>Requisitos especiales</b>	No aplica	
<b>Asuntos pendientes</b>	No aplica	

## Prototipo elemental de interfaz gráfica de usuario

**GINA v2.0 administration** Welcome, **Juan Ernesto**. [Change password](#) / [Log out](#)

[Home](#) > [Entity\\_management](#) > [Relaciones](#) > [MTI - Proceso - DM - Proceso](#)

### Change relacion History

Objeto Padre:	MTI - Proceso	+	
Objeto Hijo:	DM - Proceso	+	
Tipo Relacion:	Uno a Uno	v	

✖ Delete

### Formatos de Entrada

No aplica

### Formatos de Salida

No aplica

### Requisito 5.1: Crear acción

#### Descripción textual del requisito

<b>Precondiciones</b>		
<b>Flujo de eventos</b>		
<b>Flujo básico Crear acción</b>		
No.	Actor	Sistema
25.	Seleccionar la opción Crear acción en la pantalla de Gestión de acciones	
26.		Muestra la pantalla para entrar los datos de la nueva acción (nombre proceso, entrada, salida y tipo de acción)
27.	Introduce los datos solicitados en el formulario y selecciona la opción Save.	
28.		Valida los datos proporcionados por el usuario para la nueva acción a crear En caso de identificarse errores en la validación ver Flujo alternativo 1: Error en los datos suministrados
29.		Registra la información en el esquema de configuración de la BD del sistema generando un identificador para la nueva acción

30.		Muestra un mensaje de operación satisfactoria
<b>Pos-condiciones</b>		
5.	En el sistema queda registrado una nueva acción	
<b>Flujos alternativos</b>		
<b>Flujo alternativo 1: Error en los datos suministrados</b>		
<b>No.</b>	<b>Actor</b>	<b>Sistema</b>
13.		Muestra la pantalla de entrada de datos nuevamente señalando los datos con errores
14.	Corrige los errores cometidos y selecciona la opción Save.	
15.		Continúa en el paso 4 del Flujo básico Crear acción
<b>Pos-condiciones</b>		
5.	En el sistema queda registrado una nueva acción	
<b>Validaciones</b>		
5.		
<b>Conceptos</b>		No aplica
		No aplica
<b>Relaciones</b>	<b>Requisitos Incluidos</b>	Ninguno
	<b>Requisitos Excluidos</b>	Ninguno
<b>Requisitos especiales</b>	No aplica	
<b>Asuntos pendientes</b>	No aplica	

## Prototipo elemental de interfaz gráfica de usuario

GINA v2.0 administration
Welcome, **Juan Ernesto**. [Change password](#) / [Log out](#)

Home > Entity\_management > Acciones > Add Accion

### Add Accion

Proceso:	<input type="text" value="-----"/>	+	
Tipo:	<input type="text" value="-----"/>		
Entrada:	<input type="text"/>		
Salida:	<input type="text"/>		

### Formatos de Entrada

No aplica

### Formatos de Salida

No aplica

### Requisito 5.1.1: Crear acción validación

#### Descripción textual del requisito

<b>Precondiciones</b>		Debe existir en la BD del sistema acciones de tipo validación.
<b>Flujo de eventos</b>		
<b>Flujo básico Crear acción validación</b>		
<b>No.</b>	<b>Actor</b>	<b>Sistema</b>
31.	Seleccionar la opción Crear acción de validación en la pantalla de Gestión de acciones de validación	
32.		Muestra la pantalla para entrar los datos de la nueva acción de validación (nombre acción, operador y valor)
33.	Introduce los datos solicitados en el formulario y selecciona la opción Save.	
34.		Valida los datos proporcionados por el usuario para la nueva acción de validación a crear En caso de identificarse errores en la validación ver Flujo alternativo 1: Error en los datos suministrados
35.		Registra la información en el esquema de configuración de la BD del sistema
36.		Muestra un mensaje de operación satisfactoria
<b>Pos-condiciones</b>		
6.	En el sistema queda registrado una nueva acción de validación	
<b>Flujos alternativos</b>		
<b>Flujo alternativo 1: Error en los datos suministrados</b>		
<b>No.</b>	<b>Actor</b>	<b>Sistema</b>
16.		Muestra la pantalla de entrada de datos nuevamente señalando los datos con errores
17.	Corrige los errores cometidos y selecciona la opción Save.	
18.		Continúa en el paso 4 del Flujo básico Crear acción de validación
<b>Pos-condiciones</b>		
6.	En el sistema queda registrado una nueva acción de validación	
<b>Validaciones</b>		
6.		
<b>Conceptos</b>		No aplica
		No aplica
<b>Relaciones</b>	<b>Requisitos Incluidos</b>	Ninguno
	<b>Requisitos Excluidos</b>	Ninguno
<b>Requisitos especiales</b>	No aplica	

<b>Asuntos pendientes</b>	No aplica
---------------------------	-----------

## Prototipo elemental de interfaz gráfica de usuario

GINA v2.0 administration
Welcome, **Juan Ernesto**. [Change password](#) / [Log out](#)

[Home](#) > [Entity\\_management](#) > [AccionesValidar](#) > [Add AccionValidar](#)

### Add AccionValidar

Accion:	<input type="text" value="-----"/> <span style="float: right;">+</span>
Operador:	<input type="text"/>
Valor:	<input type="text"/>

### Formatos de Entrada

No aplica

### Formatos de Salida

No aplica

### Requisito 5.2: Eliminar acción

#### Descripción textual del requisito

<b>Precondiciones</b>	La acción que se desea eliminar debe existir en la BD del sistema	
<b>Flujo de eventos</b>		
<b>Flujo básico Eliminar acción</b>		
<b>No.</b>	<b>Actor</b>	<b>Sistema</b>
13.	Selecciona una acción del listado de acciones que se muestra en la pantalla de Gestión de acciones y luego selecciona la opción Delete	
14.		Elimina la información de la acción seleccionada del esquema de configuración de la BD
15.		Muestra un mensaje de operación satisfactoria
<b>Pos-condiciones</b>		
5.	En el sistema no existe la información relativa a la acción eliminada	
<b>Validaciones</b>		
5.		
<b>Conceptos</b>		No aplica
<b>Relaciones</b>	<b>Requisitos Incluidos</b>	Ninguno
<b>Requisitos especiales</b>	<b>Requisitos Excluidos</b>	Ninguno
<b>Asuntos pendientes</b>	No aplica	

## Prototipo elemental de interfaz gráfica de usuario

**GINA v2.0 administration** Welcome, **Juan Ernesto**. [Change password](#) / [Log out](#)

Home > Entity\_management > Acciones > DM - Proceso - input

### Change Accion History

Proceso:

Tipo:

Entrada:

Salida:

Delete

### Formatos de Entrada

No aplica

### Formatos de Salida

No aplica

### Requisito 5.2.1: Eliminar acción validación

#### Descripción textual del requisito

<b>Precondiciones</b>	La acción validación que se desea eliminar debe existir en la BD del sistema	
<b>Flujo de eventos</b>		
<b>Flujo básico Eliminar validación</b>		
<b>No.</b>	<b>Actor</b>	<b>Sistema</b>
16.	Selecciona una acción de validación del listado de acciones de validación que se muestra en la pantalla de Gestión de acciones de validación y luego selecciona la opción Delete	
17.		Elimina la información de la acción validación seleccionada del esquema de configuración de la BD
18.		Muestra un mensaje de operación satisfactoria
<b>Pos-condiciones</b>		
6.	En el sistema no existe la información relativa a la acción validación eliminada	
<b>Validaciones</b>		
6.		
<b>Conceptos</b>		No aplica
		No aplica
<b>Relaciones</b>	<b>Requisitos Incluidos</b>	Ninguno
	<b>Requisitos Excluidos</b>	Ninguno
<b>Requisitos especiales</b>	No aplica	

<b>Asuntos pendientes</b>	No aplica
---------------------------	-----------

## Prototipo elemental de interfaz gráfica de usuario

GINA v2.0 administration

Welcome, **Juan Ernesto**. Change password / Log out

Home > Entity\_management > AccionesValidar > DM - Proceso - validate

### Change AccionValidar

History

Accion:  +

Operador:

Valor:

 Delete

Save and add another

Save and continue editing

Save

### Formatos de Entrada

No aplica

### Formatos de Salida

No aplica

### Requisito 5.3: Modificar datos de acción

#### Descripción textual del requisito

<b>Precondiciones</b>	La acción que se desea modificar debe existir en la BD del sistema	
<b>Flujo de eventos</b>		
<b>Flujo básico Modificar datos de acción</b>		
<b>No.</b>	<b>Actor</b>	<b>Sistema</b>
25.	Selecciona una acción del listado de acciones que se muestra en la pantalla de Gestión de acciones y luego selecciona la opción Modificar	
26.		Muestra una pantalla donde aparecen los datos actuales de la acción a modificar (nombre proceso, entrada, salida y tipo de acción)
27.	Modifica los datos que desea en el formulario y selecciona la opción Save	
28.		Valida los datos proporcionados por el usuario para la modificación de la acción En caso de identificarse errores en la validación ver Flujo alternativo 1: Error en los datos suministrados
29.		Registra la información modificada en el esquema de configuración de la BD del sistema
30.		Muestra un mensaje de operación satisfactoria
<b>Pos-condiciones</b>		

5.	En el sistema la información de la acción está actualizada con las modificaciones	
<b>Flujos alternativos</b>		
<b>Flujo alternativo 1: Error en los datos suministrados</b>		
<b>No.</b>	<b>Actor</b>	<b>Sistema</b>
13.		Muestra la pantalla de entrada de datos nuevamente señalando los datos con errores
14.	Corrige los errores cometidos y selecciona la opción Save	
15.		Continúa en el paso 4 del Flujo básico Modificar datos de la acción
<b>Pos-condiciones</b>		
5.	En el sistema la información de la acción está actualizada con las modificaciones	
<b>Validaciones</b>		
5.		
<b>Conceptos</b>		No aplica
		No aplica
<b>Relaciones</b>	<b>Requisitos Incluidos</b>	Ninguno
	<b>Requisitos Excluidos</b>	Ninguno
<b>Requisitos especiales</b>	No aplica	
<b>Asuntos pendientes</b>	No aplica	

## Prototipo elemental de interfaz gráfica de usuario

GINA v2.0 administration
Welcome, **Juan Ernesto**. [Change password](#) / [Log out](#)

[Home](#) > [Entity\\_management](#) > [Acciones](#) > [DM - Proceso - input](#)

**Change Accion**
History

Proceso: DM - Proceso +

Tipo: Input

Entrada:

Salida:

✖ Delete
Save and add another
Save and continue editing
Save

### Formatos de Entrada

No aplica

### Formatos de Salida

No aplica

### Requisito 5.3.1: Modificar datos de acción validación

#### Descripción textual del requisito

<b>Precondiciones</b>	La acción validación que se desea modificar debe existir en la BD del sistema
-----------------------	---

<b>Flujo de eventos</b>		
<b>Flujo básico Modificar datos de acción validación</b>		
<b>No.</b>	<b>Actor</b>	<b>Sistema</b>
31.	Selecciona una acción de validación del listado de acciones validación que se muestra en la pantalla de Gestión de acciones validación y luego selecciona la opción Modificar	
32.		Muestra una pantalla donde aparecen los datos actuales de la acción validación a modificar (nombre proceso, entrada, salida y tipo de acción)
33.	Modifica los datos que desea en el formulario y selecciona la opción Save	
34.		Valida los datos proporcionados por el usuario para la modificación de la acción validación En caso de identificarse errores en la validación ver Flujo alternativo 1: Error en los datos suministrados
35.		Registra la información modificada en el esquema de configuración de la BD del sistema
36.		Muestra un mensaje de operación satisfactoria
<b>Pos-condiciones</b>		
6.	En el sistema la información de la acción validación está actualizada con las modificaciones	
<b>Flujos alternativos</b>		
<b>Flujo alternativo 1: Error en los datos suministrados</b>		
<b>No.</b>	<b>Actor</b>	<b>Sistema</b>
16.		Muestra la pantalla de entrada de datos nuevamente señalando los datos con errores
17.	Corrige los errores cometidos y selecciona la opción Save	
18.		Continúa en el paso 4 del Flujo básico Modificar datos de la acción validación
<b>Pos-condiciones</b>		
6.	En el sistema la información de la acción validación está actualizada con las modificaciones	
<b>Validaciones</b>		
6.		
<b>Conceptos</b>		No aplica
		No aplica
<b>Relaciones</b>	<b>Requisitos Incluidos</b>	Ninguno
	<b>Requisitos Excluidos</b>	Ninguno

<b>Requisitos especiales</b>	No aplica
<b>Asuntos pendientes</b>	No aplica

## Prototipo elemental de interfaz gráfica de usuario

The screenshot shows a web interface for 'GINA v2.0 administration'. The user is logged in as 'Juan Ernesto'. The breadcrumb trail is 'Home > Entity\_management > AccionesValidar > DM - Proceso - validate'. The main section is titled 'Change AccionValidar' and contains a form with the following fields:
 

- Accion:** A dropdown menu with 'DM - Proceso - validate' selected and a plus icon.
- Operador:** A text input field containing '+'. There is a plus icon to the right of the field.
- Valor:** A text input field containing '10'.

 At the bottom of the form, there is a 'Delete' button with a red 'x' icon, and three 'Save' buttons: 'Save and add another', 'Save and continue editing', and a standard 'Save' button.

### Formatos de Entrada

No aplica

### Formatos de Salida

No aplica

## Requisito 6.1: Crear proceso

### Descripción textual del requisito

<b>Precondiciones</b>	Debe existir en la BD del sistema objetos de tipo proceso.	
<b>Flujo de eventos</b>		
<b>Flujo básico Crear proceso</b>		
<b>No.</b>	<b>Actor</b>	<b>Sistema</b>
37.	Seleccionar la opción Crear proceso en la pantalla de Gestión de procesos	
38.		Muestra la pantalla para entrar los datos del nuevo proceso (nombre objeto, si es público o no)
39.	Introduce los datos solicitados en el formulario y selecciona la opción Save.	
40.		Valida los datos proporcionados por el usuario para el nuevo proceso a crear En caso de identificarse errores en la validación ver Flujo alternativo 1: Error en los datos suministrados
41.		Registra la información en el esquema de configuración de la BD del sistema.
42.		Muestra un mensaje de operación satisfactoria
<b>Pos-condiciones</b>		
7.	En el sistema queda registrado un nuevo proceso	

Flujos alternativos		
Flujo alternativo 1: Error en los datos suministrados		
No.	Actor	Sistema
19.		Muestra la pantalla de entrada de datos nuevamente señalando los datos con errores
20.	Corrige los errores cometidos y selecciona la opción Save.	
21.		Continúa en el paso 4 del Flujo básico Crear proceso
Pos-condiciones		
7.	En el sistema queda registrado un nuevo proceso	
Validaciones		
7.		
Conceptos		No aplica
		No aplica
Relaciones	Requisitos Incluidos	Ninguno
	Requisitos Excluidos	Ninguno
Requisitos especiales	No aplica	
Asuntos pendientes	No aplica	

## Prototipo elemental de interfaz gráfica de usuario

GINA v2.0 administration
Welcome, **Juan Ernesto**. [Change password](#) / [Log out](#)

Home > Entity\_management > Procesos > Add Proceso

### Add Proceso

Objeto:  +

Publico?

### Formatos de Entrada

No aplica

### Formatos de Salida

No aplica

### Requisito 6.2: Eliminar proceso

#### Descripción textual del requisito

Precondiciones	El proceso que se desea eliminar debe existir en la BD del sistema	
Flujo de eventos		
Flujo básico Eliminar proceso		
No.	Actor	Sistema
19.	Selecciona un proceso del listado de	

	procesos que se muestra en la pantalla de Gestión de procesos y luego selecciona la opción Delete	
20.		Elimina la información del proceso seleccionado del esquema de configuración de la BD
21.		Muestra un mensaje de operación satisfactoria
<b>Pos-condiciones</b>		
7.	En el sistema no existe la información relativa al proceso eliminado	
<b>Validaciones</b>		
7.		
<b>Conceptos</b>		No aplica
		No aplica
<b>Relaciones</b>	<b>Requisitos Incluidos</b>	Ninguno
	<b>Requisitos Excluidos</b>	Ninguno
<b>Requisitos especiales</b>	No aplica	
<b>Asuntos pendientes</b>	No aplica	

## Prototipo elemental de interfaz gráfica de usuario

**GINA v2.0 administration** Welcome, **Juan Ernesto**. [Change password](#) / [Log out](#)

[Home](#) > [Entity\\_management](#) > [Procesos](#) > [DM - Proceso](#)

### Change Proceso History

Objeto:

Publico?

### Formatos de Entrada

No aplica

### Formatos de Salida

No aplica

### Requisito 6.3: Modificar datos de proceso

#### Descripción textual del requisito

<b>Precondiciones</b>	El proceso que se desea modificar debe existir en la BD del sistema	
<b>Flujo de eventos</b>		
<b>Flujo básico Modificar datos de proceso</b>		
No.	Actor	Sistema
37.	Selecciona un proceso del listado de procesos que se muestra en la pantalla de Gestión de procesos y luego selecciona la opción Modificar	

38.		Muestra una pantalla donde aparecen los datos actuales del proceso a modificar (nombre objeto, si es público o no)
39.	Modifica los datos que desea en el formulario y selecciona la opción Save	
40.		Valida los datos proporcionados por el usuario para la modificación del proceso En caso de identificarse errores en la validación ver Flujo alternativo 1: Error en los datos suministrados
41.		Registra la información modificada en el esquema de configuración de la BD del sistema
42.		Muestra un mensaje de operación satisfactoria
<b>Pos-condiciones</b>		
7.	En el sistema la información del proceso está actualizada con las modificaciones	
<b>Flujos alternativos</b>		
<b>Flujo alternativo 1: Error en los datos suministrados</b>		
<b>No.</b>	<b>Actor</b>	<b>Sistema</b>
19.		Muestra la pantalla de entrada de datos nuevamente señalando los datos con errores
20.	Corrige los errores cometidos y selecciona la opción Save	
21.		Continúa en el paso 4 del Flujo básico Modificar datos del proceso
<b>Pos-condiciones</b>		
7.	En el sistema la información del proceso está actualizada con las modificaciones	
<b>Validaciones</b>		
7.		
<b>Conceptos</b>		No aplica
		No aplica
<b>Relaciones</b>	<b>Requisitos Incluidos</b>	Ninguno
	<b>Requisitos Excluidos</b>	Ninguno
<b>Requisitos especiales</b>	No aplica	
<b>Asuntos pendientes</b>	No aplica	

## Prototipo elemental de interfaz gráfica de usuario

**GINA v2.0 administration** Welcome, **Juan Ernesto**. [Change password](#) / [Log out](#)

[Home](#) > [Entity\\_management](#) > [Procesos](#) > [DM - Proceso](#)

### Change Proceso History

Objeto: DM - Proceso +

Publico?

✖ Delete
Save and add another
Save and continue editing
Save

### Formatos de Entrada

No aplica

### Formatos de Salida

No aplica

## Requisito 7: Configurar secuencia de proceso

### Descripción textual del requisito

<b>Precondiciones</b>	El proceso que se desea configurar debe estar registrado al igual que las acciones que se le van a asociar	
<b>Flujo de eventos</b>		
<b>Flujo básico Configurar secuencia de proceso</b>		
<b>No.</b>	<b>Actor</b>	<b>Sistema</b>
1.	Selecciona un proceso del listado de procesos que se muestra en la pantalla de Gestión de procesos y luego selecciona la opción Configurar	
2.		Muestra una pantalla se muestran los datos del proceso a configurar así como los controles que permiten adicionar nuevas acciones al proceso
3.	Selecciona una de las acciones para agregarla al proceso. Si no desea agregar más acciones continúa en el paso 5 del Flujo básico Configurar secuencia de proceso	
4.		Si la acción seleccionada es de tipo Validación ver Flujo alternativo 1: Acción de Validación Sino registra la acción seleccionada a continuación de la acción anterior y continúa en el paso 3 del Flujo básico Configurar secuencia de proceso.
5.		Registra la secuencia de acciones asociadas al proceso seleccionado
6.		Muestra un mensaje de operación satisfactoria
<b>Pos-condiciones</b>		
1.	En el sistema la información del proceso está actualizada con las modificaciones	

<b>Flujos alternativos</b>		
<b>Flujo alternativo 1: Acción de validación</b>		
<b>No.</b>	<b>Actor</b>	<b>Sistema</b>
1.		Solicita que se configure la secuencia de acciones en caso que el resultado de la acción de Validación sea satisfactorio
2.	Continúa en el paso 3 del Flujo básico Configurar secuencia de proceso	
3.		Continúa en el paso 4 del Flujo básico Modificar datos de objeto
<b>Pos-condiciones</b>		
1.	En el sistema la información del proceso está actualizada con las modificaciones	
<b>Validaciones</b>		
1.		
<b>Conceptos</b>		No aplica
		No aplica
<b>Relaciones</b>	<b>Requisitos Incluidos</b>	Ninguno
	<b>Requisitos Excluidos</b>	Ninguno
<b>Requisitos especiales</b>	No aplica	
<b>Asuntos pendientes</b>	No aplica	

### Prototipo elemental de interfaz gráfica de usuario

#### Formatos de Entrada

No aplica

#### Formatos de Salida

No aplica

### Requisito 8: Ejecutar proceso

#### Descripción textual del requisito

<b>Precondiciones</b>	El proceso que se desea ejecutar debe estar registrado y público para su utilización por el rol correspondiente al usuario que está solicitando su ejecución.	
<b>Flujo de eventos</b>		
<b>Flujo básico Ejecutar proceso</b>		
<b>No.</b>	<b>Actor</b>	<b>Sistema</b>
1.	Solicita que se ejecute un proceso de los que están publicados en el sistema	
2.		Carga la información correspondiente al proceso seleccionado e identifica la primera acción de su secuencia.
3.		Ejecuta las operaciones correspondientes al tipo

		de acción activa solicitando al usuario las entradas necesarias.
4.	Responde a las solicitudes de las entradas	
5.		Recibe las entradas las procesa de acuerdo a la acción activa, almacena los resultados de la acción y pasa a la siguiente. Si no quedan más acciones a ejecutar culmina el proceso sino continúa en el paso 3 del Flujo básico Ejecutar proceso.
<b>Pos-condiciones</b>		
1.	En el sistema la información de todos los objetos involucrados están actualizadas con las modificaciones	
<b>Validaciones</b>		
1.		
<b>Conceptos</b>		No aplica
		No aplica
<b>Relaciones</b>	<b>Requisitos Incluidos</b>	Ninguno
	<b>Requisitos Excluidos</b>	Ninguno
<b>Requisitos especiales</b>	No aplica	
<b>Asuntos pendientes</b>	No aplica	

**Prototipo elemental de interfaz gráfica de usuario**

**Formatos de Entrada**

No aplica

**Formatos de Salida**

No aplica

## Anexo II. Tablas de descripción de las tablas del modelo de datos.

Anexo II .Tabla 1.

<b>Nombre: Object</b>		
<b>Descripción:</b> Almacena la información y estructura de configuración referente a los objetos (procesos, relaciones o entidades) que empleara el sistema para los procesos de aduanas.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id(PK)	Serial	Id autoincrementado para identificar y diferenciar cada objeto en cuestión.
type	Varchar	Tipo de objeto
description	Varchar	Breve descripción del objeto
name	Varchar	Nombre del objeto

Anexo II .Tabla 2.

<b>Nombre: Relations</b>		
<b>Descripción:</b> Almacena la información y estructura de configuración referente a las relaciones de los objetos que empleará el sistema para los procesos de aduanas.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id_object_parent(PK, FK)	Serial	Id autoincrementado para identificar el objeto padre de la relación
id_object_child (PK, FK)	Serial	Id autoincrementado para identificar el objeto hijo de la relación
relation_type	Varchar	Identificar el tipo de relación entre los objetos

Anexo II .Tabla 3.

<b>Nombre: DD</b>		
<b>Descripción:</b> Almacena la información y estructura de configuración referente a los documentos digitales que empleará el sistema para los procesos de aduanas.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id_obj_dd (PK,FK)	Serial	Id autoincrementado para identificar y diferenciar cada documento digital en cuestión.

Anexo II .Tabla 4.

<b>Nombre: Entity</b>		
<b>Descripción:</b> Almacena la información y estructura de configuración referente a las entidades que empleará el sistema para los procesos de aduanas.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id_obj_entity(PK, FK)	Serial	Id autoincrementado para identificar y diferenciar cada entidad en cuestión. Es llave foránea de Object.
table_name	Varchar	Nombre de la tabla

Anexo II .Tabla 5.

<b>Nombre: Attributes</b>		
<b>Descripción:</b> Almacena la información y estructura de configuración referente a los atributos de los objetos (procesos, relaciones o entidades) que empleara el sistema para los procesos de aduanas.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
Id (PK)	Serial	Id autoincrementado para identificar y diferenciar cada atributo en cuestión.
name	Varchar	Nombre del atributo
description	Varchar	Breve descripción del atributo
data_type	Varchar	Almacena el tipo de dato que es el atributo
is_null	Boolean	Almacena si el atributo es nulo o no.
size	Integer	Define el tamaño del atributo

Anexo II .Tabla 6.

<b>Nombre: Process</b>		
<b>Descripción:</b> Almacena la información y estructura de configuración referente a los procesos que empleará el sistema para la aduana.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id_obj_process (PK, FK)	Serial	Un id autoincrementado para identificar y diferenciar el proceso en cuestión
sequence	Varchar	Indica la secuencia de acciones que presenta el proceso en cuestión
public	Boolean	Almacena si el proceso es público o no

Anexo II .Tabla 7.

<b>Nombre: Actions</b>		
<b>Descripción:</b> Almacena la información y estructura de configuración referente a las acciones empleará el sistema para los procesos de aduanas.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id(PK)	Serial	Id autoincrementado para identificar y diferenciar cada acción en cuestión.
input	Varchar	Punto de partida de la acción
output	Varchar	Respuesta/Salida de la acción
type	Varchar	Tipo de acción
id_obj_process (PK, FK)	Serial	Id autoincrementado para identificar y diferenciar el proceso en cuestión

Anexo II .Tabla 8.

<b>Nombre: Action_validate</b>		
<b>Descripción:</b> Almacena la información y estructura de configuración referente a las acciones de validación que empleará el sistema para los procesos de aduana.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id (PK, FK)	Serial	Id autoincrementado para identificar y diferenciar cada acción en cuestión
alternative(FK)	Integer	Acción alternativa en la secuencia de acciones para el caso en que la validación de como resultado False
operator	Varchar	Operador para validar la acción
value	Varchar	El valor contra el que se valida la acción

Anexo II .Tabla 9.

<b>Nombre: Role</b>		
<b>Descripción:</b> Almacena la información y estructura de configuración referente a los roles que empleara el sistema para los procesos de aduanas.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id(PK)	Serial	Un id autoincrementado para identificar y diferenciar cada rol
name	Varchar	Nombre del rol
description	Varchar	Breve descripción del rol
code	Varchar	Un código que identificará el tipo de rol

Anexo II .Tabla 10.

<b>Nombre: Rel_Process_Role</b>		
<b>Descripción:</b> Almacena la información y estructura de configuración referente a la relación de los roles con los procesos que empleará el sistema para la aduana.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id_obj_process (PK, FK)	Serial	Un id autoincrementado para identificar y diferenciar cada proceso en cuestión
id_role(PK, FK)	Serial	Un id autoincrementado para identificar y diferenciar cada rol
r	Boolean	Almacenara si él tiene permisos de lectura
w	Boolean	Almacenara si él tiene permisos de escritura

### Anexo III. Tablas de diseños casos de pruebas de caja negra.

Tabla Descripción de las secciones y escenarios del caso de prueba.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad
SC 1: Adicionar Objeto	EC 1.1: Salvar Objeto.	Valida que los datos introducidos son correctos y que los campos obligatorios no están vacíos.
	EC 1.2: Insertar datos incorrectos y/o campos vacíos.	Muestra un mensaje indicando que existen campos vacíos y/o con error.
SC 2: Consultar Objeto	EC 2.1: Consultar Objeto.	Muestra los objetos creados en la aplicación
SC 3: Modificar Objeto	EC 3.1: Modificar Objeto	Valida que los datos introducidos son correctos y que los campos obligatorios no están vacíos.
	EC 3.2: Insertar datos incorrectos y/o campos vacíos.	Muestra un mensaje indicando que existen campos vacíos y/o con error.
SC 4: Eliminar Objeto	EC 4.1: Eliminar Objeto	Confirmación de la eliminación del objeto
	EC 4.2: Eliminar Múltiples Objetos	Confirmación de la eliminación de múltiples objetos

Tabla Descripción de las variables

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Tipo	Lista desplegable	No	Se listan los tipos de objetos de negocio.
2	Nombre	Campo de texto	No	Cualquier cadena de caracteres.
3	Descripción	Campo de texto	No	Cualquier cadena de caracteres.
4	Action	Lista desplegable	Si	Se listan las posibles acciones a ejecutar sobre los objetos de negocio.

Tabla Matriz de datos: Caso de prueba del requisito Gestionar Objeto.

Escenario	Tipo	Nombre	Descripción	Action	Respuesta Esperada	Resultado de la Prueba	Flujo Central
<b>EC1.1</b>	<i>Entidad</i> (V)	Agente (V)	Un agente de despacho mercancía (V)	N/A	Muestra un mensaje indicando que el objeto ha sido guardada satisfactoriamente	Satisfactorio	<p>1-El usuario selecciona la opción adicionar objeto.</p> <p>2-El sistema muestra los campos para llenar con los datos del objeto.</p> <p>3-El usuario selecciona el tipo de objeto, llena los demás campos correspondientes.</p> <p>4-El usuario selecciona la opción salvar.</p> <p>5-El sistema valida que los datos introducidos son correctos y que los campos obligatorios no están vacíos.</p> <p>6-El sistema muestra un mensaje notificando que el objeto</p>

							fue guardado satisfactoriam ente terminando así el caso de uso.
<b>EC1.2</b>	<i>Null</i>	Null	Null	N/A	Muestra un mensaje indicando que el objeto tiene campos nulos.	Satisfactorio	<p>1-El usuario selecciona la opción adicionar objeto.</p> <p>2-El sistema muestra los campos para llenar con los datos del objeto.</p> <p>3-El usuario no selecciona el tipo de objeto o no llena los demás campos correspondientes.</p> <p>4-El usuario selecciona la opción salvar.</p> <p>5-El sistema valida que los datos introducidos son correctos y que los campos obligatorios no están vacíos.</p> <p>6-El sistema muestra un</p>

							mensaje notificando que posee campos nulos, dando la opción de llenarlos nuevamente terminando así el caso de uso.
<b>EC2.1</b>	N/A	N/A	N/A	N/A	Muestra los objetos existentes en la aplicación.	Satisfactorio	1-El usuario selecciona mostrar los objetos. 2-El sistema muestra los objetos existentes en la aplicación.
<b>EC3.1</b>	Proceso (V)	DM (V)	Proceso de la aduna para el despacho de mercancías (V)	N/A	Muestra un mensaje indicando que el objeto ha sido guardada satisfactoriamente	Satisfactorio	1-El sistema muestra los objetos que posee el sistema. 2-El usuario selecciona el objeto a modificar. 3-El sistema muestra los campos para modificar los datos del objeto. 4-El usuario selecciona el nuevo tipo de objeto y modifica los

							<p>demás campos correspondientes.</p> <p>5-El usuario selecciona la opción salvar.</p> <p>6-El sistema valida que los datos introducidos son correctos y que los campos obligatorios no están vacíos.</p> <p>7-El sistema muestra un mensaje notificando que el objeto fue guardado satisfactoriamente terminando así el caso de uso.</p>
<b>EC3.2</b>	<i>Null</i>	Null	Null	N/A	Muestra un mensaje indicando que el objeto tiene campos nulos.	Satisfactorio	<p>1-El sistema muestra los objetos que posee el sistema.</p> <p>2-El usuario selecciona el objeto a modificar.</p> <p>3-El sistema muestra los campos para modificar los</p>

							<p>datos del objeto.</p> <p>4-El usuario modifica los campos correspondientes y los deja nulos.</p> <p>5-El usuario selecciona la opción salvar.</p> <p>6-El sistema valida que los datos introducidos son correctos y que los campos obligatorios no están vacíos.</p> <p>7-El sistema muestra un mensaje notificando que el objeto fue guardado satisfactoriam ente terminando así el caso de uso.</p>
<b>EC4.1</b>	N/A	N/A	N/A	N/A	Muestra un mensaje de confirmación de eliminación y luego un mensaje de eliminación satisfactoria.	Satisfactoria	<p>1-El usuario selecciona el objeto a eliminar.</p> <p>2-El sistema muestra los campos de los datos del objeto.</p> <p>3-El usuario</p>

							<p>selecciona la opción eliminar.</p> <p>4-El sistema muestra un mensaje de confirmación con una lista de elementos relacionados que se eliminarán también en caso de existir.</p> <p>5-El sistema muestra un mensaje notificando que el objeto fue eliminado satisfactoriam ente terminando así el caso de uso.</p>
<b>EC4.2</b>	N/A	N/A	N/A	Eliminar Objetos Seleccionados (V)	Muestra un mensaje de confirmación de eliminación y luego un mensaje de eliminación satisfactoria.	Satisfactoria	<p>1-El usuario selecciona los objetos a eliminar.</p> <p>2-El usuario selecciona la opción Eliminar Objetos Seleccionados.</p> <p>3-El sistema muestra un mensaje de confirmación con una lista</p>

							<p>de elementos relacionados que se eliminarán también en caso de existir.</p> <p>4-El sistema muestra un mensaje notificando que el objeto fue eliminado satisfactoriam ente terminando así el caso de uso.</p>
--	--	--	--	--	--	--	--

## Anexo IV. Avales.

### Aval I

A quien pueda interesar:

Por este medio certifico que la solución obtenida como resultado del trabajo de diploma con el título: "Diseño e implementación de un núcleo de gestión de operaciones aduaneras" cubre las siguientes necesidades identificadas durante el tiempo de trabajo y la experiencia acumulada en el desarrollo de software para dicha entidad:

- Permite minimizar el impacto de los cambios que se realizan a los procesos de gestión de la AGR dándole la posibilidad a los usuarios finales de reconfigurar sus procesos y las entidades con las que estos interactúan.
- Permite definir detalladamente la estructura de las entidades definidas para el negocio aduanero de manera tal que se puede adicionar o eliminar atributos de las mismas según el criterio de los usuarios finales.
- Permite la conexión a las funcionalidades desde diferente entornos mientras esté bajo protocolo TCP/IP y realizar intercambios de información.
- Su flexibilidad permitirá a los especialistas funcionales de la AGR definir el funcionamiento de sus procesos libremente y que estos puedan ser explotados desde cualquier lugar del país.

Esta solución tiene un grupo de oportunidades de mejora que han sido debidamente identificadas y se resumen a continuación:

- No cuenta con un módulo para la descripción de documentos digitales, sin lo cual se deja de cubrir un requerimiento básico que tienen todas las soluciones de aduana existentes en el mundo.
- Es recomendable la inclusión de un módulo para definición de Interfaces de Usuario que facilite a los usuarios diseñar las pantallas que deben usar los clientes para suministrar los datos al núcleo.
- Para garantizar una total validación del funcionamiento del núcleo se recomienda desarrollar un cliente que se conecte desde una estación remota y realice intercambio de información en varios procesos que estén definidos.

Atentamente  
Ing. Alain Eduardo Rodríguez Arias  
Jefe de Departamento de Soluciones para Aduana, CEIGE

## Aval II

La Habana, 21 de Junio de 2011  
"Año 53 de la Revolución"

A quien pueda interesar:

Por este medio certifico que en el Departamento de Soluciones para Aduana el diagrama que hemos llamado de Interacción y Actividades se ha establecido como artefacto para representar la interacción entre clases en un escenario.

Las razones que provocaron la adopción de dicho artefacto radican en la necesidad que tenemos de entregar lo más detallado posible las tareas a los implementadores para lograr que nuestros diseñadores de aplicaciones que son nuestro recurso más escaso tengan un cierto nivel de desahogo para poder enfrentar varios diseños de soluciones al mismo tiempo.

Hasta el momento el uso de dicho diagrama nos ha resultado efectivo aunque no se cuenta con estadísticas de cómo se ha incrementado la productividad de nuestro equipo.

  
Atentamente  
Ing. Alain Eduardo Rodríguez Arias  
Jefe de Departamento de Soluciones para Aduana, CEIGE

## GLOSARIO

**Arancel:** Impuesto que paga el importador al introducir productos en un mercado.

**ANSI ASC X.12:**(American National Standards Institute Accredited Standards Committee X12) es la designación oficial para el grupo encargado del desarrollo y mantenimiento de los estándares de Intercambio Electrónico de Datos fundado en 1979, el cual es un comité acreditado de la ANSI.

**BPMN:** (Business Process Modeling Notation). Es una notación estándar para el modelado de procesos de negocio.

**CASE:** (Computer Aided Software Engineering) Ingeniería de Software Asistida por Computadora.

**CRUD:** (Create, Read, Update and Delete). Operaciones (creación, consulta, actualización y eliminación de entidades del negocio) sin las cuales un software no se considera completo, especialmente si está relacionado a una base de datos, dado que estas operaciones son las encargadas de la manipulación de cada entidad del negocio por parte de la aplicación.

**Daemon:** En Linux lo que sería en Windows un servicios como proceso, un proceso que se ejecuta en el fondo y brinda servicios.

**Declaración de Aduana:** Formulario impreso, debidamente autorizado por el importador/exportador, que tiene como objetivos principales: permitir la liquidación y el cobro de los derechos, impuestos u otros gravámenes que se deba pagar por las mercancías y; fiscalizar la importación/exportación de mercancías sujetas a limitaciones y/o protecciones (flora/fauna) y/o prohibiciones y/o exenciones (derechos).

**Declaración de Mercancías:** Es aquella declaración que se realiza del modo prescrito por la aduana, por la cual las personas interesadas indican qué régimen aduanero pretenden aplicar a las mercancías y suministran los detalles informativos que la Aduana requiere para la aplicación del régimen elegido.

**Depósito Aduanero:** Es el almacenamiento de mercancías que se lleva a cabo en los lugares designados al efecto, bajo potestad del Servicio de Aduanas. Allí se depositan mercancías extranjeras, ya sea con el fin de simple depósito o con el objeto que sean sometidas a una transformación.

**Documento de Embarque:** Son aquellos documentos comerciales que representan formalmente la existencia y el dominio de la mercadería a que se refiere y que son el objeto de la transacción respectiva.

**DOM:** Modelo en Objetos para la representación de Documentos. Es un modelo computacional a través de la cual los programas y scripts pueden acceder y modificar dinámicamente el contenido, estructura y estilo de los documentos HTML y XML.

**EDI:** Intercambio Electrónico de Datos.

**EDIFACT:** Es uno de los formatos usado como estándar internacional para Intercambio Electrónico de Datos.

**EFT:** Transferencia electrónica de fondos.

**Framework:** Conjunto de clases o estructuras que implementan los componentes de una aplicación genérica, así como también componentes concretos que cumplen a cabalidad tareas concreta.

**GoF:** (Gang of Four) Grupo de los Cuatro, se refiere al grupo de los autores de un conjuntos de patrones de diseño.

**Gravámenes:** Son los derechos aduaneros y cualquier otro recargo de efectos equivalentes, que inciden sobre las importaciones. No están comprendidos en este concepto, las tasas y recargos análogos cuando corresponden al costo aproximado de los servicios prestados.

**HTML:** (Hyper Text Markup Language) Lenguaje de hipertexto de marcado para la programación web.

**ISO:** Organización internacional de Estándares para certificar la calidad de un software y procesos de producción.

**I/O:** (Input/Output) término empleado en la informática para expresar Entrada/Salida.

**JDBC:** (Java Database Connectivity) es una API que permite la ejecución de operaciones sobre bases de datos desde el lenguaje de programación Java, independientemente del sistema operativo donde se ejecute o de la base de datos a la cual se accede, utilizando el dialecto SQL del modelo de base de datos que se utilice.

**MTI:** (Medios de transporte Internacional) Nave, aeronave, vagón ferroviario, camión, contenedor o cualquiera otro vehículo utilizado para el transporte de mercancías y personas.

**ODBC:** (Open Database Connectivity) es un estándar de acceso a bases de datos desarrollado por SQL Access Group en 1992, el objetivo de ODBC es hacer posible el acceder a cualquier dato desde cualquier aplicación, sin importar qué sistema de gestión de bases de datos (DBMS) almacene los datos.

**OLTP:** (On Line Transaction Processing) es un tipo de sistemas que facilitan y administran aplicaciones transaccionales, usualmente para entrada de datos y recuperación y procesamiento de transacciones

**OMA:** Organización Mundial de Aduanas.

**OMC:** Organización Mundial de Comercio.

**ONU:** Organización de Naciones Unidas.

**PC:** (Personal Computer) ordenador o computadora personal.

**PDA:** (Personal Digital Assistant), es un dispositivo móvil que funciona como administrador de información personal.

**PDF:** (Portable Document Format) es un estándar abierto para el intercambio de documentos. Este formato de ficheros fue creado por Adobe Systems en 1993 y es usado para representar documentos de una forma independiente de la aplicación, hardware y sistema operativo.

**PHP:** (PHP Hypertext Preprocessor). lenguaje script de programación web para propósitos generales. Fue creado originalmente por Rasmus Lerdorf en 1995.

**PKI:** Infraestructura de llave pública. Infraestructura empleada para la Seguridad Informática.

**Régimen Aduanero:** Dícese del tratamiento aplicable a las mercaderías sometidas al control de la aduana, de acuerdo con las leyes y reglamentos aduaneros, según la naturaleza y objetivos de la operación.

**SADEM:** Sistema Automatizado de Despacho Mercantil.

**SGDB:** (Sistema Gestor de Base de Datos). Es el software que permite la utilización y/o la actualización de los datos almacenados en una (o varias) base(s) de datos por uno o varios usuarios desde diferentes puntos de vista y a la vez.

**SSL:** (Secure Sockets Layer) protocolo de encriptación para conexiones seguras por la red o internet

**SWING GUI:** Es una biblioteca gráfica para Java. Incluye componentes para interfaz gráfica de usuario tales como cajas de texto, botones, despleables y tablas.

**UK HM Revenue & Custom CHIEF System:** Sistema de Manejo del Transporte de Mercancías para la Importación y Exportación de las Aduanas pertenecientes al departamento de Impuestos y Aduanas de Su Majestad del Reino Unido.

**Unicode:** Formato de codificación de caracteres.

**WYSIWYG:** (What You See Is What You Get), es un término empleado en la informática para describe un sistema cuyo contenido se muestra durante la edición muy similar a la salida final, la cual puede ser un documento impreso, un página web o una presentación.

**XML:** (Extensible Markup Language), es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C).