

# **UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS**

**Facultad #3**

**Diseño e implementación del componente Resoluciones del Ministerio de Finanzas y Precios.**

**Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas.**

**Autor:**

José Luis Sablón Mora

**Tutores:**

Ing. Raymond Weeden Gamboa

Ing. Lianet Pineda de la Nuez

**Habana, Junio de 2013.**

**“Año 55 de la Revolución”.**

**Declaración de Autoría**

Declaro ser autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

\_\_\_\_\_

José Luis Sablón Mora

\_\_\_\_\_

Ing. Raymond Weeden Gamboa

\_\_\_\_\_

Ing. Lianet Pineda de la Nuez

### *Agradecimientos*

*A mi familia por darme su apoyo y confiar en mí en cada momento de la carrera. A Arianna Galán Quiala por la invaluable ayuda que me ha brindado durante estos cinco años. A Lilian López por haber realizado aportes significativos al desarrollo de esta investigación. A mis amigos y amigas que durante todos estos años han compartido junto a mí. A Leodanis Wilber Polanco y Abel Irsula Tumbarell por haberme acompañado en los momentos buenos y malos de la carrera. A todas las personas que contribuyeron en mi formación, especialmente a mis tutores y profesores por el trabajo que han realizado.*

## *Dedicatoria*

*A mis padres por confiar en mí y brindarme su apoyo incondicionalmente durante estos 5 años de carrera, por darme ánimos aún cuando están lejos de mí físicamente, pero espiritualmente siempre los llevo conmigo, por sacrificarse por mí, les agradezco todo lo que soy hasta este momento y lo que puedo llegar a ser. A mi mamá por brindarme siempre su amor y cariño y estar ahí siempre que la necesite, por darme la fuerza de seguir adelante. A mi papá, por ser un ejemplo en mi carrera, espero que estés orgulloso de mí. A mi hermana por ser tan especial para mí y por estar a mi lado en cada momento de mi vida. A mis abuelos por el amor y el cariño que me han entregado. A mis tíos por los consejos y apoyo que me han brindado. Los quiero infinitamente.*

## Resumen

El uso de las Tecnologías de la Información y las Comunicaciones ha ofrecido a las empresas y organizaciones una oportunidad única para el desarrollo de sus procesos, permitiendo aumentar su eficiencia y competitividad. En Cuba ya es una realidad la informatización de sus instituciones, entre ellas se encuentra la Aduana General de la República (AGR), que se ha propuesto la automatización de todos sus procesos con el fin de brindar un servicio de excelencia.

En el momento que se realiza esta investigación la AGR no posee un sistema digital que permita la recepción de las resoluciones ministeriales, por lo que se hace necesario la realización de un componente informático que posibilite ejecutar esta tarea de forma ágil y efectiva, en un primer momento se realizará esta solución para las resoluciones del Ministerio de Finanzas y Precios, pero es necesario tener en cuenta la incorporación de otros ministerios o resoluciones en el futuro.

Como resultado del trabajo realizado y haciendo uso del Modelo de Desarrollo del Departamento de Soluciones para la Aduana, se obtuvo una solución informática que posibilita la recepción de las resoluciones ministeriales, esta solución fue creada con la filosofía de que en el futuro pueda contar con la incorporación de nuevas resoluciones. De esta forma la AGR contará con un sistema capaz de controlar la recepción de las resoluciones, permitiendo que se realice un trabajo cómodo y ágil por parte de los funcionarios de la institución.

---

## Contenido

Resumen.....	V
Introducción.....	1
1. Capítulo 1: Fundamentación Teórica .....	5
1.1. Introducción. ....	5
1.2. Conceptos generales.....	5
1.2.1. Resoluciones Ministeriales .....	5
1.2.2. Resolución de exención y bonificación .....	5
1.2.3. Resolución Cambio de Moneda .....	6
1.2.4. Resolución Condonación de Deuda.....	6
1.3. Sistemas informáticos.....	6
1.3.1. Ámbito internacional .....	6
1.3.2. Ámbito nacional .....	7
1.4. Descripción de la solución.....	7
1.5. Lenguajes y herramientas utilizadas para el desarrollo.....	7
1.5.1. Lenguajes de programación .....	8
1.5.2. Marco de trabajo .....	8
1.5.3. Entornos de Desarrollo Integrado (IDE).....	10
1.5.4. Herramientas de diseño .....	10
1.5.5. Gestor de Base de Datos .....	11
1.6. Modelo de Desarrollo de software.....	12
El desarrollo de la solución propuesta según las características que posee y la flexibilidad que ofrece el modelo de desarrollo, ejecuta durante su ciclo de vida las siguientes disciplinas:.....	12
1.6.1. Estudio preliminar .....	13
1.6.2. Modelado del negocio.....	13
1.6.3. Requisitos .....	13
1.6.4. Análisis y diseño .....	13
1.6.5. Implementación.....	13
1.6.6. Pruebas internas .....	13

---

1.7. Conclusiones parciales.....	14
2. Capítulo 2: Análisis y diseño de la solución	15
2.1. Introducción.....	15
2.2. Técnicas para la captura de requisitos .....	15
2.3. Requisitos funcionales .....	16
2.3.1. Descripción de requisitos .....	17
2.4. Técnicas de validación de requisitos.....	20
2.5. Requisitos no funcionales .....	20
2.5.1. Rendimiento.....	20
2.5.2. Fiabilidad.....	20
2.5.3. Soporte.....	21
2.5.4. Hardware.....	21
2.5.5. Software .....	21
2.5.6. Portabilidad .....	21
2.6. Modelo de diseño.....	21
2.6.1. Patrones Utilizados .....	22
2.6.2. Diagrama de secuencia.....	24
2.6.3. Diagrama de clases .....	25
2.6.4. Diagrama de paquetes .....	26
2.6.5. Modelo de datos .....	27
2.7. Validación del diseño .....	28
2.8. Conclusiones .....	29
3. Capítulo 3: Implementación y Prueba de la Solución .....	31
3.1. Introducción.....	31
3.2. Estándar de codificación.....	31
3.3. Modelo de Implementación .....	33
3.4. Diagrama de Despliegue .....	33
3.5. Diagrama de Componentes.....	35
3.6. Prueba.....	36
3.6.1. Pruebas de Caja Negra.	36

3.7. Conclusiones Parciales .....	38
4. Conclusiones.....	39
5. Recomendaciones .....	40
6. Referencia Bibliográfica.....	41
7. Bibliografía consultada .....	43
8. Glosario .....	45
9. Anexos .....	47
9.1. Anexo 1: Descripción de requisitos .....	47
9.2. Anexo 2: Casos de prueba .....	77

---

## Introducción

La Universidad de las Ciencias Informáticas (UCI), fue creada en el año 2002, como una pieza fundamental dentro del arduo proceso de informatización de la sociedad cubana(1), siendo parte además del comienzo de una nueva era de producción de software para el mercado nacional e internacional(2).

Bajo el compromiso de brindar un servicio eficaz y seguro para el desarrollo de productos informáticos cubanos, varias empresas y entidades confiaron su informatización a la UCI, formando parte de este grupo la Aduana General de la República de Cuba (AGR).

La aduana internacionalmente es reconocida como la oficina recaudadora fiscal, establecida por el gobierno nacional en los puertos marítimos, fluviales, fronterizos y aeropuertos del país, para aplicar y hacer cumplir la ley de aduanas y sus concordantes, recauda los derechos que fija el arancel y los demás que se hallen a su cargo; corre con las operaciones de entrada y despacho, tránsito y trasbordo, depósito y entrega de mercancías de importación y exportación, reprime el contrabando y el fraude a la renta nacional de aduanas y controla el comercio marítimo internacional y de cabotaje en cumplimiento de las leyes referidas, así como la entrada y salida de viajeros internacionales, y hace efectivas las prohibiciones de orden sanitario que las mismas leyes establecen.(4)

En Cuba la AGR es un órgano que permite el control en frontera, además regula y controla el tráfico del comercio exterior, la misma garantiza la seguridad y protección de la sociedad socialista y de la economía nacional, así como la recaudación fiscal y las estadísticas del comercio exterior, a través del cumplimiento de las políticas estatales de competencia aduanera para el tráfico internacional de viajeros, mercancías, medios de transporte, bultos postales. (5)

En búsqueda de un servicio con mayor calidad y un sistema de trabajo más cómodo la AGR decidió renovarse con las tecnologías emergentes en el mundo, por este motivo se integran el Departamento de Soluciones para la Aduana del Centro de Informatización y la Gestión de Entidades (CEIGE) y el Centro de Automatización y Dirección de la Información (CADI) para el desarrollo del Sistema de Gestión Integral de Aduana (GINA); que tiene como principal objetivo la informatización de la mayoría de los procesos que se realizan en la Aduana General de la República.

En la aduana, se hace necesaria la interacción directa con ministerios y entidades dentro del país. Uno de los ministerios más sobresaliente con los que la AGR interactúa es el Ministerio de Finanzas y Precios (MFP), que propone e implementa las políticas en materia fiscal, de precios, contabilidad, patrimonio, seguros del país y la administración de las finanzas públicas, a fin de garantizar el control y uso eficaz de los recursos financieros destinados a los programas de desarrollo económico y social de la Revolución(6), proceso que se realiza mediante la publicación de resoluciones ministeriales. Por la importancia de esta relación y la necesidad de que el proceso se realice con rapidez, se decide informatizar dichas operaciones.

En la actualidad las resoluciones emitidas por el MFP que son de interés para la AGR son recepcionadas de forma manual o mediante el uso de software de terceros, vías que atentan contra el correcto funcionamiento de la institución. Es importante destacar que durante la ejecución de este proceso se ve comprometida la seguridad e integridad de la información, además se pone en riesgo la pérdida de tiempo valioso para la entidad.

Las deficiencias antes descritas no van a la par del propósito que posee la AGR de agilizar sus procesos, por lo que se hace necesario crear dentro del sistema GINA, un componente que permita llevar a cabo de forma ágil y segura la recepción de las resoluciones ministeriales, además de que sea capaz de integrarse al resto de los módulos con los que cuenta el sistema.

Luego del análisis realizado de los argumentos que se exponen anteriormente se plantea el siguiente **problema a resolver** ¿Cómo agilizar el procesamiento de las resoluciones que emite el Ministerio de Finanzas y Precios referentes a los trámites comerciales en la Aduana General de la República?

El **objeto de estudio** se centra en el procesamiento electrónico de resoluciones ministeriales y se especifica como **campo de acción** el procesamiento electrónico de resoluciones del Ministerio de Finanzas y Precios en la Aduana General de la República.

Con el fin de guiar la presente investigación se define como **objetivo general** desarrollar un componente que agilice el procesamiento de las resoluciones que emite el Ministerio de Finanzas y Precios referentes a los trámites comerciales en la Aduana General de la República.

Para darle solución al objetivo general, el mismo es desglosado en los siguientes **objetivos específicos**:

1. Elaborar el marco teórico relativo a la investigación.
2. Modelar el negocio asociado al procesamiento de resoluciones que emite el MFP referentes a los trámites comerciales en la AGR.

3. Especificar los requisitos funcionales del componente de gestión de las resoluciones del MFP.
4. Modelar el diseño del componente que garantice el procesamiento de las resoluciones del MFP.
5. Implementar el componente que permita la gestión de las resoluciones del MFP en la AGR.
6. Validar la solución propuesta mediante la aplicación de pruebas.

Con el fin de darle cumplimiento a los objetivos del presente trabajo se concibieron las siguientes **tareas de investigación**:

1. Análisis del estado del arte de las soluciones de software que procesen resoluciones ministeriales.
2. Estudio del modelo de desarrollo de software, lenguajes, herramientas de modelado y tecnologías utilizadas en el proyecto.
3. Modelado y descripción de los procesos de negocio.
4. Validación del negocio.
5. Especificación de los requisitos funcionales del componente.
6. Validación de los requisitos funcionales identificados.
7. Modelado de los artefactos del diseño: diagramas de clases, diagramas de secuencia, diagrama de despliegue y modelo de datos.
8. Implementación del código fuente del componente.
9. Aplicación de pruebas al componente.

De esta forma queda planteada la siguiente **Idea a defender**

Si se desarrolla el componente Ministerio de Finanzas y Precios se agilizará el procesamiento de resoluciones emitidas por este ministerio referente a los trámites comerciales de la Aduana General de la República.

Para guiar la presente investigación se utilizan los siguientes **métodos científicos**

#### **Métodos de investigación**

1. **Método de análisis histórico – lógico:** Se utilizará para el estudio del estado del arte de los sistemas informáticos aduaneros del mundo, las técnicas y métodos utilizados para el manejo de resoluciones ministeriales.
2. **Método de modelación:** Se utilizará para realizar los modelos y diagramas referentes a la lógica del sistema en su fase de análisis y diseño.
3. **La entrevista:** Se utilizará para definir los requisitos funcionales y lograr una mayor comprensión del sistema.

4. **Implementación:** Se utilizará con el objetivo fundamental de desarrollar el sistema modelado, implementando las clases que fueron identificadas durante la etapa de diseño.

## 1. Capítulo 1: Fundamentación Teórica

### 1.1. Introducción.

Durante el desarrollo de este capítulo se abordan detalles sobre los principales elementos que conforman la fundamentación teórica de este trabajo. Se realiza un análisis de los principales sistemas que existen en Cuba y el mundo que se utilizan para la gestión de resoluciones ministeriales, especificando en cada caso sus potencialidades y debilidades. En este capítulo se exponen los conceptos de los principales elementos que fundamentan el contenido de la investigación y se describen las tecnologías y herramientas que serán utilizadas durante el proceso de desarrollo del sistema.

### 1.2. Conceptos generales

Durante el desarrollo de este epígrafe se darán a conocer los principales conceptos que debemos tener en cuenta para lograr una mejor comprensión de la investigación que se realiza.

#### 1.2.1. Resoluciones Ministeriales

Las resoluciones ministeriales se definen como una norma cuya finalidad es ejecutar y supervisar la política general del Estado dentro del ámbito de un ministerio, resolviendo casos concretos y particulares, las mismas son aprobadas por el ministro del sector, quien es el encargado de firmarlas y velar por su ejecución.<sup>(7)</sup>

#### 1.2.2. Resolución de exención y bonificación

Las exenciones pueden ser de dos tipos:

- Total: se refiere a una exención del pago para todos los códigos del arancel.
- Parcial: se refiere a una exención del pago para una lista de mercancías.

Las resoluciones de bonificaciones se crean para dejar de pagar un determinado por ciento de los derechos de aduanas, por la importación de mercancías. Las exenciones o bonificaciones pueden ser otorgadas hasta que se cumpla un determinado valor. Estos valores se pueden referir a los siguientes datos: valor factura<sup>1</sup>, derechos de aduana<sup>2</sup> y valor en aduana<sup>3</sup>. Para el caso del valor factura, en aduana y

---

<sup>1</sup> **Valor Factura:** Se anotará en moneda nacional el precio realmente pagado o por pagar de las mercancías.

<sup>2</sup> **Derechos De Aduana:** Derechos establecidos en el Arancel Aduanero y/o en la legislación nacional, a los cuales están sujetas las mercancías que entran al territorio nacional o que salen de él.

<sup>3</sup> **Valor De Aduana:** Se entiende como valor en aduana de mercancías importadas su valor de transacción. De igual modo, el valor de transacción es el precio efectivamente pagado o por pagar por las mercancías cuando éstas se vendan para su exportación al territorio aduanero de la Comunidad.

los derechos de aduana se pueden otorgar para una resolución y este valor se comparte para todos los anexos de la resolución. También se puede poner el valor tope específico para cada empresa mientras que la cantidad de artículos se controlará por subpartidas.

### **1.2.3. Resolución Cambio de Moneda**

Este tipo de resolución consiste fundamentalmente en autorizar el pago de los derechos de aduanas en CUP por la importación de mercancías que tengan dispuesto la protección en CUC. También se puede emitir este tipo de beneficio para autorizar el pago de los aranceles en CUP a empresas que tengan la obligación de realizar el mismo en CUC, sin que medie ningún código protegido. Este tipo de resolución no hace referencia a ninguna otra resolución.

Este cambio de moneda se puede otorgar por cantidad, valor en aduana, valor factura o derecho de aduana, cumpliendo los mismos requisitos de que la cantidad se controla a nivel de mercancía y el valor factura, en aduana y el derecho de aduana a nivel de empresa.

### **1.2.4. Resolución Condonación de Deuda**

Esta resolución consiste en liberar de la obligación de pago de deuda por concepto de aranceles a una entidad determinada. Este beneficio tendrá un monto tope a condonar, el cual estará relacionado con una o varias Declaraciones de Mercancías. El monto a condonar estará expresado en derechos de aduanas y se debe considerar como un sacrificio fiscal.

## **1.3. Sistemas informáticos**

En el escenario nacional e internacional se han desarrollado soluciones informáticas con el fin de optimizar los procesos que realizan las aduanas, ejemplo de ellos se encuentran el Sistema Informático Aduanero Automatizado (SIDUNEA) y Sistema Único de Aduana (SUA) por citar algunos ejemplos.

### **1.3.1. Ámbito internacional**

Existe una tendencia generalizada en estos sistemas hacia el perfeccionamiento de la gestión de los procesos, prueba vigente de ellos es SIDUNEA, instalado en más de 80 países del mundo, que se encuentra en desarrollo de nuevos módulos para evitar que los usuarios requieran de herramientas externas para realizar las operaciones de subida de resoluciones ministeriales y archivos en formato de lenguaje de marcas extensible (XML) y desde su propio sistema puedan enviar los datos referidos a sus transacciones.(8)

Este software en particular, aunque es uno de los más famosos a nivel mundial para la gestión de procesos aduaneros del mundo y en algún momento fue utilizado para la gestión de los procesos en la AGR, posee ciertas deficiencias a la hora de integrarlo al sistema aduanero de Cuba, dado que, en la AGR se llevan a cabo procesos propios que impiden la fácil adaptación de cualquier software que se desee instaurar. Otra de las limitaciones que posee este software es su carácter privativo, con licencias altamente costosas.

### **1.3.2. Ámbito nacional**

En Cuba fue desarrollado el sistema informático para la gestión de los procesos aduaneros (SUA), que posee funcionalidades que permiten realizar un trabajo cómodo y eficaz, ya que informatiza la mayoría de los procesos aduaneros, pero posee la debilidad de que no cuenta con un módulo capaz de realizar la inserción de archivos o resoluciones en formato XML dentro de su bases de datos, obligando a que necesariamente tenga que realizarse este proceso de forma manual, insertando los datos directamente en la base de datos, presentando dificultades como; posibles errores en la entrada de datos; excesiva pérdida de tiempo por parte del equipo de trabajo; necesidad de personal del equipo de desarrollo para ejecutar esta tarea; además de presentar riesgos que impliquen la disponibilidad de la información afectando directamente a la economía del país.

### **1.4. Descripción de la solución**

El procesamiento de resoluciones provenientes del MFP en la AGR constituye un factor fundamental para el funcionamiento interno de la institución, ya que la aduana aunque es la encargada de controlar, verificar y asegurar el tráfico de personas, bultos y mercancías, así como el pago de aranceles y recaudación fiscal, no es la que establece tarifas o bonificaciones para estas operaciones, es por eso que se encuentra intrínsecamente relacionada con el MFP.

Este componente tendrá la tarea fundamental de administrar el flujo de las resoluciones ministeriales emitidas por el MFP, permitiendo que el proceso de envío y recepción de estas sea rápido y seguro, además que exista una forma organizada y ágil de consultar las resoluciones, facilitando de esta manera el trabajo en la AGR.

### **1.5. Lenguajes y herramientas utilizadas para el desarrollo**

Durante el transcurso de los años se ha evidenciado un creciente interés por el perfeccionamiento del trabajo en cada uno de los proyectos que realiza el centro CEIGE, brindando de esta manera un

incremento de experiencia entre sus trabajadores, lo que apunta directamente a una mejora en la calidad y eficiencia. Este proceso de crecimiento y madurez ha dirigido a que se definan normas y estándares para cada uno de sus proyectos, quedando registradas en el Documento Línea base de la Arquitectura, las cuales a continuación se describen.

### 1.5.1. Lenguajes de programación

#### 1.5.1.1. PHP

Durante el desarrollo del sistema GINA se ha definido como lenguaje del lado del servidor<sup>4</sup> PHP (acrónimo de “PHP: Hypertext Preprocessor”) en su versión 5.3. Este lenguaje creado a partir del análisis de las principales características de C, Java y Perl, tiene como objetivo permitir el desarrollo de páginas web de forma rápida y sencilla.(9)

Entre las principales características que posee este lenguaje están:

- Puede interactuar con muchos motores de bases de datos tales como MySQL, MS SQL, Oracle, Informix, PostgreSQL, y otros muchos.(10)
- Posee una alta velocidad de ejecución, la cual es muy importante, además no crear cargas en el sistema, lo que indica que no requiere demasiados recursos para su funcionamiento.(10)
- Posee una gran estabilidad, PHP goza de la ayuda de una gran comunidad de desarrollo, permitiendo que los fallos de funcionamiento se encuentren y se reparen rápidamente. Utiliza su propio sistema de administración de recursos y dispone de un sofisticado método de manejo de variables, conformando un sistema robusto y estable.(10)
- Provee diferentes niveles de seguridad, estos pueden ser configurados de archivos de configuración que se encuentran en el servidor.(10)

### 1.5.2. Marco de trabajo

Marco de trabajo es un esquema (un esqueleto, un patrón) para el desarrollo y/o la implementación de una aplicación. Un marco de trabajo simplifica el desarrollo de una aplicación mediante la automatización de algunos de los patrones utilizados para resolver las tareas comunes. Además, un marco de trabajo proporciona estructura al código fuente, forzando al desarrollador a crear código más legible y más fácil de mantener. Por último, un marco de trabajo facilita la programación de aplicaciones, ya que encapsula operaciones complejas en instrucciones sencillas.(11)

---

<sup>4</sup> **Lenguajes en el lado del servidor:** son los lenguajes que se procesan en un servidor remoto y que generan la página web antes de enviarla al cliente.

### 1.5.2.1. Symfony

Symfony es un marco de trabajo diseñado para optimizar, gracias a sus características, el desarrollo de las aplicaciones web. Separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja.

Las clases de la capa del modelo se generan automáticamente, en función de la estructura de datos de la aplicación. La librería Propel<sup>5</sup> se encarga de esta generación automática, pues crea el esqueleto o estructura básica de las clases y genera automáticamente el código necesario.

La abstracción de la base de datos es completamente transparente para el programador, pues se realiza de forma nativa mediante PDO (PHP Data Objects). Así, si se cambia el sistema gestor de bases de datos en cualquier momento, no se debe reescribir el código, ya que tan solo es necesario modificar un parámetro en un archivo de configuración.

Symfony está desarrollado completamente con PHP 5. Ha sido probado en numerosos proyectos reales y se utiliza en sitios web de comercio electrónico de primer nivel. Su carácter libre y multiplataforma lo hacen uno de los marco de trabajo de PHP más usados en el mundo.(12)

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

- Funciona con todas las bases de datos comunes (MySQL, PostgreSQL, SQLite, Oracle, MS SQL Server)
- Sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos.
- Basado en la premisa de "convenir en vez de configurar", en la que el desarrollador sólo debe configurar aquello que no es convencional.
- Sigue la mayoría de mejores prácticas y patrones de diseño para la web.
- Preparado para aplicaciones empresariales y adaptables a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo.
- Código fácil de leer que incluye comentarios de phpDocumentor y que permite un mantenimiento muy sencillo.

---

<sup>5</sup>**Propel:** es un proyecto de software libre, es una de las mejores capas de abstracción de objetos/relacional disponibles en PHP 5.

- Fácil de extender, lo que permite su integración con librerías desarrolladas por terceros.(12)

Este marco de trabajo de desarrollo, en su versión 1.4, es la herramienta base utilizada para llevar a cabo la implementación del sistema GINA, proporcionando robustez y seguridad al futuro producto de software.

### **1.5.3. Entornos de Desarrollo Integrado (IDE)**

Los Entornos de Desarrollo Integrados (en inglés, Integrated Development Environment o IDE) son programas compuestos por una serie de herramientas que utilizan los programadores para desarrollar código. Estas herramientas pueden estar pensadas para su utilización con un único lenguaje de programación o bien puede dar cabida a varios de estos.(13)

#### **1.5.3.1. NetBeans**

NetBeans IDE es una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito en Java, pero puede servir para cualquier otro lenguaje de programación. Existe además un número importante de módulos para extender el NetBeans IDE, además es un producto libre y gratuito sin restricciones de uso. Posee un amplio soporte para el lenguaje PHP así como para el marco de trabajo de trabajo Symfony.(14)

Para el desarrollo de la solución se definió NetBeans IDE, en su versión 6.9.

### **1.5.4. Herramientas de diseño**

Las herramientas de diseño se han convertido en todo una moda entre las empresas productoras de software a nivel internacional, en la actualidad gran cantidad de estas han adquirido herramientas CASE<sup>6</sup>, con el propósito de automatizar los aspectos clave durante el proceso de desarrollo de los sistemas informáticos durante todo su ciclo de vida y de esta manera lograr un trabajo más eficiente.

#### **1.5.4.1. Visual Paradigm**

Visual Paradigm ha sido concebida para soportar el ciclo de vida completo del proceso de desarrollo del software a través de la representación de todo tipo de diagramas. Fue diseñado para una amplia gama de usuarios interesados en la construcción de sistemas de software de forma fiable a través de la utilización de un enfoque Orientado a Objetos.(15)

Algunas de sus principales características se relacionan a continuación.

- Disponibilidad en múltiples plataformas (Windows, Linux).

---

<sup>6</sup> **CASE:** siglas en inglés que se utilizan para referirse a Ingeniería de Software Asistida por Computadora

- Diseño centrado en casos de uso y enfocado al negocio que generan un software de mayor calidad.
- Capacidades de ingeniería directa e inversa.
- Soporta aplicaciones Web.
- Ingeniería inversa Java, C++, Esquemas XML, XML, NET exe/dll, CORBA IDL.
- Diagramas de flujo de datos.
- Generación de bases de datos - Transformación de diagramas de Entidad-Relación en tablas de base de datos.

Para el desarrollo de la solución se definió Visual Paradigm para UML 8.0.

### **1.5.5. Gestor de Base de Datos**

Un Sistema Gestor de Bases de Datos (SGBD) o DBMA (DataBase Management System) es una colección de programas cuyo objetivo es servir de interfaz entre la base de datos, el usuario y las aplicaciones. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. Un SGBD permiten definir los datos a distintos niveles de abstracción y manipular dichos datos, garantizando la seguridad e integridad de los mismos. (16)

#### **1.5.5.1. Oracle**

Oracle es un Sistema Gestor de Bases de Datos con características objeto-relacionales, que pertenece al modelo evolutivo de SGBD. Sus características principales son las siguientes: (17)

- Entorno cliente/servidor.
- Gestión de grandes bases de datos.
- Alto rendimiento en transacciones.
- Sistemas de alta disponibilidad.
- Disponibilidad controlada de los datos de las aplicaciones.
- Gestión de la seguridad.
- Autogestión de la integridad de los datos.
- Portabilidad.

➤ Compatibilidad.

## 1.6. Modelo de Desarrollo de software

El modelo de desarrollo de software es utilizado para el desarrollo de cualquier producto informático, el mismo propone desarrollar una serie de tareas entre la idea inicial y final del producto, además establece el orden en que se realizarán las cosas en el proyecto. El modelo provee de requisitos de entrada y salida para cada una de las actividades que se realicen durante el ciclo de vida del proyecto.(18)

En el proyecto GINA se utiliza el Modelo de Desarrollo de Software definido por CEIGE, para detallar cada una de las actividades de las fases del ciclo de vida de los proyectos (Figura 1) y los artefactos para cada una, independientemente de las herramientas o métodos que se utilicen.



**Figura 1.1:** Ciclo de vida de los proyectos para la Aduana. Tomado de (18).

El desarrollo de la solución propuesta según las características que posee y la flexibilidad que ofrece el modelo de desarrollo, ejecuta durante su ciclo de vida las siguientes disciplinas:

### **1.6.1. Estudio preliminar**

Durante el desarrollo de esta fase se llevan a cabo las actividades relacionadas con la planeación del proyecto a un alto nivel. Se realiza un estudio inicial de la organización cliente que permite obtener información fundamental acerca del alcance del proyecto y realizar estimaciones de tiempo, esfuerzo y costo, y decidir si se ejecuta o no el proyecto.(3)

### **1.6.2. Modelado del negocio**

Esta es la fase destinada a comprender los procesos de negocio de la organización. Se comprende cómo funciona el negocio que se desea automatizar para tener garantías de que el software desarrollado va a cumplir su propósito.(3)

### **1.6.3. Requisitos**

En la fase de requisitos se desarrolla un modelo del sistema que se va a construir. Incluye un conjunto de artefactos que describen todas las interacciones que tendrán los usuarios con el software y que responden a los requisitos funcionales del sistema. Se especifican los requisitos funcionales y no funcionales.(3)

### **1.6.4. Análisis y diseño**

Durante esta fase se realiza el modelado del sistema para que soporte todos los requisitos. Esto contribuye a una arquitectura sólida y estable que se convierte en un plano para la próxima fase. Los artefactos generados en esta etapa son más formales y específicos de una implementación. En caso de llevarse a cabo la reutilización de componentes de software ya desarrollados, durante esta fase se ajusta el modelado existente a los requisitos actuales.(3)

### **1.6.5. Implementación**

A partir de los resultados del análisis y diseño se implementa el sistema en términos de componentes, es decir, ficheros de código fuente, scripts, ejecutables y similares. Al reutilizar componentes software ya implementados se lleva a cabo el desarrollo necesario para ajustar a los requisitos actuales y posteriormente realizar la integración de los componentes.(3)

### **1.6.6. Pruebas internas**

Durante esta fase se desarrollan las pruebas del grupo de calidad del centro verificando el resultado de la implementación. Permite identificar posibles errores en la documentación y el software, es decir requisitos que el producto debería cumplir y que aún no los cumple.(3)

## 1.7. Conclusiones parciales

Durante el desarrollo de este capítulo se realizó un análisis de los sistemas informáticos existentes tanto a nivel nacional como internacional, que comparten características similares con la solución informática que se propone. De estos sistemas fueron estudiadas sus ventajas y debilidades, análisis que permitió definir la necesidad de desarrollar una solución informática para la recepción de resoluciones ministeriales en la AGR.

Las tecnologías seleccionadas para el desarrollo de la solución están acordes a las políticas del centro. Fue definido como lenguaje de modelado UML, el cual ayuda a una rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Como herramienta CASE fue determinado el Visual Paradigm que se utilizará para el apoyo del modelado del negocio, así como para la generación de artefactos en las etapas de definición de requisitos y de especificación de componentes. La utilización del marco de trabajo Symfony para el desarrollo de la aplicación ofrece disímiles ventajas para la implementación y el posterior mantenimiento del sistema. Fue definido el sistema gestor de base de datos Oracle que nos brinda inmensas opciones tanto en flexibilidad, rapidez, eficiencia y seguridad. En conclusión podemos determinar que el uso de herramientas y tecnologías en su mayoría libres y multiplataforma nos brinda una reducción considerable en el costo del producto.

## 2. Capítulo 2: Análisis y diseño de la solución

### 2.1. Introducción

En el presente capítulo se realizará el análisis y diseño de la solución propuesta, en el mismo quedará plasmado una descripción de los procesos que se automatizarán, así como los diagramas, patrones de diseño y arquitectura que serán utilizados en la solución, además se utilizarán métricas para ayudar a la detección de errores en fases tempranas del desarrollo de la aplicación.

### 2.2. Técnicas para la captura de requisitos

Como técnicas para la captura de requisitos fueron utilizadas:

**La entrevista:** esta técnica permite al analista adquirir conocimiento del problema y comprender los objetivos de la solución buscada. A través de esta técnica el equipo de trabajo se acerca al problema de una forma natural. Su estructura abarca cuatro pasos:

- Identificación de los entrevistados.
- Preparación de la entrevista.
- Realización de la entrevista.
- Documentación de los resultados.

Esta técnica requiere que el entrevistador sea experimentado y tenga capacidad para elegir bien a los entrevistados y obtener de ellos toda la información posible en un período de tiempo siempre limitado.(19)

Como entrevistados se identificaron a algunos trabajadores del CADI, que poseen una amplia experiencia en el trabajo con resoluciones ministeriales, además de un conocimiento avanzado en el diseño de software y gran dominio del negocio de la aplicación. Como segundo paso se les informó a los trabajadores antes mencionados la necesidad de su participación en la realización de esta técnica, luego se realizó la entrevista que en una primera instancia fue colectiva para obtener una idea general del trabajo y luego de forma individual para especificar en cada uno de los aspectos de interés. Para cada una de estas entrevistas se realizó un conjunto de preguntas previamente elaboradas, las cuales se oscilaron entre preguntas abiertas y cerradas con el único fin de lograr que las entrevistas fueran lo más efectiva posible. Por último fueron documentados los resultados obtenidos, quedando así los requisitos funcionales del sistema definidos.

**Tormenta de ideas:** es una técnica de reuniones en grupo cuyo objetivo es que los participantes muestren sus ideas de forma libre. Consiste en la mera acumulación de ideas y/o información sin evaluar

las mismas. El grupo de personas que participa en estas reuniones no debe ser muy numeroso (máximo 10 personas), una de ellas debe asumir el rol de moderador de la sesión, pero sin carácter de controlador. Como técnica de captura de requisitos es sencilla de usar y de aplicar, puesto que no requiere tanto trabajo en grupo.(21)

Esta técnica fue utilizada durante las reuniones del equipo de desarrollo con un grupo de trabajadores del CADI, el principal objetivo del uso de esta técnica fue para que los participantes mostraran sus ideas sobre cómo se debía realizar la aplicación.

A raíz de la acumulación de toda esta información se logró enriquecer la forma en que se describieron los requisitos posteriormente.

### **2.3. Requisitos funcionales**

A continuación se relacionan los requisitos funcionales con los que debe cumplir el sistema.

- Registrar documento Resoluciones de Exención y Bonificación: El sistema debe permitir el registro de las resoluciones de tipo exención y bonificación.
- Registrar documento Resoluciones de Cambio de Moneda: El sistema debe permitir el registro de las resoluciones de tipo cambio de moneda.
- Registrar documento Resoluciones de Ampliación de Exención y Bonificación: El sistema debe permitir el registro de las resoluciones de tipo ampliación de exención y bonificación.
- Registrar documento Resoluciones de Ampliación de Exención y Bonificación debido a cambio de nomenclatura: El sistema debe permitir el registro de las resoluciones de tipo exención y bonificación debido a cambio de nomenclatura.
- Registrar documento Resoluciones de Prorroga de Exención y Bonificación: El sistema debe permitir el registro de las resoluciones de tipo prórroga de exención y bonificación.
- Registrar documento Resoluciones de Suspensión de Exención y Bonificación: El sistema debe permitir el registro de las resoluciones de tipo suspensión de exención y bonificación.
- Registrar documento Resoluciones de Suspensión de Exención y Bonificación debido a cambio de nomenclatura: El sistema debe permitir el registro de las resoluciones de tipo suspensión de exención y bonificación debido a cambio de nomenclatura.
- Registrar documento Resoluciones de Cancelación de Exención y Bonificación: El sistema debe permitir el registro de las resoluciones de tipo cancelación de exención y bonificación.
- Registrar documento Resoluciones de Condonación de la Deuda: El sistema debe permitir el

registro de las resoluciones de tipo condonación de la deuda.

- Registrar documento Resoluciones de Modificación del Valor Exento: El sistema debe permitir el registro de las resoluciones de tipo modificación de valor exento.

### 2.3.1. Descripción de requisitos

A continuación se muestra la descripción de uno de los requisitos funcionales del sistema (ver, **¡Error! No se encuentra el origen de la referencia.**), para consultar los restantes remitirse al anexo número uno Descripción de requisitos del presente trabajo.

#### Especificación del requisito Registrar documento resoluciones de Exención y Bonificación.

<b>Precondiciones</b>	Recibir el objeto del componente de intercambio de datos con la información de los datos del documento Resoluciones De Exención y Bonificación.
<b>Flujo de eventos</b>	
<b>Flujo básico Registrar documento Resoluciones De Exención y Bonificación</b>	
1.	Ejecuta el Método para el procesamiento de los datos del documento Resoluciones MFP, pasándole el objeto como parámetro.
2.	Revisa que el número de resolución no exista.
3.	Registra el número de la resolución de exención, además de los datos de fechas como son fecha de elaboración.
4.	Revisa si el documento no tiene un tope a nivel de resolución.
5.	Guarda el tope por el cual está hecha la resolución (Factura, Derechos, Valor Aduana).
6.	Se verifican que las empresas estén registradas en el registro central de aduana así como que el código de la empresa beneficiada (código REEUP) exista.
7.	Verifica que la fecha de inicio del beneficio y la fecha de fin del beneficio sea mayor que el día actual.
8.	Registra el listado de empresas válidas al cual va dirigida la resolución de exención. Además de los datos específicos de la empresa (NIT, empresa beneficiada, fecha de inicio del beneficio, fecha de fin del beneficio, la empresa necesita declaración jurada). Para la empresa beneficiada la verifica por su código REEUP

9.	En caso de existir se registran el listado de contratos válidos para esa empresa en dicha resolución.
10.	En caso que venga el Exempto Protección (con valor 0 es para todas las mercancías del arancel, con valor 1 es para todo el arancel excepto las partidas protegidas en dicho anexo). Almacena dicha información.
11.	En caso de que no se haya registrado el tope a nivel de resolución se registra el valor factura, valor de derechos o valor aduana.
12.	Se valida que la mercancía este vigente en el arancel.
13.	Se registra el listado de mercancías válidas para esa empresa en esta resolución. Por cada mercancía se registra Código SACLAP, bonificación, moneda.
14.	Se registra también de las mercancías la cantidad posible a importar según esta resolución del MFP, en caso de que el tope sea por cantidad.
15.	Concluye el requisito.

**Pos-condiciones**

1.	Se guardan los datos de la Resolución de Exención y Bonificación del MFP.
----	---

**Flujos alternativos**

**Flujo alternativo 1.a Ya existe dicho número de exención.**

1	Se almacena en una lista temporal el error para ser enviado posteriormente (Ya existe dicho número de resolución).
2	Continúa en el punto 16 del flujo principal.

**Pos-condiciones**

1	Se almacenan los errores para enviarlos posteriormente.
---	---

**Flujo alternativo 2.a No existe dicha empresa.**

1	Se almacena en una lista temporal el error para ser enviado posteriormente (no existe la empresa XXXXX).
2	Continúa en el punto 6 del flujo principal.

**Pos-condiciones**

1	Se almacenan los errores para enviarlos posteriormente.
---	---

**Flujo alternativo 3.a No existen dicha empresa beneficiada.**

1	Se almacena en una lista temporal el error para ser enviado posteriormente (no existe la empresa beneficiada XXXXX).
2	Continúa en el punto 6 del flujo principal.
<b>Pos-condiciones</b>	
	Se almacenan los errores para enviarlos posteriormente.
<b>Flujo alternativo 4.a Fechas menores a la fecha actual.</b>	
1	Se almacena en una lista temporal el error para ser enviado posteriormente (la fecha de inicio y fin del beneficio para la entidad XXXX deben ser mayores a las del día actual).
2	Continúa en el punto 6 del flujo principal.
<b>Pos-condiciones</b>	
	Se almacenan los errores para enviarlos posteriormente.
<b>Flujo alternativo 5.a Mercancía no existente en el arancel.</b>	
1	Se almacena en una lista temporal el error para ser enviado posteriormente (Mercancía no existente en el arancel).
2	Continúa en el punto 13 del flujo principal.
<b>Pos-condiciones</b>	
	Se almacenan los errores para enviarlos posteriormente.
<b>Flujo alternativo 6.a Enviar errores.</b>	
1	Envía en una lista al componente de intercambio todos los errores detectados.
2	No se registran los datos del documento.
3	Concluye el requisito.
<b>Pos-condiciones</b>	
1	Se envían los errores al componente de intercambio.
<b>Validaciones</b>	
1	Se validan los datos según lo establecido en el Modelo Conceptual
<b>Requisitos especiales</b>	No aplicable
<b>Asuntos</b>	No aplicable

pendientes	
------------	--

### Prototipo elemental de interfaz gráfica de usuario

No aplicable

### Formatos de entrada/salida

#### Entradas

Objeto de PHP con la estructura del documento que se va a procesar en este caso con la estructura del documento Resoluciones de Exención y Bonificación.

#### Salidas

La lista de errores será enviado en el formato JSON que se defina.

## 2.4. Técnicas de validación de requisitos

**Revisiones de requisitos:** los requisitos son analizados sistemáticamente por un equipo de revisores. Una revisión de requisitos es un proceso manual que involucra a personas tanto de la organización del cliente como de la contratista. Las revisiones de requisitos pueden ser informales o formales.(20)

**Auditorías:** la revisión de la documentación con esta técnica consiste en un chequeo de los resultados contra una lista de chequeo predefinida o definida a comienzos del proceso, es decir sólo una muestra es revisada.(21)

## 2.5. Requisitos no funcionales

En este epígrafe se relacionan los requisitos no funcionales con los que debe cumplir la aplicación.

### 2.5.1. Rendimiento

El sistema debe poseer un nivel de eficiencia alto, debido a que está expuesto a un flujo de datos elevado, así como el volumen de la base de datos, por este motivo es necesario que el diseño del sistema y las técnicas de programación utilizadas sean óptimos para el lenguaje de programación definido.

### 2.5.2. Fiabilidad

El sistema debe registrar cualquier fallo en su funcionamiento para garantizar que los administradores cuenten con la información necesaria para corregir cualquier error que se produzca en el proceso de registro del documento.

### 2.5.3. Soporte

La aplicación cliente debe ser capaz de correr sobre cualquier plataforma. Para la parte servidora se recomienda que corra sobre plataforma Linux y que sea programado en PHP 5.3.x y con un gestor de base de datos Oracle 11g o superior.

### 2.5.4. Hardware

- La pc **cliente** debe poseer como requisitos mínimos un procesador Intel P4 o superior, 333 MHz mínimo pero recomendado 1.8 GHz, tarjeta Ethernet 100 Mbit, memoria RAM 512 Mb.
- El **servidor de aplicación** por la cantidad de datos que maneja es necesario que como requisitos mínimos cuente con un servidor Apache, el módulo php habilitado, el módulo PDO\_OCI habilitado, los módulos PDO, OCI8, XSL habilitados, cuente con PHP 5.3.X o superior, un procesador Intel Core 2 Duo 2.2 Gb, memoria RAM 1Gb o superior, 1 Discos SATA de 73 GB en RAID 1.
- En el **servidor de base de datos** es necesario contar con un CPU 2 Dual Core AMD Opteron 64 bits a 2.2 GHz y con una memoria RAM = 2 GB como requisitos mínimos.

### 2.5.5. Software

- En las pc **clientes** debe estar instalado como navegador web el Mozilla Firefox 3.5, el Internet Explorer 6 o superior. Como sistema operativo Linux o Windows NT u otra versión superior,
- El **servidor de aplicación** debe tener instalado PHP 5.3 o superior, Servidor Web Apache 2.2.10 o superior. En la Base de Datos es necesario que esté instalada una versión de Oracle 11g.

### 2.5.6. Portabilidad

- El sistema debe poder ser instalado en cualquier sistema operativo ya sea Windows o cualquier distribución de Linux.

## 2.6. Modelo de diseño

Es el proceso de diseño de la aplicación es el que determina las características principales del sistema final, establece los límites en rendimiento y calidad que la mejor implementación puede alcanzar y determina qué costos se alcanzarán.

El diseño es una actividad que comienza cuando el analista de sistemas ha producido un conjunto de requisitos funcionales lógicos para el mismo y finaliza cuando el diseñador ha especificado los componentes del sistema y las relaciones entre ellos.

Es aquí donde se toman decisiones que afectarán finalmente al éxito de la implementación del programa y con igual importancia la facilidad de mantenimiento que tendrá el software. Estas decisiones se llevan a cabo durante el diseño del software haciendo que sea un paso fundamental de la fase de desarrollo.

La importancia del diseño del software se puede definir con una única palabra: calidad. El diseño es el proceso en el que se asienta la calidad del desarrollo del software. Produce las representaciones del software en las que puede evaluarse su calidad.

El diseño sirve como base para todas las posteriores etapas del desarrollo y de la fase de mantenimiento. Sin diseño se corre el riesgo de construir un sistema inestable, que falle cuando se realicen pequeños cambios, que pueda ser difícil de probar, cuya calidad no pueda ser evaluada hasta más adelante en el proceso de ingeniería de software cuando quede poco tiempo y se haya gastado ya mucho dinero. (22)

### **2.6.1. Patrones Utilizados**

El uso del marco de trabajo Symfony para el desarrollo de la solución informática brinda disímiles ventajas para el equipo de desarrollo, una de ellas es el uso de buenas prácticas durante la implementación del sistema, que se logra mediante el uso de patrones de diseños, los cuales el marco de trabajo implementa por sí solo. A continuación se realiza una explicación de los patrones utilizados.

**Creador:** La clase DocumentoResolucionMFP.php contiene las acciones definidas para el módulo Resoluciones del Ministerio de Finanzas y Precios, en esta se ejecutan las funciones que hacen al sistema funcional. En la misma las acciones se encargan de crear los objetos de las clases que representan las entidades, evidenciando de este modo que la clase DocumentoResolucionMFP.php es el “creador” de las entidades.

**Experto:** Se evidencia este patrón puesto que Propel es la librería externa que utiliza Symfony para realizar su capa de abstracción al modelo de datos, encapsulando toda la lógica de los datos y generando las clases con funcionalidades comunes de las entidades. Por tanto cada clase creada por Propel a partir de una entidad es experta en manejar su información.

**Bajo Acoplamiento:** Plantea que debe haber pocas dependencias entre las clases. Se utiliza para mantener las clases lo menos relacionadas posibles para en caso de que ocurran cambios, estos repercutan lo menos posible en las otras.

**Alta cohesión:** Propone que cada elemento de nuestro diseño debe realizar una labor única dentro del sistema, la implementación de ciertas responsabilidades en las clases Peer, que puedan ser reutilizadas, es un ejemplo claro del uso de este patrón, demás al poseer el sistema herencias poco profundas también da muestra de una alta cohesión.

**Patrón MVC en Symfony:** El patrón Modelo Vista Controlador (MVC) consiste en la separación del código del programa en tres capas, dependiendo de su naturaleza. La lógica relacionada con los datos se incluye en el modelo, el código de la presentación en la vista y la lógica de la aplicación en el controlador.

Symfony toma lo mejor del patrón arquitectura MVC y lo implementa de forma que el desarrollo de aplicaciones sea rápido y sencillo, quedando dividido el contenido de cada capa como se muestra a continuación.(11)

**Contenido de cada capa en Symfony:**

- La capa del Modelo
  - Abstracción de la base de datos
  - Acceso a los datos
- La capa de la Vista
  - Vista
  - Plantilla
- Layout
  - La capa del Controlador
  - Controlador frontal
  - Acción

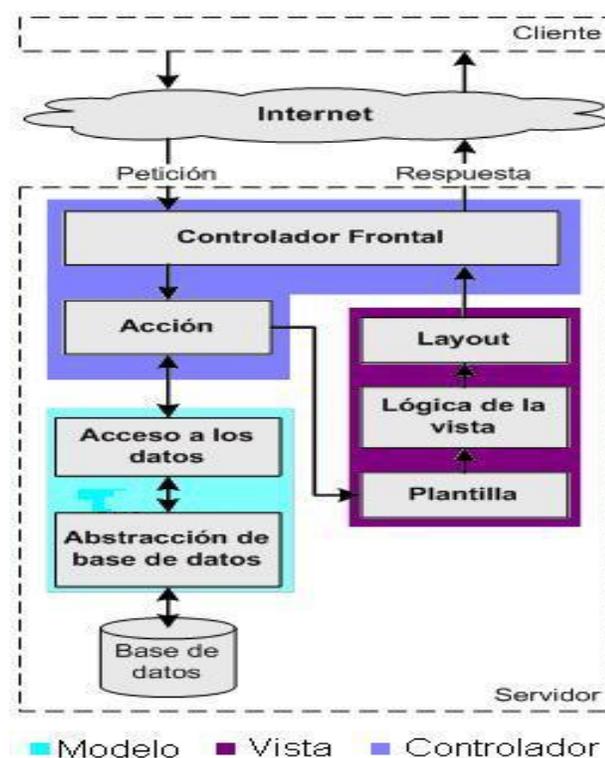


Figura 2.1: Patrón MVC. Tomado de (19).

**La capa del Modelo:** La capa modelo se puede dividir en el acceso a los datos y en la capa de abstracción de la base de datos. De esta forma, las funciones que acceden a los datos no utilizan sentencias ni consultas que dependen de una base de datos, sino que utilizan otras funciones para

realizar las consultas. Así, si se cambia de sistema gestor de bases de datos, solamente es necesario actualizar la capa de abstracción de la base de datos.(23)

**La capa de la Vista:** La capa de la vista también puede aprovechar la separación de código este caso en un layout y en una plantilla. Normalmente, el layout es global en toda la aplicación o al menos en un grupo de páginas. La plantilla sólo se encarga de visualizar las variables definidas en el controlador. (23)

**La capa del Controlador:** Entre las tareas comunes de un controlador se encuentran el manejo de las peticiones del usuario, el manejo de la seguridad, cargar la configuración de la aplicación y otras tareas similares. Por este motivo, el controlador normalmente se divide en un controlador frontal, que es único para cada aplicación, y las acciones, que incluyen el código específico del controlador de cada página. Una de las principales ventajas de utilizar un controlador frontal es que ofrece un punto de entrada único para toda la aplicación. Así, en caso de que sea necesario impedir el acceso a la aplicación, solamente es necesario editar el script correspondiente al controlador frontal. Si la aplicación no dispone de controlador frontal, se debería modificar cada uno de los controladores.(23)

### 2.6.2. Diagrama de secuencia

Los diagramas de secuencia, son un ejemplo de diagramas de iteración que muestra la secuencia de la lógica, el orden en que se suceden los mensajes. Importante, especialmente cuando se trabaja en ambientes altamente compartidos. A continuación se presenta uno de los diagramas de secuencia asociados a la funcionalidad registrar resolución de exención.

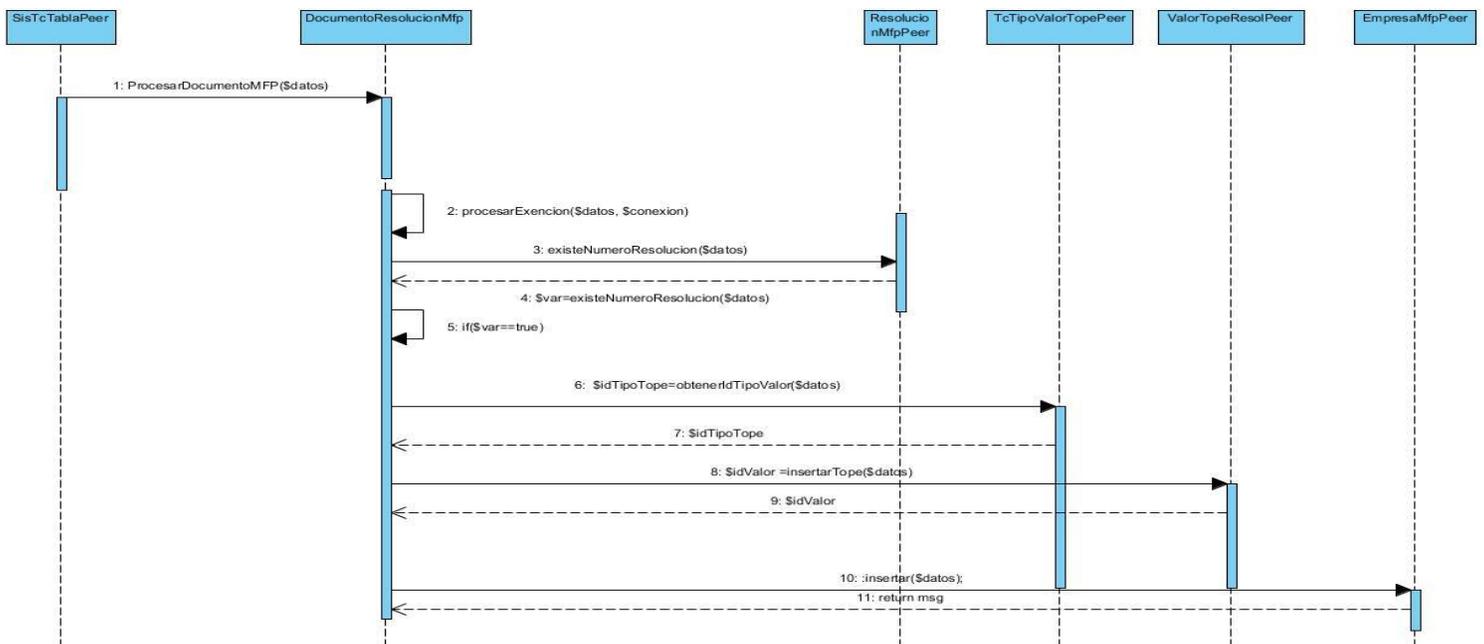


Figura 2.3: Diagrama de secuencia.

### 2.6.3. Diagrama de clases

En este epígrafe se presenta el diseño de las clases persistentes del sistema, sirve como una primera aproximación al diseño definitivo del modelo de datos.

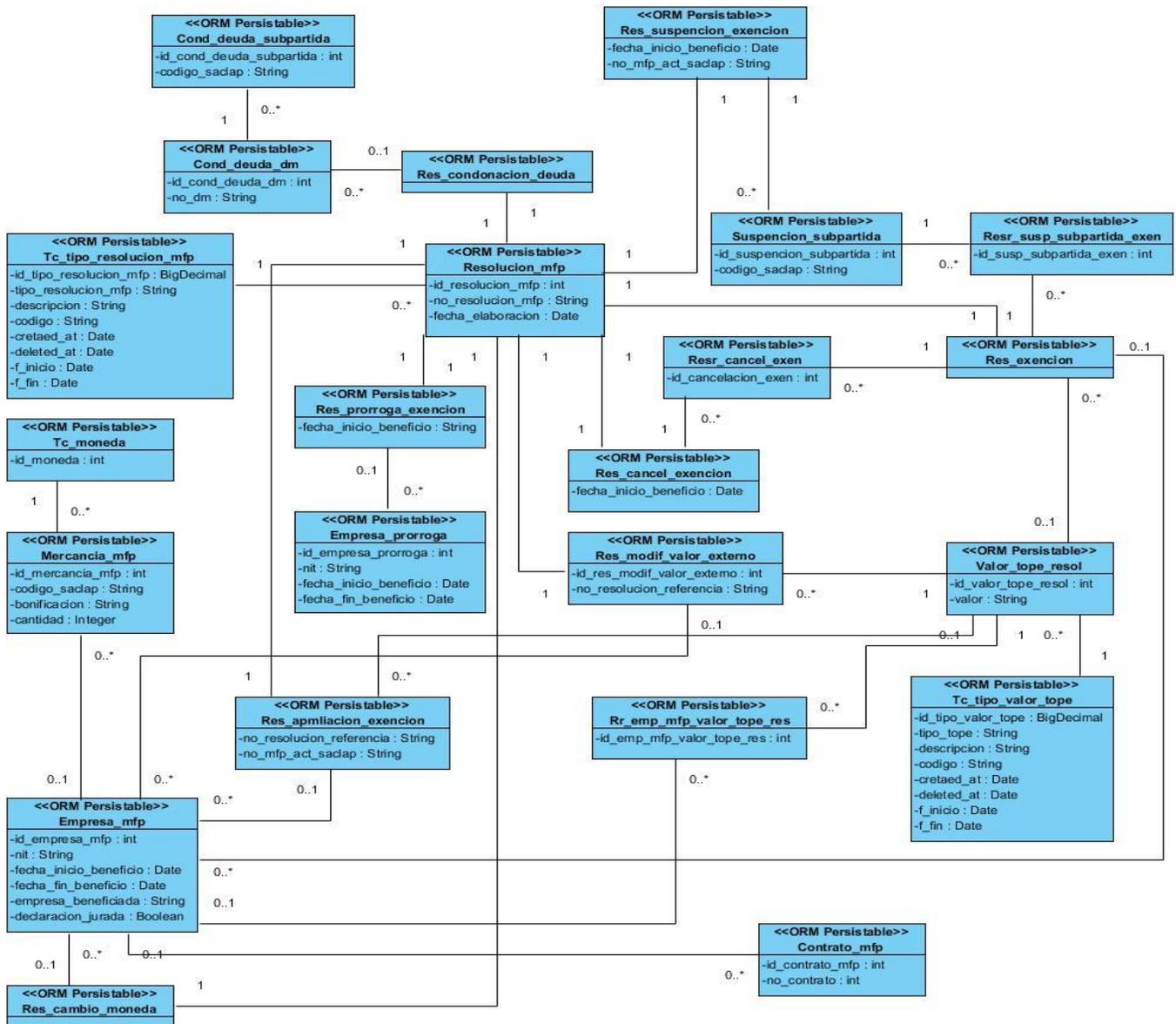


Figura 2.2: Diagrama de clases.

### 2.6.4. Diagrama de paquetes

Un diagrama de paquetes es un conjunto de agrupaciones lógicas de elementos del modelo, mostrando las dependencias entre dichas agrupaciones. Los paquetes están normalmente organizados para maximizar la coherencia interna dentro de cada paquete y minimizar el acoplamiento externo entre los paquetes. Un paquete puede contener paquetes subordinados, así como otros tipos de elementos del modelo. Todo tipo de elementos del modelo UML pueden ser organizados en paquetes.

En la **Figura 2.4** se muestra el diagrama de paquetes correspondiente a la solución que se modela. El mismo está compuesto por el paquete vuBuffer dentro del paquete VU, siendo este el encargado del envío de los datos hacia GINA. También aparece el paquete GINA dentro del cual se encuentra el paquete sfUtilPlugin, que están directamente relacionados con el desarrollo, además se encuentra el paquete mfp que es donde se desarrolla toda la lógica programada concerniente a la solución modelada en la presente investigación.

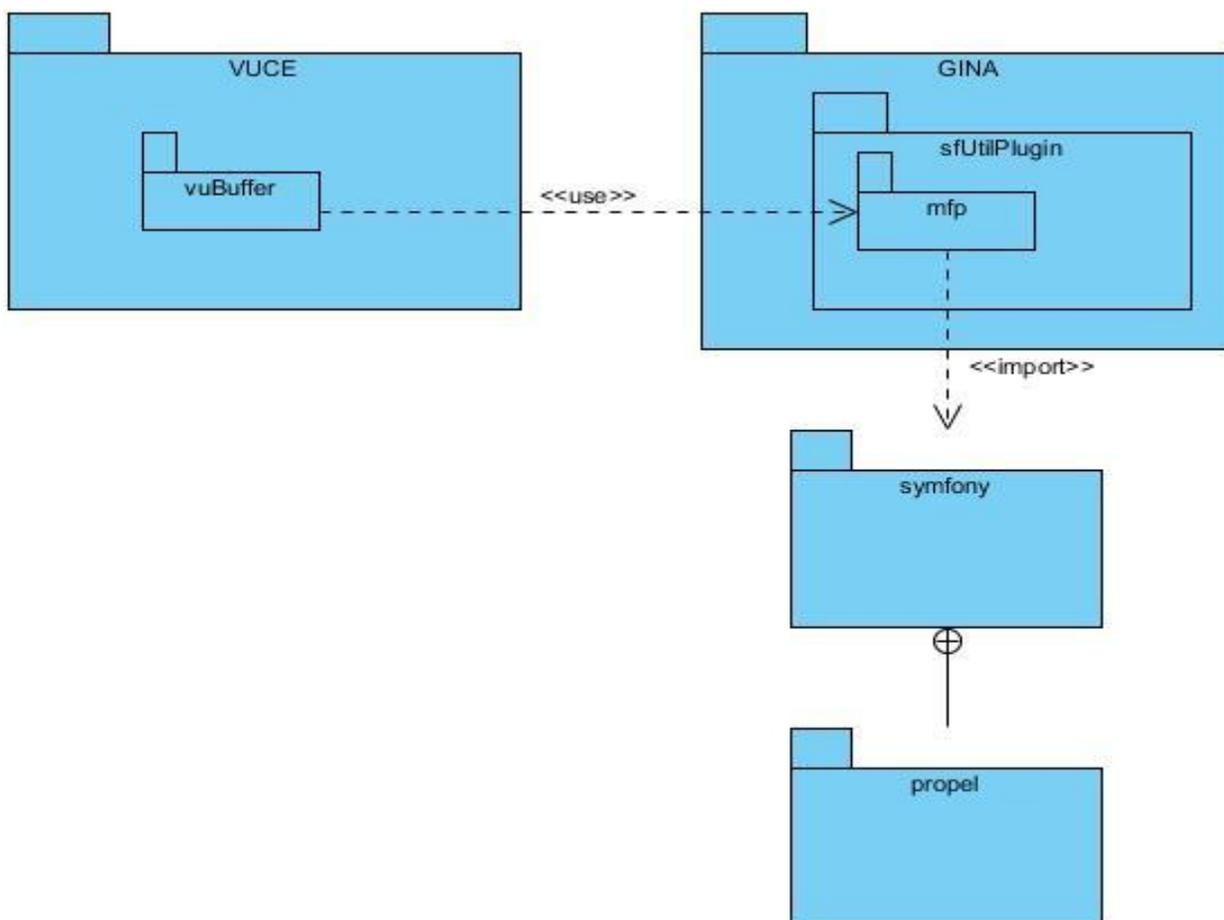


Figura 2.4: Diagrama de Paquetes.

### 2.6.5. Modelo de datos

A continuación se muestra el diagrama Entidad – Relación (ver **Figura 2.5**) perteneciente al sistema. En el mismo se muestran las entidades que forman parte de la base de datos que almacenará toda la información correspondiente a la solución en desarrollo.

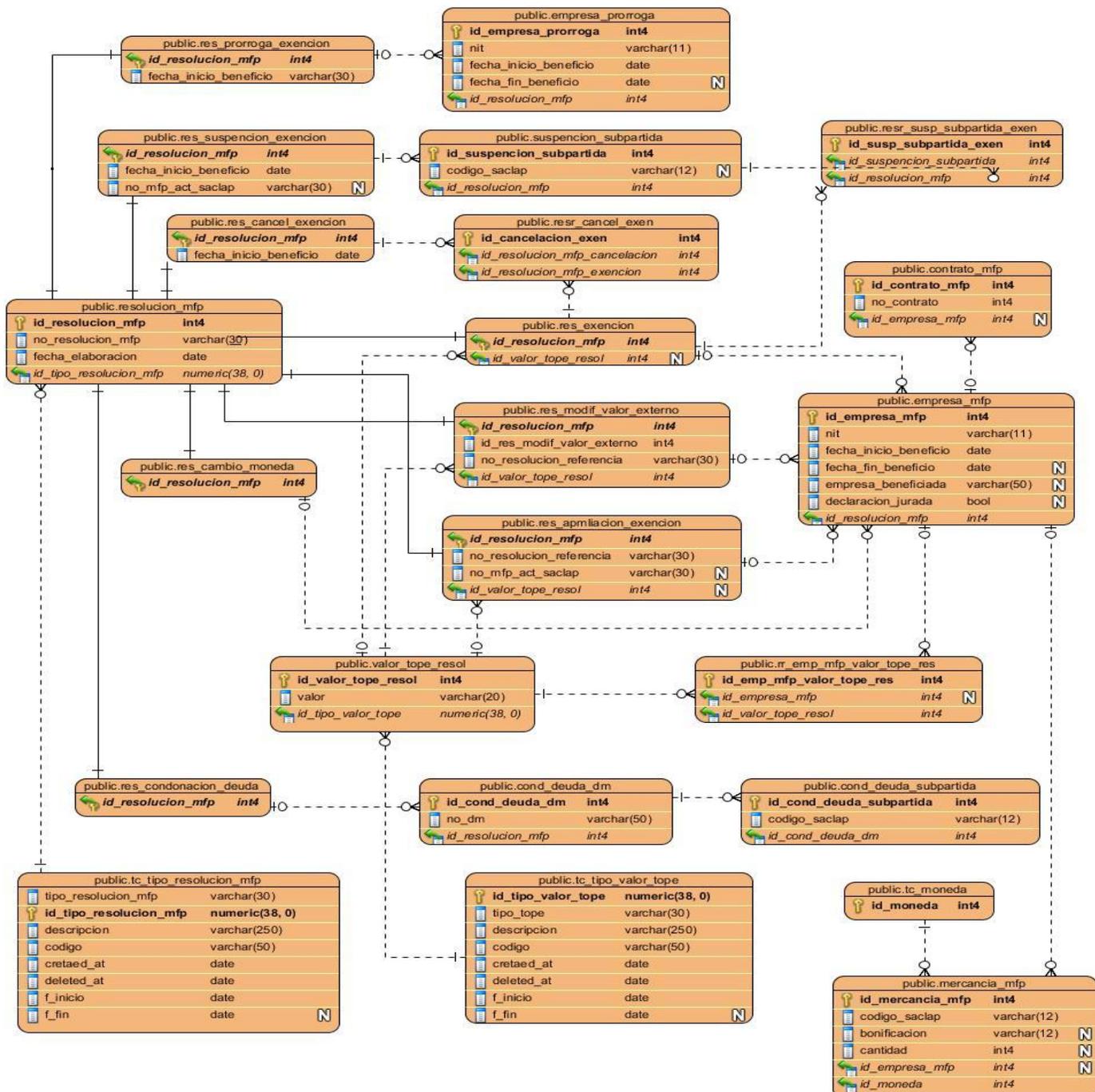


Figura 2.5: Modelo de datos.

## 2.7. Validación del diseño

El diseño está enfocado a convertir los requisitos del cliente en un modelo que al ser implementado, se obtenga el producto deseado. Generalmente constituye el punto de partida para el desarrollo de software una vez especificados los requisitos del sistema. Evidentemente realizar una validación del mismo para verificar su calidad y flexibilidad, garantiza una buena base para la implementación.

Lorenz y Kidd dividen las métricas basadas en clases en cuatro categorías: tamaño, herencia, valores internos y valores externos. Las métricas orientadas a tamaños para una clase se centran en cálculos de atributos y de operaciones para una clase individual, y promedian los valores para el sistema en su totalidad. Las métricas basadas en herencia se centran en la forma en que se reutilizan las operaciones a lo largo y ancho de la jerarquía de clases. Las métricas para valores internos de clase examinan la cohesión y asuntos relacionados con el código, y las métricas orientadas a valores externos examinan el acoplamiento y la reutilización.(24)

La validación del diseño del componente Resoluciones del Ministerio de Finanzas y Precios se realizará utilizando las métricas de Lorenz y Kidd, en específico la métrica Tamaño Operacional de Clase (TOC) por ser la que más se acomoda al diseño de la solución propuesta, esta métrica se determina por el número total de operaciones que están encapsuladas dentro de la clase. Grandes valores de esta medida muestran que la clase puede tener demasiada responsabilidad, lo cual reducirá la reusabilidad de la misma y complicará su implementación. Por otro lado, en cuanto menor sea el valor medio para el tamaño más probable es que las clases tengan menos responsabilidad y complejidad y más nivel de reutilización.

Los valores de umbrales tomados en cuenta para evaluar el diseño propuesto son los siguientes:

➤ **Responsabilidad de las clases.**

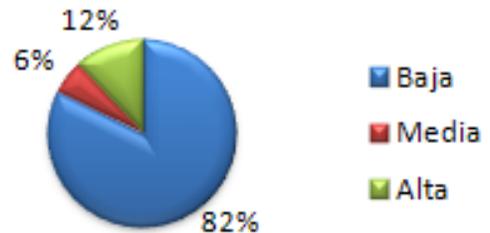
La gráfica demuestra que la responsabilidad de las clases a implementar en su generalidad es baja, aspecto importante que definirá durante la implementación del sistema una buena reutilización del código así como una fácil implementación del mismo.



➤ **Complejidad de implementación de las clases.**

En la gráfica se evidencia que la complejidad de las clases en su mayoría es baja, esto da una muestra de que la implementación del sistema será lo mas sencilla posible, lo que ofrece grandes ventajas a la hora de desarrollar el mismo.

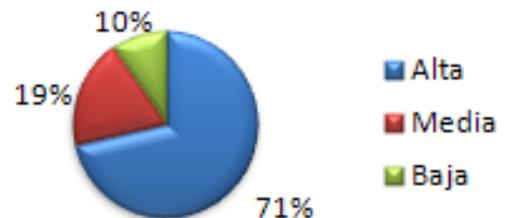
## Complejidad



➤ **Reutilización de las clases.**

La gráfica demuestra que las clases del sistema en su mayoría son reutilizables, lo que ofrecerá grandes ventajas a la hora de implementar la aplicación y evitara la redefinición de código a la hora de desarrollar el mismo.

## Reutilización



La aplicación de esta métrica nos muestra según los resultados obtenidos que los atributos de calidad de las clases se encuentran en un nivel satisfactorio (79.3%); se evidencia que las clases no se encuentran muy sobrecargadas en responsabilidad, su complejidad de implementación es baja y presentan un alto nivel de reutilización, parámetros que dan buena fe de la buena calidad del diseño de la aplicación.

## 2.8. Conclusiones

Durante el desarrollo de este capítulo se obtuvieron los artefactos generados durante la etapa de diseño de la solución. Fueron presentadas las técnicas de captura de requisitos, las que permitieron dejar claramente expuestas las funcionales con las que debe contar el software para que el producto cumpla con las expectativas del cliente. Con la validación de los requisitos fue comprobada la calidad con que se capturaron los mismos, arrojando resultados alentadores, los cuales se evidenciarán en la etapa de implementación. Se definieron los patrones que se utilizarán durante el desarrollo de la solución, exponiendo las ventajas que nos brinda la aplicación de estos a nuestro sistema. Se obtuvieron los

diagramas de clases, secuencia, paquetes y el modelo de datos, que ofrecen una visión más específica de lo que será el software. Para la validación del diseño fue utilizada la métrica Tamaño Operacional de Clase que permite obtener un criterio confiable del correcto diseño del producto.

## 3. Capítulo 3: Implementación y Prueba de la Solución

### 3.1. Introducción

En este capítulo se abordarán los aspectos más importantes referentes a la implementación y validación del sistema, se realizará el diagrama de despliegue y de componentes, además se enunciarán los estándares de codificación que se siguieron para generar el código, también se realizará la validación del sistema a través de los casos de pruebas.

### 3.2. Estándar de codificación

En la actualidad es muy común la utilización de estándares de codificación para la mayoría de los lenguajes de programación existentes. La utilización de los mismos permite una mejor comunicación entre los programadores creando condiciones para la reusabilidad y mantenimiento de los sistemas.

La adopción de un estándar de codificación sólo es viable si se sigue desde el principio hasta el final del proyecto de software. No es práctico adoptar un estándar de codificación una vez iniciado el trabajo. En el caso del Departamento de Soluciones para la Aduana, el estándar de codificación fue definido por la dirección del proyecto en sus inicios y se encuentra registrado en el documento “Propuesta de un estándar de codificación”.

#### Como regla general se tiene:

Cuando se incluyen abreviaturas en mayúsculas no se debe incluir su nombre completo, sino que se utiliza el primer nombre en mayúscula y el resto en minúsculas.

Ejemplo:

GetHtmlStatistic//Correcto

GetHTMLStatistic//Incorrecto

#### Aplicaciones

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

Se debe evitar mientras sea posible la utilización de palabras compuestas o la utilización de varias palabras, en caso de que sea palabras compuestas se utilizará la notación UpperCamelCase<sup>7</sup>.

#### Módulos

---

<sup>7</sup> **UpperCamelCase**: Consiste en escribir frases o palabras compuestas eliminando los espacios intermedios y poniendo en mayúscula la primera letra de cada palabra incluyendo la primera letra de la frase.

Deben referirse a los nombres de tablas en caso de que se trate de un módulo generado por el CRUD. En caso de que sean módulos del negocio de la aplicación debe cumplir con las mismas reglas de codificación de los nombres de las aplicaciones.

### Acciones

Symfony trae su propia nomenclatura para las clases de las acciones y sus funciones, pero no especifica claramente cuál es el nombre que se le debe poner a cada una de las funcionalidades del modelo que serán accedidas por el usuario [dígase acciones].

1. Dentro de las especificaciones del Marco de trabajo está que cada una de estas acciones debe comenzar con la palabra `execute`.
2. Todos los nombres de acciones deben estar en la nomenclatura “CamelCase<sup>8</sup>” comenzando por la palabra `execute`.
3. En caso de ser acciones referentes a un módulo de CRUD de una tabla deben ser nombres específicos como `executeNuevo`, `executeEditar`, etc.
4. Los nombres de las acciones deben especificar con la menor cantidad de palabras cual es el objetivo de la acción, de ser posible estar en infinitivo. Debe especificar bien claro cuál es la acción que se pretende ejecutar con la acción pero sin especificar los parámetros que recibe.

Ejemplo:

Correcto: `usuario/editar`

Incorrecto: `usuario/editar_dado_id`

### Plugins

Symfony especifica que la notación de los plugins siempre debe estar atada al sufijo `Plugin`. El nombre de un plugin debe iniciar con el prefijo `sf`, además debe indicarse, con la menor cantidad de palabras, cual es el objetivo del mismo.

### Nombre de las clases

1. Los nombres de las clases deben estar expresados en notación `UpperCamelCase`.
2. No se deben utilizar guiones bajos en su nombre “\_”.
3. Deben expresar con claridad cuál es el alcance y la responsabilidad de la clase.

---

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

4. Los nombres de las clases no deben estar atados a las clases de las que se deriva, cada clase debe tener un significado por ella misma, no en dependencia de la clase de la que deriva.
5. En los nombres compuestos por más de tres palabras se debe revisar el diseño, no sea que se le estén dando a la clase más responsabilidades de las que realmente tiene.

### **Nombres de los archivos de las clases**

Los nombres de los archivos de las clases deben estar compuestos por el nombre de la clase seguido de un punto y la palabra “class” y la extensión del archivo “.php”.

### **Variables**

1. Los nombres de las variables deben expresar claramente el contenido de la misma.
2. Pueden estar referidas en singular o plural.
3. Se definen al principio de las estructuras donde son utilizadas.
4. En caso de que no se le asigne un valor inicial se deben inicializar con un valor que indique el tipo de dato más general al que debe pertenecer.

Los tipos de datos cadena son definidos con comillas dobles (“”);

Los tipos de datos de caracteres se definen con comillas simples (‘’);

En caso de que se espere almacenar tipos de datos diversos no se inicializa.

### **3.3. Modelo de Implementación**

Durante la etapa de diseño se generan los artefactos que propician el correcto desarrollo de la implementación del sistema, y que posee como principal propósito desarrollar la arquitectura y el sistema como un todo, en esta etapa es donde se implementan las clases definidas durante el diseño.

El modelo de implementación describe cómo los elementos del modelo de diseño se implementan en términos de componentes, como ficheros de código fuente o ejecutables, describe cómo se organizan los componentes de acuerdo con los mecanismos de estructura disponibles en el entorno de implementación y en el lenguaje de programación utilizados, y como dependen los componente uno de los otros, este define una jerarquía de subsistemas de implementación que contiene componentes de interfaces.

### **3.4. Diagrama de Despliegue**

El diagrama de despliegue brinda la posibilidad de visualizar las relaciones físicas que existen entre los componentes de la aplicación. El mismo describe como está distribuido físicamente el sistema, además

muestra la distribución de los componentes de software entre los distintos nodos de cómputo. Este diagrama permite comprender la correspondencia entre las arquitecturas de software y hardware. En el los nodos se utilizan para modelar la topología del hardware sobre el cual se ejecuta el sistema.

Durante el desarrollo de la solución fueron tomadas en cuenta las especificaciones generales del modelo de despliegue utilizado para el Sistema GINA, el cual está implementado siguiendo la arquitectura cliente-servidor.

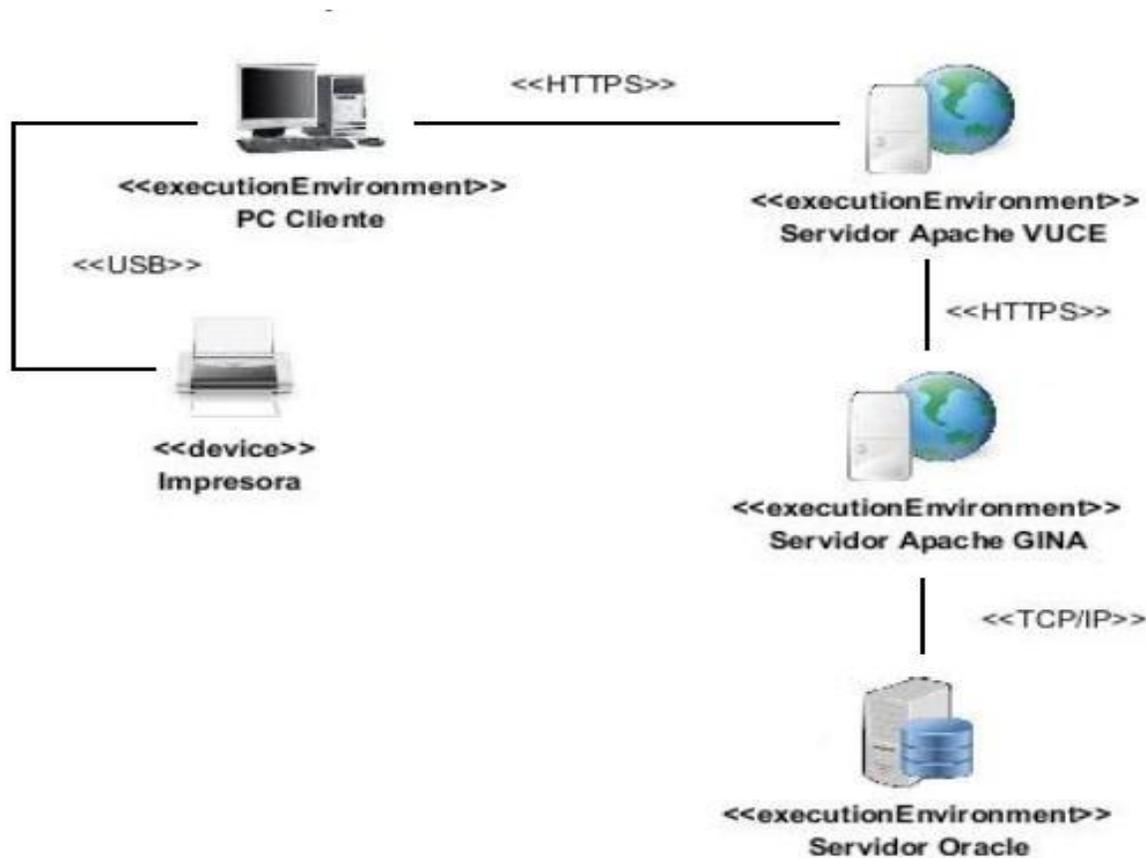


Figura 3.1: Diagrama de despliegue.

### 3.5. Diagrama de Componentes

La relación existente entre los elementos físicos son descritos mediante el diagrama de componentes, el mismo muestra la forma en que se encuentran entrelazados los diferentes componente que son utilizados en el desarrollo de la solución. Un componente se corresponde con una o más clases, interfaces o colaboraciones. A continuación se presenta en la **Figura. 3.2** el diagrama de componentes de la solución, en el mismo se representa los principales componentes por los que está formada la aplicación.

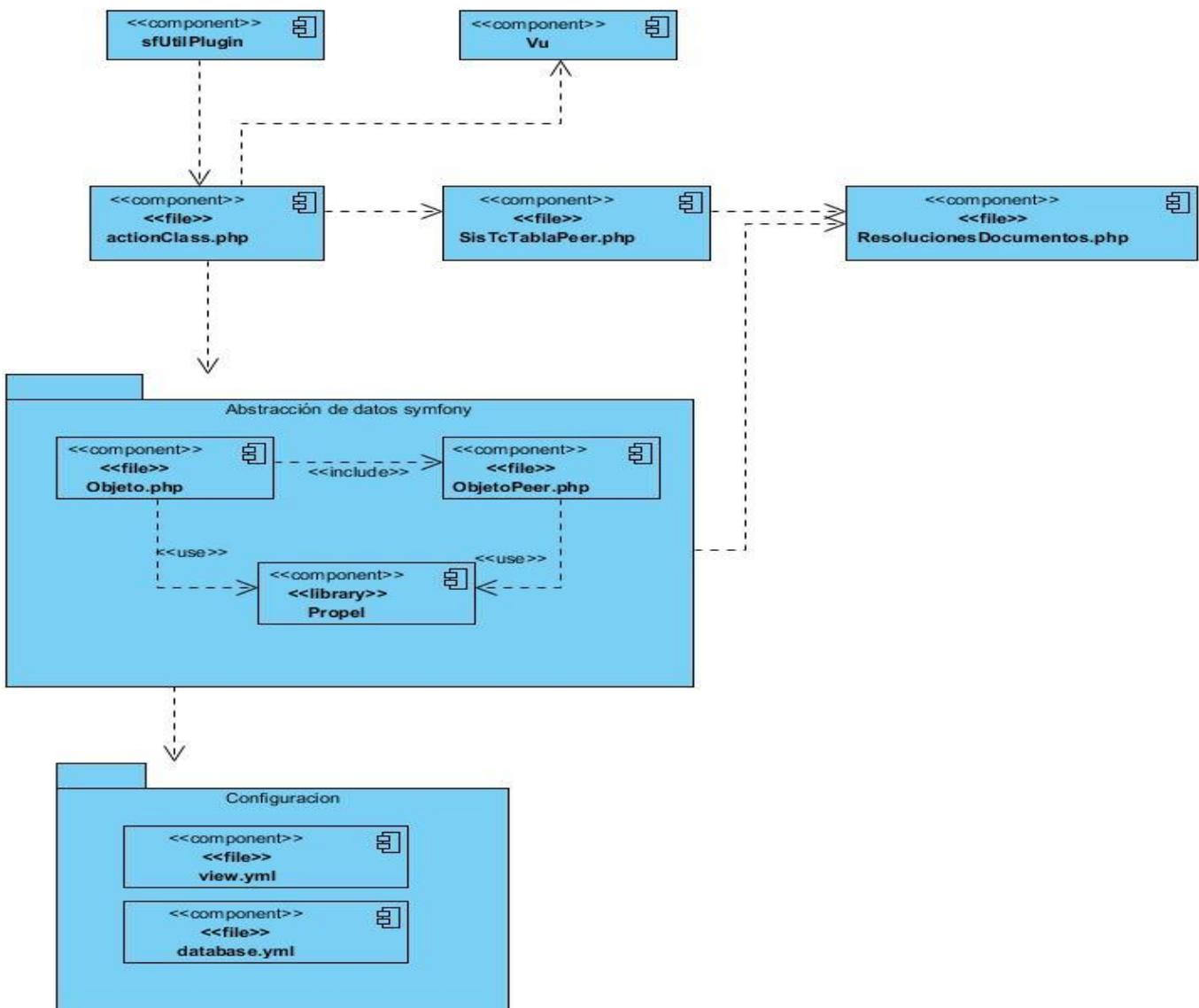


Figura 3.2: Diagrama de Componentes.

### 3.6. Prueba

El uso de las pruebas para la evaluación del software brinda la posibilidad de evaluar la calidad del producto desarrollado, aunque estas no aseguran que la calidad total del software, si permiten tener una óptica general del cumplimiento de los objetivos propuestos durante el diseño de la aplicación.

#### 3.6.1. Pruebas de Caja Negra.

Las pruebas de caja negra o de especificación tienen como objetivo fundamental verificar las relaciones de entrada y salida de una unidad. Este tipo de prueba persigue comprobar las funcionalidades que posee la aplicación, y aunque no verifica como se realizan estas funcionalidades dan una idea general de la calidad del software.

##### 3.6.1.1. Casos de prueba.

Los casos de pruebas pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce un resultado correcto. Permite obtener conjuntos de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. Intenta encontrar errores de las siguientes categorías: funciones incorrectas o ausentes; errores de interfaz, en estructuras de datos o en acceso a bases de datos externas; errores de rendimiento, de inicialización y de terminación.

Deben verificar:

- Si el producto satisface los requisitos del usuario, tal y como se describe en las especificación de los requisitos.
- Si el producto se comporta tal y como se describe en las especificaciones funcionales del diseño.

Existen muchos casos de prueba para determinar que un requisito es satisfactorio, para poder comprobar que todos los requisitos de una aplicación fueron revisados debe existir un caso de prueba para cada requisito y si el requisito tiene requisitos secundarios se debe hacer un caso de prueba para cada uno de los requisitos secundarios.

Durante la realización de las pruebas al sistema se realizaron 10 Casos de Pruebas (uno para cada requisito funcional. Luego de una detallada revisión se encontraron 4 no conformidades, en la primera

iteración, las cuales fueron corregidas, comprobándose la eliminación de las mismas en una segunda iteración, obteniéndose como resultado de la misma 0 no conformidades.

A continuación se muestra el caso de prueba para el requisito Registrar documento Resoluciones De Exención).

**Condición de ejecución**

Recibir el objeto del componente de intercambio de datos con la información de los datos del documento Resoluciones De Exención).

<b>Escenario</b>	<b>Descripción</b>	<b>Respuesta del sistema</b>	<b>Flujo central</b>
EC 1.1 Registrar Documentos.	Procesa datos, registrando, verifica y valida.	Registrar el listado de mercancías válidas para esa empresa en esta resolución.	1-Ejecuta el método. 2-Registra numero de la resolución. 3-Verifica que estén registradas las empresas. 4- Registra listado de empresas a la que va la resolución de exención. 5- Registra listado de contratos válidos. 6-Valida que la empresa pueda realizar la importación de ese tipo de mercancías. 7- Registra el listado de mercancías válidas y cantidad posible de mercancías a importar.
EC 1.2 No existen dichas empresas.	Almacena errores.	Almacena los errores para enviarlos posteriormente al componente de recepción.	1-Almacena errores. 2-Envía errores al componente de recepción.
EC 1.3 Empresa no autorizada a importar mercancías.	Almacena errores.	Almacena los errores para enviarlos posteriormente al componente de recepción.	1-Almacena errores. 2-Envía errores al componente de recepción.

EC 1.4 Envío de errores durante el proceso.	Detecta errores.	Envía en una lista al componente de intercambio todos los errores detectados.	1- Envía en una lista al componente de intercambio todos los errores detectados. 2- No se registran los datos del documento.
---	------------------	---	--

### 3.7. Conclusiones Parciales

Durante el desarrollo del capítulo fueron generados los artefactos correspondientes a la etapa de implementación, se realizó el diagrama de despliegue y el diagrama de componentes. En este capítulo se definieron los estándares de codificación que se utilizaron durante la implementación del componente, los que posibilitaron crear un código de fácil comprensión. El análisis de estos datos arrojó resultados satisfactorios por lo que concluye que la solución desarrollada se encuentra en condiciones para ser implantada.

## 4. Conclusiones

Al concluir la presente investigación se logró cumplir con los objetivos propuestos al inicio de la misma, arrojando como resultado un componente capaz de registrar las resoluciones ministeriales provenientes del Ministerio de Finanzas y Precios en la Aduana General de la República.

Durante el desarrollo de la primera etapa de la investigación se definieron las herramientas y tecnologías a utilizar para el desarrollo de la solución informática, además se realizó un estudio de algunos de los sistemas aduaneros existentes tanto a nivel nacional como internacional, análisis que permitió fundamentar las bases teóricas de la solución propuesta.

En un segundo período del desarrollo de la investigación fueron identificadas las principales funcionalidades de la solución, mediante el uso de diferentes técnicas para la captura de requisitos. Teniéndose como base el análisis realizado anteriormente, se examinaron y seleccionaron los patrones de diseño a utilizar en la solución planteada, los cuales definieron fundamentalmente la calidad con que se desarrollaría la implementación del software. Fueron diseñados los diagramas establecidos en la etapa de análisis y diseño, los cuales permitieron obtener una idea objetiva de la posterior implementación de la solución.

Durante la implementación de la solución propuesta se utilizaron herramientas, tecnologías y patrones que se definieron en fases anteriores de la investigación. Los resultados de esta etapa se evidenciaron mediante el correcto funcionamiento de la aplicación, la que fue validada a través el análisis de los resultados obtenidos de pruebas funcionales realizadas al sistema mediante las pruebas de caja negra, obteniéndose resultados satisfactorios. La realización de estas pruebas permitió además encontrar y corregir funcionalidades defectuosas en la aplicación, las que fueron solucionadas en su totalidad.

De forma general fueron solucionados todos los problemas existentes al inicio de la investigación, mediante la realización de las tareas definidas, obteniéndose como resultado el componente Resoluciones del Ministerio de Finanzas y Precios que posibilitará al sistema GINA la efectiva recepción de las resoluciones ministeriales, ofreciendo así una garantía para la comunicación entre la Aduana General de la República y el Ministerio de Finanzas y Precios.

## **5. Recomendaciones.**

A raíz de las conclusiones a las que se arriban con el desarrollo del trabajo, se recomienda:

1. Desarrollar nuevas funcionalidades para la recepción de resoluciones provenientes del Ministerio de Finanzas y Precios u otro ministerio con el cual la aduana posea algún interés en particular para su funcionamiento.
2. Que el sistema sea probado directamente en la AGR para certificar su correcto funcionamiento.

## 6. Referencia Bibliográfica

1. La UCI es fruto de la Batalla de Ideas | Atenas. In: [online]. [Accessed 11 abril 2013]. Available from: <http://www.atenas.cult.cu/node/3098>.
2. Universidad de Ciencias Informáticas. In: [online]. [Accessed 11 abril 2013]. Available from: <http://www.arqhys.com/arquitectura/universidad-ciencias-informaticas.html>.
3. ARTURO RAMÍREZ. La Aduana. In: [online]. [Accessed 24 enero 2013]. Available from: [http://www.aduanas.com.ve/boletines/boletin\\_12/aduana.htm](http://www.aduanas.com.ve/boletines/boletin_12/aduana.htm).
4. Aduana cubana. In: [online]. [Accessed 24 enero 2013]. Available from: <http://www.aduana.co.cu/>.
5. MFP. In: [online]. [Accessed 11 abril 2013]. Available from: <http://www.mfp.cu/mfp.php>.
6. Resoluciones Ministeriales. In: [online]. [Accessed 27 enero 2013]. Available from: [http://www.mef.gob.pe/index.php?option=com\\_content&view=section&id=48&Itemid=100357](http://www.mef.gob.pe/index.php?option=com_content&view=section&id=48&Itemid=100357).
7. Aduana Nacional de Bolivia 2 - RedGealc. In: [online]. [Accessed 28 enero 2013]. Available from: <http://www.redgealc.net/aduana-nacional-de-bolivia-2/conteudo/3697/pt/>.
8. Características de PHP. In: [online]. [Accessed 24 enero 2013]. Available from: [http://www.programacion.com/articulo/por\\_que\\_elegir\\_php\\_143](http://www.programacion.com/articulo/por_que_elegir_php_143).
9. ¿Por qué elegir PHP?. Programación en Castellano. In: [online]. [Accessed 22 abril 2013]. Available from: [http://www.programacion.com/articulo/por\\_que\\_elegir\\_php\\_143](http://www.programacion.com/articulo/por_que_elegir_php_143).
10. FABIEN POTENCIER, François Zaninotto. *Symfony, la guía definitiva*. S.l.: s.n., [no date].
11. FABIEN POTENCIER, François Zaninotto. *Symfony la guía definitiva*. 2008. S.l.: s.n., [no date].
12. Entornos integrados de desarrollo. In: [online]. [Accessed 27 enero 2013]. Available from: <http://curso-sobre.berlios.de/introsobre/2.0.1/sobre.html/sec-ide.html>.
13. netbeans. In: [online]. [Accessed 27 enero 2013]. Available from: [http://netbeans.org/index\\_es.html](http://netbeans.org/index_es.html).
14. Tutorial Visual Paradigm. In: [online]. [Accessed 29 enero 2013]. Available from: <http://es.scribd.com/doc/36636137/Tutorial-Visual-Paradigm>.
15. ¿Qué es un Sistema Gestor de Bases de Datos o SGBD? In: [online]. [Accessed 27 enero 2013]. Available from: <http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sgbd/>.
16. El SGBDR Oracle. In: [online]. [Accessed 27 enero 2013]. Available from: <http://www2.rhernando.net/modules/tutorials/doc/bd/oracle.html>.

17. Modelos de Desarrollo. In: [online]. [Accessed 22 abril 2013]. Available from: <http://www.slideshare.net/inventa2/modelos-de-desarrollo#btnLast>.
18. WILLIAM GONZÁLEZ OBREGÓN. *Modelo de Desarrollo de Software del Centro CEIGE*. S.l.: s.n.
19. JUAN JAVIER DANS MORENO. *Modelación e implementación del módulo Documentos del subsistema Depósitos de Aduana*. Habana: Universidad de las Ciencias Informáticas, 2011.
20. MARÍA JOSÉ ESCALONA y NORA KOCH. *Ingeniería de Requisitos en Aplicaciones para la Web – Un estudio comparativo*. S.l.: s.n., [no date].
21. IAN SOMMERVILLE. *Ingeniería de Software*. Séptima. S.l.: s.n., [no date]. ISBN 84-7829-074-5.
22. DANIEL ALBERTO OJEDA ESTRADA y REINIER SILVERIO FIGUEROA. *Diseño e implementación del módulo Despacho Comercial (DC) para el Sistema de Gestión Integral de Aduana*. Universidad de Ciencias Informáticas: s.n., 2012.
23. NATACHA ÁLVAREZ RODRÍGUEZ. *Diseño e implementación del Módulo Despacho de las Embarcaciones de Recreo para el Sistema de Gestión Integral de la Aduana*. S.l.: s.n., [no date].
24. Capitulo\_23 Métricas Técnicas Para Sistemas Orientados a Objetos. In: [online]. [Accessed 23 abril 2013]. Available from: <http://es.scribd.com/doc/32165537/Capitulo-23-Metricas-Tecnicas-Para-Sistemas-Orient-a-Dos-a-Objetos>.

---

## 7. Bibliografía consultada

**Collin, Peter.** Publishing Dictionary of Computing, 4th Edition.

**Conallen, Jim.** 1999. Modeling Web Application Architectures with UML. 1999.

**Markiewicz, Marcus Eduardo y de Lucena, Carlos J.P.** El Desarrollo del Framework Orientado al Objeto.

**Meléndrez, Edelsys Hernández.** 2006. Cómo escribir una tesis. La Habana: s.n., 2006.

**S., Carlos Mario Vélez.** 2005. Apuntes de metodología de la investigación. Medellín: s.n., 2005.

**Potencier, Fabien.** 2009. *El tutorial Jobeet.* 2009.

1. ANGEL ALVAREZ, M. *Qué es Java script.* [Consultado el: 13 enero de 2011].

Disponible en: <http://www.desarrolloWeb.com/articulos/25.php>.

2. Potencier, Fabien and Zaninotto, François. 2008. Guía definitiva de Symfony. 2008.

3. José Antonio Cobo Rodríguez. Línea Base Arquitectónica para el Polo Sistemas Tributarios y de Aduanas. Universidad de las Ciencias Informáticas, Ciudad de la Habana, Cuba: junio de 2008.

4. Yisel Rodríguez Pérez, Rafael Andrés Céspedes Basteiro. Diseño del Módulo Control de Personas del Sistema Único de Aduanas. Universidad de las Ciencias Informáticas, Ciudad de la Habana, Cuba: abril de 2009.

5. Roberto Carlos García Andino. Diseño e Implementación del Módulo Medios de Transporte Internacional. Universidad de las Ciencias Informáticas, Ciudad de la Habana, Cuba: junio 2009.

6. APACHE. Comunidad de desarrolladores y usuarios de apache [Consultado el: 15 febrero de 2011].  
Disponible en: <http://www.apache.org/>.

7. NETBEANS. Comunidad de usuarios de NetBeans [Consultado el: 14 enero de 2011].

Disponible en: <http://www.netbeans.org/>.

8. PRESSMAN. R. S. Ingeniería de software, un enfoque práctico. McGraw-Hill, 2001. [Consultado el: 24 marzo de 2011].

9. SÁNCHEZ, J. I. P. Metodología para el Desarrollo de Software. 2005, [Consultado el: 6 febrero de 2011]  
Disponible en: [www.lcc.uma.es/~jignacio/index\\_archivos/TEMA4.pdf](http://www.lcc.uma.es/~jignacio/index_archivos/TEMA4.pdf).

10. VISUAL PARADIGM INTERNATIONAL. Visual Paradigm for UML [Consultado el: 15 enero de 2011].

Disponible en: <http://www.visual-paradigm.com/product/vpuml/>.

11. Patrones de Diseño GRASP [Consultado el: 24 febrero de 2011].

Disponible en: <http://es.scribd.com/doc/53315954/Patrones-de-Diseno-GRASP>

68

12. Patrones de Diseño GoF [Consultado el: 24 febrero de 2011].

Disponible en: <http://es.scribd.com/doc/55128700/6/Los-Patrones-GoF-Originales>

13. DAEDALUS. 2006. Diseño de Sistemas. [Online] 2006.

Disponible en: <http://www.daedalus.es/AreasISDiseno-E.php>.

14. "Understanding and Using Patterns in Software Development." Riehle, D. y Züllighoven, H. 1996,

Theory and Practice of Object Systems.

15. Potencier, Fabien y Zaninotto, François. "Symfony | Open-Source PHP Web Framework". [En

Línea] Sitio Oficial del proyecto Symfony. <http://www.symfony-project.org>.

16. Rumbaugh, James. El Lenguaje Unificado de Modelado. Manual de Referencia.

17. ADUANA GENERAL DE LA REPÚBLICA. RESOLUCIÓN No. 187-2008. 2008.

18. ADUANA GENERAL DE LA REPUBLICA. Metodología Interna para el Despacho

19. Larman, Craig. Introducción al análisis y diseño orientado a objeto. UML y Patrones.

---

## 8. Glosario

**UCI:** Universidad de las Ciencias Informáticas.

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

**AGR:** Aduana General de la República de Cuba.

**CADI:** Centro de Automatización para la Información y la Dirección.

**GINA:** Gestión Integral Aduanera.

**Resoluciones ministeriales:** Es la norma cuya finalidad es ejecutar y supervisar la política general del estado dentro del ámbito de un ministerio, resolviendo casos concretos y particulares.

**Artefacto:** pieza de información tangible que es creada, modificada y usada por los trabajadores al realizar actividades.

**CASE:** *Computer Aided Software Engineering*, en su traducción al español significa Ingeniería de Software Asistida por Computación, es la aplicación de tecnología informática a las actividades, las técnicas y las metodologías propias de desarrollo de software, su objetivo es acelerar el proceso de automatizar o apoyar una o más fases del ciclo de vida del desarrollo de sistemas.

**IDE:** Entorno de Desarrollo Integrado. Consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI).

**Clase:** descripción de un conjunto de objetos que comparten los mismos atributos, operaciones, relaciones y semántica.

**PHP:** *PHP Hypertext Pre-processor* es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas.

**JSON:** *JavaScript Object Notation* es un formato ligero para el intercambio de datos. JSON es un subconjunto de la notación literal de objetos de JavaScript que no requiere el uso de XML.

**Symfony:** *framework* de trabajo para la realización de aplicaciones web escrito en PHP 5.

**Diagrama:** representación gráfica de un conjunto de elementos. Visualizan un sistema desde diferentes perspectivas.

**Especificación de requisitos:** captura los requerimientos de software para el sistema completo o una porción del mismo.

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

**Mercancías:** todos los bienes corporales muebles de comercio o no, con la sola excepción de los Efectos Personales de los viajeros.

**Software:** se refiere a los programas y datos almacenados en un ordenador.

**UML:** es el Lenguaje de Modelado Unificado para detallar, construir, visualizar y documentar las partes o artefactos de un software.

**ORM:** (*del inglés, Object Relational Mapping*) Mapeo de Objetos Relacional es una técnica de programación para convertir tipos de datos incompatibles entre sistemas de bases de datos y lenguajes orientados a objetos.

## 9. Anexos

### 9.1. Anexo 1: Descripción de requisitos

**Especificación del requisito Registrar documento resoluciones de Exención y Bonificación.**

<b>Precondiciones</b>	Recibir el objeto del componente de intercambio de datos con la información de los datos del documento Resoluciones De Exención y Bonificación.
<b>Flujo de eventos</b>	
<b>Flujo básico Registrar documento Resoluciones De Exención y Bonificación</b>	
1	Ejecuta el Método para el procesamiento de los datos del documento Resoluciones MFP, pasándole el objeto como parámetro.
2	Revisa que el número de resolución no exista.
3	Registra el número de la resolución de exención, además de los datos de fechas como son fecha de elaboración.
4	Revisa si el documento no tiene un tope a nivel de resolución.
5	Guarda el tope por el cual está hecha la resolución (Factura, Derechos, Valor Aduana).
6	Se verifican que las empresas estén registradas en el registro central de aduana así como que el código de la empresa beneficiada (código REEUP) exista.
7	Verifica que la fecha de inicio del beneficio y la fecha de fin del beneficio sea mayor que el día actual.
8	Registra el listado de empresas válidas al cual va dirigida la resolución de exención. Además de los datos específicos de la empresa (NIT, empresa beneficiada, fecha de inicio del beneficio, fecha de fin del beneficio, la empresa necesita declaración jurada). Para la empresa beneficiada la verifica por su código REEUP
9	En caso de existir se registran el listado de contratos válidos para esa empresa en dicha resolución.
10	En caso que venga el Excepto Protección (con valor 0 es para todas las mercancías

	del arancel, con valor 1 es para todo el arancel excepto las partidas protegidas en dicho anexo). Almacena dicha información.
11	En caso de que no se haya registrado el tope a nivel de resolución se registra el valor factura, valor de derechos o valor aduana.
12	Se valida que la mercancía este vigente en el arancel.
13	Se registra el listado de mercancías válidas para esa empresa en esta resolución. Por cada mercancía se registra Código SACLAP, bonificación, moneda.
14	Se registra también de las mercancías la cantidad posible a importar según esta resolución del MFP, en caso de que el tope sea por cantidad.
15	Concluye el requisito.
<b>Pos-condiciones</b>	
1	Se guardan los datos de la Resolución de Exención y Bonificación del MFP.
<b>Flujos alternativos</b>	
<b>Flujo alternativo 1.a Ya existe dicho número de exención.</b>	
1	Se almacena en una lista temporal el error para ser enviado posteriormente (Ya existe dicho número de resolución).
2	Continúa en el punto 16 del flujo principal.
<b>Pos-condiciones</b>	
1	Se almacenan los errores para enviarlos posteriormente.
<b>Flujo alternativo 2.a No existe dicha empresa.</b>	
1	Se almacena en una lista temporal el error para ser enviado posteriormente (no existe la empresa XXXXX).
2	Continúa en el punto 6 del flujo principal.
<b>Pos-condiciones</b>	
1	Se almacenan los errores para enviarlos posteriormente.
<b>Flujo alternativo 3.a No existen dicha empresa beneficiada.</b>	
1	Se almacena en una lista temporal el error para ser enviado posteriormente (no existe la empresa beneficiada XXXXX).
2	Continúa en el punto 6 del flujo principal.

<b>Pos-condiciones</b>	
	Se almacenan los errores para enviarlos posteriormente.
<b>Flujo alternativo 4.a Fechas menores a la fecha actual.</b>	
1	Se almacena en una lista temporal el error para ser enviado posteriormente (la fecha de inicio y fin del beneficio para la entidad XXXX deben ser mayores a las del día actual).
2	Continúa en el punto 6 del flujo principal.
<b>Pos-condiciones</b>	
	Se almacenan los errores para enviarlos posteriormente.
<b>Flujo alternativo 5.a Mercancía no existente en el arancel.</b>	
1	Se almacena en una lista temporal el error para ser enviado posteriormente (Mercancía no existente en el arancel).
2	Continúa en el punto 13 del flujo principal.
<b>Pos-condiciones</b>	
	Se almacenan los errores para enviarlos posteriormente.
<b>Flujo alternativo 6.a Enviar errores.</b>	
1	Envía en una lista al componente de intercambio todos los errores detectados.
2	No se registran los datos del documento.
3	Concluye el requisito.
<b>Pos-condiciones</b>	
1	Se envían los errores al componente de intercambio.
<b>Validaciones</b>	
2	Se validan los datos según lo establecido en el Modelo Conceptual
<b>Requisitos especiales</b>	No aplicable
<b>Asuntos pendientes</b>	No aplicable

**Prototipo elemental de interfaz gráfica de usuario**

No aplicable

## Formatos de entrada/salida

### Entradas

Objeto de PHP con la estructura del documento que se va a procesar en este caso con la estructura del documento Resoluciones de Exención y Bonificación.

### Salidas

La lista de errores será enviado en el formato JSON que se defina.

## Especificación de requisito Registrar documento Resoluciones de Cambio de Moneda.

<b>Precondiciones</b>	Recibir el objeto del componente de intercambio de datos con la información de los datos del documento Resoluciones de Condonación de la Deuda.
<b>Flujo de eventos</b>	
<b>Flujo básico Registrar documento Resoluciones de Cambio de Moneda</b>	
1	Ejecuta el Método para el procesamiento de los datos del documento Resoluciones MFP, pasándole el objeto como parámetro.
2	Revisa que el número de resolución no exista.
3	Registra el número de la resolución de Cambio de Moneda, además de la fecha de elaboración.
4	Se verifican que las empresas estén registradas en el registro central de aduana.
5	Verifica que el código de la empresa beneficiada (código REEUP), exista.
6	Verifica que la fecha de inicio del beneficio y la fecha de fin del beneficio sea mayor que el día actual.
7	Registra el listado de empresas válidas al cual va dirigida la resolución de cambio de moneda. Además de los datos específicos de la empresa (NIT, empresa beneficiada, fecha de inicio del beneficio, fecha de fin del beneficio).
8	En caso de existir se registran el listado de contratos válidos para esa empresa en dicha resolución, además del valor factura, valor de derechos, o valor aduana en

	caso de que el tope sea por empresa.
<b>9</b>	Se registra el listado de mercancías válidas para esa empresa en esta resolución. Por cada mercancía se registra Código SACLAP.
<b>10</b>	Se registra también de las mercancías la cantidad posible a importar según esta resolución del MFP, en caso de que el tope sea por cantidad.
<b>11</b>	Se registra a las mercancías la moneda de pago CUC.
<b>12</b>	Concluye el requisito.
<b>Pos-condiciones</b>	
<b>1</b>	Se guardan los datos de la Resolución de Cambio de Moneda del MFP.
<b>Flujos alternativos</b>	
<b>Flujo alternativo 1.a Ya existe dicho número de resolución.</b>	
<b>1</b>	Se almacena en una lista temporal el error para ser enviado posteriormente (Ya existe dicho número de resolución).
<b>2</b>	Continúa en el punto 12 del flujo principal.
<b>Pos-condiciones</b>	
<b>1</b>	Se almacenan los errores para enviarlos posteriormente.
<b>Flujo alternativo 2.a No existe dicha empresa.</b>	
<b>1</b>	Se almacena en una lista temporal el error para ser enviado posteriormente (No existe dicha empresa XXXXX).
<b>2</b>	Continúa en el punto 4 del flujo principal.
<b>Pos-condiciones</b>	
<b>1</b>	Se almacenan los errores para enviarlos posteriormente.
<b>Flujo alternativo 3.a No existen dicha empresa beneficiada.</b>	
<b>1</b>	Se almacena en una lista temporal el error para ser enviado posteriormente (no existe la empresa beneficiada XXXXX).
<b>2</b>	Continúa en el punto 4 del flujo principal.
<b>Pos-condiciones</b>	
	Se almacenan los errores para enviarlos posteriormente.
<b>Flujo alternativo 4.a Fechas menores a la fecha actual.</b>	

1	Se almacena en una lista temporal el error para ser enviado posteriormente (la fecha de inicio y fin del beneficio para la entidad XXXX deben ser mayores a las del día actual).
2	Continúa en el punto 4 del flujo principal.
<b>Pos-condiciones</b>	
	Se almacenan los errores para enviarlos posteriormente.
<b>Flujo alternativo 5.a Enviar errores.</b>	
1	Envía en una lista al componente de intercambio todos los errores detectados.
2	No se registran los datos del documento.
3	Concluye el requisito.
<b>Pos-condiciones</b>	
1	Se envían los errores al componente de intercambio.
<b>Validaciones</b>	
1	Se validan los datos según lo establecido en el Modelo Conceptual
<b>Requisitos especiales</b>	No aplicable
<b>Asuntos pendientes</b>	No aplicable

**Prototipo elemental de interfaz gráfica de usuario**

No aplicable

**Formatos de entrada/salida****Entradas**

Objeto de PHP con la estructura del documento que se va a procesar en este caso con la estructura del documento Resoluciones de Cambio de Moneda.

**Salidas**

La lista de errores será enviado en el formato JSON que se defina.

**Especificación de requisito Registrar documento Resoluciones de Ampliación de Exención y Bonificación.**

<b>Precondiciones</b>	Recibir el objeto del componente de intercambio de datos con la información de los datos del documento Resoluciones de Ampliación Exención y Bonificación.
<b>Flujo de eventos</b>	
<b>Flujo básico Registrar documento Resoluciones de Ampliación de Exención y Bonificación</b>	
1	Ejecuta el Método para el procesamiento de los datos del documento Resoluciones MFP, pasándole el objeto como parámetro.
2	Verifica que exista una resolución de Exención de referencia
3	Revisa que el número de resolución no exista.
4	Registra el número de la resolución de ampliación, el número de la resolución de referencia, además de fecha de elaboración.
5	En caso de venir un valor tope a nivel de resolución, se verifica si este es menor que el existente si esto ocurre se verifica si el valor consumido es mayor que el valor nuevo.
6	Se verifica si existen nuevos anexos a adicionar, en caso de existir se verifica que la fecha de inicio del beneficio y la fecha de fin del beneficio sea mayor que el día actual.
7	Se adiciona el anexo a la resolución de referencia, con la fecha de inicio y fecha de fin con que venga el anexo. El anexo debe contener todos los datos necesarios.

8	En caso de que en el fichero venga un anexo que ya exista. Se verifica si vienen nuevos contratos, los cuales son adicionados. Si existen nuevas mercancías las mismas se adicionan manteniendo la fecha de inicio y la fecha de fin del anexo al cual se están adicionados.
9	Concluye el requisito.
<b>Pos-condiciones</b>	
1	Se guardan los datos de la Resolución de Ampliación Exención y Bonificación del MFP.
<b>Flujos alternativos</b>	
<b>Flujo alternativo 1.a No existe dicha resolución de exención.</b>	
1	Se almacena en una lista temporal el error para ser enviado posteriormente (No existe dicha resolución de referencia).
2	Continúa en el punto 8 del flujo principal.
<b>Pos-condiciones</b>	
1	Se almacenan los errores para enviarlos posteriormente.
<b>Flujo alternativo 2.a Ya existe dicho número de ampliación.</b>	
1	Se almacena en una lista temporal el error para ser enviado posteriormente (Ya existe dicho número de resolución de ampliación).
2	Continúa en el punto 8 del flujo principal.
<b>Pos-condiciones</b>	
1	Se almacenan los errores para enviarlos posteriormente.

<b>Flujo alternativo 3.a Valor tope no se puede registrar.</b>	
1	Se almacena en una lista temporal el error para ser enviado posteriormente (El valor tope no puede ser menor que el valor ya consumido en la resolución).
2	Continúa en el punto 8 del flujo principal.
<b>Pos-condiciones</b>	
	Se almacenan los errores para enviarlos posteriormente.
<b>Flujo alternativo 4.a Fechas menores a la fecha actual.</b>	
1	Se almacena en una lista temporal el error para ser enviado posteriormente (la fecha de inicio y fin del beneficio para la entidad XXXX deben ser mayores a las del día actual).
2	Continúa en el punto 6 del flujo principal.
<b>Pos-condiciones</b>	
	Se almacenan los errores para enviarlos posteriormente.
<b>Flujo alternativo 5.a Existen errores al adicionar el anexo.</b>	
1	Se almacena en una lista temporal el error para ser enviado posteriormente (Existen errores al adicionar el anexo XXXXX).
2	Continúa en el punto 7 del flujo principal
<b>Pos-condiciones</b>	
1	Se envían los errores al componente de intercambio.
<b>Flujo alternativo 6.a Existen errores al adicionar las mercancías.</b>	
1	Se almacena en una lista temporal el error para ser enviado posteriormente (Existen errores al adicionar la mercancía XXXXX).
2	Continúa en el punto 7 del flujo principal.
<b>Pos-condiciones</b>	
1	Se envían los errores al componente de intercambio.
<b>Flujo alternativo 7.a Enviar errores.</b>	
1	Envía en una lista al componente de intercambio todos los errores detectados.
2	No se registran los datos del documento.
3	Concluye el requisito

<b>Pos-condiciones</b>	
1	Se envían los errores al componente de intercambio.
<b>Validaciones</b>	
1	Se validan los datos según lo establecido en el Modelo Conceptual
<b>Requisitos especiales</b>	No aplicable
<b>Asuntos pendientes</b>	No aplicable

### Prototipo elemental de interfaz gráfica de usuario

No aplicable

### Formatos de entrada/salida

#### Entradas

Objeto de PHP con la estructura del documento que se va a procesar en este caso con la estructura del documento Resoluciones de Ampliación de Exención y Bonificación.

#### Salidas

La lista de errores será enviado en el formato JSON que se defina.

### Especificación de requisito Registrar documento Resoluciones de Ampliación de Exención y Bonificación debido a cambio de nomenclatura.

<b>Precondiciones</b>	Recibir el objeto del componente de intercambio de datos con la información de los datos del documento Resoluciones de Ampliación Exención y Bonificación debido al cambio de nomenclatura.
<b>Flujo de eventos</b>	
<b>Flujo básico Registrar documento Resoluciones de Ampliación Exención y Bonificación debido al cambio de nomenclatura</b>	

---

1	Ejecuta el Método para el procesamiento de los datos del documento Resoluciones MFP, pasándole el objeto como parámetro.
2	Verifica que exista una resolución de Exención de referencia
3	Revisa que el número de resolución no exista.
4	Verifica que exista una resolución de cambio de nomenclatura de referencia.
5	Registra el número de la resolución de ampliación, el número de la resolución de referencia, además de fecha de elaboración.
6	En caso de venir un valor tope a nivel de resolución, se verifica si este es menor que el existente si esto ocurre se verifica si el valor consumido es mayor que el valor nuevo.
7	Se verifica si existen nuevos anexos a adicionar, en caso de existir se verifica que la fecha de inicio del beneficio y la fecha de fin del beneficio sea mayor que el día actual.
8	Se adiciona el anexo a la resolución de referencia, con la fecha de inicio y fecha de fin con que venga el anexo. El anexo debe contener todos los datos necesarios
9	En caso de que en el fichero venga un anexo que ya exista. Se verifica si vienen nuevos contratos, los cuales son adicionados. Si existen nuevas mercancías las mismas se adicionan manteniendo la fecha de inicio y la fecha de fin del anexo al cual se están adicionados.
10	Concluye el requisito.

<b>Pos-condiciones</b>	
1	Se guardan los datos de la Resolución de Ampliación Exención y Bonificación debido a cambio de nomenclatura del MFP.
<b>Flujos alternativos</b>	
<b>Flujo alternativo 1.a No existe dicha resolución de exención.</b>	
1	Se almacena en una lista temporal el error para ser enviado posteriormente (No existe dicha resolución de referencia).
2	Continúa en el punto 9 del flujo principal.
<b>Pos-condiciones</b>	
1	Se almacenan los errores para enviarlos posteriormente.
<b>Flujo alternativo 2.a Ya existe dicho número de ampliación.</b>	
1	Se almacena en una lista temporal el error para ser enviado posteriormente (Ya existe dicho número de resolución de ampliación).
2	Continúa en el punto 9 del flujo principal.
<b>Pos-condiciones</b>	
1	Se almacenan los errores para enviarlos posteriormente.
<b>Flujo alternativo 3.a No existe dicha resolución de cambio de nomenclatura.</b>	
1	Se almacena en una lista temporal el error para ser enviado posteriormente (No existe dicha resolución de cambio de nomenclatura).
2	Continúa en el punto 10 del flujo principal.
<b>Pos-condiciones</b>	
	Se almacenan los errores para enviarlos posteriormente.

**Flujo alternativo 4.a Valor tope no se puede registrar.**

1	Se almacena en una lista temporal el error para ser enviado posteriormente (El valor tope no puede ser menor que el valor ya consumido en la resolución).
2	Continúa en el punto 10 del flujo principal.

**Pos-condiciones**

	Se almacenan los errores para enviarlos posteriormente.
--	---

**Flujo alternativo 5.a Fechas menores a la fecha actual.**

1	Se almacena en una lista temporal el error para ser enviado posteriormente (la fecha de inicio y fin del beneficio para la entidad XXXX deben ser mayores a las del día actual).
2	Continúa en el punto 7 del flujo principal

**Pos-condiciones**

1	Se envían los errores al componente de intercambio.
---	---

**Flujo alternativo 6.a Existen errores al adicionar el anexo.**

1	Se almacena en una lista temporal el error para ser enviado posteriormente (Existen errores al adicionar el anexo XXXXX).
2	Continúa en el punto 8 del flujo principal.

**Pos-condiciones**

1	Se envían los errores al componente de intercambio.
---	---

**Flujo alternativo 7.a Existen errores al adicionar la mercancía.**

1	Se almacena en una lista temporal el error para ser enviado posteriormente (Existen errores al adicionar la mercancía XXXXX).
2	Continúa en el punto 9 del flujo principal.

**Pos-condiciones**

1	Se envían los errores al componente de intercambio.
---	---

**Flujo alternativo 8.a Enviar errores.**

1	Envía en una lista al componente de intercambio todos los errores detectados.
2	No se registran los datos del documento.
3	Concluye el requisito

<b>Pos-condiciones</b>	
1	Se envían los errores al componente de intercambio.
<b>Validaciones</b>	
1	Se validan los datos según lo establecido en el Modelo Conceptual
<b>Requisitos especiales</b>	No aplicable
<b>Asuntos pendientes</b>	No aplicable

### Prototipo elemental de interfaz gráfica de usuario

No aplicable

### Formatos de entrada/salida

#### Entradas

Objeto de PHP con la estructura del documento que se va a procesar en este caso con la estructura del documento Resoluciones de Ampliación de Exención y Bonificación debido a cambios en la nomenclatura.

#### Salidas

La lista de errores será enviado en el formato JSON que se defina.

### Especificación de requisito Registrar documento Resoluciones de Prorroga de Exención y Bonificación.

<b>Precondiciones</b>	Recibir el objeto del componente de intercambio de datos con la información de los datos del documento Resoluciones de Prorroga de Exención y Bonificación.
<b>Flujo de eventos</b>	
<b>Flujo básico Registrar documento Resoluciones de Prorroga de Exención y Bonificación</b>	
1	Ejecuta el Método para el procesamiento de los datos del documento Resoluciones MFP, pasándole el objeto como parámetro.

2	Verifica que el número de resolución de prórroga no exista.
3	Verifica que exista una resolución de Exención de referencia.
4	Verifica la existencia de los anexos en la resolución de referencia.
5	Verifica que la fecha de caducidad de los anexos la resolución de referencia sea un día menor que la fecha de inicio de la prórroga.
6	Registra el número de la resolución de Prórroga de Exención y Bonificación, además de los datos de fechas como son fecha de elaboración, fecha de inicio del beneficio y fecha de fin del beneficio. Además del número de resolución de referencia.
7	En caso de venir un tope para la resolución se modifica dicho valor (Factura, Derechos, Valor Aduana).
8	Concluye el requisito.
<b>Pos-condiciones</b>	
1	Se guardan los datos de la Resolución de Prórroga de Exención y Bonificación del MFP.
<b>Flujos alternativos</b>	
<b>Flujo alternativo 1.a Ya existe dicho número de resolución de prórroga.</b>	
1	Se almacena en una lista temporal el error para ser enviado posteriormente (Ya existe dicho número de resolución de prórroga).

2	Continúa en el punto 8 del flujo principal.
<b>Pos-condiciones</b>	
1	Se almacenan los errores para enviarlos posteriormente.
<b>Flujo alternativo 2.a no existe dicha resolución de referencia.</b>	
1	Se almacena en una lista temporal el error para ser enviado posteriormente (No existe dicha resolución de referencia).
2	Continúa en el punto 8 del flujo principal.
<b>Pos-condiciones</b>	
1	Se almacenan los errores para enviarlos posteriormente.
<b>Flujo alternativo 3.a No existe dicho anexo en la resolución de referencia.</b>	
1	Se almacena en una lista temporal el error para ser enviado posteriormente (No existe dicho anexo en la resolución de referencia).
2	Continúa en el punto 8 del flujo principal.
<b>Pos-condiciones</b>	
	Se almacenan los errores para enviarlos posteriormente.
<b>Flujo alternativo 4.a No coordinan la fecha de fin del anexo de la resolución y la de inicio de la prórroga.</b>	
1	Se almacena en una lista temporal el error para ser enviado posteriormente (No coordina la fecha de fin del anexo de la resolución y la fecha de inicio de la prórroga).
2	Continúa en el punto 8 del flujo principal.
<b>Pos-condiciones</b>	
	Se almacenan los errores para enviarlos posteriormente.
<b>Flujo alternativo 5.a Enviar errores.</b>	
1	Envía en una lista al componente de intercambio todos los errores detectados.

2	No se registran los datos del documento.
3	Concluye el requisito
<b>Pos-condiciones</b>	
1	Se envían los errores al componente de intercambio.
<b>Validaciones</b>	
1	Se validan los datos según lo establecido en el Modelo Conceptual
<b>Requisitos especiales</b>	No aplicable
<b>Asuntos pendientes</b>	No aplicable

### Prototipo elemental de interfaz gráfica de usuario

No aplicable

### Formatos de entrada/salida

#### Entradas

Objeto de PHP con la estructura del documento que se va a procesar en este caso con la estructura del documento Resoluciones de Prorroga de Exención y Bonificación.

#### Salidas

La lista de errores será enviado en el formato JSON que se defina.

### Especificación de requisito Registrar documento Resoluciones de Suspensión de Exención y Bonificación.

<b>Precondiciones</b>	Recibir el objeto del componente de intercambio de datos con la información de los datos del documento Resoluciones de Suspensión de Exención y Bonificación.
<b>Flujo de eventos</b>	
<b>Flujo básico Registrar documento Resoluciones de Suspensión de Exención y Bonificación.</b>	

1	Ejecuta el Método para el procesamiento de los datos del documento Resoluciones MFP, pasándole el objeto como parámetro.
2	Verifica que el número de resolución de Suspensión no exista.
3	Registra el número de la resolución de Suspensión, además de los datos de fechas como son fecha de elaboración y fecha de fin del beneficio.
4	Registra un listado de mercancías, donde registra el código de la mercancía y un listado de resoluciones donde va a ser eliminada esa mercancía. Verifica que cada una de estas resoluciones exista.
5	En cada una de las resoluciones registra a la empresa que se le va a eliminar dicha mercancía, registrándole la fecha de fin de la mercancía como la fecha de fin del beneficio.
6	Concluye el requisito.

**Pos-condiciones**

1	Se guardan los datos de la Resolución Suspensión de Exención y Bonificación del MFP.
---	--

**Flujos alternativos****Flujo alternativo 1.a Ya existe dicho número de resolución de suspensión.**

1	Se almacena en una lista temporal el error para ser enviado posteriormente (Ya existe dicha resolución de suspensión).
2	Continúa en el punto 6 del flujo principal.

**Pos-condiciones**

1	Se almacenan los errores para enviarlos posteriormente.
<b>Flujo alternativo 2.a No existen dicha resolución.</b>	
1	Se almacena en una lista temporal el error para ser enviado posteriormente (No existe dicha resolución).
2	Continúa en el punto 4 del flujo principal.
<b>Pos-condiciones</b>	
1	Se almacenan los errores para enviarlos posteriormente.
<b>Flujo alternativo 3.a Enviar errores.</b>	
1	Envía en una lista al componente de intercambio todos los errores detectados.
2	No se registran los datos del documento.
3	Concluye el requisito
<b>Pos-condiciones</b>	
1	Se envían los errores al componente de intercambio.
<b>Validaciones</b>	
1	Se validan los datos según lo establecido en el Modelo Conceptual
<b>Requisitos especiales</b>	No aplicable
<b>Asuntos pendientes</b>	No aplicable

**Prototipo elemental de interfaz gráfica de usuario**

No aplicable

**Formatos de entrada/salida****Entradas**

Objeto de PHP con la estructura del documento que se va a procesar en este caso con la estructura del documento Resoluciones de Suspensión de Exención y Bonificación.

### Salidas

La lista de errores será enviado en el formato JSON que se defina.

### Especificación de requisito Registrar documento Resoluciones de Suspensión de Exención y Bonificación debido a cambio de nomenclatura.

<b>Precondiciones</b>	Recibir el objeto del componente de intercambio de datos con la información de los datos del documento Resoluciones de Suspensión de Exención y Bonificación debido a cambio de nomenclatura.
<b>Flujo de eventos</b>	
<b>Flujo básico Registrar documento Resoluciones de Suspensión de Exención y Bonificación debido al cambio de nomenclatura</b>	
1	Ejecuta el Método para el procesamiento de los datos del documento Resoluciones MFP, pasándole el objeto como parámetro.
2	Verifica que el número de resolución de Suspensión no exista.
3	Verifica que exista una resolución de cambio de nomenclatura de referencia.
4	Registra el número de la resolución de Suspensión, además de los datos de fechas como son fecha de elaboración y fecha de fin del beneficio.
5	Registra un listado de mercancías, donde registra el código de la mercancía y un listado de resoluciones donde va a ser eliminada esa mercancía. Verifica que cada una de estas resoluciones exista.

6	En cada una de las resoluciones registra a la empresa que se le va a eliminar dicha mercancía, registrándole la fecha de fin de la mercancía como la fecha de fin del beneficio.
6	Concluye el requisito.
<b>Pos-condiciones</b>	
1	Se guardan los datos de la Resolución Suspensión de Exención y Bonificación debido al cambio de nomenclatura del MFP.
<b>Flujos alternativos</b>	
<b>Flujo alternativo 1.a Ya existe dicho número de resolución de suspensión.</b>	
1	Se almacena en una lista temporal el error para ser enviado posteriormente (ya existe dicho número de resolución de suspensión).
2	Continúa en el punto 7 del flujo principal.
<b>Pos-condiciones</b>	
1	Se almacenan los errores para enviarlos posteriormente.
<b>Flujo alternativo 2.a No existe dicha resolución de cambio de nomenclatura.</b>	
1	Se almacena en una lista temporal el error para ser enviado posteriormente (No existe dicha resolución de cambio de nomenclatura).
2	Continúa en el punto 7 del flujo principal.
<b>Pos-condiciones</b>	
1	Se almacenan los errores para enviarlos posteriormente.
<b>Flujo alternativo 3.a No existen dicha resolución.</b>	

1	Se almacena en una lista temporal el error para ser enviado posteriormente (No existe dicha resolución).
2	Continúa en el punto 5 del flujo principal.
<b>Pos-condiciones</b>	
1	Se almacenan los errores para enviarlos posteriormente.
<b>Flujo alternativo 4.a Enviar errores.</b>	
1	Envía en una lista al componente de intercambio todos los errores detectados.
2	No se registran los datos del documento.
3	Concluye el requisito
<b>Pos-condiciones</b>	
1	Se envían los errores al componente de intercambio.
<b>Validaciones</b>	
1	Se validan los datos según lo establecido en el Modelo Conceptual
<b>Requisitos especiales</b>	No aplicable
<b>Asuntos pendientes</b>	No aplicable

**Prototipo elemental de interfaz gráfica de usuario**

No aplicable

**Formatos de entrada/salida****Entradas**

Objeto de PHP con la estructura del documento que se va a procesar en este caso con la estructura del documento Resoluciones de Suspensión de Exención y Bonificación debido al cambio de nomenclatura.

**Salidas**

La lista de errores será enviado en el formato JSON que se defina.

**Especificación de requisito Registrar documento Resoluciones de Cancelación de Exención y Bonificación.**

<b>Precondiciones</b>	Recibir el objeto del componente de intercambio de datos con la información de los datos del documento Resoluciones de Cancelación de Exención y Bonificación.
<b>Flujo de eventos</b>	
<b>Flujo básico Registrar documento Resoluciones de Cancelación de Exención y Bonificación</b>	
1	Ejecuta el Método para el procesamiento de los datos del documento Resoluciones MFP, pasándole el objeto como parámetro.
2	Verifica que el número de resolución de Cancelación no exista.
3	Registra el número de la resolución de Cancelación, además de los datos de fechas como son fecha de elaboración y fecha de fin del beneficio.
4	Registra un listado de resoluciones a las cuales se desea cancelar. En caso de que no existan anexos para las resoluciones, se cancela la resolución completa, estas se cancelan con la fecha de fin del beneficio registrada anteriormente. Si existe algún anexo, se cancela dicho anexo.
6	Concluye el requisito.
<b>Pos-condiciones</b>	
1	Se guardan los datos de la Resolución Cancelación de Exención del MFP.
<b>Flujos alternativos</b>	
<b>Flujo alternativo 1.a Ya existe dicho número de resolución de cancelación.</b>	

1	Se almacena en una lista temporal el error para ser enviado posteriormente (Ya existe dicho número de resolución de cancelación).
2	Continúa en el punto 5 del flujo principal.
<b>Pos-condiciones</b>	
1	Se almacenan los errores para enviarlos posteriormente.
<b>Flujo alternativo 2.a No existe dicha resolución o anexo para cancelar.</b>	
1	Se almacena en una lista temporal el error para ser enviado posteriormente (No existe dicha resolución o anexo para cancelar).
2	Continúa en el punto 4 del flujo principal.
<b>Pos-condiciones</b>	
1	Se almacenan los errores para enviarlos posteriormente.
<b>Flujo alternativo 4.a Enviar errores.</b>	
1	Envía en una lista al componente de intercambio todos los errores detectados.
2	No se registran los datos del documento.
3	Concluye el requisito
<b>Pos-condiciones</b>	
1	Se envían los errores al componente de intercambio.
<b>Validaciones</b>	
1	Se validan los datos según lo establecido en el Modelo Conceptual
<b>Requisitos especiales</b>	No aplicable
<b>Asuntos pendientes</b>	No aplicable

**Prototipo elemental de interfaz gráfica de usuario**

No aplicable

**Formatos de entrada/salida****Entradas**

Objeto de PHP con la estructura del documento que se va a procesar en este caso con la estructura del documento Resoluciones objeto Cancelación de Exención y Bonificación.

**Salidas**

La lista de errores será enviado en el formato JSON que se defina.

**Especificación de requisito Registrar documento Resoluciones de Condonación de la Deuda.**

<b>Precondiciones</b>	Recibir el objeto del componente de intercambio de datos con la información de los datos del documento Resoluciones de Condonación de la Deuda.
<b>Flujo de eventos</b>	
<b>Flujo básico Registrar documento Resoluciones De Condonación de la Deuda</b>	
1	Ejecuta el Método para el procesamiento de los datos del documento Resoluciones MFP, pasándole el objeto como parámetro.
2	Revisa que el número de resolución no exista.
3	Registra el número de la resolución de condonación de la deuda, además de la fecha de elaboración.
4	Se verifica que existan los números de las Declaraciones de Mercancías.

5	Se verifica que las mercancías de una DM estén relacionadas en la DM original.
6	Registra el listado de las declaraciones de mercancías con sus mercancías específicas.
7	Se le cancela la deuda a las declaraciones de mercancías registradas.
8	Concluye el requisito.
<b>Pos-condiciones</b>	
1	Se guardan los datos de la Resolución de Condonación de la Deuda del MFP.
<b>Flujos alternativos</b>	
<b>Flujo alternativo 1.a Ya existe dicho número de resolución de condonación de la deuda.</b>	
1	Se almacena en una lista temporal el error para ser enviado posteriormente (ya existe dicho número de condonación de la deuda).
2	Continúa en el punto 8 del flujo principal.
<b>Pos-condiciones</b>	
1	Se almacenan los errores para enviarlos posteriormente.
<b>Flujo alternativo 2.a No existe dicha declaración de mercancía.</b>	
1	Se almacena en una lista temporal el error para ser enviado posteriormente (no existe dicho número de declaración de mercancía).

2	Continúa en el punto 4 del flujo principal.
<b>Pos-condiciones</b>	
1	Se almacenan los errores para enviarlos posteriormente.
<b>Flujo alternativo 3.a No existe esta mercancía en la DM original.</b>	
1	Se almacena en una lista temporal el error para ser enviado posteriormente (no existe dicha mercancía en la DM original).
2	Continúa en el punto 5 del flujo principal.
<b>Pos-condiciones</b>	
1	Se almacenan los errores para enviarlos posteriormente.
<b>Flujo alternativo 4.a Enviar errores.</b>	
1	Envía en una lista al componente de intercambio todos los errores detectados.
2	No se registran los datos del documento.
3	Concluye el requisito
<b>Pos-condiciones</b>	
1	Se envían los errores al componente de intercambio.
<b>Validaciones</b>	
1	Se validan los datos según lo establecido en el Modelo Conceptual
<b>Requisitos especiales</b>	No aplicable
<b>Asuntos pendientes</b>	No aplicable

**Prototipo elemental de interfaz gráfica de usuario**

No aplicable

**Formatos de entrada/salida**

**Entradas**

Objeto de PHP con la estructura del documento que se va a procesar en este caso con la estructura del documento Resoluciones De Condonación de la Deuda

**Salidas**

La lista de errores será enviado en el formato JSON que se defina.

**Especificación de requisito Registrar documento Resoluciones De Modificación del Valor Exento.**

<b>Precondiciones</b>	Recibir el objeto del componente de intercambio de datos con la información de los datos del documento Resoluciones De Modificación del Valor Exento.
<b>Flujo de eventos</b>	
<b>Flujo básico Registrar documento Resoluciones De Modificación del Valor Exento</b>	
1	Ejecuta el Método para el procesamiento de los datos del documento Resoluciones MFP, pasándole el objeto como parámetro.
2	Revisa que el número de resolución no exista.
3	Registra el número de la resolución de exención, además de los datos de fechas como son fecha de elaboración.
4	Revisa si el documento tiene un tope a nivel de resolución.
5	Guarda el tope por el cual está hecha la resolución (Factura, Derechos, Valor Aduana).

6	Se verifican que las empresas estén registradas en el registro central de aduana.
7	Registra el listado de empresas válidas al cual va dirigida la resolución de exención. Además de los datos específicos de la empresa (NIT, empresa beneficiada, fecha de inicio del beneficio, fecha de fin del beneficio, la empresa necesita declaración jurada).
8	Concluye el requisito.
<b>Pos-condiciones</b>	
1	Se guardan los datos de la Resolución de Modificación del Valor Exento del MFP.
<b>Flujos alternativos</b>	
<b>Flujo alternativo 1.a Ya existe dicho número de exención.</b>	
1	Se almacena en una lista temporal el error para ser enviado posteriormente (Ya existe dicho número de resolución).
2	Continúa en el punto 3 del flujo principal.
<b>Pos-condiciones</b>	
1	Se almacenan los errores para enviarlos posteriormente.
<b>Flujo alternativo 2.a No existen dichas empresas.</b>	
1	Se almacena en una lista temporal el error para ser enviado posteriormente (no existe la empresa XXXXX).
2	Diferencia dicha empresa como No válida.

3	Continúa en el punto 7 del flujo principal.
<b>Pos-condiciones</b>	
1	Se almacenan los errores para enviarlos posteriormente.
<b>Flujo alternativo 3.a Empresa no autorizada a importar mercancías.</b>	
1	Almacena los errores para enviarlos posteriormente al componente de recepción.
2	Continúa en el punto 8 del flujo principal.
<b>Pos-condiciones</b>	
1	Se almacenan los errores para enviarlos posteriormente.
<b>Flujo alternativo 4.a Enviar errores.</b>	
1	Envía en una lista al componente de intercambio todos los errores detectados.
2	No se registran los datos del documento.
3	Concluye el requisito
<b>Pos-condiciones</b>	
1	Se envían los errores al componente de intercambio.
<b>Validaciones</b>	
1	Se validan los datos según lo establecido en el Modelo Conceptual
<b>Requisitos especiales</b>	No aplicable
<b>Asuntos pendientes</b>	No aplicable

**Prototipo elemental de interfaz gráfica de usuario**

No aplicable

**Formatos de entrada/salida**

**Entradas**

Objeto de PHP con la estructura del documento que se va a procesar en este caso con la estructura del documento Resoluciones De Modificación del Valor Exento.

**Salidas**

La lista de errores será enviado en el formato JSON que se defina.

**9.2. Anexo 2: Casos de prueba****Condiciones de ejecución**

Recibir el objeto del componente de intercambio de datos con la información de los datos del documento Resoluciones De Bonificación).

**SC: Registrar documento Resoluciones De Bonificación).**

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Registrar Documentos.	Registra, verifica y valida.	Registra para cada mercancía la bonificación que se le aplicará.	1- Ejecuta el método. 2- Registra número de la resolución. 3- Verifica que estén registradas las empresas. 4- Registra el listado de contratos válidos. 5- Valida que la empresa pueda realizar la importación de ese tipo de mercancías. 6- Registra el listado de mercancías válidas. 7- Registra también de las mercancías la cantidad posible a importar. 8- Registra para cada mercancía la bonificación que se le aplicará.
EC 1.2 No existen	Almacena y	Almacena en una lista	1- Almacena en una lista temporal

dichas empresas.	envía	temporal el error para ser enviado posteriormente.	el error. 2- Envía error posteriormente.
EC 13 Empresa no autorizada a importar mercancías.	Almacena y envía	Almacena los errores para enviarlos al componente de recepción.	1- Almacena los errores. 2- Envía errores al componente de recepción.
EC 1.4 Enviar errores.	Detecta errores.	Envía en una lista al componente de intercambio todos los errores detectados.	1- Envía en una lista al componente de intercambio todos los errores detectados. 2- No se registran los datos del documento.

### Condiciones de ejecución

Recibir el objeto del componente de intercambio de datos con la información de los datos del documento Resoluciones De Cambio de Moneda).

### SC: Registrar documento Resoluciones De Cambio de Moneda).

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Registrar Documentos.	Procesa datos, registra, verifica y valida.	Hace el cambio de moneda para CUC.	1- Ejecuta el método. 2- Registra número de la resolución y el listado de empresas. 3- Verifica que estén registradas las empresas. 4- Registran el

			<p>listado de contratos válidos. 5- Valida que la empresa pueda realizar la importación. 6- Registra el listado de mercancías válidas para esa empresa. 7- Registra las mercancías. 8- Se les hace el cambio de moneda para CUC.</p>
EC 1.2 No existen dichas empresas.	Almacena y envía.	Almacena en una lista temporal el error para ser enviado posteriormente.	1- Almacena en una lista temporal el error. 2- Envía error posteriormente.
EC 1.3 Empresa no autorizada a importar mercancías.	Almacena y envía.	Almacena los errores para enviarlos al componente de recepción.	1- Almacena los errores. 2- Envía errores al componente de recepción.

EC 1.4 Enviar errores.	Detecta errores.	Envía en una lista al componente de intercambio todos los errores detectados.	1- Envía en una lista al componente de intercambio todos los errores detectados. 2- No se registran los datos del documento.
------------------------	------------------	---	--

### Condiciones de ejecución

Recibir el objeto del componente de intercambio de datos con la información de los datos del documento Resoluciones De Ampliación Exención).

### SC: Registrar documento Resoluciones De Ampliación Exención).

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Registrar Documentos.	Procesa datos, verifica y registra.	Verificar existencia de resolución.	1- Ejecuta el método. 2- Verifica que exista la resolución Exención de referencia. 3- Registra el número de la resolución de

			ampliación de exención, listado de las empresas.
EC 1.2 No existe dicha resolución de exención.	Verifica, almacena y envía.	Almacena en una lista temporal el error para ser enviado posteriormente.	1- Almacena en una lista temporal el error. 2- Envía error posteriormente.
EC 1.3 Enviar errores.	Registra y detecta errores.	Envía en una lista al componente de intercambio todos los errores detectados.	1- Envía en una lista al componente de intercambio todos los errores detectados. 2- No se registran los datos del documento.

### Condiciones de ejecución

Recibir el objeto del componente de intercambio de datos con la información de los datos del documento Resoluciones De Ampliación Bonificación).

### SC: Registrar documento Resoluciones De Ampliación Bonificación).

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Registrar Documentos.	Procesa de datos,	Registra el listado de las	1- Ejecuta el

	verifica y registra.	empresas, contratos o mercancías.	<p>método. 2- Verifica que exista una resolución de Bonificación de referencia.</p> <p>3- Registra el número de la resolución de ampliación de Bonificación además de los datos de fechas.</p> <p>4- Registra el listado de las empresas, contratos o mercancías.</p>
EC 1.2 No existe dicha resolución de Bonificación.	Verifica, almacena y envía.	Almacena en una lista temporal el error para ser enviado posteriormente.	1- Almacena en una lista temporal el error. 2- Envía error posteriormente.
EC 1.3 Enviar errores.	Registra errores.	Envía en una lista al componente de intercambio todos los errores detectados.	1- Envía en una lista al componente de intercambio todos los

			errores detectados. 2- No se registran los datos del documento.
--	--	--	---

### Condiciones de ejecución

Recibir el objeto del componente de intercambio de datos con la información de los datos del documento Resoluciones De Exención Eventual).

### SC: Registrar documento Resoluciones De Exención Eventual).

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Registrar Documentos.	Procesa datos, verifica que exista la resolución.	Registra el listado de mercancías válidas y cantidad posible a importar.	1- Ejecuta el método. 2- Verifica que exista una resolución de Bonificación de referencia. 3- Registra el número de la resolución Eventual. 4- Verifican que las empresas estén registradas en el registro

			<p>central de aduana. 5- Registra el listado de empresas al cual va dirigida la resolución. 6- Registra contratos válidos para esa empresa. 7- Valida que se pueda realizar la importación mercancías. 8- Registra el listado de mercancías válidas y cantidad posible a importar.</p>
EC 1.2 No existe dicha resolución.	Verifica que no existe la resolución, almacena y envía	Almacena en una lista temporal el error para ser enviado posteriormente.	1- Almacena en una lista temporal el error. 2- Envía error posteriormente.

EC 1.3 No existen dichas empresas.	Se ejecuta este escenario cuando verifica que no existe la resolución.	Almacena en una lista temporal el error para ser enviado posteriormente.	1- Almacena en una lista temporal el error. 2- Envía error posteriormente. 3- En caso de no existir ninguna empresa válida el requisito termina.
EC 1.3 Empresa no autorizada a importar dicha mercancía	Verifica que no existe la resolución, almacena y envía.	Almacena en una lista temporal el error para ser enviado posteriormente.	1- Almacena en una lista temporal el error. 2- Envía error posteriormente. 3- En caso de no existir ninguna empresa válida el requisito termina.
EC 1.3 Enviar errores.	Registra y envía errores.	Envía en una lista al componente de intercambio todos los errores detectados.	1- Envía en una lista al componente de intercambio todos los errores

			detectados. 2- No se registran los datos del documento.
--	--	--	--

### Condiciones de ejecución

Recibir el objeto del componente de intercambio de datos con la información de los datos del documento Resoluciones De Prorroga de Exención).

### SC: Registrar documento Resoluciones De Prorroga de Exención).

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Registrar Documentos.	Procesa datos, verifica y registra.	Registra el número de la resolución de Prórroga de Exención.	1- Ejecuta el método. 2- Verifica que exista una resolución de Exención de referencia, fecha de caducidad. 3- Registra el número de la resolución de Prórroga de Exención.
EC 1.2 No existe dicha	Verifica que no existe la	Almacena en una	1- Almacena en

resolución.	resolución, almacena y envía	lista temporal el error para ser enviado posteriormente.	una lista temporal el error. 2- Envía error posteriormente.
EC 1.2 No coordinan la fecha de fin de la resolución y la de inicio de la prórroga.	Verifica que no existe la fecha de caducidad de la resolución, almacena y envía.	Almacena en una lista temporal el error para ser enviado posteriormente.	1- Almacena en una lista temporal el error. 2- Envía error posteriormente.
EC 1.3 Enviar errores.	Registrar errores durante el proceso de envío.	Envía en una lista al componente de intercambio todos los errores detectados.	1- Envía en una lista al componente de intercambio todos los errores detectados. 2- No se registran los datos del documento.

### Condiciones de ejecución

Recibir el objeto del componente de intercambio de datos con la información de los datos del documento Resoluciones De Prorroga de Bonificación).

**SC: Registrar documento Resoluciones De Prorroga de Bonificación).**

Escenario	Descripción	Respuesta del	Flujo central
-----------	-------------	---------------	---------------

		<b>sistema</b>	
EC 1.1 Registrar Documentos.	Procesa datos, verifica y registra.	Registra el número de la resolución de Prórroga de Bonificación	1- Ejecuta el método. 2- Verifica que exista una resolución de Bonificación de referencia. 3- Verifica fecha de caducidad de la resolución de referencia. 4- Registra el número de la resolución de Prórroga de Bonificación
EC 1.2 No existe dicha resolución.	Verifica que no existe la resolución de Exención de referencia, almacena y envía	Almacena en una lista temporal el error para ser enviado posteriormente.	1- Almacena en una lista temporal el error. 2- Envía error posteriormente.
EC 1.3 No coordinan la fecha de fin de la resolución y la de inicio de la prórroga	Verifica que no existe la fecha de caducidad de la resolución, almacena y envía	Almacena en una lista temporal el error para ser enviado posteriormente.	3- Almacena en una lista temporal el error. 2- Envía error posteriormente.

EC 1.4 Enviar errores.	Registra errores durante el proceso, almacena y envía.	Envía en una lista al componente de intercambio todos los errores detectados.	1- Envía en una lista al componente de intercambio todos los errores detectados. 2- No se registran los datos del documento.
------------------------	--	---	--

### Condiciones de ejecución

Recibir el objeto del componente de intercambio de datos con la información de los datos del documento Resoluciones De Conjunta).

### SC: Registrar documento Resoluciones De Conjunta).

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Registrar Documentos.	Procesa datos, registra y verifica.	Registra el listado de empresas, contratos válidos, mercancías y cantidad posible a importar según esta resolución del MFP.	1- Ejecuta el método. 2- Registra el número de la resolución Conjunta y datos de fecha. 3- Verifica que las empresas estén registradas en el registro central de aduana. 4- Registra el listado de empresas, contratos válidos, mercancías y cantidad posible a importar según esta resolución del MFP.

EC 1.2 No existen dichas empresas.	Verifica que no están registradas las empresas , almacena y envía.	Almacena en una lista temporal el error para ser enviado posteriormente	1- Almacena en una lista temporal el error. 2- Envía error posteriormente. 3- Verifica que hayan empresas válidas.
EC 1.3 Empresa no autorizada a importar mercancías.	Verifica que la empresa no pueda realizar la importación, almacena y envía.	Almacena los errores para enviarlos posteriormente al componente de recepción.	1- Almacena los errores. 2- Envía errores al componente de recepción.
EC 1.5 Registrar mercancías a importar.	Registra cantidad de mercancías, almacena y envía.	Almacena los errores para enviarlos posteriormente al componente de recepción.	2- Almacena los errores. 2- Envía errores al componente de recepción.
EC 1.6 Enviar errores.	Verifica y envía errores.	Envía en una lista al componente de intercambio todos los errores detectados.	1- Envía en una lista al componente de intercambio todos los errores detectados. 2- No se registran los datos del documento.

### Condiciones de ejecución

Recibir el objeto del componente de intercambio de datos con la información de los datos del documento Resoluciones De Suspensión de Exención).

### SC: Registrar documento Resoluciones De Suspensión de Exención).

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Registrar Documentos.	Procesa datos y registra.	Registra el número de la resolución de Suspensión de	1- Ejecuta el método. 2- Registra el

		Exención, datos de fechas, registra un listado de mercancías y resoluciones donde va a ser eliminada.	número de la resolución Suspensión de Exención, datos de fechas, registra un listado de mercancías y resoluciones donde va a ser eliminada.
EC 1.2 No existen dicha combinación de resolución empresa y mercancía.	Verifica que no están registradas las empresas que se le va a eliminar mercancía. Almacena y envía	Se almacena en una lista temporal el error para ser enviado posteriormente.	1- Se almacena en una lista temporal el error. 2- Envía errores posteriormente.
EC 1.3 Enviar errores.	Verifica y envía errores.	Envía en una lista al componente de intercambio todos los errores detectados.	1- Envía en una lista al componente de intercambio todos los errores detectados. 2- No se registran los datos del documento.

### Condiciones de ejecución

Recibir el objeto del componente de intercambio de datos con la información de los datos del documento Resoluciones De Suspensión de Bonificación).

**SC: Registrar documento Resoluciones De Suspensión de Bonificación).**

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Registrar Documentos.	Procesa datos y registra.	Registra el número de la resolución de Suspensión de Bonificación, listado de mercancías, y resoluciones donde va a ser eliminada.	1- Ejecuta el método. 2- Registra el número de la resolución, listado de mercancías, y resoluciones donde va a ser eliminada.
EC 1.2 No existen dicha combinación de resolución empresa y mercancía.	Verifica que no están registradas las empresas que se le va a eliminar mercancía. Almacena y envía.	Se almacena en una lista temporal el error para ser enviado posteriormente.	1- Se almacena en una lista temporal el error. 2- Envía errores posteriormente.
EC 1.3 Enviar errores.	Se ejecuta este escenario cuando verifica errores durante el proceso.	Envía en una lista al componente de intercambio todos los errores detectados.	1- Envía en una lista al componente de intercambio todos los errores detectados. 2- No se registran los datos del documento.

**Condiciones de ejecución**

Recibir el objeto del componente de intercambio de datos con la información de los datos del documento Resoluciones De Cancelación de Exención).

**SC: Registrar documento Resoluciones De Cancelación de Exención).**

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Registrar Documentos.	Procesa datos y registra	Registra el número de la resolución, listado de resoluciones a cancelar, un listado de mercancías y resoluciones donde va a ser eliminada.	1- Ejecuta el método. 2- Registra el número de la resolución, listado de resoluciones a cancelar, un listado de mercancías y resoluciones donde va a ser eliminada.
EC 1.2 No existen dicha resolución para cancelar.	Se ejecuta este escenario cuando verifica que no están registradas alguna de estas resoluciones.	Se almacena en una lista temporal el error para ser enviado posteriormente.	1- Se almacena en una lista temporal el error. 2- Envía errores posteriormente.
EC 1.3 Enviar errores.	Se ejecuta este escenario cuando verifica errores durante el proceso.	Envía en una lista al componente de intercambio todos los errores detectados.	1- Envía en una lista al componente de intercambio todos los errores detectados. 2- No

		se registran los datos del documento.
--	--	---------------------------------------

### Condiciones de ejecución

Recibir el objeto del componente de intercambio de datos con la información de los datos del documento Resoluciones De Cancelación de Bonificación).

### SC: Registrar documento Resoluciones De Cancelación de Bonificación).

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Registrar Documentos.	Procesa y registra datos.	Registra el número de la resolución de Cancelación de Bonificación, listado de resoluciones a cancelar.	1- Ejecuta el método. 2- Registra el número de la resolución de Cancelación de Bonificación, listado de resoluciones a cancelar.
EC 1.2 No existen dicha resolución para cancelar.	Verifica que no están registradas alguna de estas resoluciones. Almacena y envía.	Se almacena en una lista temporal el error para ser enviado posteriormente.	1- Se almacena en una lista temporal el error. 2- Envía errores posteriormente.

---

EC 1.3 Enviar errores.	Verifica errores durante el proceso.	Envía en una lista al componente de intercambio todos los errores detectados.	1- Envía en una lista al componente de intercambio todos los errores detectados. 2- No se registran los datos del documento.
------------------------	--------------------------------------	---	---