

Universidad de las Ciencias Informáticas
FACULTAD 4



Título: SISTEMA PARA EL PROCESO DE CONTROL DE LA EJECUCIÓN DE LA PLANIFICACIÓN MATERIAL Y FINANCIERA REALIZADA POR LOS ÓRGANOS CONSUMIDORES DEL APARATO CENTRAL DEL MINFAR.

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

Autor: Danay Puentes Liza.

Tutor: Ing. Saumel Tejeda Díaz.

Ciudad de La Habana, Julio de 2008

“Año del 50 Aniversario del Triunfo de la Revolución”

DECLARACIÓN DE AUTORÍA

Declaro ser la única autora de este trabajo y reconozco a la Unidad de Compatibilización, Integración y Desarrollo (UCID) y a la Universidad de las Ciencias Informáticas (UCI) los derechos patrimoniales del mismo, con carácter exclusivo.

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

Danay Puentes Liza

Firma del Autor

Saumel Tejeda Díaz

Firma del Tutor

DATOS DE CONTACTO

El tutor de este trabajo de diploma es el Ing. .Saumel Tejeda Díaz, graduado en el año 2005 en el ITM, Instituto Técnico Militar, tiene 3 años de experiencia .Actualmente trabaja en el centro UCID, Unidad de Compatibilización, Integración y Desarrollo, donde ejerce las funciones de Jefe de Producción y Jefe de Línea de investigación, su dirección de correo electrónico es stejeda@uci.cu .



“ Si alguna vez nuestro trabajo nos pareciera bueno, debemos luchar por hacerlo mejor, luchar por hacerlo perfecto, sabiendo de antemano que no hay obra humana totalmente perfecta. ”

Fidel Castro Ruz.

AGRADECIMIENTOS

A Fidel Castro por brindarme la posibilidad de formarme profesionalmente, a mi mamá por representar todo en mi vida, a mis abuelos, a mi hermanita Daray por ser el regalo más valioso que me ha dado la vida, a Angelito por brindarme siempre su ayuda, a mi tía Elsa y a Lecusay, por saber que puedo contar con ellos, a mi prima Yanefys y a Yoan por su confianza y cariño, a Mairita por su ayuda y preocupación por mis problemas, a toda mi familia en general.

De manera muy particular a mi tutor Saumel, a su novia Gerxys y a Sánchez por alentarme y darme fuerza en los momentos que creía que ya no podía continuar, muchas gracias por su ayuda y apoyo.

Según dicen el agradecimiento es la memoria del corazón y no podría dejar de agradecer a todas las personas que sin tener vínculos carnales me han brindado siempre su apoyo y cariño incondicionalmente, personas tan especiales como Yanefys Díaz por los lazos de amistad que nos han mantenido unidas desde mis estudios en la Vocacional, a Kenia, a Yany; a personas inolvidables y excepcionales que conocí en la UCI y han marcado una pauta en mi vida como Yanet que le agradezco mucho por sus consejos y preocupación, a Leiser por haberme soportado tantas malcriadeces y llantos, por su preocupación por mis estudios, por su ayuda y por ser el amor de mi vida, a Yanelis Pupo por tantas cosas lindas que hizo por mi, a Mercy por ayudarme tanto, comprenderme y tolerar mi mal carácter, a Mavi por su ayuda incondicional y por haberse convertido en una amiga maravillosa, a Nilien por darme la posibilidad de contar con ella siempre que lo necesité, por estar a mi lado en buenos y malos momentos, a Nimia por su confianza y cariño, a Yisel Pupo por ser una amiga extraordinaria y por el apoyo que siempre encontré en ella, al Gallego por haberme ayudado desde mi primer año y por su preocupación por mi bienestar; a Yosbel y Yela por brindarme su amistad en el transcurso de este curso; tantas personas que quizás no pueda volver a ver pero que siempre estarán en mi corazón.

DEDICATORIA

Dedico este triunfo a mi mamá Elva Liza por representar todo en mi vida , por haberme ayudado a alcanzar este sueño dándome las fuerzas y el aliento para seguir adelante en los momentos de flaqueza; a mi abuelita Elda Palmero por el amor que siempre me ha brindado, por su comprensión y por ser la abuela más buena del mundo; a mi abuelito Onelio Liza por ser tan especial para mi ,por su preocupación y cariño, a Angelito por su apoyo y el empeño que siempre ha puesto en educarme y por darme la posibilidad de conocer lo que significa el cariño de un padre; a mi hermanita Daray Amanda por el amor y cariño que siento por ella.

A toda mi familia en general y a mis amistades que de una forma u otra han colocado su granito de arena en mi formación espiritual y profesional.

RESUMEN

El Ministerio de las Fuerzas Armadas Revolucionarias(MINFAR) que tiene como principal objetivo social la defensa de la Revolución, se encuentra dentro de un proceso de informatización que incluye la implantación de Sistemas Empresariales, Planeamiento de Recursos Estratégicos para las Empresas(ERP), para garantizar la integridad de la información y la toma de decisiones seguras. El MINFAR está dividido en varios departamentos como el de Planificación Material y Financiera en el cual se trabaja con grandes volúmenes de información y se efectúan un sin número de procesos, entre ellos el proceso de Control de la Ejecución de la Planificación Material y Financiera de los Órganos Consumidores del MINFAR, constituyendo el objetivo de este trabajo de diploma desarrollar la idea de cómo modelar un sistema que resuelva los problemas existentes dentro del mismo.

Para ello se definirá el modelo del negocio para enmarcar la situación problemática que se planteará en la Introducción, luego se desarrollará un diagrama de actividades para definir cuáles específicamente se deben informatizar, es decir, que se les pueda dar una solución en un sistema informático. Posteriormente definidas estas se realizará el levantamiento de requisitos funcionales y no funcionales y se decidirá, analizando los mismos, cómo se agruparán para definir el diagrama de Casos de Uso del Sistema y quiénes serán los actores del mismo. Siendo el análisis y el diseño los flujos de trabajo que aportan mayor información, se ocupan en el núcleo técnico de la Ingeniería de Software, se definirá la arquitectura para desarrollar el sistema, los diagramas de clases para ambos flujos, con mecanismos de diseño definidos para el MINFAR, se diseñará la Base de Datos que responderá a las necesidades del sistema con una descripción de las tablas definidas en la misma, culminándose con los diagramas de despliegue y componentes correspondientes al tema.

PALABRAS CLAVE

Planificación, MINFAR, sistema empresarial, ERP, análisis, diseño, requisitos, Distribución Recursos, Confirmación de Pago.

TABLA DE CONTENIDOS

INTRODUCCIÓN.....	1
CAPÍTULO 1. TECNOLOGÍAS ACTUALES A CONSIDERAR	4
1.1 Introducción.....	4
1.2 Sistemas Automatizados existentes vinculados al campo de acción.	4
1.3 Tendencias y tecnologías actuales.....	4
1.3.1 Planeamiento de recursos para las Empresas (ERP)	4
1.3.2 Surgimiento de Internet.....	5
1.3.3 Un servicio de Internet: La World Wide Web (WWW).....	5
1.3.4 Aplicaciones Web.	5
1.3.5 Arquitectura Cliente-Servidor.	5
1.3.6 Técnica y tecnologías del lado del cliente.	6
1.3.7 Extensible Markup Language (XML).	9
1.3.8 Lenguaje Web del lado del servidor.	9
1.3.9 Proceso de Desarrollo de Software.....	11
1.3.10 El Lenguaje Unificado de Modelado (en lo adelante UML) como lenguaje de modelado.....	12
1.3.11 Herramientas CASE.....	12
1.3.12 Herramientas de apoyo.....	13
1.4 Conclusiones.....	14
CAPÍTULO 2. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA.....	15
2.1 Introducción.....	15
2.2 Propuesta del sistema.	15
2.3 Modelado del Negocio.....	15
2.4 Actor y Trabajadores del Negocio.....	16
2.5 Diagrama de Casos de Uso del Negocio.....	17
2.6 Especificación de los Casos de Uso del Negocio.	17
2.6.1 Descripciones Textuales de los Casos de Uso del Negocio.....	17
2.6.2 Diagramas de Actividades de los Casos de Uso del Negocio.	24
2.7 Diagrama de Clases del Modelo de Objetos.....	26
2.9 Requerimientos.	27
2.9.1 Requerimientos Funcionales.....	27
2.9.2 Requerimientos No Funcionales.	29

2.10 Modelado del Sistema.....	32
2.11 Actores del Sistema.....	32
2.12 Diagrama de Casos de Uso del Sistema.....	33
2.13 Casos de Uso expandidos.....	34
2.14 Conclusiones.....	76
CAPÍTULO 3. ANÁLISIS Y DISEÑO DEL SISTEMA	77
3.1 Introducción.....	77
3.2 Modelo de Análisis.....	77
3.3 Modelo de Diseño.....	80
3.4 Arquitectura.....	81
3.5 Mecanismo de Diseño.....	82
Mecanismo de Seguridad:.....	83
Mecanismo de Acceso a Datos:.....	84
3.6 Diagramas de Interacción.....	86
3.8 Diseño de la Base de Datos.....	96
3.10 Estándares de Diseño.....	97
3.11 Estándares de Programación.....	98
3.12 Conclusiones.....	100
CAPÍTULO 4. ESBOZO DE IMPLEMENTACIÓN.....	101
4.1 Introducción.....	101
4.2 Modelo de Implementación.....	101
4.4 Conclusiones.....	103
CONCLUSIONES.....	104
RECOMENDACIONES.....	105
REFERENCIAS BIBLIOGRÁFICAS.....	106
BIBLIOGRAFÍAS.....	107
GLOSARIO DE TÉRMINOS.....	108
ANEXOS.....	110

INTRODUCCIÓN

En el entorno actual de los negocios, con el avance creciente de las tecnologías, se demandan cada vez aplicaciones más rápidas, ligeras y robustas. Estas tecnologías pueden llegar a proporcionar recursos estratégicos, no por la tecnología en sí misma que está disponible ampliamente, sino por lo fácil que es personalizarla y construir con ella sistemas propietarios para las empresas.

Muchas empresas han adoptado la tecnología como una parte importante de su estrategia y toma de decisiones, jugando un papel determinante los ERP¹ (Enterprise Resource Planning, Planeamiento de Recursos para las Empresas) que aparecen como una herramienta fundamental tanto para la integración de las diferentes áreas y sistemas de la empresa, como para la ayuda en la toma de decisiones estratégicas (LÓPEZ 2002-2007).

Una empresa que cuente con un sistema informático que no disponga de un ERP, en función de sus necesidades, puede encontrarse con muchas aplicaciones de software cerradas, que no se pueden personalizar y no se optimizan para su negocio.

Un eficiente proceso de planificación se centra en la necesidad de organizar de manera coherente lo que se quiere lograr ,implicando tomar decisiones previas a la práctica sobre qué es lo que se desea realizar ,para qué se hará y cómo se puede lograr de la mejor manera. Éste es el sentido más profundo de la acción planificadora: prever para acertar, disponer para evitar la confusión y el desorden, proyectar para resolver con desenvoltura y eficacia.

En nuestro país el Ministerio de las Fuerzas Armadas Revolucionarias (en lo adelante MINFAR), actualmente, se encuentra enmarcado dentro de un proceso de informatización, teniendo en cuenta el soporte tecnológico, desde el punto de vista de las comunicaciones, capacidad de almacenamiento y procesamientos de información que le permiten proyectar la integración de los diferentes procesos que componen el sistema de planificación que se realiza por el Aparato Central del MINFAR. Permitiendo la realización de un proceso de previsión, estructuración y asignación de recursos a los diferentes organismos vinculados al mismo, para lograr una organización y optimización de los recursos materiales y financieros con el fin de cumplir los objetivos de trabajo y las metas trazadas en un tiempo determinado.

Como parte importante de la Planificación Material y Financiera en el MINFAR se encuentra el proceso del Control de la Ejecución de la Planificación realizada por los Órganos del Aparato Central del MINFAR,este proceso de negocio incluye tareas fundamentales como la entrega de la planificación para su aprobación y registro de los recursos asignados para cada órgano consumidor, el control de las asignaciones de los cheques para la adquisición de los recursos solicitados y la recepción de las confirmaciones de pago para actualizar los productos y las mercancías de los mismos, tomando como

punto de partida los lineamientos generales definidos por la Dirección de Economía (en lo adelante DE).

Las actividades relacionadas con este proceso se realizan en la actualidad de forma manual o con herramientas ofimáticas. Este mecanismo trae aparejado una serie de problemas que van desde el retraso de la ejecución de la planificación hasta una gran complejidad en el proceso de tomar decisiones con la objetividad, veracidad y rapidez que se requiere, unido a esto podemos encontrar problemas con la gestión de la información, muchas veces convirtiéndola en ineficiente debido a la gran dispersión y la presencia de un flujo de información lento.

Como resultado de lo expresado anteriormente se puede arribar a la conclusión de que la inexistencia de un sistema informático que comprenda un enfoque integral del problema, propicia una amplia gama de complicaciones que afectan la eficiencia del proceso del Control de la Ejecución de la Planificación de los Órganos del Aparato Central del MINFAR. Todas estas razones dan lugar al siguiente **problema a resolver**: ¿Cómo realizar el diseño de un sistema informático para la gestión de la información dentro del proceso del Control de la Ejecución de la Planificación Material y Financiera realizada por los Órganos del Aparato Central del MINFAR ?

Este problema se enmarca en el **objeto de estudio**: proceso para el Control de la Ejecución de la Planificación Material y Financiera realizada por los Órganos del Aparato Central del MINFAR.

Para resolver el problema planteado se propone como **objetivo**: desarrollar el diseño de un sistema informático para el proceso del Control de la Ejecución de la Planificación Material y Financiera de los Órganos del Aparato Central del MINFAR.

De acuerdo con la propuesta anterior se proponen los siguientes **objetivos específicos**:

- Estudiar a fondo las etapas del proceso del Control de la Ejecución de la Planificación Material y Financiera realizada por los Órganos del Aparato Central del MINFAR.
- Modelar los principales procesos del negocio y garantizar sus mejoras proporcionalmente a su desarrollo.
- Realizar el análisis y diseño del sistema.
- Realizar esbozo de implementación: modelo implementación y despliegue.

El objetivo que delimita el **campo de acción** es la realización de un diseño informático para el proceso del Control de la Ejecución de la Planificación Material y Financiera realizada por los Órganos del Aparato Central del MINFAR.

Para guiar la investigación se plantea la siguiente **hipótesis**: si se cuenta con un diseño informático con una adecuada planificación y consistencia se favorecerá el desarrollo de la aplicación para el proceso del Control de la Ejecución de la Planificación Material y Financiera realizada por los Órganos del Aparato Central del MINFAR en cuanto a: un manejo fiable y óptimo de la información, posibilitando tomas de decisiones seguras.

Esta investigación estará guiada por las siguientes acciones:

- Realizar un estudio profundo sobre la etapa de Control de Ejecución de la Planificación
- Análisis del estado de la informatización dentro de las áreas vinculadas a la etapa anteriormente mencionada.
- Análisis y valoración sobre el nivel de conocimiento de los usuarios en torno a la Informática y habilidades de trabajo con sistemas informáticos con características de ERP.
- Estudio sobre sistemas existentes que trabajan el proceso de control y estado de la misma; así como la filosofía de interacción con los usuarios.
- Desarrollar el análisis y el diseño de las tareas fundamentales vinculadas al proceso del Control de la Ejecución de la Planificación.

El presente trabajo consta de introducción, cuatro capítulos, conclusiones, recomendaciones, referencias bibliográficas, glosario de términos y anexos:

En el capítulo 1, **Tecnologías actuales a considerar**, se abordan los principales aspectos de las tecnologías fundamentales que sustentan el desarrollo de sistemas informáticos asociadas a la misma.

El capítulo 2, **Descripción de la solución propuesta**, se enfoca en los aspectos fundamentales de modelamiento del negocio, captura de requisitos y análisis del sistema.

En el capítulo 3, **Construcción de la solución propuesta**, se centra en los principales aspectos sobre la arquitectura del sistema, patrones y algunos mecanismos de diseño definidos para mejorar y optimizar el desarrollo del mismo.

En el capítulo 4, **Esbozo de implementación**, se centra en la realización de los diagramas de componentes y despliegue.

CAPÍTULO 1. TECNOLOGÍAS ACTUALES A CONSIDERAR

1.1 Introducción

En el presente capítulo se ofrecen algunas características y ventajas de los ERP que constituyen la tendencia a seguir en la Planificación Material y Financiera dentro del MINFAR.

Se abordan importantes conceptos que son necesarios conocer para la creación de una aplicación de gestión Web y para entender correctamente en qué consiste la misma. También se hace referencia al estado de las técnicas, tecnologías, metodologías y software utilizadas en la confección de la propuesta de solución, abordando breves reseñas no menos necesarias como: el surgimiento de Internet, su principal servicio: la World Wide Web (WWW), así como conceptos y características de aplicaciones Web, de la arquitectura Cliente-Servidor y los Sistemas Gestores de Base de Datos.

1.2 Sistemas Automatizados existentes vinculados al campo de acción.

En particular, con relación a las características que presenta el MINFAR ,se puede decir que en la Universidad ,Cuba y el mundo no existen sistemas de gestión empresarial que contenga funciones para el control de la ejecución de la planificación que se ajusten a las características propias y específicas del MINFAR, a pesar de la existencia de sistemas de gestión empresarial avanzados .Por lo tanto no existe ningún sistema que contenga alguna similitud con el que se quiere desarrollar ,teniendo como principal motivo que la implantación de un sistema para la planeación de los recursos empresariales (ERP), es un proyecto muy complejo debido a su profundo impacto en los procesos de las empresas, significando un cambio cultural en la misma ,cambio en los procesos de negocios y en las disciplinas de trabajo .

1.3 Tendencias y tecnologías actuales.

1.3.1 Planeamiento de recursos para las Empresas (ERP)

Una planificación de recursos para las empresas constituye un sistema de gestión de la información estructurado para satisfacer la demanda de soluciones de gestión empresarial, basada en el ofrecimiento de una solución completa para las empresas permitiendo evaluar, implementar y gestionar más fácilmente su negocio. Las soluciones ERP tienen características implícitas como la modularidad, integración de la información (dato único), universalidad, estandarización e interfaces con otras aplicaciones. Constituyendo sistemas abiertos en la mayoría de los casos multiplataforma (LEÓN 2001-2007).

1.3.2 Surgimiento de Internet.

Internet es una palabra del idioma inglés que significa “en la red”. Sus orígenes se remontan a un proyecto militar estadounidense que pretendía obtener una red de comunicaciones segura que se pudiese mantener aunque fallase alguno de sus puntos de apoyo. Así nació ARPA², red de información iniciada en 1960, que se propuso conectar equipos localizados en sitios dispersos, dando paso a la gran red de redes, que no es más que la existencia de millones de computadoras interconectadas, que pueden ser clientes o servidores, independientemente de que algunos estuvieran fuera de servicio en un momento determinado dentro de un conflicto militar. Los sistemas o protocolos (TCP/IP) que permitieron finalmente tal interconexión fueron desarrollados en 1973 por el informático Vinton Cerf y el ingeniero, también estadounidense, Robert Kahn, conocidos como Protocolo de Internet (IP) y Protocolo de Control de Transmisión (TCP).

1.3.3 Un servicio de Internet: La World Wide Web (WWW).

El Web, WWW o World Wide Web es un sistema mundial de información sencillo y poderoso, constituido por documentos (llamados páginas Web) y vínculos a otros documentos o páginas. Para navegar en la Web se usan programas llamados “navegadores” o “browsers”. A veces el término Web se usa para referirse al Internet en general ya que la Web es una de las aplicaciones más populares; pero también es sólo una, de las muchas aplicaciones de Internet.

1.3.4 Aplicaciones Web.

Una aplicación Web es un sistema informático que los usuarios utilizan para acceder a un servicio Web mediante Internet³ o a través de una Intranet⁴. Estas aplicaciones han tomado gran auge y son populares debido a la practicidad del navegador Web como cliente ligero, su facilidad para actualizar y mantener aplicaciones Web sin distribuir e instalar software en miles de clientes. Aplicaciones como los webmails, wikis, weblogs, tiendas en línea y la wikipedia son ejemplos bien conocidos de aplicaciones Web. Existen diferentes formas de estructurar estas aplicaciones, comúnmente como una aplicación de tres-capas. En su sentido más común el navegador Web constituye la primera capa, un motor usando alguna tecnología Web dinámica (ejemplo: PHP, Java Servlets o ASP) es la segunda capa y una base de datos como última capa.

1.3.5 Arquitectura Cliente-Servidor.

La arquitectura cliente servidor se basa en la distribución de tareas, el cliente (un usuario de PC) solicita un servicio (como por ejemplo imprimir) que un servidor le proporciona (un procesador

conectado a la LAN⁵). Para establecer un canal de comunicación entre dos programas que se ejecutan en dos computadoras distintas o incluso en una misma computadora, un programa debe iniciar la conexión y el otro aceptarla. En esto se basa el modelo Cliente-Servidor, cuando un cliente o computadora remota necesita enviar o descargar información hacia o desde el servidor, éste establece una conexión por donde circula la información y luego la cierra. El cliente interactúa directamente con el usuario, procesa sus solicitudes y muestra los resultados.

Así por ejemplo la World Wide Web utiliza un modelo cliente-servidor, los visualizadores o navegadores (clientes) tienen la función de manipular las solicitudes de documentos hechas por los usuarios. Ellos determinan con qué computadora se realizará la conexión, descargan los documentos solicitados y se los muestran al usuario en su pantalla. Los servidores Web son los responsables de la otra parte de la conexión, esperan una solicitud proveniente de un usuario y una vez hecha ésta, transmiten el documento al visualizador.

1.3.6 Técnica y tecnologías del lado del cliente.

Un cliente Web (navegador o browser) es una aplicación de software que está formada por el código HTML⁶ que forma la página Web, con opción a código ejecutable mediante los lenguajes de scripting de los navegadores (Java Script, VBScript, etc.) o mediante pequeños programas en Java (applets). Las tecnologías del lado del cliente evitan la recarga de trabajo de los servidores delegando la ejecución de la planificación al cliente. Entre los navegadores más populares están, el Netscape, Internet Explorer, Mozilla. Los lenguajes más comunes que se ejecutan en el cliente son: VBScript y Java Script.

1.3.6.1 Lenguaje Web del lado del cliente: Java Script

En las aplicaciones Web, el cliente es el encargado de ejecutarlas en la máquina del usuario. Son las aplicaciones tipo Java o Java Script: el servidor proporciona el código de las aplicaciones al cliente y éste, mediante el navegador, las ejecuta. Es necesario, por tanto, que el cliente disponga de un navegador con capacidad para ejecutar aplicaciones (también llamadas scripts). Normalmente, los navegadores permiten ejecutar aplicaciones escritas en lenguaje Java o Java Script, aunque pueden añadirse más lenguajes mediante el uso de plugins. Java Script es un lenguaje que no requiere de compilación, se considera un lenguaje de programación interpretado, utilizado principalmente en páginas Web. Su sintaxis es semejante al lenguaje Java y C. No es un lenguaje puramente orientado a objetos porque no dispone de herencia, se basa en prototipos donde las nuevas clases se generan clonando a las clases base (prototipos) y extendiendo su funcionalidad (TORRE 2006).

Java Script:

Es un lenguaje de programación utilizado para crear pequeños programas encargados de realizar acciones dentro del ámbito de una página Web. Se trata de un lenguaje de programación del lado del cliente, porque es el navegador el que soporta la carga de procesamiento. Gracias a su compatibilidad con la mayoría de los navegadores modernos, es el lenguaje de programación del lado del cliente más utilizado.

¿Por qué Java Script?

La importancia de Java Script radica en la posibilidad de agregar respuestas inmediatas en las páginas HTML, antes de la utilización de este lenguaje la interactividad se limitaba a la activación de vínculos de hipertextos para acceder a otras páginas. Con Java Script es posible detectar y modificar eventos tales como oprimir un botón o un vínculo, cargar páginas, intercambiar imágenes cuando se mueve un cursor sobre ellas, o validar formularios antes de enviarlos. Entre las acciones típicas que se pueden realizar en Java Script tenemos dos vertientes. Por un lado los efectos especiales sobre páginas Web, para crear contenidos dinámicos y elementos de la página que tengan movimiento, cambien de color o cualquier otro dinamismo. Por el otro, Java Script nos permite ejecutar instrucciones como respuesta a las acciones del usuario, con lo que podemos crear páginas interactivas con programas como calculadoras, agendas, o tablas de cálculo (TORRE 2006).

1.3.6.2 AJAX: Una técnica de desarrollo Web.

AJAX es una técnica de desarrollo Web para crear aplicaciones interactivas, es un acrónimo de Asynchronous Java Script And XML⁷ (Java Script y XML asíncronos).

Dichas aplicaciones se ejecutan en el lado del Cliente y mantiene comunicación asíncrona con el servidor, en segundo plano, sin que el usuario se percate de dichas comunicaciones a nivel de tiempos de respuestas, se considera un conjunto de tecnologías que se desarrollan por sí mismas y se unen en poderosas nuevas formas, incorporando: presentación basada en estándares usando XHTML⁸ y CSS⁹, exhibición e interacción dinámicas usando el Document Object Model, intercambio y manipulación de datos usando XML y XSLT¹⁰, recuperación de datos asíncrona usando XMLHttpRequest¹¹ y Java Script poniendo todo junto.

AJAX combina de forma simultánea la técnica de página única convirtiéndose en una alternativa sólida para un nuevo desarrollo de interfaz entre el Usuario y los Servidores de Información que existen en Internet. No necesitan refrescar la página completa para actualizar información solicitada, pueden simplemente actualizar parte de la página en cualquier momento, dándole al usuario una respuesta instantánea a sus consultas por la página Web. Se puede decir que al aplicar este conjunto de

tecnologías se establece un motor AJAX entre el usuario y el servidor, tomando como base que la aplicación de AJAX permite que la interacción del usuario con la aplicación suceda asincrónicamente (independientemente de la comunicación con el servidor). Así el usuario nunca estará mirando una ventana en blanco del navegador y un icono de reloj de arena esperando a que el servidor haga algo. (Ver Fig.1)

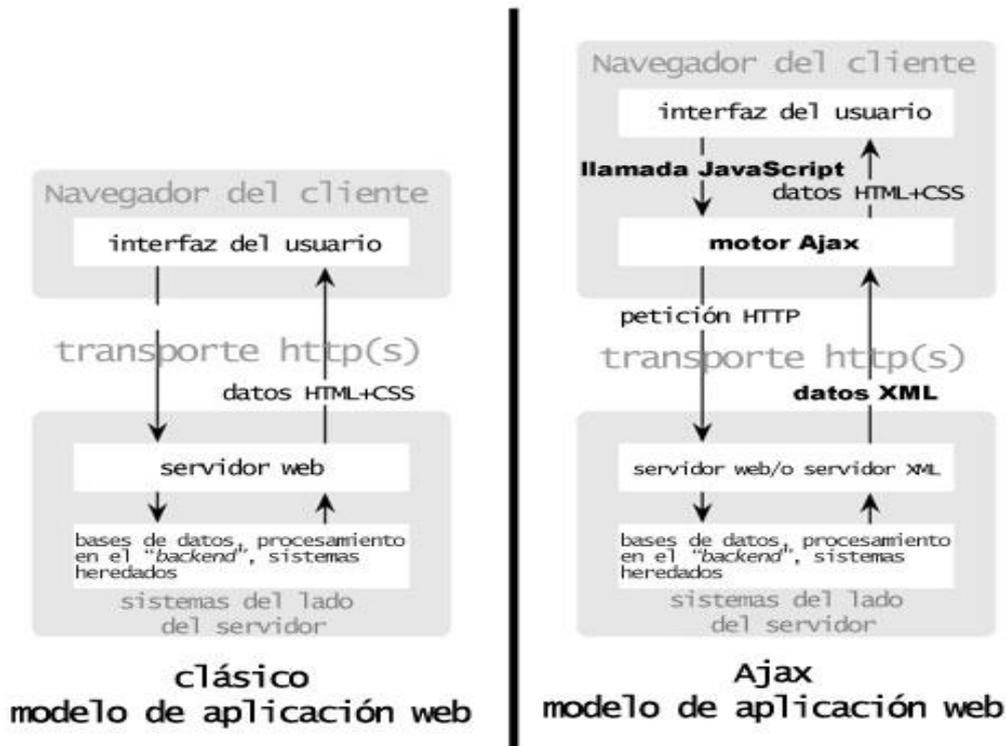


Figura.1: El modelo tradicional para las aplicaciones Web (izq.) comparado con el modelo de AJAX (der).

1.3.6.3 Hojas de Estilo en Cascada.

Las hojas de estilo en cascada (Cascading Style Sheets, CSS), permiten desarrollar la creatividad en el diseño con una intuición sin precedentes. La idea que se encuentra detrás del desarrollo de CSS es separar la estructura de un documento de su presentación. Las CSS constituyen un mecanismo para asociar estilos de composición a documentos estructurados, del tipo HTML o XML. Aplicables a cualquier navegador, admiten un mayor control sobre los distintos elementos de una página, permitiendo definir el estilo de las fuentes, el color, el espaciado del texto, la posición del contenido, e incluso variaciones en el sonido en los elementos auditivos. Estos estilos pueden definirse para luego ser aplicados al código de cualquier documento.

1.3.7 Extensible Markup Language (XML).

XML es una tecnología sencilla que tiene a su alrededor otras que la complementan y la hacen mucho más grande y con unas posibilidades mucho mayores. Tiene un papel muy importante en la actualidad ya que permite la compatibilidad entre sistemas para compartir la información de una manera segura, fiable y fácil. La tecnología XML trata de ofrecer una solución al problema de expresar información estructurada de la manera más abstracta y reutilizable posible.

1.3.8 Lenguaje Web del lado del servidor.

Un lenguaje del lado del servidor es aquel que se ejecuta en el servidor Web, justo antes de que se envíe la página a través de Internet al cliente. Las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones en red, y otras tareas para crear la página final que verá el cliente. El cliente solamente recibe una página con el código HTML resultante de la ejecución de la PHP. Como la página resultante contiene únicamente código HTML, es compatible con todos los navegadores.

Las aplicaciones de servidor suelen ser la opción por la que se opta en la mayoría de las ocasiones para realizar aplicaciones Web. La razón es que, al ejecutarse ésta en el servidor y no en la máquina del cliente, éste no necesita ninguna capacidad adicional, como sí ocurre en el caso de querer ejecutar aplicaciones Java script o Java. Así pues, cualquier cliente dotado de un navegador Web básico puede utilizar este tipo de aplicaciones, las mismas están formadas por un programa o script que es ejecutado por el servidor Web, y cuya salida se envía al navegador del cliente. Los principales beneficios de la aplicación de la tecnología de servidores de aplicación son la centralización y la disminución de la complejidad en el desarrollo de aplicaciones (TORRE 2006).

1.3.8.1 Script del lado del servidor.

Los lenguajes de scripts que se ejecutan en el servidor Web son interpretados por alguna aplicación alojada en dicho servidor y que se envían al cliente en un formato comprensible para él, son independientes del cliente por lo que son mucho más flexibles en relación al cambio de un navegador a otro, o con respecto a las versiones del mismo. El cliente solo verá el código HTML terminado.

PHP:

PHP (Acrónimo de Hipertext Preprocesor) es un lenguaje de script interpretado en el lado del servidor gratuito e independiente de plataforma, rápido, con una gran librería de funciones y mucha

documentación utilizado para la generación de páginas Web dinámicas, similar al ASP de Microsoft o el JSP de Sun, embebido en páginas HTML y ejecutado en el servidor (HENST).

¿Por qué PHP?

La meta del lenguaje es permitir rápidamente a los desarrolladores la generación dinámica de páginas. Es un lenguaje de alto nivel empleado para el desarrollo de aplicaciones Web permitiendo crear prácticamente cualquier funcionalidad imaginable. La principal característica de PHP es que es un lenguaje que se ejecuta en el servidor, y por tanto, es independiente del navegador que use el visitante, además de que permite manejar bases de datos desde una interfaz web. Analizando esto el más adecuado es el PHP por ser libre, multiplataforma, su flexibilidad de comunicación con los principales gestores de bases de datos y por su potencialidad en funcionalidades y rapidez (ALFARO 2007).

1.3.8.2 Sistemas de Gestión de Base de Datos (en lo adelante SGBD).

Un Sistema Gestor de Bases de Datos (en lo adelante SGBD), es un conjunto de programas que permiten introducir y almacenar datos, ordenarlos y manipularlos, en fin realizar diversas operaciones con la información de una base de datos. El principal propósito de los SGBD es trabajar de forma sencilla, ordenada y clara con un conjunto de datos que posteriormente nos servirá de mucho apoyo para poder tener una factible manipulación de los mismos. Poseen grandes ventajas como lo son, el control de la redundancia, restricción de los accesos no autorizados y además ofrecen recursos para definir y garantizar el cumplimiento de las restricciones de integridad. Dentro de los tipos fundamentales de SGBD tenemos los libres (PostgreSQL¹², MySQL, etc.), los gratuitos (Microsoft SQL Server Compact Edición, entre otros) y los comerciales (Oracle, WindowBase, etc.).

1.3.8.3 ¿Por qué PostgreSQL?

PostgreSQL es un Sistema de Gestión de Bases de Datos Objeto- Relacionales de dominio público, está ampliamente considerado como el sistema de bases de datos de código abierto más avanzado del mundo. PostgreSQL proporciona un gran número de características, que lo hace mucho más conveniente para trabajar con el, por ejemplo posee modelos de negocios más rentables con instalaciones a gran escala, no existe la posibilidad de ser auditado para verificar cumplimiento de licencia en ningún momento, es muy flexible para hacer investigación y desarrollo sin necesidad de incurrir en costos adicionales de licenciamiento. Además este SGBD posee estabilidad y confiabilidad legendarias, incluye el código fuente disponible para todos sin costo, PostgreSQL usa una estrategia de almacenamiento de filas, para conseguir una mejor respuesta en ambientes de grandes volúmenes,

posee además herramientas gráficas de alta calidad para administrar las bases de datos. Soporta distintos tipos de datos: además del soporte para los tipos base, también soporta datos de tipo fecha, monetarios, elementos gráficos, datos sobre redes, cadenas de bits, índices, reglas y vistas (TIENDALINUX.COM.).

1.3.9 Proceso de Desarrollo de Software.

Un Proceso de Desarrollo de Software es la definición del conjunto de actividades que guían los esfuerzos de las personas implicadas en el proyecto, a modo de plantilla que explica los pasos necesarios para terminar el proyecto. Definiendo: "Quién", "Qué", "Cuándo" y "Cómo" hay que realizar las cosas necesarias para alcanzar un determinado producto de software. El proceso de desarrollo de software requiere por un lado un conjunto de conceptos, una metodología y un lenguaje propio. Este es el encargado de guiar el desarrollo por cada uno de los pasos que garanticen un producto final de calidad. Debe brindar la posibilidad al equipo de desarrollo de documentar lo aspectos esenciales para en un futuro dar soporte al software.

1.3.9.1 Metodologías de desarrollo de software.

Un proceso de software detallado y completo suele denominarse "Metodología". Sin embargo existen numerosas propuestas metodológicas que inciden en distintas dimensiones del proceso de desarrollo. Vale destacar que todo desarrollo de software es riesgoso y difícil de controlar, por lo que es necesario aplicar una metodología que permita obtener como resultado un producto de calidad. Las metodologías guían el proceso de desarrollo y la clave del éxito de un proyecto de software es la elección correcta de la metodología, que puede conducir al programador a desarrollar un buen sistema de software, logrando el crecimiento de la calidad y la productividad del producto (MENDOZA 2004).

La elección de la metodología adecuada es más importante que utilizar las mejores y más potentes herramientas.

Metodología para el desarrollo unificado, Rational Unified Process (en lo adelante RUP):

RUP¹³ es un proceso de desarrollo de software donde durante el paso por las diferentes etapas de desarrollo se van a "transformar los requerimientos de los usuarios en un sistema software". La metodología RUP, llamada así por sus siglas en inglés Rational Unified Process, divide en 4 fases el desarrollo del software.

- Fase de Conceptualización: El Objetivo en esta etapa es determinar la visión del proyecto.
- Fase de Elaboración: En esta etapa el objetivo es determinar la arquitectura óptima.
- Fase de Construcción: En esta etapa el objetivo es llegar a obtener la capacidad operacional inicial.

- Fase de Transición: El objetivo es llegar a obtener el release del proyecto.

RUP es un proceso de desarrollo de software donde se asignan tareas y responsabilidades cuyos objetivos son asegurar la producción de software de calidad dentro de plazos y presupuestos predecibles. El ciclo de vida de RUP se caracteriza por ser: Dirigido por los casos de uso, Centrado en la arquitectura, Iterativo e incremental (ITERA.SA).

1.3.10 El Lenguaje Unificado de Modelado (en lo adelante UML) como lenguaje de modelado.

El Lenguaje Unificado de Modelado prescribe un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos, y describe la semántica esencial de lo que estos diagramas y símbolos significan. UML¹⁴ se puede usar para modelar distintos tipos de sistemas: sistemas de software, sistemas de hardware, y organizaciones del mundo real (GRACIA 2005). UML ofrece nueve diagramas en los cuales modelar sistemas:

- Diagramas de Casos de Uso para modelar los procesos.
- Diagramas de Secuencia para modelar el paso de mensajes entre objetos.
- Diagramas de Colaboración para modelar interacciones entre objetos.
- Diagramas de Estado para modelar el comportamiento de los objetos en el sistema.
- Diagramas de Actividad para modelar el comportamiento de los Casos de Uso, objetos u operaciones.
- Diagramas de Clases para modelar la estructura estática de las clases en el sistema.
- Diagramas de Objetos para modelar la estructura estática de los objetos en el sistema.
- Diagramas de Componentes para modelar componentes.
- Diagramas de Implementación para modelar la distribución del sistema.

1.3.11 Herramientas CASE.

Las Herramientas CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Ordenador) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software haciendo que sea mucho más rápido y menos costoso. Estas herramientas nos pueden ayudar en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores entre otras. Es conocida como un conjunto de métodos y técnicas a través de las cuales se hacen

útiles a las personas comprender las capacidades de las computadoras, por medio de programas, de procedimientos y su respectiva documentación (PEREYRA. 2008).

En nuestro sistema a desarrollar utilizaremos como herramienta a Visual Paradigm.

1.3.11.1 ¿Por que Visual Paradigm?

Visual Paradigm es una herramienta CASE visual que utiliza UML como lenguaje de modelado. Se integra con las diversas herramientas Java, tales como Eclipse/IBM Web Sphere , Oracle JDeveloper entre otras. Ayuda a construir aplicaciones de forma rápida, mejor y de forma económica. Esta herramienta tiene unas características gráficas muy cómodas que facilitan la realización de los diagramas de modelado que sigue el estándar de UML, que son: Diagramas de clases, Casos de Uso, Comunicación, Secuencia, Estado, Actividad, Componentes, etc. Permite relacionar elementos presentes en un enunciado con los diagramas UML correspondientes, aunque no contempla el tratamiento directo en el texto de la relación entre los componentes del enunciado (WAN).

Ventajas de utilizar Visual Paradigm:

- Navegación intuitiva entre el modelo visual y el código.
- Sincronización entre el código fuente y el modelo en tiempo real.
- Entorno visual de modelado superior.
- Soporte para toda la notación UML.
- Sofisticados y automáticos diagramas de capas.
- Análisis de textos.
- Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.
- Disponibilidad de integrarse en los principales.
- Disponibilidad en múltiples plataformas.
- Diseño centrado en casos de uso y enfocado al negocio que generan un software de mayor calidad.
- Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- Capacidades de ingeniería directa (versión profesional) e inversa.

1.3.12 Herramientas de apoyo.

Navegador:

El navegador sobre el cual correrá la aplicación es el Mozilla Firefox este permite abrir por defecto las nuevas páginas Web en pestañas, cada una de esas pestañas tiene su propio botón de cerrado.

Firefox incorpora bloqueo de ventanas emergentes, marcadores dinámicos, soporte para estándares abiertos y un mecanismo para añadir funcionalidades mediante extensiones. Posee un corrector ortográfico para evitar que se cometan errores de ortografía en las entradas de información que se hagan. Tiene una sugerencia de búsqueda que se va desplazando a medida que se va introduciendo el texto que se desea buscar. Mantiene a salvo a la aplicación de programas espías e impostores usando el poder de una comunidad de desarrollo que le da soporte. Estas y muchas otras posibilidades brinda este navegador del cual se hará uso, aunque el sistema también correrá sobre el Internet Explorer.

Macromedia Dreamweaver 8:

Dreamweaver 8 es la herramienta de desarrollo Web líder del mercado y permite a sus usuarios diseñar, desarrollar y mantener de forma eficaz sitios y aplicaciones Web basadas en normas, con Dreamweaver 8, los desarrolladores Web lo abarcan todo, desde la creación y mantenimiento de sitios Web básicos hasta aplicaciones avanzadas compatibles con las mejores prácticas y las tecnologías más recientes. Brinda múltiples herramientas visuales de diseño y un entorno de codificación adaptable a lenguajes de programación Web (PHP), trabaja con hojas de estilos CSS, permite la comparación de archivos para determinar que ha cambiado, permite el trabajo directo del lado del servidor, etc.

1.4 Conclusiones.

En este capítulo se detallaron las condiciones y problemas que rodean el objeto de estudio; y a través de los conceptos y definiciones planteadas, se determinaron las condiciones específicas que envuelven al mismo. En base a esto se obtuvieron los objetivos generales y específicos para este trabajo, se planteó una solución al problema, se analizaron las características de diferentes herramientas, para la creación de un software o aplicación así como algunas metodologías. Para el desarrollo de la aplicación Web se tuvo en cuenta las restricciones que impone el MINFAR como requisito en la elaboración de la misma.

Después de este análisis y la fundamentación realizada el lenguaje de programación que se utilizará es PHP y Java Script para el control de las diferentes funciones a realizar en el cliente, como gestor de base de datos se estableció el PostgreSQL.

Se empleará la metodología RUP, que permite dirigir el proceso por casos de uso y tener un mejor control de la calidad en cada etapa del proceso, como lenguaje de modelado UML y la herramienta que se usará como entorno de creación de diagramas para UML el Visual Paradigm.

CAPÍTULO 2. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA.

2.1 Introducción.

En el presente capítulo, se dará una propuesta de solución de lo que funcionalmente debe realizar el sistema para erradicar los problemas existentes anteriormente mencionados. Se transitará a lo largo de la fase de inicio por cada uno de los flujos de trabajo desde el negocio hasta el sistema. Todo esto se desarrollará por medio de un conjunto de artefactos y modelos resultantes de la aplicación de la metodología de desarrollo de software. Estos modelos deben cumplir una serie de propiedades, entre ellas la de ser coherentes y relacionados. Uno de los modelos útiles previo al desarrollo de un software es el modelo del negocio.

2.2 Propuesta del sistema.

En el proceso del Control de la Ejecución de la Planificación Material y Financiera realizada por los Órganos del Aparato Central del MINFAR, el V_Departamento juega un papel determinante, es el responsable de entregar la aprobación de la planificación realizada por el Viceministro I al Departamento de Economía para comenzar con el proceso del Control de la Ejecución de la Planificación Material y Financiera. El Jefe del Departamento de Economía realiza la distribución de los recursos asignados para cada órgano consumidor y le orienta a los operadores oficiales que comienzan a registrar y controlar las operaciones que se realizan con las aprobaciones, mercancías, contratos y confirmaciones de pago, estas últimas ocurren después que el órgano consumidor recibe el cheque de acceso a los recursos. Las planificaciones, aprobaciones de la planificación, deben poder ser insertadas, modificadas, eliminadas y mostradas al igual que los contratos, mercancías, confirmaciones de pago y la distribución de recursos; cuando se registra una confirmación de pago se liquidan recursos en mercancías y contratos. Se necesitan obtener reportes de las confirmaciones de pago, distribución de recursos y planificación, dando la posibilidad de imprimir.

2.3 Modelado del Negocio.

El modelado del negocio es una técnica para comprender y definir los procesos del negocio, roles y responsabilidades de la organización en los modelos de casos de uso del negocio y de objetos.

Los objetivos fundamentales que se persiguen al realizarse el modelado del negocio, son:

- Comprender la estructura y la dinámica de la organización en la cual se va a implantar el sistema.
- Comprender los problemas actuales de la organización e identificar las mejoras potenciales.

- Asegurar que los usuarios finales y desarrolladores tengan un entendimiento común de la organización.

2.4 Actor y Trabajadores del Negocio.

Un actor puede representar a una persona física, otro sistema, un dispositivo, siendo siempre un tercero fuera del sistema que colabora con él. La definición de actores, sirve para definir el contexto externo del sistema, esto es, delimitar los elementos que se encuentran fuera y dentro del mismo, y un actor juega un rol para cada caso de uso en el que colabora. Todo lo que interactúa con el ambiente del negocio debe modelarse como actor.

Actor del negocio	Justificación
Órganos Consumidores	Entrega la propuesta de planificación para que sea aprobada y solicita los recursos materiales y financieros planificados en su propuesta, proporciona además la confirmación de pago para que se actualicen las operaciones sobre los recursos que le han sido asignados.

Tabla 1. Descripción de los actores del negocio.

El trabajador, por otro lado, define el comportamiento y responsabilidades (rol) de un individuo, grupo de individuos, sistema automatizado o máquina, que trabajan en conjunto como un equipo. Para cada trabajador se especifican las responsabilidades y habilidades requeridas, estos realizan las actividades y son propietarios de elementos.

Trabajadores del negocio	Justificación
V Departamento	Este trabajador es quien recibe la propuesta para la planificación del presupuesto del año entrante por parte del Órgano Consumidor y se encarga de tramitar su aprobación.
Jefe del Departamento de Economía.	Este trabajador recibe la planificación aprobada, realiza la distribución de los recursos del Órgano Consumidor y la imprime, firma el cheque de acceso a los recursos materiales y financieros.
Operador Oficial	Este trabajador es quien registra la planificación aprobada, las mercancías y los contratos, recibe las confirmaciones de pagos de los Órganos Consumidores y actualiza los contratos y mercancías de los mismos.

Tabla 2. Descripción de los trabajadores del negocio.

2.5 Diagrama de Casos de Uso del Negocio.

Para lograr los propósitos del negocio y obtener una visión general de los diferentes procesos del mismo se construye el diagrama de casos de uso del negocio, donde cada proceso se representa como un caso de uso y solo aparecerán los actores del negocio correspondientes a los roles externos al sistema. Este diagrama permite mostrar los límites y el entorno de la organización (Ver Figura 2).

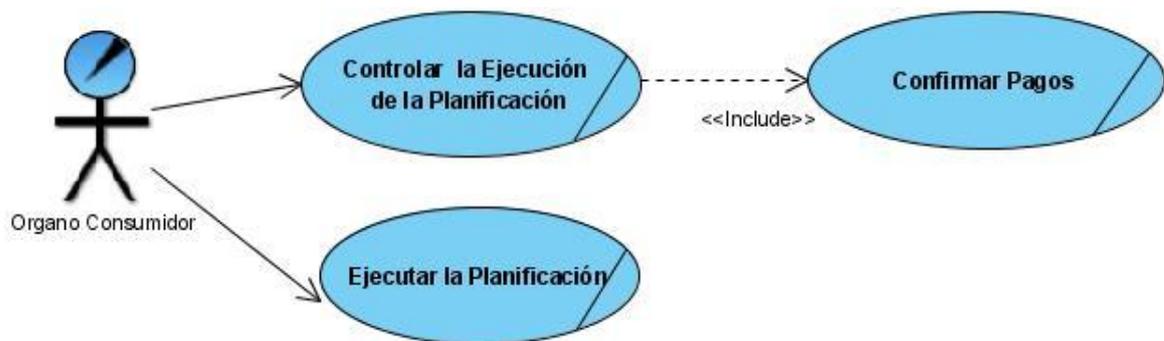


Figura 2. Diagrama de Casos de Uso del Negocio.

2.6 Especificación de los Casos de Uso del Negocio.

Descripción textual:

La descripción textual de un caso de uso de negocio se formaliza en un documento generalmente llamado "Especificación del Caso de Uso de negocio". Para realizar la misma, inicialmente se rellena una plantilla de descripción, y después, a partir de la información reflejada en dicha plantilla, se construye un conjunto de diagramas (*diagramas de actividades*) que describen completamente el caso de uso del negocio.

2.6.1 Descripciones Textuales de los Casos de Uso del Negocio.

Caso de uso:	Ejecutar la Planificación
Actores del negocio	Órgano Consumidor

DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

Trabajadores del negocio	V Departamento, Viceministro I, Jefe del departamento de Economía, Operador Oficial.
Resumen	El caso de uso comienza cuando los Órganos Consumidores entregan la propuesta para la aprobación de la Planificación Material y Financiera al V Departamento, este se encarga de tramitar la propuesta de planificación, presentarla al Viceministro I para su aprobación, cuando queda aprobada el V Departamento la recibe y se la entrega al Departamento de Economía .El Jefe del Departamento de Economía le envía la planificación aprobada al Operador Oficial para que registre la planificación, las mercancías y los contratos, confecciona reportes sobre información de planificación y los envía al Jefe del Departamento de Economía quien realiza la distribución de recursos y la imprime, se archiva una copia y se le entrega la distribución de recursos a los Órganos Consumidores, el caso de uso finaliza cuando estos la reciben .
Casos de usos asociados	ninguno
Precondiciones	La Propuesta de Planificación debe quedar aprobada para que se ejecute la planificación.
Flujo Normal de Eventos	
Flujo normal	
Acción del actor	Respuesta del proceso de negocio
1. El Órgano Consumidor entrega al V Departamento la propuesta para que se le realice la planificación del presupuesto del año entrante.	2. El V Departamento recibe el documento de la Propuesta de Planificación y se encarga de tramitar la aprobación de la planificación dando inicio al subproceso Tramitar Propuesta de Planificación (véase Sección 1). Si la Propuesta de Planificación es aprobada el V Departamento entrega la planificación aprobada al Jefe del Departamento de Economía, sino véase Flujo Alterno 2 .
	3. El Jefe del Departamento de Economía recibe planificación aprobada.
	4. El Jefe del Departamento de Economía envía la planificación aprobada al Operador Oficial.
	5. El Operador Oficial recibe la planificación aprobada.

DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

	6. El Operador Oficial registra la planificación aprobada.
	7. El Operador Oficial registra las mercancías y los contratos de la planificación aprobada.
	8. El Operador Oficial registra el Órgano Consumidor de la planificación aprobada.
	9. El Operador Oficial confecciona reportes con la información de la planificación registrada.
	10. El Operador Oficial le avisa al Jefe del Departamento de Economía que terminó la confección de los reportes.
	11. El Jefe del Departamento de Economía recibe el aviso de terminación de reportes de planificación.
	12. El Jefe del Departamento de Economía revisa los reportes de planificación.
	13. El Jefe del Departamento de Economía realiza la distribución material y financiera de los recursos.
	14. El Jefe del Departamento de Economía imprime la distribución material y financiera de los recursos.
	15. El Jefe del Departamento de Economía archiva una copia de la distribución material y financiera de los recursos.
	16. El Jefe del Departamento de Economía entrega la distribución material y financiera al Órgano Consumidor.
17. El Órgano Consumidor recibe la distribución material y financiera de recursos.	
Secciones.	
Sección 1	
Acción del actor	Respuesta del proceso de negocio
	1.El V Departamento entrega la propuesta de planificación al Viceministro I.
	2. El Viceministro I recibe la propuesta de planificación.
	3. El Viceministro I revisa y aprueba la propuesta de planificación y

DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

	la envía al V Departamento, si la propuesta de planificación es rechazada véase Flujo Alterno 1 .
	4. El V Departamento recibe la Planificación Aprobada (Regresa al paso 3 del flujo normal).
Flujos Alternos	
Flujo Alterno 1	
Acción del actor	Respuesta del proceso de negocio
	1. El Viceministro I informa la desaprobación de la Propuesta de Planificación al V Departamento terminando el proceso.
Flujo Alterno 2	
	1. El V Departamento informa al Órgano Consumidor la desaprobación de la Propuesta de Planificación terminando el proceso.
Pos-Condiciones	La planificación es ejecutada, se realiza la distribución material y financiera de la misma donde el Jefe del Departamento de Economía la envía al Órgano Consumidor.
Mejoras	La informatización de este caso de uso favorecerá a la rapidez y eficiencia en el proceso Control de la Ejecución de la Planificación Material y Financiera.
Prioridad	Crítico.

Tabla. 3 Descripción Textual Caso de Uso “Ejecutar la Planificación”.

Caso de uso:	Controlar la Ejecución de la Planificación
Actores del negocio	Órgano Consumidor
Trabajadores del negocio	Jefe del Departamento de Economía, Operador Oficial.
Resumen	El caso de uso inicia cuando el Órgano Consumidor solicita la necesidad de recursos materiales y financieros al Operador Oficial, este le solicita el código del Órgano Consumidor para verificar si se encuentra registrado y buscar si el contrato de la misma es válido y las mercancías disponibles, si estos resultados son positivos se lo comunica al Jefe del Departamento de Economía para que firme el cheque de acceso a recursos materiales y financieros y enviárselo al

DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

	Órgano Consumidor terminando el caso de uso.
Casos de usos asociados	Confirmar Pago (relación de inclusión).
Precondiciones	El Órgano Consumidor debe estar registrado y el contrato debe ser válido.
Flujo Normal de Eventos	
Flujo normal	
Acción del actor	Respuesta del proceso de negocio
1. El actor solicita recursos materiales y financieros al Operador Oficial.	2. El Operador Oficial le solicita el código del Órgano Consumidor.
3. El actor proporciona el código al Operador Oficial.	
	4. El Operador Oficial busca el Órgano Consumidor, si no es encontrado véase Flujo Alterno 1 .
	5. El Operador Oficial consulta la validez del contrato de recursos materiales y financieros, si no es válido véase Flujo Alterno 2 .
	6. El Operador Oficial consulta la disponibilidad de las mercancías, sino están disponibles véase Flujo Alterno 3 .
	7. El Operador Oficial informa el resultado de la búsqueda al Jefe del Departamento de Economía.
	8. El Jefe del Departamento de Economía firma el cheque de acceso a recursos materiales y financieros.
	9. El Jefe del Departamento de Economía entrega el cheque de acceso de recursos materiales y financieros al Órgano Consumidor.
10. El actor recibe el cheque de acceso recursos materiales y financieros.	
Flujos Alternos	
Flujo Alterno 1	
Acción del actor	Respuesta del proceso de negocio

DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

	1. El Operador Oficial informa al Órgano Consumidor que no se encuentra registrado y se termina el proceso de negocio.
Flujo Alternativo 2	
	1. El Operador Oficial informa al Órgano Consumidor que el contrato de recursos materiales y financieros no es válido y se termina el proceso de negocio.
Flujo Alternativo 3	
	1. El Operador Oficial informa al Órgano Consumidor que las mercancías no están disponibles y se termina el proceso de negocio.
Pos condiciones:	Al terminar el proceso de negocio el Órgano Consumidor recibe el cheque de acceso a recursos materiales y financieros.

Tabla. 4 Descripción Textual Caso de Uso “Controlar la Ejecución de la Planificación”.

Caso de uso:	Confirmar Pago
Actores del negocio	Órgano Consumidor
Trabajadores del negocio	Operador Oficial, Jefe del Departamento de Economía.
Resumen	El caso de uso inicia cuando el Órgano Consumidor entrega la confirmación de pago de las compras que ha realizado al Operador Oficial, estos la reciben. El Operador Oficial busca al Órgano Consumidor al que pertenece la confirmación de pago, registra la misma y realiza la liquidación de recursos, actualizando los contratos y las mercancías. El Operador Oficial entrega la confirmación de pago al Jefe del Departamento de Economía, quien lo recibe y archiva terminando el caso de uso.
Casos de usos asociados	Ninguno
Precondiciones	El Órgano Consumidor debe entregar la confirmación de pago.
Flujo Normal de Eventos	
Flujo normal	
Acción del actor	Respuesta del proceso de negocio

DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

1. El actor entrega la confirmación de pagos al Operador Oficial.	2. El Operador Oficial recibe la confirmación de pagos.
	3. El Operador Oficial busca al Órgano Consumidor al que pertenece la confirmación de pagos.
	4. El Operador Oficial registra la confirmación de pagos.
	5. El Operador Oficial realiza la liquidación de recursos con la información de la confirmación de pagos.
	6. El Operador Oficial actualiza los contratos y las mercancías del Órgano Consumidor correspondiente.
	7. El Operador Oficial entrega la confirmación de pagos al Jefe del Departamento de Economía.
	8. El Jefe del Departamento de Economía recibe la confirmación de pagos.
	9. El Jefe del Departamento de Economía archiva la confirmación de pagos.
Poscondiciones:	Al terminar el proceso quedan actualizados los contratos y las mercancías del Órgano Consumidor que entrega la confirmación de pagos.

Tabla. 5 Descripción Textual Caso de Uso Incluido “Confirmar Pago”.

2.6.2 Diagramas de Actividades de los Casos de Uso del Negocio.

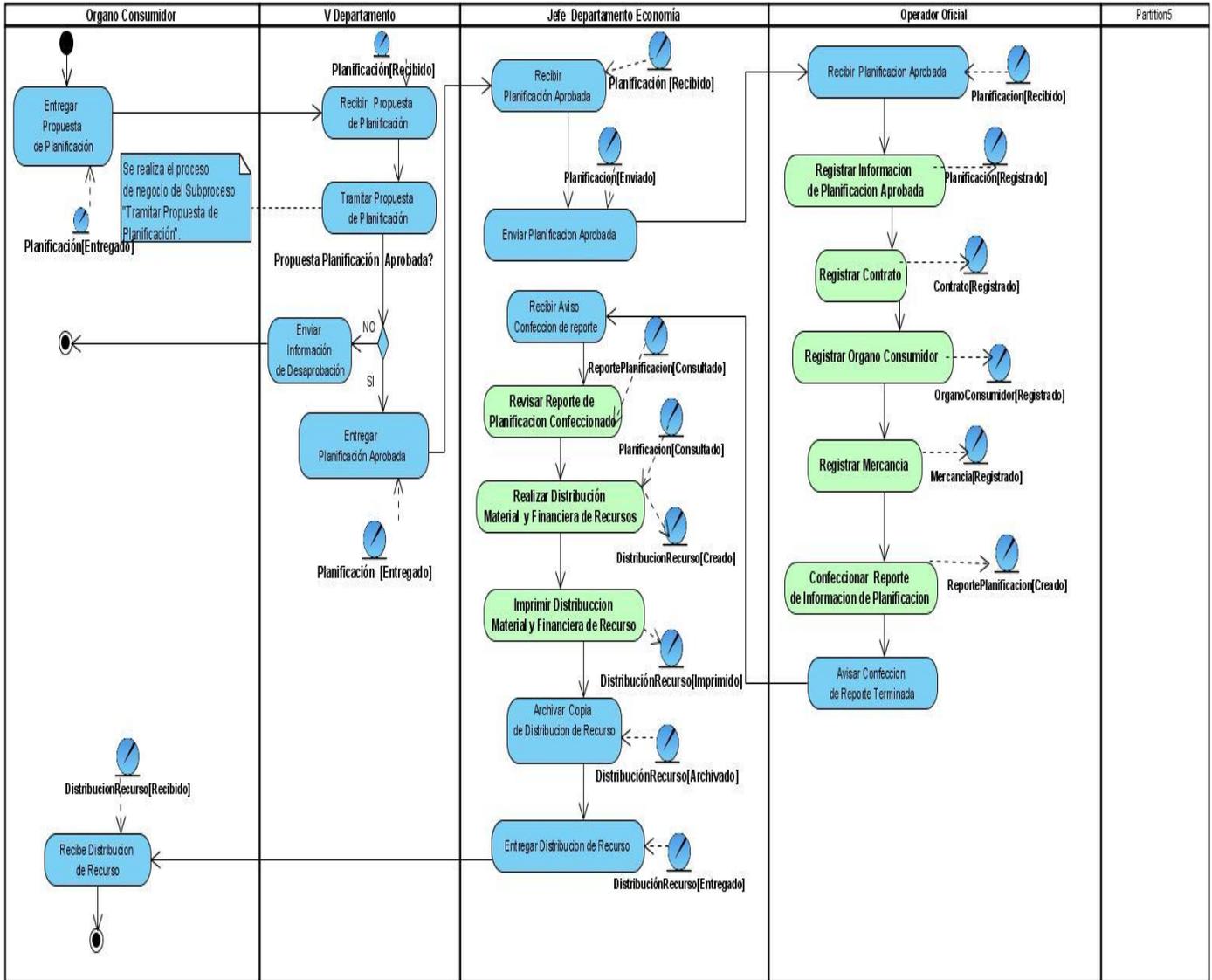


Figura 3. Diagrama de actividades “Ejecutar la Planificación”.

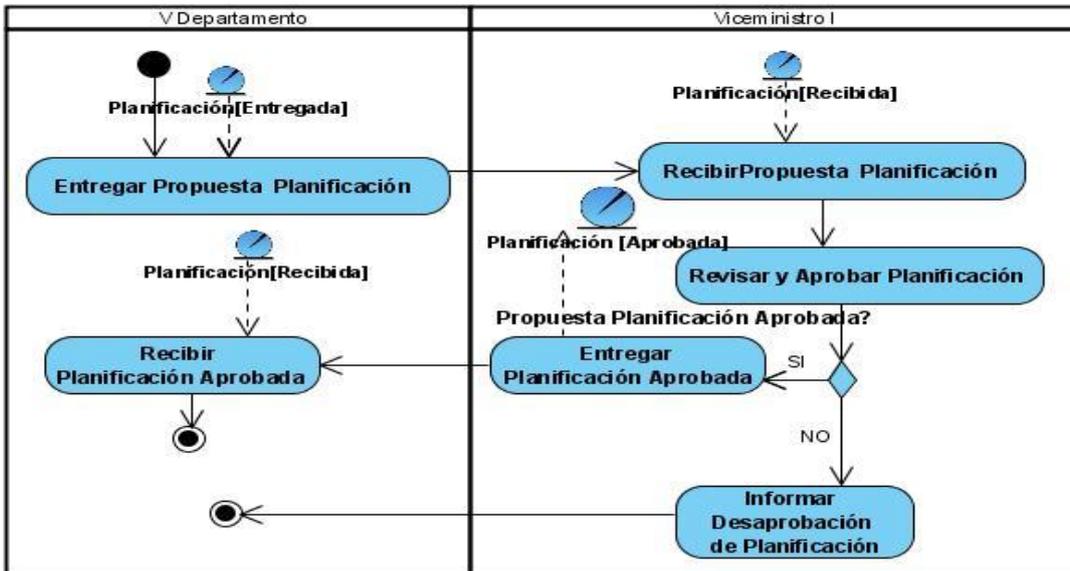


Figura 4. Diagrama de actividades del Subproceso “Tramitar Propuesta de Planificación”.

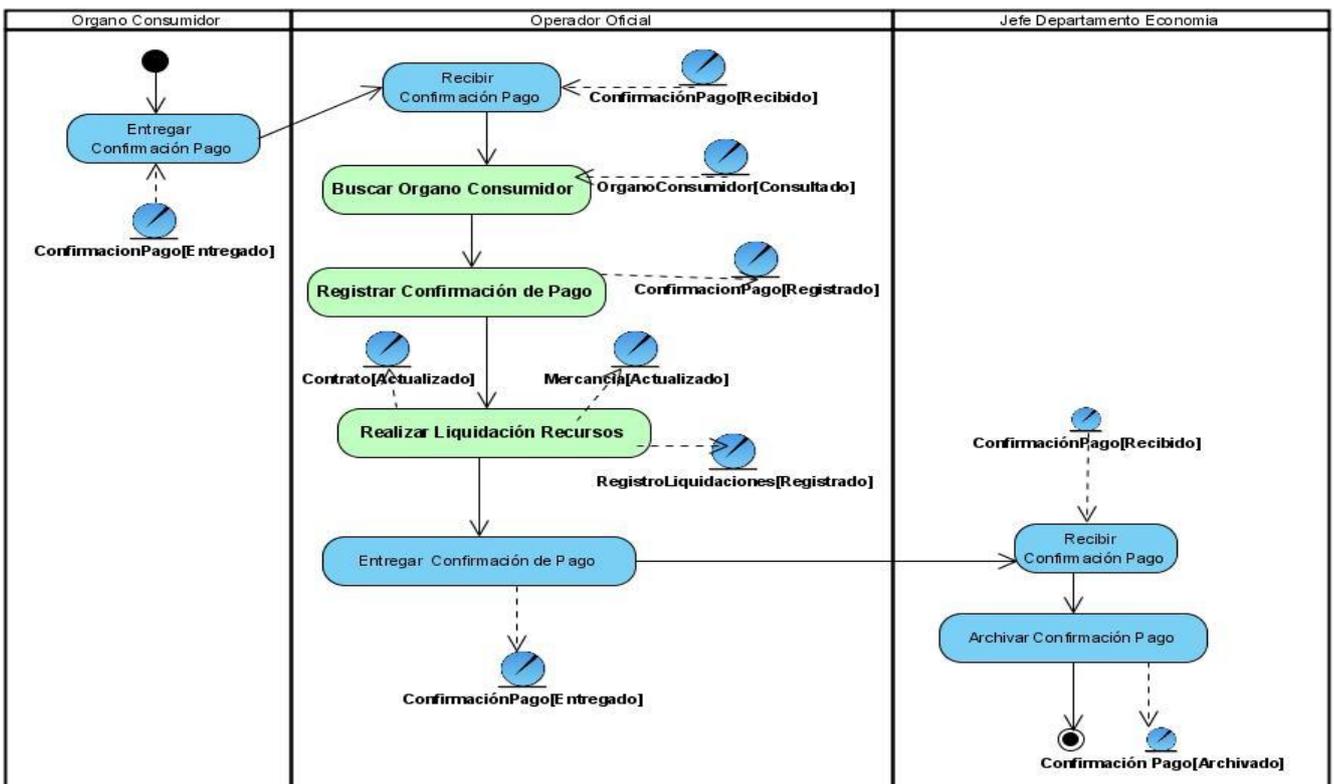


Figura 5. Diagrama de actividades “Confirmar Pago”.

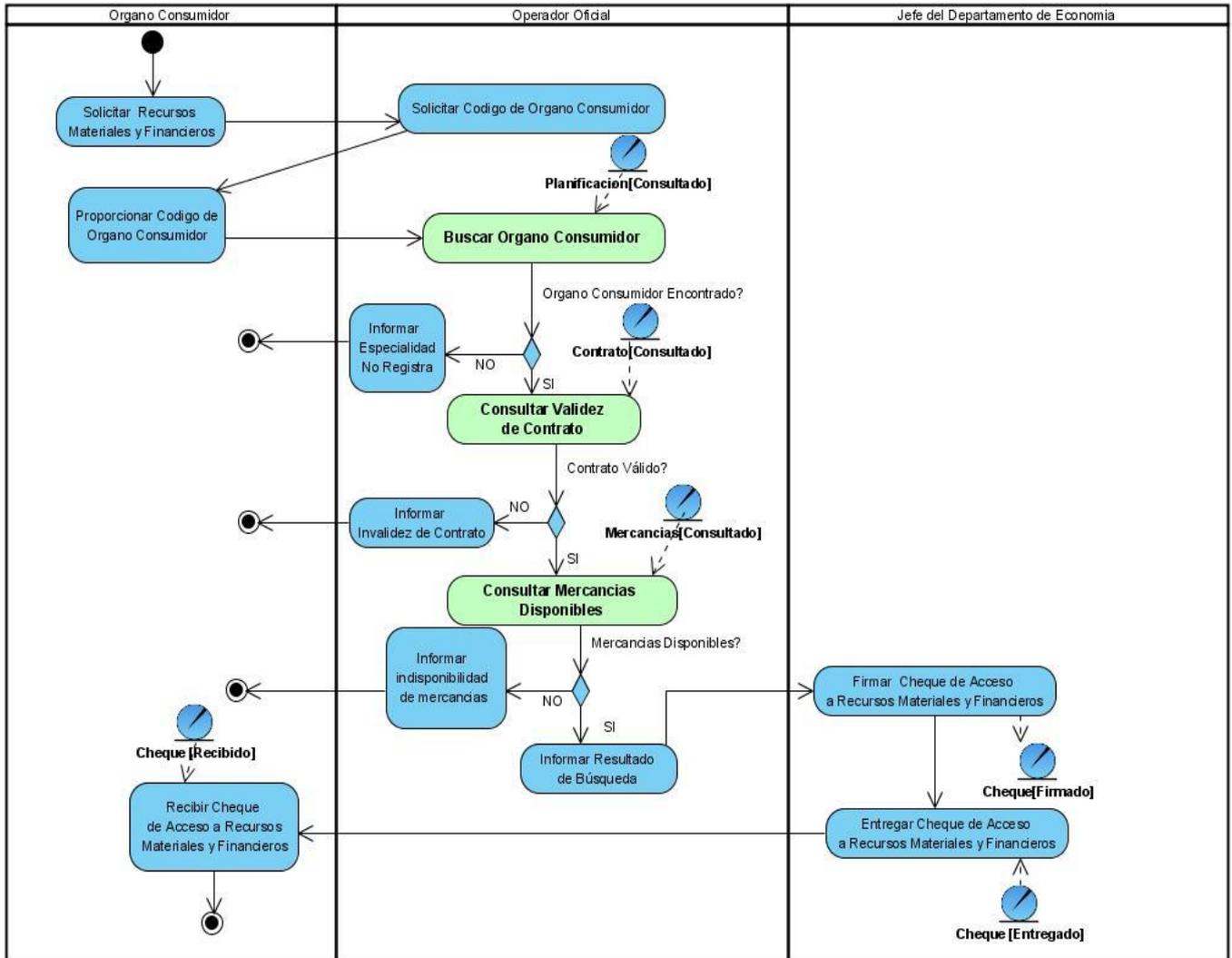


Figura 6. Diagrama de actividades “Controlar la Ejecución de la Planificación”.

2.7 Diagrama de Clases del Modelo de Objetos.

Este diagrama muestra la relación existente entre los trabajadores y las entidades del negocio y proporciona un acercamiento a la identificación de los futuros actores y entidades del sistema (Ver Figura 7).

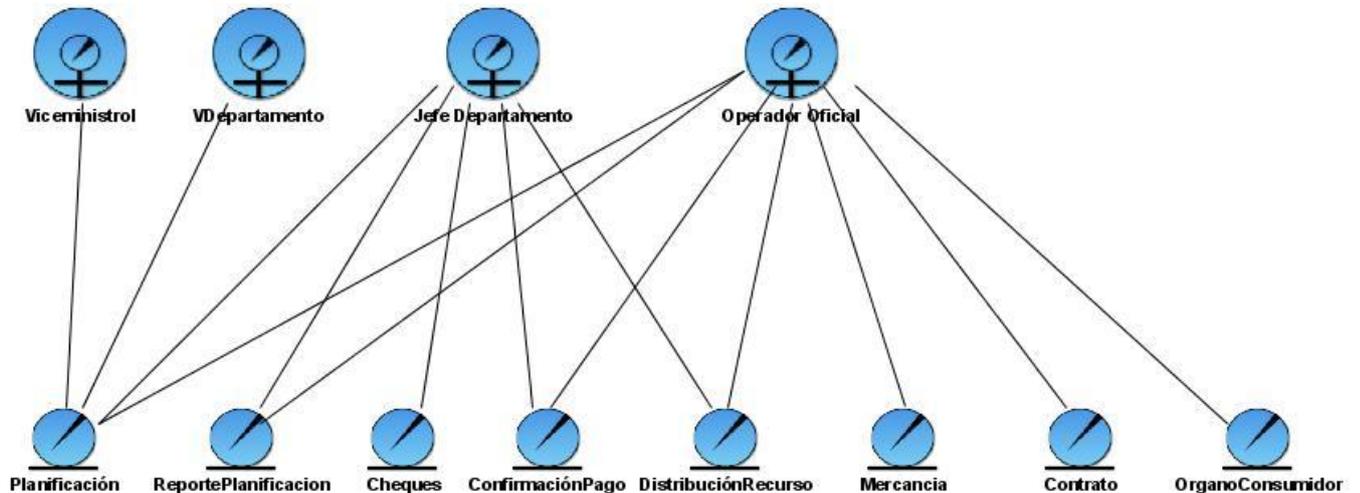


Figura 7. Modelo de Objeto.

2.9 Requerimientos.

Una de las principales tareas en el ciclo de desarrollo de un sistema es la de determinar los requerimientos del sistema de información que constituyen una condición o capacidad que necesita un usuario para resolver un problema o lograr un objetivo. El propósito principal del flujo de trabajo de RUP, captura de requisitos, es guiar el desarrollo hacia el sistema correcto donde se describan con claridad y sin ambigüedades el comportamiento del mismo. Así como lograr una comunicación efectiva entre el cliente (incluyendo a los usuarios) y a los desarrolladores sobre qué debe y qué no debe hacer el sistema.

Los requisitos se pueden clasificar en: funcionales y no funcionales.

2.9.1 Requerimientos Funcionales.

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir. Su meta principal es identificar y documentar las acciones que en realidad debe ejecutar el sistema para que cumpla con los objetivos planteados al inicio de este trabajo.

Todas estas acciones se convierten en requisitos funcionales y de acuerdo a los objetivos propuestos el sistema debe ser capaz de:

- RF1 Gestionar Planificación
 - 1.1 Insertar Planificación
 - 1.2 Eliminar Planificación

1.3 Modificar Planificación

1.4 Mostrar Planificación

RF2 Gestionar Distribución de Recurso

2.1 Insertar Distribución de Recurso

2.2 Modificar Distribución de Recurso

2.3 Eliminar Distribución de Recurso

2.4 Mostrar Distribución de Recurso

RF 3 Gestionar Mercancía

3.1 Insertar Mercancía

3.2 Modificar Mercancía

3.3 Eliminar Mercancía

3.4 Mostrar Mercancía

RF 4 Gestionar Contrato

4.1 Insertar Contrato

4.2 Modificar Contrato

4.3 Eliminar Contrato

4.4 Mostrar Contrato

RF5 Gestionar Confirmación de Pago

5.1 Insertar Confirmación de Pago

5.2 Eliminar Confirmación de Pago

5.3 Modificar Confirmación de Pago

5.4 Mostrar Confirmación de Pago

RF 6 Gestionar Órgano Consumidor

6.1 Insertar Órgano Consumidor

6.2 Modificar Órgano Consumidor

6.3 Eliminar Órgano Consumidor

6.4 Mostrar Órgano Consumidor

RF 7 Mostrar Recuperaciones Especificas

7.1 Mostrar reportes de Distribución de Recursos

7.1.1 Imprimir el reporte visualizado

7.2 Mostrar reportes de Planificación

7.2.1 Imprimir el reporte visualizado

7.3 Mostrar reportes de Confirmación de Pagos

7.3.1 Imprimir el reporte visualizado

7.4 Mostrar reportes de Contratos

7.4.1 Imprimir el reporte visualizado

7.5 Mostrar reportes de Mercancías

7.5.1 Imprimir el reporte visualizado

RF 8 Seleccionar Criterios de Búsqueda.

2.9.2 Requerimientos No Funcionales.

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Estas propiedades o cualidades constituyen las características que hacen al producto atractivo, usable, rápido o confiable. Los requerimientos no funcionales forman una parte significativa de la especificación. Son importantes para que clientes y usuarios puedan valorar las características no funcionales del producto y de esta forma poder decir si el producto tiene la calidad requerida.

Apariencia o interfaz externa:

- Diseño sencillo, con pocas entradas, permitiendo que no sea necesario mucho entrenamiento para utilizar el sistema.

- Debe tener opciones que permitan a los usuarios visualizar reportes y actualizaciones, contendrá un menú dinámico para que actúe con el usuario en dependencia de sus necesidades. Debe estar concebido para simular una aplicación de escritorio.
- Empleo de los colores de la entidad a la cual pertenecerá el sistema a desarrollar: verde, gris, blanco y azul principalmente.

Usabilidad:

- El sistema debe ser de fácil manejo para los usuarios que tengan niveles básicos sobre la computación o hayan realizado algún trabajo previo con sistemas similares.
- Debe tener una opción de ayuda sobre las principales operaciones que se realizan y sus íconos respectivos para lograr un menor tiempo de aprendizaje.
- Debe facilitar, principalmente, el ser manejado por usuarios que estén vinculados al proceso de Planificación Material y Financiera del MINFAR.

Rendimiento:

- Esta aplicación está concebida para un ambiente cliente/servidor así que los tiempos de respuestas deben ser generalmente rápidos (no debe exceder los 10 segundos) al igual que la velocidad de procesamiento de la información.
- El tiempo de búsqueda de información así como la generación de reportes y documentos debe ser en el menor tiempo posible pues se deben generar pantallas dinámicas, implicando esto que el acceso a la base de datos debe ser lo más disponible, rápido y consistente posible.

Soporte:

- Para el servidor de aplicaciones: Se requiere que esté instalado un intérprete de ficheros PHP 5.x y con las últimas actualizaciones del lenguaje.
- Para el cliente: Se requiere que esté instalado un navegador que interprete JavaScript y versiones HTML 3.0 o superior.
- Para el servidor de base de datos: Se requiere que esté instalado un gestor de base de datos que soporte grandes volúmenes de datos, maneje la concurrencia y transacciones.

Portabilidad:

- Requiere de un sistema operativo el cual soporte el navegador Mozilla Firefox e Internet Explorer.

Requerimientos de Hardware:

Para las computadoras del cliente:

- Se requiere tengan tarjeta de red.
- Se requiere tengan al menos 64 MB de memoria RAM.
- Se requiere al menos 100MB de disco duro.
- Procesador 512 MHz como mínimo.
- Para los servidores:
 - Se requiere tarjeta de red.
 - Se requiere tenga al menos 256MB de RAM.
 - Se requiere al menos 1GB de disco duro.
 - Procesador 1.2 GHz como mínimo.

Requerimientos de Software:

- El sistema se desarrollará con tecnología PHP versión 5.0. Correrá sobre un servidor con el sistema operativo UNIX (Linux). Se recurrirá a la tecnología Apache versión 2.0 o superior para el servidor Web. El sistema incluirá una base datos implementada en PostgreSQL versión 8.0 o superior.
- En las computadoras de los clientes debe estar instalado el navegador Mozilla Firefox. Versión 1.5 o superior. La comunicación de las computadoras clientes con el servidor será a través de conexiones de fibra óptica, a una velocidad constante de 100 Mbps o superior.

Seguridad:

- Chequear si el usuario que está accediendo al sistema esta autenticado y brindarle servicio de autenticación.
- Mostrar las operaciones de acuerdo al rol del usuario y no más.
- Mantener la integridad de la información, es decir que no se perderá durante su almacenamiento o transporte.
- Mostrar opción de advertencia antes de borrar cualquier documento o información que pueda existir.
- Garantizar que las funcionalidades del sistema se muestren de acuerdo al nivel de usuario que este activo.

Disponibilidad:

- El sistema deberá estar disponible las 24 horas del día para todos los usuarios con derechos a utilizarlo.

Confiabilidad:

- La información manejada por el sistema está protegida de acceso no autorizado y de divulgación.

Fiabilidad:

- La tasa de fallos del sistema no debe exceder 2 fallos por semana.

Legales:

- El sistema se basa en el manual de normas y principios establecidos por el MINFAR.

2.10 Modelado del Sistema.

El modelado del sistema representa la funcionalidad completa de un sistema mostrando su interacción con los agentes externos, ayudando al cliente, a los usuarios y desarrolladores a llegar a un acuerdo sobre como utilizar el sistema. Los actores utilizan el sistema al interactuar con los casos de uso que no son más que la especificación de una secuencia de acciones lógicas, incluyendo variantes, que el sistema puede llevar a cabo, y que producen un resultado observable de valor para un actor concreto. Las representaciones del sistema se hacen mediante relaciones entre los actores y los casos de uso del sistema. Los diagramas de casos de usos del sistema representan los conjuntos de funcionalidades afines que el sistema debe cumplir para satisfacer todos los requerimientos funcionales que tiene a su cargo.

2.11 Actores del Sistema.

Estos no son más que trabajadores que participan directamente en el negocio y/o nuevos actores que surgen para ejecutar funcionalidades que aparecen debido a la presencia de requisitos especiales (Ver Tabla 4).

Actores del Sistema	Justificación
Jefe del Departamento de Economía	Este trabajador recibe la planificación aprobada, realiza la distribución de los recursos del Órgano Consumidor y la imprime, firma el cheque de acceso a los recursos materiales y financieros.

<p>Operador Oficial</p>	<p>Este trabajador es quien registra la planificación aprobada, las mercancías y los contratos, recibe las confirmaciones de pagos de los Órganos Consumidores, realiza la liquidación de recursos, actualiza los contratos y mercancías de los mismos.</p>
--------------------------------	---

Tabla 6. Descripción de los Actores del Sistema.

2.12 Diagrama de Casos de Uso del Sistema.

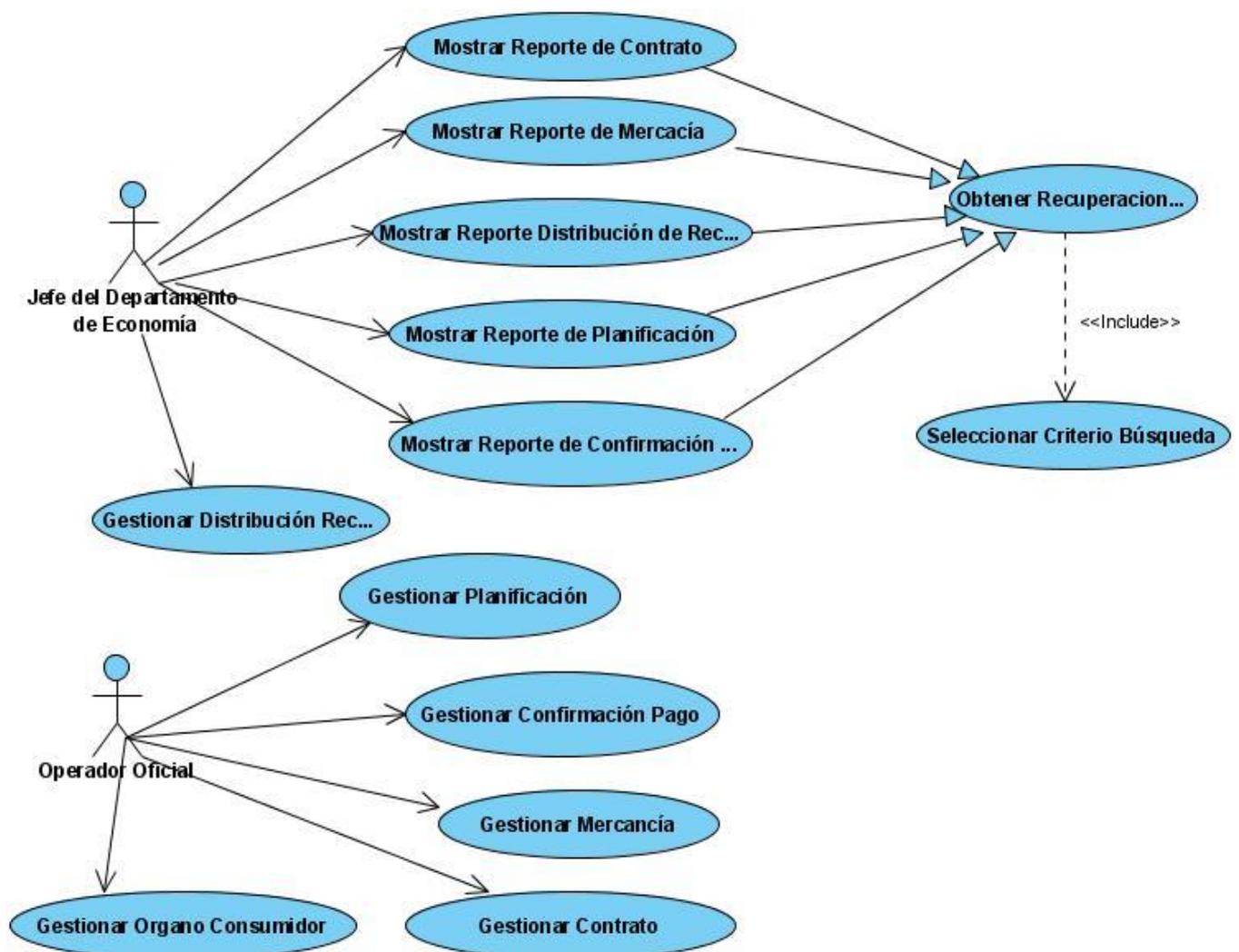
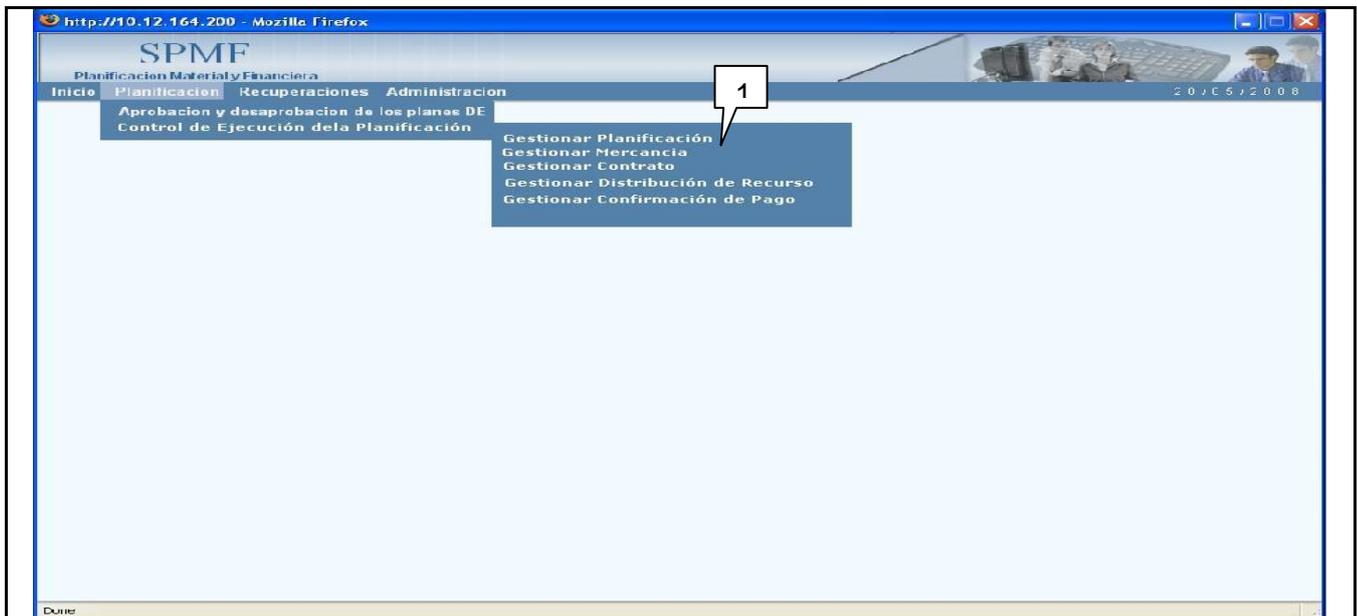


Figura 8 Diagrama de Casos de Uso del Sistema.

2.13 Casos de Uso expandidos.

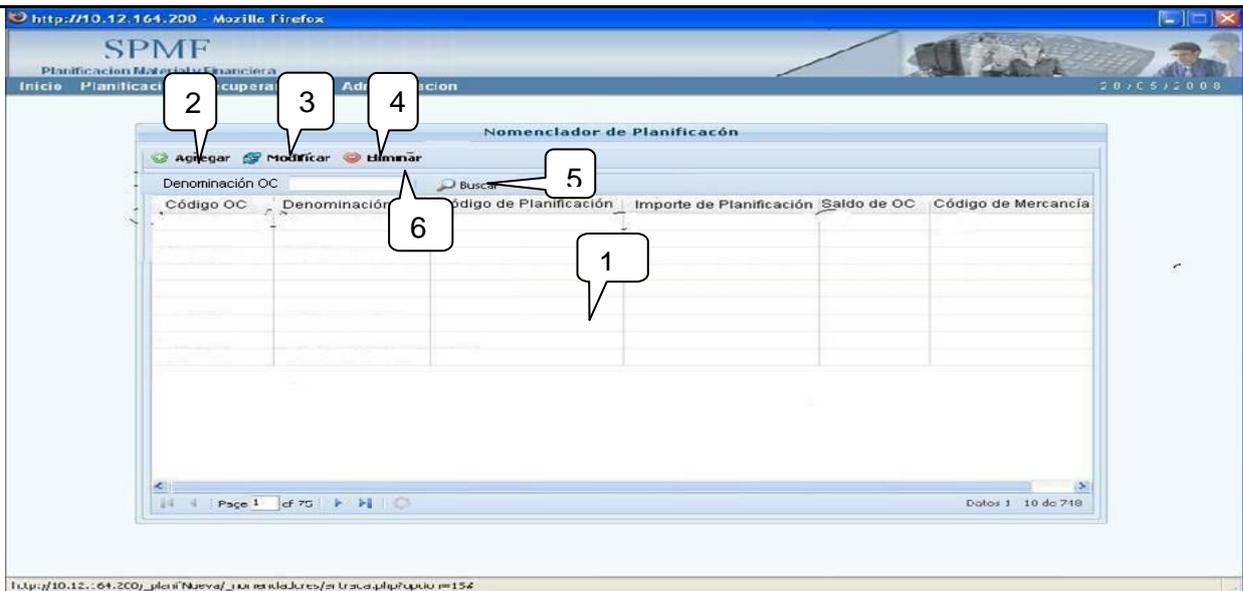
La descripción puede ser elaborada de forma breve o extendida y debe ir acompañada del prototipo respectivo. El prototipo del sistema que se construye en este punto da una visión de las pantallas diseñadas para cada caso de uso, pero con comportamiento estático, que se presenta al usuario para verificar los requerimientos funcionales.

Caso de Uso	Gestionar Planificación.
Actores	Operador Oficial.
Propósito	El propósito de este caso de uso es que el Operador Oficial pueda realizar actualizaciones y mostrar la planificación de recursos materiales y financieros del Órgano Consumidor.
Resumen	El caso de uso se inicia cuando el Operador Oficial desea gestionar la planificación de los Órganos Consumidores. El sistema muestra la interfaz de gestionar planificación, donde el Operador Oficial puede agregar una nueva planificación, eliminar o modificar algunos datos ya existentes. Finalmente se archivan los datos en el sistema y se muestra la planificación actualizada. Terminando así el caso de uso.
Tipo	Real y expandido.
Precondiciones.	El actor se debe encontrar trabajando con el sistema y tener acceso a gestionar planificación.
Referencias	RF.1, 1.1, 1.2, 1.3, 1.4
Casos de Uso relacionados:	
Interfaz I	



(1) Submenú para seleccionar la acción que desea realizar.

Interfaz II



(1) Grid para mostrar los datos (si existen) de la Distribución de Recursos, inicialmente vacío, hasta que se inserte por primera vez una planificación. **Nombre:** GridDatos, **tipo:** objeto

(2) Botón para Agregar planificación. **Nombre:** Agregar, **tipo:** button

(3) Botón para Modificar planificación. **Nombre:** Modificar, **tipo:** button

(4) Botón para Eliminar planificación. **Nombre:** Eliminar, **tipo:** button

(5) Botón para Buscar planificación. **Nombre:** Buscar, **tipo:** button

(6)ComboBox para seleccionar la denominación del Órgano Consumidor.**Nombre:** den_oc,tipo :string

Flujo normal de eventos

Acción del actor	Respuesta del sistema
1. El actor desea gestionar la Planificación Material y Financiera y selecciona la opción<Gestionar Planificación > del menú principal Interfaz I	2. El sistema visualiza la Interfaz II.
3. El actor selecciona la opción que desea realizar sobre la planificación. <ul style="list-style-type: none"> • Agregar planificación • Modificar planificación • Eliminar planificación 	4. El sistema realiza las operaciones según haya sido la selección del actor. <ul style="list-style-type: none"> • Agregar planificación (2) (ver sección Insertar). • Modificar planificación (3) (ver sección Modificar). • Eliminar planificación (4) (ver sección Eliminar).

Sección Insertar

Interfaz III



- (1) Textfield para insertar la fecha en que se registra la planificación. **Nombre:**fecha,**tipo** date
- (2) Textfield para cargar el código del órgano consumidor. **Nombre:**cod_oc,**tipo** string
- (3) ComboBox seleccionar la Denominación del Órgano Consumidor. **Nombre:** den_oc, **tipo:** string
- (4)Textfield para cargar el código del contrato.**Nombre:**cod_contrato,**tipo:**string
- (5) Textfield para insertar importe de planificación.**Nombre:**importe_plani, **tipo:** double.
- (6) Textfield para insertar saldo del Órgano consumidor. **Nombre:** saldo, **tipo:** double.

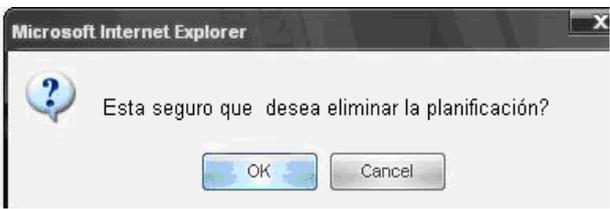
DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

<p>(7) ComboBox para seleccionar la denominación de la mercancía. Nombre :den_mercancia,tipo:string</p> <p>(8) Botón para aceptar la inserción de los datos :Nombre :Aceptar,tipo:button</p> <p>(9) Botón para cancelar la inserción de los datos :Nombre Cancelar,tipo :button</p>	
Acción del actor	Respuesta del sistema
<p>1. El actor desea introducir una planificación y selecciona la opción <Insertar>. Interfaz II (2)</p>	<p>2. El sistema le muestra el formulario (Interfaz III) y las opciones Aceptar y Cancelar activas.</p>
<p>3. El actor introduce los datos de la planificación. En la opción:</p> <p>(1) Fecha de registro.</p> <p>(2) Código del Contrato.</p> <p>(3) Denominación de Mercancía.</p> <p>(4) Denominación de Órgano Consumidor.</p> <p>(5) Saldo del Órgano Consumidor.</p>	
<p>4. El actor selecciona <Aceptar> Interfaz III (8).</p>	<p>5. El sistema verifica que los datos introducidos sean correctos y que no exista una planificación con ese código. Guarda los cambios en sistema y envía mensaje al actor.</p> <div style="text-align: center;">  </div>
<p>6. El actor oprime el botón <OK>.</p>	
<p>Sección Modificar</p>	
<p>Interfaz IV</p>	



- (1) Textfield para insertar la fecha en que se registra la planificación. **Nombre:** fecha, **tipo** date
- (2) Textfield para cargar el código del Órgano Consumidor. **Nombre:** cod_oc, **tipo** string
- (3) ComboBox seleccionar la Denominación del Órgano Consumidor. **Nombre:** den_oc, **tipo:** string
- (4) Textfield para insertar el código del contrato. **Nombre:** cod_contrato, **tipo:** string
- (5) Textfield para insertar importe de planificación. **Nombre:** importe_plani, **tipo:** double.
- (6) Textfield para insertar saldo del Órgano Consumidor. **Nombre:** saldo, **tipo:** double.
- (7) ComboBox para seleccionar la denominación de la mercancía. **Nombre** :den_mercancia, **tipo:** string
- (8) Botón para aceptar la inserción de los datos :Nombre :Aceptar, **tipo:** button
- (9) Botón para cancelar la inserción de los datos :Nombre Cancelar, **tipo** :button

Acción del actor	Respuesta del sistema
1. El actor busca la planificación que desea modificar y selecciona la opción <Modificar> en la Interfaz II (3).	2. El sistema muestra la Interfaz IV con los datos de la planificación y muestra las opciones Aceptar y Cancelar activas.
3. El actor realiza los cambios necesarios por teclado y selecciona <Aceptar>.	4. El sistema verifica que los datos modificados sean correctos. Guarda los cambios y envía mensaje al actor. 
5. El actor oprime el botón <ok>.	
Sección Eliminar	
Acción del actor	Respuesta del sistema

<p>1. El actor busca la planificación que desea eliminar del documento de planificación Interfaz II (5) y selecciona la opción <Eliminar> en la Interfaz II (3).</p>	<p>2. El sistema pide confirmación al actor</p> 
<p>3. El actor responde "OK"</p>	<p>4. El sistema elimina la planificación señalada por el usuario. Guarda los datos en el sistema. Devuelve el documento de planificación actualizado.</p>
<p>Sección Mostrar</p>	
<p>Acción del actor</p>	<p>Respuesta del sistema</p>
<p>1. El actor desea mostrar una planificación específica, introduce el código del Órgano Consumidor y selecciona la opción <Buscar> Interfaz II (5).</p>	<p>2. El sistema muestra en el Grid de la Interfaz II la planificación buscada.</p>
<p>Cursos Alternos</p>	
<p>Sección Insertar.</p>	
<p>Línea 4. En caso que el actor no introduce los datos de la planificación y oprima <Aceptar> Interfaz III (8), el sistema envía mensaje al actor.</p>	
	
<p>Línea 5. Si el actor introduce una planificación que ya existe, el sistema envía mensaje al actor.</p>	
	
<p>Línea 4. Si el actor decide <Cancelar> el sistema no registra los cambios y el sistema regresa a la interfaz inicial.</p>	
<p>Sección Modificar.</p>	
<p>Línea 1. Si el actor no ha seleccionado la planificación a modificar y selecciona <Modificar> el</p>	

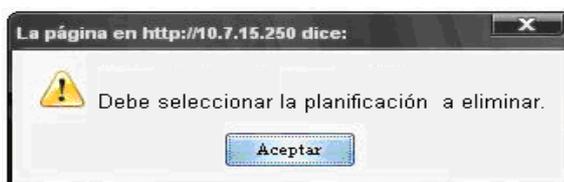
sistema le envía un mensaje.



Línea 3. Si el actor decide <Cancelar> el sistema regresa a la interfaz inicial.

Sección Eliminar.

Línea 1. Si el actor no ha seleccionado la planificación a eliminar y selecciona <Eliminar> el sistema le envía un mensaje.



Línea 3. Si el actor decide <Cancelar> el sistema regresa a la interfaz inicial.

Pos-condiciones	El documento de planificación queda actualizado.
Prioridad	Crítico

Tabla 7. Descripción Textual del caso de uso “Gestionar Planificación”.

Caso de Uso	Gestionar Mercancía
Actores	Operador Oficial.
Propósito	El propósito de este caso de uso es que el Operador Oficial pueda realizar actualizaciones y mostrar las mercancías destinadas a los Órganos Consumidores.
Resumen	El caso de uso se inicia cuando el Operador Oficial desea gestionar las mercancías de los Órganos Consumidores. El sistema muestra la interfaz de gestionar mercancías, donde el Operador Oficial puede agregar una nueva mercancía, eliminar o modificar algunos datos ya existentes. Finalmente se archivan los datos en el sistema y se muestra la mercancía actualizada. Terminando así el caso de uso.
Tipo	Real y expandido.
Precondiciones.	El actor se debe encontrar trabajando con el sistema y tener acceso a

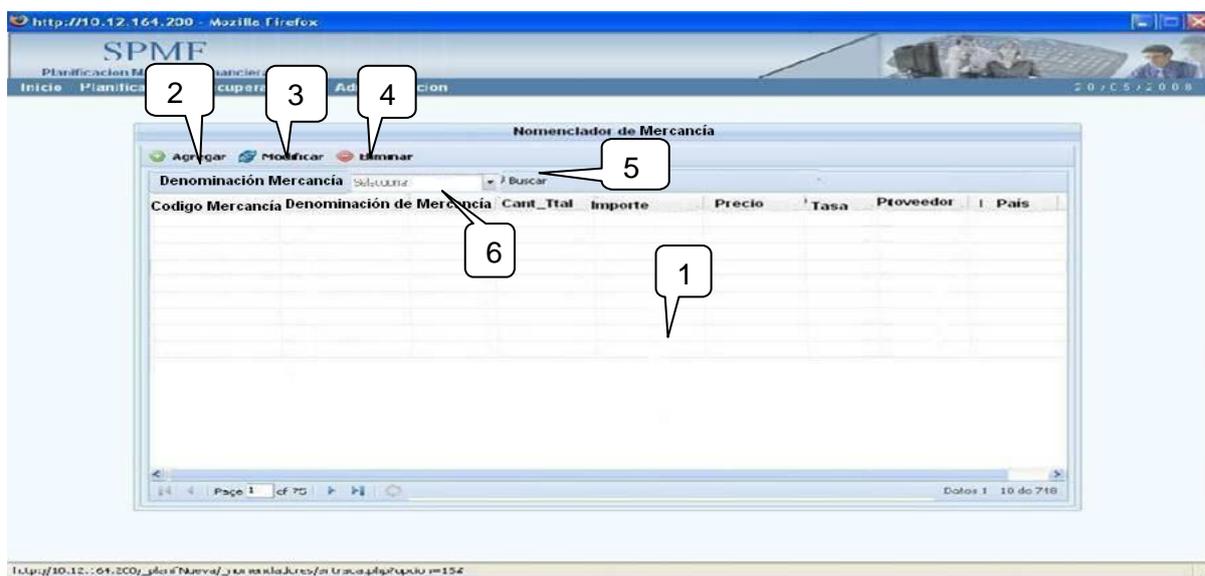
DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

	gestionar mercancía.
Referencias	RF.3, 3.1, 3.2, 3.3, 3.4
Casos de Uso relacionados:	

Ver Interfaz I de la Tabla 7 Descripción expandida del Caso de Uso: "Gestionar Planificación"

(1) Submenú para seleccionar la acción que desea realizar.

Interfaz II



(1) Grid para mostrar los datos (si existen) de las Mercancías, inicialmente vacío, hasta que se inserte por primera vez una mercancía. **Nombre:** GridDatos, **tipo:** objeto

(2) Botón para Agregar mercancía. **Nombre:** Agregar, **tipo:** button

(3) Botón para Modificar mercancía. **Nombre:** Modificar, **tipo:** button

(4) Botón para Eliminar mercancía. **Nombre:** Eliminar, **tipo:** button

(5) Botón para Buscar mercancía. **Nombre:** Buscar, **tipo:** button

(6) ComboBox para seleccionar la denominación de la mercancía. **Nombre:** den_oc, **tipo:** string

Flujo normal de eventos

Acción del actor	Respuesta del sistema
1. El actor desea gestionar las mercancías asignadas al Órgano Consumidor y selecciona la opción <Gestionar	2. El sistema visualiza la Interfaz II .

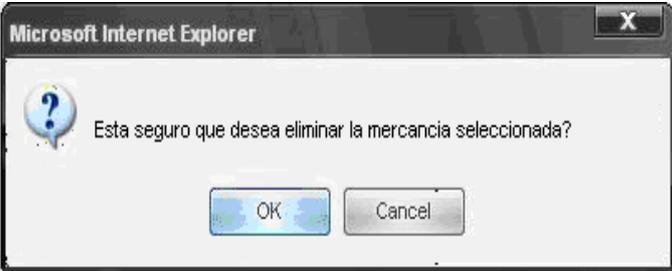
<p>Mercancía > del menú principal</p> <p>Interfaz I</p>	
<p>3. El actor selecciona la opción que desea realizar sobre las mercancías.</p> <ul style="list-style-type: none"> • Agregar mercancía. • Modificar mercancía. • Eliminar mercancía. 	<p>4. El sistema realiza las operaciones según haya sido la selección del actor.</p> <ul style="list-style-type: none"> • Agregar mercancía (2) (ver sección Insertar). • Modificar mercancía (3) (ver sección Modificar). • Eliminar mercancía (4) (ver sección Eliminar).
<p>Sección Insertar</p>	
<p>Interfaz III</p>	
	
<p>(1) ComboBox para seleccionar el país del que procede. Nombre:país, tipo:string</p> <p>(2) ComboBox para seleccionar la denominación de la mercancía. Nombre:name_mercancia, tipo:string</p> <p>(3) Textfield para cargar el código de la mercancía. Nombre :cod_mercancia, tipo:string</p> <p>(4) Textfield para insertar la tasa de mercancía. Nombre:tasa, tipo:double</p> <p>(5) Textfield para insertar el importe de la mercancía. Nombre:importe, tipo:double</p> <p>(6) ComboBox para seleccionar el proveedor. Nombre:proveedor, tipo:string</p> <p>(7) Botón Aceptar. Nombre: Aceptar, tipo: button</p> <p>(8) Botón Cancelar la operación. Nombre: Cancelar, tipo: button.</p>	
<p>Acción del actor</p>	<p>Respuesta del sistema</p>
	<p>1. El sistema le muestra el formulario (Interfaz III) y las opciones Aceptar y Cancelar activas.</p>
<p>2. El actor introduce los datos de la mercancía. En la opción:</p>	

<p>(1) Denominación de Mercancía. (2) Cantidad total (3) Importe de Mercancía. (4) Precio de Mercancía. (5) Tasa. (6) Proveedor. (7) País</p>	
<p>3. El actor selecciona <Aceptar> Interfaz III (7).</p>	<p>4. El sistema verifica que los datos introducidos sean correctos. Guarda los cambios en sistema y envía mensaje al actor.</p> 
<p>5. El actor oprime el botón <ok>.</p>	<p>6. Devuelve las mercancías actualizadas visualizándolas en el Grid Interfaz II (1)</p>

Sección Modificar

Interfaz IV

	
<p>(1) ComboBox para seleccionar el país del que procede. Nombre:país, tipo:string</p> <p>(2) ComboBox para seleccionar la denominación de la mercancía. Nombre:name_mercancia, tipo:string</p> <p>(3) Textfield para cargar el código de la mercancía. Nombre :cod_mercancia, tipo:string</p> <p>(4) Textfield para insertar la tasa de mercancía. Nombre:tasa, tipo:double</p> <p>(5) Textfield para insertar el importe de la mercancía. Nombre:importe, tipo:double</p> <p>(6) ComboBox para seleccionar el proveedor. Nombre:proveedor, tipo:string</p> <p>(7) Botón Aceptar. Nombre: Aceptar, tipo: button</p>	

(8) Botón Cancelar la operación. Nombre: Cancelar, tipo: button.	
Acción del actor	Respuesta del sistema
1. El actor selecciona la mercancía que desea modificar y selecciona la opción <Modificar> en la Interfaz II (3).	2. El sistema muestra la Interfaz II con los datos de la mercancía y muestra las opciones Aceptar y Cancelar activas.
3. El actor realiza los cambios necesarios por teclado y selecciona <Aceptar>.	4. El sistema verifica que los datos modificados sean correctos. Guarda los cambios en el sistema. Devuelve la mercancía actualizada y envía mensaje al actor. 
Sección Eliminar	
Acción del actor	Respuesta del sistema
1. El actor selecciona la mercancía que desea eliminar y selecciona la opción <Eliminar> en la Interfaz II (4).	2. El sistema pide confirmación al actor 
3. El actor responde "OK"	4. El sistema elimina la mercancía señalada por el usuario. Guarda los datos en el sistema. Devuelve las mercancías actualizadas.
Sección Mostrar	
Acción del actor	Respuesta del sistema
1. El actor desea mostrar una mercancía para un Órgano Consumidor específico, introduce el código del Órgano Consumidor y selecciona la opción <Buscar> Interfaz II (5).	2. El sistema muestra en el Grid de la Interfaz II las mercancías del Órgano Consumidor buscado.

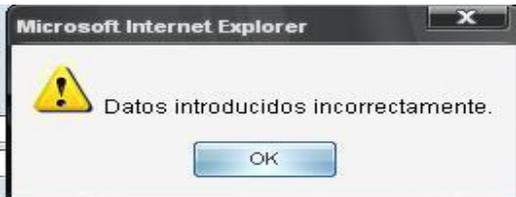
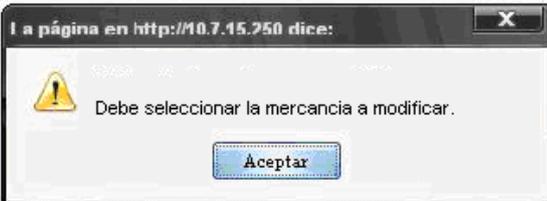
Cursos Alternos	
Sección Insertar.	
<p>Línea 3. En caso que el actor no introduce los datos de la mercancía y oprima <Aceptar> Interfaz III (7), el sistema envía mensaje al actor.</p>	
	
<p>Línea 3. Si el actor decide <Cancelar> el sistema no registra los cambios y el sistema regresa a la interfaz inicial.</p>	
Sección Modificar.	
<p>Línea 1. Si el actor no ha seleccionado la mercancía a modificar y selecciona <Modificar> el sistema le envía un mensaje.</p>	
	
<p>Línea 3. Si el actor decide <Cancelar> el sistema regresa a la interfaz inicial.</p>	
Sección Eliminar.	
<p>Línea 1. Si el especialista no ha seleccionado la mercancía a eliminar y selecciona <Eliminar> el sistema le envía un mensaje.</p>	
	
<p>Línea 3. Si el actor decide <Cancelar> el sistema regresa a la interfaz inicial.</p>	
Pos-condiciones	Las mercancías quedan actualizadas.
Prioridad	Crítico

Tabla 8. Descripción Textual del caso de uso “Gestionar Mercancía”.

Caso de Uso	Gestionar Contrato
Actores	Operador Oficial.

DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

Propósito	El propósito de este caso de uso es que el Operador Oficial pueda realizar actualizaciones y mostrar los contratos de los Órganos Consumidores.
Resumen	El caso de uso se inicia cuando el Operador Oficial desea gestionar los contratos de los Órganos Consumidores. El sistema muestra la interfaz de gestionar contrato, donde el Operador Oficial puede agregar un nuevo contrato, eliminar o modificar algunos datos ya existentes. Finalmente se archivan los datos en el sistema y se muestra el contrato actualizado. Terminando así el caso de uso.
Tipo	Real y expandido.
Precondiciones.	El actor se debe encontrar trabajando con el sistema y tener acceso a gestionar contratos.
Referencias	RF.4 , 4.1, 4.2, 4.3, 4.4
Casos de Uso relacionados:	

Interfaz I

Ver Interfaz I de la Tabla 7 Descripción expandida del Caso de Uso: "Gestionar Planificación "

(1) Submenú para seleccionar la acción que desea realizar.

Interfaz II

- (1) Grid para mostrar los datos (si existen) de los Contratos, inicialmente vacío, hasta que se inserte por primera vez un contrato. **Nombre:** GridDatos, **tipo:** objeto
- (2) Botón para Agregar contrato. **Nombre:** Agregar, **tipo:** button
- (3) Botón para Modificar contrato. **Nombre:** Modificar, **tipo:** button
- (4) Botón para Eliminar contrato. **Nombre:** Eliminar, **tipo:** button
- (5) Botón para Buscar contrato. **Nombre:** Buscar, **tipo:** button
- (6) ComboBox para seleccionar la denominación del Órgano Consumidor. **Nombre:** den_oc, **tipo:** :string

Flujo normal de eventos	
Acción del actor	Respuesta del sistema
<p>1. El actor desea gestionar los contratos de los Órganos Consumidores y selecciona la opción <Gestionar Contrato > del menú principal Interfaz I</p>	<p>2. El sistema visualiza la Interfaz II.</p>
<p>3. El actor selecciona la opción que desea realizar sobre los contratos.</p> <ul style="list-style-type: none"> Agregar contratos. Modificar contratos. Eliminar contratos. Mostrar contratos. 	<p>4. El sistema realiza las operaciones según haya sido la selección del actor.</p> <ul style="list-style-type: none"> Agregar contratos (2) (ver sección Insertar). Modificar contratos (3) (ver sección Modificar). Eliminar contratos (4) (ver sección Eliminar). Mostrar contratos (ver sección mostrar)

Sección Insertar

Interfaz III

The screenshot shows a form titled "Insertar Contrato" with the following elements and callouts:

- 1:** Callout pointing to the "Denominación OC" dropdown menu.
- 2:** Callout pointing to the "Código de Contrato" text input field.
- 3:** Callout pointing to the "Importe de Contrato" text input field.
- 4:** Callout pointing to the "Saldo de Contrato" text input field.
- 5:** Callout pointing to the "Tipo de Moneda" dropdown menu.
- 6:** Callout pointing to the "Tasa de Contrato" text input field.
- 7:** Callout pointing to the "Fecha de Inicio" date picker.
- 8:** Callout pointing to the "Fecha de Vencimiento" date picker.
- 9:** Callout pointing to the "Anticipo Pago" dropdown menu.
- 10:** Callout pointing to the "Aceptar" button.
- 11:** Callout pointing to the "Cancelar" button.

DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

<p>(1) ComboBox para la denominación del Órgano Consumidor. Nombre: cod_oc, tipo: string</p> <p>(2) Textfield para insertar el código del contrato. Nombre: cod_contrato, tipo: string</p> <p>(3) Textfield para insertar el importe del contrato. Nombre: importe, tipo: double.</p> <p>(4) Textfield para insertar el saldo del contrato. Nombre: saldo, tipo: double.</p> <p>(5) ComboBox seleccionar el tipo de moneda. Nombre: moneda, tipo: string.</p> <p>(6) Textfield para insertar la tasa del contrato. Nombre: tasa, tipo: double.</p> <p>(7) DateField para insertar la fecha de inicio. Nombre: fecha_ini, tipo: date.</p> <p>(8) DateField para insertar la fecha de fin. Nombre: fecha_fin, tipo: date.</p> <p>(9) ComboBox para seleccionar si tiene anticipación de pago o no. Nombre: anticipo, tipo: bool.</p> <p>(10) Botón Aceptar. Nombre: Aceptar, tipo: button</p> <p>(11) Botón Cancelar la operación. Nombre: Cancelar, tipo: button.</p>	
Acción del actor	Respuesta del sistema
	<p>1. El sistema le muestra el formulario (Interfaz III y las opciones Aceptar y Cancelar activas.</p>
<p>2. El actor introduce los datos del contrato por teclado. En la opción:</p> <p>(1) Código de Contrato.</p> <p>(2) Código OC.</p> <p>(3) Saldo de Contrato</p> <p>(4) Importe de Contrato.</p> <p>(5) Tipo Moneda</p> <p>(6) Anticipo de Pago</p> <p>(7) Tasa.</p> <p>(8) Fecha inicio.</p> <p>(9) Fecha fin.</p>	
<p>3. El actor selecciona <Aceptar> Interfaz III (10).</p>	<p>4. El sistema verifica que los datos introducidos sean correctos. Guarda los cambios. Devuelve el contrato actualizado visualizándolo en el Grid Interfaz II (1) y envía mensaje al actor.</p>



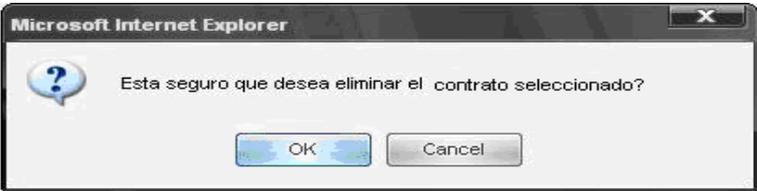
Sección Modificar

Interfaz IV



- (1) ComboBox para seleccionar la denominación del ÓrganoConsumidor. **Nombre:** den_oc, **tipo:** string
- (2) Textfield para insertar el código del contrato. **Nombre:** cod_contrato, **tipo:** string
- (3) Textfield para insertar el importe del contrato. **Nombre:** importe, **tipo:** double.
- (4) Textfield para insertar el saldo del contrato. **Nombre:** saldo, **tipo:** double.
- (5) ComboBox seleccionar el tipo de moneda. **Nombre:** moneda, **tipo:** string.
- (6) Textfield para insertar la tasa del contrato. **Nombre:** tasa, **tipo:** double.
- (7) DateField para insertar la fecha de inicio. **Nombre:** fecha_ini, **tipo:** date.
- (8) DateField para insertar la fecha de fin. **Nombre:** fecha_fin, **tipo:** date.
- (9) ComboBox para seleccionar si tiene anticipación de pago o no. **Nombre:** anticipo, **tipo:** bool.
- (10) Botón Aceptar. **Nombre:** Aceptar, **tipo:** button
- (11) Botón Cancelar la operación. **Nombre:** Cancelar, **tipo:** button.

Acción del actor	Respuesta del sistema
<p>1. El actor selecciona el contrato que desea modificar y selecciona la opción <Modificar> en la Interfaz II (3).</p>	<p>2. El sistema muestra la Interfaz III con los datos del contrato cargados y muestra las opciones Aceptar y Cancelar activas.</p>

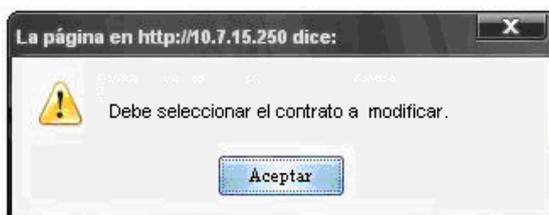
<p>3. El actor realiza los cambios necesarios por teclado y selecciona la opción <Aceptar></p>	<p>4. El sistema verifica que los datos modificados sean correctos. Guarda los cambios en el sistema. Devuelve el contrato actualizado y envía mensaje al actor.</p> 
<p>Sección Eliminar</p>	
<p>Acción del actor</p>	<p>Respuesta del sistema</p>
<p>1. El actor selecciona el contrato que desea eliminar y selecciona la opción <Eliminar> en la Interfaz II (4).</p>	<p>2. El sistema pide confirmación al actor</p> 
<p>3. El actor responde "OK"</p>	<p>4. El sistema elimina el contrato señalado por el usuario. Guarda los datos en el sistema. Devuelve los contratos actualizados.</p>
<p>Sección Mostrar</p>	
<p>Acción del actor</p>	<p>Respuesta del sistema</p>
<p>1. El actor desea mostrar un contrato para un Órgano Consumidor específico, introduce el código del Órgano Consumidor y selecciona la opción <Buscar> Interfaz II (5).</p>	<p>2. El sistema muestra en el Grid de la Interfaz II el contrato buscado.</p>
<p>Cursos Alternos</p>	
<p>Sección Insertar.</p>	
<p>Línea 4. En caso que el actor introduce los datos del contrato y oprima <Aceptar> Interfaz III (10), el sistema envía mensaje al actor.</p>	
	
<p>Línea 4. Si el actor introduce un contrato que ya existe, el sistema envía mensaje al actor.</p>	



Línea 3. Si el actor decide <Cancelar> el sistema no registra los cambios y el sistema regresa a la interfaz inicial.

Sección Modificar.

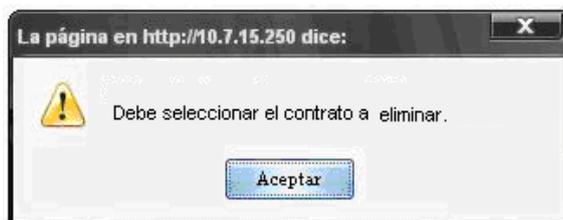
Línea 1. Si el actor no ha seleccionado el contrato a modificar y selecciona <Modificar> el sistema le envía un mensaje.



Línea 3. Si el actor decide <Cancelar> el sistema regresa a la interfaz inicial.

Sección Eliminar.

Línea 1. Si el actor no selecciona el contrato a eliminar y selecciona <Eliminar> el sistema le envía un mensaje.



Línea 3. Si el actor decide <Cancelar> el sistema regresa a la interfaz inicial.

Pos-condiciones	Los Contratos quedan actualizados.
Prioridad	Crítico

Tabla 9. Descripción Textual del caso de uso “Gestionar Contrato”.

Caso de Uso	Gestionar Órgano Consumidor.
Actores	Operador Oficial.
Propósito	El propósito de este caso de uso es que el Operador Oficial pueda realizar actualizaciones y mostrar los Órganos

DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

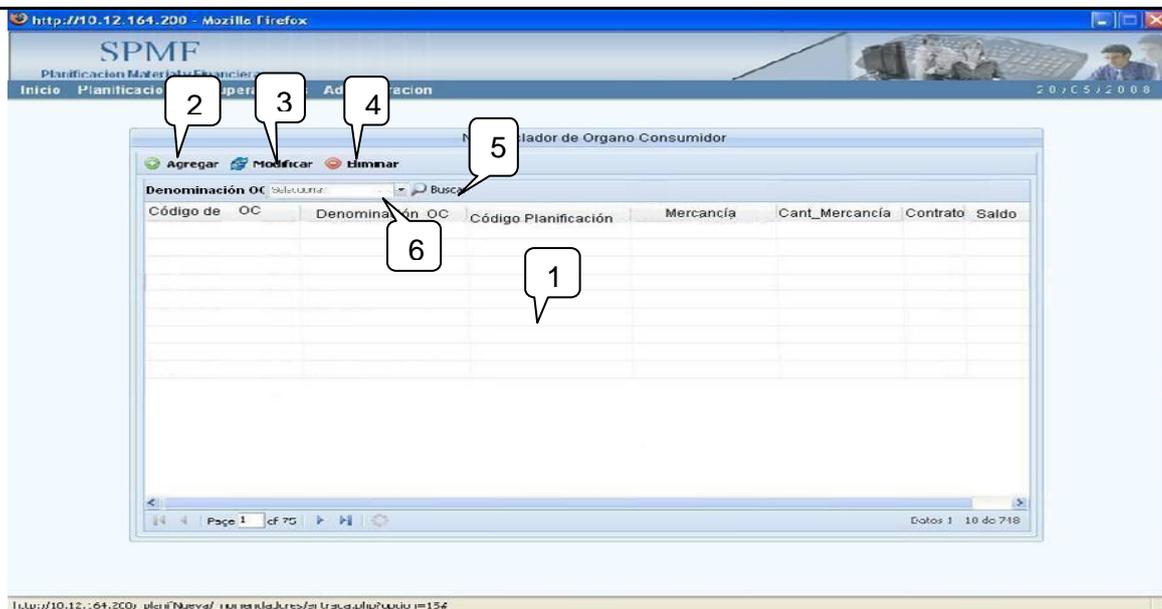
	Consumidores.
Resumen	El caso de uso se inicia cuando el Operador Oficial desea gestionar los órganos consumidores. El sistema muestra la interfaz de gestionar Órgano Consumidor, donde el Operador Oficial puede agregar un nuevo Órgano Consumidor, eliminar o modificar algunos datos ya existentes. Finalmente se archivan los datos en el sistema y se muestra el Órgano Consumidor actualizado. Terminando así el caso de uso.
Tipo	Real y expandido.
Precondiciones.	El actor se debe encontrar trabajando con el sistema y tener acceso a gestionar Órgano Consumidor.
Referencias	RF.6, 6.1, 6.2, 6.3, 6.4
Casos de Uso relacionados:	

Interfaz I

Ver Interfaz I de la Tabla 7 Descripción expandida del Caso de Uso: "Gestionar Planificación "

(1) Submenú para seleccionar la acción que desea realizar.

Interfaz II



(1) Grid para mostrar los datos (si existen) de los Órganos Consumidores, inicialmente vacío, hasta que se inserte por primera vez un Órgano Consumidor. **Nombre:** GridDatos, **tipo:** objeto

- (2) Botón para Agregar Órgano Consumidor. **Nombre:** Agregar, **tipo:** button
- (3) Botón para Modificar Órgano Consumidor. **Nombre:** Modificar, **tipo:** button
- (4) Botón para Eliminar Órgano Consumidor. **Nombre:** Eliminar, **tipo:** button
- (5) Botón para Buscar Órgano Consumidor. **Nombre:** Buscar, **tipo:** button
- (6)Textfield para buscar la denominación del Órgano Consumidor.**Nombre:** den_oc,**tipo** :string

Flujo normal de eventos

Acción del actor	Respuesta del sistema
<p>1. El actor selecciona la opción que desea realizar sobre los Órganos Consumidores.</p> <ul style="list-style-type: none"> • Agregar Órgano Consumidor. • Modificar Órgano Consumidor. • Eliminar Órgano Consumidor. • Mostrar Órgano Consumidor. 	<p>2. El sistema realiza las operaciones según haya sido la selección del actor.</p> <ul style="list-style-type: none"> • Agregar Órgano Consumidor (2) (ver sección Insertar). • Modificar Órgano Consumidor (3) (ver sección Modificar). • Eliminar Órgano Consumidor (4) (ver sección Eliminar). • Mostrar órgano consumidor (ver sección Mostrar)

Sección Insertar

Interfaz III



- 1) Textfield para mostrar el código del Órgano Consumidor.**Nombre:**cod_oc,**tipo:** string
- (2) Textfield para insertar el código del contrato.**Nombre:**cod_contrato,**tipo:**string
- (3) ComboBox para seleccionar la denominación de la mercancía. **Nombre:** cod_merca, **tipo:** string.
- (4) extfield para cargar el código de la planificación .**Nombre:** cod_plani, **tipo:** string.
- (5) Textfield para insertar la cantidad de mercancía. **Nombre:** cant_merca, **tipo:** double.

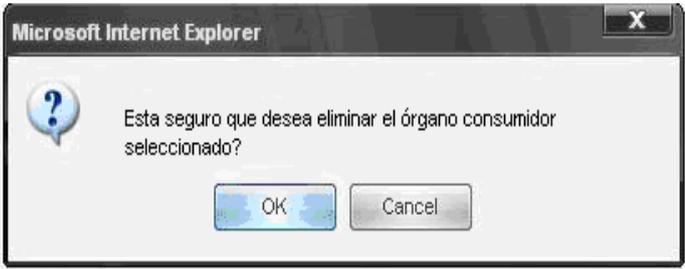
DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

<p>(6) Textfield para insertar el saldo asignado. Nombre: saldo, tipo: double</p> <p>(7) ComboBox seleccionar Denominación. Nombre: deno, tipo: string.</p> <p>(8) ComboBox seleccionar Sector o Rama. Nombre: Sector, tipo: string.</p> <p>(9) Botón Aceptar. Nombre: Aceptar, tipo: button</p> <p>(10) Botón Cancelar la operación. Nombre: Cancelar, tipo: button.</p>	
Acción del actor	Respuesta del sistema
1. El actor selecciona la opción <Insertar>.	2. El sistema le muestra el formulario (Interfaz III) y las opciones Aceptar y Cancelar activas.
<p>3. El actor introduce los datos del órgano consumidor por teclado. En la opción:</p> <p>(1) Código de OC.</p> <p>(2) Denominación OC.</p> <p>(3) Denominación Mercancía.</p> <p>(4) Cantidad Mercancía Asignada</p> <p>(5) Código de Contrato.</p> <p>(6) Saldo Asignado por Contrato</p> <p>(7) Denominación OC</p> <p>(8) Sector o Rama</p>	
4. El actor selecciona <Aceptar> Interfaz III (9).	<p>5. El sistema verifica que los datos introducidos sean correctos. Guarda los cambios y envía mensaje al actor.</p> 
6. El actor selecciona la opción <OK>.	
Sección Modificar	
Interfaz IV	



- 1) Textfield para mostrar el código del Órgano Consumidor. **Nombre:** cod_oc, **tipo:** string
- (2) Textfield para insertar el código del contrato. **Nombre:** cod_contrato, **tipo:** string
- (3) ComboBox para seleccionar la denominación de la mercancía. **Nombre:** cod_merca, **tipo:** string.
- (4) Textfield para cargar el código de la planificación. **Nombre:** cod_plani, **tipo:** string.
- (5) Textfield para insertar la cantidad de mercancía. **Nombre:** cant_merca, **tipo:** double.
- (6) Textfield para insertar el saldo asignado. **Nombre:** saldo, **tipo:** double
- (7) ComboBox seleccionar Denominación. **Nombre:** deno, **tipo:** string.
- (8) ComboBox seleccionar Sector o Rama. **Nombre:** Sector, **tipo:** string.
- (9) Botón Aceptar. **Nombre:** Aceptar, **tipo:** button
- (10) Botón Cancelar la operación. **Nombre:** Cancelar, **tipo:** button.

Acción del actor	Respuesta del sistema
1. El actor selecciona la denominación del Órgano Consumidor que desea modificar y selecciona la opción buscar.	2. El sistema busca y muestra el órgano consumidor en el Grid de la Interfaz II
3. El actor selecciona el Órgano Consumidor y la opción <Modificar> en la Interfaz II (3).	4. El sistema muestra en la Interfaz IV los datos del Órgano Consumidor cargados y las opciones Aceptar y Cancelar activas.
5. El actor realiza los cambios necesarios por teclado y selecciona la opción < Aceptar> (9).	6. El sistema verifica que los datos modificados sean correctos. Guarda los cambios y envía mensaje al actor. 
7. El actor selecciona la opción <OK>.	

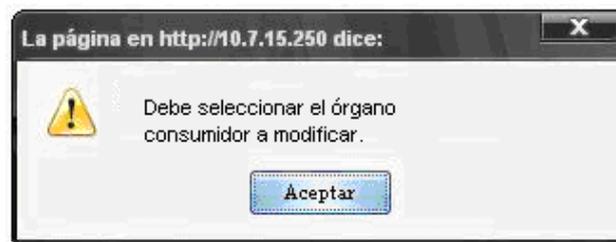
Sección Eliminar	
Acción del actor	Respuesta del sistema
1. El actor selecciona la denominación del Órgano Consumidor que desea eliminar.	2. El sistema verifica los datos introducidos y busca el Órgano Consumidor y lo muestra en el Grid de la Interfaz II .
3. El actor selecciona el Órgano Consumidor y la opción <Eliminar> Interfaz II (4).	4. El sistema pide confirmación al actor 
5. El actor responde "OK"	6. El sistema elimina el Órgano Consumidor señalado por el usuario y guarda los cambios
Sección Mostrar	
Acción del actor	Respuesta del sistema
1. El actor desea mostrar un Órgano Consumidor específico, selecciona la denominación del Órgano Consumidor y selecciona la opción <Buscar> Interfaz II (5).	2. El sistema busca y le muestra el Órgano Consumidor visualizado en el Grid (1) de la Interfaz II .
Cursos Alternos	
Sección Insertar.	
<p>Línea 3. En caso que el actor no introduce los datos del Órgano Consumidor y oprima <Aceptar> Interfaz III (9), el sistema envía mensaje al actor.</p> 	
<p>Línea 6. Si el actor introduce un Órgano Consumidor que ya existe, el sistema envía mensaje al actor.</p>	



Línea 5. Si el actor decide <Cancelar> el sistema no registra los cambios y el sistema regresa a la interfaz inicial.

Sección Modificar.

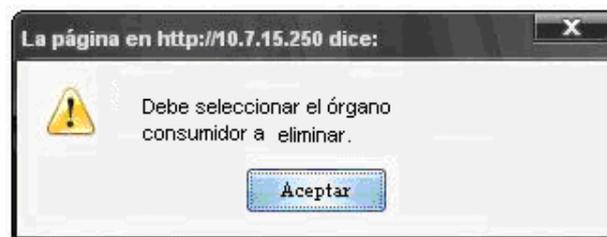
Línea 3. Si el actor no ha seleccionado el Órgano Consumidor a modificar y selecciona <Modificar> el sistema le envía un mensaje.



Línea 5. Si el actor decide <Cancelar> el sistema regresa a la interfaz inicial.

Sección Eliminar.

Línea 3. Si el especialista no ha seleccionado Órgano Consumidor a eliminar y selecciona <Eliminar> el sistema le envía un mensaje.



Línea 5. Si el actor decide <Cancelar> el sistema regresa a la interfaz inicial.

Sección Mostrar.

Línea 1. Si el actor introduce el código del Órgano Consumidor y no es encontrado el sistema le envía un mensaje.

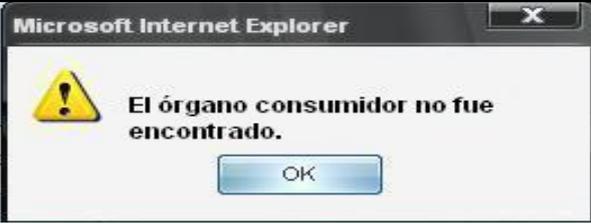
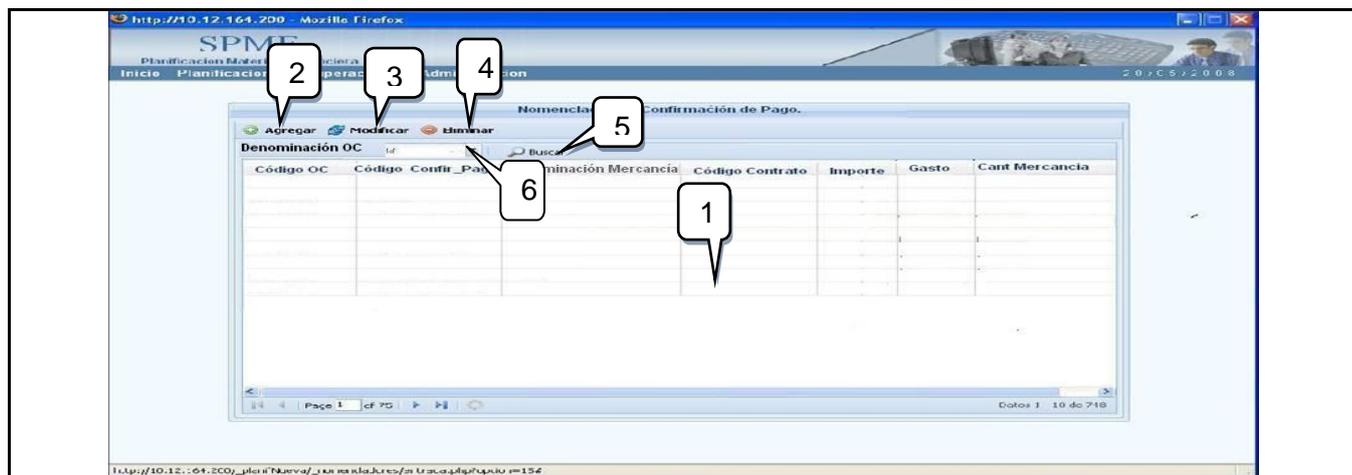
	
Pos-condiciones	Los Órganos Consumidores quedan actualizados.
Prioridad	Crítico

Tabla 10. Descripción Textual del caso de uso “Gestionar Órgano Consumidor”.

Caso de Uso	Gestionar Confirmación Pago.
Actores	Operador Oficial
Propósito	El propósito de este caso de uso es que el Operador Oficial pueda realizar actualizaciones y mostrar las confirmaciones de pago.
Resumen	El caso de uso se inicia cuando el Operador Oficial desea gestionar confirmaciones de pago. El sistema muestra la interfaz de gestionar confirmaciones de pago, donde el Operador Oficial puede agregar una nueva confirmación de pago, eliminar, modificar o mostrar algunos datos ya existentes. Terminando así el caso de uso.
Tipo	Real y expandido .
Precondiciones.	El actor se debe encontrar trabajando con el sistema y tener acceso a gestionar confirmación de pago.
Referencias	RF.5, 5.1, 5.2, 5.3, 5.4
Casos de Uso relacionados:	
Interfaz I	
Ver Interfaz I de la Tabla 7 Descripción expandida del Caso de Uso:”Gestionar Planificación ”	
(1) Submenú para seleccionar la acción que desea realizar.	
Interfaz II	



- (1) Grid para mostrar los datos (si existen) de la Distribución de Recursos, inicialmente vacío, hasta que se inserte por primera vez una confirmación de pago. **Nombre:** GridDatos, **tipo:** objeto
- (2) Botón para Agregar confirmación de pago. **Nombre:** Agregar, **tipo:** button
- (3) Botón para Modificar confirmación de pago. **Nombre:** Modificar, **tipo:** button
- (4) Botón para Eliminar confirmación de pago. **Nombre:** Eliminar, **tipo:** button
- (5) Botón para Buscar confirmación de pago. **Nombre:** Buscar, **tipo:** button
- (6) ComboBox para seleccionar denominación del Órgano Consumidor. **Nombre:** codoc, **tipo:** string

Flujo normal de eventos

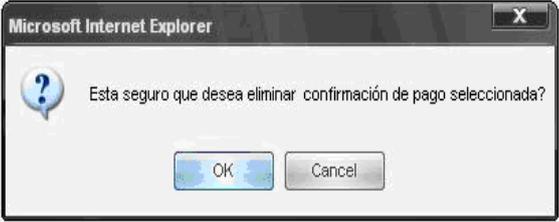
Acción del actor	Respuesta del sistema
<p>1. El actor desea gestionar el documento de confirmación de pago y selecciona la opción <Gestionar Confirmación de Pago > del menú principal Interfaz I</p>	<p>2. El sistema visualiza la Interfaz II.</p>
<p>3. El actor selecciona la opción que desea realizar sobre la confirmación de pago.</p> <ul style="list-style-type: none"> • Agregar confirmación pago. • Modificar confirmación pago. • Eliminar confirmación pago. • Mostrar confirmación pago. 	<p>4. El sistema realiza las operaciones según haya sido la selección del actor.</p> <ul style="list-style-type: none"> • Agregar confirmación pago (2) (ver sección Insertar). • Modificar confirmación pago (3) (ver sección Modificar). • Eliminar confirmación pago (4) (ver sección Eliminar). • Mostrar confirmación pago (ver sección Mostrar)

Sección Insertar	
Interfaz III	
	
<p>(1) Textfield para cargar el Código del Órgano Consumidor. Nombre: cod_oc, tipo: string.</p> <p>(2) ComboBox para seleccionar la denominación del Órgano Consumidor. Nombre: den_oc, tipo: string</p> <p>(3) Textfield para Insertar Importe de Confirmación de Pago. Nombre: importe, tipo: double</p> <p>(4) Textfield Insertar Cantidad de Mercancía adquirida. Nombre: cant_mercancia, tipo: string</p> <p>(5) ComboBox para seleccionar denominación Mercancía. Nombre: den_mercancia, tipo: entero</p> <p>(6) Textfield Insertar Gasto Total de Confirmación de Pago. Nombre: gasto_total, tipo: double.</p> <p>(7) Textfield insertar Código de Contrato. Nombre: cod_contrato, tipo: string</p> <p>(8) Botón Aceptar. Nombre: Aceptar, tipo: button</p> <p>(9) Botón Cancelar la operación. Nombre: Cancelar, tipo: button.</p>	
Acción del actor	Respuesta del sistema
<p>1.El actor selecciona la opción <Agregar> de la Interfaz II (2)</p>	<p>2. El sistema le muestra el formulario (Interfaz III) y las opciones Aceptar y Cancelar activas.</p>
<p>3. El actor introduce los datos de la confirmación pago. En la opción:</p> <p>(1) Importe de Confirmación Pago.</p> <p>(2) Denominación de OC.</p> <p>(3) Cantidad de Mercancía.</p> <p>(4) Denominación de Mercancía</p> <p>(5) Gasto total</p> <p>(6) Código Contrato.</p>	

DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

4. El actor selecciona <Aceptar> Interfaz III (8).	5. El sistema verifica que los datos introducidos sean correctos, introduce la confirmación de pago
	6. El sistema liquida los recursos correspondientes a contratos y mercancías y envía mensaje al actor. 
7. El actor selecciona la opción <OK>.	
Sección Modificar	
Interfaz IV	
	
<p>(1) Textfield para cargar el Código del Órgano Consumidor. Nombre: cod_oc, tipo: string.</p> <p>(2) ComboBox para seleccionar la denominación del Órgano Consumidor. Nombre: den_oc, tipo: string</p> <p>(3) Textfield para Insertar Importe de Confirmación de Pago. Nombre: importe, tipo: double</p> <p>(4) Textfield Insertar Cantidad de Mercancía adquirida. Nombre: cant_mercancia, tipo: string</p> <p>(5) ComboBox para seleccionar denominación Mercancía. Nombre: den_mercancia, tipo: entero</p> <p>(6) Textfield Insertar Gasto Total de Confirmación de Pago. Nombre: gasto_total, tipo: double</p> <p>(7) Textfield insertar Código de Contrato. Nombre: cod_contrato, tipo: string</p> <p>(8) Botón Aceptar. Nombre: Aceptar, tipo: button</p> <p>(9) Botón Cancelar la operación. Nombre: Cancelar, tipo: button.</p>	
Acción del actor	Respuesta del sistema
1. El actor busca por la denominación del Órgano	2. El sistema muestra en el Grid de la

DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

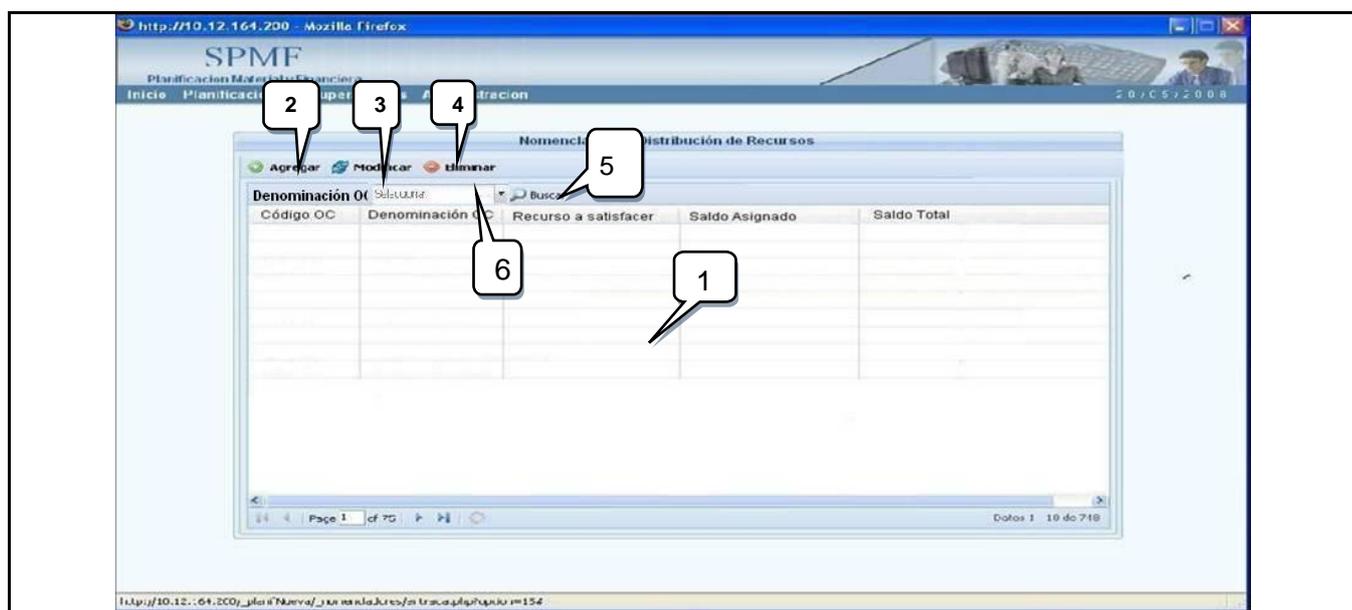
Consumidor la confirmación de pago.	Interfaz II la confirmación de pago.
3. El actor selecciona la confirmación de pago y la opción <Modificar> en la Interfaz II (3).	4. El sistema muestra los datos cargados de la confirmación de pago en la Interfaz IV y las opciones Aceptar y Cancelar activas.
5. El actor realiza los cambios necesarios por teclado, selecciona la opción <Aceptar>.	6. El sistema verifica que los datos modificados sean correctos. Guarda los cambios y envía mensaje al actor. 
7. El actor selecciona la opción <OK>.	
Sección Eliminar	
Acción del actor	Respuesta del sistema
1. El actor busca por la denominación del Órgano Consumidor la confirmación de pago.	2. El sistema busca y muestra la confirmación de pago en el Grid de la Interfaz II (1).
3. El actor selecciona la confirmación de pago que desea eliminar y la opción <Eliminar> en la Interfaz II (4).	4. El sistema pide confirmación para eliminar. 
5. El actor selecciona la opción "OK".	6. El sistema elimina la confirmación de pago señalado por el actor y guarda los cambios.
Sección Mostrar	
Acción del actor	Respuesta del sistema
1. El actor desea mostrar una confirmación de pago para un Órgano Consumidor específico, selecciona la	2. El sistema busca y le muestra la confirmación de pago en el Grid (1) de la

DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

denominación del Órgano Consumidor y la opción <Buscar> Interfaz II (5).	Interfaz II.
Pos-condiciones	El documento de confirmación de pago queda actualizado.
Prioridad	Crítico

Tabla 11. Descripción Textual del caso de uso “Gestionar Confirmación de Pago”.

Caso de Uso	Gestionar Distribución de Recursos.
Actores	Jefe del Departamento de Economía.
Propósito	El propósito de este caso de uso es que el actor pueda realizar actualizaciones y mostrar la distribución de recursos.
Resumen	El caso de uso se inicia cuando el actor desea gestionar la distribución de recursos. El sistema muestra la interfaz de gestionar distribución de recursos, donde el Operador Oficial puede agregar una nueva distribución de recursos, eliminar o modificar algunos datos ya existentes. Finalmente se archivan los datos en el sistema y se muestra la distribución de recursos actualizada. Terminando así el caso de uso.
Tipo	Real y expandido.
Precondiciones.	El actor se debe encontrar trabajando con el sistema y tener acceso a gestionar distribución de recursos.
Referencias	RF.2, 2.1, 2.2, 2.3, 2.4
Casos de Uso relacionados:	
Interfaz I	
Ver Interfaz I de la Tabla 7 Descripción expandida del Caso de Uso:”Gestionar Planificación ”	
(1) Submenú para seleccionar la acción que desea realizar.	
Interfaz II	



- (1) Grid para mostrar los datos (si existen) de la Distribución de Recursos, inicialmente vacío, hasta que se inserte por primera vez una distribución de recursos. **Nombre:** GridDatos, **tipo:** objeto
- (2) Botón para Agregar distribución de recursos. **Nombre:** Agregar, **tipo:** button
- (3) Botón para Modificar distribución de recursos. **Nombre:** Modificar, **tipo:** button
- (4) Botón para Eliminar distribución de recursos. **Nombre:** Eliminar, **tipo:** button
- (5) Botón para Buscar distribución de recursos. **Nombre:** Buscar, **tipo:** button
- (6) ComboBox seleccionar denominación del Órgano Consumidor a buscar. **Nombre:** den_oc, **tipo:** string.

Flujo normal de eventos

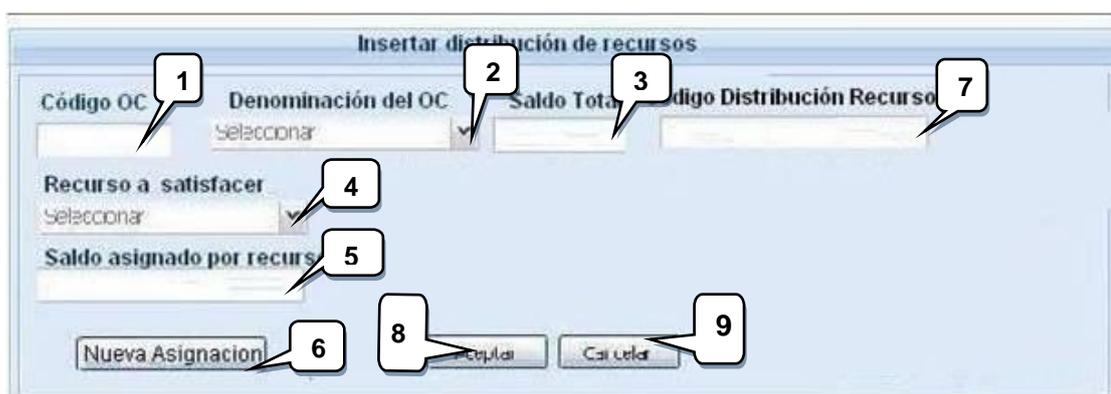
Acción del actor	Respuesta del sistema
1.El actor desea gestionar la distribución de recursos y selecciona la opción <Gestionar Distribución de Recursos> del menú principal Interfaz I	2. El sistema visualiza la Interfaz I .
3. El actor selecciona la opción que desea realizar sobre la distribución de recurso. <ul style="list-style-type: none"> • Agregar distribución de recurso. • Modificar distribución de recurso. 	4. El sistema realiza las operaciones según haya sido la selección del actor. <ul style="list-style-type: none"> • Agregar distribución de recurso (2) (ver sección Insertar). • Modificar distribución de recurso (3) (ver sección Modificar).

- Eliminar distribución de recurso.
- Mostrar distribución recurso

- Eliminar distribución de recurso (4) (ver sección Eliminar).
- Mostrar distribución de recursos (ver sección Mostrar).

Sección Insertar

Interfaz III



- (1) Textfield para cargar el Código del Órgano Consumidor. **Nombre:** cod_oc, **tipo:** string.
- (2) ComboBox para seleccionar la Denominación del Órgano Consumidor. **Nombre:** den_oc, **tipo:** string
- (3) Textfield Insertar Saldo Total. **Nombre:** saldo_t, **tipo:** entero
- (4) Textfield seleccionar Recurso a Satisfacer. **Nombre:** recur_sat, **tipo:** string
- (5) Textfield Insertar Saldo Asignado por recurso. **Nombre:** sal_asig, **tipo:** entero
- (6) Botón Nueva Asignación: generaría nuevos campos para insertar otro(s) recurso(s) a satisfacer. **Nombre:** Nueva Asignación, **tipo:** button
- (7) Textfield para cargar Código de Distribución de Recursos. **Nombre:** cod_distrib, **tipo:** string
- (8) Botón Aceptar. **Nombre:** Aceptar, **tipo:** button
- (9) Botón Cancelar la operación. **Nombre:** Cancelar, **tipo:** button.

Acción del actor	Respuesta del sistema
1. El actor selecciona la opción <Agregar> en la Interfaz II(6).	2. El sistema le muestra el formulario (Interfaz III) y las opciones Aceptar y Cancelar activas.
3. El actor desea agregar un nuevo recursos a satisfacer y su saldo asignado y presiona la opción <Nueva Asignación>.	4. El sistema muestra actualizada la Interfaz III agregando dos nuevos cada vez que el actor seleccione <Nueva Asignación>.

Asignación> (6).	
<p>5. El actor introduce los datos de la distribución de recurso por teclado. En la opción:</p> <p>(1) Código de OC.</p> <p>(2) Denominación del Órgano</p> <p>(3) Recurso(s) a Satisfacer</p> <p>(4) Saldo asignado por recurso.</p> <p>(5) saldo total</p> <p>(7)Código de Distribución de Recursos.</p> <p>(9)Código de Planificación</p>	
6. El actor selecciona la opción <Aceptar>.	7. El sistema verifica que los datos introducidos sean correctos. Inserta la nueva distribución de recursos y envía mensaje al actor.
8. El actor selecciona la opción <OK>.	

Sección Modificar

Interfaz IV

The screenshot shows a window titled "Modificar Distribución de Recursos". It contains several input fields and buttons. Callouts 1 through 9 point to the following elements:

- 1: Textfield for "Código OC" (value: 011A)
- 2: ComboBox for "Denominación del OC" (value: DataFAR)
- 3: Textfield for "Saldo Total" (value: 1000)
- 4: Textfield for "Recurso a satisfacer" (value: Equipos)
- 4: Textfield for "Recurso a Satisfacer" (value: PPA)
- 5: Textfield for "Saldo asignado por recurso" (value: 800)
- 5: Textfield for "Saldo Asignado por recurso" (value: 200)
- 6: "Nueva Asignación" button
- 8: "Aceptar" button
- 9: "Cancelar" button
- 7: Textfield for "Código Distribución Recursos" (value: A23)

- (1) Textfield para cargar el Código del Órgano Consumidor. **Nombre:** cod_oc, **tipo:** string.
- (2) ComboBox para seleccionar la Denominación del Órgano Consumidor. **Nombre:** den_oc, **tipo:** string
- (3) Textfield Insertar Saldo Total. **Nombre:** saldo_t, **tipo:** entero
- (4) Textfield Seleccionar Recurso a Satisfacer. **Nombre:** recur_sat, **tipo:** string
- (5) Textfield Insertar Saldo Asignado por recurso. **Nombre:** sal_asig, **tipo:** entero

<p>(6) Botón Nueva Asignación: generaría nuevos campos para insertar otro(s) recurso(s) a satisfacer. Nombre: Nueva Asignación, tipo: button</p> <p>(7) Textfield para cargar Código de Distribución de Recursos.Nombre:cod_distrib,tipo:string</p> <p>(8) Botón Aceptar. Nombre: Aceptar, tipo: button</p> <p>(9) Botón Cancelar la operación. Nombre: Cancelar, tipo: button.</p>	
Acción del actor	Respuesta del sistema
1. El actor busca por la denominación del Órgano Consumidor la distribución de recursos.	2. El sistema verifica el dato introducido, busca y muestra en el Grid de la Interfaz II la distribución de recursos.
3. El actor selecciona la distribución de recursos y la opción <Modificar> en la Interfaz II (3).	4. El sistema muestra los datos cargados de la distribución de recursos en la Interfaz IV y las opciones Aceptar y Cancelar activas.
5. El actor modifica los datos deseados y selecciona la opción <Aceptar>.	6. El sistema verifica que los datos modificados sean correctos. El sistema guarda los cambios y envía mensaje al actor. 
7. El actor selecciona la opción <OK>.	
Sección Eliminar	
Acción del actor	Respuesta del sistema
1. El actor busca por la denominación del Órgano Consumidor la distribución de recursos que desea eliminar.	2. El sistema busca la distribución de recursos y la muestra en el Grid de la Interfaz II (1).
3. El actor selecciona la distribución de recursos y la opción <Eliminar> en la Interfaz II (4).	4. El sistema pide confirmación al actor.
5. El actor responde <OK>.	6. El sistema elimina la distribución de recursos.
Sección Mostrar	
Acción del actor	Respuesta del sistema

1. El actor desea mostrar una distribución de recursos de un Órgano Consumidor específico, selecciona la denominación del Órgano Consumidor y la opción <Buscar> Interfaz II (5).	2. El sistema busca la distribución de recursos y la muestra marcada en el Grid de la Interfaz II.
--	---

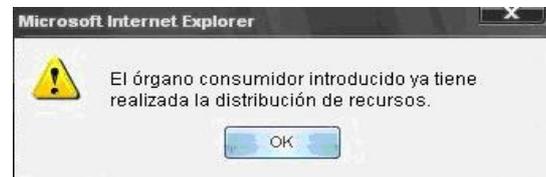
Cursos Alternos

Sección Insertar.

Línea 4. En caso que el actor introduce los datos incorrectamente y oprime <Aceptar> **Interfaz III (8)**, el sistema envía mensaje al actor.



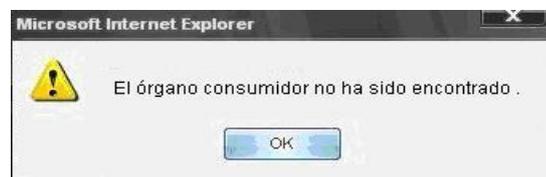
Línea 4. Si el actor introduce un órgano consumidor que tiene realizada la distribución de recursos, el sistema envía mensaje al actor.



Línea 3. Si el actor decide <Cancelar> el sistema no registra los cambios y el sistema regresa a la interfaz inicial.

Sección Modificar.

Línea 2. Si el Órgano Consumidor no es encontrado, el sistema envía un mensaje al actor.



Línea3. Si el actor decide <Cancelar> el sistema regresa a la interfaz inicial.

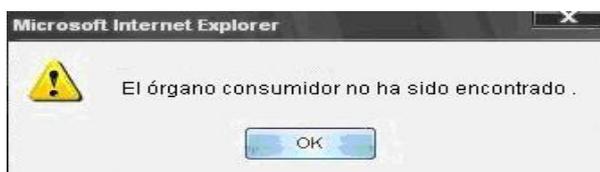
Línea 4. Si el actor introduce los datos incorrectamente el sistema le envía un mensaje al actor.



Sección Eliminar.

Línea 3. Si el actor decide <Cancelar> el sistema no registra los cambios y el sistema regresa a la interfaz inicial.

Línea 2. Si el Órgano Consumidor no es encontrado, el sistema envía un mensaje al actor.



Línea 3. Si el actor decide <Cancelar> el sistema regresa a la interfaz inicial.

Pos-condiciones	El documento de confirmación de pago queda actualizado.
Prioridad	Crítico

Tabla 12. Descripción Textual del caso de uso “Gestionar Distribución de Recurso”.

Nombre del caso de uso	Seleccionar Criterios de Búsqueda. (Include)
Actores	Jefe del Departamento de Economía.
Propósito	El propósito de este caso de uso es que el actor pueda seleccionar los diferentes criterios por los cuales desea generar un reporte determinado.
Resumen	El caso de uso se inicia cuando el actor desea seleccionar algunos criterios para hacer más específica una recuperación. El caso de uso termina cuando el actor oprime el botón aceptar.
Tipo	Real y expandido.
Referencias	RF8
Caso de Uso relacionado:	
Interfaz I	



- (1) Textfield para cargar el código del OC. **Nombre:** cod_oc, **tipo:** string
- (2) ComboBox para seleccionar la Denominación de OC. **Nombre:** den_oc, **tipo:** string.
- (3) ComboBox para seleccionar el tipo de reporte. **Nombre:** tipo_reporte, **tipo:** string.
- (4) Textfield para cargar el código del tipo seleccionado. **Nombre:** cod_contrato, **tipo:** string..
- (5) Botón Aceptar para filtrar los datos. **Nombre:** Aceptar, **tipo:** button.
- (6) Botón Cancelar para cancelar la operación. **Nombre:** Cancelar, **tipo:** button.

Flujo normal de eventos

Acción del actor	Respuesta del sistema
	1. El sistema muestra la Interfaz I con los diferentes criterios por los cuales se puede mostrar un reporte.
2. El actor selecciona los diferentes criterios de búsqueda por los que va a filtrar el reporte.	
3. El actor selecciona <Aceptar> (5).	4. El sistema verifica los datos introducidos y busca el o los reportes según criterios de búsqueda.

Cursos Alternos

Línea 3 En caso que el actor selecciona la opción <Cancelar> el sistema regresa a la interfaz inicial.

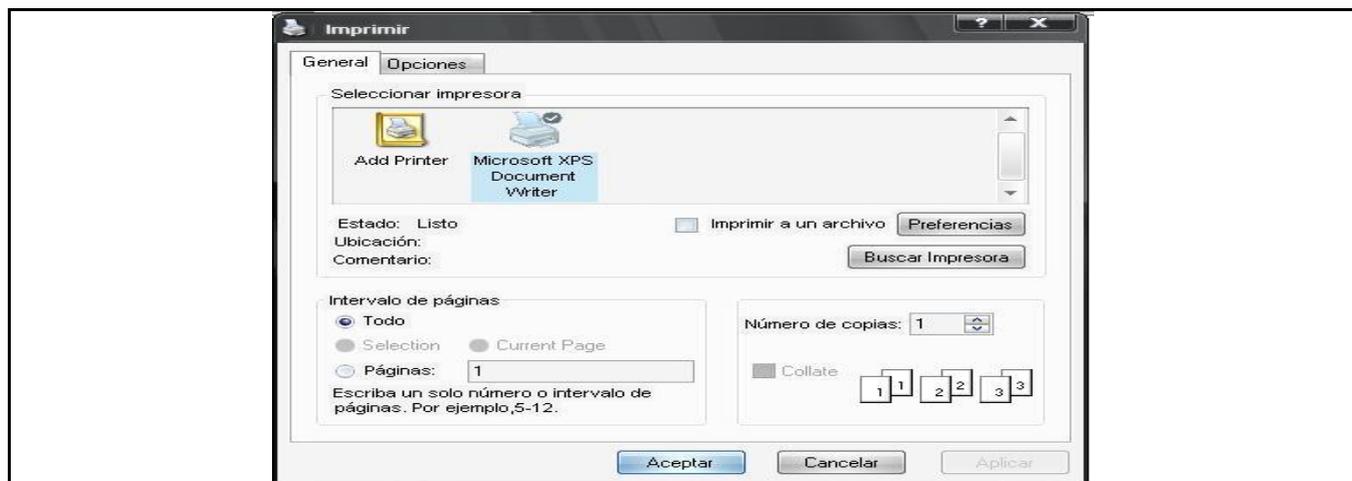
Pos-condiciones	El sistema el busca los criterios para buscar el reporte.
Prioridad	Secundario

Tabla 13. Descripción Textual del caso de uso “Seleccionar Criterios de Búsqueda.”.

Caso de Uso	Obtener Reporte (Caso de uso base generalizado)
Actores	Jefe del Departamento de Economía.
Propósito	El propósito de este caso de uso es que el actor pueda obtener reporte.

DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

Resumen	El caso de uso se inicia cuando el Caso Uso “Mostrar Reporte de Contrato”, “Mostrar Reporte de Mercancía”, “Mostrar Reporte de Planificación”, “Mostrar Reporte de Distribución de Recurso”, o “Mostrar Reporte de Confirmación de Pago”	
Tipo	Real y expandido	
Referencias	RF 7, 7.1, 7.2, 7.3, 7.4, 7.1.1, 7.1.2, 7.1.3, 7.1.4, 7.1.5	
Sección “ Seleccionar Reporte”		
Interfaz I		
		
(1) Submenú para seleccionar la acción que desea realizar.		
Flujo normal de eventos		
Acción del actor	Respuesta del sistema	
1. El actor desea obtener un reporte y selecciona la opción <Recuperaciones> del menú principal Interfaz I (1).	2. El sistema ejecuta el Caso de Uso: “Seleccionar Criterios de Búsqueda”	
Se invoca la sección “Visualizar Reporte” del caso de uso:		
<ul style="list-style-type: none"> -“Mostrar Reporte de Confirmación de Pago”. --“Mostrar Reporte de Distribución de Recursos”. - “Mostrar Reporte de Planificación “. - “Mostrar Reporte de Contrato “. - “Mostrar Reporte de Mercancía “. 		
Sección “ Imprimir”		
Interfaz II		



Flujo normal de eventos	
Acción del actor	Respuesta del sistema
1. El actor selecciona <Imprimir>.	2. El sistema le muestra la Interfaz II para que configure el reporte a imprimir.
3. El actor configura el reporte selecciona <Aplicar> y luego <Aceptar>.	4. El sistema imprime el reporte.
Cursos Alternos	
Sección Imprimir	
Línea 3. Si el actor decide <Cancelar> el sistema regresa a la interfaz inicial.	
Pos-condiciones	Reportes Conformados.

Tabla 14. Descripción Textual del caso de uso “Obtener Reporte.”

Caso de Uso	Mostrar Reporte de Planificación (Especialización)
Actores	Jefe del Departamento de Economía.
Propósito	El propósito de este caso de uso es que el actor pueda obtener un reporte de planificación.
Resumen	El caso de uso se inicia cuando el actor desea obtener un reporte de planificación para su análisis o impresión del documento generado. El sistema muestra una interfaz que le permite obtener el reporte de la planificación por diferentes criterios. Una vez efectuada la búsqueda muestra la planificación. El caso de uso termina cuando se muestra el reporte.
Precondiciones	El reporte debe quedar conformado.

Referencias	RF7, 7.2 , 7.2.1
Sección “Seleccionar Reporte”	
Se invoca a la sección: “Seleccionar Reporte” del caso de uso generalizado :“Obtener Reporte”	
Sección “ Visualizar Reporte”	
Interfaz I	
	
(1) Botón Imprimir. Nombre: Imprimir tipo: button (2) Grid para mostrar los datos de Reporte .Nombre: GridDatos, tipo: objeto	
Acción del actor	Respuesta del sistema
	1. El sistema visualiza el reporte de planificación.
2. El actor analiza el reporte. En caso que desee: Imprimir ir sección “Imprimir”.	
Sección “ Imprimir”	
Se invoca a la sección: “Imprimir” del caso de uso generalizado: “Obtener Reporte”.	
Pos-condiciones	Obtiene el reporte de planificación.
Prioridad	Secundario

Tabla 15. Descripción Textual del caso de uso “Mostrar Reporte de Planificación.”

Caso de Uso	Mostrar Reporte de Confirmación de pagos(Especialización)
Actores	Jefe del Departamento de Economía.
Propósito	El propósito de este caso de uso es que el actor pueda obtener un reporte de confirmación de pagos.
Resumen	El caso de uso se inicia cuando el actor desea obtener un reporte de

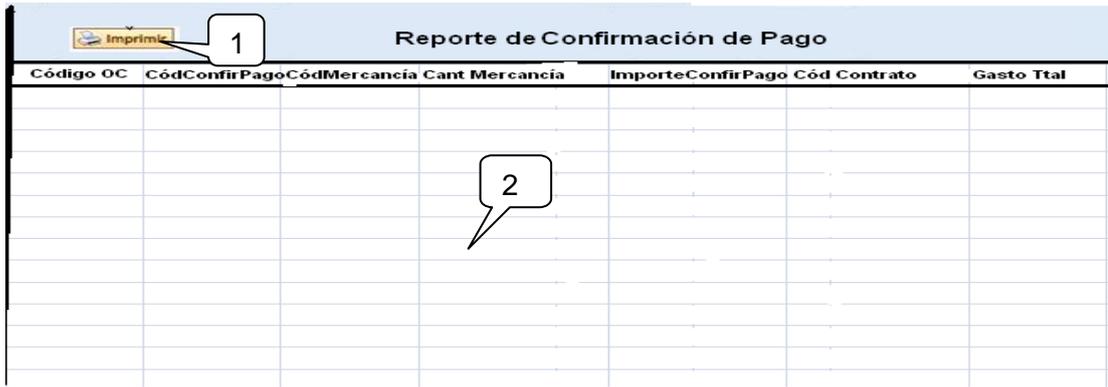
	confirmación de pagos para su análisis o impresión del documento generado. El sistema muestra una interfaz que le permite obtener el reporte de la confirmación de pagos por diferentes criterios. Una vez efectuada la búsqueda muestra la confirmación pago. El caso de uso termina cuando se muestra el reporte.
Precondiciones	El reporte debe quedar visualizado.
Referencias	RF 7.3 , 7.3.1
Sección “Seleccionar Reporte”	
Se invoca a la sección: “Seleccionar Reporte” del caso de uso generalizado :“Obtener Reporte”	
Sección “ Visualizar Reporte”	
Interfaz I	
	
(1) Botón Imprimir. Nombre: Imprimir tipo: button	
(2) Grid para mostrar los datos de Reporte .Nombre: GridDatos, tipo: objeto	
Acción del actor	Respuesta del sistema
	1. El sistema visualiza el reporte de la confirmación de pago.
2. El actor analiza el reporte. En caso que desee: Imprimir ir sección “Imprimir “.	
Sección “ Imprimir”	
Se invoca a la sección: “Imprimir” del caso de uso generalizado: “Obtener Reporte”.	
Pos-condiciones	Obtiene el reporte de confirmación de pago
Prioridad	Secundario

Tabla 16. Descripción Textual del caso de uso “Mostrar Reporte de Confirmación de pagos.”

DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

Caso de Uso	Mostrar Reporte de Distribución de Recursos. (Especialización)	
Actores	Jefe del Departamento de Economía.	
Propósito	El propósito de este caso de uso es que el actor pueda obtener un reporte de distribución de recursos.	
Resumen	El caso de uso se inicia cuando el actor desea obtener un reporte de distribución de recursos para su análisis o impresión del documento generado. El sistema muestra una interfaz que le permite obtener el reporte de la distribución de recursos por diferentes criterios. Una vez efectuada la búsqueda muestra la distribución de recursos. El caso de uso termina cuando se muestra el reporte.	
Precondiciones	El reporte debe quedar visualizado.	
Referencias	RF 7.1 ,7.1.1	
Flujo normal de eventos		
Acción del actor	Respuesta del sistema	
Sección “Seleccionar Reporte”		
Se invoca a la sección: “Seleccionar Reporte” del caso de uso generalizado :“Obtener Reporte”		
Sección “ Visualizar Reporte”		
Interfaz I		
		
(1) Botón Imprimir. Nombre: Imprimir tipo: button		
(2) Grid para mostrar los datos de Reporte .Nombre: GridDatos, tipo: objeto		
Acción del actor	Respuesta del sistema	
	1. El sistema visualiza el reporte de la distribución de recursos.	

2. El actor analiza el reporte. En caso que desee: Imprimir ir sección "Imprimir".	
Sección " Imprimir"	
Se invoca a la sección: "Imprimir" del caso de uso generalizado: "Obtener Reporte".	
Pos-condiciones	Obtiene el reporte de distribución de recursos.
Prioridad	Secundario

Tabla 17. Descripción Textual del caso de uso "Mostrar Reporte de Distribución de Recursos."

2.14 Conclusiones.

Hasta el momento se ha visto como a partir del análisis de los procesos del negocio y de los artefactos obtenidos en el modelamiento del mismo se identificaron los requerimientos, los cuales ayudarán a desarrollar las funcionalidades que el software a construir debe cumplir. Estas se representaron mediante el Diagrama de Casos de Uso y finalmente se describieron paso a paso todas las acciones de los actores del sistema con los casos de uso con los que interactúan. Todo esto constituye la base para empezar la construcción del sistema.

CAPÍTULO 3. ANÁLISIS Y DISEÑO DEL SISTEMA

3.1 Introducción.

En este capítulo se procede a la confección de la propuesta de solución, basándose en el análisis hecho anteriormente. Para el desarrollo de este capítulo se tuvieron en cuenta los requisitos funcionales, como no funcionales obtenidos durante el flujo de la captura de requisitos. El mismo constará con varios artefactos que tienen como objetivos mostrar el diseño y la implementación del sistema propuestos como solución.

3.2 Modelo de Análisis.

El modelo de análisis es la primera representación técnica del sistema, por lo que puede considerarse como una primera aproximación al modelo de diseño, y es por tanto una entrada fundamental para las actividades de diseño e implementación subsiguientes. A partir de su realización se refinan y estructuran los requisitos obtenidos con anterioridad proporcionando una visión general del sistema.

El modelo de análisis debe lograr tres objetivos primarios:

- Describir lo que requiere el cliente.
- Establecer una base para la creación de un diseño de software.
- Definir un conjunto de requisitos que se pueda validar una vez que se construye el software.

A continuación se muestran los diagramas de clases del análisis correspondiente a cada uno de los casos de usos que se describieron en el capítulo anterior.

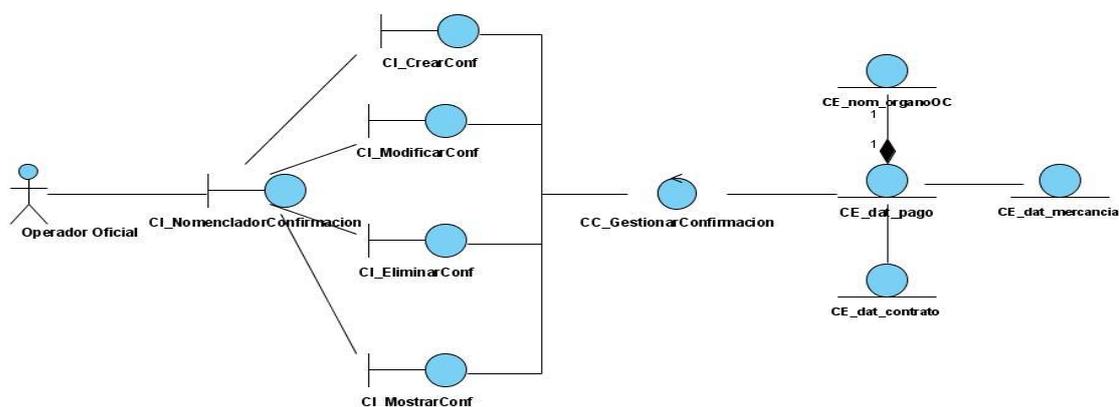


Figura 9. Diagrama de clases del análisis “Gestionar Confirmación de Pagos”.

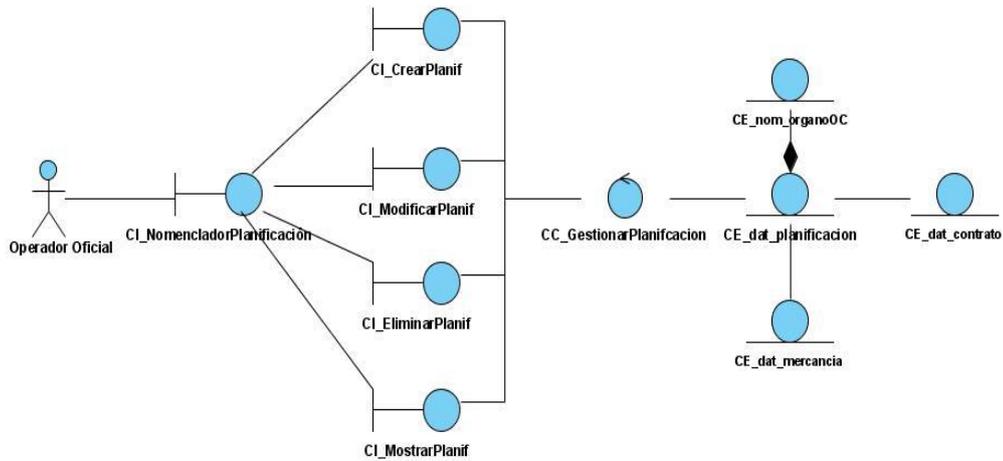


Figura 10. Diagrama de clases del análisis “Gestionar Planificación”.

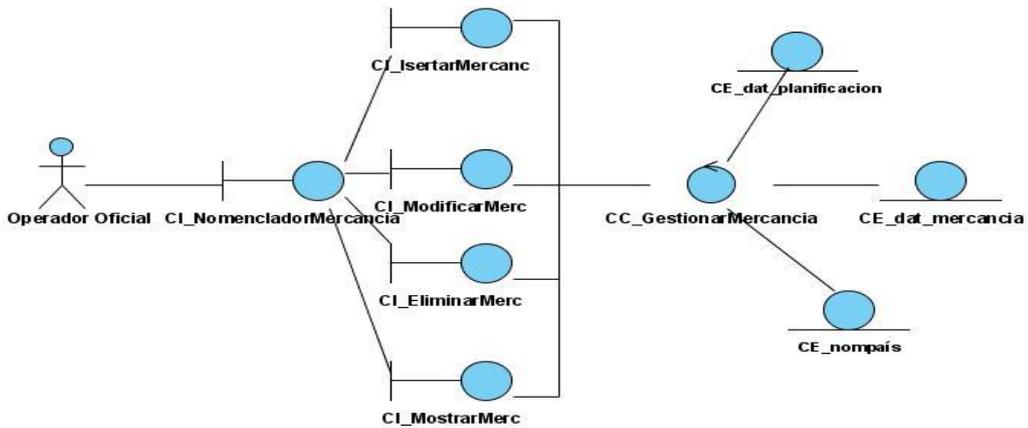


Figura 11. Diagrama de clases del análisis “Gestionar Mercancía”.

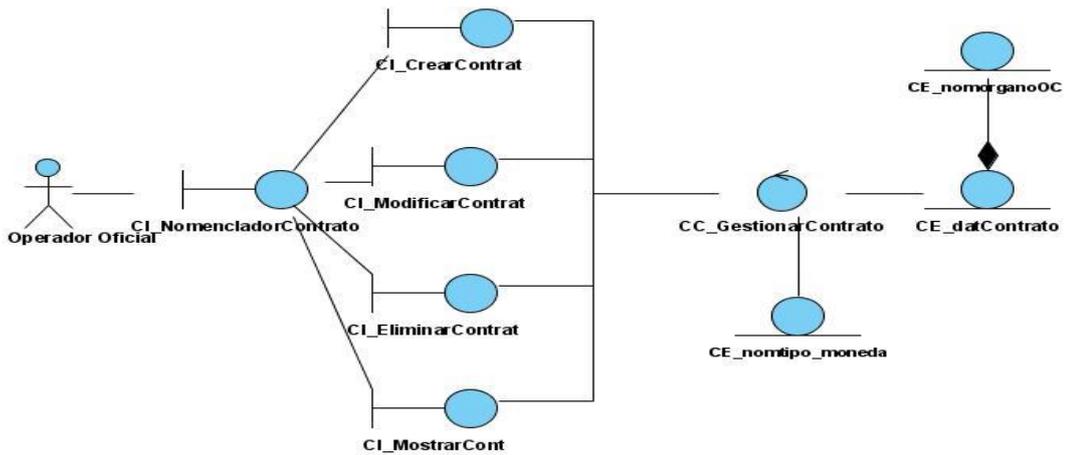


Figura 12. Diagrama de clases del análisis “Gestionar Contrato”.

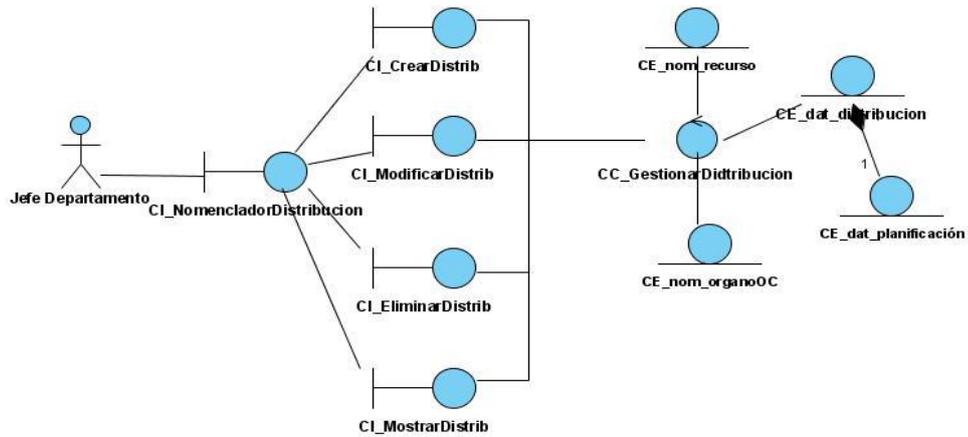


Figura 13. Diagrama de clases del análisis “Gestionar Distribución de Recursos”.

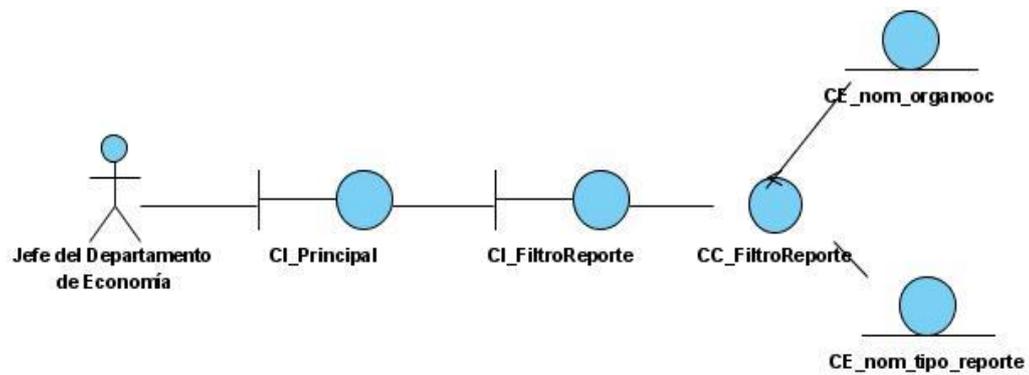


Figura 14. Diagrama de clases del análisis “Seleccionar Criterios de Búsqueda”.

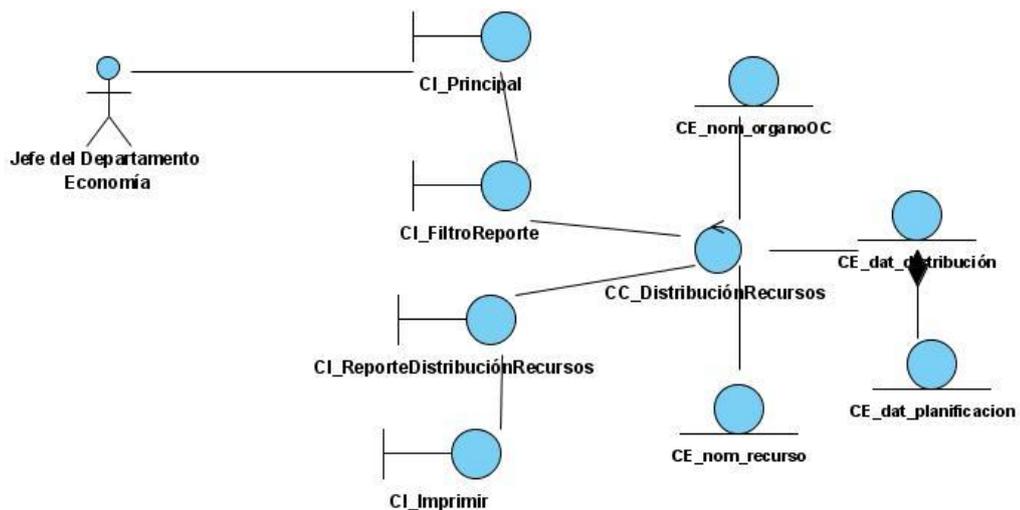


Figura 15. Diagrama de clases del análisis “Mostrar Reporte de Distribución Recursos”.

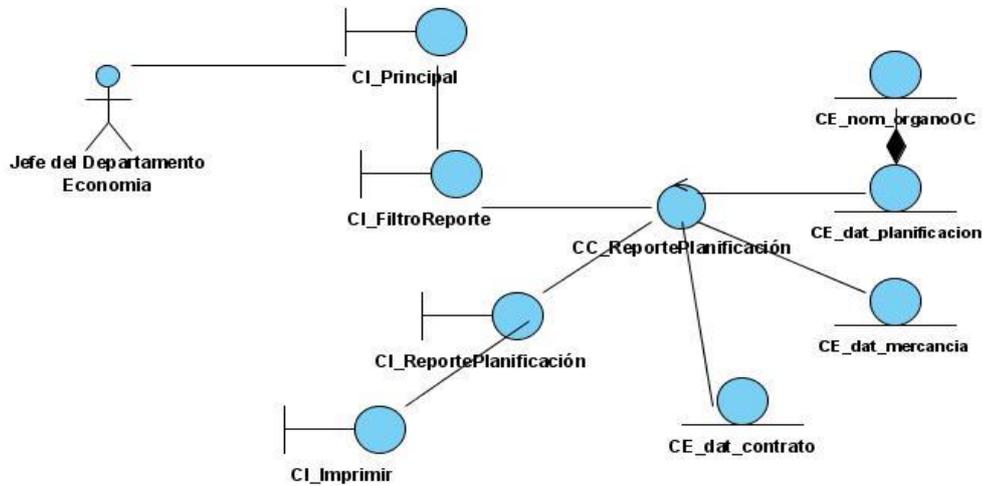


Figura 16. Diagrama de clases del análisis “Mostar Reporte de Planificación”.

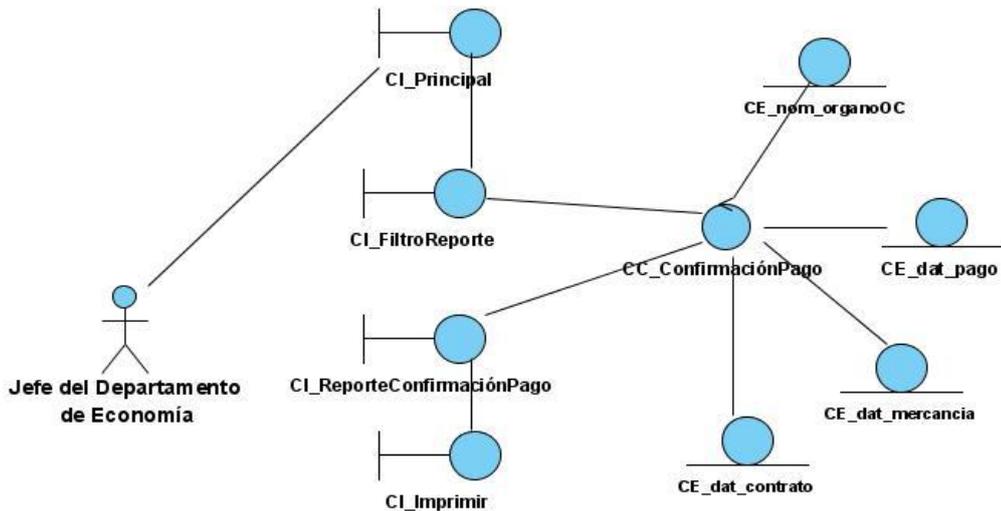


Figura 17. Diagrama de clases del análisis “Mostar Reporte de Confirmación Pago”.

3.3 Modelo de Diseño.

El diseño constituye el centro de atención de la propuesta de solución, el mismo contribuye a una arquitectura estable y sólida. En el diseño se modela el sistema y de forma incluida la arquitectura para que soporte todos los requisitos, incluyendo los no funcionales y las restricciones que se le suponen. Una entrada esencial en el diseño es el resultado del análisis, o sea el modelo de análisis, que proporciona una comprensión detallada de los requisitos.

3.4 Arquitectura.

La arquitectura es el esqueleto o base de una aplicación que permite su análisis desde varios puntos de vista. Establece los fundamentos para que analistas, diseñadores, y programadores trabajen en una línea común que permita alcanzar los objetivos y necesidades del sistema informático, existen diferentes estilos arquitectónicos dentro de ellos el estilo de arquitectura en capas.

Como parte vital dentro de la arquitectura se encuentran los patrones de arquitectura.

¿Qué es un patrón?

Un patrón describe un problema que ocurre una y otra vez en el entorno y describe también el núcleo de la solución al problema, de forma que puede reutilizarse continuamente.

Para el desarrollo de la aplicación se propone utilizar el patrón de Capas.

¿Por qué patrón Capas?

Este patrón define cómo organizar el modelo de diseño través de capas, donde los componentes de una capa solo pueden hacer referencia a componentes en capas inmediatamente inferiores, por lo que se dice que las capas pueden estar físicamente distribuidas. Este patrón simplifica la comprensión y la organización del desarrollo de sistemas complejos, reduciendo las dependencias de forma que las capas más bajas no son conscientes de ningún detalle o interfaz de las superiores.

Esta arquitectura brinda la ventaja de aislar definitivamente la lógica de negocios de todo lo que tenga que ver con el origen de datos, ya que desde el manejo de la conexión, hasta la ejecución de una consulta, la manejará la capa de Acceso a Datos. De este modo, ante cualquier eventual cambio, solo se deberá cambiar una capa en específico (ALEJANDROTERUEL 2001).

Ver **Figura 18** para conocer estructura de la Arquitectura Multi-Capas.

- **Capa de presentación o capa de interfaz de usuario:** es la forma de ofrecer al usuario un modo de interactuar con la aplicación. Las interfaces de usuario se implementan utilizando formularios, controles u otro tipo de tecnología que permita procesar y dar formato a los datos de los usuarios, así como adquirir y validar los datos entrantes procedentes de éstos.
- **Capa de lógica de negocio¹⁵:** se encapsula las responsabilidades y los procesos que en ella se comprenden, su funcionamiento basado en el uso de clases o componentes y a su vez compuestos estos por métodos que de una forma u otra establecen la relación entre los componentes de presentación y los componentes de acceso a datos. Para el acceso al subsistema de la capa de acceso a datos se presenta una interfaz denominada “Factoría típica”, clase que implementa el patrón para clases con un objetivo y funcionalidades similares, o sea todo flujo de información entre estas dos capas es a través de esta clase.

- **Capa de Acceso a Datos:** permite conocer el formato de los datos que se intercambian a través de los componentes de acceso a datos y la capa lógica de negocio, permiten el acceso a los datos a través de los mismos. Además usan la tecnología de acceso a datos PDO e implementa un interfaz de programación para la gestión de los datos.
- **Capa Datos:** Contiene el almacenamiento de Datos.

¿Por qué una Arquitectura Multi-Capas?

- **Abstracción total acerca del origen de datos:** las distintas capas se especializan absolutamente en la funcionalidad que deben brindar sin importar cual es el origen de los datos procesados.
- **Bajo costo de desarrollo y mantenimiento de las aplicaciones:** esta arquitectura brinda un control más cercano de cada componente, así como también la posibilidad de una verdadera reutilización del código.
- **Mejor calidad en las aplicaciones:** como las aplicaciones son construidas en unidades separadas, estas pueden ser probadas independientemente y con mucho mas detalle, esto conduce a obtener un producto mucho más sólido.
- **Reutilización de código:** la concepción natural de un sistema desarrollado con esta arquitectura, promueve la reutilización de sus componentes en varias partes del propio desarrollo y de futuros sistemas.



Figura 18. Arquitectura Multi_Capas.

3.5 Mecanismo de Diseño.

Los mecanismos de diseño se modelan para comunicar la manera más óptima en que debe darse solución a problemas repetitivos en la aplicación. Aunque su desarrollo y mantenimiento es opcional, se recomienda su uso en entornos de desarrollo complejos. Con su elaboración de modela un

conocimiento que ayudara al desarrollo de la aplicación actual y a futuras iteraciones. En los mecanismos de diseño intervienen diversos elementos de la aplicación (clases, subsistemas).

Los mismos reportan beneficios como:

- Mantener la homogeneidad en el diseño.
- Reutilizar soluciones anteriormente probadas.
- Reutilizar documentación.

Un ejemplo típico de mecanismos son los patrones de diseño.

¿Qué es un Patrón de Diseño?

Un patrón de diseño es una solución a un problema de diseño no trivial que es efectiva y reusable, igualmente es una descripción de clases y objetos comunicándose entre sí que resuelve un problema de diseño general en un contexto particular. Un patrón de diseño identifica: Clases, Instancias, Roles, Colaboraciones y la distribución de responsabilidades. En la actualidad son muchos los patrones de diseños utilizados en la construcción de una aplicación y en base a las responsabilidades asignadas se clasifican en:

- Experto.
- Creador.
- Alta Cohesión.
- Bajo Acoplamiento.
- Controlador.

Dentro del patrón Creador definidos por la GoF¹⁶ (Gang of Four) se encuentra el patrón de Factoría. Este patrón es un modelo que utiliza abstracción de clases para crear y relacionar objetos sin conocer de qué clase son. Se utiliza cuando la aplicación no sabe de antemano el tipo de objeto que se va a crear, es en tiempo de ejecución cuando toma la decisión (LARMAN 1999).

Mecanismo de Seguridad:

Las aplicaciones Web permiten el acceso de usuarios a recursos centrales como el servidor Web y, a través de éste, a otros como los servidores de base de datos. Con la implementación correcta de medidas de seguridad pueden protegerse los recursos y proporcionar un entorno seguro para los usuarios que trabajen en la aplicación. Administrar centralmente la seguridad de la aplicación es un factor esencial para controlar el acceso a la misma, esto se logra a través del empleo de variables de sesión, en la cual se registra la identidad del usuario, una vez se autentica en el sistema. Al constituir

esta aplicación como módulo de un ERP, las variables de sesión se obtienen a través de otro módulo encargado de la seguridad, que brinda esta posibilidad por medio de un servicio Web, el cual proporciona una interfaz cControlacceso para la autenticación de los usuarios, esta interfaz contiene un método público llamado logueo, que recibe como parámetro el usuario, la contraseña y el módulo al que desea entrar, este método devolverá un mensaje de error en caso de que exista algún problema. En caso de éxito en la autenticación el sistema devuelve además de las variables de sesión necesarias para tratar al usuario en el sistema, el menú y la barra de herramienta que será diferente para cada usuario en dependencia del rol. El rol del usuario es una de las variables de sesión de mayor importancia ya que esto indica al sistema los privilegios que tiene el usuario autenticado.

A continuación se muestra en la **Figura 19** el mecanismo de diseño para la seguridad del sistema basado en el uso de servicios Web. .

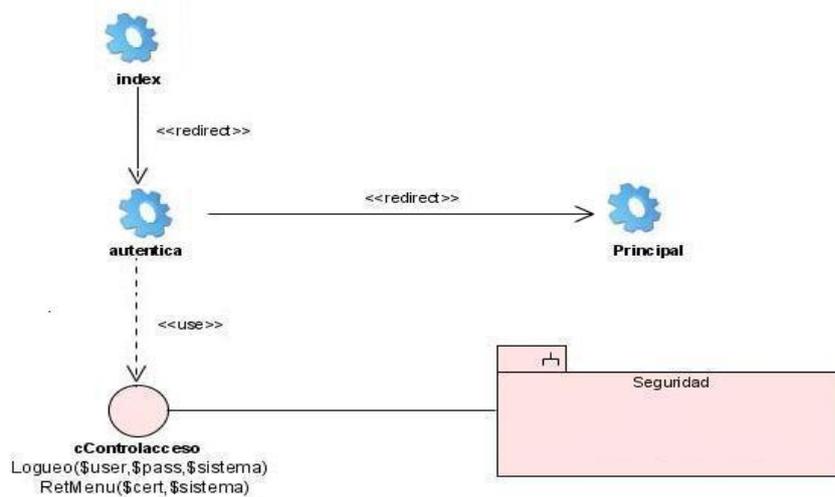


Figura 19. Mecanismo de Seguridad.

Mecanismo de Acceso a Datos:

¿Por qué definir un mecanismo de acceso a datos?

El mecanismo de acceso a datos facilita el acceso a los datos del sistema, creando puntos de referencias para desarrolladores, permitiendo obtener diagramas más entendibles que proporcionen una mayor comunicación con el equipo de desarrollo, trazando una línea común y posibilitando la reutilización.

La vista estática de este mecanismo de acceso a datos (Figura 19) visualiza un conjunto de clases que interactúan para dar acceso y manipulación de los datos de la persistencia desde el nivel más bajo, es

decir, utilizando los objetos nativos brindados por el entorno de desarrollo PHP como son la extensión PDO que define una ligera interfaz, para el acceso a bases de datos., siguiendo así hasta la abstracción del acceso a datos a través de meBase de la cual heredan las clases particulares del sistema como Típicas y meSimples. Para dar la responsabilidad a una clase que encapsulara las instancias de estos objetos se definió la clase FactoríaTípica, que está concebida para instanciar todas las típicas del sistema para utilizarlas de algún modo.

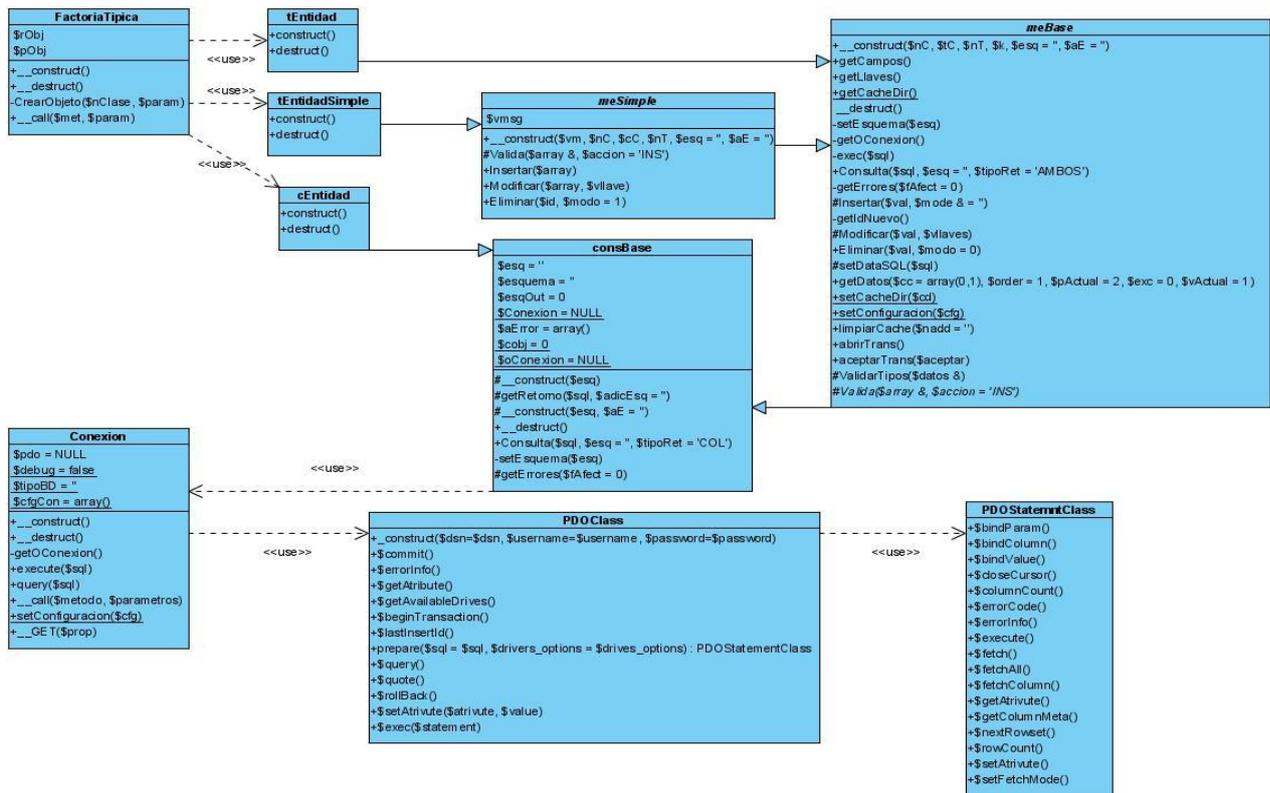


Figura 20. Diagrama general del “Mecanismo de Acceso a Datos”.

A continuación se muestra una descripción más detallada de estas clases.

FactoríaTípica: Clase que implementa la interfaz del modelo de persistencia con el resto de los subsistemas. A través de esta clase se crean y se manipulan los objetos de las típicas simples, los nomencladores y las demás típicas. Es una puerta entre la capa de Acceso a Datos y la capa de Lógica de Negocio. Implementa un método de instancia de clases típicas.

tEntidadSimple: Es una clase que representa a las clases típicas (nomencladores simples) en general de la aplicación. Estas típicas son de una implementación muy sencilla, pues la mayoría de las líneas que normalmente habrían que codificar quedaron encapsuladas en la clase base de las mismas. Para la implementación de esta clase se decidió aplicar el patrón de diseño Table Data Gateway, que

consiste en crear una instancia por cada tabla existente en la BD. Sus métodos consisten en las operaciones básicas que se realizan sobre estas tablas, insertar, modificar y eliminar. Hereda de la clase abstracta meSimple.

meSimple: Clase abstracta heredera de meBase, y a la vez base para la implementación de las típicas que responderán a los nomencladores simples del modelo de persistencia dado. Redefine las operaciones básicas con la funcionalidad de Validación dada. Redefine las operaciones básicas que pudieran realizarse a una entidad (insertar, eliminar, modificar) para los nomencladores simples.

meBase: Es una clase abstracta que constituye la base para el resto de las que implementen funcionalidades para el trabajo con las entidades del sistema desarrollado. Implementa además, las operaciones básicas que pudieran realizarse a una entidad como pudieran ser: INSERT, DELETE, UPDATE así como un conjunto de consultas. Esta clase encapsula todo lo relacionado con la conexión al gestor de la bases de datos.

consBase: Esta clase es la base en toda la jerarquía de Acceso a Datos y es empleada para aportar contenido dinámico a las plantillas. Encapsula el objeto conexión. Implementa la operación de Consulta.

Conexión: Esta clase es la encargada de establecer la conexión con el servidor de la base de datos a través de un objeto PDO de la librería de PHP.

PDO: Es un modelo de acceso a datos para PHP que brinda una capa de abstracción para el acceso a BD desde PHP.

3.6 Diagramas de Interacción

Los diagramas de interacción muestran las interacciones entre objetos mediante transferencia de mensajes entre objetos o subsistemas. Los diagramas de interacción se clasifican en Diagramas de Colaboración y Diagramas de Secuencias.

Un diagrama de colaboración muestra las interacciones entre objetos organizadas entorno a los objetos y enlaces entre ellos; por su parte, un diagrama de secuencia muestra las interacciones entre objetos, ordenadas en secuencia temporal durante un escenario concreto.

A continuación se muestran los diagramas de secuencia para las realizaciones del caso de uso Gestionar Planificación. Para observar otros diagramas de secuencia ver **anexo 1**.

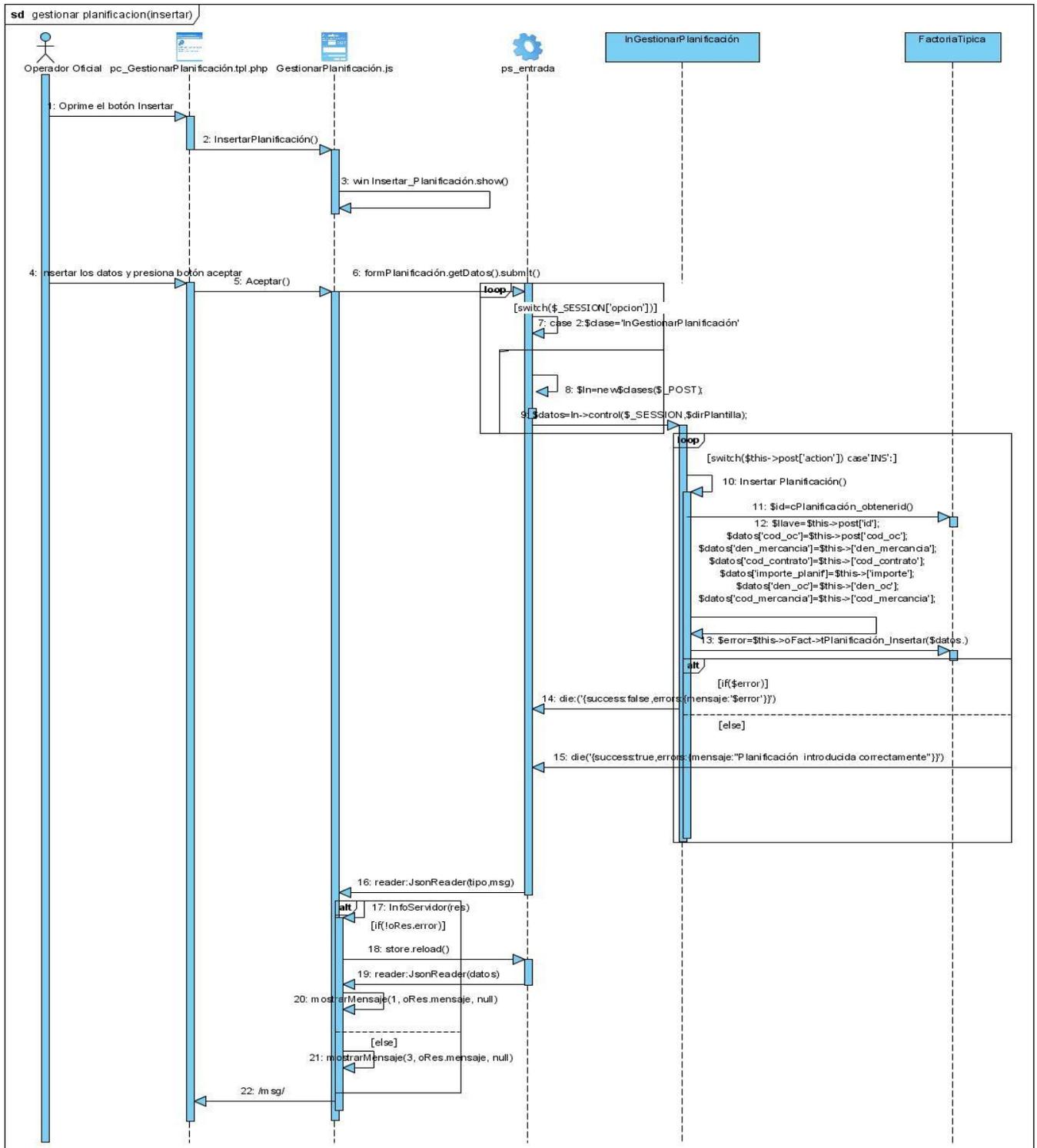


Figure 22 Diagrama de secuencia “Realización Insertar (Caso de Uso Gestionar Planificación)”.

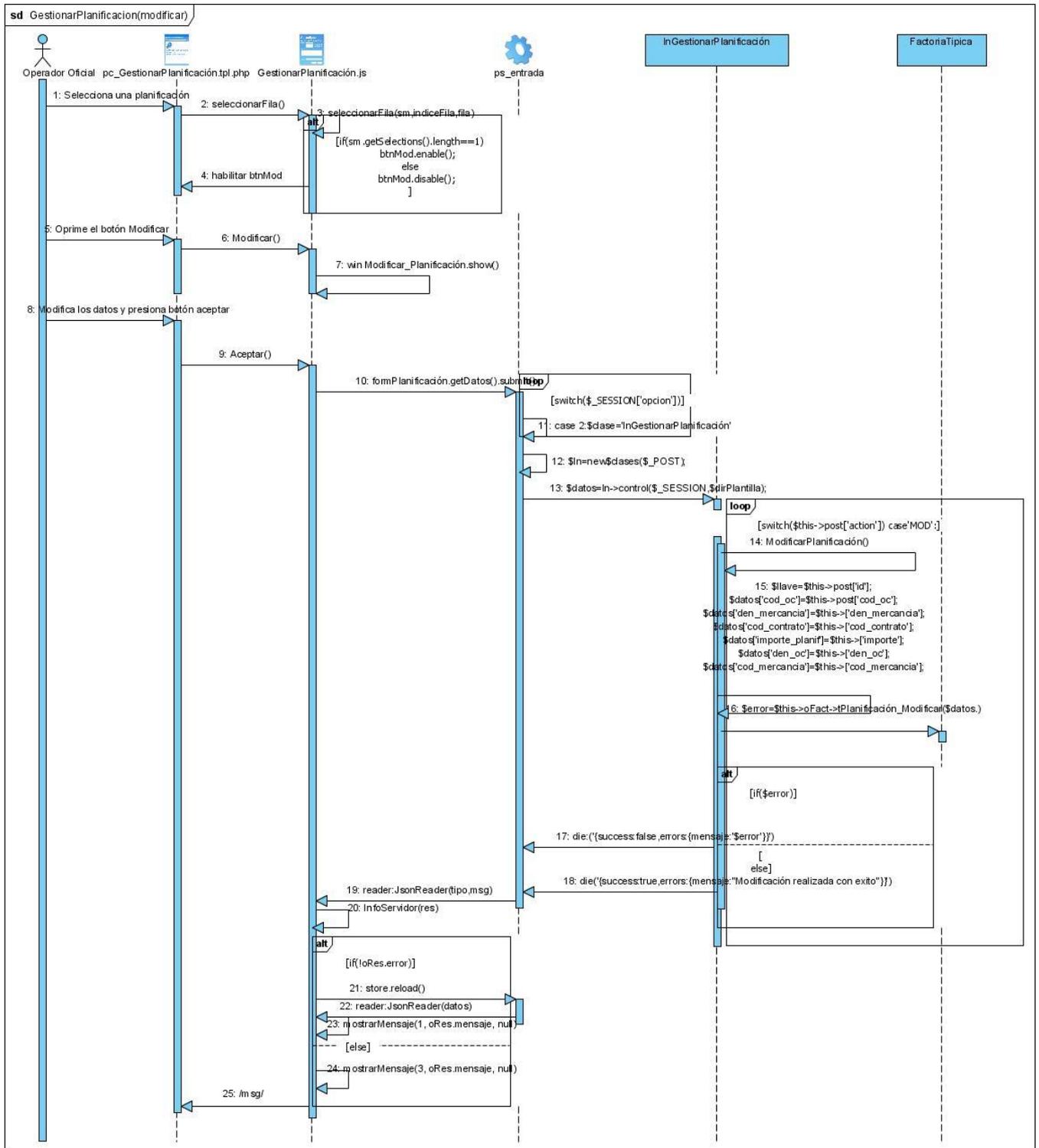


Figure 23 Diagrama de secuencia “Realización Modificar (Caso de Uso Gestionar Planificación)”.

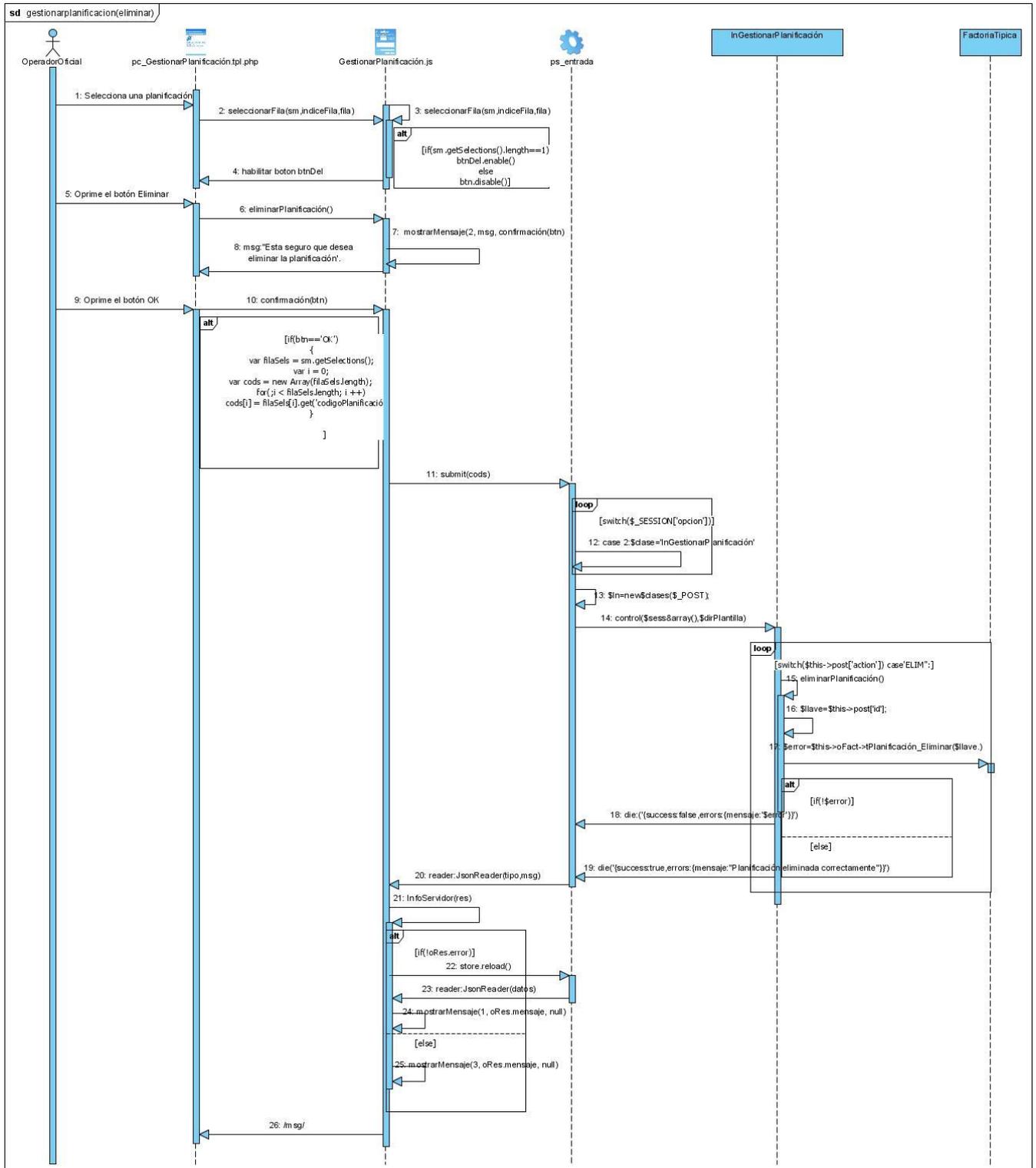


Figure 24 Diagrama de secuencia “Realización Eliminar (Caso de Uso Gestionar Planificación)”.

3.7 Diagrama de Clases del Diseño.

Los diagramas de clases del diseño permiten adquirir un nivel correcto de abstracción y detalle modelando los artefactos del sistema, páginas, enlaces entre ellas, código que irán creando las páginas y su contenido dinámico una vez que estén en el navegador del cliente. El objetivo fundamental del diseño es crear una entrada apropiada y un punto de partida para la implementación, capturando los requisitos o subsistemas individuales, interfaces y clases. El diseño convierte el modelo de análisis en diseños de datos, arquitectónicos, de interfaz y a nivel de componentes del software.

Este sistema tiene concebido por cada caso de uso lo siguiente:

- Una clase de lógica de negocio (In_**NombreCU**) por cada caso de uso.
- Un fichero **GLOBAL** para la configuración de las direcciones de las clases a través de la función `__autoload` y algún elemento de configuración.
- La página servidora tendrá las instancias de los objetos In_**NombreCU**.
- Una página servidora **principal** que redireccionará a la servidora de cada Caso de Uso por módulo.
- Una interfaz **cControlacceso** que permite la autenticación de los usuarios.
- Una clase **InBase**: Es la clase base de lógica de negocio, de la cual heredarán todas las clases "In_NombreCU". En ella se define el objeto de la clase Factoría Típica.
- Una clase **FactoríaTípica** que a través de la cual se crean y se manipulan los objetos de acceso a datos como las tablas y consultas. Es una interfaz entre la Lógica de Negocio y la capa de Acceso a Datos. Basada en el patrón Factoría.
- El fichero **ext-all** para construir librerías de componentes ext y funciones generales.
- El fichero **ext-base** que son las librerías de extensiones puras de ext; tiene diferentes especificaciones de extensiones.

Con el objetivo de optimizar la modelación del diseño de la aplicación en desarrollo, se construyó un diagrama genérico de clases del diseño (**Figura 20**), que tiene como misión la representación de los elementos comunes para todos los diagramas de clases a desarrollar, destacando en color azul el nombre de las clases que varían en dependencia del diagrama que se esté realizando, por tanto los posteriores diagramas solamente contendrán las páginas clientes, servidoras, las lógicas de negocio, y en caso de tener formularios y ficheros java script también serán incluidos.

A continuación se muestran los diagramas de clases del diseño.

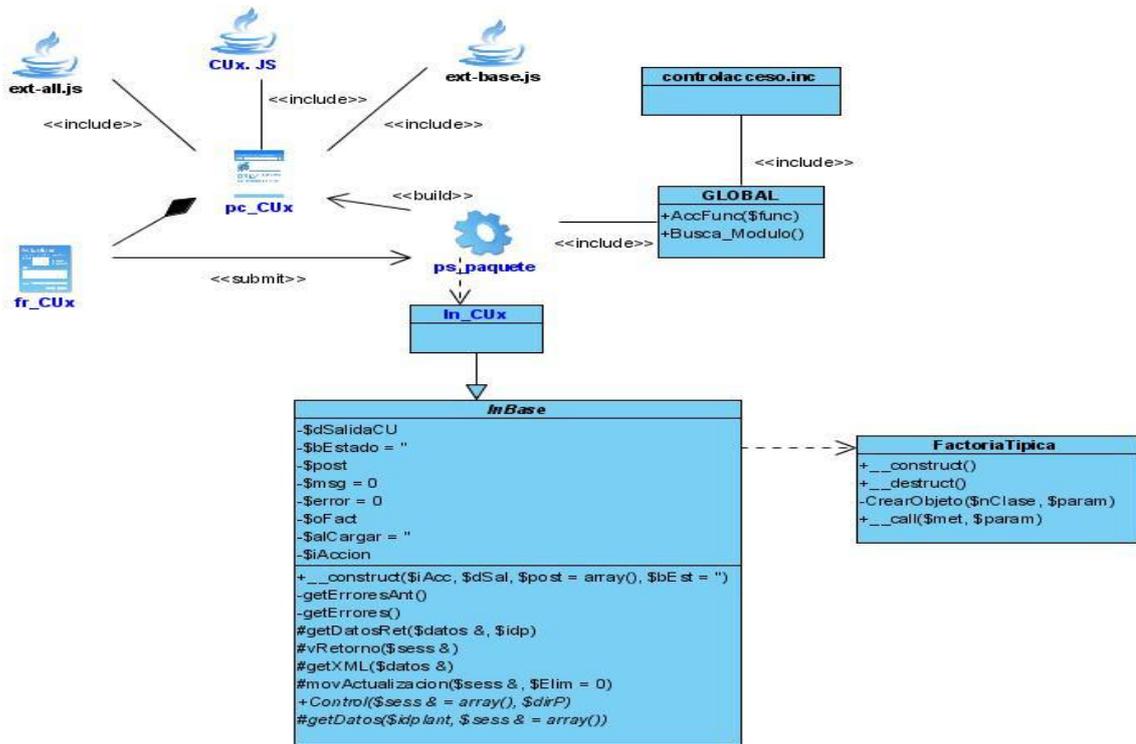


Figura 25. Diagrama de clases del diseño “Genérico”.

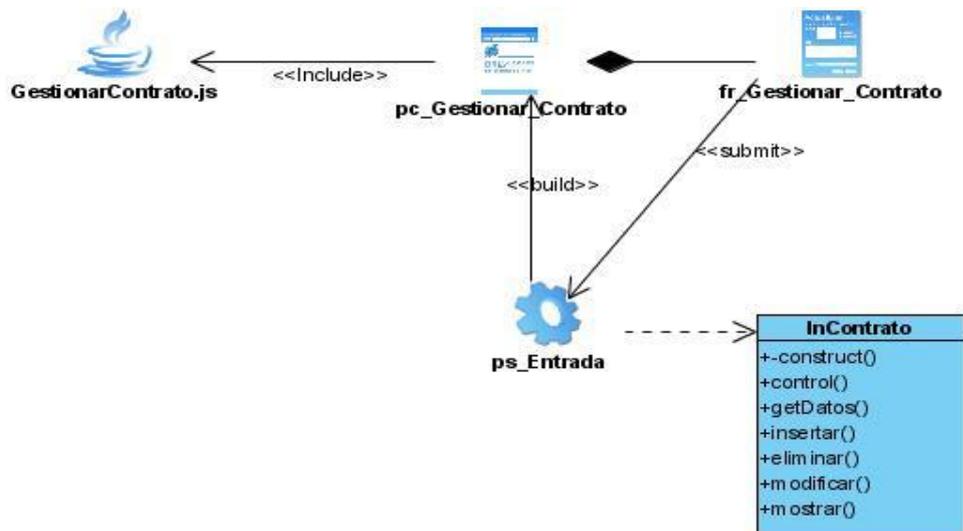


Figura 26. Diagrama de clases del diseño “Gestionar Contrato”.

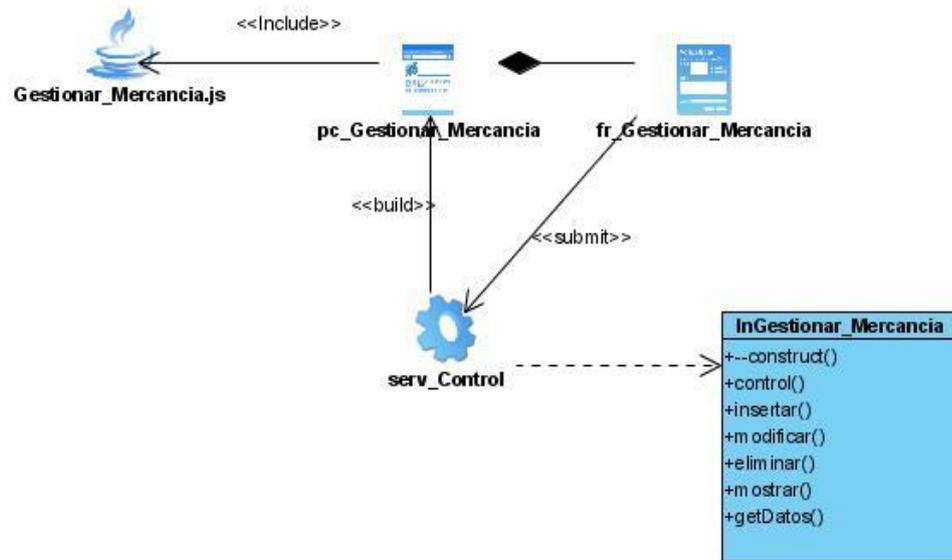


Figura 27. Diagrama de clases del diseño “Gestionar Mercancía”.

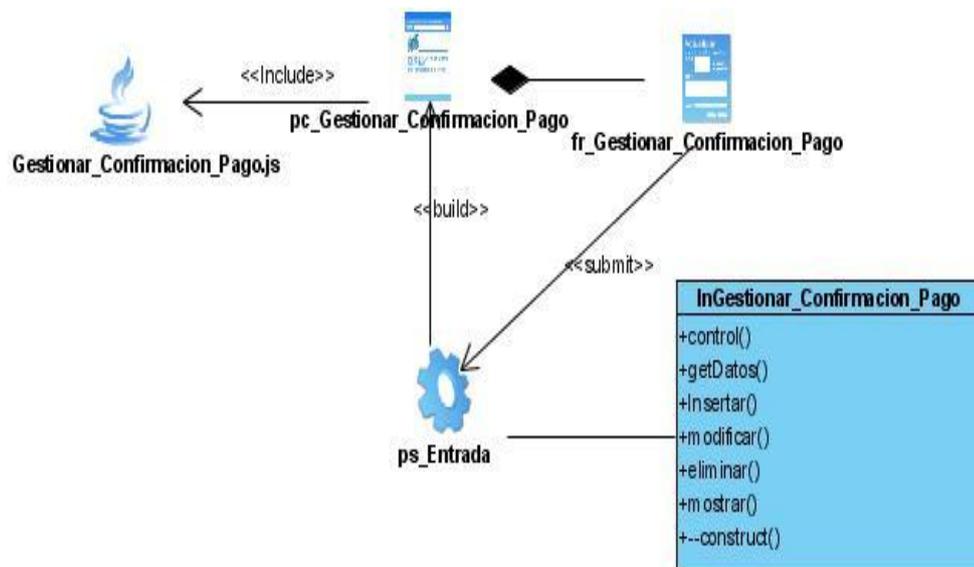


Figura 28. Diagrama de clases del diseño “Gestionar Confirmación de Pago”.

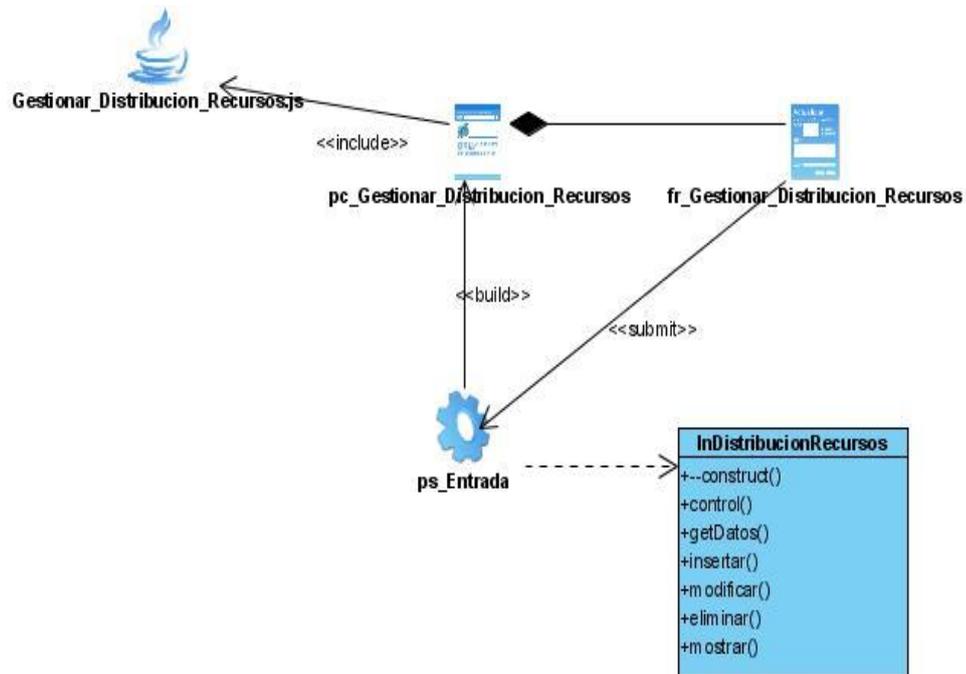


Figura 29. Diagrama de clases del diseño “Gestionar Distribución de Recursos”.

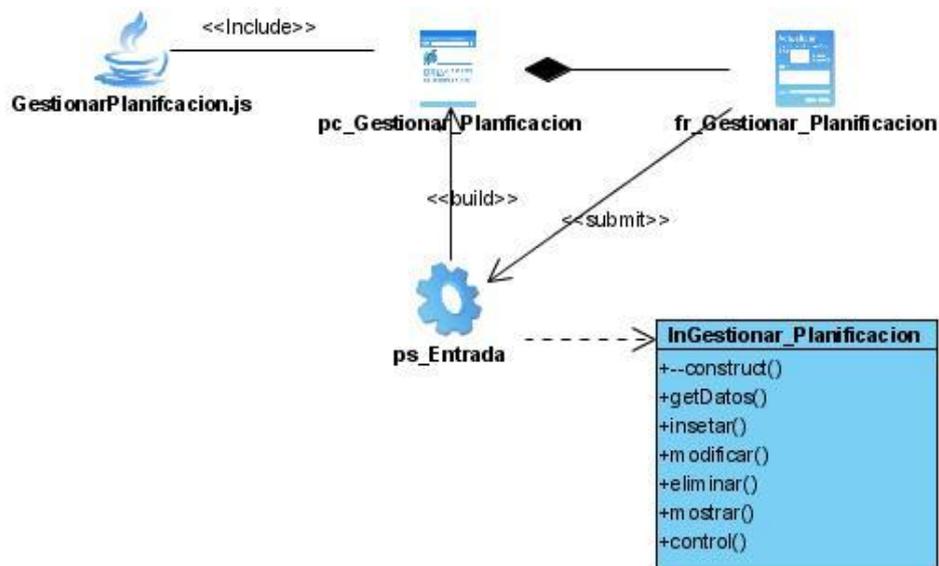


Figura 30. Diagrama de clases del diseño “Gestionar Planificación”.

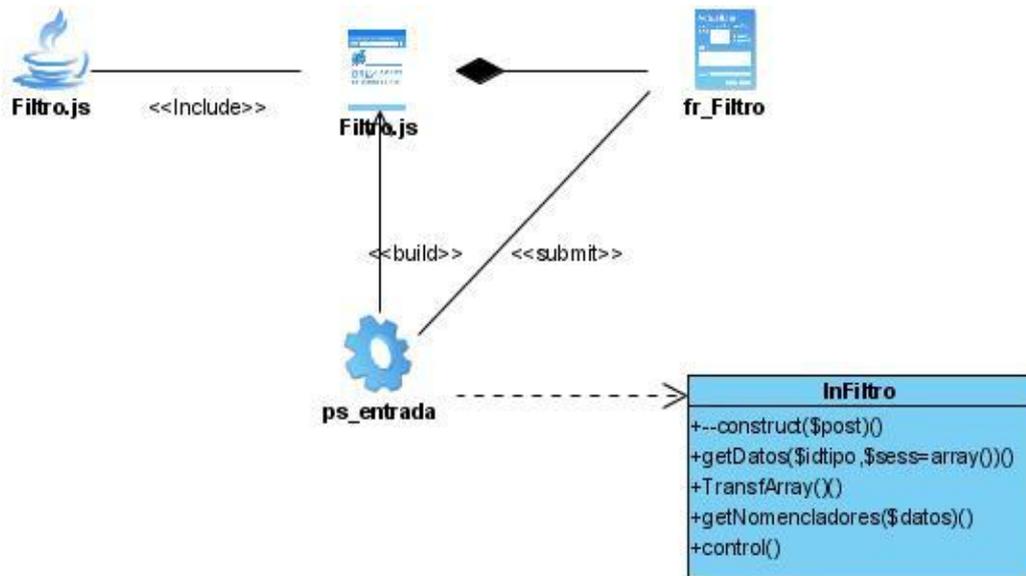


Figura 31. Diagrama de clases del diseño “Seleccionar Criterio de Búsqueda”.

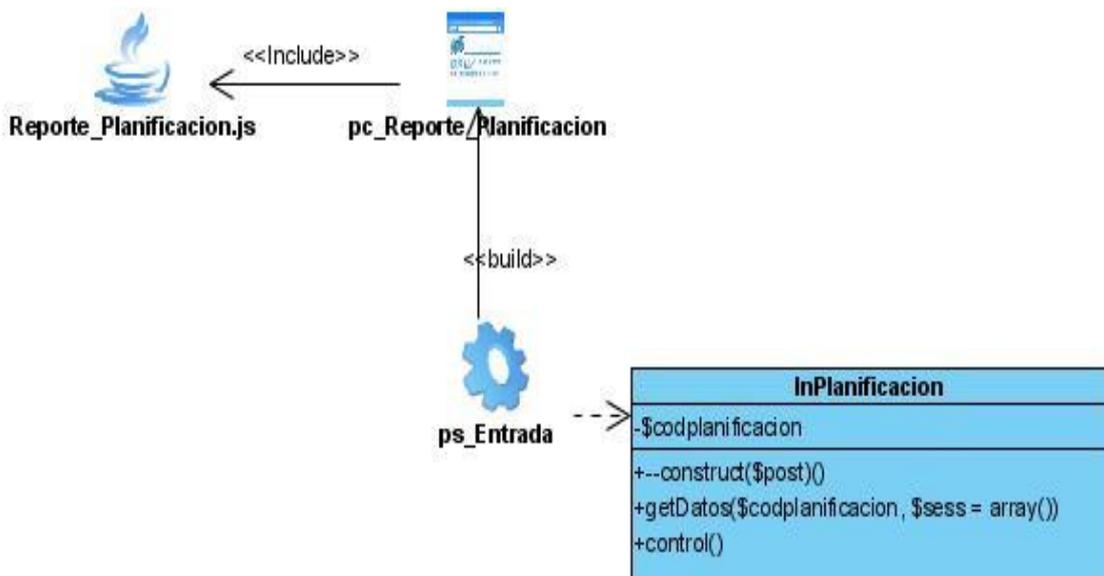


Figura 32. Diagrama de clases del diseño “Mostrar Reporte Planificación”.

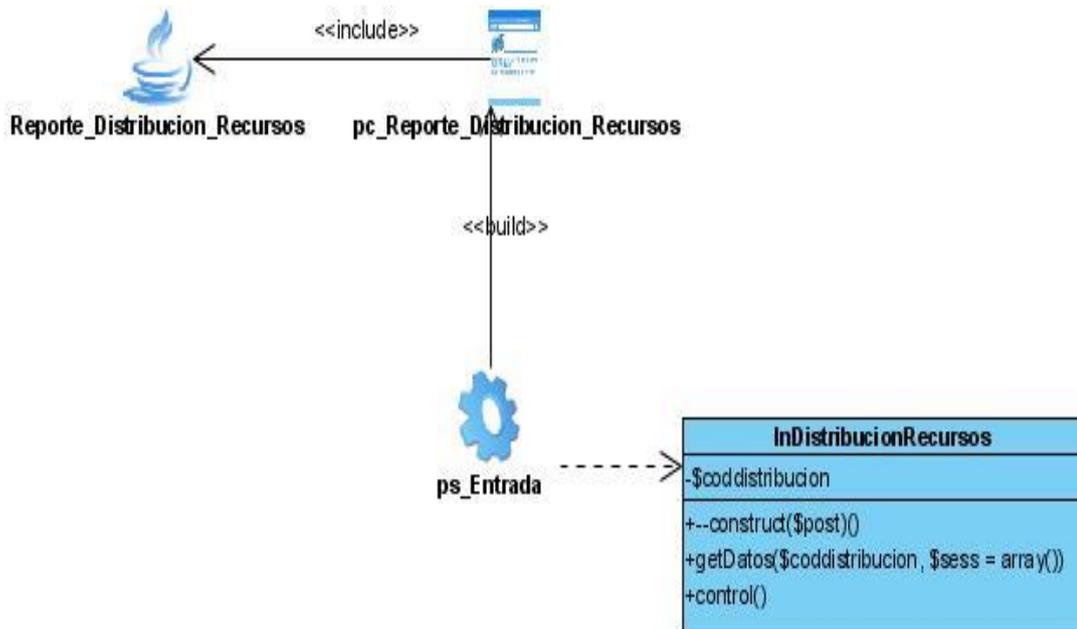


Figura 33. Diagrama de clases del diseño “Mostrar Reporte de Distribución de Recursos”.

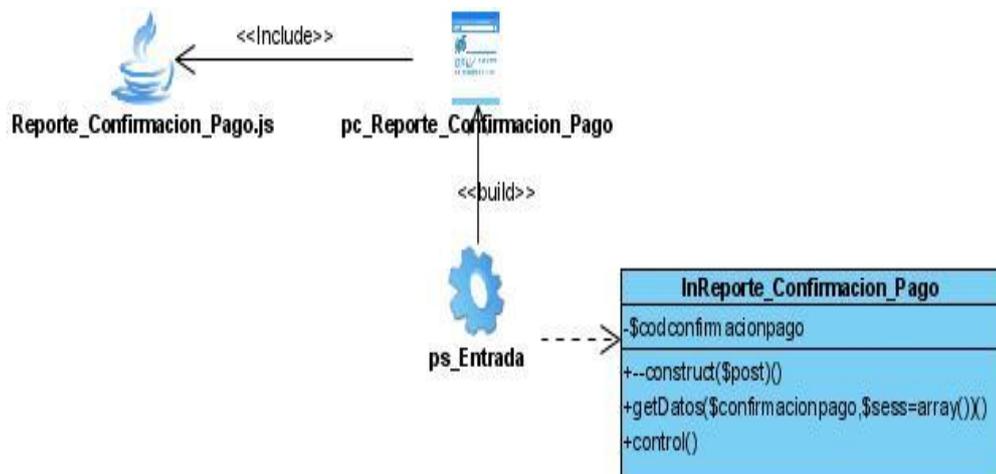


Figura 34. Diagrama de clases del diseño “Mostrar Reporte de Confirmación de Pago”

3.8 Diseño de la Base de Datos.

Diagrama Entidad Relación de la BD:

Los diagramas o modelos entidad-relación son una herramienta para el modelado de datos de un sistema de información. Estos modelos expresan entidades relevantes para dicho sistema, sus

interrelaciones y propiedades. Además en él se modela el tratamiento de la información con carácter persistente dentro del sistema, para ver la descripción de las tablas ver **anexo 3**.

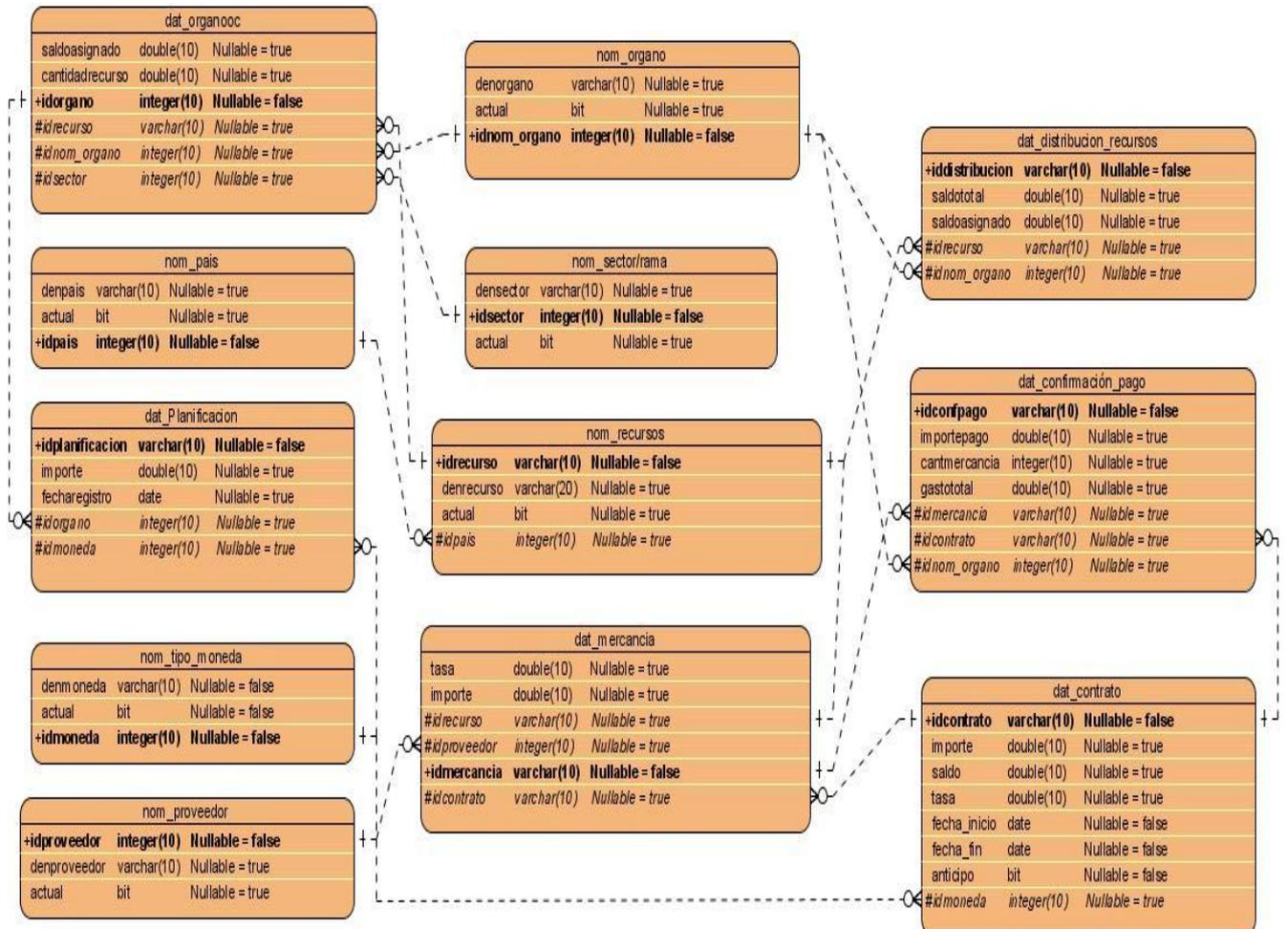


Figura 35. Diagrama que representa el Modelo de Datos.

3.10 Estándares de Diseño.

La página principal de la aplicación, se concibe como un portal, con un menú, que no debe exceder de 3 niveles de profundidad, donde se agrupan las funcionalidades del sistema.

Los iconos de las acciones principales que puede realizar el usuario (modificar, eliminar, insertar, buscar) se mostrarán en la parte superior de la página. Las páginas deben tener una cabecera (banner) representativa, un área de trabajo y una barra menú con las opciones, además tener una hoja de estilo en común para lograr la uniformidad, es decir, se trabajará con la familia de fuentes Arial, Geneva, Helvética, Verdana, el tamaño de la misma no debe diferir mucho de 11px. Los colores con los que se trabajarán serán tonalidades claras basadas en el azul combinado con el color blanco o gris.

3.11 Estándares de Programación.

Resulta ventajoso utilizar un estándar para escribir código, entre dichas ventajas, tenemos las siguientes:

- Reducir errores
- Escribir un código comprensible y fácil de leer
- Garantizar una buena comunicación entre los programadores del equipo
- Facilitar el mantenimiento del software

A continuación se muestra los principales estándares aplicados:

Legibilidad		
Objetivo: Nombrar las clases e instancias de las mismas de forma estándar para todas las aplicaciones.		
Apariencia de clases y objetos	Primera letra en mayúscula y serán en singular.	Los nombres de las clases y las instancias de las mismas deben comenzar con la primera letra en mayúscula y el resto en minúscula, en caso de que sea un nombre compuesto se empleará notación PascalCasing ¹⁷ . Ejemplo: MiClase (). Los prefijos a utilizar serán los siguientes: t----- Clases Típicas In----- Lógica de Negocio. c-----Consulta me-----Manejadora de Entidad
Nombre de clases y objetos	Relacionados al propósito	Permitir inferir por el nombre el propósito de la clase o instancia.. Para el caso de las instancias es recomendable que se denoten así: Para la clase: tNomumedida su instancia será \$Oumedida, de forma tal que la primera letra indique que es un objeto y el resto, la clase a la que pertenece.
Apariencia de atributos	Primera letra en minúscula	El nombre que se le da a los atributos de las clases debe comenzar con la primera letra en minúscula, en caso de que sea un nombre compuesto se empleará notación CamelCasing ¹⁸ .
Nombre de atributos	Nemotécnicos	El nombre empleado, debe permitir que con sólo leerlo se conozca el propósito del mismo dentro de la clase.
Apariencia de las funciones	Primera letra en mayúscula	Los nombres de las funciones deben comenzar con la primera letra en mayúscula y el resto en minúscula, en caso de que sea un nombre compuesto se empleará notación PascalCasing .Si son funciones que obtienen un dato se emplea el prefijo get y si fijan algún valor se emplea el prefijo set .
Nombre de	Nemotécnicos	El nombre empleado, debe permitir que con sólo leerlo se conozca

las funciones		el propósito de la misma dentro de la clase.
Declaración de parámetro en funciones	Agrupados por tipos primero los string, los numéricos y valores por defecto.	Los parámetros que se le pasan a las funciones se recomienda sean declarados de forma tal que estén agrupados por el tipo de dato que contienen. Ejemplo: BuscaUnidad (\$nTabla(string), \$nCampos(string), \$kIndice (entero)).
Variables y constantes		
Apariencia de constantes	Todas sus letras en mayúscula	Se deben declarar las constantes con todas sus letras en mayúscula.
Apariencia de variables:	Primera letra en minúscula	El nombre que se le da a las variables debe comenzar con la primera letra en minúscula, en caso de que sea un nombre compuesto se empleará notación CamelCasing.
Nombres de las variables y constantes	Nemotécnicos	El nombre empleado, debe permitir que con sólo leerlo se conozca el propósito de la misma. Ejemplo: \$nCampos.
Declaración de constantes y asignación a variables	Una por cada línea	Se recomienda declarar una constante por cada línea y con las asignaciones a las variables sucede lo mismo. Ejemplo: define("CONSTANT1", "value1"); define("CONSTANT2", "value2"); \$nTabla='nomproducto'; \$kIndice=0;
Indentación		
Objetivo: Lograr una estructura uniforme para los bloques de código así como para los diferentes niveles de anidamiento.		
0 espacios en blanco desde la izquierda en	Require Include Class	No se empleará ningún espacio en blanco desde la izquierda para las instrucciones antes mencionadas. Se tomará como inicio de la página el tag PHP <?
2 espacios en blanco desde la izquierda en	Function Define	Se dejarán dos espacios en blanco desde la izquierda en las instrucciones antes mencionadas.
2 espacios en blanco desde	Inicio y fin de bloque	Se recomienda dejar dos espacios en blanco desde la instrucción anterior para el inicio y fin de bloque. Lo mismo sucede para el

la referencia en		caso de las instrucciones If, else, For, While, Do While, Switch, Foreach.
Niveles de anidación	Hasta 5 niveles	Se recomienda emplear hasta 5 niveles de anidación en instrucciones If, For, While.
Comentarios, separadores, líneas y espacios en blanco		
Objetivo: Establecer un modo común para comentar el código de forma tal que sea comprensible con sólo leerlo una vez.		
Ubicación de comentarios	Al inicio de cada clase o función y al final de cada bloque de código.	Se recomienda comentar al inicio de la clase o función especificando el objetivo de la misma así como los parámetros que usa (especificar tipos de dato, y objetivo del parámetro) entre otras cosas. Y se comenta también cuando se cierran los ciclos, clases, instrucciones if y otras.
Separador de instrucciones	Se emplea el punto y coma.	Se recomienda usar el separador al final de cada instrucción y no en la línea de abajo. Ejemplo: define ("CONSTANT", "value1");
Líneas en blanco	Se emplean antes de cada función.	Se recomienda dejar una línea en blanco antes de la definición de cada función para dar claridad al código.
Espacios en blanco	Entre operadores lógicos y aritméticos.	Se recomienda usar espacios en blanco entre estos operadores para lograr una mayor legibilidad en el código. Ejemplo: \$nTabla = 'nomproducto'; if ((\$nTabla) && (\$nFields))

3.12 Conclusiones

En este capítulo se mostró como a través del análisis y diseño, se transformó el modelo de análisis en un modelo de diseño, es decir, en una estructura de clasificadores y realizaciones de casos de uso. Además se construyeron varios artefactos para llevar a cabo el proceso de implementación del sistema. También se identificaron otras funcionalidades que se deben tener en cuenta para futuras versiones del sistema. Se utilizaron diagramas de clases Web para explicar la lógica del negocio del sistema, se crearon mecanismos de diseño que simplifican el modelado y se modeló la base de datos.

CAPÍTULO 4. ESBOZO DE IMPLEMENTACIÓN

4.1 Introducción.

En este capítulo se procede a la implementación del sistema que permite crear la línea base ejecutable de la arquitectura, la misma comienza con la culminación del diseño y se refiere a la implementación en términos de componentes; para el desarrollo de este capítulo se tuvieron en cuenta algunos artefactos que fueron generados en el flujo de análisis y diseño. A partir de los cuales se desarrolló la implementación del sistema en términos de componentes, es decir, ficheros de código fuente, scripts, ficheros de código binario, ejecutables y similares.

4.2 Modelo de Implementación.

El modelo de implementación constituye la vista de Implementación de la arquitectura, y como tal guía las labores de construcción del sistema. Este contiene fundamentalmente los subsistemas de implementación, incluyendo las dependencias y otras informaciones necesarias para su utilización. (Ver figura 36)

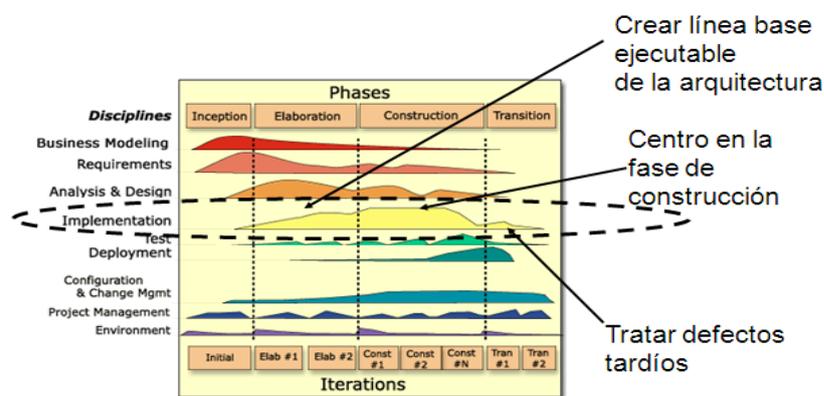


Figura 36 Ciclo vida de RUP.

Para lograr una mejor comprensión de los componentes que forman el sistema, se presentarán los diagramas de componentes que describen los elementos físicos y lógicos del sistema y sus relaciones. Los componentes representan todos los tipos de elementos software que entran en la fabricación de aplicaciones informáticas; pueden ser simples archivos, paquetes, bibliotecas cargadas dinámicamente. Otra parte importante del modelo de implementación la constituyen los subsistemas de implementación que proporcionan una forma de organizar los artefactos en partes más manejables y una interfaz es un descriptor de las operaciones visibles de una clase, un componente u otra

entidad. Una interfaz no indica cómo debe implementarse las operaciones, solamente indica la existencia y visibilidad de operaciones que especifican un servicio. Un componente que implementa una interfaz ha de implementar todas las operaciones definidas por la interfaz, como también un subsistema de implementación que proporciona una interfaz tiene que contener componentes que proporcionen la interfaz u otros subsistemas (recursivamente) que proporcionen la interfaz.

A continuación se muestra en la Figura 36 el diagrama de componentes general. Ver anexo 2 para otros diagramas de componentes específicos.

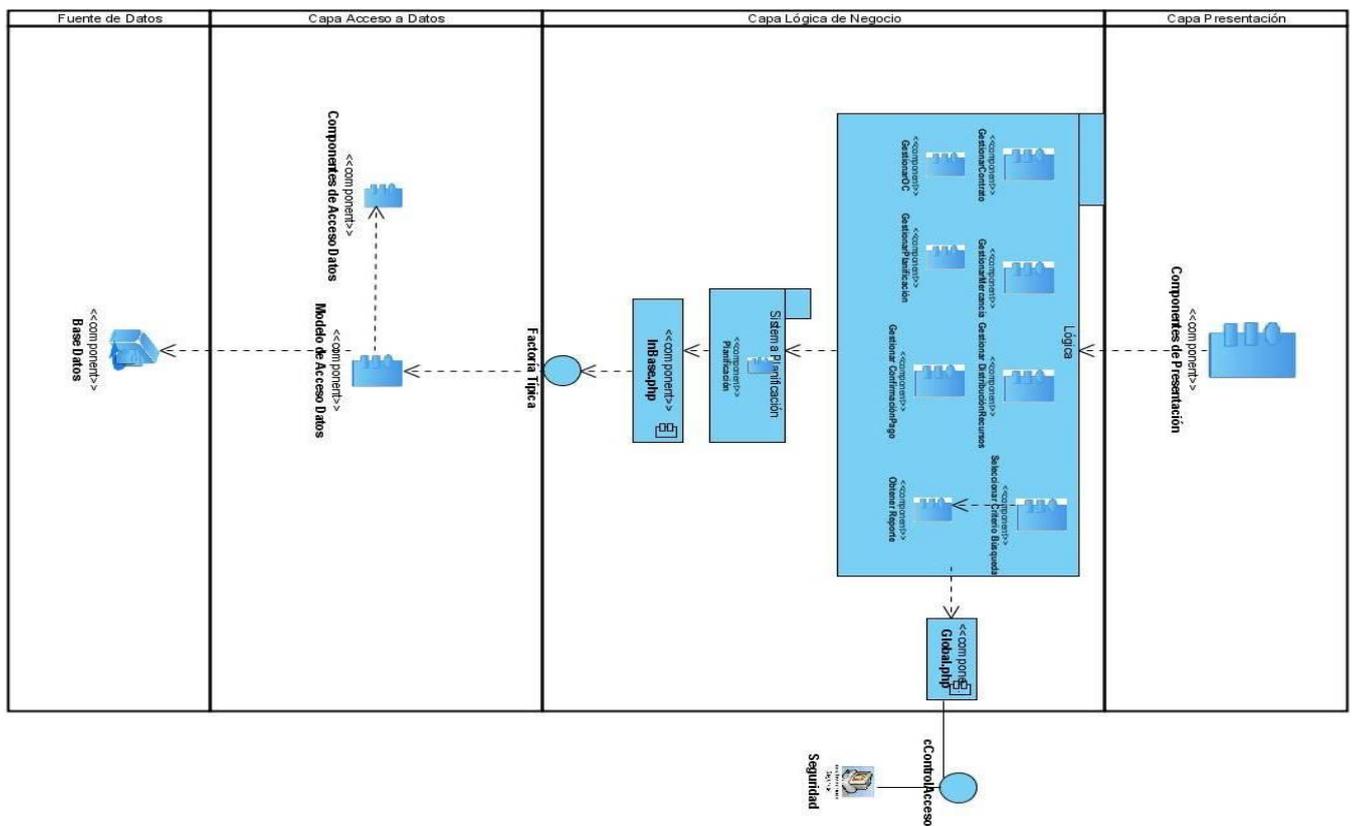


Figura 37. Diagrama de Componentes General.

4.3 Modelo de Despliegue.

El diagrama de despliegue representa la arquitectura de tiempo de ejecución de los procesadores, dispositivos y los componentes de software que se ejecutan en esa arquitectura. Es la última descripción física de la topología del sistema y describe la estructura de las unidades de hardware. Además, representa la disposición de las instancias de componentes de ejecución en instancias de

nodos conectados por enlaces de comunicación. Un nodo es un recurso de ejecución tal como un procesador, un dispositivo o memoria. En los procesadores es donde se encuentran alojados los componentes.

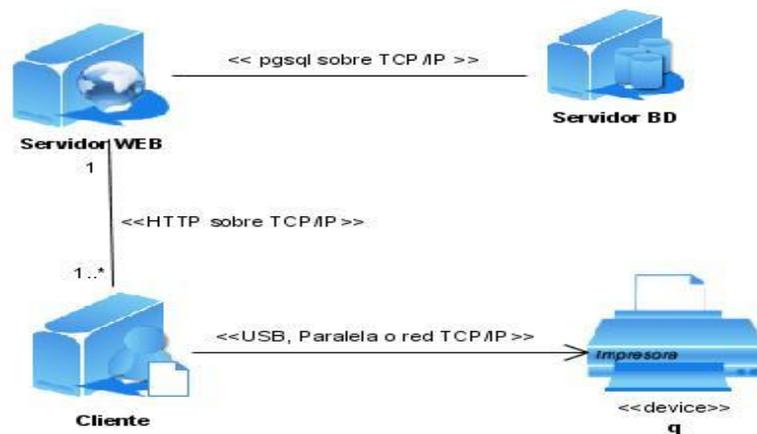


Figura 38. Diagrama de Despliegue.

El diagrama de despliegue mostrado anteriormente representa un modelo ideal que está en correspondencia con la arquitectura cliente-servidor en capas, compuesto por un Servidor Web, un Servidor de Base de Datos, una computadora para visualizar las operaciones sobre ambos servidores, un dispositivo que en este caso es una Impresora y las conexiones entre ellos.

4.4 Conclusiones

En este capítulo se mostró como a través de un esbozo de la implementación, se produjo un refinamiento de la vista de la arquitectura del modelo de despliegue, donde los componentes ejecutables fueron asignados a nodos. Se utilizaron diagramas de componentes para representar a través de un grafo los componentes de software unidos por medio de relaciones de dependencia; con los cuales se modeló la vista estática del sistema.

CONCLUSIONES

Este Trabajo de Diploma ha permitido pasar por los diferentes flujos de trabajo que poseen las cuatro fases de desarrollo para un proyecto de software, como lo propone RUP. Permitiendo en cada flujo de trabajo lograr el desarrollo y perfeccionamiento del diseño del producto de software, jugando un papel determinante las iteraciones que le garantizan al desarrollador la mejora en la calidad.

Cada uno de los capítulos desarrollados ha proporcionado la posibilidad de abrir nuevas fronteras en el conocimiento personal y en la búsqueda de la solución para la situación problemática que posee el MINFAR en cumplimentar el proceso del control de la ejecución de la planificación, así como ha permitido la realización y cumplimiento de los objetivos trazados para culminar exitosamente con el diseño del sistema que se quiere desarrollar.

RECOMENDACIONES

Se recomienda realizarle las descripciones textuales a los casos de uso “Mostrar Reporte de Contrato” y “Mostrar Reporte de Mercancías”, así como sus realizaciones llegando hasta los diagramas de componentes. Además se sugiere, guiándose por el análisis y diseño anteriormente definido y por el desarrollo de los capítulos anteriormente expuestos, que se implemente un sistema que satisfaga y cumpla los requisitos del software y la situación problemática existente en el proceso del Control de la Ejecución de la Planificación Material y Financiera realizada por los Órganos del Aparato Central del MINFAR.

REFERENCIAS BIBLIOGRÁFICAS

- ALEJANDRO TERUEL. *Arquitectura de Capas* 2001. [Citado el: 3 de marzo del 2008]. Disponible en: <http://www ldc.usb.ve/~teruel/ci3715/clases/argCapas.html>.
- ALFARO, R. *¿Por qué elegir PHP?*, 2007. [Citado el: 24 de enero del 2008]. Disponible en: <http://www.mmug.cl/articulos.php?id=283>
- GRACIA, J. M. *UML: Diagramas UML. ¿Qué es UML?*, 2005.
- HENST, C. V. D. *¿Qué es el PHP?*, Citado el: 24 de enero del 2008]. Disponible en: <http://www.maestrosdelweb.com/editorial/phpintro/>
- LARMAN, C. . *UML y Patrones Introducción al análisis y diseño orientado a objetos*, 1999. [Citado el: 20 de abril del 2008]. Disponible en: http://alarcos.inf-cr.uclm.es/per/fgarcia/isoftware/doc/LabTr5_Multicapa.pdf.
- LEÓN, M. O. D. *ERP: su presencia en Latinoamérica. de Gerencia.com.* , 2001-2007. [Citado el: 20 de enero del 2008]. Disponible en: http://www.degerencia.com/articulo/erp_su_presencia_en_latinoamerica.
- LÓPEZ, C. *IMPACTO DE LOS ERP'S EN LAS EMPRESAS*, 2002-2007. [Citado el :10 de enero del 2008]. Disponible en: <http://www.gestiopolis.com/canales2/gerencia/erpbetty.htm>.
- LÓPEZ., A. L. and G. R. GÓMEZ. *Procesos de la Ingeniería de Software*. Disponible en: <http://209.85.215.104/search?q=cache:XrbKlpwopM4J:ccc.inaoep.mx/~grodrig/Descargas/IngSoft/ch06.pdf+validaci%C3%B3n+de+requerimientos&hl=es&ct=clnk&cd=23&gl=cu>
- MENDOZA, M. *Metodologías De Desarrollo De Software*, 2004. [Citado el :10 de febrero del 2008]. Disponible en: http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html
- PEREYRA., M. T. *Herramientas CASE.*, 2008. [Disponible en: <http://www.elprisma.com/apuntes/curso.asp?id=13324>
- . *Introducción a la Ingeniería de Software.*, 2007- 2008. [Disponible en: <http://teleformacion.uci.cu/mod/resource/view.php?id=6655>
- TIENDALINUX.COM. *Ventajas de PostgreSQL*, 5 de abril del 2008]. Disponible en: http://soporte.tiendalinux.com/portal/Portfolio/postgresql_ventajas_html.
- TORRE, A. D. L. *Lenguajes del lado servidor o cliente.*, 2006. [Citado el: 22 de Enero del 2008]. Disponible en: http://www.adelat.org/media/docum/nuke_publico/lenguajes_del_lado_servidor_o_cliente.html.
- WAN, C. S. *¿Por qué Visual Paradigm para UML?*

BIBLIOGRAFÍAS

COTA, A. *Ingeniería de Software.*, 1994. p.

GALVES., J. A. *Ingeniería de Software.*

JACOBSON, I. B., GRADY; RUMBAURGH JAMES. *El proceso unificado de software.* . la Habana, Editorial Felix Varela, 2004. p.

ITERA.SA. *Rational Unified Process*, Citada el: 20 de febrero de 2008]. Disponible en: http://www.itera.com.mx//index.php?option=com_content&task=view&id=18&Itemid=42

RODRÍGUEZ, A. *Hacia la definición de los Procesos del Negocio.*, 2005.

SOFTWARE., D. D. I. Y. G. D. *Fase de Inicio.Modelo del Negocio.*, 2007-2008. [Disponible en: <http://teleformacion.uci.cu/mod/resource/view.php?id=6680>

PRESMAN. *Ingeniería de Software: Un enfoque práctico.* Sexta Edición. México DF., Editorial McGraw Hill., p.

SOMMERVILLE, I. *Ingeniería de Software.* Séptima edición. México DF., Editorial Pearson, 2005. p.

GLOSARIO DE TÉRMINOS

- 1 **ERP:** *Enterprise Resource Planning*, sistema de información gerencial que integra muchas de las prácticas de los negocios asociados con las operaciones de producción.
- 2 **ARPA:** es acrónimo de la expresión en inglés *Advanced Research Projects Agency* ("Agencia de Proyectos de Investigación Avanzada"), denominación del organismo del Departamento de Defensa de Estados Unidos creado en 1958 como consecuencia tecnológica de la llamada Guerra Fría, y del que surgieron los fundamentos de ARPANET, red que dio origen a Internet.
- 3 **Internet:** Es una red de redes a escala mundial de millones de computadoras interconectadas con el conjunto de protocolos TCP/IP. También se usa este nombre como sustantivo común y por tanto en minúsculas para designar a cualquier red de redes que use las mismas tecnologías que la Internet, independientemente de su extensión o de que sea pública o privada.
- 4 **Intranet:** Una intranet es una red de ordenadores de una red de área local (LAN) privada empresarial o educativa que proporciona herramientas de Internet.
- 5 **LAN:** (Local Area Network) Red de Area Local.o red direccional de comunicaciones de amplia banda de frecuencias que normalmente opera en una zona geográfica limitada. Su función es permitir a los usuarios de ordenadores comunicarse, compartir el uso de periféricos.
- 6 **HTML:** Acrónimo inglés de Hyper Text Markup Language (lenguaje de marcación de hipertexto), es un lenguaje de marcas diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas Web, basándose en tags¹⁹.
- 7 **XML:** Acrónimo de "EXtensible Markup Language". Es decir, lenguaje de marcas extensible, es de reciente creación (febrero de 1998). XML es un metalenguaje, o sea, sirve para crear lenguajes. Es más amplio, más rico y más dinámico que HTML.
- 8 **XHTML:** Acrónimo inglés de eXtensible Hyper Text Markup Language (lenguaje extensible de marcado de hipertexto), es el lenguaje de marcado pensado para sustituir a HTML como estándar para las páginas web. XHTML es la versión XML de HTML, por lo que tiene, básicamente, las mismas funcionalidades, pero cumple las especificaciones, más estrictas, de XML.
- 9 **CSS:** (Cascade Style Sheet) Hoja de Estilo en Cascada. Conjunto de instrucciones escritas en HTML que definen las apariencias de una página Web con el objetivo de que sus estilos se parezcan.
- 10 **XSLT: XSLT o Transformaciones XSL** es un estándar que presenta una forma de transformar documentos XML en otros e incluso a formatos que no son XML.
- 11 **XMLHttpRequest (XHR):** Es un conjunto de APIs,interfaz de programación de aplicaciones, que pueden ser usadas por JavaScript y otros lenguajes para transferir XML u otra información textual

hacia o desde un servidor Web usando HTTP²⁰, mediante el establecimiento de un canal de comunicación independiente entre páginas Web del lado del cliente y del lado del servidor. El mayor avance de XMLHTTP es la habilidad de actualizar dinámicamente una página Web sin recargar la página entera o usando plugins.

- 12 **PostgreSQL:** Es un servidor de base de datos relacional libre, liberado bajo la licencia BSD.
- 13 **RUP:** El Proceso Racional Unificado o RUP (Rational Unified Process), es un proceso de desarrollo de software.
- 14 **UML:** Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modelling Language) es el lenguaje de modelado de sistemas de software más conocido en la actualidad.
- 15 **Lógica de negocio:** Se denomina capa de negocio (e incluso de lógica del negocio) pues es aquí donde se establecen todas las reglas que deben cumplirse. Recibiendo las peticiones del usuario y enviando las respuestas tras el proceso.
- 16 **GoF:** En español significa la pandilla de los cuatro, formada por Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides., quienes recopilaron y documentaron 23 patrones de diseño aplicados usualmente por expertos diseñadores de software orientado a objetos.
- 17 **Notación PascalCasing:** Los identificadores y nombres de variables, métodos y funciones están compuestos por múltiples palabras juntas iniciando cada palabra con letra mayúscula. Ejemplo: NotacionPascalCasing.
- 18 **Notación CamelCasing:** Los identificadores y nombres de variables, métodos y funciones están compuestos por múltiples palabras juntas iniciando cada palabra con letra mayúscula excepto la primera palabra que debe iniciar con minúscula. Ejemplo: notacionCamelCasing.
- 19 **Tags:** Instrucciones que le dicen al texto como deben mostrarse y atributos (parámetros que dan valor al tag).
- 20 **HTTP:** Es el protocolo de la Web (WWW), usado en cada transacción. Las letras significan Hyper Text Transfer Protocol, es decir, protocolo de transferencia de hipertexto. El hipertexto es el contenido de las páginas web, y el protocolo de transferencia es el sistema mediante el cual se envían las peticiones de acceder a una página Web, y la respuesta de esa Web.

ANEXOS

Anexo 1 Diagramas de Secuencia.

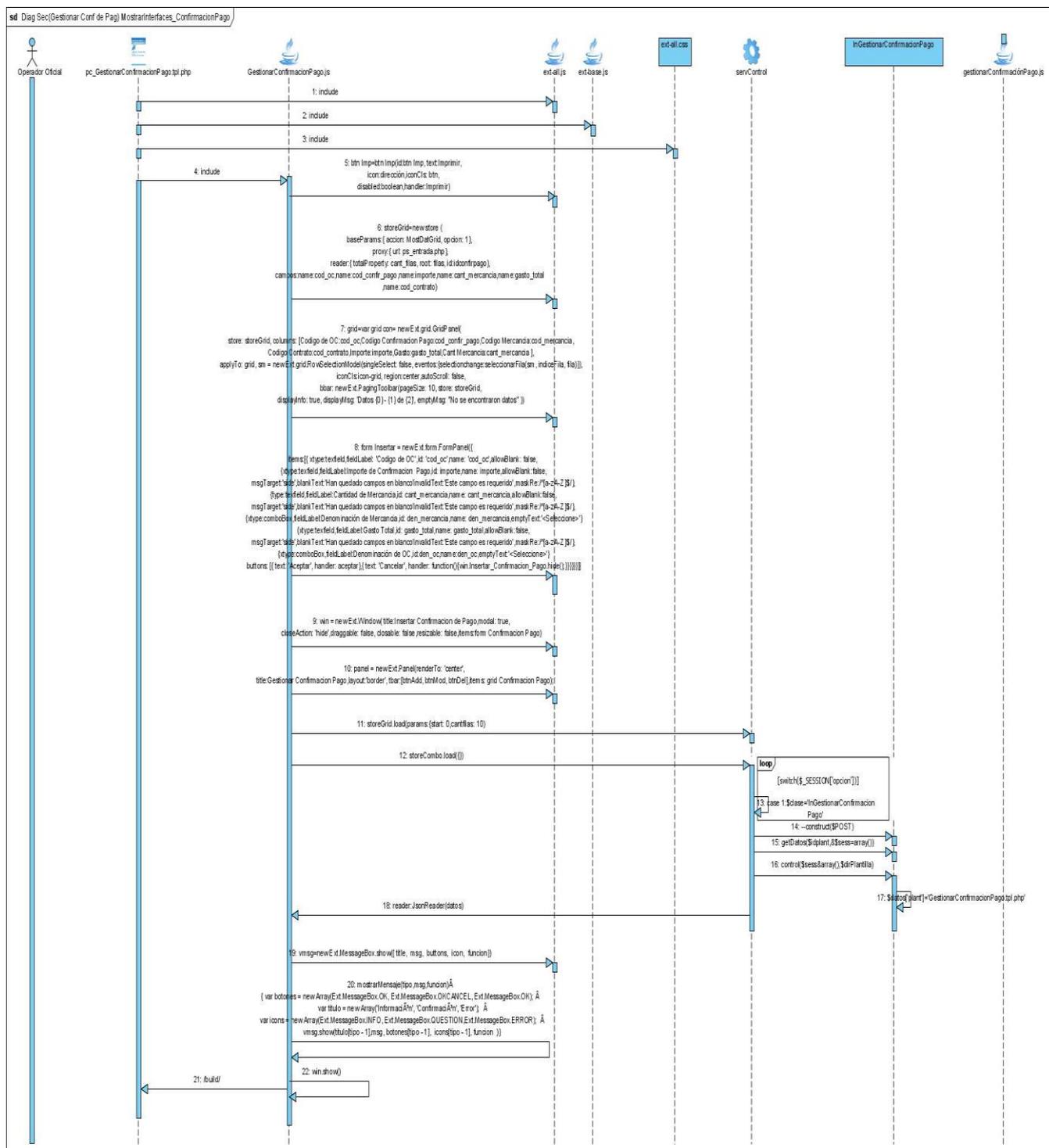


Figura 37 Diagrama de secuencia “Realización Mostrar Interfaz (Caso de Uso Gestionar Confirmación Pago)”.

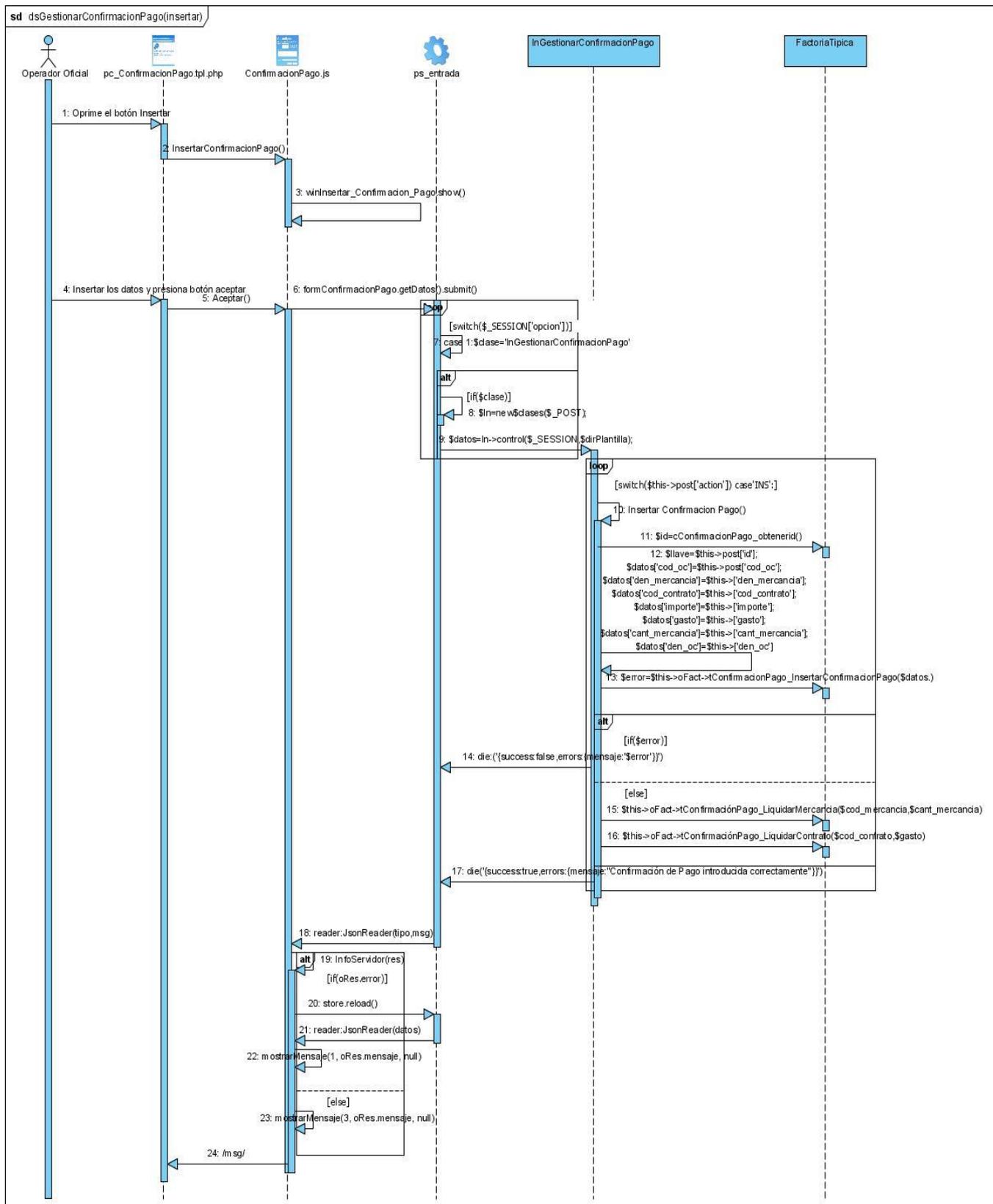


Figura 38 Diagrama de secuencia “Realización Insertar (Caso de Uso Gestionar Confirmación Pago)”.

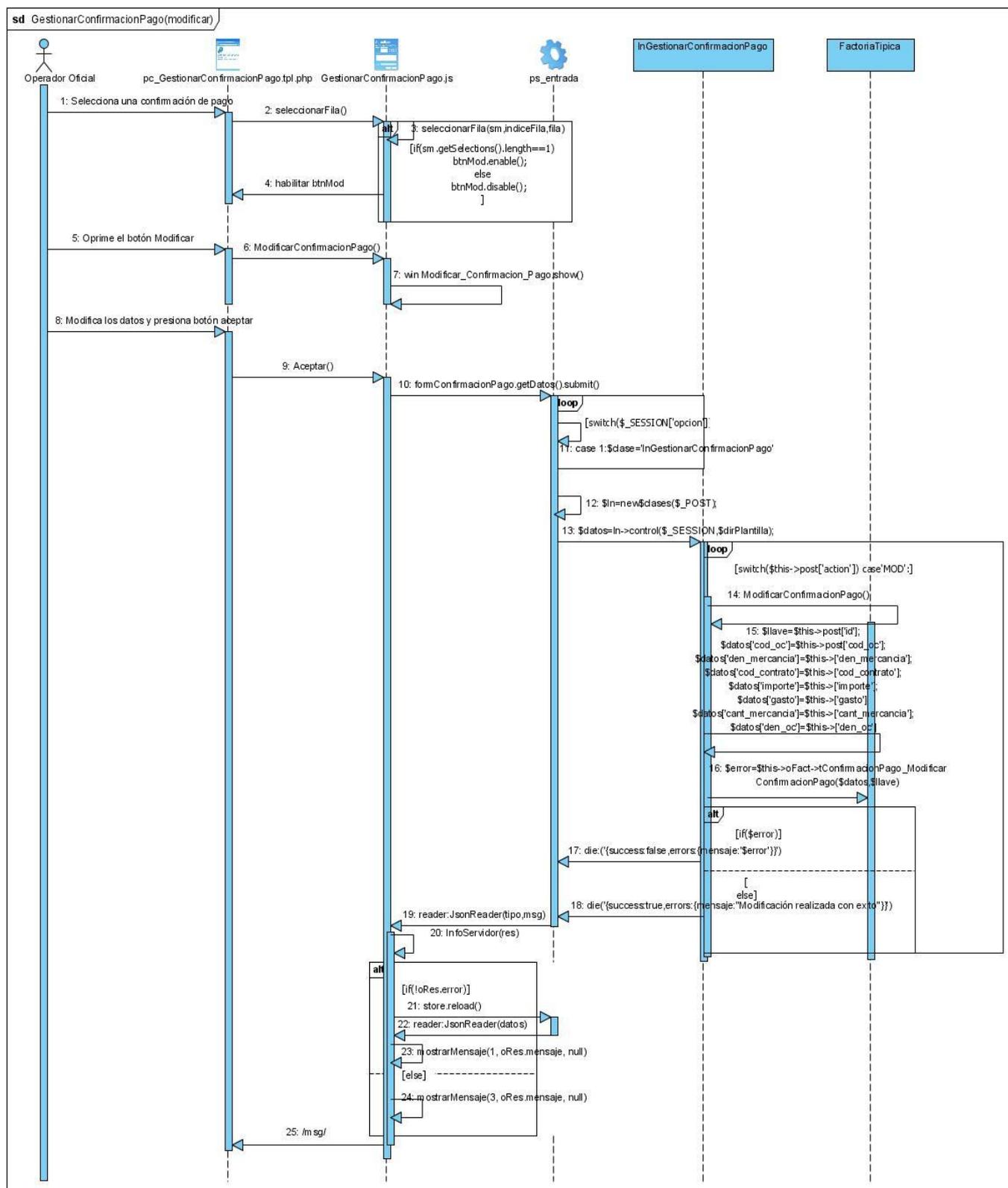


Figura 39 Diagrama de secuencia “Realización Modificar (Caso de Uso Gestionar Confirmación de Pago)”.

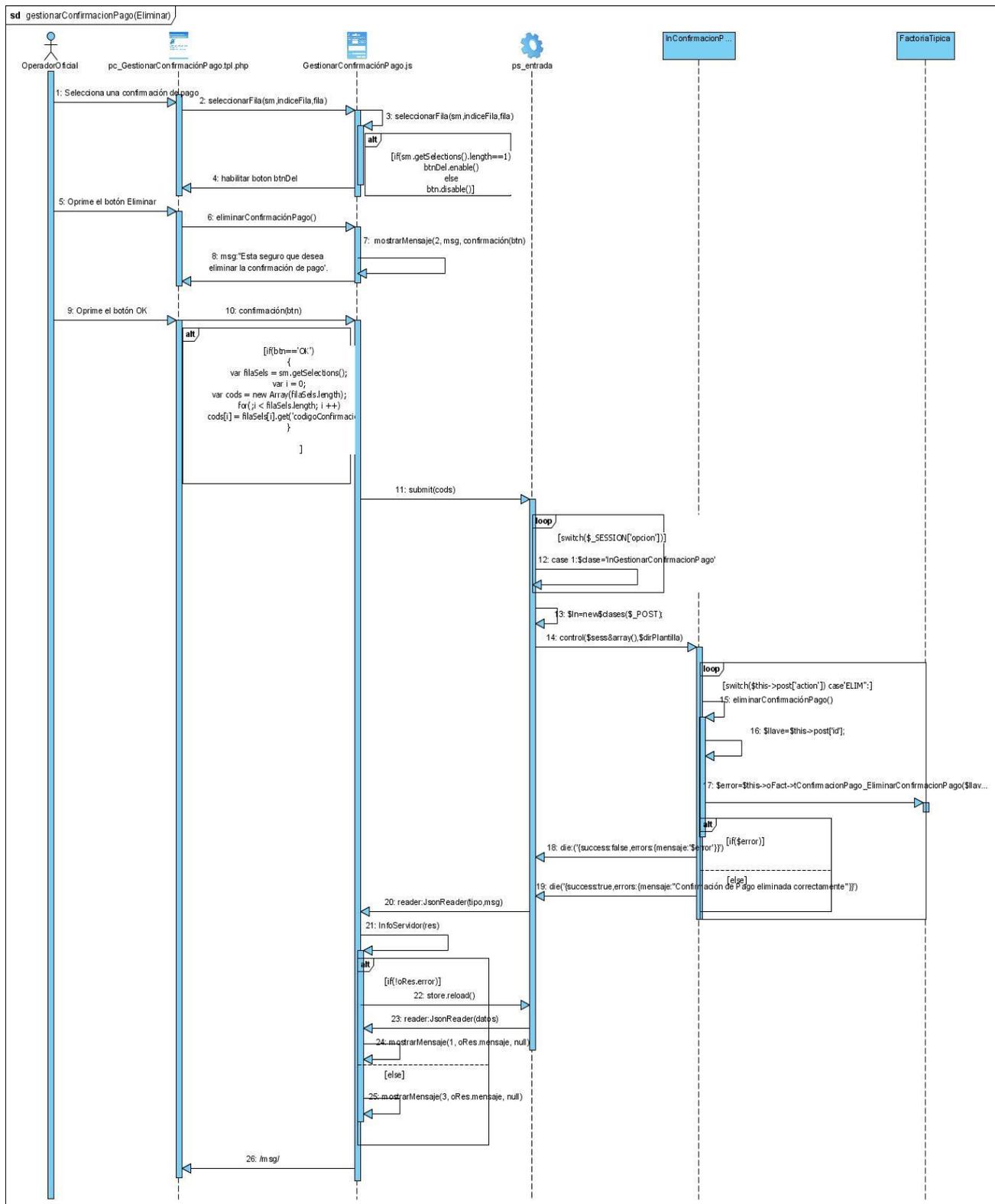


Figura 40 Diagrama de secuencia “Realización Eliminar (Caso de Uso Gestionar Confirmación de Pago)”.

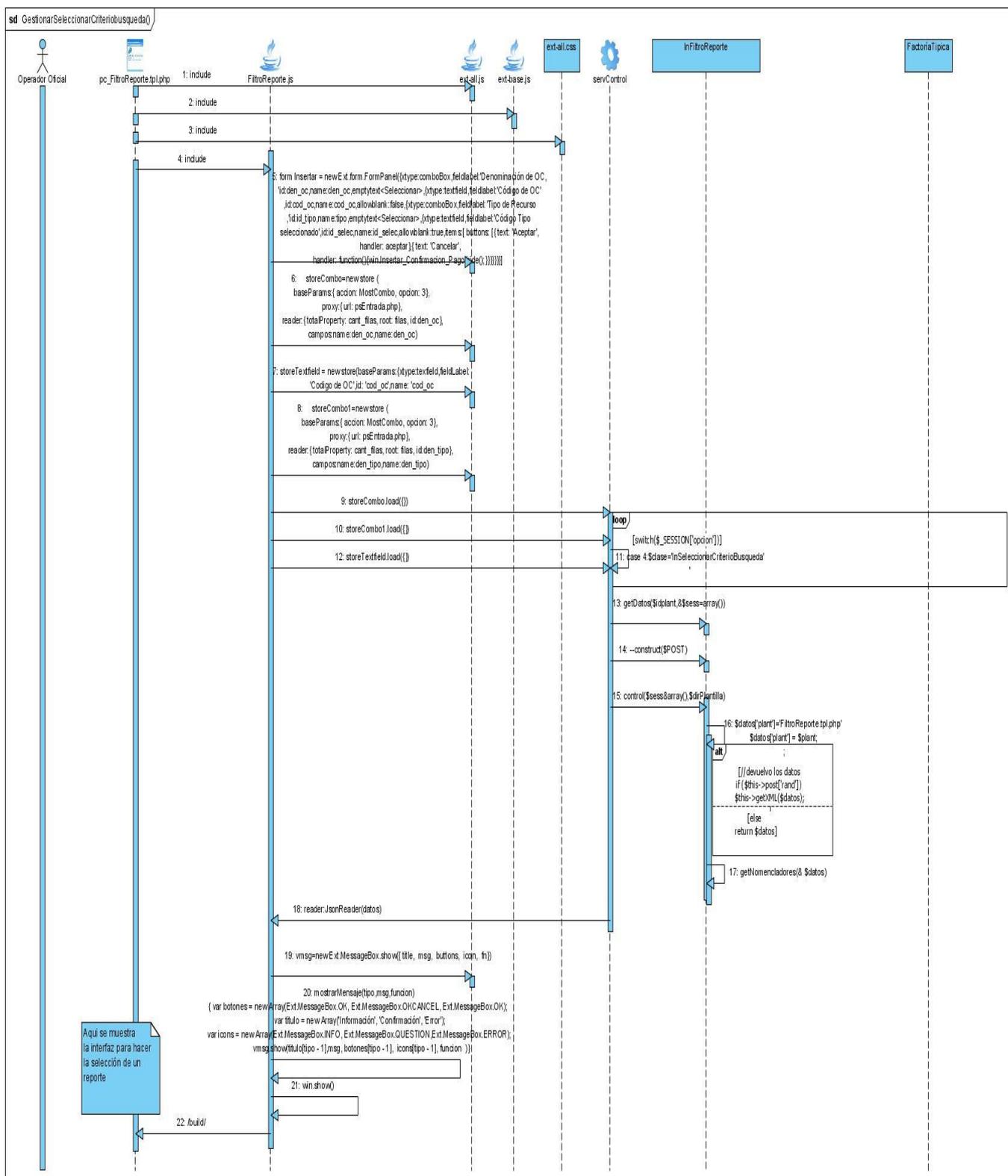


Figure 41 Diagrama de secuencia “Realización de Caso de Uso “Seleccionar Criterio Búsqueda”.

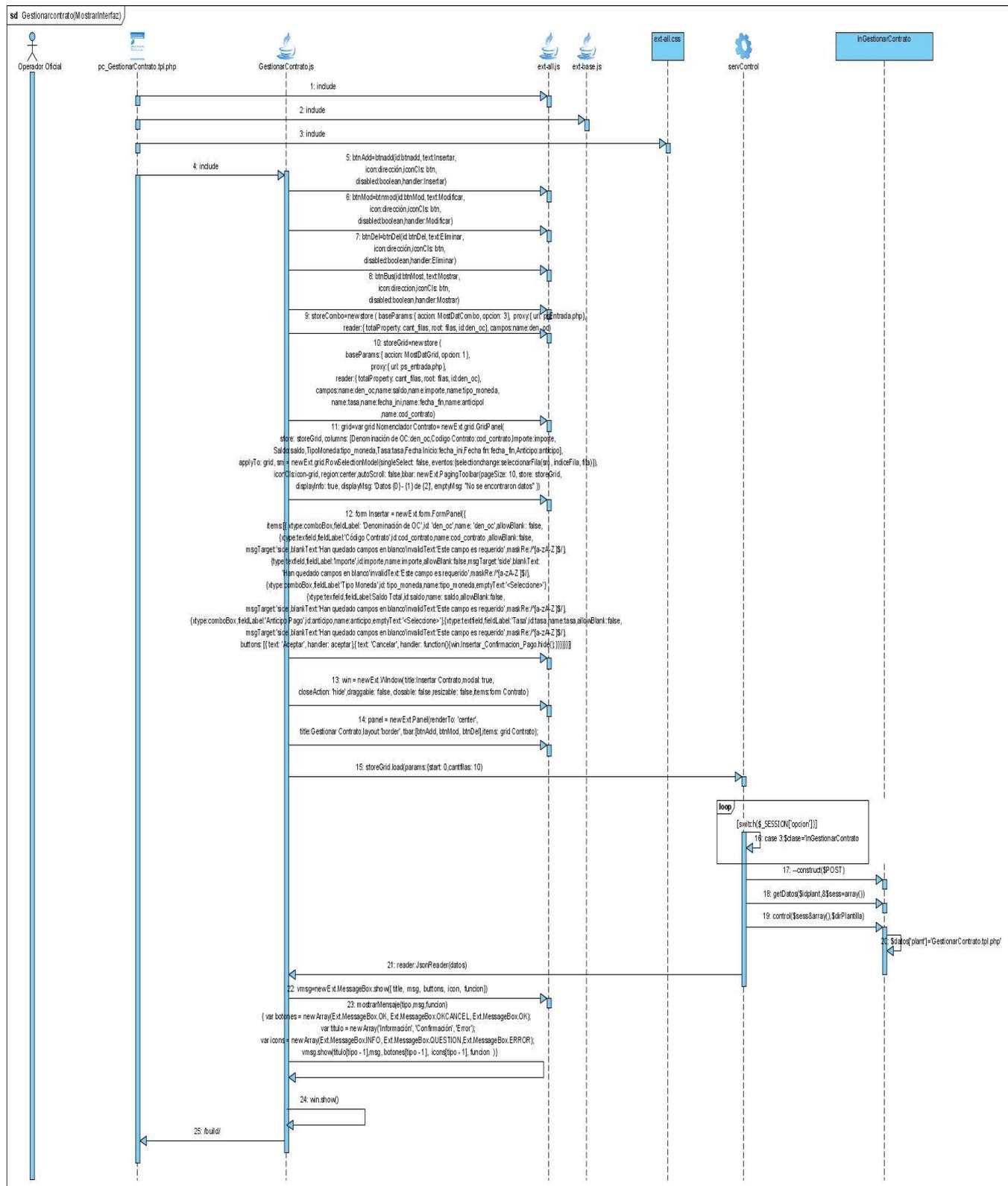


Figura 42 Diagrama de secuencia “Realización Mostrar Interfaz (Caso de Uso Gestionar Contrato)”.

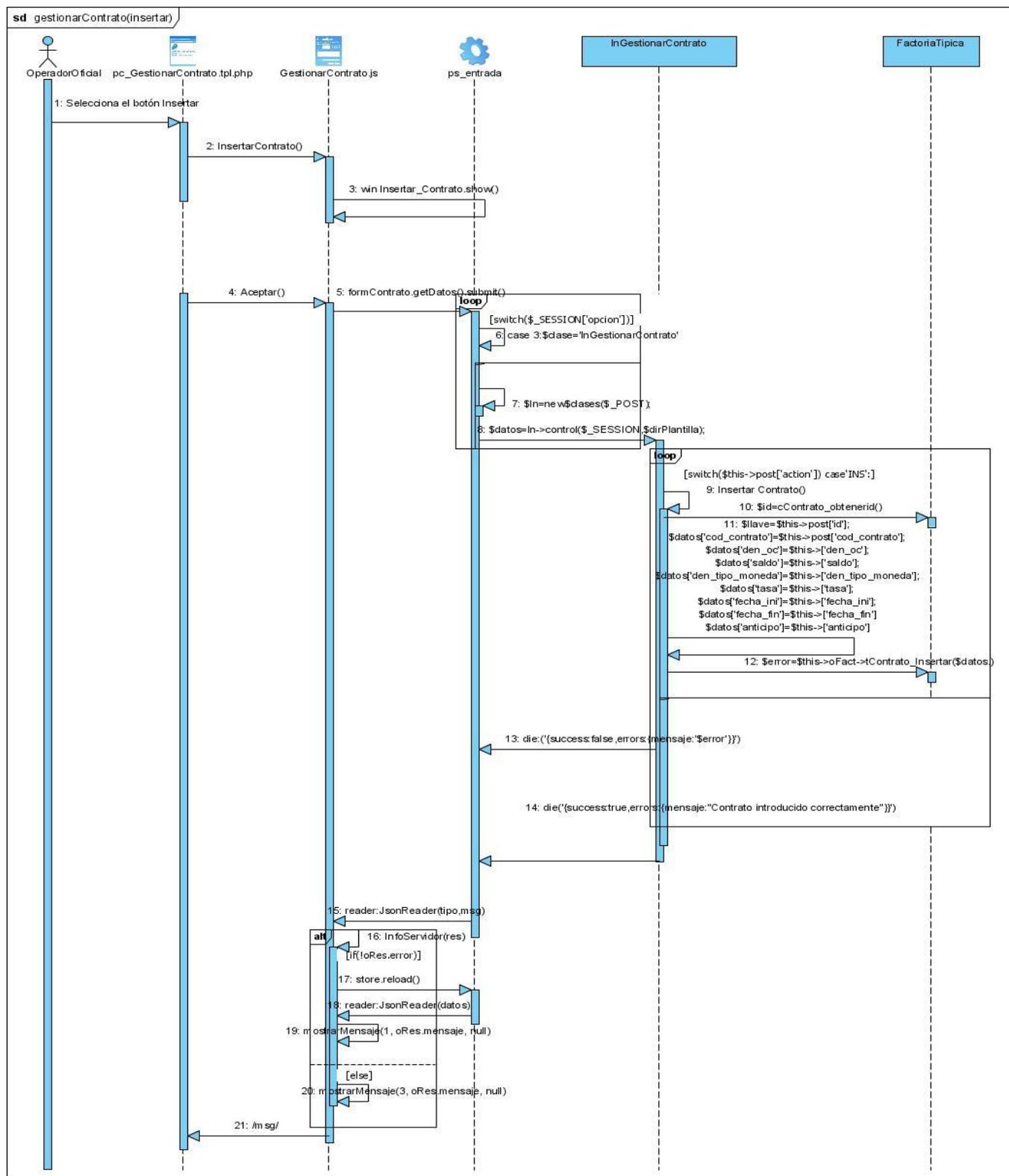


Figura 43 Diagrama de secuencia “Realización Insertar (Caso de Uso Gestionar Contrato)”.

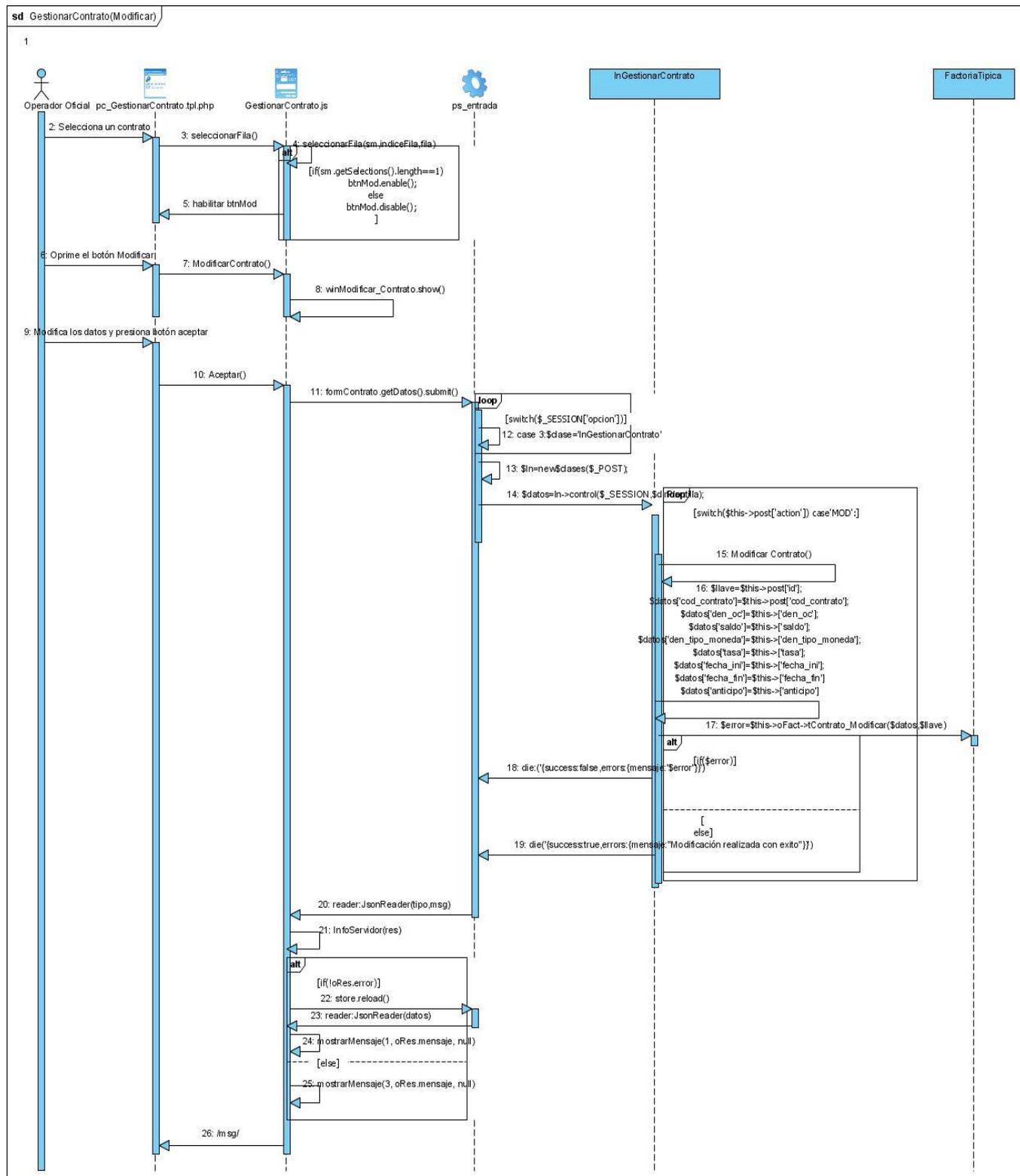


Figura 44 Diagrama de secuencia “Realización Modificar (Caso de Uso Gestionar Contrato)”.

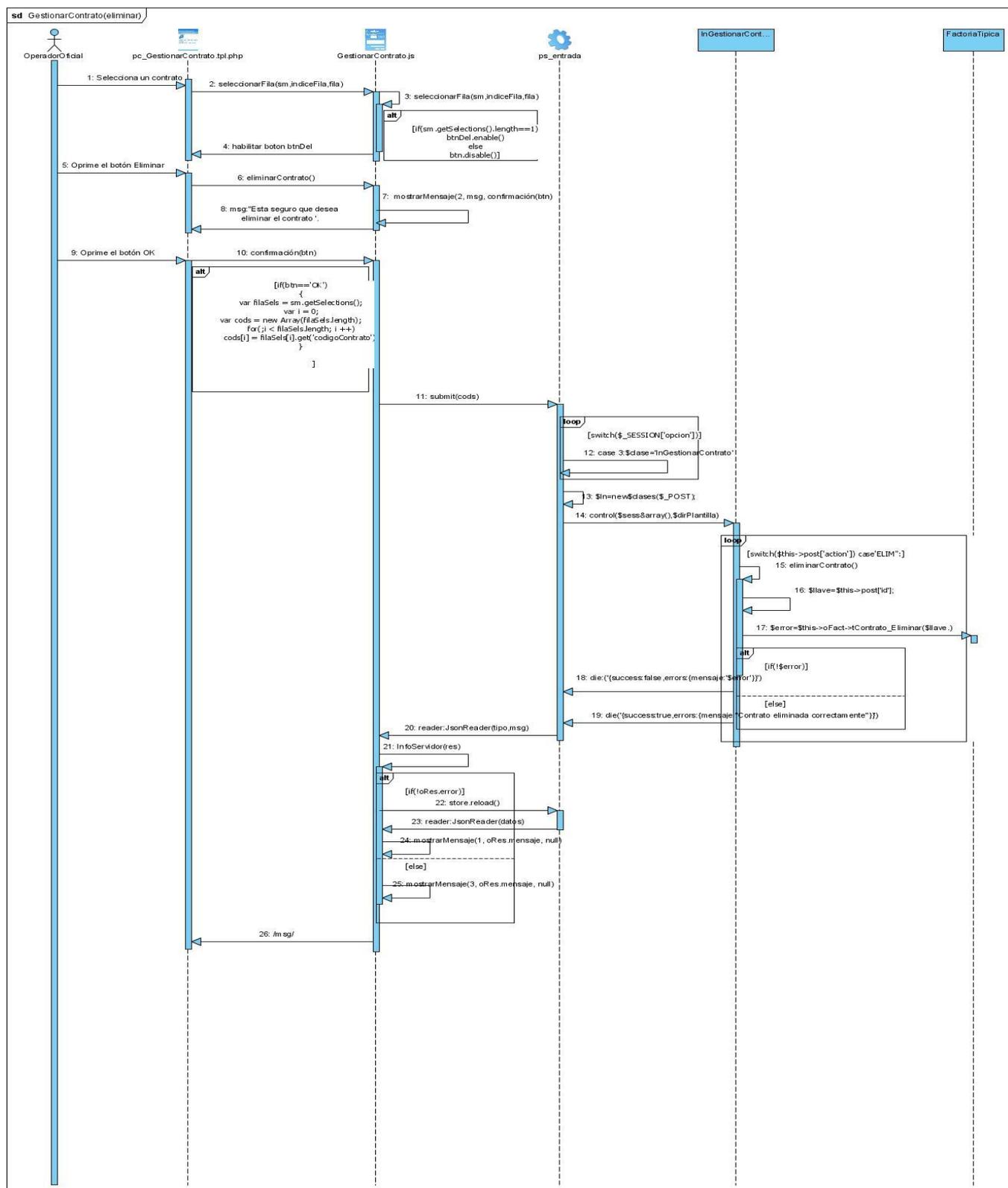


Figura 45 Diagrama de secuencia “Realización Eliminar (Caso de Uso Gestionar Contrato)”.

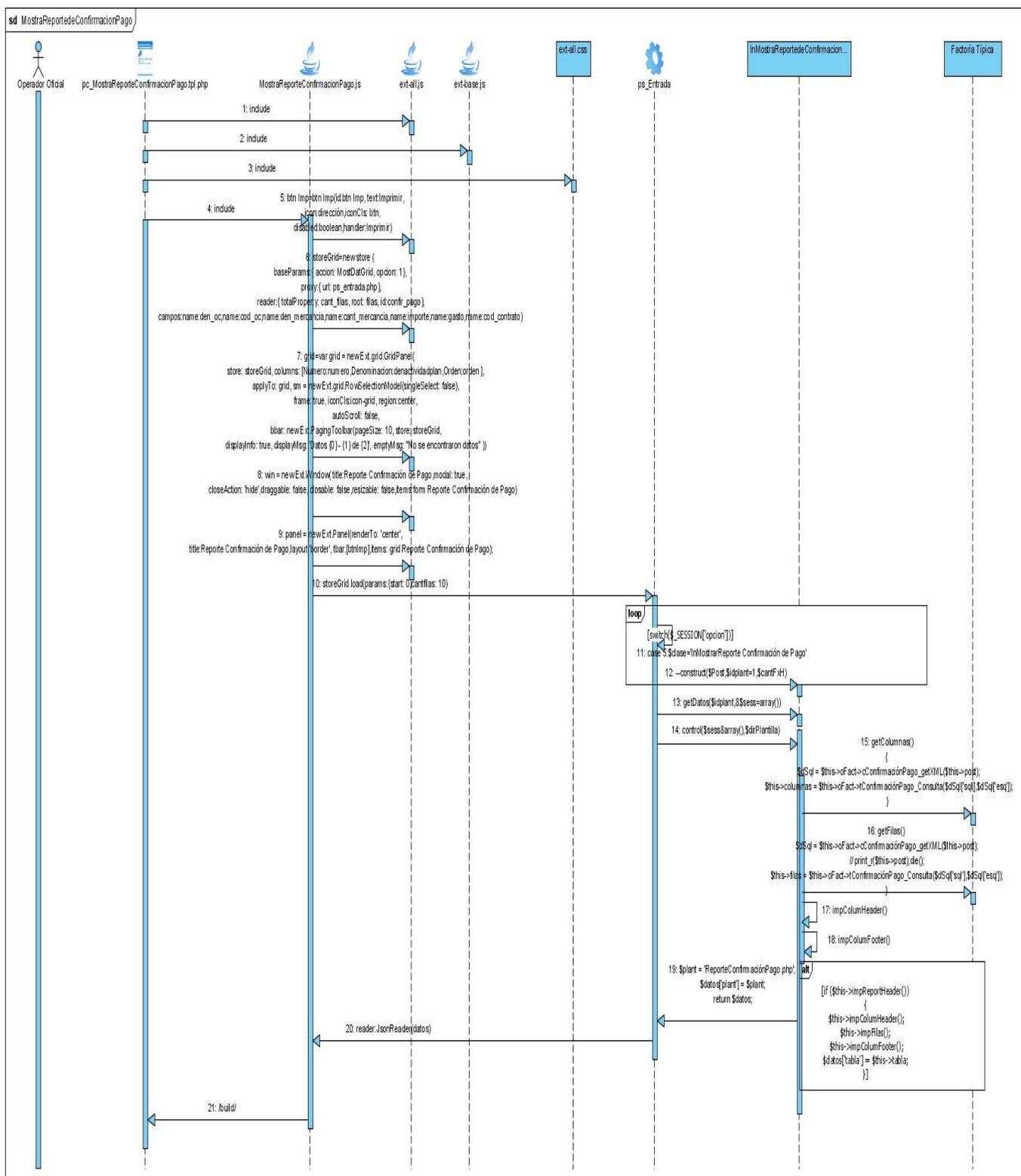


Figura 46 Diagrama de secuencia “Realización Caso de Uso Mostrar Reporte de Confirmación Pago”.

Anexo 2 Diagrama de Componentes

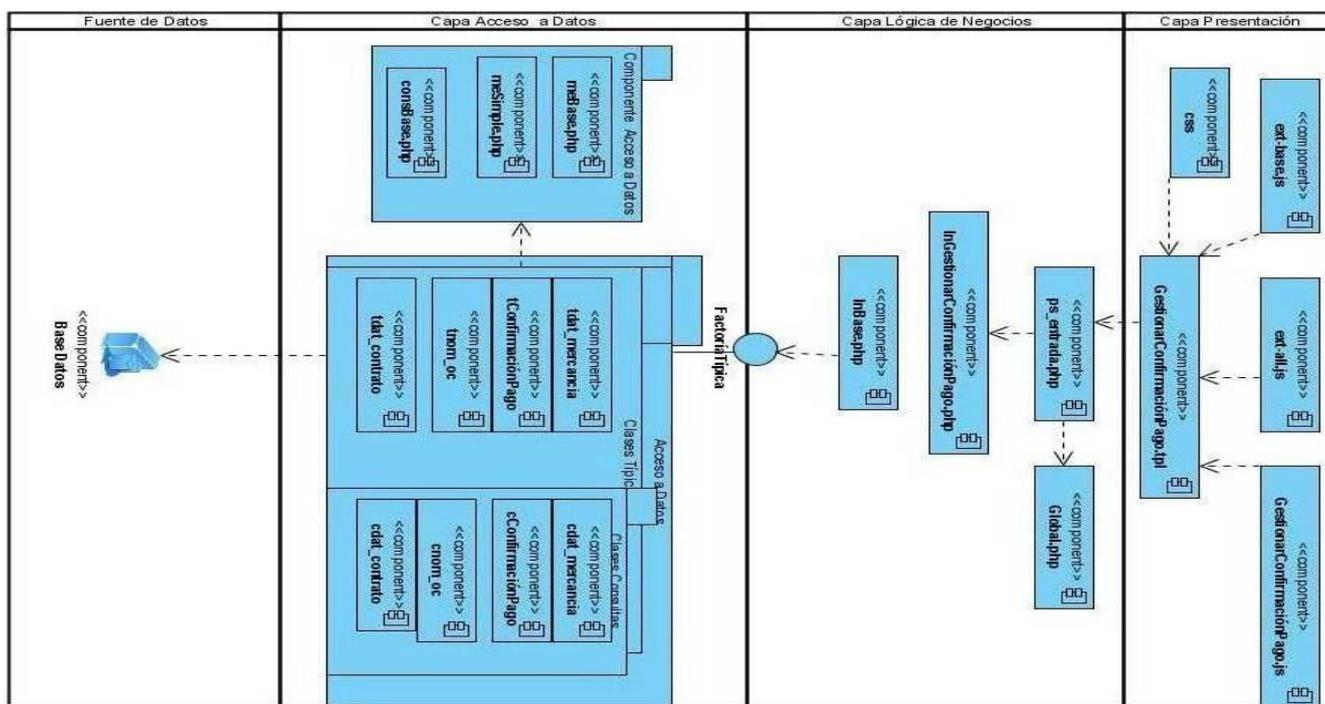


Figura 47 Diagrama de Componentes (Caso de Uso Gestionar Confirmación Pago).

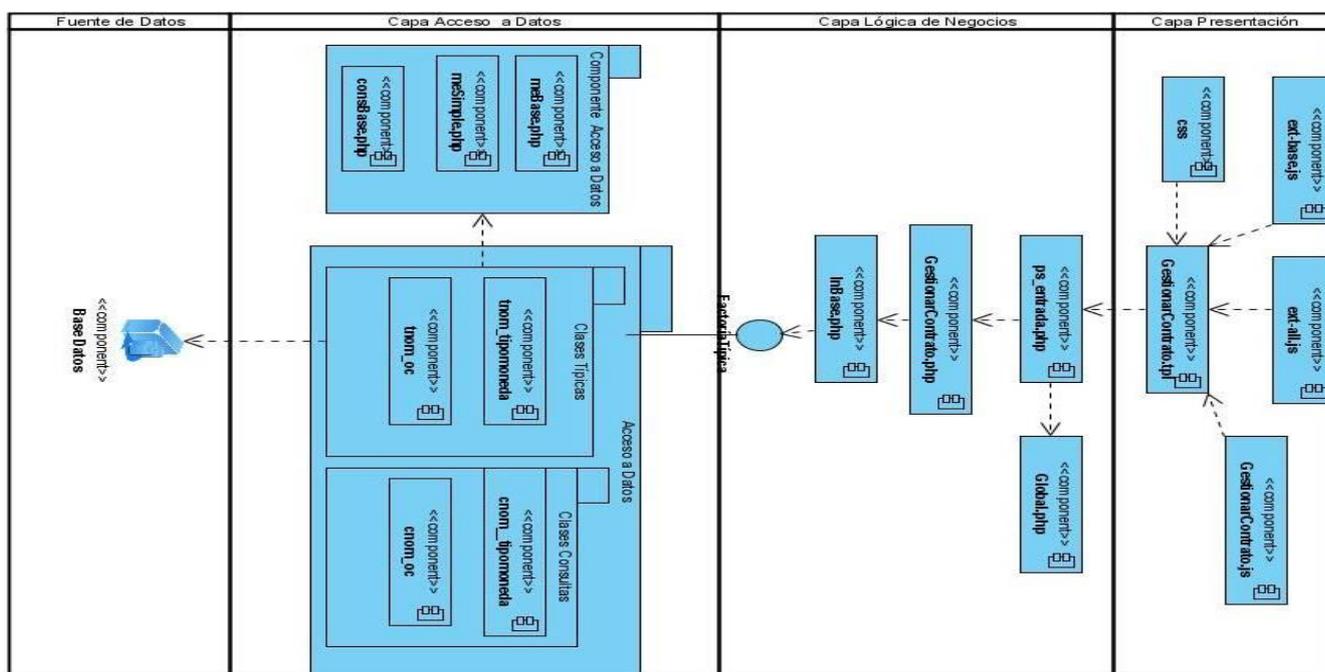


Figura 48 Diagrama de Componentes (Caso de Uso Gestionar Contrato).

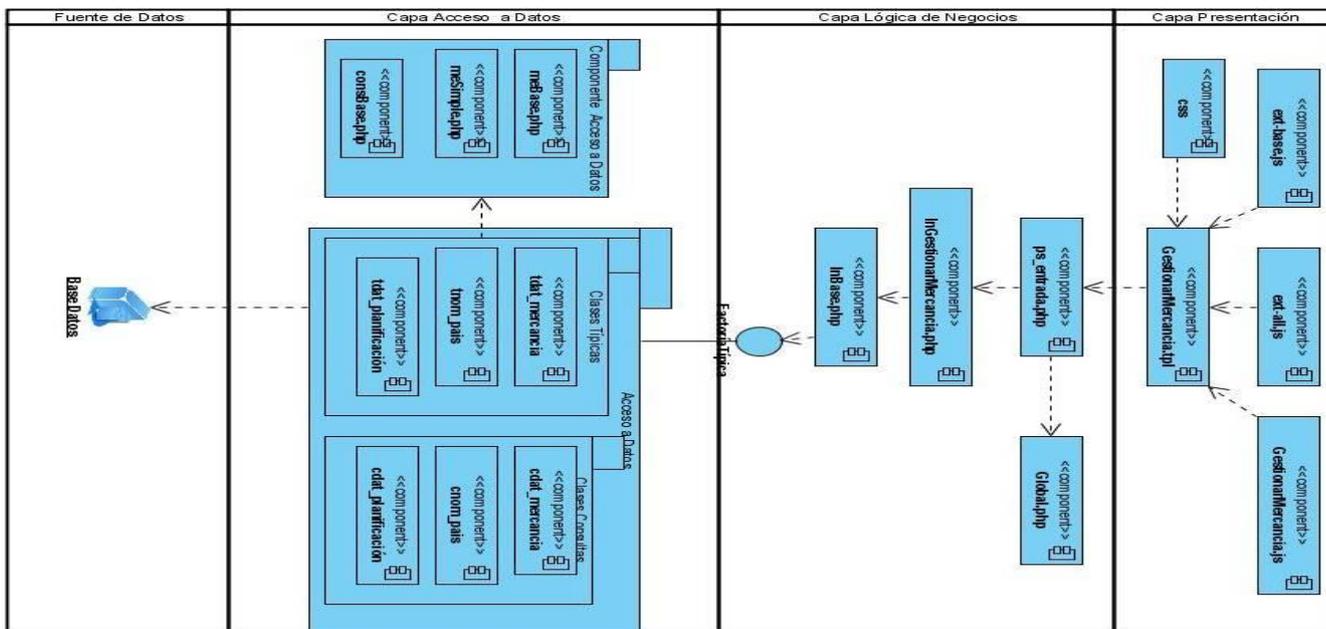


Figura 49 Diagrama de Componentes (Caso de Uso Gestionar Mercancía).

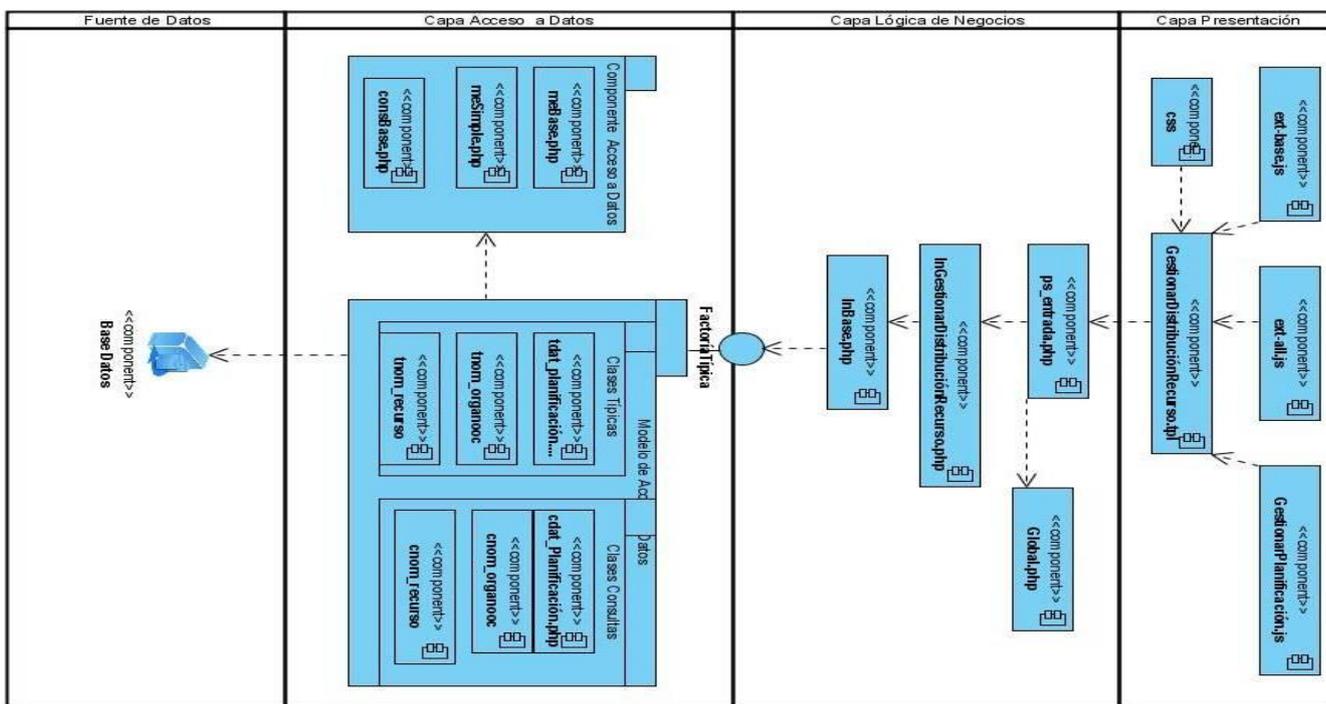


Figura 50 Diagrama de Componentes (Caso de Uso Gestionar Distribución Recursos).