



# Sistema de Reserva, Factura y Coordinación para el Acuario Nacional de Cuba

---

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

---

**Autora:**

Coralí Triana González

**Tutores:**

Ing. Lisandra Gibert Estrada

MSc. Francisco Longino Franquiz

**Co-Tutores:**

Ing. Maikel Pereira Ojeda

Ing. Osay González Fuentes

Ing. Marcel Mesa Hernández

Ing. Daniel Rodríguez Soberats

**Declaración de Autoría**

Declaro ser la autora de la presente tesis y autorizo a la Universidad de las Ciencias Informáticas a hacer uso de ella para lo que necesite, cediéndole de esta forma los derechos patrimoniales de la misma, con carácter exclusivo.

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

**Autores:** Coralí Triana González

\_\_\_\_\_

**Tutores:** Ing. Lisandra Gibert Estrada

\_\_\_\_\_

*A mi madre por serlo todo para mí, mi apoyo, mi guía y mi consejera, por estar siempre para mí en los momentos difíciles y apoyar todos mis planes por muy locos que parezcan.*

*A mi abuela por ser un ejemplo de mujer a seguir para que se sienta orgullosa de su nieta más chiquita.*

*A mi tía Elsa y a Gilbert por siempre estar pendientes de mí y a mis primos Dayana, Marta Beatríz y Jose Luis.*

*A mi papá para que tenga otro motivo para sentirse orgulloso de su hija.*

*A Maikel por ayudarme a ser siempre mejor, por enseñarme a ser independiente, por su amor, por haber tenido esta genial idea y por compartir parte de su vida conmigo.*

*A Jose por ser mi amigo, mi compañero de sueños, mi flaco rico...*

*A Osay por su amistad y apoyo en los momentos donde lo necesité, sin él que no hubiese podido cumplir esta tarea.*

*A Yudanis y Mayliuvis los "machu" y Marcel "el negro" por haberme ayudado y aguantado tanto tiempo.*

*A Daniel y Lisandra mis tutores por su ayuda en el desarrollo de este trabajo.*

*A Longino y principalmente a Haydeé Zamora mi amiga sin ella no lo hubiese logrado, al menos no con calidad, además de siempre facilitarme las cosas.*

*A la gente de Alfaomega pero principalmente a Dunet y Yisel y a los que volví locos cuando no sabía symfony: Alejandro, Orestes, Ernesto Vladimir, Adrian y Yaismel.*

*A todos los de mi grupo anterior los que quedan y los que no, pero especialmente a Olía, Celia, Linet, Karen, Judith, Yadira, Yisel, Yunior, Randy y Pepo.*

## RESUMEN

El tema central del presente trabajo es el desarrollo de un sistema de gestión de información para el control de los procesos de reservación, coordinación y factura para solucionar problemas de eficiencia, accesibilidad y seguridad en el tratamiento de la información que presenta la Oficina de Reserva, Factura, Coordinación y Gestión de Cobro del Acuario Nacional de Cuba. Para esto se realiza un estudio de las tendencias actuales en el desarrollo de este tipo de sistemas; además, se hace el análisis y selección de las herramientas y tecnologías más adecuadas para este en particular. Asimismo, se analizan las metodologías de desarrollo de software, los lenguajes de modelado, las herramientas de autor, así como los lenguajes de programación, y ello permite justificar las selecciones hechas para el desarrollo de la aplicación. Se lleva a cabo la implementación del software a partir de los artefactos obtenidos del análisis y el diseño y finalmente se describe el estudio la factibilidad y beneficios en términos de tiempo y dinero ahorrado al país. Se espera que con la puesta en práctica del “Sistema para la Reserva, Factura y Coordinación” se beneficie de forma en general el accionar de la oficina comercial, además de una reducción considerable de la redundancia de la información guardada, rapidez en las consultas de información, registro y consulta de trazas de modificación, entre otras.

## Índice

Introducción .....	1
Capítulo 1 . Fundamento teórico.....	7
1.1 Introducción .....	7
1.2 Productos similares .....	7
Sistemas de reservación a nivel internacional.....	7
Sistemas de reservación a nivel nacional .....	8
1.3 Valoración de los sistemas de referencia .....	8
1.4 Proceso de desarrollo .....	9
Proceso Unificado de Desarrollo .....	9
Extreme Programming (XP) .....	11
SCRUM.....	12
Agile Unified Procces .....	12
Selección de la metodología para guiar el desarrollo del software.....	14
1.5 Lenguaje de modelado .....	14
Principales características de UML.....	15
1.6 Lenguajes y tecnologías para el desarrollo .....	15
Tecnologías del lado del cliente.....	16
Lenguaje del lado del servidor .....	18
1.7 Frameworks.....	18
Capa de presentación .....	18
Capa de lógica de negocio .....	19
Capa de acceso a datos.....	20
1.8 Sistema gestor de base de datos .....	20
1.9 Servidor Web .....	21
1.10 Entorno de desarrollo .....	21
1.11 Herramientas CASE para el modelado UML .....	22
ArgoUML.....	22
Visual Paradigm.....	22
Selección de la herramienta CASE .....	23
1.12 Conclusiones .....	23
Capítulo 2 . Propuesta de solución.....	24
2.1 Funcionamiento del negocio y elementos de automatización.....	24
Realizar una reservación .....	24
Confirmar una reservación.....	25
Facturar una reservación .....	25
Mejoras esperadas a partir de la puesta en marcha del sistema .....	25
2.2 Modelo de Dominio .....	26
2.3 Lista de conceptos modelados en el Dominio y definiciones correspondientes .....	26
2.4 Lista de requisitos .....	27

---

Requisitos funcionales: .....	27
Requisitos no funcionales .....	33
2.5 Modelo de Casos de Uso del Sistema .....	35
2.6 Descripción de Casos de Uso en formato resumido .....	37
2.7 Prototipos de interfaz de usuario.....	47
2.8 Vista de CU arquitectónicamente significativos.....	49
2.9 Patrones de diseño utilizados en la solución.....	51
2.10 Diagrama de despliegue de la aplicación.....	52
2.11 Diagrama de las clases del diseño utilizando estereotipos web .....	53
2.12 Modelo de Implementación .....	57
4.2.1 Diagrama de componentes .....	57
2.13 Modelo de datos del Sistema .....	61
2.14 Conclusiones .....	62
Capítulo 3 Estudio de factibilidad.....	63
3.1 Método de estimación. ....	63
3.2 Información necesaria. ....	64
3.3 Factores de Complejidad.....	66
3.4 Resultados de la estimación. ....	68
3.5 Conclusiones. ....	69
Conclusiones .....	70
Recomendaciones.....	71
Referencias Bibliográficas .....	72



## Introducción

El Acuario Nacional de Cuba es una de las instituciones recreativas más accesibles al pueblo cubano, y uno de los más prestigiosos centros de investigación de nuestro país. La Revolución Cubana, dentro de sus numerosas acciones en pro del bienestar del pueblo, por elevar su cultura y por el desarrollo de la ciencia y la tecnología en Cuba, ha dedicado muchos años de esfuerzo y recursos a garantizar el progreso y la evolución de esta entidad de referencia.

Un acuario público es un lugar que contiene especies marinas expuestas al público, con fines educativos, de investigación o comerciales. La idea de crear un acuario en Cuba surgió desde los mismos inicios de la Revolución. Originalmente se creó un Centro Experimental, sin la intención de recibir público ni servir de entretenimiento; el único objetivo de este centro era llevar a cabo investigaciones sobre especies marinas. Sin embargo, comenzaron las visitas debido al interés que despertaban los animales que se hallaban en algunas peceras. Por esta razón, se decidió acondicionar la instalación –que no era más que una casa situada en la Avenida 1era, y calle 60 en Miramar– para abrirla al público formalmente, y el 12 de abril de 1961 se inauguró la llamada "primera ampliación del acuario"; desde este momento la recepción de visitantes se convierte en una de sus razones de existir. La interacción y la atención al público fue adquiriendo cada vez más importancia cuando, a finales de los 70, el Acuario Nacional comenzó sus actividades con mamíferos marinos, los cuales llegaron a ser la mayor atracción en sus espectáculos. El número de visitantes en 1991 llegó a ser de aproximadamente un millón de cubanos.

Por esta razón, surge la necesidad de establecer un procedimiento de reservaciones mejor respaldado y nace lo que hoy es la oficina de reserva, coordinación, factura y gestión de cobro del Acuario Nacional de Cuba, en el sector comercial. Este departamento es el encargado de tramitar las reservaciones (ya sean para espectáculos, de alquiler o gastronómicas), además de coordinar los recursos de la institución para que puedan efectuarse las visitas, facturarlas y negociar el pago.

A esta oficina se dirigen los clientes cuando desean efectuar una reservación; estos son atendidos por dos técnicos en promoción y comunicación integral que se encargan de mostrarles las ofertas y aclarar cualquier duda que tenga el cliente, para que finalmente reserve, de acuerdo con su elección. Esta reservación queda guardada en un documento Excel que se almacena en un servidor, una semana antes de efectuarse la visita el cliente debe llamar para confirmar. Todos los domingos se hace un cronograma con las actividades o reservaciones confirmadas de la semana, el





cual se muestra en una pizarra para que todo el personal esté informado del programa que se brindará cada día. Toda la coordinación de los activos (locales), la gastronomía y los espectáculos se hace internamente, vía telefónica. La reservación se puede facturar a partir de que se confirme y se imprimen copias, de modo que cada entidad conserve una; estas se archivan después de haberse efectuado el pago.

La oficina tiene buenos resultados en el cumplimiento de su función, pero para ser más eficiente se necesita un mejor aprovechamiento de las nuevas tecnologías de la informática y las comunicaciones (TIC), en su quehacer diario, lo que contribuiría a solucionar las siguientes dificultades:

- ✓ El procesamiento de la información de las reservaciones y la coordinación de recursos muestran inconvenientes, debido a que una parte de ella se encuentra en archivos compartidos, en formato digital, ubicados en una red informática de clientes ligeros; y la otra parte se halla en formato físico, su tratamiento e intercambio se dificulta y no siempre se encuentra a disposición de quienes deben acceder a ella.
- ✓ La seguridad de esta información es escasa, ya que cualquier persona con acceso a una PC (Computadora Personal) de la red de clientes ligeros podría, en determinada situación, hacer cambios significativos en los datos (eliminar o modificar los archivos relacionados con la facturación) sin que queden registros de estos.
- ✓ Los directivos, a la hora de presentar informes a los superiores o planificar estrategias en función de los resultados, no cuentan con retroalimentación, ya que su estilo de trabajo no les brinda las facilidades en el procesamiento de los datos, como estadísticas y gráficos, lo que dificulta la toma de decisiones.
- ✓ El estado actual de los procesos de reservación y planificación de locales y recursos se lleva de manera manual, y se informa telefónicamente, por lo que no quedan registros de la comunicación y pueden existir equivocaciones.
- ✓ Los activos de hardware y tecnología de la empresa están subutilizados, pues los usuarios se limitan a utilizar los productos de ofimática y consumir información de la intranet, cuando hay numerosas aplicaciones que podrían mejorar la calidad del servicio al público, unido a las condiciones de los propios trabajadores de la institución.

Mencionadas las dificultades fundamentales y con la intención de solucionarlas se plantea la siguiente interrogante como el **problema de** la presente **investigación**:



¿Cómo contribuir a la mejora de los procesos que se realizan en la oficina de reserva, coordinación, factura y gestión de cobro del Acuario Nacional de Cuba?

Para solucionar este problema se establece, como objeto de estudio: los procesos de reserva y facturación de acuarios.

El **campo de acción** en que se enmarca el trabajo son los procesos de la Oficina de Reserva, Coordinación, Factura y Gestión de Cobro del Acuario Nacional de Cuba.

Con el propósito de encontrar una solución al problema planteado se define como **objetivo general**: Implementar un sistema informático de gestión de información para la oficina de reserva, coordinación, factura y gestión de cobro del Acuario Nacional, que permita mejorar la eficiencia y seguridad de sus procesos, y la accesibilidad y confiabilidad de sus datos, proporcionándole un mejor aprovechamiento a las TIC.

Este objetivo general se puede desglosar, para su mejor comprensión, en los siguientes objetivos específicos:

- ✓ Caracterizar los procesos principales del negocio e identificar mejoras potenciales.
- ✓ Definir las funcionalidades y características que debe tener el sistema.
- ✓ Implementar una aplicación informática que cumpla con los requerimientos definidos.
- ✓ Documentar el análisis de factibilidad.

Para guiar la investigación se considera la siguiente **idea a defender**: Si se implementa un sistema informático que automatice los procesos de la oficina de reserva, coordinación y gestión de cobro del Acuario Nacional, se contribuirá a la mejora de la eficiencia, accesibilidad, integridad y seguridad en el tratamiento de la información, así como las condiciones de trabajo y calidad de los servicios.

Para dar cumplimiento a los objetivos específicos se planifican las siguientes **tareas**:

- ✓ Análisis del marco teórico de la investigación y determinación de la posición teórica del investigador.
- ✓ Identificación y especificación de las funcionalidades y de los requerimientos no funcionales apoyados parcialmente en la tarea anterior.
- ✓ Realización del análisis y diseño del sistema utilizando estereotipos web.
- ✓ Implementación de las clases definidas en la tarea asociada al diseño del sistema.



- ✓ Implementación de la estructura de almacenamiento persistente, el acceso a esta y su actualización.
- ✓ Documentación del estudio de costo.

### **1. Métodos investigativos**

Para llevar a cabo la investigación, se emplearon los métodos del nivel teórico y el empírico.

#### Métodos teóricos:

- ✓ *Análisis histórico-lógico:* Se utilizó este método para realizar el estudio del estado del arte, o sea, para investigar acerca de otras aplicaciones o soluciones similares y de los lenguajes y metodologías de desarrollo de software existentes, así como los frameworks y herramientas de desarrollo; también para describir la metodología, las herramientas y los lenguajes a utilizar en el análisis, diseño e implementación del sistema de apoyo al trabajo en la oficina.
- ✓ *Análisis y síntesis:* Se identificaron y observaron todas las partes o sectores relacionados en el negocio (Sector Comercial, Sector de Gastronomía y Sector de Economía) y se estudiaron individualmente, para finalmente después de reunirlos entender cómo funciona la organización en su totalidad.

#### Métodos empíricos:

- ✓ *Entrevista:* El uso de esta técnica de recopilación de información permitió, mediante una conversación profesional con los trabajadores de la institución, adquirir información acerca de los procesos que se llevaban a cabo en la oficina.

### **2. Resultados esperados:**

Con el desarrollo del presente trabajo de diploma se pretende lograr:

- ✓ La implementación del sistema de reserva y coordinación para el Acuario Nacional de Cuba.
- ✓ Un mejor aprovechamiento de la tecnología en la oficina comercial del Acuario Nacional de Cuba.



- ✓ Una mayor fiabilidad, seguridad y velocidad en el manejo de los datos asociados a las reservas y a la coordinación de eventos y recursos.
- ✓ La documentación referente al desarrollo del sistema para soporte futuro y transferencia tecnológica.



### 3. Estructura de los capítulos

**Capítulo 1:** Fundamento teórico. Se exponen los elementos teóricos que sustentan el problema científico y los objetivos del trabajo. Se realiza un análisis de las metodologías y herramientas de desarrollo que se pueden utilizar y se justifica la selección de cada una de ellas.

**Capítulo 2:** Características del sistema. Se describen los procesos de negocio actuales, y se proponen adaptaciones, que más adelante garantizarán la consecución de los objetivos planteados. Se especifican los requisitos que debe cumplir el sistema, así como el diagrama de casos de uso y las descripciones de ellos. Se realiza el diseño del sistema, así como los diagramas de clases con estereotipos web que brindan una visión clara de los componentes del producto y se generan otros artefactos. Se describe cómo está implementado el sistema.

**Capítulo 3:** Análisis de factibilidad. Se documenta la aplicación del método de estimación definido en Calisoft para los proyectos productivos de la universidad al sistema propuesto, con el objetivo de estimar los beneficios que aporta, principalmente económicos, la implantación del sistema.



## Capítulo 1 . Fundamento teórico

### 1.1 Introducción

Al iniciarse el desarrollo de un producto de software, los primeros pasos están en función de definir el tipo de aplicación y su ambiente de desarrollo, por lo que se llevará a cabo el estudio de las soluciones similares que ya existen en la gestión de información, en lo relativo a reservaciones, facturas y estadísticas. También se estudiarán cuestiones referentes a las tecnologías y herramientas que se usan para dar cumplimiento al objetivo general de la investigación. Ello se realiza con el fin de tomar ideas y conocer los patrones de conducta de estas herramientas, de forma que la investigación aporte una guía para el desarrollo del Sistema para la Reservación y Coordinación del Acuario Nacional de Cuba.

### 1.2 Productos similares

El desarrollo de sistemas de gestión de información y las soluciones informáticas, en general, tienen muchas cosas en común, o sea, puntos de coincidencia. Por eso es tan importante valorar los productos en determinada área, en busca de que exista ya la solución al problema que se intenta resolver, o que sirvan de referencia para el desarrollo de la nueva solución. En este sentido, se estudiaron los principales sistemas asociados al objeto de estudio y se seleccionaron algunos para plasmar en este documento.

#### Sistemas de reservación a nivel internacional

- Walt Disney World ® -Reservation Landing

Sistema de reservas para el Mundo del Walt Disney, incluye reservas gastronómicas, de habitaciones para pasar temporada, visitas guiadas, espectáculos, cursos, entre otros. Este sistema cumple una gran parte de los requisitos planteados por el cliente; pero se debe destacar que sería imposible su uso, debido a que ellos tienen todos los derechos reservados sobre el sistema.

Disponible en: <http://bookwdw.reservations.disney.go.com>

- Vacations Made Easy



Sistema de reservaciones para el disfrute de atracciones, shows y parques de diversiones en Orlando, Florida. El paquete completo de vacaciones incluye la reserva del hospedaje. Se puede afirmar que cumple parcialmente con los requisitos planteados por el cliente, lo que lo convierte en un ejemplo a la hora de definir el sistema a implementar, pero está hecho a la medida y los propietarios poseen todos los derechos sobre él.

Disponible en: <http://espanol.vacationsmadeeasy.com/OrlandoFL>

### **Sistemas de reservación a nivel nacional**

- VÍAZUL

Es un sistema sencillo, cuya única funcionalidad es realizar las reservaciones de pasajes en ómnibus VÍAZUL para diferentes rutas de destinos. Aunque no cumple con ninguno de los requisitos mencionados por el cliente, se puede tomar como referencia. Resulta atractivo realizar un análisis de cómo se aborda el problema de realizar reservaciones en un horario ya utilizado para un recurso, y además podría ser útil a la hora de diseñar la interacción con el sistema en un futuro, cuando se cumpla el requisito de publicar en internet la funcionalidad de reservar.

Disponible en: <http://www.viazul.com>

- Directorio Turístico de Cuba

Portal dedicado a brindar información de interés sobre la Isla a los turistas además de reservar hospedaje en los diferentes hoteles del país, autos, vuelos y traslados. No cumple con ninguno de los requisitos mencionados por el cliente, pero al igual que el sistema analizado anteriormente, se puede tomar como ejemplo para algunos de los procesos, por la similitud que existe con la propuesta.

Disponible en: <http://www.dtcuba.com>

### **1.3 Valoración de los sistemas de referencia**

Los sistemas internacionales analizados cuentan con un gran cúmulo de funcionalidades que incluyen, parcialmente, las solicitadas por el cliente, ejemplo:

Gestión de reservaciones para espectáculos.

Gestión de reservaciones para entradas a parques temáticos.



Gestión de reservaciones para alquileres evitando solapamientos.

Por otro lado, los nacionales no cumplen con casi ninguna de estas características. No se encontró ningún sistema que cumpliera, al menos, con las funcionalidades básicas, como la gestión de reservas y la coordinación de los recursos y activos asociados a esta. De hecho, los sistemas que más se asemejan a lo que queremos lograr son privados, ya que los de códigos abiertos encontrados se dedican por lo general a reservas para el hospedaje en hoteles. Por ello se decidió implementar una solución específica, de acuerdo con los requisitos funcionales planteados.

#### **1.4 Proceso de desarrollo**

Los últimos años, con la especialización y el desarrollo creciente de software, han traído consigo el auge de las metodologías de desarrollo, en el afán de perfeccionar el proceso de construcción de un software. Antes, el desarrollo en cascada era la única forma utilizada para ello, lo cual ha cambiado mucho. Dos corrientes, o tipos de metodologías, dominan en la actualidad el desarrollo de software: los llamados métodos pesados y los métodos ligeros. La diferencia fundamental entre ambos es que mientras los métodos pesados intentan conseguir el objetivo común por medio de orden y documentación, los métodos ligeros (también denominados métodos ágiles o adaptativos) tratan de mejorar la calidad del software a través de una comunicación directa e inmediata entre las personas que intervienen en el proceso. Conjuntamente con estas vertientes, existen otros procesos de desarrollo, que toman características distintivas de ambas.

En la actualidad existen una gran variedad de metodologías orientadas al desarrollo de software, entre las que se pueden citar: Proceso Unificado de Desarrollo (RUP), metodología de desarrollo pesada por excelencia; Programación Extrema o *eXtreme Programming* (XP por sus siglas en inglés), de las metodologías ágiles más difundidas; SCRUM, también ágil y que se usa a veces complementada con XP; Microsoft Solution Framework (MSF) para desarrollo ágil; y Agil Unified Procces, una metodología creada por Scott W. Ambler que se puede considerar mixta. Todas estas constituyen metodologías líderes para guiar el proceso de desarrollo de software, razón por la cual cuatro de ellas se estudiarán a continuación.

#### **Proceso Unificado de Desarrollo**

El Proceso Unificado de Desarrollo (RUP, por sus siglas en inglés) es una de las metodologías más generales que existen actualmente; por su finalidad, no está restringida a guiar el desarrollo de software, sino cualquier tipo de proyecto. La estrategia de este proceso es conseguir su objetivo por





medio de orden y documentación, lo que lo convierte en un fiel exponente de los métodos pesados. RUP define cuatro fases (inicio, elaboración, construcción y transición) y dentro de cada una de ellas el equipo de trabajo pasa por todos los flujos que son transversales a las fases, inclusive en varias iteraciones.

RUP se caracteriza por ser **dirigido por casos de uso**, a través de los cuales permite definir lo que el usuario desea a partir de la captura de requisitos. Es **centrado en la arquitectura**, lo que brinda una visión completa del sistema, a través de la descripción de los procesos del negocio que son más importantes, para comprenderlo, desarrollarlo y producirlo de una forma eficaz. **Iterativo e incremental**, donde cada fase se desarrolla en iteraciones, de forma tal que se pueda dividir en pequeños proyectos para mejorar su comprensión y desarrollo.

Un proyecto realizado siguiendo RUP se divide en cuatro fases:

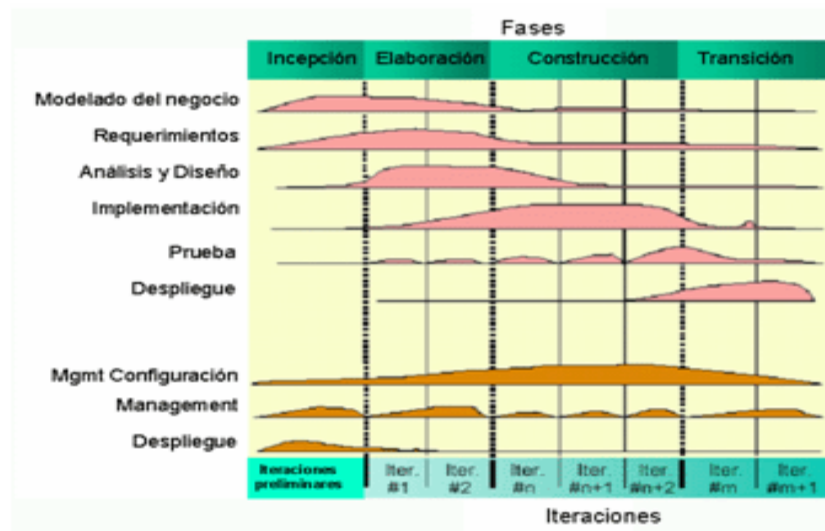


Figura 1. Fases y flujos de RUP

1. Inicio: brinda la posibilidad de definir el alcance y objetivo del proyecto.
2. Elaboración: permite definir la arquitectura del sistema, lo cual provee una base estable para el diseño e implementación de la siguiente fase.
3. Construcción: en la que se obtiene la capacidad operativa del producto, al completar el desarrollo del sistema basado en la línea base de la arquitectura.



4. Transición: se realiza una liberación del producto, garantizando que esté listo para ser instalado y entregado a los usuarios

RUP es más adecuado para proyectos grandes, dado que requiere un equipo de trabajo capaz de administrar un proceso complejo en varias etapas; en proyectos pequeños es posible que no se pueda cubrir los costos de dedicación del equipo de profesionales necesarios. Los requerimientos de los diversos inversores pueden ser diferentes o contradictorios, o disputarse recursos limitados, por lo que debe encontrarse un balance que satisfaga los deseos de todos y el control de calidad no debe realizarse al final de cada iteración, sino en todos los aspectos de la producción (1).

Por tal motivo, esta metodología garantiza obtener los artefactos necesarios durante el desarrollo de la presente investigación, que derivan en un producto con calidad, bien documentado y ejecutado en tiempo.

### **Programación Extrema**

eXtreme Programming (XP por sus siglas en inglés) es una metodología ágil, centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software que promueve el trabajo en equipo; se preocupa por el aprendizaje de los desarrolladores y propicia un buen clima de trabajo. Se basa en la retroalimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios.

XP se define como una metodología especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, donde existe un alto riesgo técnico (2).

Una característica distintiva de XP es la programación en parejas, con el objetivo de que el código sea revisado y validado antes de ser escrito; la refactorización de código está presente durante todo el desarrollo, y no es más que escribir el mismo código fuente nuevamente, buscando claridad, pero sin cambiar la funcionalidad resultante. Las parejas no serán siempre las mismas, sino que se pretende que cada desarrollador haya formado dupla, al menos una vez, con todos los demás, de donde se desprende que el código es de propiedad colectiva y cada uno es responsable por todo el proyecto.



## **SCRUM**

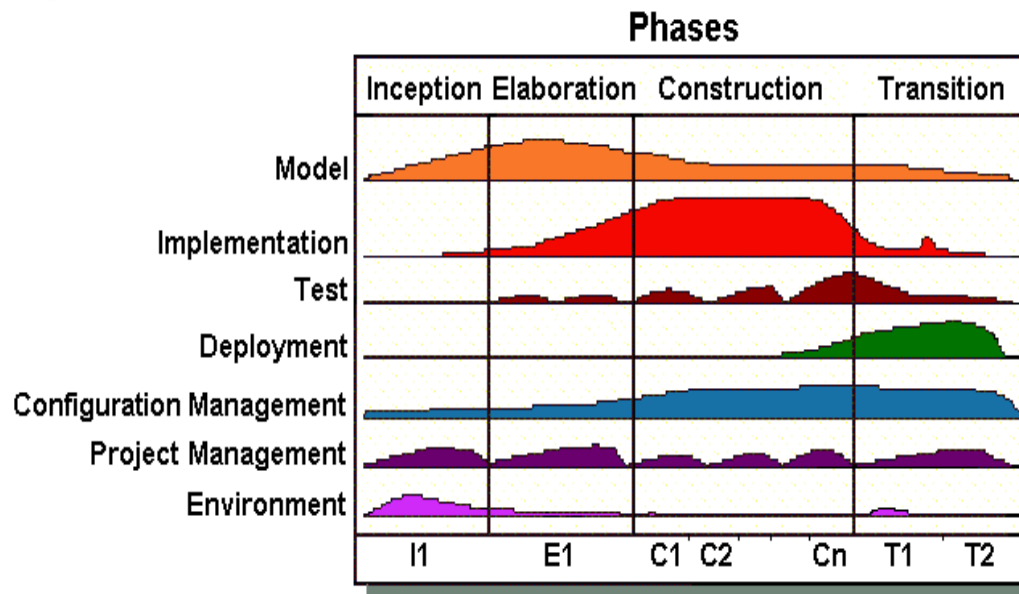
SCRUM es una metodología ágil de desarrollo de software que plantea que los ciclos de desarrollo de los proyectos sean lo más rápidos posible. Esta metodología se ajusta a los cambios que puedan presentarse y mantiene una estrecha relación con el cliente. Establece un desarrollo incremental del producto con entregas frecuentes al cliente, garantizando así que este se encuentre al tanto del progreso del sistema.

SCRUM es muy fácil de aprender, ya que requiere de muy poco esfuerzo para utilizarlo, pero no se tiene una perspectiva global del proyecto más allá de una lista de tareas. Además, no genera toda la documentación que se obtiene con otras metodologías, por lo que no es muy conveniente utilizarlo solo.

### **Agile Unified Procces**

Agile Unified Process, en español Proceso Unificado Ágil, de Scott Ambler, es una versión simplificada del RUP, la cual describe en una forma simple y fácil de entender, y brinda un enfoque de desarrollo de software utilizando técnicas ágiles y conceptos del RUP. En comparación con las disciplinas del RUP, que son 9, el AUP tiene solamente 7, algunas de las cuales son combinaciones de dos disciplinas del RUP.

El ciclo de vida del AUP es el siguiente:



**Figura 2. Fases y flujos de AUP**

Es notable es que las disciplinas han cambiado. En primer lugar, la disciplina Modelo abarca el Modelado de Negocios, Requisitos y Análisis y Diseño del RUP. El Modelo es una parte importante del AUP y, como se puede ver, no domina el proceso. En segundo lugar, las disciplinas Configuración y Gestión del Cambio es ahora la Gestión de Configuración, en su desarrollo ágil de Gestión del Cambio son parte de los esfuerzos de gestión de requerimientos, que forma parte de la disciplina Modelo.

Se pueden mencionar algunos principios, como:

- ✓ El personal necesita saber lo que está haciendo. Las personas no leen la documentación de los procesos en detalle, sino que, en ocasiones, quieren una orientación de alto nivel y/o formación. El producto AUP proporciona enlaces a muchos de los detalles si uno está interesado, pero no obliga a seguirlos.
- ✓ Simplicidad. Todo se describe concisamente usando unas páginas, no miles de páginas.
- ✓ Agilidad. El AUP se ajusta a los valores y principios de la Alianza Ágil.
- ✓ El centrarse en las actividades importantes. La atención se centra en las actividades que realmente cuentan.



Independencia de las herramientas. Se puede utilizar cualquier herramienta con la AUP. Es recomendable usar las herramientas que mejor se adapten al trabajo, que a menudo son simples, o incluso las de código abierto.

## **Selección de la metodología para guiar el desarrollo del software**

RUP es una metodología que genera la documentación necesaria para la transferencia tecnológica, pero la influencia de las metodologías ágiles en el desarrollo de software, actualmente, hace que muchas de sus características puedan combinarse con esta, para lograr así el uso de prácticas y principios que doten de mayor riqueza el desarrollo de un software. La combinación de elementos ágiles, como el desarrollo guiado por pruebas, la gestión rápida de cambios, o la refactorización, entre otros, con el orden propuesto por RUP, se considera como una herramienta poderosa que se ajusta a la visión del proceso de desarrollo de software. Por lo que se decide utilizar Agile Unified Process como metodología para guiar el desarrollo del proyecto.

### **1.5 Lenguaje de modelado**

Entre los lenguajes de modelado analizados, se encuentran:

- ✓ Lenguaje para la Modelación de Aplicaciones Multimedia Educativas (**ApEM-L**, por sus siglas en inglés).
- ✓ Lenguaje para la Modelación Orientada a Objetos de Aplicaciones Multimedia (**OMMMA-L**, por sus siglas en inglés).
- ✓ Lenguaje Unificado de Modelado (**UML**, por sus siglas en inglés).

Este último es el seleccionado para modelar el sistema a desarrollar, debido a que tanto ApEM-L como OMMMA-L se usan netamente para modelar aplicaciones con tecnologías multimedia, razón por la cual no se especifican las características de ambos en la presente investigación.

UML constituye el estándar más aceptado internacionalmente. Es un lenguaje de modelado visual que se usa para visualizar, especificar, construir y documentar artefactos de un sistema de software.



## Principales características de UML

- ✓ Permite modelar sistemas utilizando técnicas orientadas a objetos (OO).
- ✓ Mediante UML se pueden especificar todas las decisiones de análisis, diseño e implementación, construyéndose así modelos precisos, no ambiguos y completos (3).
- ✓ Permite documentar todos los artefactos de un proceso de desarrollo (requisitos, arquitectura, pruebas, versiones, etc.)
- ✓ Cubre las cuestiones relacionadas con el tamaño propio de los sistemas complejos y críticos.
- ✓ Es un lenguaje muy expresivo que cubre todas las vistas necesarias para desarrollar y luego desplegar los sistemas.

El Lenguaje Unificado de Modelado proporciona a los desarrolladores un vocabulario que incluye tres categorías: elementos, relaciones y diagramas (3).

RUP propone usar UML para llevar la documentación del sistema, facilitar la etapa del diseño y posterior construcción o desarrollo, transmitir ideas y ayudar al equipo a comunicarlás, lo cual constituyó un elemento de peso a la hora de su selección.

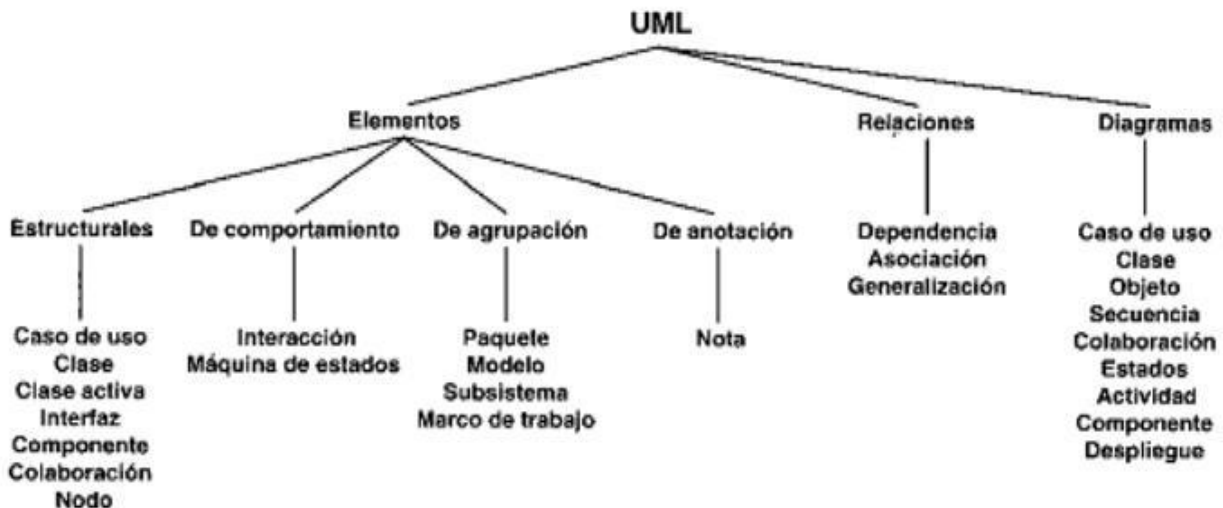


Figura 3. Estructura de UML

## 1.6 Lenguajes y tecnologías para el desarrollo



En la actualidad, el uso de Ajax (Asynchronous JavaScript and XML) para el desarrollo de aplicaciones web se torna prácticamente una necesidad, debido a las ventajas que reporta, entre las cuales se encuentran: mayor dinamismo al software, mejoras en tiempo de respuesta, mejor interacción entre el sistema y el usuario, etc. Por ello se tiene en cuenta para el desarrollo de la aplicación web un conjunto de tecnologías que, al unirse, forman a Ajax.

**Ajax:** En realidad el término es un acrónimo de Asynchronous JavaScript and XML, que se puede traducir como “JavaScript asíncrono y XML”. En un artículo publicado en 2005, Jesse James Garrett lo define como: “Ajax no es una tecnología en sí mismo. En realidad, se trata de varias tecnologías independientes que se unen de formas nuevas y sorprendentes”. Las tecnologías que forman Ajax son XHTML y CSS para crear una presentación basada en estándares, DOM para la interacción y manipulación dinámica de la presentación. XML, XSLT y JSON para el intercambio y la manipulación de información, XMLHttpRequest para el intercambio asíncrono de la información y JavaScript para unir todas las demás tecnologías. (4)

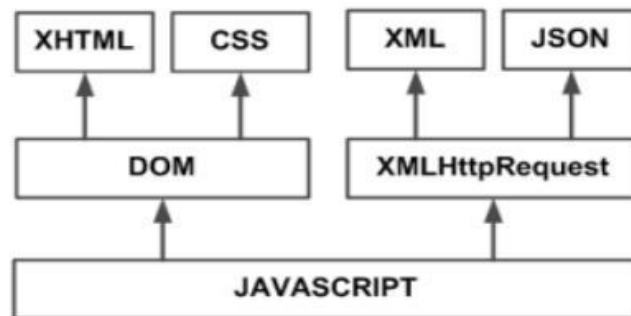


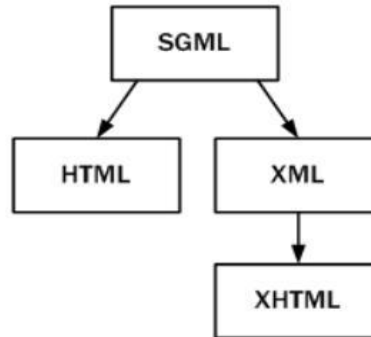
Figura 4. Componentes de AJAX

### Tecnologías del lado del cliente

**XHTML** (Lenguaje de Marcado de Hipertexto Extensible) **Versión 1.1:** El lenguaje XHTML es muy similar al lenguaje HTML (Lenguaje de Marcado de Hipertexto). De hecho, XHTML no es más que una adaptación de HTML al lenguaje XML, sólo añade pequeñas mejoras y modificaciones menores. Técnicamente, HTML es descendiente directo del lenguaje SGML (Lenguaje de Marcado Generalizado), mientras que XHTML lo es de XML (que a su vez también es descendiente de SGML). (5). Este lenguaje proporciona una mejor velocidad de navegación ya que las páginas creadas y maquetadas correctamente se arman más rápido en los exploradores que si no lo están, ya sea porque hay errores de sintaxis o porque se usan elementos con un fin para el cual no fueron



creados. Además, proporciona un mejor entendimiento del código principalmente para los diseñadores; pudiéndose actualizar fácilmente la web.



**Figura 5. Esquema de evolución HTML y XHTML**

**XML** (Lenguaje de Etiquetado Extensible) **Versión 1.1**: Es un lenguaje muy simple pero estricto, representa un papel fundamental en el intercambio de una gran variedad de datos. Es un lenguaje muy similar a HTML pero su función principal es describir datos y no mostrarlos como es el caso de HTML. XML permite la lectura de datos a través de diferentes aplicaciones que usen este (6).

**CSS** (Hojas de Estilo en Cascada) **Versión 2**: Es un lenguaje de hojas de estilo creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos y su presentación, y es imprescindible para crear páginas web complejas. Separar la definición de los contenidos y la de su aspecto presenta numerosas ventajas, ya que obliga a crear documentos HTML/XHTML bien definidos y con significado completo (también llamados “documentos semánticos”). Además, mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferentes (7).

**JavaScript**: Es un lenguaje de programación del lado del cliente que se utiliza principalmente para aportar dinamismo a la interfaz. Una interfaz dinámica es aquella que incorpora efectos, como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario, además se usa para validar la entrada de datos del lado del cliente. Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios. JavaScript es una marca registrada de la empresa Sun Microsystems (8).





## Lenguaje del lado del servidor

Luego de investigar acerca de los lenguajes del lado del servidor, específicamente de PHP versión 5.2.5 y ASP.NET, se han encontrado semejanzas entre estos, tales como: son de uso gratis, se utilizan para construir páginas web dinámicas, presentan soporte para una gran cantidad de bases de datos (MySQL, Oracle, PostgreSQL, entre otras) y permiten la integración con varias bibliotecas externas, lo que posibilita agregar funcionalidades nuevas. Como producto de código abierto, PHP goza de la ayuda de una gran comunidad de programadores que permite que los fallos de funcionamiento se encuentren y reparen rápidamente, a diferencia de ASP.NET que solo puede ser asistido por Microsoft. El código PHP se pone al día continuamente, con mejoras y extensiones de lenguaje para ampliar sus capacidades (9).

Otra característica que se tiene en cuenta para la selección es el servidor que debe dar soporte al lenguaje: PHP es soportado por varios servidores.

### 1.7 Frameworks

El concepto framework o marco de trabajo se emplea en muchos ámbitos del desarrollo de sistemas de software, pero en general se refiere a una estructura de software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. En otras palabras, se puede considerar como una aplicación genérica incompleta y configurable, a la que se le puede añadir las últimas piezas para construir una aplicación concreta. Los objetivos principales que persigue un framework son: acelerar el proceso de desarrollo, reutilizar el código ya existente y promover buenas prácticas de desarrollo con el uso de patrones (10).

#### Capa de presentación

Dentro de los frameworks de la capa de presentación, relacionados con Javascript, se estudiaron EXTJS 2.2 y JQuery 1.4.3. Ambos tienen características similares como son: simplifican la manera de interactuar con los documentos HTML, manipulan el árbol de objetos dinámicos (DOM), manejan eventos, desarrollan animaciones y agregan interacción con la tecnología AJAX a páginas web, son libres, gratis y de código abierto, son compatibles con los navegadores Internet Explorer 7, Firefox 3.6, Opera 10.00, Safari 4.0 y Chrome 7.0. (11)

Se decidió utilizar JQuery debido a que es fácil de usar y aprender, el equipo de trabajo tiene experiencia en el uso de este framework, se integra con el lenguaje PHP y con el framework



Symfony, presenta varios plugins que extienden sus funcionalidades, además de la amplia documentación y comunidad de usuarios, aspectos muy importantes en la retroalimentación y rectificación de errores con el uso de esta herramienta.

### Capa de lógica de negocio

Es necesario definir el concepto de Modelo-Vista-Controlador (MVC): este es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario y la lógica de control en tres componentes distintos. El patrón MVC se ve frecuentemente en aplicaciones web.

(12)

**Modelo:** Es la representación específica de la información con la cual el sistema opera. La lógica de datos asegura la integridad de estos y permite derivar nuevos datos; por ejemplo, no permite comprar un número de unidades negativo, calcula si hoy es el cumpleaños del usuario o los totales, impuestos o importes en un carrito de compra.

**Vista:** Presenta el modelo en un formato adecuado para interactuar con este, usualmente la interfaz de usuario.

**Controlador:** Este responde a eventos, usualmente acciones del usuario, e invoca cambios en el modelo y, probablemente, en la vista.

Luego de un análisis de los frameworks utilizados para optimizar el desarrollo de aplicaciones web, desarrollados en el lenguaje del lado del servidor PHP, y que separan la lógica de negocio, la lógica de servidor y la presentación de la aplicación web, como son Zend Framework, CakePHP y Symfony, se encontraron características similares, como son: implementan el patrón MVC; proporcionan una estructura al código fuente, forzando al desarrollador a crear un código más legible para un futuro mantenimiento; encapsulan operaciones complejas en instrucciones sencillas, reutilizando código; se pueden ejecutar tanto en plataformas \*nix (Unix, Linux, etc.) como en plataformas Windows; presentan CRUD (Crear, Obtener, Actualizar y Borrar), integrado para la interacción con la base de datos; usan programación orientada a objetos; son independientes del sistema de base de datos, soportando varios gestores como MySQL, Oracle y PostgreSQL, lo cual es importante para si en un futuro se cambia el gestor; también soportan integración con el mapeador de objetos relacional (ORM por sus siglas en inglés) y con el framework de la capa de presentación jQuery.(13)

Se decide utilizar el framework Symfony 1.4.11 debido a que la comunidad de usuarios y la documentación son muy amplias, por lo que es importante para la retroalimentación, superación y



reparación de errores. La comunidad provee diversos plugins que pueden ser reutilizables. El equipo de trabajo cuenta con vasta experiencia en el uso de este framework, el cual es actualizado constantemente con mejoras y soluciones a los errores encontrados.

### **Capa de acceso a datos**

Se analizaron los dos ORM más conocidos y utilizados, Doctrine y Propel. Ambos presentan características similares ya que soportan cualquier operación usual en un CRUD; pueden generar las clases PHP del modelo, Propel basado en XML y Doctrine basado en YAML (Ain't Markup Language), este último es más legible para las personas. Ambos soportan la generación de sus respectivos markups desde una base de datos existente, son para PHP; brindan la posibilidad de escribir (opcionalmente) consultas a la base de datos en un lenguaje orientado a objetos, lo cual posibilita a los desarrolladores una alternativa al SQL, así como mantener una máxima flexibilidad sin necesidad de duplicar código. Se sitúan en la parte superior de la capa de abstracción de la base de datos, lo que permite abstracción al gestor utilizado, como MySQL, Oracle, PostgreSQL, etc. Soportan validación de datos en los modelos y relaciones entre modelos. (14)

Se decide usar Doctrine debido a que es muy rápido y eficaz, tanto es así que en la versión de Propel 2.0 se usarán muchas características de Doctrine 2.0. Doctrine posee una amplia documentación, presenta soporte para migraciones. Posee una característica que facilita aún más las consultas a la base de datos, que es la presencia de métodos mágicos, los cuales permiten obtener información por cualquier campo de la tabla en cuestión. Además, el equipo de trabajo tiene experiencia en el uso de este ORM.

### **1.8 Sistema gestor de base de datos**

Analizando algunos de los gestores de bases de datos más utilizados a nivel mundial –MySQL y PostgreSQL– se encontraron características similares, como son: funcionan en los principales sistemas operativos, en los que se incluyen Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64) y Windows; son totalmente compatibles con ACID (Atomicidad, Consistencia, Aislamiento y Durabilidad); tienen soporte completo para claves foráneas, uniones, vistas, disparadores y procedimientos almacenados. PostgreSQL posee una gran escalabilidad y es capaz de ajustarse al número de CPUs y a la cantidad de memoria que posee el sistema de forma óptima, por lo que es capaz de soportar una mayor cantidad de peticiones simultáneas de manera correcta. No obstante, la concurrencia de acceso a la base de datos esperada en el sistema que se propone



es de 20 usuarios. Por otra parte, los mayores inconvenientes que se pueden encontrar a este gestor son:

- Consume gran cantidad de recursos.
- Tiene un límite de espacio por fila, y su aumento con lleva a una disminución considerable del rendimiento.

Por otro lado, MySQL tiene una gran velocidad a la hora de realizar las operaciones, lo que le hace uno de los gestores que ofrecen mayor rendimiento. Debido a su bajo consumo es apto para ser ejecutado en una máquina con escasos recursos, como lo es el servidor destinado a la aplicación. Tiene una probabilidad muy reducida de corromper los datos, incluso en los casos en los que los errores no se produzcan en el propio gestor, sino en el sistema en el que está. (15)

Se decidió utilizar MySQL, debido a su gran rendimiento, además de ser un requisito no funcional solicitado por los clientes, ya que es el gestor utilizado en la institución.

### **1.9 Servidor Web**

Se estudiaron los servidores más usados, según las estadísticas, desde agosto de 1995 hasta febrero de 2011: Internet Information Services 6.1 (IIS) y Apache 2.2.6. (16)

Se decide utilizar este último, ya que presenta licencia de software libre y de código abierto; mientras que IIS tiene licencia de software propietario. Apache es multiplataforma, a diferencia de IIS que funciona solo sobre la plataforma Windows, lo que es una limitación. Además, Apache es el servidor por excelencia para PHP. (17)

Apache tiene una alta comunidad de usuarios, lo que posibilita que se le dé solución tanto a errores del propio servidor como a errores de configuración; es seguro ya que permite protección de ficheros. (18)

### **1.10 Entorno de desarrollo**

Al estudiarse dos de los entornos de desarrollo integrado (IDE por sus siglas en inglés) más utilizados para la elaboración de aplicaciones web, como son Eclipse y NetBeans, se verificó que ambos IDEs son similares en cuanto a las funcionalidades y ventajas que ofrecen, como son: el soporte para PHP, son libres, gratuitos, sin restricciones de uso y de código abierto. Se decidió escoger NetBeans en su versión 6.9 como IDE, debido a las ventajas con respecto a Eclipse tales como: rapidez, lo que implica facilidad en cuanto la completamiento de código en los lenguajes de



programación utilizados, además del completamiento de código para los métodos mágicos que usa el ORM Doctrine. (19)

### **1.11 Herramientas CASE para el modelado UML**

La utilización de una herramienta CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Ordenador) mejora la comunicación entre usuarios y especialistas, permite la facilidad de revisión de aplicaciones instaladas, es capaz de generar automáticamente las instrucciones del programa fuente y da la posibilidad de la generación de documentación técnica.

Entre las herramientas CASE orientadas a UML se encuentran ArgoUML y Visual Paradigm, que constituyen dos de las herramientas de desarrollo de ingeniería de software para Linux más usadas en la actualidad.

#### **ArgoUML**

ArgoUML es una herramienta desarrollada en Java que permite crear modelos UML compatibles con los estándares de la versión 1.4 de este lenguaje. Incluye una interfaz muy intuitiva, estable y de sencillo manejo. Los tipos de diagramas que se pueden crear son: diagramas de clases, de estados, de actividad, de casos de uso, de colaboración, de despliegue y de secuencia. (20)

Otras características presentes en ArgoUML son:

- ✓ Compatibilidad con el estándar XMI (XML Metadata Interchange, XML de Intercambio de Metadatos): es posible utilizar los modelos creados con UML en otros programas.
- ✓ Generación de código para los lenguajes Java, C++, C, PHP4 y PHP5.
- ✓ Framework modular para ingeniería inversa.
- ✓ Críticas de diseño o sugerencias realizadas por ArgoUML para mejorarlo.

#### **Visual Paradigm**

Es una herramienta diseñada para desarrollar software con soporte a la programación orientada a objetos. Busca reducir la duración del ciclo de desarrollo, brindando ayuda a arquitectos, analistas, diseñadores y desarrolladores; permite el uso de un lenguaje estándar común a todo el equipo de desarrollo, que facilita la comunicación y tiene capacidades de ingeniería directa e inversa. Usa UML como lenguaje de modelado. Presenta una interfaz de uso intuitiva y con muchas facilidades a la hora de modelar los diagramas que soportan la Ingeniería de Requerimientos. (21)



Una de las características más importantes de su uso es que brinda la posibilidad de sincronización del modelo de diseño y el código en todo el ciclo de desarrollo, una vez que se integra con el NetBeans IDE; ello permite la facilidad de programar directamente sobre el código fuente generado y, a su vez, actualizar el diseño con cambios que se realicen en la programación. Tiene una alta disponibilidad de múltiples versiones para cada necesidad, al igual que en las plataformas *Linux*, *MacOS* y *Windows*.

Presenta un innovador analizador textual donde se introduce texto extraído de conversaciones con el cliente, se definen actores, entidades, casos de uso disponibles para la generación de artefactos posteriores. También posee una herramienta de generación de reportes en formato PDF o HTML, configurable y selectiva, e importa o exporta formatos estándares de otras herramientas CASE como el Rational Rose.

### **Selección de la herramienta CASE**

Por todas las potencialidades anteriormente citadas, tales como: el innovador analizador textual, la constante sincronización del modelo de diseño y el código fuente durante todo el ciclo de desarrollo del software, además de su excelente generación de código y soporte para UML 2.1, se ha escogido Visual Paradigm en su versión 6.4 como herramienta CASE.

### **1.12 Conclusiones**

Teniendo en cuenta el estudio realizado previamente sobre las principales tendencias en el campo de la informática, se toma como decisión desarrollar una aplicación web utilizando el lenguaje PHP, con la integración de los frameworks jQuery 1.4.3 y Symfony 1.4.9, buscando ganar en velocidad de elaboración e incorporar buenas prácticas de desarrollo con el uso de patrones. Para la construcción del sistema se determina hacer uso de las herramientas NetBeans y Visual Paradigm, debido a las facilidades que estas brindan. Para ganar en rendimiento se utilizará el sistema gestor de bases de datos MySQL. Todo este proceso será controlado y orientado por la metodología AUP, la cual constituye una guía de cómo se debe desarrollar un software.



## Capítulo 2 . Propuesta de solución

El modelado del sistema, utilizando la Agile Unified Procces, tiene una mayor presencia en las fases de inicio y elaboración para la presente investigación. El objetivo de esta disciplina es entender el negocio de la organización, el dominio que se aborda en el proyecto, y determinar una solución viable para contribuir a la mejora de los procesos que se realizan en la Oficina de Reserva, Coordinación, Factura y Gestión de Cobro del Acuario Nacional de Cuba.

En este capítulo se describe el flujo de acciones realizadas para viabilizar la implementación del sistema; se muestran los aspectos principales, escogidos de la documentación generada y además se complementa con otras explicaciones y comentarios que no perdurarán como parte de la documentación del proyecto.

### 2.1 Funcionamiento del negocio y elementos de automatización

Después del contacto inicial en el que se concibió el proyecto, se definieron las metas que se desea lograr por parte del colectivo, formado por los especialistas del sector comercial del Acuario Nacional y la autora. El resultado de estos encuentros fue una mejor comprensión de los deseos del cliente, así como un mejor entendimiento del proyecto por parte de la autora y una visión común de los objetivos.

El negocio a automatizar gira en torno a tres procesos elementales del negocio, a partir de los cuales se concibió adicionar una serie de posibilidades y cambios implícitos en la nueva solución, para facilitar los procesos y elevar la calidad del resultado final. Las partes distintivas de lo que ocurre habitualmente en el negocio a automatizar son: la realización de una reservación, con variantes; la confirmación de esta y, finalmente, la facturación.

A continuación se realiza una descripción informal de los procesos elementales del negocio a partir del desglose de sus actividades, que permiten mostrar cómo se realiza el trabajo de las personas implicadas mediante la identificación las mejoras potenciales.

#### **Realizar una reservación**

El cliente es atendido por el técnico en promoción y comunicación integral, que es el encargado de mostrarle todas las ofertas disponibles en ese momento y aclarar cualquier duda, para que finalmente se llegue a un acuerdo que incluye la fecha, la cantidad de personas para las que se reserva, los servicios de que van a disfrutar, etc. La oficina conserva los datos del cliente para contactarlo en caso de cambios y viceversa. Cabe aclarar que el cliente puede ser tanto un



representante de una institución o empresa, como alguien independiente que simplemente quiere disfrutar de los servicios brindados.

### **Confirmar una reservación**

Una semana antes de la fecha pactada durante la reservación, el cliente debe llamar por teléfono o presentarse personalmente para confirmar su presencia, y redefinir los términos del contrato en caso de ser necesario o desearlo. Opcionalmente se realiza el pago de los servicios, descrito a continuación. Una vez realizado este paso, no se puede alterar la reservación.

### **Facturar una reservación**

La factura se puede hacer desde el mismo día que se confirma, pero a veces se formaliza después de hacerse efectiva la visita. Consiste en desglosar el consumo del cliente y dar una serie de subtotales y total a pagar. De la factura se hacen dos copias, una para el cliente y otra se archiva en el Acuario Nacional.

### **Mejoras esperadas a partir de la puesta en marcha del sistema**

Todo esto puede realizarse más eficientemente auxiliándose de la infraestructura tecnológica con que cuenta la institución y utilizando el Sistema de Reserva y Coordinación propuesto.

El accionar de la oficina comercial se beneficiará, de forma general, con la seguridad de la información, lo cual se garantiza con los niveles de acceso brindados por el sistema. La redundancia de la información guardada va a ser mínima, y esta será accesible a través del sistema informático. Las consultas de información serán más rápidas, esto se garantiza con el uso de filtros de búsqueda. Si ya el cliente ha realizado una reservación con anterioridad, los datos de él estarían almacenados y disponibles para agilizar la negociación. La verificación de solapamiento de fechas y otras reglas que puedan existir en el negocio serían realizadas instantáneamente, al seleccionar los servicios deseados.

Otra ventaja, durante la confirmación, sería la verificación del rango de fecha en el que se puede confirmar una reservación; y durante la facturación, la verificación del estado actual de la reservación. El anexo de órdenes a la factura, o sea, del consumo adicional de productos y tables gastronómicos, también agiliza potencialmente el funcionamiento de la oficina y el servicio en general.

Un beneficio extra puede resultar del registro automático de cambios sobre las entidades más trascendentales: reservas y facturas. Un pequeño sistema de mensajería interna entre los usuarios puede mejorar la coordinación de eventos. Los directivos podrán beneficiarse de variados tipos de





reportes que ayuden a la toma de decisiones. La planificación de las hojas de operaciones, que actualmente se hacen utilizando una pizarra, podrá ser automatizada a partir de la información guardada anteriormente.

## 2.2 Modelo de Dominio

Aunque existen en la organización tres procesos elementales del negocio, la definición de roles y responsabilidades no está completamente definida en la organización, así como la asignación de todas las responsabilidades de las personas implicadas en el negocio, es por esta razón que se hace necesario la realización de un modelado del dominio, con el objetivo de identificar todas las relaciones entre los principales conceptos de la organización.

La figura muestra como la reservación es el elemento fundamental en el negocio, que es solicitada por los tipos de cliente que existen, y que es realizada por el trabajador de la oficina comercial. Con dicha reservación se relacionan los diferentes servicios de los que pretenda disfrutar el cliente y en algún momento se factura.

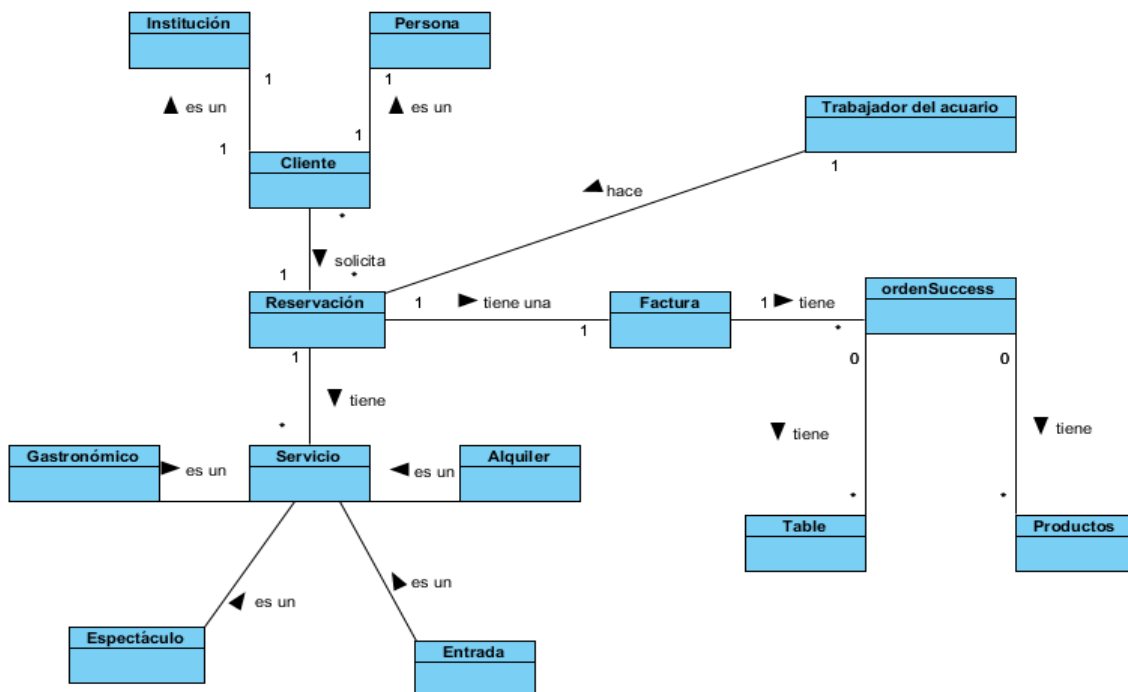


Figura 6. Modelo de dominio para Sistema de Reserva y Coordinación

## 2.3 Lista de conceptos modelados en el Dominio y definiciones correspondientes

Reservación: Asignación que se hace de los locales (alquiler), la gastronomía y los espectáculos, con la intención de que pueda ser utilizada por el cliente en el plazo convenido.



Producto: Cualquier elemento que se ofrece para su consumo, con fines tanto nutricionales como psicológicos.

Table: Conjunto de productos que se pagan en una misma moneda y que pueden formar parte de las órdenes consumidas por los clientes.

Orden: Conjunto de productos gastronómicos y tables consumidos por el cliente que se anexan a la factura.

Servicios: Son las ofertas que brinda el Acuario, que pueden ser reservadas.

Espectáculo: Tipo de servicio que brinda el Acuario. Ej.: Delfines, Ballet Acuático, etc.

Gastronómico: Tipo de servicio que brinda el Acuario. Ej.: Almuerzo, Comida, Buffet.

Alquiler: Tipo de servicio que brinda el Acuario, aplicado solo a los locales.

Factura: Documento que refleja toda la información de una reservación, fundamentalmente la entrega de productos o la provisión de servicios, junto a la fecha de devengo, además de indicar la cantidad a pagar.

Cliente: Persona o institución que solicita la reservación.

Institución: Tipo de cliente referente a empresa o institución que debe estar representada por una persona.

Persona: Tipo de cliente.

Trabajador del acuario: Persona encargada de realizar las reservaciones y facturarlas.

## **2.4 Lista de requisitos**

El sitio oficial de AUP recomienda el tratamiento de requisitos como una lista priorizada que evolucione en el tiempo (esto apoya la gestión de cambios, no se opone a estos). Las reglas del negocio y los requerimientos técnicos se deben tener en cuenta para su generación y cambio. Basado en el modelado del dominio y teniendo en cuenta los elementos anteriores se obtiene el siguiente listado de requisitos, los que fueron validados y aceptados en su totalidad por el cliente:

### **Requisitos funcionales:**

**RF1** Autenticar Usuario.

**RF2** Gestionar local.

**RF2.1** Adicionar local.

**RF2.1.1** Listar las monedas en las que se va a alquilar el local.





- RF8** Gestionar usuario.
  - RF8.1** Adicionar usuario.
    - RF8.1.1** Listar grupos disponibles.
    - RF8.1.2** Listar premisos disponibles.
  - RF8.2** Modificar usuario.
    - RF8.2.1** Listar grupos disponibles.
    - RF8.2.2** Listar premisos disponibles.
  - RF8.3** Eliminar usuario.
- RF9** Consultar usuarios.
  - RF9.1** Listar usuarios.
    - RF9.1.1** Paginar listado de usuarios.
- RF10** Gestionar grupos de usuarios.
  - RF10.1** Adicionar grupo de usuarios.
    - RF10.1.1** Listar usuarios disponibles.
    - RF10.1.2** Listar premisos disponibles.
  - RF10.2** Modificar grupo de usuarios.
    - RF10.2.1** Listar usuarios disponibles.
    - RF10.2.2** Listar premisos disponibles.
  - RF10.3** Eliminar grupos de usuarios.
- RF11** Consultar grupos de usuarios.
  - RF11.1** Listar grupos de usuarios.
    - RF11.1.1** Paginar listado de los grupos de usuarios.
- RF12** Gestionar permisos de usuarios.
  - RF12.1** Adicionar permisos de usuarios.
    - RF12.1.1** Listar usuarios disponibles.
    - RF12.1.2** Listar grupos disponibles.
  - RF12.2** Modificar permisos de usuarios.
    - RF12.2.1** Listar usuarios disponibles.
    - RF12.2.2** Listar grupos disponibles.
  - RF12.3** Eliminar permisos de usuarios.
- RF13** Consultar permisos de usuarios.
  - RF13.1** Listar permisos de usuarios.
    - RF13.1.1** Paginar listado de los permisos de usuarios.



- RF14** Gestionar productos gastronómicos.
  - RF14.1** Adicionar productos gastronómicos.
  - RF14.2** Modificar productos gastronómicos.
  - RF14.3** Eliminar productos gastronómicos.
- RF15** Consultar productos gastronómicos.
  - RF15.1** Listar productos gastronómicos.
    - RF15.1.1** Pagar listado de productos gastronómicos.
  - RF15.2** Filtrar productos gastronómicos.
- RF16** Gestionar tables gastronómicos.
  - RF16.1** Adicionar tables gastronómicos.
    - RF16.1.1** Listar productos de la moneda en la moneda seleccionada.
    - RF16.1.2** Calcular el precio total si se paga en Moneda Nacional.
  - RF16.2** Modificar tables gastronómicos.
    - RF16.2.1** Listar productos de la moneda en la moneda seleccionada.
    - RF16.2.2** Calcular el precio total si se paga en Moneda Nacional.
  - RF16.3** Mostrar tables gastronómicos.
  - RF16.4** Eliminar tables gastronómicos.
- RF17** Consultar tables gastronómicos.
  - RF17.1** Listar tables gastronómicos.
    - RF17.1.1** Pagar listado de tables gastronómicos.
  - RF17.2** Filtrar tables gastronómicos.
- RF18** Gestionar reservación.
  - RF18.1** Adicionar reservación.
    - RF18.1.1** Adicionar cliente persona.
    - RF18.1.2** Adicionar cliente institución.
    - RF18.1.3** Buscar cliente persona.
    - RF18.1.4** Buscar cliente institución.
    - RF18.1.5** Asignar espectáculos a la reservación.
    - RF18.1.6** Eliminar espectáculos a la reservación.
    - RF18.1.7** Asignar servicios gastronómicos a la reservación.
    - RF18.1.8** Eliminar servicios gastronómicos a la reservación.
    - RF18.1.9** Asignar alquileres a la reservación.







- RF25.1.1** Filtrar semanalmente.
  - RF25.2** Generar hoja de operaciones por gastronomía en locales.
    - RF25.2.1** Filtrar semanalmente.
  - RF25.3** Generar reporte del dinero recaudado mensualmente por alquiler de locales.
  - RF25.4** Generar reporte del dinero recaudado mensualmente por gastronomía.
  - RF25.5** Generar reporte del dinero recaudado mensualmente por espectáculos.
  - RF25.6** Generar reporte del dinero recaudado mensualmente en cada una de las monedas y por cada uno de los servicios.
  - RF25.7** Generar reporte de la cantidad de espectáculos reservados mensualmente.
  - RF25.8** Generar reporte de la cantidad de alquileres reservados mensualmente.
  - RF25.9** Generar reporte de la cantidad de servicios gastronómicos reservados mensualmente.
  - RF25.10** Generar reporte de la cantidad de servicios reservados mensualmente.
  - RF25.11** Generar reporte de la cantidad de clientes nacionales y extranjeros que han realizado alguna reservación mensualmente.
  - RF25.12** Imprimir reportes.
  - RF25.13** Descargar reportes en formato JPEG, PNG, PDF, SVG.
- RF26** Gestionar mensajes.
- RF26.1** Enviar mensaje a uno o varios usuarios en la aplicación.
  - RF26.2** Eliminar mensaje.
  - RF26.3** Mostrar los últimos 5 mensajes recibidos.

Una vez definida las funcionalidades del sistema, correspondió identificar las cualidades o propiedades que complementan el funcionamiento del mismo; para ello se identificaron los requerimientos no funcionales, que se describen a continuación:

### **Requisitos no funcionales**

#### Requisitos de interfaz externa

- RNF1.** Las etiquetas de los campos, mensajes de error y demás textos deben presentarse de forma similar.
- RNF2.** Las validaciones, en caso de error, se deben señalar mostrando un mensaje de error. Ejemplo: “Requerido” o “Solo dígitos”.
- RNF3.** El menú debe estar disponible en cualquiera de las páginas de la aplicación.
- RNF4.** El nombre del usuario debe estar en la parte superior de la interfaz.





**RNF5.** El diseño y los colores utilizados serán afines al manual de identidad de la institución (blanco, azul y amarillo) y el logo de la misma deberá estar visible en todo momento.

#### Requisitos de usabilidad

**RNF6.** El sistema debe ser fácil de usar para personas con conocimientos de computación básicos, dígame básicos como manejo de los recursos de ofimática, etc.

**RNF7.** Cada página de entrada a algún módulo en el sitio debe tener una explicación breve acerca de lo que el usuario puede realizar en ella. Ejemplo: “Administrar Reservas En este módulo usted podrá gestionar la información correspondiente a las reservas tanto para personas como para instituciones, es decir, podrá crear, modificar y eliminar reservas, así como realizar búsquedas filtradas en el listado inicial. Además podrá observar el historial de cambios o modificaciones de cada una de estas”.

**RNF8.** El sistema debe mostrar el estado estadístico de facturación, reservación y coordinación de la institución.

**RNF9.** La mensajería interna será en forma de post it haciéndola más amigable e intuitiva.

#### Requisitos de rendimiento

**RNF10.** Los tiempos de respuesta deben ser menores de 5 segundos.

#### Requisitos de soporte

**RNF11.** El sistema debe ser escalable en cuanto a los valores y la cantidad de nomencladores. Ejemplo: Si se inserta una nueva moneda en el nomenclador nom\_moneda la facturas automáticamente agregaran otra columna precio – importe, los productos, tables locales y espectáculos deben brindar la opción de pagarse y/o alquilarse en la nueva moneda, etc.

**RNF12.** Debe ser capaz de asimilar un nuevo módulo sin que se afecten los que ya existen.

#### Requisitos de seguridad

**RNF13.** El acceso al sistema estará restringido por usuario y contraseña.

**RNF14.** Las contraseñas de guardan encriptadas en la base de datos, lo que permite mayor seguridad.

**RNF15.** Los usuarios en la aplicación estarán agrupados por roles o niveles de acceso, que a su vez tendrán definidos permisos sobre la aplicación. Ejemplo: el tecnicopci tiene el permiso o



credencial reservas\_all que le permite gestionar y consultar las reservaciones, facturas y clientes.

**RNF16.** El sistema debe mostrar la trazabilidad de las entidades reservación y factura de acuerdo a los historiales de cambio realizados en todo momento.

#### Requisitos de portabilidad

**RNF17.** El sistema debe ser multiplataforma, es decir que puede instalarse lo mismo en Linux que en Windows.

**RNF18.** El usuario debe acceder a la aplicación desde cualquier sistema operativo.

#### Requisitos de software

**RNF19.** Las computadoras cliente deben tener instalado el navegador web Mozilla Firefox 3.6, Internet Explorer 7.0 o Chrome 10 o superior, en cada uno de los navegadores mencionados.

**RNF20.** Las computadoras cliente deben tener instalado Adobe Acrobat Reader.

**RNF21.** El servidor debe tener instalado un servidor MySQL 5.5 y un servidor de aplicaciones Apache 2.2.6 o superior.

#### Requisitos de hardware

**RNF22.** La aplicación debe consumir los mínimos recursos.

**RNF23.** Debe funcionar sobre una infraestructura de redes. Ejemplo: red de clientes ligeros del Acuario Nacional de Cuba.

#### Requisitos de accesibilidad

**RNF24.** La aplicación debe estar accesible desde cualquier lugar de la red.

**RNF25.** La información real y actualizada debe estar disponible para la consulta en todo momento.

## **2.5 Modelo de Casos de Uso del Sistema**

A través del procesamiento de los artefactos generados anteriormente, se realizó el Modelo de Casos de Uso, estos constituyen la guía del proceso de desarrollo y dan una visión del sistema a desarrollar. En la figura se observa la jerarquía de actores que interactúan con el sistema, en la cual el Técnico de Promoción y Comunicación Integral juega el rol principal, el actor Gastronómico ejecuta acciones complementarias, así como el administrador del sistema y los directivos son usuarios consumidores de información exclusivamente. Todos deben estar autenticados y pueden hacer uso de un sencillo sistema de mensajería.

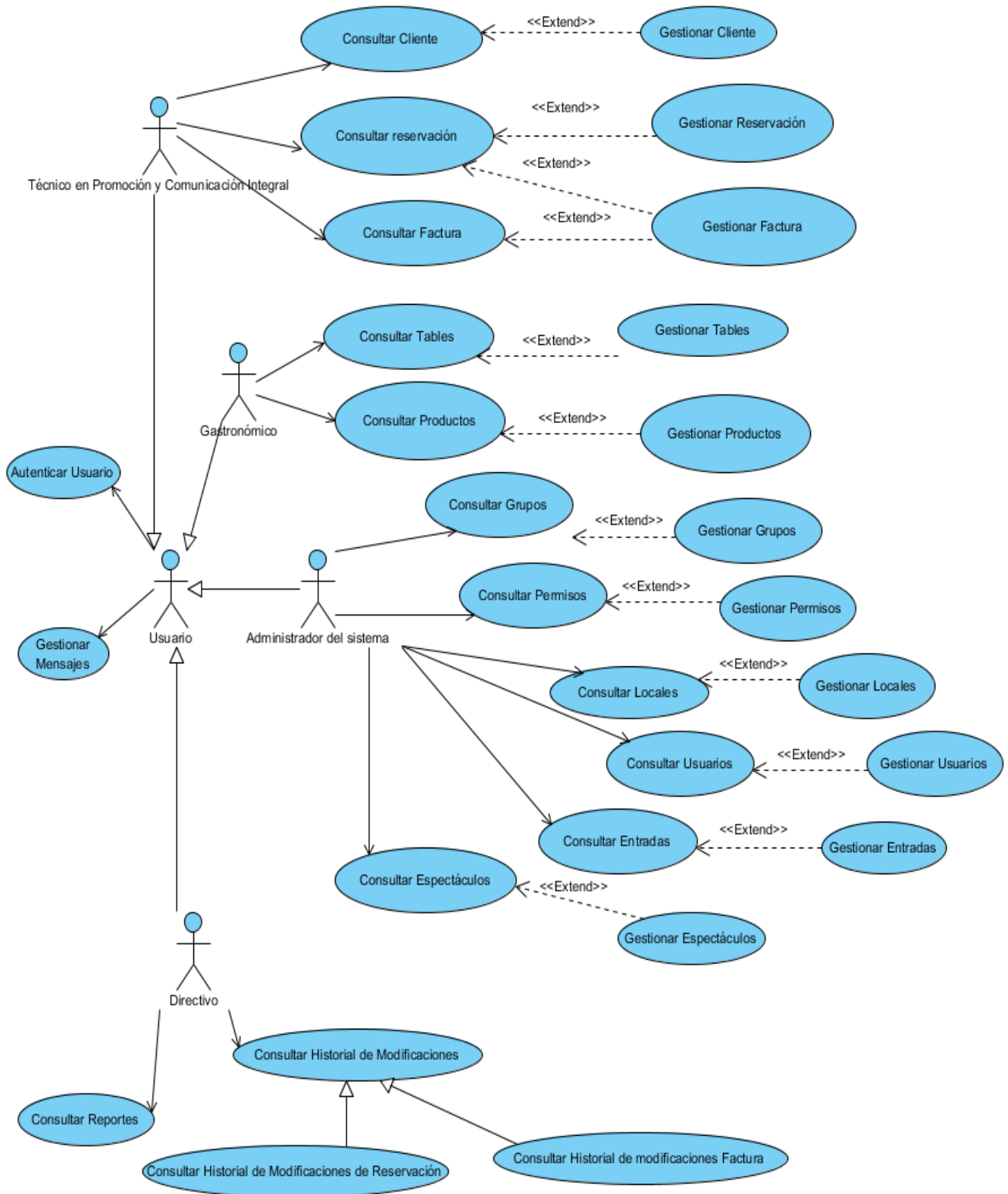


Figura 7. Modelo de Casos de Uso del sistema de Reserva y Coordinación del Acuario Nacional



## 2.6 Descripción de Casos de Uso en formato resumido

A través de un análisis de los requisitos, se estructuraron, se agruparon y se convirtieron en casos de uso, estos casos de uso se describen como parte de la documentación a generar y para comprender el flujo de acciones y respuestas del sistema, el formato de descripción que se utilizó para el sistema en cuestión es la descripción resumida. Se seleccionaron para ilustrar su desarrollo en este documento un subconjunto de los Casos de Uso atendiendo primeramente a la importancia para el cliente, y en segundo lugar, algunos Casos de Uso representativos de un grupo mayor que se puede encontrar en los anexos.

Nombre	Gestionar Mensajes.
Objetivo	Gestionar los mensajes entre los usuarios de la aplicación.
Actor	Usuario.
Complejidad	Baja.
Nivel	Usuario.
Precondiciones	Debe haberse autenticado el usuario en el sistema.
Poscondiciones	Se incluyó o eliminó el mensaje por el actor.
Resumen del caso de uso	
El caso de uso se inicia cuando el actor selecciona la opción que le permite crear un mensaje nuevo, el sistema dará la posibilidad de insertar los datos que se necesitan, el mensaje podrá ser enviado a uno o varios usuarios. Los mensajes se mostrarán ordenados cronológicamente, es decir, los últimos 5 recibidos. Si el actor elige la opción de eliminar un mensaje, el sistema permitirá eliminarlo, terminando así el caso de uso.	
Requerimientos funcionales	
<b>RF26</b>	

*Descripción resumida Gestionar Mensajes*

Nombre	Consultar Reservaciones.
Objetivo	Consultar las reservaciones existentes.



Actor	Técnico en Promoción y Comunicación Integral.
Complejidad	Baja.
Nivel	Usuario.
Precondiciones	Debe haberse autenticado el usuario en el sistema. Debe haberse generado el escritorio de trabajo de la aplicación de Reservas y el módulo Reservaciones.
Poscondiciones	Se consultó la reservación por el actor.
Resumen del caso de uso	
<p>El caso de uso se inicia cuando se genera el escritorio de trabajo del módulo y se muestra el listado de las reservaciones, con los datos: nombre de la entidad, hora de entrada, fecha, servicios reservados y las acciones que se pueden realizar sobre cada una de ellas. El actor podrá, además, realizar búsquedas sobre el listado filtrándola por los campos: nombre, fecha, usuarios y/o confirmada; entonces el sistema mostrará el listado con las reservaciones que coinciden con los criterios especificados. El actor puede consultar, además, las reservaciones que ya han sido facturadas y no aparecen en el listado inicial.</p>	
Requerimientos funcionales	
<b>RF19</b>	

*Descripción resumida Consultar Reservaciones*

Nombre	Gestionar Reservaciones.
Objetivo	Gestionar las reservaciones en la aplicación.
Actor	Técnico en Promoción y Comunicación Integral.
Complejidad	Alta.
Nivel	Usuario.
Precondiciones	Debe haberse autenticado el usuario en el sistema. Deben existir los datos en el nomenclador nom_tipo_cliente. Deben existir los datos en el nomenclador nom_moneda. Deben existir los datos en el nomenclador nom_tipo_gastronomico Debe haberse generado el escritorio de trabajo de la aplicación de Reservas y el módulo Reservaciones.



Poscondiciones	Se incluyó, modificó, vio o confirmó la reservación por el actor.
<b>Resumen del caso de uso</b>	
<p>El caso de uso se inicia cuando el actor selecciona la opción que le permite realizar una acción sobre la reservación. El actor puede incluir, ver, modificar o confirmar una reservación.</p> <p>En el caso de que seleccione la opción de incluir una nueva reservación, ya sea para una persona o para una institución, el sistema dará la posibilidad de insertar los datos que se necesitan para llenar esta plantilla, dividiéndolos en tres grupos fundamentales:</p> <ul style="list-style-type: none"><li>✓ Datos de la entidad o del cliente: en el que opcionalmente el actor podrá verificar que el cliente ya exista en el sistema buscándolo por el carnet de identidad y, de ser así, se completarían los restantes datos para agilizar este trámite.</li><li>✓ Datos de la reservación: es recomendable llenar el campo fecha antes de pasar al último grupo para evitar solapamientos en los alquileres.</li><li>✓ Datos de los servicios: aquí el actor asociará, a la reservación, los servicios que desea el cliente, y dará la opción desde ese momento de especificar en qué moneda quiere pagar cada uno de estos, el horario y la cantidad de personas en el que se consumirá, siempre que esté disponible. El cliente podrá eliminar estos servicios si así lo desea, pero para crear la reservación debe estar asociado, al menos, un servicio.</li></ul> <p>Si el actor elige la opción de modificar una reservación, el sistema mostrará los datos mencionados anteriormente y, una vez realizados los cambios, guardará las modificaciones. Si el actor elige la opción de ver una reservación, el sistema mostrará toda la información detallada de la misma.</p> <p>Una semana antes de efectuarse la reservación se activará la opción confirmar y, en ese rango, el actor podrá confirmarla, para poder facturarla posteriormente.</p>	
<b>Requerimientos funcionales</b>	
<b>RF18</b>	

*Descripción resumida Gestionar Reservaciones*

Nombre	Gestionar Factura.
Objetivo	Gestionar las reservaciones en la aplicación.



Actor	Técnico en Promoción y Comunicación Integral.
Complejidad	Alta.
Nivel	Usuario.
Precondiciones	Debe haberse autenticado el usuario en el sistema. Deben existir los datos en el nomenclador nom_moneda. Debe haberse generado el escritorio de trabajo de la aplicación de Reservas y el módulo Factura.
Poscondiciones	Se incluyó, modificó, vio o eliminó la factura por el actor.
<b>Resumen del caso de uso</b>	
<p>El caso de uso se inicia cuando el actor selecciona la opción que le permite realizar una acción sobre la factura. El actor puede incluir, ver, modificar o eliminar una reservación.</p> <p>En el caso de que desee incluir una nueva factura, deberá consultar el listado de las reservaciones y podrá facturar todas las que estén confirmadas; en el caso de tener asignado un servicio gastronómico, opcionalmente el actor podrá anexarle a la factura una orden con los productos y los tables consumidos.</p> <p>Si el actor elige la opción de modificar una factura, el sistema mostrará los datos que pueden ser editados y, una vez realizados los cambios, guardará las modificaciones.</p> <p>Si el actor elige la opción de ver una factura, el sistema mostrará toda la información detallada de la misma.</p>	
<b>Requerimientos funcionales</b>	
<b>RF20</b>	

*Descripción resumida Gestionar Factura*

Nombre	Gestionar Cliente.
Objetivo	Gestionar clientes en la aplicación.
Actor	Técnico en Promoción y Comunicación Integral.
Complejidad	Baja.
Nivel	Usuario
Precondiciones	Debe haberse autenticado el usuario en el sistema. Debe haberse generado el escritorio de trabajo de la aplicación de



	Reservas y el módulo Cliente.
Poscondiciones	Se modificó el cliente por el actor.
<b>Resumen del caso de uso</b>	
El caso de uso se inicia cuando el actor selecciona la opción que le permite modificar un cliente; el sistema mostrará los datos que pueden ser editables dentro de este y, una vez realizados los cambios, guardará las modificaciones, terminando así el caso de uso.	
<b>Requerimientos funcionales</b>	
<b>RF23</b>	

*Descripción resumida Gestionar Cliente*

Nombre	Gestionar Tables.
Objetivo	Gestionar los tables gastronómicos en la aplicación.
Actor	Gastronómico.
Complejidad	Media.
Nivel	Usuario.
Precondiciones	Debe haberse autenticado el usuario en el sistema. Deben existir los datos en el nomenclador nom_moneda. Debe haberse generado el escritorio de trabajo de la aplicación de Gastronomía y el módulo Table.
Poscondiciones	Se incluyó, modificó, vio o eliminó el table por el actor.
<b>Resumen del caso de uso</b>	
El caso de uso se inicia cuando el actor selecciona la opción que le permite realizar una acción sobre el table. El actor puede incluir, ver, modificar o eliminar un producto. En caso de que seleccione la opción de incluir un nuevo table, el sistema mostrará el listado de productos que se pagan en la moneda seleccionada para conformar el table; en el caso de seleccionar Moneda Nacional el precio final del table se calculará de forma automática, de lo contrario el actor debe especificarlo. Si el actor elige la opción de modificar un table, el sistema mostrará los datos que pueden ser editables dentro de este y, una vez realizados los cambios, guardará las modificaciones. Si el actor elige la opción de ver un table, el sistema mostrará toda su información. Si el actor elige la opción de eliminar un table, el	





sistema permitirá eliminarlo, terminando así el caso de uso.
Requerimientos funcionales
<b>RF16</b>

*Descripción resumida Gestionar Tables*

Nombre	Gestionar Espectáculos.
Objetivo	Gestionar los distintos tipos de espectáculos que podrán ser usados posteriormente en las reservaciones y facturas.
Actor	Administrador del Sistema.
Complejidad	Media.
Nivel	Usuario.
Precondiciones	Debe haberse autenticado el usuario en el sistema. Deben existir los datos en el nomenclador nom_moneda. Debe haberse generado el escritorio de trabajo de la aplicación de Administración y el módulo Espectáculos.
Poscondiciones	Se incluyó, modificó o eliminó el espectáculo por el actor.

Resumen del caso de uso
El caso de uso se inicia cuando el actor selecciona la opción que le permite realizar una acción sobre el espectáculo. El actor puede incluir, eliminar y modificar un espectáculo. En caso de que seleccione la opción de incluir un espectáculo, el sistema dará la posibilidad de insertar los datos que se necesitan para llenar esta plantilla. Si el actor elige la opción de modificar un espectáculo, el sistema mostrará los datos que pueden ser editables dentro del espectáculo y, una vez realizados los cambios, guardará las modificaciones. Si el actor elige la opción de eliminar un espectáculo, el sistema permitirá eliminarlo, terminando así el caso de uso.
Requerimientos funcionales
<b>RF4</b>

*Descripción resumida de Gestionar Espectáculos*

Nombre	Gestionar Locales.
--------	--------------------



Objetivo	Gestionar los locales que podrán ser usados posteriormente en los alquileres de las reservaciones y facturas.
Actor	Administrador del Sistema.
Complejidad	Media.
Nivel	Usuario.
Precondiciones	Debe haberse autenticado el usuario en el sistema. Deben existir los datos en el nomenclador nom_moneda. Debe haberse generado el escritorio de trabajo de la aplicación de Administración y el módulo Locales.
Poscondiciones	Se incluyó, modificó, filtró, activó o desactivó el local por el actor.
Resumen del caso de uso	
<p>El caso de uso se inicia cuando el actor selecciona la opción que le permite realizar una acción sobre el local. El actor puede incluir, modificar, activar o desactivar un local. En caso de que seleccione la opción de incluir un local, el sistema dará la posibilidad de insertar los datos que se necesitan para llenar esta plantilla. Si el actor elige la opción de modificar un local, el sistema mostrará los datos que pueden ser editables dentro del local y, una vez realizados los cambios, guardará las modificaciones. Si el actor elige la opción de filtrar el listado de locales, el sistema mostrará el listado con los locales que coinciden con los criterios especificados. Si el actor elige la opción de desactivar un local, el sistema permitirá eliminarlo, temporalmente. Si el actor elige la opción activar del listado de locales temporalmente eliminados o desactivados, el local volverá a estar activo en el listado inicial, terminando así el caso de uso.</p>	
Requerimientos funcionales	
<b>RF2</b>	

*Descripción resumida Gestionar Locales*

Nombre	Consultar Historial de Modificaciones de Reservación
Objetivo	Consultar los las modificaciones hechas a una reservación.
Actor	Directivo.
Complejidad	Baja.



Nivel	Usuario.
Precondiciones	Debe haberse autenticado el usuario en el sistema. Debe haberse generado el escritorio de trabajo de la aplicación de Reservas y el módulo Reservaciones. Se consultó el historial de modificaciones por el actor.
Poscondiciones	Se consultó el historial de modificaciones de las reservas por el actor.
Resumen del caso de uso	
<p>El caso de uso el actor selecciona la opción “Ver” del listado de modificaciones de las reservas; el sistema muestra un instancia de esta reservación antes y después de haber sido modificada y resalta los cambios en tres colores distintos:</p> <ul style="list-style-type: none"> <li>✓ Verde: Se agregó el elemento a la reservación (por ejemplo: se agregó un servicio nuevo).</li> <li>✓ Amarillo: Se modificó o cambió el valor del elemento por otro.</li> <li>✓ Rojo: Se eliminó el elemento.</li> </ul>	
Requerimientos funcionales	
<b>RF22.1</b>	

*Descripción resumida Consultar Historial de Modificaciones de Reservación*

Nombre	Consultar Historial de Modificaciones de Factura.
Objetivo	Consultar los las modificaciones hechas a una factura.
Actor	Directivo.
Complejidad	Baja.
Nivel	Usuario.
Precondiciones	Debe haberse autenticado el usuario en el sistema. Debe haberse generado el escritorio de trabajo de la aplicación de Reservas y el módulo Facturas. Se consultó el historial de modificaciones por el actor.
Poscondiciones	Se consultó el historial de modificaciones de las facturas por el actor.
Resumen del caso de uso	
El caso de uso el actor selecciona la opción “Ver” del listado de modificaciones de las	



facturas; el sistema muestra un instancia de esta factura antes y después de haber sido modificada y resalta los cambios en tres colores distintos:

- ✓ Verde: Se agregó el elemento a la reservación.
- ✓ Amarillo: Se modificó o cambió el valor del elemento por otro.
- ✓ Rojo: Se eliminó el elemento.

#### Requerimientos funcionales

#### **RF22.2**

*Descripción resumida Consultar Historial de Modificaciones de Factura*

Nombre	Consultar Reportes.
Objetivo	Consultar los reportes generados para una mejor planificación.
Actor	Directivo.
Complejidad	Baja.
Nivel	Usuario.
Precondiciones	Debe haberse autenticado el usuario en el sistema. Debe haberse generado el escritorio de trabajo de la aplicación de Administración y el módulo Reportes.
Poscondiciones	Se consultó el reporte por el actor.

#### Resumen del caso de uso

El actor puede consultar distintos tipos de reportes:

- ✓ En el caso de que seleccione la opción de ver la hoja de operaciones semanal por alquiler de locales, el sistema mostrará el listado de los locales con sus respectivos horarios de alquiler; el actor especificará entonces la semana que desea consultar y el sistema resaltará los locales que han sido alquilados y por quién.
- ✓ En el caso de que seleccione la opción de ver la hoja de operaciones semanal por gastronomía en locales, el sistema mostrará el listado de los locales con sus respectivos horarios de comida; el actor especificará entonces la semana que desea consultar y el sistema mostrará, mediante una barra, qué tan llenos están los locales en cada uno de los horarios.
- ✓ En el caso de que el actor seleccione ver el dinero recaudado por alquiler de locales, mensualmente, el sistema mostrará una gráfica de barras que represente esta



información.

- ✓ En el caso de que el actor seleccione ver el dinero recaudado por gastronomía, mensualmente, el sistema mostrará una gráfica de barras que represente esta información.
- ✓ En el caso de que el actor seleccione ver el dinero recaudado por espectáculos, mensualmente, el sistema mostrará una gráfica de barras que represente esta información.
- ✓ En el caso de que el actor seleccione ver el dinero recaudado por cada uno de los servicios, mensualmente, el sistema mostrará una gráfica de barras que represente esta información.
- ✓ En el caso de que el actor seleccione ver la cantidad de espectáculos reservados, mensualmente, el sistema mostrará una gráfica de línea que represente esta información.
- ✓ En el caso de que el actor seleccione ver la cantidad de alquileres reservados,, mensualmente el sistema mostrará una gráfica de línea que represente esta información.
- ✓ En el caso de que el actor seleccione ver la cantidad de servicios gastronómicos reservados, mensualmente el sistema mostrará una gráfica de línea que represente esta información.
- ✓ En el caso de que el actor seleccione ver la cantidad de servicios reservados, mensualmente, el sistema mostrará una gráfica de barras que represente esta información.
- ✓ En el caso de que el actor seleccione ver la cantidad de clientes nacionales y extranjeros que han reservado algún servicio, el sistema mostrará una gráfica de línea que represente esta información.

Cada uno de los reportes podrá ser imprimido y descargado en los siguientes formatos: PDF, JPEG, PNG y SVG.

Requerimientos funcionales

**RF25**

*Descripción resumida Consultar Reportes*



## 2.7 Prototipos de interfaz de usuario

Se desarrollaron los prototipos de interfaz de usuario de varias de las principales pantallas. Si bien es cierto que los prototipos iniciales sufrieron muchos cambios a través de las iteraciones en el desarrollo del proyecto, el objetivo se cumplió en el momento: entender las principales pantallas/páginas de la interfaz de usuario, lo que permitió identificar la apariencia básica del sistema. Las principales interfaces fueron modeladas con herramientas simples, como recomienda AUP; las imágenes escaneadas de las principales pantallas se muestran a continuación:

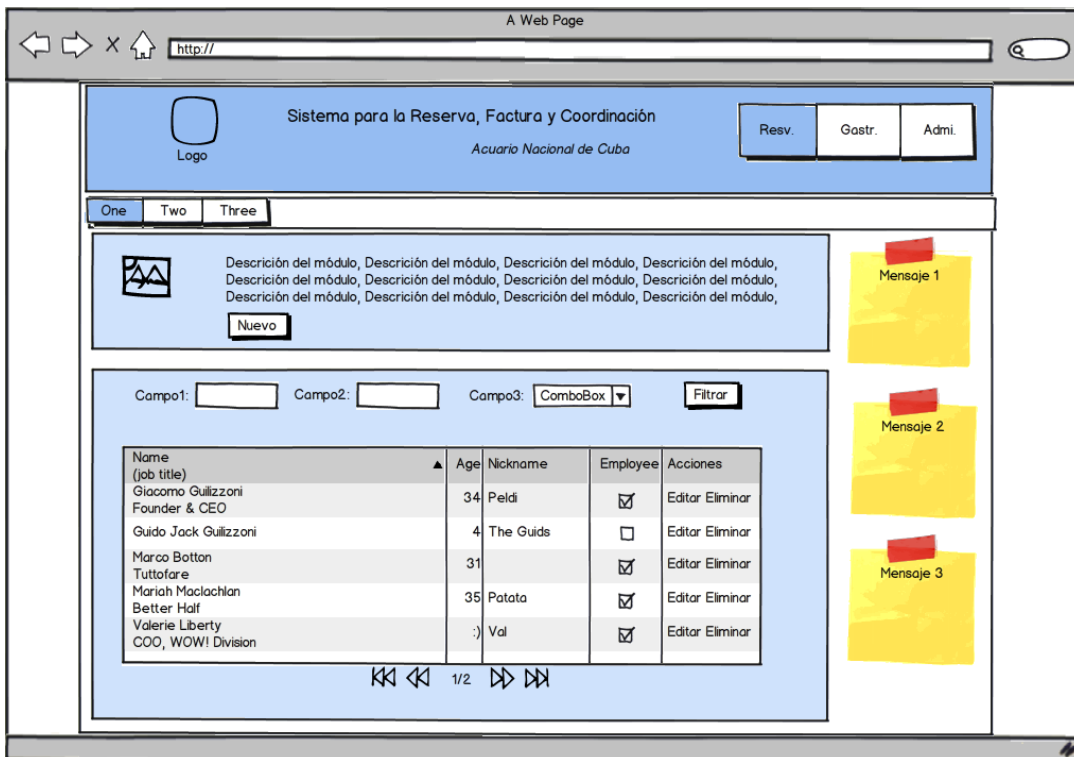


Figura 8. Prototipo de interfaz de usuario Consultar Reservación

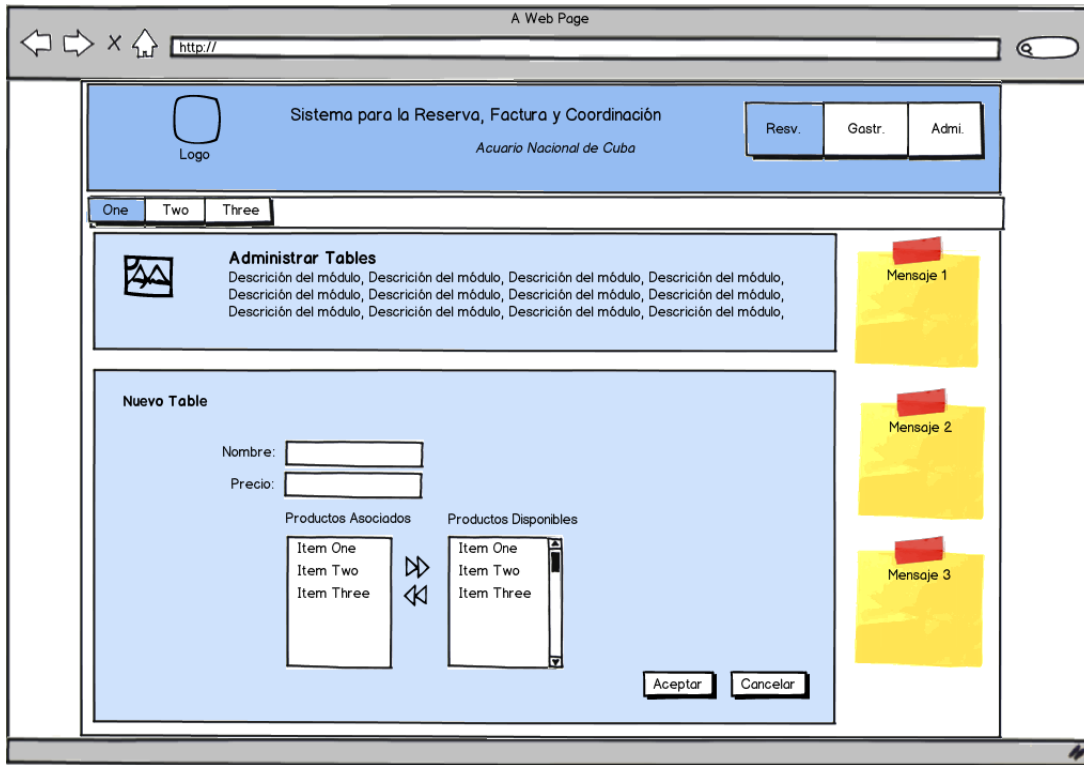


Figura 9. Prototipo de interfaz de usuario Nuevo Table



Figura 10. Prototipo de interfaz de usuario Reportes



The screenshot shows a web browser window with the URL 'http://'. The page title is 'Sistema para la Reserva, Factura y Coordinación' and the subtitle is 'Acuario Nacional de Cuba'. There are three navigation buttons: 'Resv.', 'Gastr.', and 'Admi.'. Below the header, there are three tabs: 'One', 'Two', and 'Three'. The main content area is divided into three sections:

- Administración de Reservas:** A section with a small icon and a list of descriptions.
- Nueva Reservación:** A form with three sub-sections:
  - Datos de la Entidad:** Fields for 'Nombre:', 'Tipo:', 'Correo:', 'Teléfono:', 'CI:', and 'Dirección:'.
  - Datos de la Reservación:** Fields for 'Hora entrada:', 'Hora salida:', 'Fecha:', and a text area for 'Observaciones:'.
  - Datos de los Servicios:** A table with columns 'Servicio', 'Local', 'Cantidad de personas', and 'Horario'. Below the table are four buttons: '+ Espectáculo', '+ Gastronómico', '+ Alquiler', and '+ Entrada'.
- Mensajes:** A vertical sidebar on the right with three yellow sticky notes labeled 'Mensaje 1', 'Mensaje 2', and 'Mensaje 3'.

Figura 11. Prototipo de interfaz de usuario Nueva Reservación

## 2.8 Vista de CU arquitectónicamente significativos

Esta vista permitirá definir los escenarios o casos de uso que serán de interés en cada iteración del ciclo de vida del proyecto; así como describir aquellos escenarios o casos de uso que tengan un alto grado de importancia para la arquitectura, puesto que encapsulan gran parte de la funcionalidad central del sistema. A estos se les llama casos de uso arquitectónicamente significativos, porque son aquellos que describen funcionalidades imprescindibles para el sistema y a través de ellos se valida la arquitectura propuesta para este.



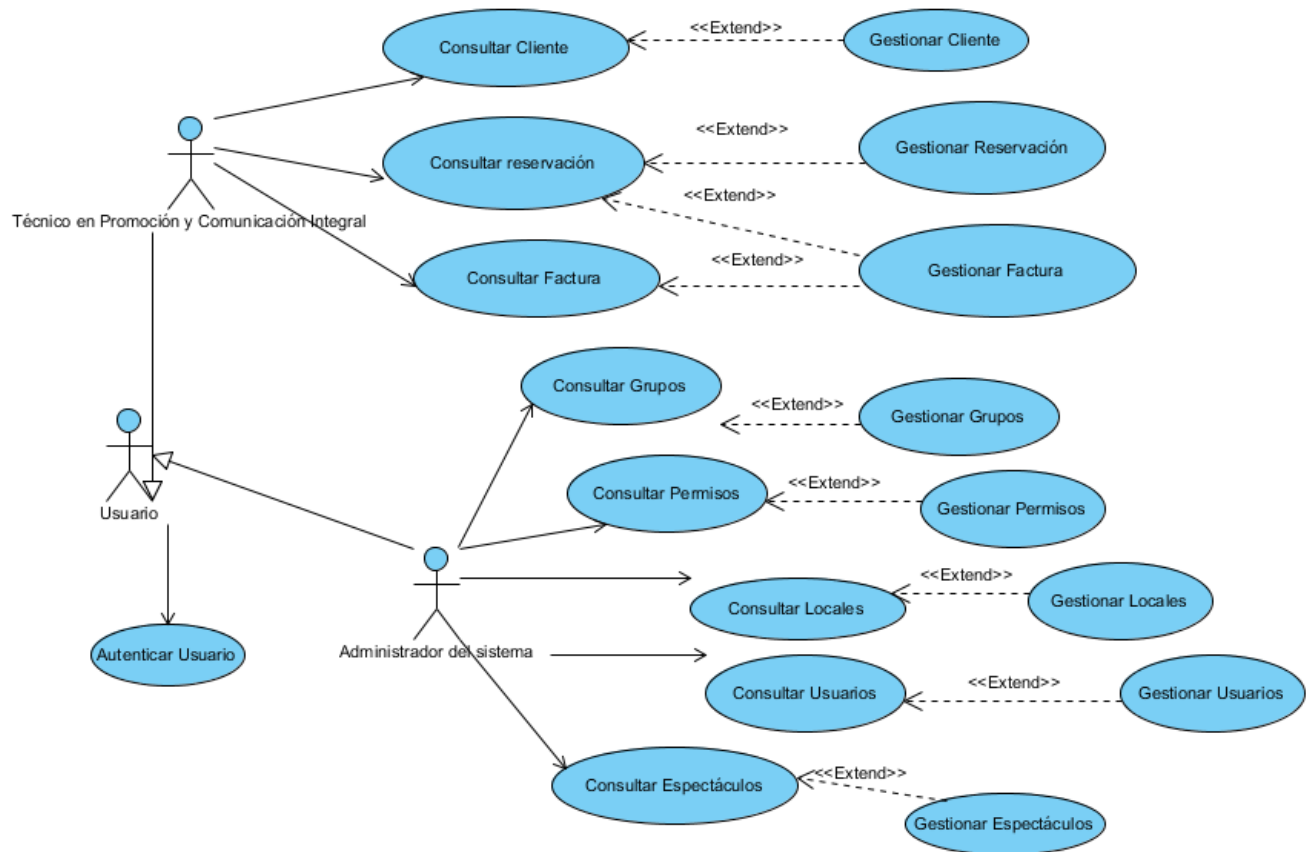


Figura 12. Vista de Casos de Uso arquitectónicamente significativos



## 2.9 Patrones de diseño utilizados en la solución

Un patrón es una descripción de un problema y la solución, a la que se le da un nombre, y que además se puede aplicar a nuevos contextos.

### GRASP (patrones de software para la asignación general de responsabilidad)

Los patrones GRASP describen los principios fundamentales de diseño de objetos para la asignación de responsabilidades. Constituyen un apoyo para la enseñanza, que ayuda a entender el diseño de objeto esencial y aplica el razonamiento para el diseño de una forma sistemática, racional y explicable.

- ✓ Experto: la responsabilidad de realizar una labor es de la clase que tiene o puede tener los datos involucrados. Una clase contiene toda la información necesaria para realizar la labor que tiene encomendada. Este patrón es muy utilizado al trabajar con Symfony, ya que con la librería Doctrine para mapear la base de datos, se evidencia su continua utilización. Doctrine genera las clases para la gestión de las entidades con las responsabilidades fijadas, según lo plantea el patrón. (22)
- ✓ Creador: se asigna la responsabilidad de que una clase B cree un objeto de la clase A, solamente cuando: B contiene a A, B es una agregación (o composición) de A, B almacena a A, B tiene los datos de inicialización de A (datos que requiere su constructor), B usa a A. En la clase actions.class.php se encuentran las acciones específicas para el módulo visita, solicitante, visitante, área y visitante. En cada acción se crean los objetos de las clases, que son los que crean instancias de las clases del modelo, por lo que se demuestra que la clase actions.class.php es la “creadora” de estas entidades. Un ejemplo del uso de este patrón es en la acción executeCreate(\$request), que crea instancias de las clases entidades que contienen los datos a utilizar. (22)
- ✓ Alta cohesión: cada elemento del diseño debe realizar una labor única dentro del sistema, no desempeñada por el resto de los elementos y auto-identificable, manteniendo la complejidad dentro de los límites manejables. El utilizar Symfony permite crear y trabajar con clases, con una alta cohesión, ya que da la posibilidad de organizar el trabajo de acuerdo con la estructura del proyecto. Un ejemplo claro es la clase actions.class.php, ya que posee funcionalidades con una fuerte relación, que tienen un sentido común y un único propósito, a la vez que son las responsables de controlar las acciones de las plantillas. (22)
- ✓ Bajo acoplamiento: debe haber poca dependencia entre las clases para poder extraerlas de un modo independiente y reutilizarlas. Este patrón se pone de manifiesto a partir de que el modelo



posee las clases que implementan la lógica de negocio y acceso a datos, y estas clases no tienen relaciones con las de la vista o el controlador, por lo que la dependencia es baja.

- ✓ Controlador: asigna a clases específicas la responsabilidad de controlar el flujo de eventos del sistema. Esto facilita la centralización de actividades (validaciones, seguridad, etc.). El controlador no las realiza, sino las delega en otras clases con las que mantiene un modelo de alta cohesión. Este patrón se pone de manifiesto en las clases `sfFrontController`, `sfWebFrontController`, `sfContext`, los "actions" y el `index.php` del ambiente. La arquitectura del framework (MVC) brinda una capa específicamente para los controladores, que son el núcleo de este, y especifica la presencia de este patrón. (22)

#### GOF (banda de los cuatro)

- ✓ Decorador/Decorator: patrón estructural que se pone de manifiesto a través del archivo `Layout.php`, que es el que contiene el Layout de las páginas. Este archivo posee el código HTML que es común en todas las páginas de la aplicación, por lo que también es conocido como plantilla global. Finalmente, el Layout es el que decora cada plantilla y constituye una implementación de este patrón. (22)
- ✓ Solitario/Singleton: este es un patrón de creación que es utilizado en la clase `sfRouting` utilizada por el controlador frontal (`sfWebFrontController`); esta clase se encarga de enrutar todas las peticiones que se realizan a la aplicación, por lo que es muy utilizada. (22)
- ✓ Patrón Command: es un patrón de comportamiento. Se demuestra el uso de este patrón en la clase `sfWebFrontController`, específicamente en el método `dispatch`, el cual es el encargado de determinar cuál módulo y acción utilizar, en dependencia de la solicitud realizada por el usuario. (22)
- ✓ Patrón Registry: este patrón es un medio simple y eficiente de compartir datos y objetos en la aplicación, sin tener que preocuparse de mantener numerosos parámetros o hacer uso de variables globales, por lo que brinda muchos beneficios a los programadores en la POO. Se pone de manifiesto en la clase `sfConfig`, la cual se encarga de almacenar todas las variables de uso global en la aplicación. (22)

## **2.10 Diagrama de despliegue de la aplicación**

El modelo de despliegue define la arquitectura física del sistema. Se usa para modelar de manera detallada los nodos físicos y las asociaciones de comunicación que existen entre ellos. Del mismo modo queda especificado qué hardware, sistemas operativos, software de interfaces y soporte



conformarán el nuevo sistema. A continuación se presenta el diagrama de despliegue propuesto para el sistema.

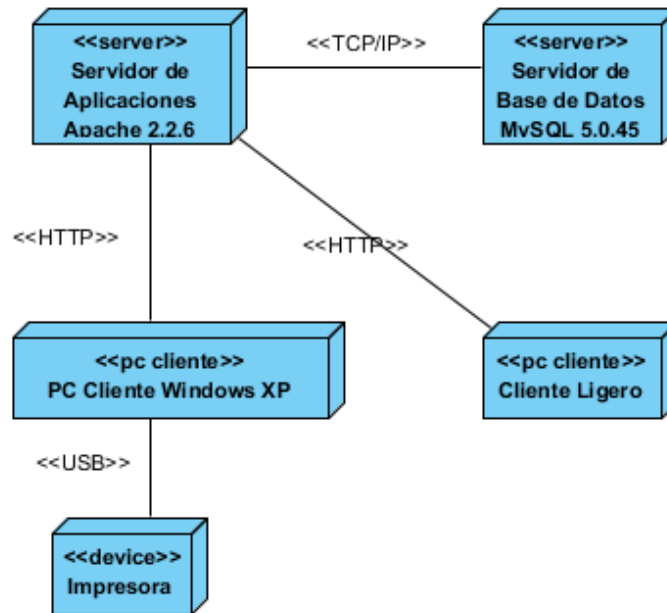


Figura 13. Diagrama de despliegue

### 2.11 Diagrama de las clases del diseño utilizando estereotipos web

El diagrama de clases permite tener una visión estática de cómo se realizará la implementación de los casos de uso. En esta sección se muestran varios diagramas, representativos del sistema, no se incluyen todos por razones de formato, pero se pueden encontrar en su totalidad en los anexos.

La figura muestra el paquete Vista, donde se encuentran los componentes fundamentales para mostrar al usuario la información. El paquete Controlador contiene la clase de las acciones del módulo, o que representa el punto de contacto de la vista con el resto del sistema. Finalmente las clases del Modelo, en este caso las relacionadas con los clientes, interactúan con el sistema de almacenamiento persistente.

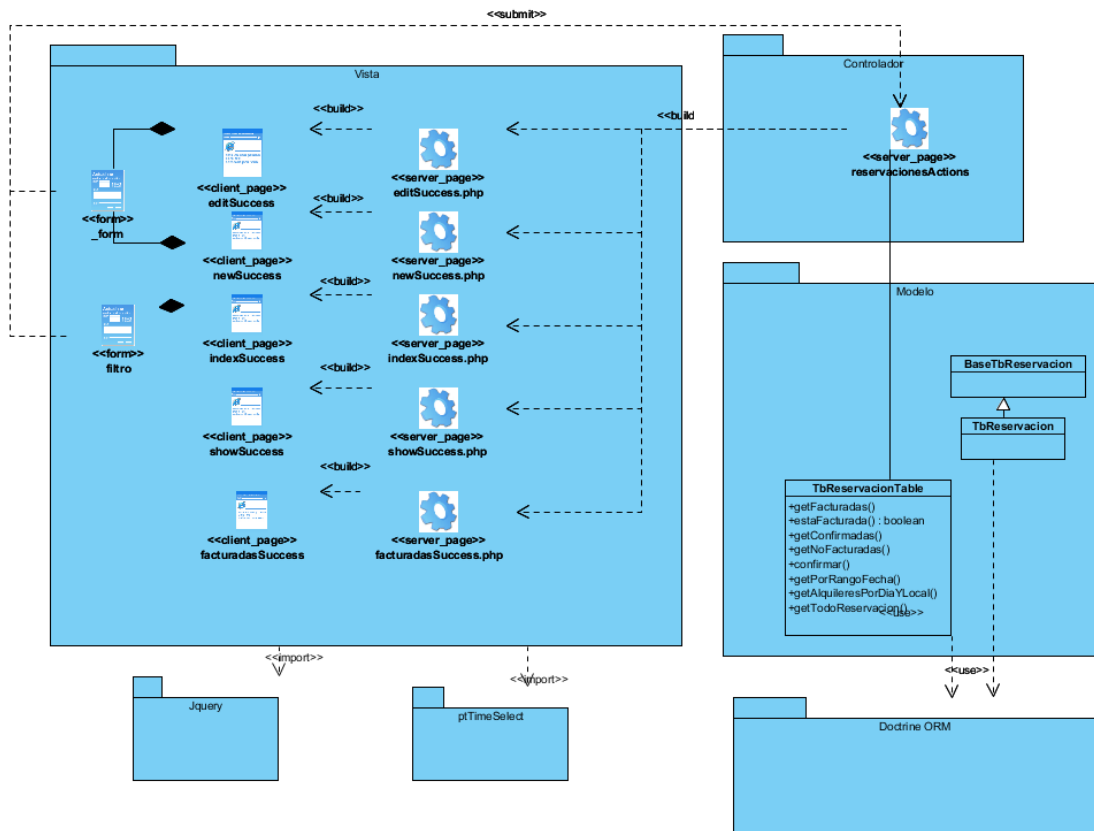


Figura 14. Diagrama de Clases del módulo Reservasiones

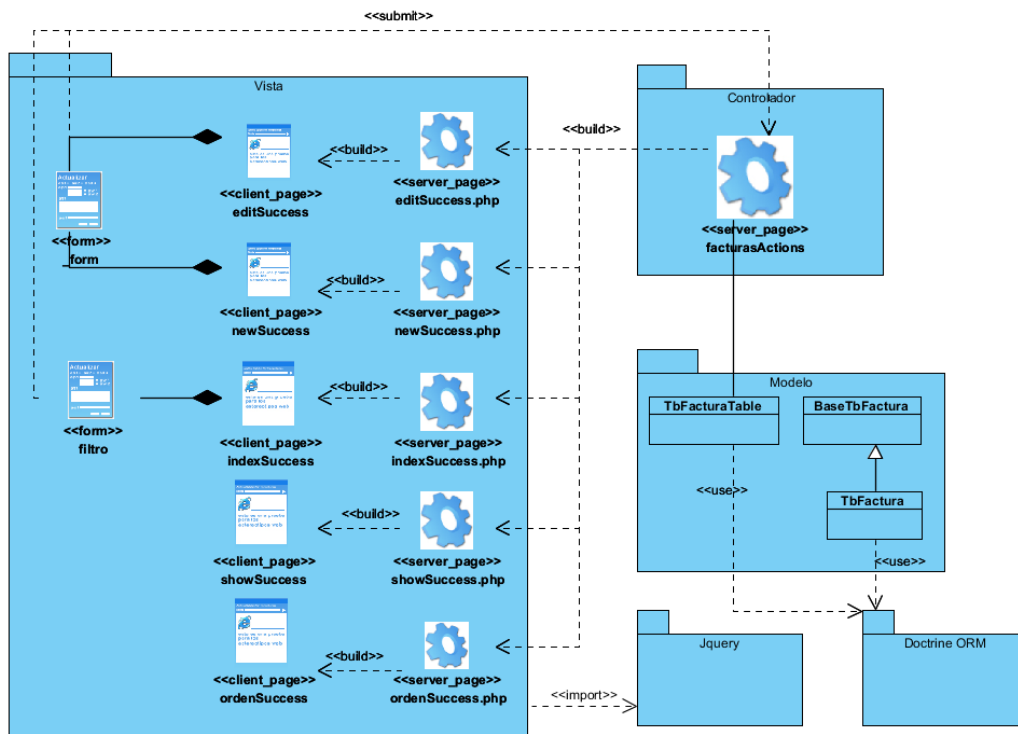




Figura 15. Diagrama de Clases del módulo Factura

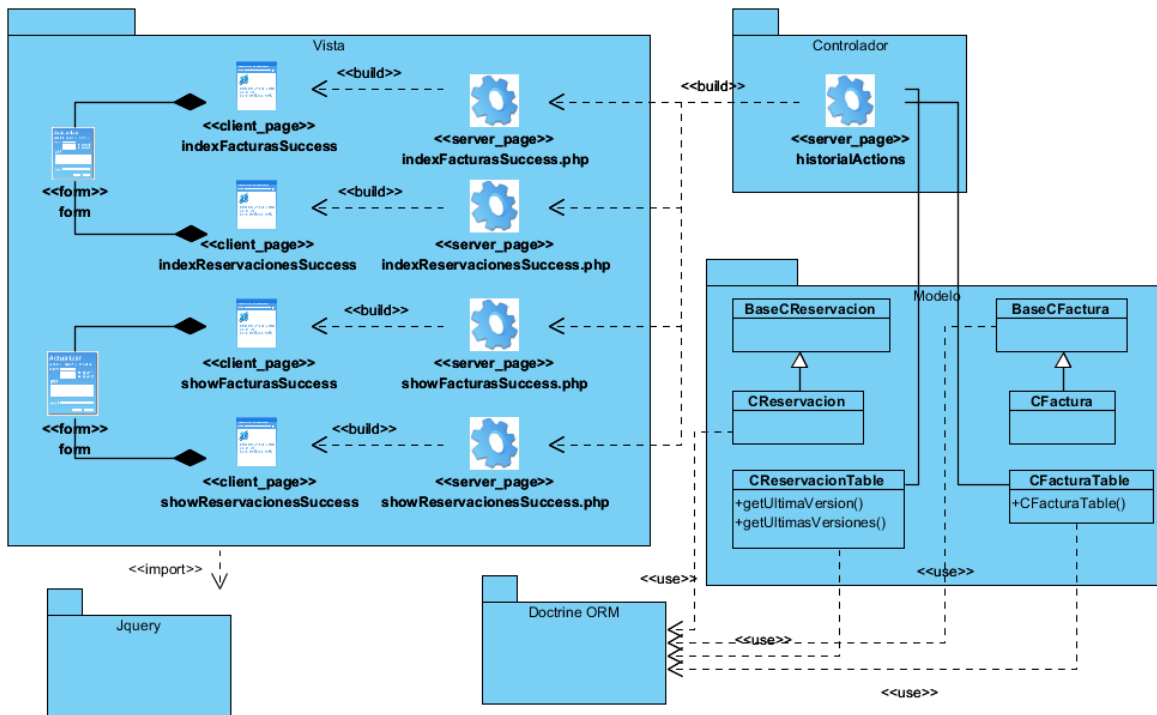


Figura 16. Diagrama de Clases del módulo Historial

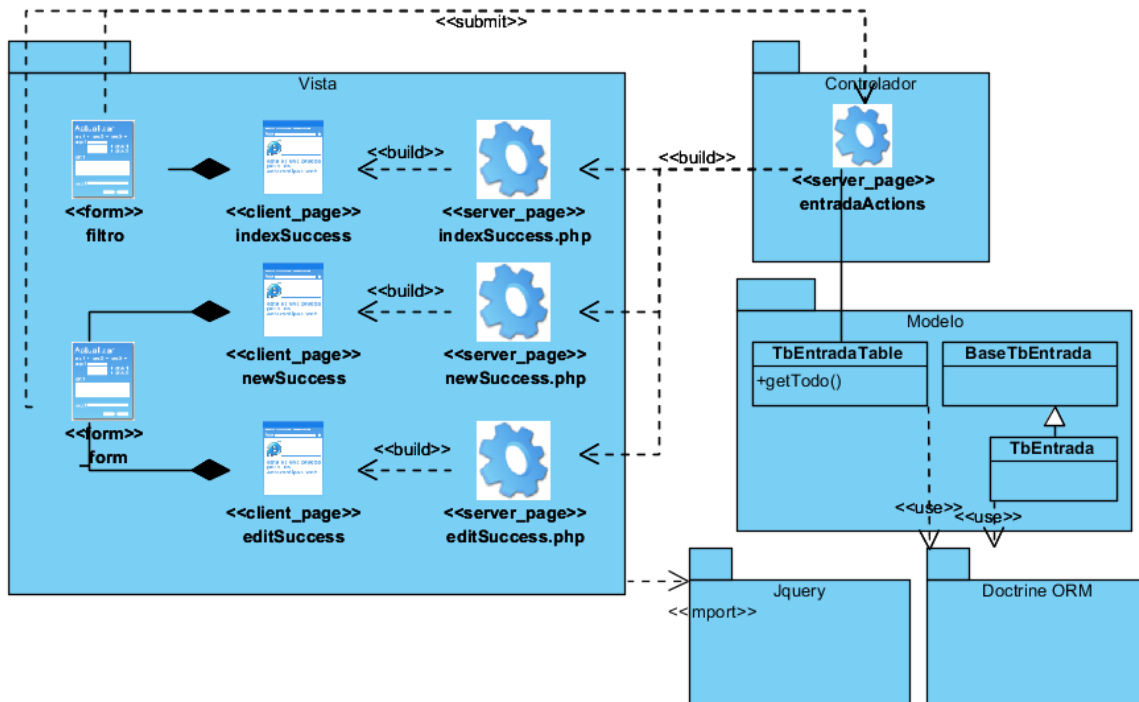


Figura 17. Diagrama de Clases del módulo Entrada

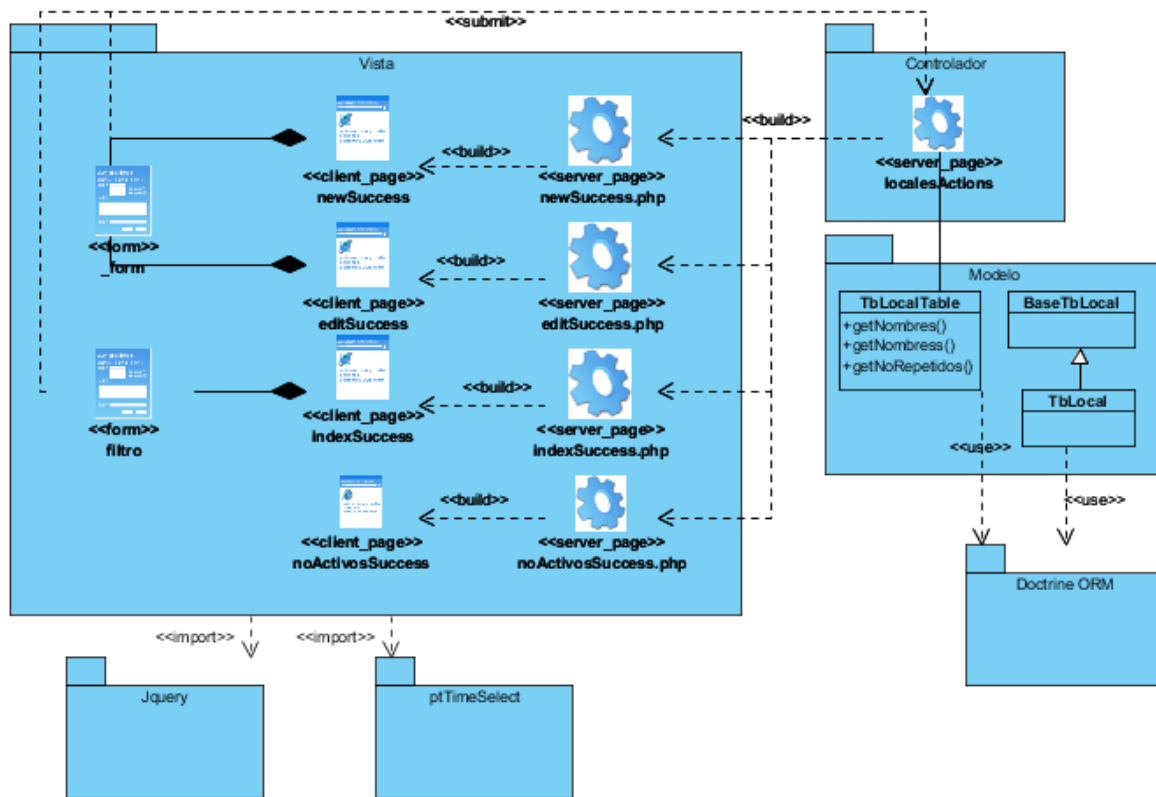


Figura 18. Diagrama de Clases del módulo Locales

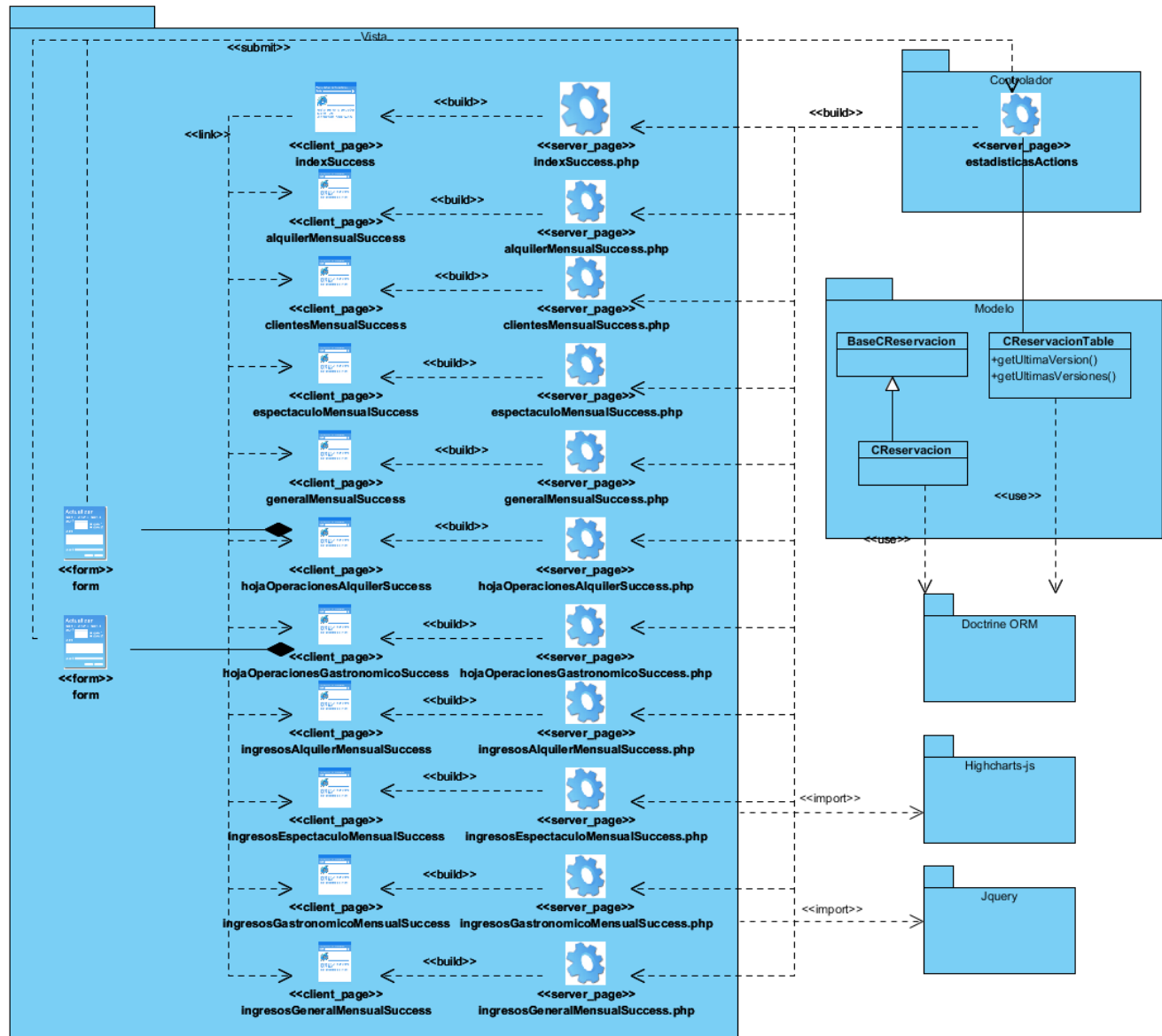


Figura 19. Diagrama de Clases del módulo Reportes

## 2.12 Modelo de Implementación

El propósito principal de la implementación es desarrollar la arquitectura y el sistema como un todo, se realiza a partir de los artefactos del diseño. El modelo de implementación es la entrada principal de las etapas de prueba que siguen a la implementación.

### 4.2.1 Diagrama de componentes





Los diagramas de componentes se utilizan para estructurar el modelo de implementación en términos de subsistemas de implementación y mostrar las relaciones entre los elementos de implementación.

El uso más importante de estos diagramas es mostrar la estructura de alto nivel del modelo de implementación, especificando que:

- ✓ Muestra las dependencias entre tipos de componentes.
- ✓ Organiza los subsistemas de implementación en capas.

A continuación se presentan varios de los diagramas de componentes propuestos para algunos módulos representativos del sistema, los restantes se pueden encontrar en los anexos:

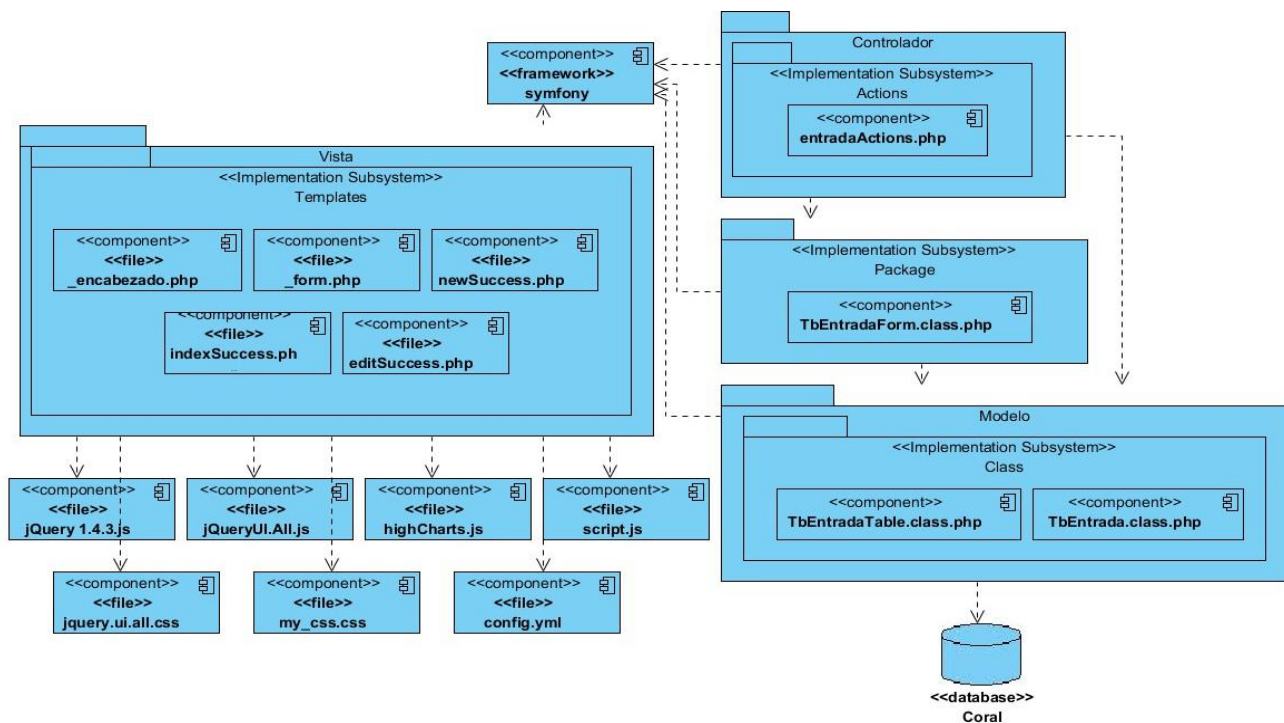


Figura 20. Diagrama de componentes del módulo Entradas

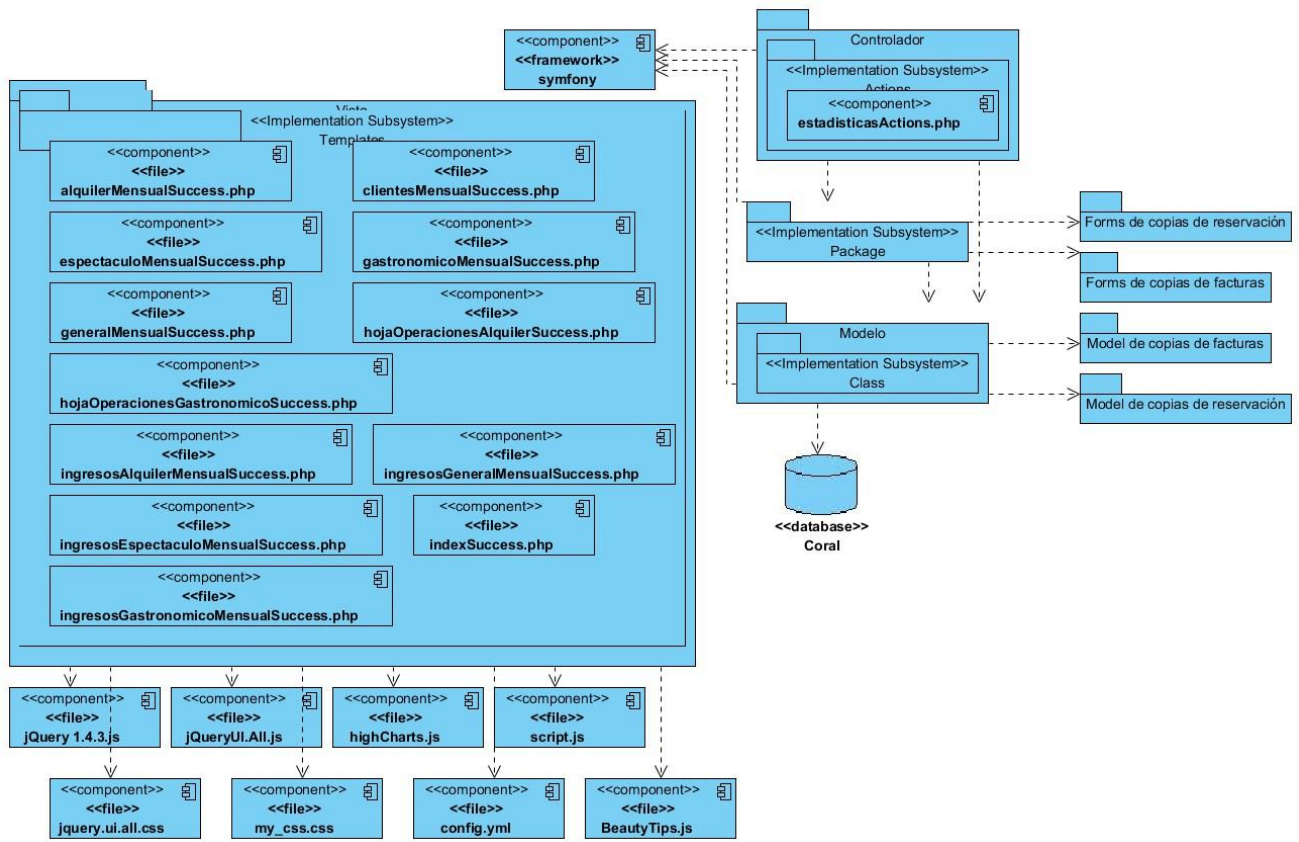


Figura 21. Diagrama de componentes del módulo Reportes

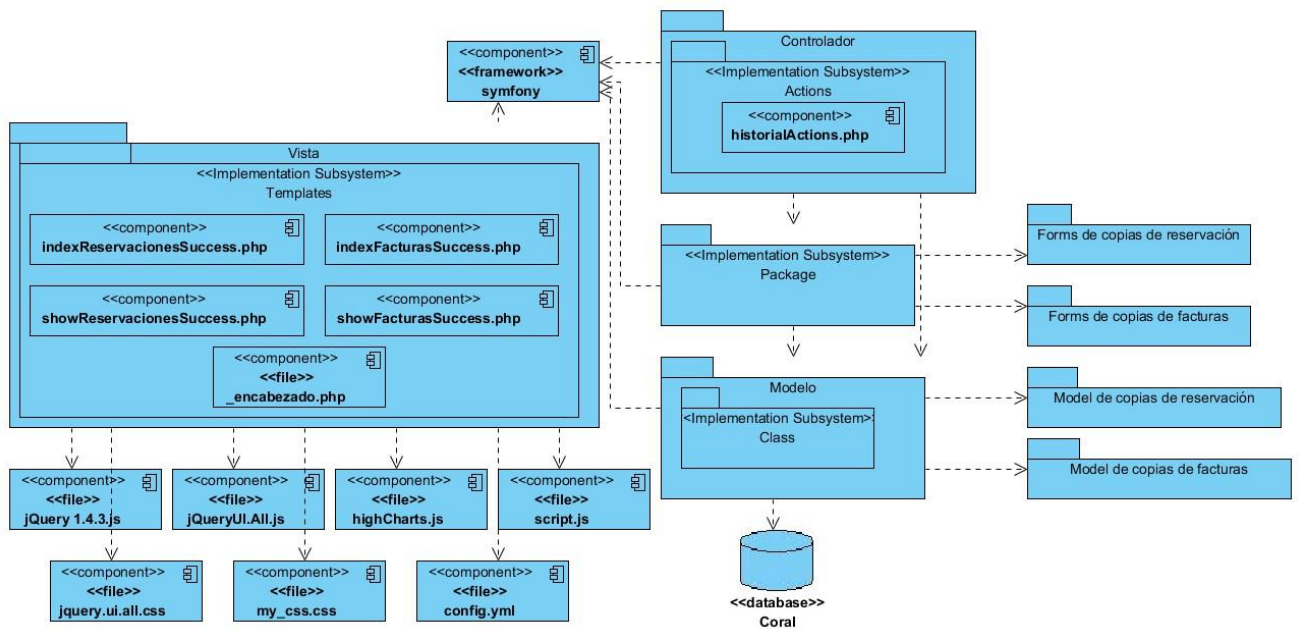


Figura 22. Diagrama de componentes del módulo Historial

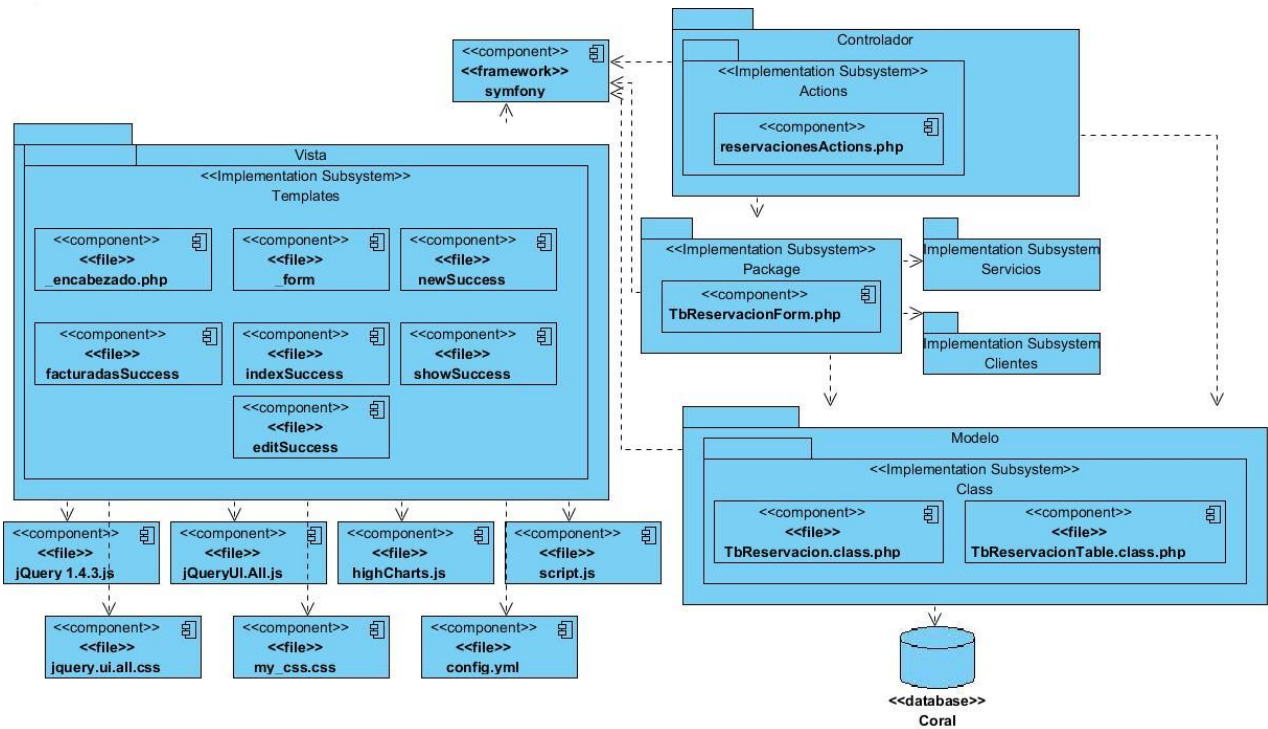


Figura 23. Diagrama de componentes del módulo Reservas

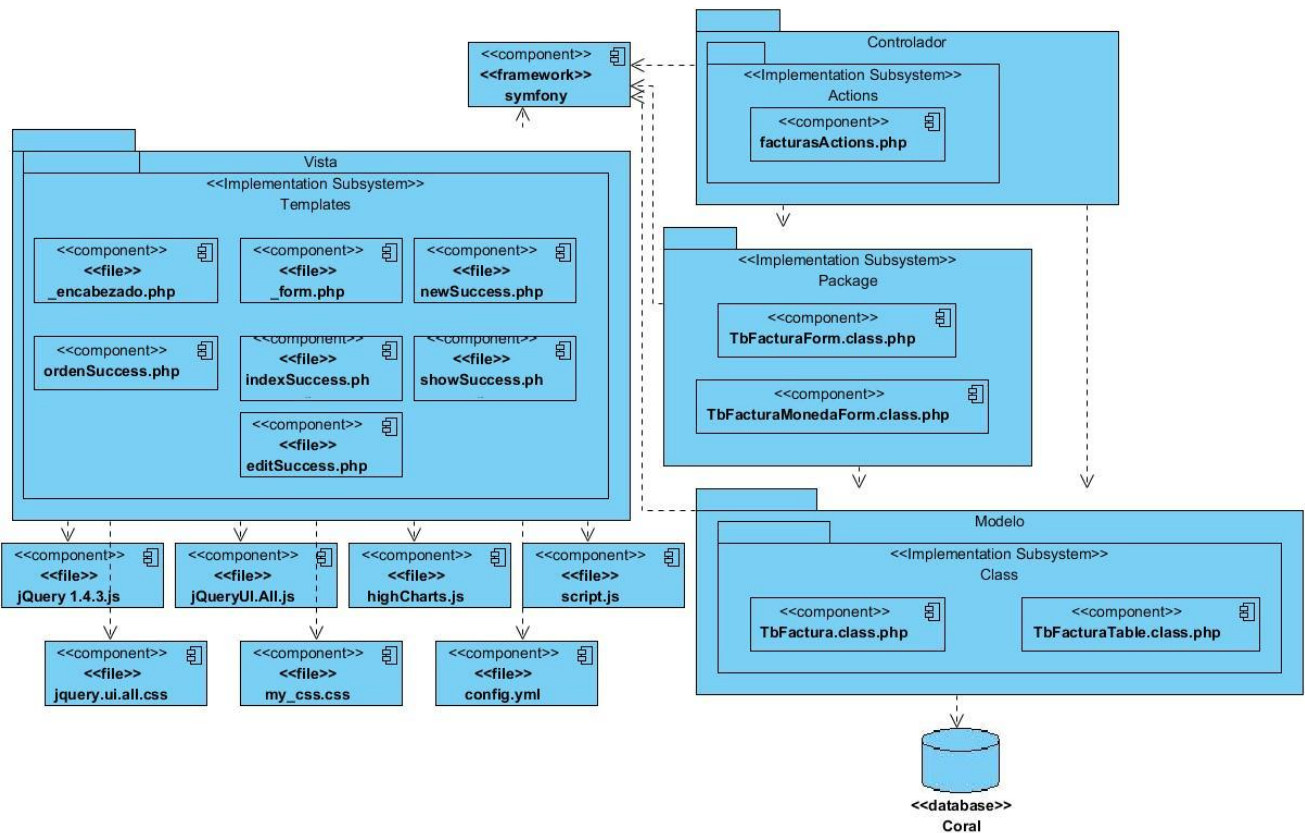


Figura 24. Diagrama de componentes del módulo Facturas





## **2.14 Conclusiones**

La captura de requisitos, haciendo uso de las técnicas de recopilación de información, constituyó el punto de entrada para la elaboración de los Casos de Uso y el flujo de trabajo de análisis y diseño. Se obtuvo, como resultado, los diagramas de clases del diseño, el diagrama de despliegue, el modelo de datos y finalmente los diagrama de componentes, que corresponden al modelo de implementación. Para estos últimos se consideró, principalmente, las facilidades que los frameworks aportaban y los patrones de diseño que, sin forzar al programador, ayudan a la solución propuesta.



## Capítulo 3 Estudio de factibilidad

La estimación del costo de un software forma una parte importante y necesaria en el desarrollo del mismo. Generalmente esta se realiza en la etapa inicial del desarrollo cuando lo que se tiene es a lo sumo una visión general del proyecto. Estudios realizados demuestran que una mala estimación da como resultados atrasos en la entrega del producto, costos por encima de lo planificado y riesgo de que el producto a entregar no cumpla con el objetivo para el que fue construido. De dicha experiencia viene la necesidad de estimar correctamente los esfuerzos de un proyecto.

Con una correcta estimación no solo se obtiene el costo sino también la estimación de los resultados del proyecto y los valores de tiempo y recursos requeridos.

La estimación tiene como objetivo determinar la posibilidad real de llevar a cabo el desarrollo de un proyecto, es decir, el estudio de factibilidad de acuerdo a las diferentes restricciones.

Las restricciones están dadas por las características propias del proyecto y/o grupo de trabajo:

- ✓ Organizativas: por la estructura, procesos y personas.
- ✓ Económicas: se analiza la relación costo-beneficio.
- ✓ Técnicas: referente a las habilidades que posee el grupo de trabajo, experiencia con que cuenta y los recursos disponibles.
- ✓ Tiempo: se tiene en cuenta cuando existen requerimiento de fecha de cumplimiento.

### 3.1 Método de estimación.

Existen varios métodos de estimación para definir la factibilidad de un proyecto; en la Universidad de las Ciencias Informáticas se desarrolló uno debido a la necesidad de estimar el tamaño, costo y esfuerzo requerido para desarrollar un producto software. Dado que la misma no cuenta con una base histórica, se tuvo en cuenta los datos dispersos de algunos proyectos y la evaluación de algunos factores que, según criterio de expertos, pueden influir en las estimaciones del proyecto.

(23) Entre los que se encuentran:

- ✓ El Factor Cliente (FC): permite determinar un nivel de incertidumbre asociado a la madurez que tiene el cliente en materia de asimilación de proyectos informáticos.
- ✓ Factor de Valor Agregado (VA): determina algunas acciones que pueden representar esfuerzo adicional al desarrollo de software, se incluye en todas las etapas, fundamentalmente en la de Análisis y Diseño.



- ✓ Factor de Ambiente (FA): es interno, se determina a través de analizar las condiciones de la estructura de producción en la que se desarrollará el proyecto, no influye en el precio final, pero si en las valoraciones para la realización del proyecto y se utilizará como indicador para saber si la estructura de producción ha creado condiciones para cumplir los compromisos. Se aplicará en el precio de negociación y previo a la contratación.
- ✓ Factor Complejidad Técnica (CT): es asociado a la complejidad de la tecnología y requerimientos no funcionales del sistema, influye directamente en el esfuerzo y fundamentalmente en la etapa de implementación.

El presente trabajo ha descrito la realización del software en todo su ciclo, utilizando los recursos de la Universidad y propios de la autora, la cual considera importante incluir este análisis con el objetivo de demostrar la factibilidad de la implementación del producto, teniendo en cuenta el tiempo en que se realizó y el costo que podría haber acarreado al Acuario Nacional en caso de haber contratado a una empresa de desarrollo de software para la realización del sistema. El estudio de la factibilidad fue realizado para el proyecto utilizando el método de estimación de la UCI. Se utilizó el método de estimación definido en Calisoft para los proyectos productivos de la universidad: <http://estima.prod.uci.cu/>

### 3.2 Información necesaria.

Lo primero que se tomó en cuenta para el estudio de factibilidad es la información básica y necesaria del producto a desarrollar, como la cantidad de módulos y la complejidad de los mismos, las personas que trabajaran en su desarrollo, etc.

La tabla muestra los módulos o paquetes de funcionalidad que forman parte de los datos de entrada para aplicar el método de estimación, los mismos están clasificados de acuerdo a la complejidad que prima en los casos de uso relacionados con estos.

Lista de Paquetes Funcionales		
Nro.	Nombre del Paquete Funcional	Categoría
1	Módulo Entradas	Pequeño
2	Módulo Espectáculos	Mediano
3	Módulo Estadísticas	Pequeño
4	Módulo Locales	Mediano
5	Módulo Clientes	Pequeño
6	Módulo Facturas	Grande
7	Módulo Historial	Pequeño
8	Módulo Principal	Pequeño



9	Módulo Productos	Mediano
10	Módulo Reservas	Grande
11	Módulo de Seguridad	Mediano
12	Módulo Table	Mediano

La tabla muestra la cantidad de paquetes funcionales del sistema, clasificando los mismos de acuerdo a distintos criterios. Se debe especificar que el paquete que se reutilizará es plugin de Symfony sfGuardUser, que se utilizará para el manejo de los usuarios, los restantes serán desarrollados por el autor.

Formulario	Cantidad
Paquetes Funcionales con Reutilización	1
Paquetes Funcionales a Desarrollar	11
Sistemas a Integrar o Legar	0
Paquetes Funcionales del Sistema	12
Sistemas a Integrar o Legar Desconocidos	
Dispositivos Externos	
Dispositivos Externos con Driver o SDK	
Estándares y Normas a cumplir	
Estándares y Normas específicas.	

La tabla muestra el tiempo en semanas que se dedicará a las pruebas, despliegue y soporte, se debe considerar que todo este proceso en la presente investigación acontecerá en una sola semana, pero en la tabla no se especifica así, ya que se entiende por **Total** la suma de todas las semanas y el software no permite la entrada de números decimales en estos campos.

Actividad	Tiempo (Semanas)
Pruebas aceptación	1
Pruebas piloto	0
Despliegue	0
Soporte	0
<b>Total</b>	<b>1</b>

Se contará además con un consultor de alto nivel el MSc. Francisco Longino Franquiz, vicedirector comercial del Acuario Nacional y un profesional.

Rol	Cantidad
Técnico	
Profesional	1
Consultor de Alto Nivel	1
<b>Total Estimado</b>	<b>2</b>





### 3.3 Factores de Complejidad

A continuación se relacionan los factores de complejidad evaluados para el Sistema de Reserva y Coordinación del Acuario Nacional.

Factor del Complejidad Técnica		
Factor	Descripción	Complejidad
Rendimiento de la aplicación.	Velocidad de procesamiento, el tiempo de respuesta, consumo de recursos, rendimiento efectivo total y eficacia de la aplicación.	Medio
Rendimiento de la Plataforma.	Velocidad de procesamiento, el tiempo de respuesta, consumo de recursos, rendimiento efectivo total y eficacia de la Plataforma.	Medio
Complejidad de Diseño Gráfico	Nivel de complejidad en el conjunto de operaciones técnicas-proyectuales necesarias para la información visual, al objeto de dotarla de la mayor cantidad posible de atributos eficaces, comprensibles y persuasivos para la fácil y completa percepción del mensaje a transmitir.	Medio Bajo
Procesamiento interno complejo.	Procesamientos lógicos o matemáticos intensivos en la aplicación	No
Incluye objetivos especiales de Seguridad.	La disponibilidad de mecanismos que controlen o protejan los programas o los datos.	No

Tabla 1. Factor del Complejidad Técnica.

Factor de ambiente		
Factor	Descripción	Complejidad
Temática conocida por la estructura de producción (Facultad, Centro, etc )	¿La estructura de producción tiene experiencia en el desarrollo de proyectos sobre esta temática o tiene personal capacitado en esta temática?	Si
La estructura de producción tiene conocimientos sobre Base de Datos.	¿La estructura de producción domina los conocimientos de Base de Datos? Las Bases de Datos son sistemas formado por un conjunto de datos almacenados en discos que permiten el acceso directo a ellos y un conjunto de programas que manipulen ese conjunto de datos.	Si
La estructura de producción tiene conocimientos sobre Leguaje de Programación.	¿La estructura de producción domina los conocimientos sobre el lenguaje de programación que especifica el cliente o con el que se pretende realizar el proyecto?	Si



La estructura de producción tiene conocimientos sobre el Control de Versiones.	¿La estructura de producción domina los conocimientos y herramientas para el control de versiones? El control de versiones es la gestión de los diversos cambios que se realizan sobre los elementos de algún producto o una configuración del mismo.	Si
La estructura de producción tiene conocimientos sobre la Gestión de Proyectos.	¿La estructura de producción tiene experiencia en el desarrollo de proyectos exitosos? La Gestión de proyectos es la disciplina de organizar y administrar recursos de manera tal que se pueda culminar todo el trabajo requerido en el proyecto dentro del alcance, el tiempo, y coste definidos.	Si
Proyecto de continuidad.	Se ha desarrollado una parte del proyecto y se pretende continuar el desarrollo.	No
Disponibilidad de la estructura de producción de líder de Proyecto.	¿La estructura de producción cuenta con personal con experiencia en la gestión de proyectos?	No
La estructura de producción cuenta con todos los desarrolladores dedicados a tiempo completo en el proyecto	¿Los desarrolladores involucrados (estudiantes y profesores) solo están dedicados a la producción a tiempo completo?	Si

**Tabla 2. Factor de ambiente.**

<b>Factor de cliente</b>		
<b>Factor</b>	<b>Descripción</b>	<b>Complejidad</b>
Existe sistema anterior.	¿El proyecto se ha automatizado en otro momento?	No
Existen resultados de informatización.	¿Existen algunas funcionalidades o paquetes funcionales informatizados?	No
Existe Dirección de Informatización o Informática.	¿Se cuenta en la organización con personal calificado en los conocimientos informáticos y/o con una Dirección o grupo dedicado a la Informatización?	Si
Existe infraestructura tecnológica en la organización.	La infraestructura tecnológica representa la integración de un conjunto de elementos de hardware (servidores, puestos de trabajo, redes, enlaces de telecomunicaciones, etc.), software (sistemas operativos, bases de datos, lenguajes de programación, herramientas de administración, etc.) y servicios (soporte técnico, seguros, comunicaciones, etc.) que en conjunto dan soporte a las aplicaciones (sistemas informáticos) de una empresa u organización.	Si



Existe base legal en la organización.	¿La organización cuenta con una base fundamentada jurídicamente y apoyada en la ley?	Si
Es el primer proyecto con la organización.	¿Es la primera vez que se realiza un proyecto con la organización?	Si
Existe una estructura clara en la organización.	¿Los procesos y áreas funcionales están claramente definidos?	Si
Existe un especialista para atender al proyecto.	¿En la organización se cuenta con un especialista en disposición de atender el proyecto?	Si
Están definidas las funciones de las áreas.	¿Las actividades que realizan en cada área están definidas?	No
Estabilidad de los requisitos.	¿Los requisitos funcionales y no funcionales se mantendrán estables y sin posibilidad de cambios?	No
El cliente es el usuario de la aplicación.	¿El cliente que solicita la realización del proyecto es el usuario que interactuará con la aplicación?	Si

**Tabla 3. Factor de cliente.**

Factor del valor agregado		
Factor	Descripción	Complejidad
Ayuda Integrada.	La aplicación debe contar con una ayuda integrada al sistema.	No
Mantener una interfaz familiar al negocio.	El cliente de la aplicación solicita que la interfaz o el diseño gráfico del sistema cuenten con características particulares o explícitas.	Si
Aplicación multidioma.	La aplicación debe ser desarrollada en varios idiomas.	No

**Tabla 4. Factor del valor agregado.**

### 3.4 Resultados de la estimación.

Finalmente, después de aplicar el método se obtuvo información relevante sobre el desarrollo del proyecto y en base a esto se puede determinar si es factible ayudar a la organización a lograr los objetivos y cubrir todas las metas con los recursos actuales.



Actividad	En Horas	En Semanas	En Meses	En Años
Estudio Preliminar	160.0	4.0	1.0	0.08
Modelamiento del Negocio	360.0	9.0	2.25	0.19
Requerimientos	520.0	13.0	3.25	0.27
Análisis y Diseño	520.0	13.0	3.25	0.27
Implementación	840.0	21.0	5.25	0.44
Pruebas Internas	280.0	7.0	1.75	0.15
Pruebas de Liberación	80.0	2.0	0.5	0.04
Pruebas de Aceptación	40.0	1.0	0.25	0.02
Pruebas Piloto	0.0	0.0	0.0	0.0
Despliegue	0.0	0.0	0.0	0.0
Soporte	0.0	0.0	0.0	0.0
<b>Tiempo Mínimo de Desarrollo</b>	<b>1423.6</b>	<b>35.59</b>	<b>8.9</b>	<b>0.74</b>
<b>Tiempo Máximo de Desarrollo</b>	<b>1423.6</b>	<b>35.59</b>	<b>8.9</b>	<b>0.74</b>

Costo de Software (Tarifas Horarias)	\$37053.7 USD
--------------------------------------	---------------

### 3.5 Conclusiones.

El desarrollo del sistema es rentable para la empresa ya que no es necesario grandes gastos de recursos en tecnología de punta, ni sistemas de cómputo de última generación. Se realizará además para usuarios con baja o ninguna experiencia en el uso de computadoras y posee una interfaz amigable, de fácil uso. Según los datos obtenidos a través de la estimación, se concluye que operativa, técnica y económicamente es factible desarrollar el producto y aunque el costo de producción asciende a \$ 37 053.00 USD, es más que una pérdida una oportunidad de ahorro para el país, ya que este producto se realizará de forma gratuita ahorrando de esta forma gastos al Acuario Nacional de Cuba y al país.



## **Conclusiones**

Se realizó un estudio del marco teórico de la investigación lo que posibilitó determinar que no existe un sistema que se pueda adaptar a la solicitud del cliente.

La identificación y especificación de las funcionalidades del sistema así como los requerimientos no funcionales permitieron desarrollar el análisis y diseño utilizando estereotipos web que posibilitó la implementación del Sistema de Reserva, Factura y Coordinación del Acuario Nacional de Cuba.

El estudio de costo de la propuesta presentada, teniendo en cuenta los factores de complejidad técnica, ambiente, valor agregado y cliente, evidenció el beneficio y ahorro económico que representa el desarrollo del sistema para el país.



## **Recomendaciones**

- ✓ Enriquecer el módulo de Reportes con otros tipos de estadísticas.
- ✓ Desarrollar una guía virtual educativa con el objetivo de capacitar a los trabajadores que interactuaran con el sistema.
- ✓ Desarrollar un manual de usuario interactivo para la aplicación.
- ✓ Internacionalizar el sistema.



## Referencias Bibliográficas

1. *An introduction*. **Kruchten, Philippe**. 2003, Rational Unified Process.
2. *Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP)*. **Penadés, Patricio Leteier y M. Carmen**. España: Universidad Politécnica de Valencia : s.n., 2006.
3. *El lenguaje Unificado de Modelado*. **Rumbaugh, James, Ivar, Jacobson y Grady, Booch**.
4. **Pérez, J. E.** Introducción a Ajax. [En línea] 2008. [Citado el: 3 de enero de 2011.] <http://www.librosweb.es/>.
5. —. Introducción a XHTML. [En línea] 2008. [Citado el: 21 de enero de 2012.] <http://librosweb.es>.
6. **Tidwell, Doug**. Introducción a XML. [En línea] [Citado el: 27 de enero de 2012.] <http://www.ibm.com/developerWorks>.
7. **Pérez, J. E.** Introducción al CSS. [En línea] 2009. [Citado el: 2012 de enero de 2.] <http://librosweb.es>.
8. —. Introducción a JavaScript. [En línea] 2009. [Citado el: 2012 de enero de 6.] <http://librosweb.es>.
9. **Hinostroza, R. R.** Características de PHP. [En línea] 2005. [Citado el: 2012 de febrero de 15.] <http://www.linuxcentro.net/linux/staticpages/index/php?page=CaracteristicasPHP..>
10. **Gutiérrez, J. J.** Qué es un framework web? [En línea] 2009. [Citado el: 2012 de febrero de 6.] [http://www.lsi.us.es/~javierj/investigacion\\_ficheros/Framework.pdf](http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf).
11. **SlideShare**. [En línea] [Citado el: 6 de marzo de 2012.] <http://slideshare.net/tutorialsruby/d422guiaparaeldesarrollodegadgets10-2620805>.
12. **Alvarez, Miguel Angel**. desarrolloweb.com. [En línea] 2010. [Citado el: 23 de marzo de 2012.] <http://desarrolloweb.com>.
13. **java.sun.com**. [En línea] [Citado el: 3 de abril de 2012.] <http://java.sun.com>.
14. Doctrine. [En línea] 2010. [Citado el: 5 de abril de 2012.] <http://www.doctrine>.
15. Scribd. [En línea] 2010. [Citado el: 15 de enero de 2012.] <http://es.scribd.com/doc/54285643/Mysql-vs-Postgresql>.



16. **Ltd, Netcraft.** Netcraft Ltd. [En línea] [Citado el: 15 de diciembre de 2011.] <http://news.netcraft.com/archives/2011>.
17. GNU Operating System. [En línea] [Citado el: 15 de diciembre de 2011.] <http://www.gnu.org/licences/licence-list.es.html>.
18. [En línea] [Citado el: 15 de diciembre de 2011.] 27. <http://www.abadiadigital.com/articulo/google-confirma-que-apache-es-el-doble-de-seguro-que-iis/>.
19. **Guerrero, S.** Netbeans 6.8 con soporte para PHP 5.3 y Symfony. [En línea] 2009. [Citado el: 23 de enero de 2012.] <http://www.sergioguerrero.es/php/netbeans-soporte-php53-symfony>.
20. Linperial International Systems S.A de C.V. *Comunidad Limperial de Desarrolladores*. [En línea] 2010. <http://www.linperial.com/comunities/forums/developers/?q=node/60>.
21. Sitio Web oficial Visual-Paradigm. [En línea] 2011. <http://www.visual-paradigm.com/product/vpuml>.
22. [adictosaltrabajo.com](http://www.adictosaltrabajo.com). [En línea] 22 de diciembre de 2003. [Citado el: 23 de febrero de 2012.] <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=grasp>.
23. **Rodríguez Brito, Dayami** . *5406\_Bases del Método de Estimación UCI*. La Habana, Cuba : s.n., 2012.
24. [En línea] [Citado el: 15 de diciembre de 2011.] 27. <http://www.abadiadigital.com/articulo/google-confirma-que-apache-es-el-doble-de-seguro-que-iis/>.