

Universidad de las Ciencias Informáticas

Facultad 1



Título: Sistema Control de Hoteles. Módulos Administración y Control de estancia.

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

Autor(es): Antonio Proenza Betancourt.

Lisandra Marcheco Rodríguez.

Tutor(es): Ing. Ronaldo Castro Milán.

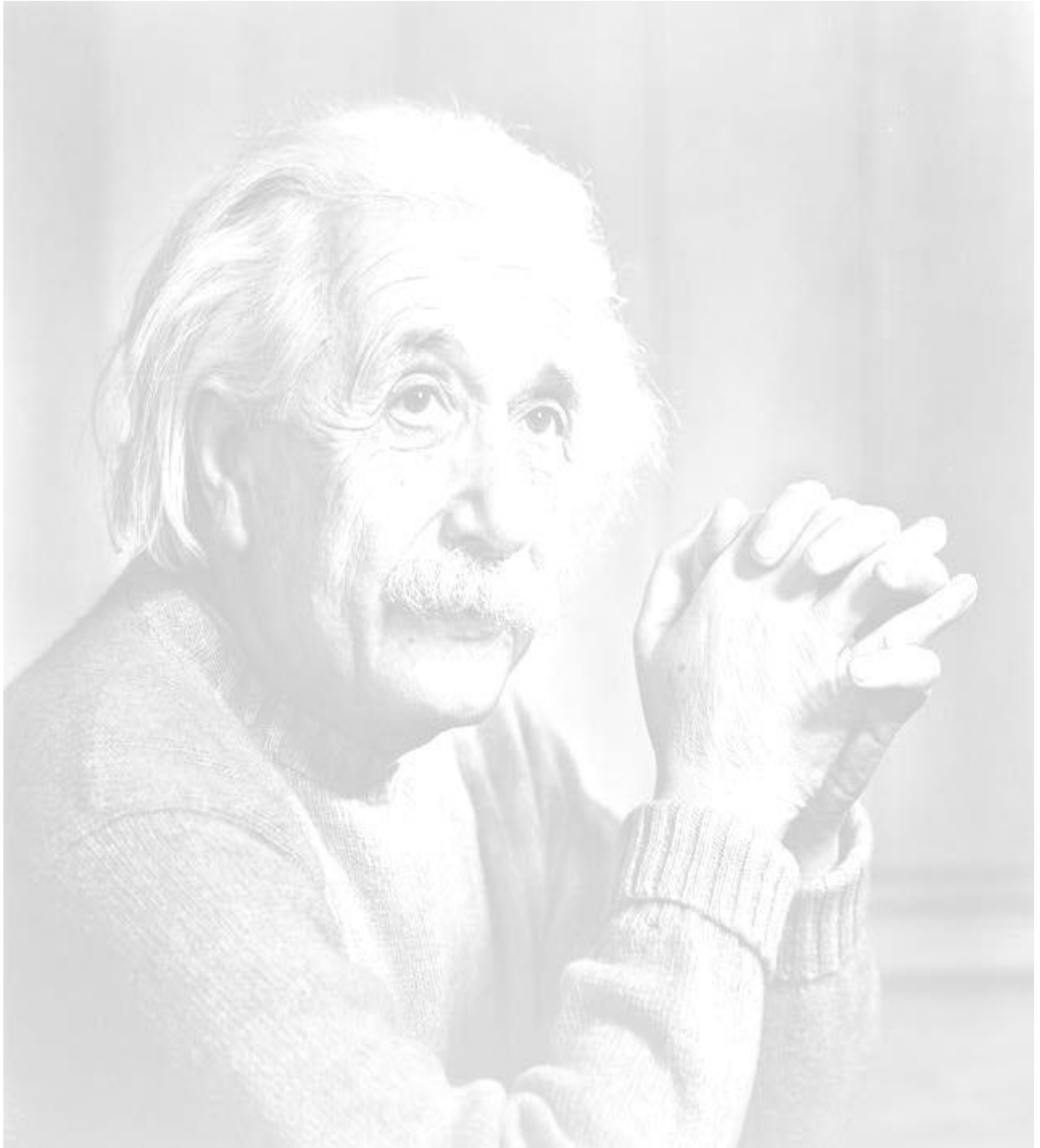
Ing. Adiry Hernández Regueiro.

Co-Tutor: Ing. Joe Luis Borrero Senciales.

La Habana

“Año 54 Aniversario del Triunfo de la Revolución”

Junio 2012



“Nunca consideres el estudio como una obligación, sino como una oportunidad para penetrar en el bello y maravilloso mundo del saber.”

Albert Einstein

Declaración de Autoría

Declaración de autoría

Declaramos que somos los únicos autores del trabajo “Sistema Control de Hoteles. Módulos Administración y Control de Estancias” y autorizamos a la Facultad 1 de la Universidad de la Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmamos la presente a los ____ días del mes _____ del año_____.

Autores:

Antonio Proenza Betancourt

Lisandra Marcheco Rodríguez

Tutores:

Ronaldo Castro Milán

Adiary Hernández Regueiro

Agradecimientos

Lisandra Marcheco Rodríguez

Agradezco a todas aquellas personas que de alguna forma han influido en mi formación como profesional, a familiares, amigos, profesores, y al tutor.

Agradezco a mi familia en general por el apoyo que me han dado, en especial a mis padres queridos que amo con todo mi corazón y que siempre tengo presente en mi mente, a mis dos hermanos, a Alexey por su comprensión y por su ayuda, gracias por servirme de ejemplo y a Robertico por compartir a mi lado todos estos años y ser mi cómplice, los quiero.

Agradezco a mis amistades, Zaida, Maidelis, Geny y en especial a Yeimy por toda tu ayuda en estos últimos años, gracias por ser quien eres y por brindarme tu amistad, te admiro por tu sinceridad y por tu empeño, nunca cambies.

Agradezco a mis suegros que son mi familia también y que siempre me han brindado mucho amor, los quiero mucho.

En especial debo agradecer a mi novio que más que mi novio es mi amigo porque siempre está a mi lado ayudándome a ser mejor persona, porque es a la vez mi padre aconsejándome y cuidándome cuando estoy lejos del mío, y por ser mi amor más grande y verdadero, gracias a ti por lo mucho que me has dado, te amo.

Antonio Proenza Betancourt

Ante todo quisiera agradecerle a todos aquellos que han hecho posible de una forma u otra mi formación como Ingeniero en Ciencias Informáticas, especialmente a mis familiares, profesores y amigos.

También quisiera agradecer a Marcel, Darién, Pepe André, Dariel, Onnier y al tocallo, entre otros compañeros y amigos que me han soportado en estos años y muchos de ellos no están presentes en la universidad.

En estos agradecimientos no cabe olvidar mis seres más queridos, mi Madre, mi papá Rubén y mis hermanos que son todo lo que tengo.

También debo agradecerles a mis suegros que han sido especiales en los últimos 3 años de mi vida.

Le agradezco a mi novia que la quiero con todo el amor de mi vida, muchas gracias por existir y sobre todo por entrar en mi vida y marcarla para siempre.

Dedicatoria

“Dedicamos este trabajo a nuestros seres más amados, nuestros padres que siempre nos han apoyado en nuestras decisiones y nos han servido de guía para cumplir nuestros sueños.”

Resumen

La presente investigación, que lleva por título “Sistema Control de Hoteles. Módulos Administración y Control de Estancias”, tiene como objetivo principal el desarrollo de una aplicación web que mejore la disponibilidad de la información referente a los huéspedes en las instalaciones de hospedaje, para el Servicio Administrativo de Identificación, Migración y Extranjería (SAIME) en la República Bolivariana de Venezuela.

El SAIME utiliza al Sistema Registro de Huéspedes Extranjeros en Establecimientos (SRHEE), para facilitar el control de los huéspedes que se encuentren alojados en las distintas instalaciones de hospedaje de la República Bolivariana de Venezuela, pero con dicho sistema la información se obtiene cada ocho días y el SAIME necesita una solución que permita obtener la información en tiempo real.

Para el desarrollo de la solución se realizó un estudio del estado del arte de los sistemas relacionados con la gestión hotelera. Para darle cumplimiento a las funcionalidades que componen los Módulos Administración y Control de Estancias, se siguieron los pasos que propone el Proceso Unificado de Desarrollo de Software, se describieron las principales herramientas y tecnologías utilizadas, y se realizó un diseño basado en patrones. La presente investigación garantizará la disponibilidad en tiempo real de la información referente a los huéspedes extranjeros, propiciando la reducción de actividades manuales y la verificación de la información manejada.

Palabras Clave: aplicación web, disponibilidad, información, instalaciones de hospedaje.

Índice de Contenido

Introducción.....	1
Capítulo I: Fundamentación Teórica	4
1.1. Introducción.....	4
1.2. Conceptos asociados al dominio del problema	4
1.3. Sistemas homólogos.....	4
1.3.1 Sistemas de Gestión Hotelera.....	4
1.4. Sistemas de Control Migratorio	7
1.5. Metodología de Desarrollo de Software.....	8
1.5.1 Proceso Unificado de desarrollo de Software (RUP).....	8
1.6. Lenguajes.....	9
1.6.1 Lenguajes de Programación.....	9
1.6.2 Lenguaje de Modelado	11
1.7. Herramientas	12
1.7.1 Modelado	12
1.7.2 Sistema Gestor de Base de Datos.....	13
1.7.3 Entorno de desarrollo integrado.....	13
1.7.4 Servidor Web.....	14
1.7.5 <i>Framework</i>	14
1.8. Conclusiones parciales.....	17
Capítulo II: Diseño de la solución	18
2.1. Introducción.....	18
2.2. Modelo de Dominio	18
2.2.1 Definición de las clases del Modelo de Dominio	19
2.3. Requisitos Funcionales.....	19
2.4. Requisitos No Funcionales.....	20
2.5. Actores del Sistema.....	21
2.6. Diagrama de Casos de Uso del Sistema.....	21
2.7. Descripción de Casos de usos del Sistema.....	22
2.8. Modelos de Clases del Análisis	38
2.9. Arquitectura de Software	41
2.9.1 Patrón de arquitectura Cliente/Servidor	41
2.9.2 Patrón de arquitectura Modelo Vista Controlador	42
2.10. Patrones de Diseño	42

Índice

2.10.1	Patrones GRASP (<i>General Responsibility Assignment Software Patterns</i>).....	42
2.11.	Modelos de Clases del Diseño.....	43
2.12.	Diagramas de interacción.....	46
2.13.	Diseño de la Base de Datos	48
2.14.	Conclusiones parciales	48
Capítulo III: Implementación y Prueba		49
3.1.	Introducción.....	49
3.2.	Implementación.....	49
3.3.	Diagrama de Despliegue	49
3.4.	Diagrama de Componentes	50
3.5.	Estándares de Codificación	50
3.5.1	Convención de nomenclatura	51
3.5.2	Estructuras de control.....	52
3.6.	Seguridad.....	54
3.7.	Proceso de pruebas.....	55
3.8.	Resultado de las pruebas	69
3.9.	Conclusiones parciales.....	69
Conclusiones.....		70
Recomendaciones.....		71
Referencias bibliográficas.....		72
Glosario de Términos		74
Anexos		75
Anexo 1 Descripciones de Casos de Uso.....		75
Anexo 2 Modelos de Clases del Análisis.....		78
Anexo 3 Modelos de Clases del Diseño.....		80
Anexo 4 Diagramas de interacción.....		84
Anexo 5 Diseño de casos de prueba.....		86

Índice de Figuras

<i>Figura 1 Diagrama de Clases del Modelo de Dominio</i>	<i>18</i>
<i>Figura 2 Diagrama de Casos de Uso del Sistema.....</i>	<i>22</i>
<i>Figura 3 Diagrama de Clases del Análisis CU Registrar Huésped.....</i>	<i>39</i>
<i>Figura 4 Diagrama de Clases del Análisis CU Emitir Reportes.....</i>	<i>39</i>
<i>Figura 5 Diagrama de Clases del Análisis CU Buscar Persona.....</i>	<i>40</i>
<i>Figura 6 Diagrama de Clases del Análisis CU Actualizar Registro de Huésped</i>	<i>40</i>
<i>Figura 7 Diagrama de Clases del Análisis CU Finalizar Estancia.....</i>	<i>40</i>
<i>Figura 8 Diagrama de Clases del Diseño CU Registrar Huésped</i>	<i>44</i>
<i>Figura 9 Diagrama de Clases del Diseño CU Mostrar Reportes.....</i>	<i>45</i>
<i>Figura 10 Diagrama de Clases del Diseño CU Actualizar Registro.....</i>	<i>46</i>
<i>Figura 11 Diagrama de Secuencia CU Actualizar Registro</i>	<i>47</i>
<i>Figura 12 Diagrama de Secuencia CU Finalizar Estancia</i>	<i>47</i>
<i>Figura 13 Modelo Físico de la Base de Datos</i>	<i>48</i>
<i>Figura 14 Diagrama de Despliegue</i>	<i>49</i>
<i>Figura 15 Diagrama de Componentes</i>	<i>50</i>
<i>Figura 16 Estándar de codificación: Convención de nomenclatura de variables</i>	<i>51</i>
<i>Figura 17 Estándar de codificación: Convención de nomenclatura de constantes.....</i>	<i>51</i>
<i>Figura 18 Estándar de codificación: Convención de nomenclatura de clases.....</i>	<i>51</i>
<i>Figura 19 Estándar de codificación: Convención de nomenclatura de funciones</i>	<i>52</i>
<i>Figura 20 Estándar de codificación: Estructuras de control.....</i>	<i>53</i>
<i>Figura 21 Estándar de codificación: Estructuras de control.....</i>	<i>53</i>
<i>Figura 22 Estándar de codificación: Estructuras de control.....</i>	<i>54</i>
<i>Figura 23 Estándar de codificación: Estructuras de control.....</i>	<i>54</i>
<i>Figura 24 Pruebas del Sistema</i>	<i>69</i>
<i>Figura 25 Diagrama de Clases del Análisis CU Emitir Reportes</i>	<i>78</i>
<i>Figura 26 Diagrama de Clases del Análisis CU Gestionar Usuario.....</i>	<i>79</i>
<i>Figura 27 Diagrama de Clases del Análisis CU Autenticar Usuario</i>	<i>79</i>
<i>Figura 28 Diagrama de Clases del Diseño CU Buscar Persona.....</i>	<i>80</i>
<i>Figura 29 Diagrama de Clases del Diseño CU Finalizar Estancia.....</i>	<i>81</i>
<i>Figura 30 Diagrama de Clases del Diseño CU Registrar Acompañante</i>	<i>82</i>
<i>Figura 31 Diagrama de Clases del Diseño CU Autenticar Usuario.....</i>	<i>83</i>
<i>Figura 32 Diagrama de Clases del Diseño CU Gestionar Usuario.....</i>	<i>83</i>
<i>Figura 33 Diagrama de Secuencia CU Buscar Persona</i>	<i>84</i>
<i>Figura 34 Diagrama de Secuencia CU Mostrar Reportes.....</i>	<i>85</i>
<i>Figura 35 Diagrama de Secuencia CU Registrar Acompañante.....</i>	<i>85</i>
<i>Figura 36 Diagrama de Secuencia CU Registrar Huésped.....</i>	<i>86</i>

Índice de tablas

<i>Tabla 1 Comparación entre Frameworks.....</i>	<i>15</i>
<i>Tabla 2 Descripción de Actores del Sistema.....</i>	<i>21</i>
<i>Tabla 3 Descripción CU Registrar Huésped.....</i>	<i>26</i>
<i>Tabla 4 Descripción CU Buscar Persona.....</i>	<i>28</i>
<i>Tabla 5 Descripción CU Registrar Acompañante</i>	<i>29</i>
<i>Tabla 6 Descripción CU Actualizar Registro de Huéspedes.....</i>	<i>32</i>
<i>Tabla 7 Descripción CU Finalizar Estancia</i>	<i>34</i>
<i>Tabla 8 Descripción CU Gestionar Usuario.....</i>	<i>38</i>
<i>Tabla 9 Diseño de Casos de Prueba basados en requisitos: Actualizar Registro.....</i>	<i>62</i>
<i>Tabla 10 Diseño de Casos de Prueba basados en requisitos: Finalizar Estancia</i>	<i>64</i>
<i>Tabla 11 Diseño de Casos de Prueba basados en requisitos: Adicionar Usuario</i>	<i>68</i>
<i>Tabla 12 Descripción CU Emitir Reportes</i>	<i>76</i>
<i>Tabla 13 Descripción CU Autenticar usuario.....</i>	<i>78</i>
<i>Tabla 14 Diseño de Casos de Prueba basado en requisitos: Eliminar Usuario.....</i>	<i>87</i>
<i>Tabla 15 Diseño de Casos de Prueba basado en requisitos: Registrar huésped</i>	<i>93</i>
<i>Tabla 16 Diseño de Casos de Prueba basado en requisitos: Buscar persona.....</i>	<i>96</i>
<i>Tabla 17 Diseño de Casos de Prueba basado en requisitos: Registrar Acompañante.....</i>	<i>100</i>
<i>Tabla 18 Diseño de Casos de Prueba basado en requisitos: Listar huésped por fecha</i>	<i>102</i>
<i>Tabla 19 Diseño de Casos de Prueba basado en requisitos: Listar huésped por período</i>	<i>105</i>
<i>Tabla 20 Diseño de Casos de Prueba basado en requisitos: Autenticar usuario.....</i>	<i>106</i>
<i>Tabla 21 Diseño de Casos de Prueba basado en requisitos: Modificar usuario</i>	<i>112</i>

Introducción

El Servicio Administrativo de Identificación, Migración y Extranjería (SAIME), es un organismo dependiente del Estado Venezolano y adscrito al Ministerio del Poder Popular para Relaciones Interiores y de Justicia, encargado de la identificación y registro de los ciudadanos en Venezuela, así como de la emisión del pasaporte venezolano y las solicitudes formadas por ciudadanos extranjeros.

El SAIME ha trazado metas de trabajo para poder posicionarse como una institución de marcada referencia nacional por la excelencia del servicio que presta, con todos los procesos automatizados y un alto grado de profesionalismo y eficiencia, donde los trámites de identificación, migración y control de extranjeros se hacen de una manera ágil dentro del marco de la Ley.

El área de Migración es la encargada de establecer las normas y procedimientos del control migratorio de los ciudadanos venezolanos y extranjeros, que ingresan o egresan a través de los diferentes puntos de control a nivel nacional, además le corresponde llevar el control de los huéspedes extranjeros que se hospedan en las distintas instalaciones de Venezuela, las cuales pueden ser: pensiones, posadas y hoteles de cualquier categoría.

Para ello utiliza el SRHEE¹, desarrollado en Cuba por la Universidad de Ciencias Informáticas en la Fase II del Proyecto Identidad. Este sistema gestiona los procesos que se llevan a cabo en las instalaciones hoteleras y ofrece la posibilidad de conocer oportunamente la condición legal de los ciudadanos que acceden al territorio nacional.

Al realizar un análisis del funcionamiento de dicho sistema se identificó la siguiente **situación problemática**: el sistema desarrollado es una aplicación de escritorio, por tanto resulta muy difícil desplegarla por todas las instalaciones hoteleras; los establecimientos que utilizan esta aplicación necesitan enviar al SAIME cada ocho días la información recopilada de los huéspedes, por lo que no se obtiene en tiempo real, esta aplicación además cuenta con una base de datos local y en la mayoría de los hoteles existe más de una PC en la recepción, desde donde se hace el registro de huéspedes, por tanto se envían al SAIME tantos ficheros XML con la información como PC de registro se tiene en el hotel.

¹ Sistema Registro de Huéspedes Extranjeros en Establecimientos.

Atendiendo a esta problemática el presente trabajo de diploma se plantea como **problema a resolver**: ¿Cómo mejorar la disponibilidad de la información referente al proceso de control de huéspedes extranjeros en instalaciones de hospedaje, para el SAIME en la República Bolivariana de Venezuela?

Se define como **objeto de estudio**: Proceso de control de huéspedes extranjeros en instalaciones de hospedaje.

Se plantea como **objetivo general**: Desarrollar un sistema web a partir del SRHEE que permita la disponibilidad de la información necesaria de los huéspedes, en las instalaciones de hospedaje de la República Bolivariana de Venezuela.

El **campo de acción** enmarca: El registro de huéspedes extranjeros en instalaciones de hospedaje en la República Bolivariana de Venezuela.

Para el desarrollo de esta investigación se propone la siguiente **idea a defender**: la realización de los Módulos Administración y Control de Estancias para el Sistema Control de Hoteles en la República Bolivariana de Venezuela, facilitará la disponibilidad de la información referente a los huéspedes en las instalaciones de hospedaje.

El cumplimiento efectivo del objetivo trazado dependerá de las siguientes **tareas de la investigación**:

- Análisis del estado del arte de sistemas relacionados con la gestión hotelera a nivel mundial.
- Análisis bibliográfico que permita fundamentar las herramientas informáticas utilizadas para el desarrollo de la aplicación web.
- Análisis de la documentación asociada al Sistema Registro de Huéspedes.
- Identificación de los requisitos funcionales y no funcionales del sistema.
- Diseño de la interfaz de usuario de la aplicación web.
- Desarrollo del Módulo Administración para realizar la gestión de los usuarios y roles para el Sistema Control de hoteles.
- Desarrollo del Módulo Control de Estancia para realizar los procesos asociados al registro y control de los huéspedes.
- Documentación detallada de todo el proceso de desarrollo para que sirva como un elemento de consulta.
- Realización de pruebas asociadas al proceso de desarrollo.

- Evaluación de los resultados obtenidos a partir de las pruebas realizadas. Al finalizar como **posibles resultados** se obtendrán:
- Documentación de todo el proceso de desarrollo del sistema.
- Desarrollo de los Módulos Administración y Control de Estancias para el Sistema Control de Hoteles, que permitan el registro y la verificación de la información referente a los huéspedes en instalaciones de hospedaje de la República Bolivariana de Venezuela.

En el transcurso de la investigación se utilizarán los siguientes **métodos teóricos**.

Métodos Científicos:

Histórico – Lógico: permitirá realizar un estudio del estado del arte, haciendo un análisis de los fenómenos relacionados con los sistemas informáticos implementados a nivel mundial y los que se han desarrollado en la República Bolivariana de Venezuela asociados al control migratorio y la gestión hotelera, describiendo las ventajas y desventajas de los mismos.

Analítico – Sintético: a partir de la bibliografía consultada servirá para organizarla y sintetizarla, y de este modo seleccionar la información necesaria respecto a las herramientas y metodología de desarrollo a utilizar en la solución del Sistema Control de Hoteles en la República Bolivariana de Venezuela.

La presente investigación cuenta con tres **Capítulos**:

Capítulo I: Fundamentación Teórica. Se realiza un análisis de las tecnologías actuales relacionadas con el control migratorio y la gestión hotelera y se abordarán las características, ventajas y desventajas de las herramientas y tecnologías escogidas para el desarrollo de la solución.

Capítulo II: Diseño de la solución. Se analizan los elementos fundamentales a tener en cuenta en el desarrollo del producto, realizando una valoración crítica del diseño para los módulos. Definiendo los patrones de diseño y la arquitectura, facilitando el entendimiento de la solución a través de las descripciones y diagramas realizados.

Capítulo III: Implementación y Prueba. Se realizara el diagrama de despliegue y el de componentes, definiendo las pautas usadas para la implementación del sistema y las principales funcionalidades de la aplicación. Se realizara la validación de la solución propuesta aplicando las pruebas necesarias para demostrar el correcto funcionamiento del sistema a partir de las métricas especificadas.

Capítulo I: Fundamentación Teórica

1.1. Introducción

En el presente capítulo se abordará sobre los sistemas informáticos que han sido desarrollados a nivel mundial especializados en la gestión hotelera, se describirán las características de la metodología de desarrollo y de las herramientas que serán utilizadas. Se seleccionarán los lenguajes de programación y el *framework* para el desarrollo del sistema propuesto.

1.2. Conceptos asociados al dominio del problema

Control Migratorio

Hace referencia a la capacidad del Estado de verificar, vigilar, supervisar o autorizar la entrada, permanencia y salida de nacionales o extranjeros de un estado, y de regular las consecuencias de su traslado, estableciendo para ello reglas de funcionamiento (Hidalgo, 2011).

Control de estancias

Proceso para controlar todos los extranjeros que se encuentran hospedados en los hoteles y establecimientos de Venezuela.

1.3. Sistemas homólogos

El Sistema Control de Hoteles se encargará del registro de huéspedes extranjeros en las diferentes instalaciones de hospedaje. Por tanto se realizó un análisis de sistemas relacionados con la gestión hotelera, con el fin de fortalecer el conocimiento de los procesos que desarrollan.

La mayoría de estos sistemas han sido desarrollados bajo las necesidades específicas de los organismos de administración de un país determinado, por lo que son soluciones privadas y con muy poca documentación, que no se adecuan al negocio existente hoy en las instalaciones de hospedaje de la República Bolivariana de Venezuela.

1.3.1 Sistemas de Gestión Hotelera

Sistemas orientados a la administración de datos de una empresa que proporciona servicios de hospedaje, restaurante y servicios o productos varios. Estos programas, intercomunicados entre sí, cubren el proceso de hospedaje que van desde la reservación, pasando por la recepción, el registro de consumos dentro del hotel y hasta la emisión de un estado de cuenta y facturación (Figuroa, 2006).

1.3.1.1 Internacionales

AM/Hotel

Es un programa informático desarrollado por la empresa AMSystem² dirigido a la completa gestión administrativa del hotel. El programa ha sido desarrollado manteniendo una interfaz de usuario bastante sencilla e intuitiva para su manejo (AMSystem, 2010).

Permite dividir la actividad hotelera por años y hoteles. Las características de AM/HOTEL son:

- Control de habitación: El programa brinda detalles de la gestión y uso de cada una de las habitaciones declaradas.
- Control de reservas y recepción: Son los principales procesos de un hotel y los más cuidados, ya que del buen funcionamiento y registro de estos procesos dependen el resto de las operaciones.
- Balance del día: Permite sacar un pequeño pero detallado balance de la situación del día.
- Gestión y control de usuarios o grupos de usuarios a través de perfiles.

ActualHotel

Una herramienta llena de características destinadas a la gestión hotelera, procesos intuitivos y potentes, control absoluto del egreso de extranjeros, reservas, manejo de extras, bar, restaurante, telefonía, personal, existencias de mercadería, proveedores y compras. Controla las distintas cajas ubicadas en las dependencias del hotel, emite informes de facturación y estadísticas.

Módulos disponibles:

- Módulo recepción
- Módulo administración
- Módulo reservación
- Módulo proveedores
- Módulo restaurante

Sistema de Control Hotelero (SCH) del Departamento Administrativos de Seguridad (DAS)

El Departamento Administrativo de Seguridad (DAS) en Colombia puso en marcha un avanzado Sistema de Control Hotelero que le permitirá a la entidad contar con información en tiempo real sobre la ubicación y presencia de ciudadanos extranjeros en hoteles, pensiones, hostales, residencias, fincas, casas y apartamentos en todo el

² Empresa española productora de software.

Capítulo I: Fundamentación Teórica

territorio nacional. El objetivo del Sistema de Control Hotelero (SCH) es tener una herramienta de Control Migratorio que va a permitir confrontar la información que entregan los extranjeros cuando se hospedan en un hotel del país con la base de datos de la entidad. Así mismo, hacer un control más efectivo sobre aquellos establecimientos que hospedan extranjeros y no lo reportan.

El convenio de cooperación entre las entidades, DAS-Cotelco³, busca el intercambio de información que permita proyectar el crecimiento del sector, atender la demanda, mejorar los servicios y, simultáneamente, realizar el control migratorio. La idea es trabajar conjuntamente contra delitos como el abuso de menores, el tráfico de migrantes, la trata de personas y otros tipos de delitos de carácter transnacional, que están afectando a un número importante de colombianos (Diario Portafolio, 2011).

1.3.1.2 Nacionales

GesHotel

Sistema realizado en Cuba por técnicos del antiguo Instituto Nacional de Turismo (INTUR) utilizado en la carpeta de algunos hoteles. Entre las características principales de GesHotel destacan que todo el entorno es muy intuitivo, pudiendo gestionarse casi todo desde sus gráficos. Las operaciones que puede realizar son las siguientes: situación de la reserva de una habitación (si está reservada, ocupada o libre), control de todas las reservas solicitadas, agenda empresarial para llevar los datos de los clientes, rápido cambio automático de ubicaciones por cambio de fechas, gestión de anulaciones, modificaciones, envío por email o por SMS para confirmar la reserva del cliente, entre otras (Febles, 2005).

InterHotel

Solución desarrollada entre Cuba y España que se instaló prácticamente en todos los hoteles administrados por las cadenas nacionales en la década de los 90. Es un sistema diseñado expresamente para automatizar las operaciones de Gestión Hotelera de manera integral, teniendo en cuenta los intereses de los clientes. El sistema se adapta a las características de cualquier instalación turística, independientemente de sus dimensiones o complejidad (Febles, 2005).

1.3.1.3 Universidad de las Ciencias Informáticas

Sistema Registro de Huéspedes Extranjeros en Establecimientos

En la Universidad de las Ciencias Informáticas específicamente en el Centro de Identificación y Seguridad Digital se encuentra el proyecto Identidad. En la Fase II de dicho proyecto se realizó el SRHEE para la República Bolivariana de Venezuela. Este

³ Asociación Hotelera y Turística de Colombia.

Capítulo I: Fundamentación Teórica

es un sistema diseñado para cada hotel y es una aplicación de escritorio desarrollada en el lenguaje C# con una base de datos local en SQLite.

Esta aplicación permite registrar huéspedes nacionales y extranjeros con sus acompañantes y posteriormente genera un fichero encriptado con esta información que se importa al SAIME a través del Sistema Control de Hoteles en Oficinas. Este sistema emite reportes sobre toda la información registrada en el establecimiento, pero no permite que el SAIME obtenga la información en tiempo real sino cada ocho días cuando se envían los reportes.

Huésped casero

Sistema utilizado por las personas cubanas que arriendan sus viviendas a cubanos o extranjeros, pero desde hace algún tiempo se está elaborando en la Universidad de Ciencias Informáticas una aplicación que sustituye los sistemas obsoletos que actualmente utilizan los inspectores a cargo del tema, el sistema está enfocado para que los "caseros" desde sus viviendas reporten sus arrendamientos, pero está en desarrollo aún.

1.4. Sistemas de Control Migratorio

El Sistema Control de Hoteles además de realizar el registro de los huéspedes extranjeros facilitará al SAIME el control migratorio que realiza de estos, para comprender los procesos que realizan los sistemas de control migratorio se analizaron algunas soluciones existentes.

Sistema Integrado de Captura Migratoria (S.I.Ca.M)

Es una aplicación desarrollada por Técnicos Argentinos en la Dirección de Sistemas de la Dirección Nacional de Migraciones. El mismo está orientado a satisfacer las necesidades en materia de gestión y control de los puestos migratorios que atienden tránsitos aéreos, navales y terrestres.

Dicho sistema fue implementado en modalidad piloto en el área metropolitana y en aeropuertos internacionales del interior del país, y debido a la positiva evaluación efectuada se adoptó para el control y registro de los tránsitos migratorios, incluyendo patrones biométricos en la medida de la disponibilidad de dicha tecnología. Este sistema ha merecido el reconocimiento de diversos países por la efectividad y celeridad en el control migratorio para quienes ingresan o egresan del país, cuenta con un banco de datos integrado a los archivos de la policía nacional e internacional.

Capítulo I: Fundamentación Teórica

Sus funcionalidades básicas comprenden:

Chequeo de restricciones migratorias: antes de realizarse el tránsito, el S.I.Ca.M verifica que la persona no esté incluida en la lista de control, la cual está conformada por todos los pedidos de captura y prohibiciones que realizan Juzgados y Organismos competentes o alertar a los funcionarios sobre procedimientos especiales a tener en cuenta en el registro.

Registro del tránsito: toda persona que se somete al Control Migratorio ya sea en un paso aéreo, terrestre o marítimo, queda registrada en la base informática del S.I.Ca.M con sus datos personales y los propios del tránsito: fecha de cruce, nacionalidad, nombre del paso de cruce y datos del inspector que registra (DNM, 2011).

Sistema de Información y Control Migratorio de Guatemala (S.I.Co.M.Gua)

Sistema electrónico de registro de entradas y salidas migratorias en el país. Se trata de un sistema moderno con dispositivos de lectura electrónica de pasaportes. Con capacidad de identificar la información digital que por diversos sistemas gráficos, se incorpora en las nuevas libretas de pasaportes de la mayoría de países del mundo. El sistema está llamado a ser un sistema nacional completo, moderno y ágil para el registro y control migratorios, lo que facilita considerablemente los trámites individuales. Tiene particular importancia para el desarrollo de la industria turística, uno de los sectores estratégicos del crecimiento económico y cultural de Guatemala (DNM, 2003).

1.5. Metodología de Desarrollo de Software

Las metodologías surgen ante la necesidad de utilizar una serie de procedimientos, técnicas, herramientas y soporte documental a la hora de desarrollar un software. Se clasifican en dos tipos, las metodologías ágiles y las metodologías pesadas. Las más utilizadas para el desarrollo de software son: La Programación Extrema (XP), Scrum⁴ y el Proceso Unificado de Software (RUP).

Se seleccionó a RUP como metodología de desarrollo siguiendo la línea de trabajo anterior, pues El Sistema Control de Hoteles será desarrollado sobre la base del SRHEE, y el proceso de desarrollo de dicho sistema estuvo guiado por RUP.

1.5.1 Proceso Unificado de desarrollo de Software (RUP)

Rational Unified Process (RUP), es un proceso de amplio marco que ofrece buenas prácticas para el software y los sistemas de entrega, la aplicación eficaz y la gestión

⁴ Marco de trabajo para la gestión y desarrollo de software.

Capítulo I: Fundamentación Teórica

de proyectos. Constituye una de las metodologías estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

RUP es un proceso de desarrollo de software pesado, se basa en casos de uso para describir lo que se espera del software y está muy orientado a la arquitectura del sistema, documentándose lo mejor posible, basándose en UML (Unified Modeling Language) como herramienta principal. Se caracteriza por ser guiado por casos de uso, centrado en arquitectura, iterativo e incremental, desarrollo basado en componentes, utilización de un único lenguaje de modelado y un proceso integrador.

Se divide en cuatro fases y nueve flujos de trabajo para la mejor organización del software, cada fase se desarrolla mediante iteraciones que generan artefactos logrando ser una de las metodologías más importantes para establecer un acuerdo entre desarrolladores y clientes, está diseñada para proyectos de gran envergadura, esta metodología genera gran cantidad de documentación permitiendo un mejor entendimiento del proyecto (Ivar Jacobson, 2000).

1.6. Lenguajes

Teniendo en cuenta que la solución es una aplicación web, a continuación se detallan las características principales de los lenguajes utilizados en el desarrollo.

1.6.1 Lenguajes de Programación

Un lenguaje de programación es un idioma artificial diseñado para expresar procesos que pueden ser llevados a cabo por máquinas como las computadoras. Pueden usarse para crear programas que controlen el comportamiento físico y lógico de una máquina, para expresar algoritmos con precisión, o como modo de comunicación humana. Está formado por un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones (Rojas, 2007).

1.6.1.1 *Hypertext Preprocessor (PHP)*

El lenguaje de programación utilizado por el lado del servidor será PHP, es un lenguaje de programación de estilo clásico, con variables, sentencias condicionales, bucles y funciones, no es un lenguaje de marcas como podría ser HTML o XML. Pero a diferencia de Java o Java Script que se ejecutan en el navegador, PHP se ejecuta en el servidor, por eso permite acceder a los recursos que tenga el servidor.

El objetivo del lenguaje PHP es desarrollar sitios dinámicos en forma sencilla y rápida, Una de sus características más potentes es su soporte para gran cantidad de bases

Capítulo I: Fundamentación Teórica

de datos. Entre su soporte pueden mencionarse MySQL, Oracle, PostgreSQL, entre otras. PHP también ofrece la integración con las varias bibliotecas externas, que permiten que el desarrollador haga casi cualquier cosa desde generar documentos en Pdf hasta analizar código XML (Monso, 2005).

Ventajas

- Es un lenguaje multiplataforma.
- Orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos.
- El código fuente escrito en PHP es invisible al navegador web y al cliente ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador. Esto hace que la programación en PHP sea segura y confiable.
- Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL.
- Posee una amplia documentación.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite aplicar técnicas de programación orientada a objetos.

1.6.1.2 JavaScript

JavaScript será utilizado como lenguaje de programación por el lado del cliente para la creación de las vistas mostradas al usuario, y su principal función estará encaminada a las validaciones dentro de las páginas HTML. Es multiplataforma, orientado a eventos con manejo de objetos, cuyo código se incluye directamente en el mismo documento HTML. Este lenguaje es muy semejante a otros como C#, Java o PHP, tanto en su formato como en su sintaxis, aunque tiene sus propias características (Brandendaugh, 2000).

- Es simple, no hace falta tener conocimientos de programación para poder hacer un programa en JavaScript.
- Es dinámico, responde a eventos en tiempo real. Eventos como presionar un botón, pasar el puntero del mouse sobre un determinado texto o el simple hecho de cargar la página o caducar un tiempo.

1.6.1.3 *Hypertext Markup Language (HTML)*

HTML⁵ será utilizado como lenguaje de programación por el lado del cliente, es un lenguaje muy sencillo que permite describir documentos hipertextos, se basa en especificar en el texto la estructura lógica del contenido (títulos, párrafos de texto normal, enumeraciones, definiciones, citas, entre otros), así como los diferentes efectos que se quieren dar. Se basa en el uso de etiquetas, que podrán incluir una serie de atributos o parámetros, en su mayoría opcionales. (Kemedey, 2004)

1.6.2 Lenguaje de Modelado

Los lenguajes de modelado son un conjunto estandarizado de símbolos y sus relaciones que permiten expresar un Sistema de Información mediante un esquema teórico, el cual representará su diseño (Herranz, 2010).

1.6.2.1 *Unified Modeling Language (UML)*

Para el proceso de modelado se utilizará Unified Modeling Language (UML), es un lenguaje para especificar, construir, visualizar y documentar los artefactos de un sistema de software orientado a objetos (OO). Prescribe un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos, y describe la semántica esencial de lo que estos diagramas y símbolos significan. UML se puede usar para modelar distintos tipos de sistemas: sistemas de software, sistemas de *hardware* y organizaciones del mundo real. UML ofrece ocho diagramas en los cuales modelar sistemas (Ivar Jacobson, 2000).

Estáticos:

- Diagramas de Casos de Uso para modelar los procesos del negocio.
- Diagramas de Clases para modelar la estructura estática de las clases en el sistema.
- Diagramas de Objetos para modelar la estructura estática de los objetos en el sistema.
- Diagramas de Componentes para modelar componentes.
- Diagrama de Despliegue.

Dinámicos:

- Diagramas de Estado para modelar el comportamiento de los objetos en el sistema.

⁵ Hypertext Markup Language

- Diagramas de Actividad para modelar el comportamiento de los Casos de Uso, objetos u operaciones.
- Diagrama de Interacción.

Hoy en día, UML está consolidado como el lenguaje estándar en el análisis y diseño de sistemas de cómputo. Mediante este es posible establecer la serie de requerimientos y estructuras necesarias para plasmar un sistema de software previo al proceso intensivo de escribir código.

1.7. Herramientas

Para el desarrollo de la solución será necesaria la utilización de varias herramientas, a continuación se muestran las que serán utilizadas.

1.7.1 Modelado

Herramientas que propician un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación. Permiten la reutilización del software, portabilidad y estandarización de la documentación, además del uso de las distintas metodologías propias de la Ingeniería del Software (Pressman, 2002).

1.7.1.1 Visual Paradigm como Herramienta CASE

Para la creación de los diagramas UML se utilizará como herramienta de modelado a Visual Paradigm, esta herramienta soporta aplicaciones web, permite generar imágenes y reportes de muy buena calidad, además es muy fácil de instalar (Fuster, 2006).

Características:

- Producto de calidad.
- Soporta aplicaciones web.
- Las imágenes y reportes generados, son de muy buena calidad.
- Varios idiomas.
- Generación de código para Java y exportación como HTML.
- Fácil de instalar y actualizar.
- Compatibilidad entre ediciones.

Ofrece:

- Entorno de creación de diagramas para UML.
- Diseño centrado en casos de uso y enfocado al negocio que generan un software de mayor calidad.
- Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- Capacidades de ingeniería directa (versión profesional) e inversa.
- Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.
- Disponibilidad de múltiples versiones, para cada necesidad.
- Disponibilidad de integrarse en los principales IDEs⁶.
- Disponibilidad en múltiples plataformas.

1.7.2 Sistema Gestor de Base de Datos

Un Sistema Gestor de Base de Datos es un sistema de software que permite la definición de bases de datos así como la elección de las estructuras de datos necesarios para el almacenamiento y búsqueda de los datos, ya sea de forma interactiva o a través de un lenguaje de programación (Álvarez, 2007).

1.7.2.1 Oracle

El Sistema Control de Hoteles utilizará la Base de Datos del Proyecto Identidad, esta fue desarrollada con el Sistema Gestor de Base de Datos Oracle. Oracle es una herramienta cliente/servidor para la gestión de Bases de Datos con soporte de transacciones, estabilidad, y multiplataforma. Se considera a Oracle como uno de los sistemas de bases de datos más completos, destacando:

- Soporte de transacciones, estabilidad, escalabilidad y Soporte multiplataforma.
- Ha sido diseñado para que las organizaciones puedan controlar y gestionar grandes volúmenes de contenidos.

1.7.3 Entorno de desarrollo integrado

Un Entorno de Desarrollo Integrado es un programa informático compuesto por un conjunto de herramientas de programación. Puede dedicarse en exclusiva a un sólo lenguaje de programación o bien, poder utilizarse para varios. Consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (Altamirano, 2008).

⁶ Entorno de desarrollo integrado, llamado también IDE (sigla en inglés de Integrated Development Environment).

1.7.3.1 Netbeans

Como entorno de desarrollo para la realización de la solución se utilizará Netbeans. Netbeans es un entorno de desarrollo integrado que permite la integración de todas las herramientas requeridas para poder desarrollar y se basa en la creación de aplicaciones a partir de un conjunto de componentes de software llamados módulos.

1.7.4 Servidor Web

Un servidor web es un programa que se ejecuta continuamente en un computador, manteniéndose a la espera de peticiones de ejecución que le hará un cliente o un usuario de Internet. Se encarga de contestar a estas peticiones de forma adecuada, entregando como resultado una página web o información de todo tipo de acuerdo a los comandos solicitados.

1.7.4.1 Apache

Como servidor Web se utilizará Apache pues corre en varios Sistemas Operativos, lo que lo hace prácticamente universal. Está diseñado para ser un Servidor Web potente y flexible que pueda funcionar en la más amplia variedad de plataformas y entornos. Es una tecnología gratuita de código abierto, altamente configurable de diseño modular, que permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor.

Ventajas:

- Modular.
- Código abierto.
- Multiplataforma.
- Extensible.
- Popular.

1.7.5 Framework

La evolución en el desarrollo de aplicaciones web para múltiples usos, utilizando a PHP como lenguaje de programación y teniendo como objetivo mejorar la calidad de los productos que se obtengan, ha propiciado el surgimiento de los *frameworks*, un *framework* es una estructura de soporte definida mediante la cual otro proyecto de software puede ser organizado y desarrollado (Chávez, 2009).

Para conocer las principales características funcionales de los *frameworks* más utilizados a nivel mundial se realizó un análisis estableciendo parámetros comparativos entre ellos, con el objetivo de determinar cuál utilizar en el desarrollo de la solución.

Características	Cake PHP	Symfony	Zend
Arquitectura de aplicaciones			
Incorporación del patrón Modelo Vista Controlador orientado a objetos.	X	X	X
Mapeado de objetos a bases de datos relacionales (ORM).	X	X	
Independiente del manejador de base de datos.	X	X	X
Implementación de código HTML			
Uso de plantillas en PHP.	X	X	X
Usabilidad y acceso rápido			
Almacenamiento en caché de configuración de las aplicaciones.		X	
Herramientas de programación			
Almacenamiento de logs de funcionamiento del Frameworks.		X	
Extensibilidad y opciones adicionales			
Soporte para envío de correo electrónico.		X	

Tabla 1 Comparación entre Frameworks

1.7.5.1 Zend Framework

Framework de código abierto para PHP desarrollado por Zend⁷, implementa el patrón MVC⁸, es cien por ciento orientado a objetos y sus componentes tienen un bajo acoplamiento, puede ser usado en forma independiente y brinda un estándar de codificación que se debería seguir en los proyectos (González, 2009).

1.7.5.2 CakePHP

Framework que brinda un excelente entorno para el desarrollo y mantenimiento de aplicaciones en lenguaje PHP. Utiliza patrones de diseño habituales como MVC que reduce los costos de desarrollo y ayuda a los programadores a escribir menos código. También proporciona una estructura de organización básica, desde los nombres de los archivos hasta los de las tablas de la base de datos, manteniendo toda la aplicación consistente y lógica.

⁷ Empresa encargada de la mayor parte de las mejoras hechas a PHP.

⁸ Patrón arquitectónico Modelo-Vista-Controlador.

1.7.5.3 Symfony

Permite optimizar el desarrollo de las aplicaciones Web, se basa en el patrón MVC permitiendo separar el código del programa en tres capas, es compatible con la mayoría de los gestores de base de datos y es multiplataforma.

Razones para usar Symfony:

- Probado: ha sido probado con éxito durante años en varias aplicaciones gigantescas (*Dailymotion*⁹) y en otros miles de sitios pequeños y medianos.
- Soporte: sigue una política de tipo LTS¹⁰, por la que las versiones estables se mantienen durante tres años sin cambios pero con una continua corrección de errores.
- Licencia: se publica bajo licencia MIT, se pueden desarrollar aplicaciones web comerciales, gratuitas y de software libre.
- Código: desde su primera versión Symfony ha sido creado sólo para PHP 5, para obtener el máximo rendimiento de PHP y aprovechar todas sus características.
- Documentado: es el *framework* mejor documentado, ya que ha publicado cinco libros gratuitos de calidad y siempre actualizados. Además, toda la documentación está traducida al español.
- Calidad: su código fuente incluye más de 9.000 pruebas unitarias y funcionales.
- Internacionalización: está traducido a más de 40 idiomas e incluye todas las herramientas necesarias para traducir fácilmente las aplicaciones.

Conclusión del Epígrafe

El *framework* seleccionado para el desarrollo de la solución es Symfony, por los beneficios que aporta a diferencia de los restantes analizados. A diferencia de otros *frameworks*, proporciona herramientas que simplifican el desarrollo de las aplicaciones web. Su comprensión se centra en el conocimiento de conceptos básicos de la programación moderna, sobre todo acerca de la programación orientada a objetos, el mapeo de objetos a base de datos y el desarrollo rápido de aplicaciones, teniendo como único requisito técnico para su manipulación el conocimiento previo de PHP 5.

⁹ Dailymotion es un sitio Web de alojamiento de videos.

¹⁰ *Long Term Support* en español Soporte a Largo Plazo.

1.8. Conclusiones parciales

- Las soluciones implementadas a nivel mundial relacionadas con la gestión hotelera, no se adecuan al negocio existente en los establecimientos de la República Bolivariana de Venezuela.
- El proceso de desarrollo estará guiado por RUP siguiendo la línea de trabajo del SRHEE, utilizando a UML como lenguaje de modelado para la modelación de los artefactos generados.
- Se desarrollará la solución con el *framework* Symfony a partir de los beneficios que aporta al desarrollo de aplicaciones web utilizando PHP como lenguaje de programación.

Capítulo II: Diseño de la solución

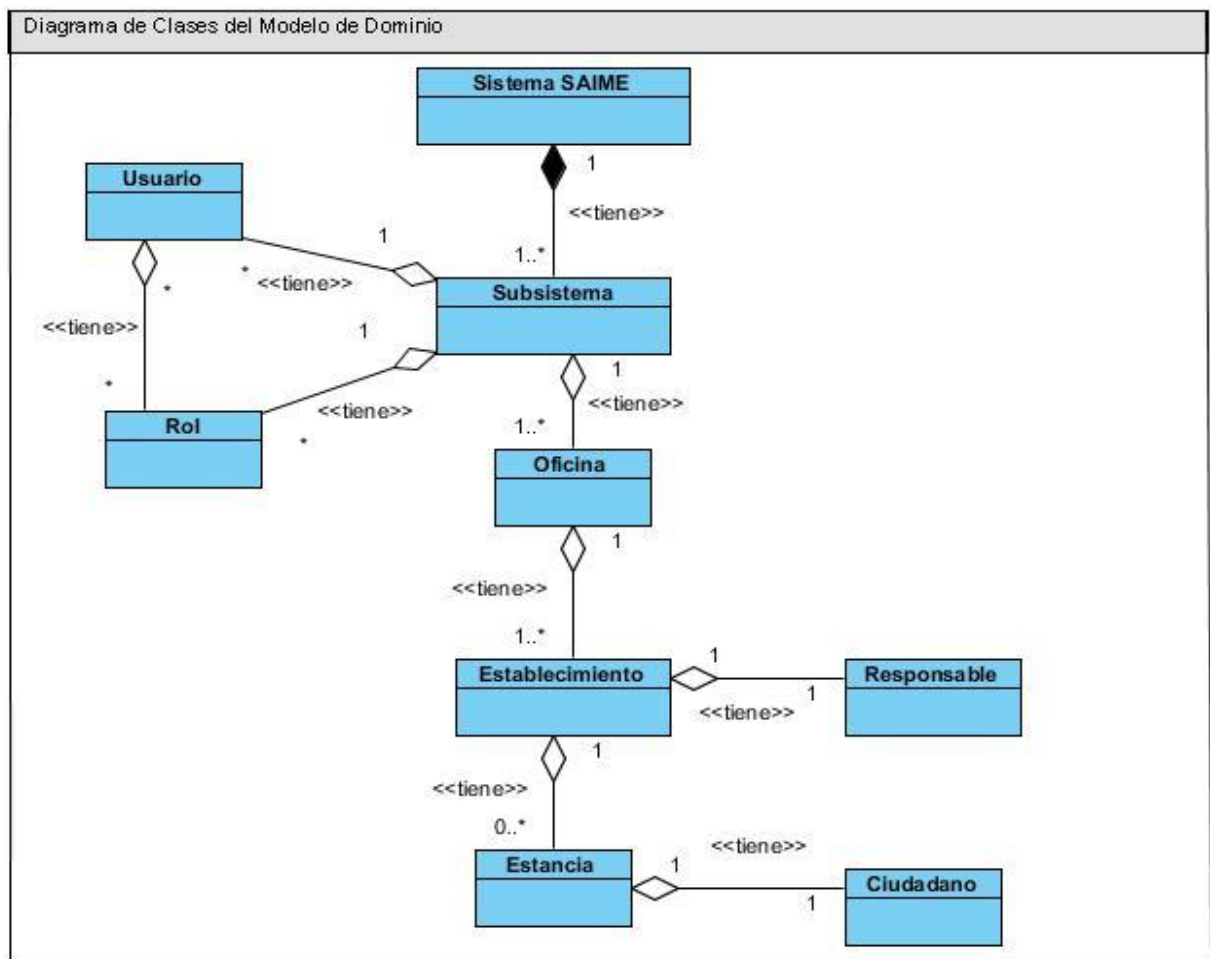
2.1. Introducción

En el presente capítulo se muestran todas las características que tendrá el sistema, partiendo del estudio de los procesos actuales del negocio, describiéndose la arquitectura y los patrones de diseño. Se identifican los requerimientos funcionales y no funcionales, y se realizarán los diferentes diagramas de clases, de interacción y los prototipos de interfaz para posibles implementaciones.

2.2. Modelo de Dominio

Luego de realizar un análisis se identificó un negocio muy pequeño, solo puede identificarse un caso de uso, por tanto se determinó no modelar el negocio a través de casos de uso, se realiza un Modelo de Dominio y se representa a través de conceptos relacionados entre sí.

Figura 1 Diagrama de Clases del Modelo de Dominio



Capítulo II: Diseño de la solución

2.2.1 Definición de las clases del Modelo de Dominio

Sistema SAIME: Es el Sistema del Servicio Administrativo de Identificación, Migración y Extranjería, en él se desarrollan un conjunto de procesos, que se implementan a través de varias aplicaciones.

Subsistema: Concepto utilizado para generalizar todos aquellos subsistemas que componen al sistema.

Usuario: Define un concepto administrativo para garantizar seguridad en el sistema, a través de él, un funcionario del sistema podrá autenticarse y dependiendo de sus roles, podrá efectuar determinadas operaciones.

Rol: Es un concepto relacionado con la autorización de los usuarios u operadores del sistema para utilizar las funcionalidades que este brinda, y a través de la administración se le asigna o revoca en dependencia de lo que deba hacer en ese momento.

Oficina: Es la estación de trabajo donde existe una o varias aplicaciones y donde los usuarios pueden trabajar con el sistema y brindar un conjunto de servicios.

Establecimiento: Lugar donde una persona se hospeda o pasa un tiempo determinado.

Responsable: Persona encargada de registrar todos los datos de cada ciudadano que se hospede en el establecimiento.

Estancia: Período de tiempo en el que la persona permanece en el establecimiento.

Ciudadano: Persona de la cual se requiere conocer sus datos para registrar la entrada o salida al país.

2.3. Requisitos Funcionales

RF1) Autenticar Usuario

RF2) Gestionar Usuario.

RF3) Registrar Huésped.

RF4) Buscar Persona.

RF5) Registrar Acompañante.

RF6) Actualizar Registro.

RF7) Emitir Reportes.

RF8) Finalizar Estancia.

2.4. Requisitos No Funcionales

Requerimientos de software.

Cliente

- Utilizar en la PC cliente el sistema operativo Windows XP SP2 y conexión a red.
- Navegador web: Mozilla Firefox.

Servidor

- Utilizar como Gestor de BD Oracle.
- Utilizar como servidor Web (aplicaciones): Apache.

Requerimientos de *hardware*.

- PC Pentium III o superior.
- Mínimo 256 Mb RAM.
- Instalación de *drivers* de impresora.

Usabilidad.

- El sistema podrá ser usado por cualquier persona que tenga los conocimientos básicos del negocio.
- La solución debe tener una interfaz gráfica que facilite el entendimiento de las funcionalidades.

Apariencia o interfaz externa.

- Todas las interfaces de usuario que se definan para el sistema respetarán los patrones de diseño establecidos para la organización.
- Las ventanas del sistema contendrán claro y bien estructurados los datos, y al mismo tiempo permitirán la interpretación correcta e inequívoca de la información.
- Se utilizará un criterio que permita la identificación visual de los elementos en la interfaz a través del uso de colores y formatos de fuente para la letra, entre otras técnicas.
- Mostrar todos los textos y mensajes en pantalla en español.
- Diseñar su funcionamiento de modo que sea intuitivo, y requiera de información mínima.

Ayuda y documentación.

- Desarrollar manual de usuario y procedimientos relativo a la utilización de la aplicación y tecnología a utilizar.

Capítulo II: Diseño de la solución

Seguridad.

Confidencialidad:

- La información manejada por el sistema estará protegida de acceso no autorizado y divulgación.
- Solo el administrador tendrá acceso total al sistema.
- Los usuarios deben estar autenticados para poder realizar alguna operación.
- Deben asignarse roles a los usuarios definiendo las operaciones que pueden realizar según los permisos que estos tengan.

Integridad:

- La información manejada por el sistema será objeto de cuidadosa protección contra la corrupción y estados inconsistentes.
- Los cambios en el sistema solo pueden ser realizados por las personas autorizadas y de la forma autorizada.

Disponibilidad:

- El sistema se mantendrá disponible 24 horas diarias durante los siete días de la semana.
- Las salvas se realizarán durante horarios nocturnos que no afecten los servicios de la organización.

2.5. Actores del Sistema

Un actor no es más que un conjunto de roles que los usuarios desempeñan cuando interaccionan con los Casos de Uso. Los actores representan a terceros fuera del sistema que colaboran con el mismo.

Actor	Descripción
Administrador	Gestiona todo lo relacionado con el manejo de usuarios y la asignación de roles.
Responsable del Establecimiento	Capta los datos de los huéspedes y realiza todas las funciones relacionadas al manejo de los datos de los mismos.

Tabla 2 Descripción de Actores del Sistema

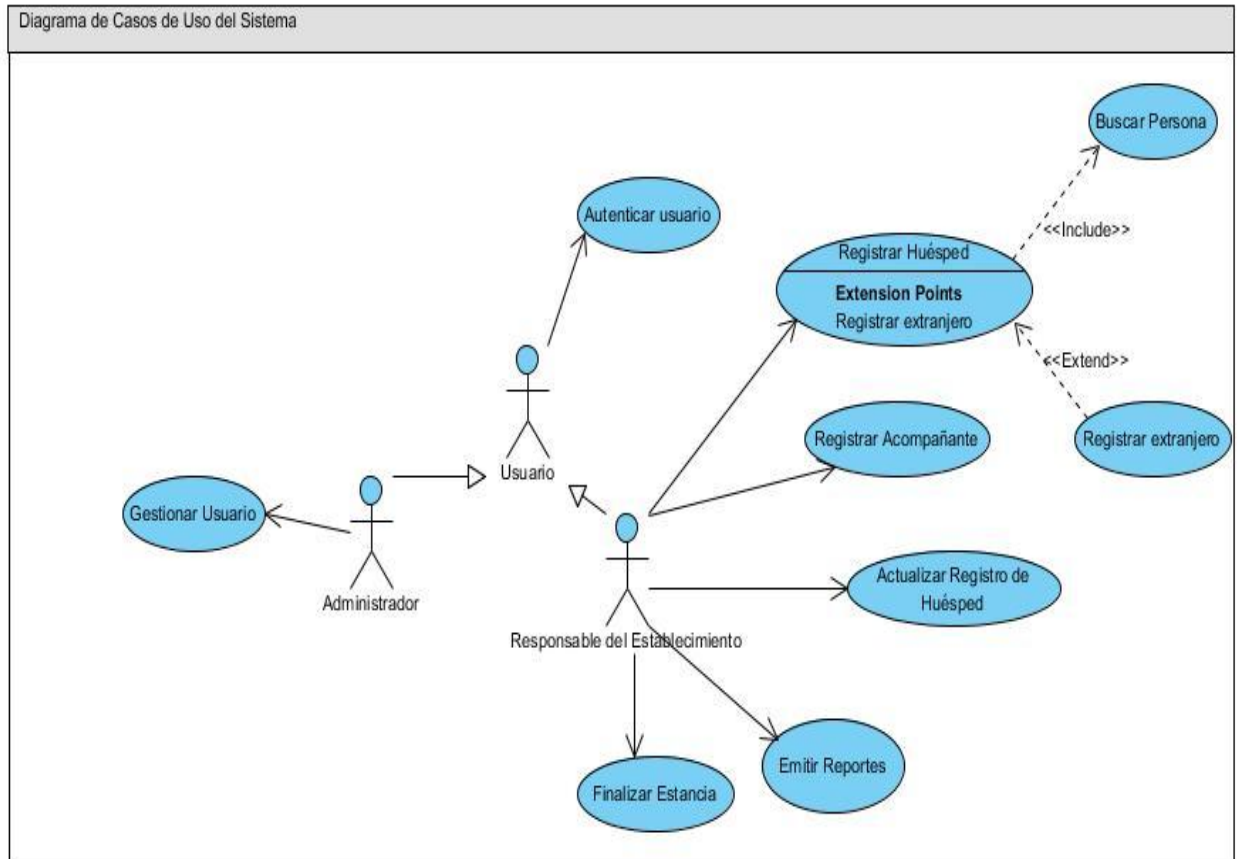
2.6. Diagrama de Casos de Uso del Sistema

Un caso de uso es una secuencia de transacciones que son desarrolladas por un sistema en respuesta a un evento que inicia un actor sobre el propio sistema. Los diagramas de Casos de Uso sirven para especificar la funcionalidad y el

Capítulo II: Diseño de la solución

comportamiento de un sistema mediante su interacción con los usuarios y otros sistemas (Vega, 2010).

Figura 2 Diagrama de Casos de Uso del Sistema



2.7. Descripción de Casos de usos del Sistema

A continuación se muestran las descripciones para cinco funcionalidades del sistema, las demás se encuentran en el **Anexo 1 Descripciones de Casos de Uso**.

Caso de Uso:	Registrar Huésped.
Actores:	Responsable del Establecimiento (inicia).
Resumen:	El Caso de Uso inicia cuando el Responsable del Establecimiento selecciona la opción Registrar Huésped para controlar la estancia de cada extranjero en el establecimiento que se hospede, finalizando el Caso de Uso con el registro de la estancia en el hotel.
Precondiciones:	El Responsable del Establecimiento debe pedir el documento de identificación al huésped.

Capítulo II: Diseño de la solución

Referencias	RF – 3.
Prioridad	Alta.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El Caso de Uso se inicia cuando el Responsable del Establecimiento selecciona la opción “Registrar Huésped”.	2. El sistema muestra la interfaz correspondiente para realizar el registro de un huésped, con los campos correspondientes y con las opciones: 2.a. Campo “Nombre(s)”. 2.b. Campo “Apellido(s)”. 2.c. Campo “Fecha Nacimiento”. 2.d. Campo “Sexo”. 2.e. Campo “Nacionalidad”. 2.f. Campo “Número de documento”. 2.g. Campo “Tipo de documento”. 2.h. Campo “Tipo de Visa”. 2.i. Campo “Número de Visa”. 2.j. Campo “Fecha de Emisión de la Visa”. 2.k. Campo “Fecha de inicio de estancia”. 2.l. Campo “Fecha de fin de estancia reportada”. 2.m. Campo “País de Origen”. 2.n. Campo “País de Destino”. 2.o. Opción “Posee Acompañante”. 2.p. Opción “Buscar”. 2.q. Opción “Registrar”. 2.r. Opción “Cancelar”. 2.s. Opción “Terminar”.

Capítulo II: Diseño de la solución

<p>3. El Responsable del Establecimiento introduce los datos relacionados al huésped:</p> <ul style="list-style-type: none"> 3.a. Campo “Nombre(s)”. 3.b. Campo “Apellido(s)”. 3.c. Campo “Fecha de Nacimiento”. 3.d. Campo “Sexo”. 3.e. Campo “Nacionalidad” 3.f. Campo “Número de documento”. 3.g. Campo “Tipo de documento”. 3.h. Opción “Cedulado”. 3.i. Campo “Tipo de Visa”. 3.j. Campo “Número de Visa”. 3.k. Campo “Fecha de Emisión de la Visa”. 3.l. Campo “Fecha de Inicio de la estancia”. 3.m. Campo “Fecha de Fin de la estancia”. 3.n. Campo “País de Origen”. 3.o. Campo “País Destino”. 3.p. Opción “Posee Acompañante”. <p>4. El Responsable realiza una de las siguientes acciones:</p> <ul style="list-style-type: none"> 4.a. “Buscar”. 4.b. “Registrar”. 4.c. “Cancelar”. 4.d. “Terminar”. 	<p>5. Según su selección:</p> <ul style="list-style-type: none"> 5.a. Si seleccionó “Buscar” el sistema realiza una búsqueda, ver Punto de Inclusion-1, si no continua con el paso 3 del Flujo Normal. 5.b. Si seleccionó “Registrar” el sistema verifica que el Responsable introdujo todos los datos, si no continúa en el Flujo Alternativo 1: “Datos Incompletos”. 5.c. Si seleccionó “Cancelar” el sistema no efectúa el registro del huésped, y continúa en el paso 2 del Flujo Normal. 5.d. Si seleccionó “Terminar” el sistema verifica que haya concluido la acción 5.e. correspondiente a registrar un huésped y finaliza el Caso de Uso, si no ir a Flujo Alternativo 2: “Terminar”. <p>6. El sistema verifica que los datos estén correctos, si no lo están continúa en el Flujo Alternativo 3: “Datos Incorrectos”.</p> <p>7. El sistema registra los datos, y muestra:</p> <ul style="list-style-type: none"> 7.a. El mensaje: “El registro se ha efectuado correctamente”. 7.b. Opción “Aceptar”.
<p>8. El Responsable del Establecimiento selecciona la opción “Aceptar”.</p>	<p>9. Continúa en el paso 2 del Flujo Normal.</p>
Flujo Alternativo 1: “Datos Incompletos”	
Acción del Actor	Respuesta del Sistema

Capítulo II: Diseño de la solución

	<p>1.1. En caso de estar incompletos los datos el sistema muestra:</p> <p>1.1.a El mensaje: “Debe completar los datos”.</p> <p>1.1.b Opción “Aceptar”.</p>
1.2. El Responsable del Establecimiento selecciona la opción “Aceptar”.	1.3. El sistema muestra la interfaz con los campos como han sido llenados y permite completarlos, continúa en el paso 3 del Flujo Normal.
Flujo Alternativo 2: “Terminar”	
Acción del Actor	Respuesta del Sistema
	<p>2.1. Si no ha concluido la acción correspondiente a registrar un huésped el sistema muestra el mensaje: “La acción actual no se ha culminado correctamente, esta será cancelada”.</p> <p>2.2. El sistema muestra:</p> <p>2.2.a Opción “Aceptar”.</p> <p>2.2.b Opción “Cancelar”.</p>
<p>2.3. El Responsable del Establecimiento realiza una de las siguientes acciones:</p> <p>2.3.a “Aceptar”.</p> <p>2.3.b “Cancelar”.</p>	<p>2.4. Según su selección:</p> <p>2.4.a Si seleccionó “Aceptar” no se registra el huésped y finaliza el Caso de Uso.</p> <p>2.4.b Si seleccionó “Cancelar” regresa al paso 2 del Flujo Normal.</p>
Flujo Alternativo 3: “Datos Incorrectos”	
Acción del Actor	Respuesta del Sistema
	<p>3.1. En caso de estar incorrectos los datos el sistema muestra:</p> <p>3.1.a El mensaje: “Debe introducir los datos correctamente”.</p>

Capítulo II: Diseño de la solución

	3.1.b Opción "Aceptar".
3.2. El Responsable del Establecimiento selecciona la opción "Aceptar".	3.3. El sistema muestra la interfaz con los campos como han sido llenados y permite corregirlos, continúa en el paso 3 del Flujo Normal.
Puntos de Inclusión	
1) CU "Buscar Persona".	
Poscondiciones éxito	Se hizo efectivo el registro de los huéspedes en el establecimiento.
Poscondiciones fallo	No se hizo efectivo el registro de los huéspedes en el establecimiento.

Tabla 3 Descripción CU Registrar Huésped

Caso de Uso:	Buscar persona.
Caso de Uso Base:	Registrar Huésped.
Actores:	Responsable del Establecimiento (inicia).
Resumen:	El Caso de Uso inicia cuando el sistema muestra la interfaz correspondiente a registrar un huésped. Se realiza la búsqueda por nacionalidad, número de documento de identificación y tipo de documento de identificación para comprobar que el huésped no haya estado hospedado en el establecimiento recientemente, finaliza el Caso de Uso cuando se obtienen los datos del huésped.
Precondiciones:	El Responsable del Establecimiento debe pedir el documento de identificación al huésped.
Referencias	RFH – 4.
Prioridad	Alta.

Flujo Normal de Eventos

Capítulo II: Diseño de la solución

Acción del Actor	Respuesta del Sistema
	1. El Caso de Uso se inicia cuando el sistema muestra la interfaz para registrar un huésped con los siguientes campos: <ul style="list-style-type: none"> 1.a. Campo “Nacionalidad”. 1.b. Campo “Número de Documento”. 1.c. Campo “Tipo de Documento”. 1.d. Opción “Buscar”.
2. El Responsable del Establecimiento introduce los datos. <ul style="list-style-type: none"> 2.a. Campo “Nacionalidad”. 2.b. Campo “Número de Documento”. 2.c. Campo “Tipo de Documento”. 3. El Responsable del Establecimiento selecciona la opción “Buscar”.	4. El sistema verifica si el Responsable introdujo todos los datos requeridos para realizar la búsqueda, si no ir a Flujo Alterno 1: “Datos Incompletos”. 5. El sistema verifica que los datos estén correctos, si no lo están continúa en el Flujo Alterno 2: “Datos Incorrectos”. 6. El sistema verifica si el huésped ha estado hospedado recientemente en el establecimiento, si no continúa en el Flujo Alterno 3: “No encontrado”. 7. El sistema devuelve los datos personales del huésped. 8. Finaliza el Caso de Uso.
Flujo Alterno 1: “Datos Incompletos”	
Acción del Actor	Respuesta del Sistema
	1.1. En caso de estar incompletos los datos el sistema muestra: <ul style="list-style-type: none"> 1.1.a El mensaje: “Debe completar los datos”. 1.1.b Opción “Aceptar”.
1.2. El Responsable del Establecimiento selecciona la opción “Aceptar”.	1.3. El sistema muestra la interfaz con los campos como han sido llenados y permite completarlos, continúa en el paso 3 del Flujo Normal.

Capítulo II: Diseño de la solución

Flujo Alternativo 2: “Datos Incorrectos”	
Acción del Actor	Respuesta del Negocio
	2.1. En caso de estar incorrectos los datos el sistema muestra: 2.1.a El mensaje: “Debe introducir los datos correctamente”. 2.1.b Opción “Aceptar”.
2.2. El Responsable del Establecimiento selecciona la opción “Aceptar”.	2.3. El sistema muestra la interfaz con los campos como han sido llenados y permite corregirlos, continúa en el paso 3 del Flujo Normal.
Flujo Alternativo 3: “No encontrado”	
Acción del Actor	Respuesta del Negocio
	3.1. En caso de no ser encontrado, el sistema muestra: 3.1.a El mensaje: “El huésped nunca se ha registrado en esta instalación”. 3.1.b Opción “Aceptar”.
3.2. El Responsable del Establecimiento selecciona la opción “Aceptar”.	3.3. Finaliza el Caso de Uso.
Poscondiciones éxito	Se realizó la búsqueda exitosamente y el huésped fue encontrado.
Poscondiciones fallo	No existe ningún huésped con estas características que se haya hospedado antes en el establecimiento.

Tabla 4 Descripción CU Buscar Persona

Caso de Uso:	Registrar Acompañante.
Actores:	Responsable del Establecimiento (inicia).
Resumen:	El Caso de Uso inicia cuando el sistema muestra la interfaz correspondiente a registrar acompañante y el Responsable del

Capítulo II: Diseño de la solución

	Establecimiento asigna los acompañantes a un extranjero determinado, finaliza en Caso de uso con el registro del acompañante.
Precondiciones:	El Responsable del Establecimiento debe pedir el documento de identificación al huésped.
Referencias	RFH – 5.
Prioridad	Media.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El Caso de Uso se inicia cuando el Responsable del Establecimiento selecciona la opción “Registrar Acompañantes”.	2. El sistema muestra una interfaz con el listado de todos los huéspedes registrados en la instalación.
3. El Responsable del Establecimiento selecciona el extranjero al que quiere asignarle un acompañante.	4. El sistema muestra un listado con los extranjeros disponibles para asignarlos como acompañantes del extranjero seleccionado.
5. El Responsable del Establecimiento selecciona al extranjero que desea asignar como acompañante y presiona el botón “Registrar”.	6. El sistema muestra el mensaje “Acompañante registrado con éxito” y el acompañante registrado pasa al listado de acompañantes. 7. Finaliza el Caso de Uso.
Poscondiciones éxito	Se asignó correctamente el acompañante del extranjero deseado.
Poscondiciones fallo	No se asignó acompañante para el extranjero deseado.

Tabla 5 Descripción CU Registrar Acompañante

Capítulo II: Diseño de la solución

Caso de Uso:	Actualizar Registro de Huésped.
Actores:	Responsable del Establecimiento (inicia).
Resumen:	El Caso de Uso inicia cuando el sistema muestra la interfaz Actualizar Registro de Huésped para modificar la fecha de salida de los huéspedes en el caso que se hayan retirado antes o después de la fecha prevista inicialmente, el Caso de uso termina cuando es modificada la fecha de salida.
Precondiciones:	El Responsable del Establecimiento debe pedir el documento de identificación al huésped.
Referencias	RFH – 6.
Prioridad	Media.

Flujo Normal de Eventos

Acción del Actor	Respuesta del Sistema
1. El Caso de Uso se inicia cuando el Responsable del Establecimiento selecciona la opción “Actualizar Registro de Huésped”.	2. El sistema muestra una interfaz con el listado de todos los huéspedes registrados en la instalación.
3. El Responsable del Establecimiento selecciona al Huésped que desea modificarle la “Fecha de salida”.	4. El sistema muestra los datos del extranjero seleccionado en los campos. <ul style="list-style-type: none"> 4.a. Datos del Viaje <ul style="list-style-type: none"> 4.a.1 Campo “País Origen”. 4.a.2 Campo “País Destino”. 4.b. Datos del Huésped <ul style="list-style-type: none"> 4.b.1 Campo “Nombre(s)”. 4.b.2 Campo “Apellidos”. 4.b.3 Campo “Nacionalidad”. 4.b.4 Campo “Número de documento de identificación”. 4.b.5 Campo “Sexo”. 4.b.6 Campo “Tipo visa”. 4.b.7 Campo “Número de visa”. 4.b.8 Campo “Fecha de

Capítulo II: Diseño de la solución

	<p>nacimiento”.</p> <p>4.b.9 Campo “Fecha entrada”.</p> <p>4.b.10 Campo “Fecha de salida”.</p> <p>4.c. Opción “Actualizar”.</p> <p>4.d. Opción “Terminar”.</p>
<p>5. El Responsable del Establecimiento modifica la “Fecha de salida” y selecciona la opción:</p> <p>5.a. Opción “Actualizar”.</p> <p>5.b. Opción “Terminar”.</p>	<p>6. Según su selección:</p> <p>6.a. Si seleccionó “Actualizar” se muestra un mensaje de conformación “Desea actualizar el registro seleccionado”.</p> <p>6.a.1 Opción “Aceptar”.</p> <p>6.a.2 Opción “Cancelar”.</p> <p>6.b. Si seleccionó “Terminar” ver Flujo alternativo “Terminar”.</p>
<p>7. El Responsable del Establecimiento selecciona la opción:</p> <p>7.a. Opción “Aceptar”.</p> <p>7.b. Opción “Cancelar”.</p>	<p>8. Según su selección:</p> <p>8.a. Si seleccionó “Aceptar” se muestra el mensaje “Registro actualizado con éxito”.</p> <p>8.a.1 Opción “Aceptar”.</p> <p>8.b. Si seleccionó “Cancelar” no se actualiza el registro, finaliza el Caso de uso.</p>
<p>9. El Responsable del Establecimiento presiona la opción “Aceptar”.</p>	<p>10. Finaliza el Caso de uso.</p>

Flujo Alternativo 1: “Datos Incompletos”

Acción del Actor	Respuesta del Sistema
	<p>1.1. En caso de estar incompletos los datos el sistema muestra:</p> <p>1.1.a El mensaje: “Debe completar los datos”.</p> <p>1.1.b Opción “Aceptar”.</p>
<p>1.2. El Responsable del Establecimiento presiona la opción “Aceptar”.</p>	<p>1.3. El sistema muestra la interfaz con los campos como han sido llenados y permite completarlos, continúa en</p>

Capítulo II: Diseño de la solución

	el paso 4 del Flujo Normal.
Flujo Alternativo 2: “Datos Incorrectos”	
Acción del Actor	Respuesta del Negocio
	<p>2.1. En caso de estar incorrectos los datos el sistema muestra:</p> <p>2.1.a El mensaje: “Debe introducir los datos correctamente”.</p> <p>2.2. Opción “Aceptar”.</p>
2.3. El Responsable del Establecimiento selecciona la opción “Aceptar”.	2.4. El sistema muestra la interfaz con los campos como han sido llenados y permite corregirlos, continúa en el paso 4 del Flujo Normal.
Flujo Alternativo 3: “Terminar”	
Acción del Actor	Respuesta del Negocio
	3.1. Si no ha concluido la acción correspondiente a actualizar el registro muestra el mensaje: “La acción actual no se ha culminado correctamente, esta será cancelada”.
3.2. El Responsable del Establecimiento realiza una de las siguientes acciones: 1.2.a “Aceptar”. 3.2.b “Cancelar”.	3.3. Según su selección: 3.3.a Si seleccionó “Aceptar” no se actualiza el registro y finaliza el Caso de Uso. 3.3.b Si seleccionó “Cancelar” regresa al paso 4 del Flujo
Poscondiciones éxito	Se actualizó el registro del huésped correctamente.
Poscondiciones fallo	No se actualizó el registro del huésped.

Tabla 6 Descripción CU Actualizar Registro de Huéspedes

Capítulo II: Diseño de la solución

Caso de Uso:	Finalizar Estancia Hotel.
Actores:	Responsable del Establecimiento (inicia).
Resumen:	El Caso de Uso inicia cuando el Responsable del Establecimiento selecciona la opción Finalizar Estancia, se muestran todas las estancias que están abiertas, se selecciona la estancia a finalizar, terminando el Caso de Uso con la finalización de la estancia.
Precondiciones:	Debe existir al menos una estancia abierta.
Referencias	RF – 8.
Prioridad	Alta.

Flujo Normal de Eventos

Acción del Actor	Respuesta del Sistema
1. El Caso de Uso se inicia cuando el Responsable del Establecimiento selecciona la opción “Finalizar Estancia”	2. El sistema muestra la interfaz correspondiente con los datos de las personas que han realizado las estancias y las opciones: 2.a. Campo “Nombre(s)”. 2.b. Campo “Apellido(s)”. 2.c. Campo “Fecha Nacimiento”. 2.d. Campo “Sexo”. 2.e. Campo “Nacionalidad”. 2.f. Campo “Número de documento”. 2.g. Campo “Fecha inicio de estancia”. 2.h. Campo “Fecha fin”. 2.i. Opción “Finalizar”. 2.j. Opción “Terminar”
3. El Responsable del Establecimiento selecciona la estancia a finalizar e	5. Según su selección: 5.a. Si seleccionó “Finalizar” el

Capítulo II: Diseño de la solución

<p>introduce la fecha de salida.</p> <p>4. El Responsable del Establecimiento realiza una de las siguientes acciones:</p> <p>4.a. “Finalizar”.</p> <p>4.b. “Terminar”.</p>	<p>sistema verifica que el Responsable del Hotel introdujo todos los datos, si no continúa en el Flujo Alternativo 1: “Datos Incompletos”.</p> <p>5.b. Si seleccionó “Terminar” el sistema verifica que haya concluido la acción correspondiente a finalizar una estancia y finaliza el Caso de Uso.</p> <p>6. El sistema finaliza la estancia, y muestra:</p> <p>6.a. El mensaje: “Estancia finalizada con éxito”.</p> <p>6.b. Opción “Aceptar”.</p>
<p>7. El Responsable del Establecimiento selecciona la opción “Aceptar”.</p>	<p>8. Continúa en el paso 2 del Flujo Normal.</p>
Flujo Alternativo 1: “Datos Incompletos”	
Acción del Actor	Respuesta del Sistema
	<p>1.1. En caso de estar incompletos los datos el sistema muestra:</p> <p>1.1.a El mensaje: “Debe completar los datos”.</p> <p>1.1.b Opción “Aceptar”.</p>
<p>1.2. El Responsable del Establecimiento selecciona la opción “Aceptar”.</p>	<p>1.3. El sistema muestra la interfaz con los campos como han sido llenados y permite completarlos, continúa en el paso 3 del Flujo Normal.</p>
Poscondiciones éxito	Se hizo efectiva la finalización de la estancia.
Poscondiciones fallo	No se hizo efectiva la finalización de la estancia.

Tabla 7 Descripción CU Finalizar Estancia

Capítulo II: Diseño de la solución

Caso de Uso:	Gestionar Usuario.
Actores:	Administrador (inicia).
Resumen:	El Caso de Uso inicia cuando el administrador selecciona la opción Gestionar Usuario, luego realiza la acción de adicionar, eliminar o modificar un usuario, el sistema actualiza y termina el Caso de uso.
Precondiciones:	El administrador debe estar autenticado.
Referencias	RF-2
Prioridad	Alta.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El Caso de Uso se inicia cuando el Administrador selecciona la opción Gestionar Usuario.	2. El sistema muestra las opciones: 2.a. Opción "Adicionar usuario". 2.b. Opción "Modificar usuario". 2.c. Opción "Eliminar usuario".
3. El Administrador selecciona la opción deseada: 3.a. Opción "Adicionar usuario". 3.b. Opción "Modificar usuario". 3.c. Opción "Eliminar usuario".	2. Según su selección: 4.a. Si seleccionó "Adicionar usuario" ver Escenario 1 "Adicionar usuario". 4.b. Si seleccionó "Modificar usuario" ver Escenario 2 "Modificar usuario". 4.c. Si seleccionó "Eliminar usuario" ver Escenario 3 "Eliminar usuario".
Escenario 1 "Adicionar usuario"	
	1.1. El sistema muestra una interfaz con los campos: 1.1.a Campo "Nombres". 1.1.b Campo "Apellidos". 1.1.c Campo "Fecha de nacimiento". 1.1.d Campo "Sello". 1.1.e Campo "Contraseña".

Capítulo II: Diseño de la solución

	<p>1.1.f Campo “Rol”.</p> <p>1.1.g Opción “Aceptar”.</p> <p>1.1.h Opción “Cancelar”.</p>
<p>1.2. El administrador llena los campos:</p> <p>1.2.a Campo “Nombres”.</p> <p>1.2.b Campo “Apellidos”.</p> <p>1.2.c Campo “Fecha de nacimiento”.</p> <p>1.2.d Campo “Sello”.</p> <p>1.2.e Campo “Contraseña”.</p> <p>1.2.f Campo “Rol”.</p> <p>1.2.g Opción “Aceptar”.</p> <p>1.2.h Opción “Cancelar”.</p> <p>1.3. Realiza una de las siguientes acciones:</p> <p>1.3.a Aceptar.</p> <p>1.3.b Cancelar.</p>	<p>1.4. Según su selección:</p> <p>1.4.a Si seleccionó “Aceptar” el sistema verifica que el Administrador introdujo todos los datos, si no continúa en el Flujo Alterno 1: “Datos Incompletos”.</p> <p>1.4.b El sistema verifica que los datos estén correctos, si no lo están continúa en el Flujo Alterno 2: “Datos Incorrectos”.</p> <p>1.4.c Si seleccionó “Cancelar” el sistema no adiciona al usuario, y continúa en el paso 3 del Flujo Normal.</p> <p>1.4.d El sistema registra los datos, y muestra:</p> <p>1.4.e El mensaje: “El usuario se ha añadido correctamente”.</p> <p>1.4.f Opción “Aceptar”.</p>
<p>1.5. Presiona el botón “Aceptar”.</p>	<p>1.6. Finaliza el Caso de uso.</p>
Escenario 2 “Modificar Usuario”	
	<p>2.1. El sistema muestra una interfaz con el listado de todos los usuarios que han sido añadidos.</p>
<p>2.2. El Administrador selecciona el usuario al que desea modificarle los datos y presiona el botón “Aceptar”.</p>	<p>2.3. El sistema muestra los datos del usuario seleccionado.</p> <p>2.3.a Campo “Nombres”.</p> <p>2.3.b Campo “Apellidos”.</p> <p>2.3.c Campo “Fecha de nacimiento”.</p> <p>2.3.d Campo “Sello”.</p> <p>2.3.e Campo “Contraseña”.</p>

Capítulo II: Diseño de la solución

	<p>2.3.f Campo “Rol”.</p> <p>2.3.g Opción “Aceptar”.</p> <p>2.3.h Opción “Cancelar”.</p>
<p>2.4. El Administrador modifica los datos:</p> <p>2.4.a Campo “Nombres”.</p> <p>2.4.b Campo “Apellidos”.</p> <p>2.4.c Campo “Fecha de nacimiento”.</p> <p>2.4.d Campo “Sello”.</p> <p>2.4.e Campo “Contraseña”.</p> <p>2.4.f Campo “Rol”.</p> <p>2.5. Realiza una de las siguientes acciones:</p> <p>2.5.a Aceptar.</p> <p>2.5.b Cancelar.</p>	<p>2.6. Según su selección:</p> <p>2.6.a Si seleccionó “Aceptar” el sistema verifica que el Administrador introdujo todos los datos, si no continúa en el Flujo Alternativo 1: “Datos Incompletos”.</p> <p>2.6.b El sistema verifica que los datos estén correctos, si no lo están continúa en el Flujo Alternativo 2: “Datos Incorrectos”.</p> <p>2.6.c Si seleccionó “Cancelar” el sistema no modifica los datos del usuario, y continúa en el paso 3 del Flujo Normal.</p> <p>2.6.d El sistema actualiza los datos, y muestra:</p> <p>2.6.e El mensaje: “Los datos del usuario se han modificado correctamente”.</p> <p>2.6.f Opción “Aceptar”.</p>
<p>2.7. Presiona el botón “Aceptar”.</p>	<p>2.7.1.1. Finaliza el Caso de uso.</p>
Escenario 3 “Eliminar usuario”	
	<p>3.1. El sistema muestra una interfaz con el listado de todos los usuarios que han sido añadidos.</p>
<p>3.2. El Administrador selecciona el usuario al que desea eliminar y presiona el botón “Aceptar”.</p>	<p>3.3. El sistema muestra un mensaje de confirmación “Desea eliminar este usuario”.</p>
<p>3.4. Presiona el botón “Aceptar”.</p>	<p>3.5. Elimina al usuario del listado de usuarios disponibles.</p>

Capítulo II: Diseño de la solución

	3.6. Finaliza el Caso de uso.
Flujo Alternativo 1: “Datos Incompletos”	
	1.1. En caso de estar incompletos los datos el sistema muestra: 1.1.a El mensaje: “Debe completar los datos”. 1.1.b Opción “Aceptar”.
1.2. El Administrador presiona el botón “Aceptar”.	1.3. El sistema muestra la interfaz con los campos como han sido llenados y permite completarlos.
Flujo Alternativo 2: “Datos Incorrectos”	
	2.1. En caso de estar incorrectos los datos el sistema muestra: 2.1.b El mensaje: “Debe introducir los datos correctamente”. 2.1.b Opción “Aceptar”.
2.2. El Administrador presiona el botón “Aceptar”.	2.3. El sistema muestra la interfaz con los campos como han sido llenados y permite corregirlos.
Poscondiciones éxito	Se realizó la gestión de los usuarios permitiendo modificarlos, adicionarlos y eliminarlos.
Poscondiciones fallo	No se pudo adicionar, modificar o eliminar los usuarios.

Tabla 8 Descripción CU Gestionar Usuario

2.8. Modelos de Clases del Análisis

Los diagramas de clases del análisis muestran sus clases participantes y sus relaciones. Se utilizan para coordinar todos los requisitos sobre una clase y sus objetos que pueden tener diferentes casos de uso. (Jacobson, 2000)

El modelo de clases del análisis está estructurado por clases y paquetes estereotipados que proporcionan la estructura de la vista interna del sistema. Este

Capítulo II: Diseño de la solución

modelo define realizaciones de casos de uso, y cada una de ellas representa el análisis de un caso de uso del modelo de casos de uso.

A continuación se muestran los diagramas de clases del análisis para cinco Casos de Uso, los demás se encuentran en el **Anexo 2 Modelos de Clases del Análisis**.

Figura 3 Diagrama de Clases del Análisis CU Registrar Huésped

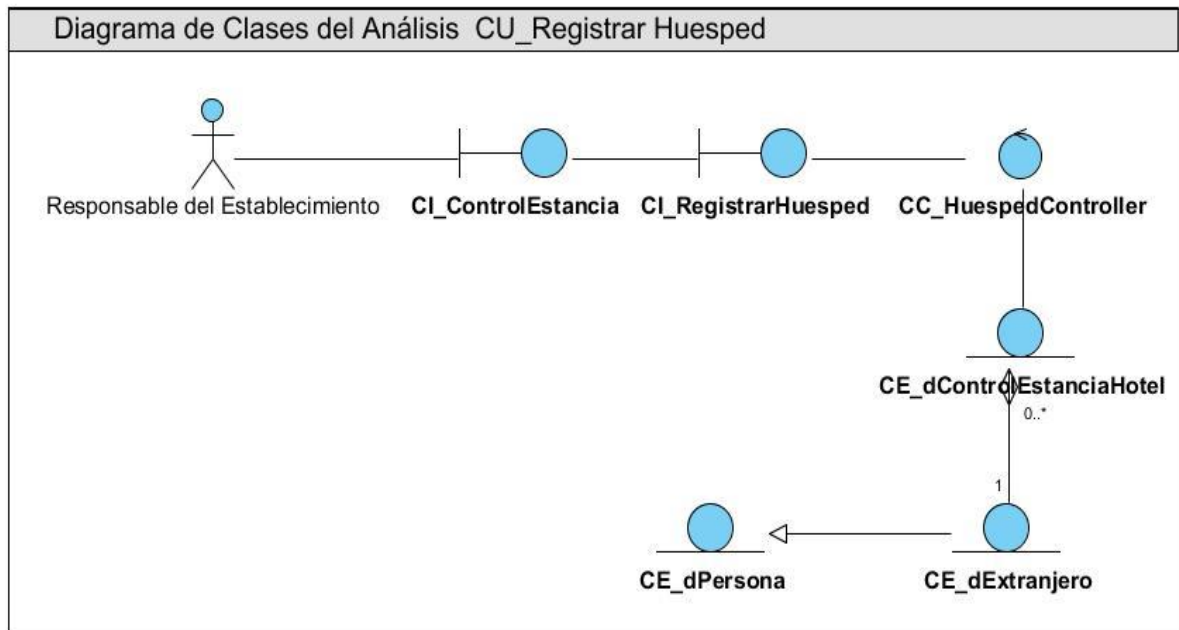
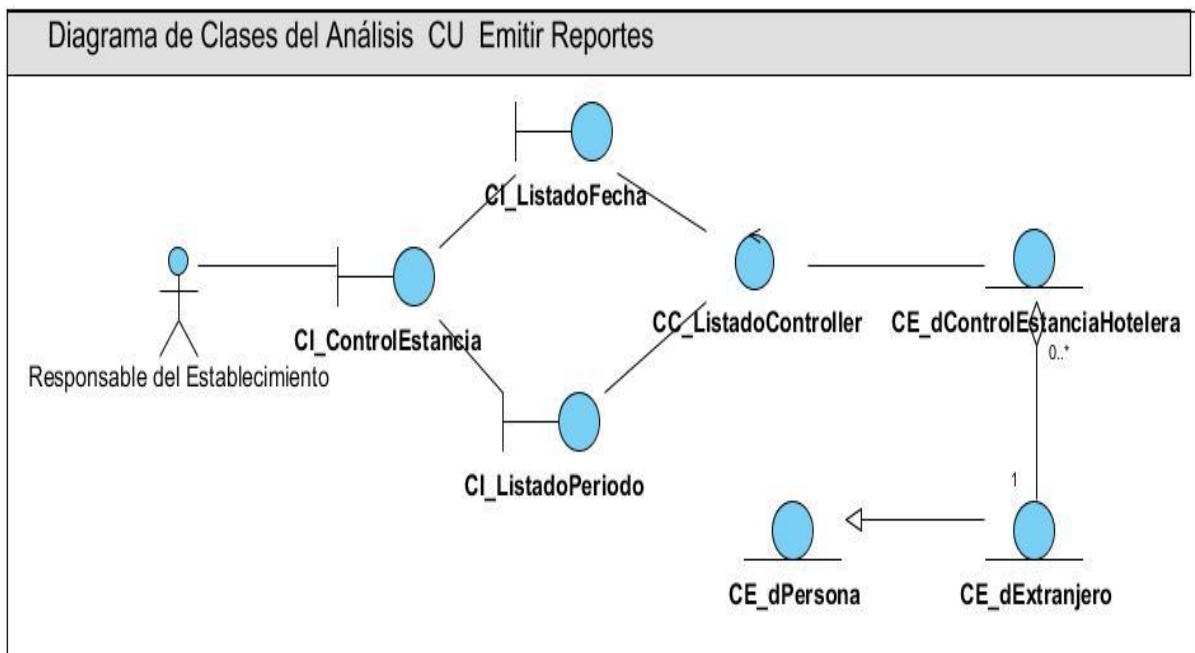


Figura 4 Diagrama de Clases del Análisis CU Emitir Reportes



Capítulo II: Diseño de la solución

Figura 5 Diagrama de Clases del Análisis CU Buscar Persona

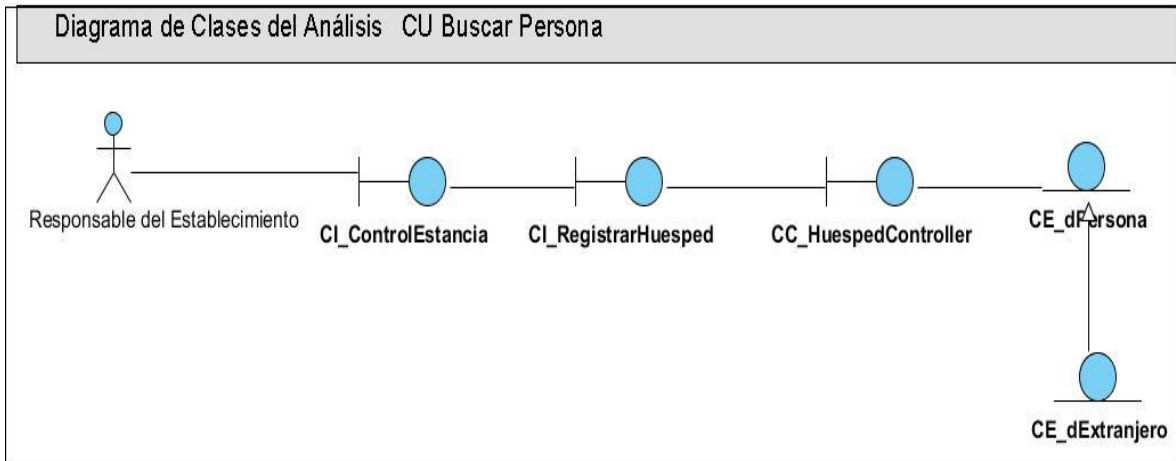


Figura 6 Diagrama de Clases del Análisis CU Actualizar Registro de Huésped

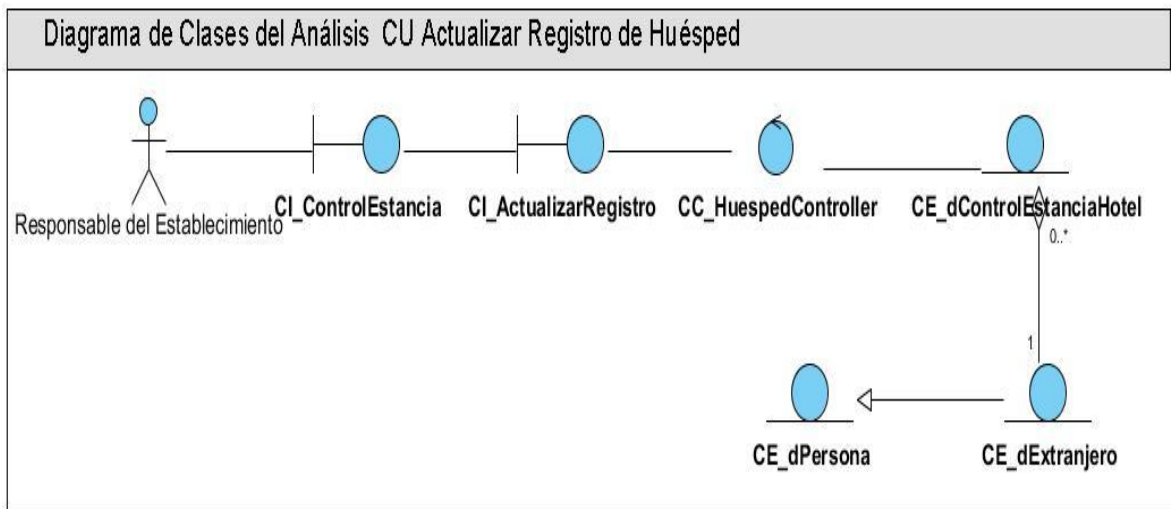
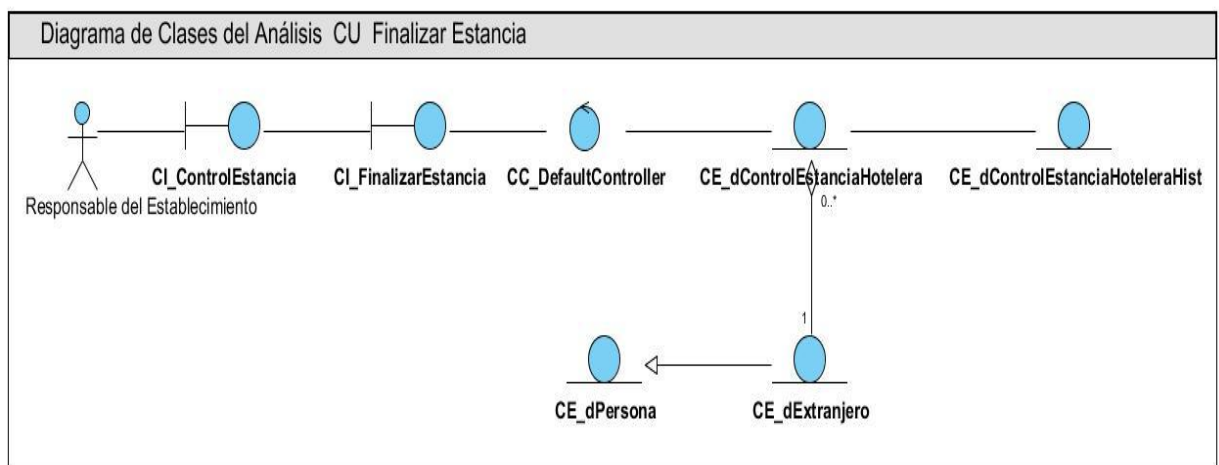


Figura 7 Diagrama de Clases del Análisis CU Finalizar Estancia



2.9. Arquitectura de Software

La Arquitectura de Software es una vista del sistema que incluye los componentes principales del mismo, la conducta de esos componentes según se percibe desde el resto del sistema y las formas en que los componentes interactúan y se coordinan para alcanzar la misión del sistema. (Clements, 2003)

2.9.1 Patrón de arquitectura Cliente/Servidor

La arquitectura Cliente/Servidor es aquella en la que confluyen una serie de aplicaciones basadas en dos categorías que cumplen funciones diferentes (una requiere servicios y la otra los brinda), pero que a la vez, pueden realizar tanto actividades en forma conjunta como independientemente. Esas dos categorías son justamente, cliente y servidor. En el caso del cliente, es aquel que requiere un servicio del servidor.

En esta categoría se realizan funciones de *software* basándose en el *hardware*, pero en caso de no tener la capacidad de procesar los datos necesarios, recurre al servidor y espera a que este le brinde los servicios solicitados. El cliente es una estación de trabajo o computadora que está conectada a una red a través de la cual puede acceder al servidor.

Las funciones que lleva a cabo el proceso cliente se resumen en los siguientes puntos:

- Administrar la interfaz de usuario.
- Interactuar con el usuario.
- Procesar la lógica de la aplicación y hacer validaciones locales.
- Generar requerimientos de base de datos.
- Recibir resultados del servidor.
- Formatear resultados.

Las funciones que lleva a cabo el proceso servidor se resumen en los siguientes puntos:

- Aceptar los requerimientos de base de datos que hacen los clientes.
- Procesar requerimientos de base de datos.
- Formatear datos para transmitirlos a los clientes.
- Procesar la lógica de la aplicación y realizar validaciones a nivel de base de datos.

2.9.2 Patrón de arquitectura Modelo Vista Controlador

Modelo Vista Controlador (MVC): 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. Facilita agregar nuevos tipos de datos según sea requerido por la aplicación ya que son independientes del funcionamiento de las otras capas.

Modelo: Es la representación de la información que maneja la aplicación. Representa los datos puros que puestos en el contexto del sistema proveen de información al usuario.

Vista: Es la representación del modelo en forma gráfica, disponible para la interacción con el usuario. En el caso de una aplicación Web, la “Vista” es una página HTML con contenido dinámico sobre el cual el usuario puede realizar operaciones.

Controlador: Es la capa encargada de manejar y responder las solicitudes del usuario, procesando la información necesaria y modificando el Modelo en caso de ser necesario.

2.10. Patrones de Diseño

Para la realización del diseño se tomaron en cuenta un conjunto de patrones para la descripción del problema y la esencia de su solución, de forma que la solución se puede reutilizar en diferentes situaciones: “Los patrones de diseño de software son soluciones reutilizables de problemas recurrentes que aparecen durante el proceso de diseño de software orientado a objetos” (Ordóñez, 2004).

2.10.1 Patrones GRASP (*General Responsibility Assignment Software Patterns*)

Se utilizaron los Patrones GRASP para asignar correctamente las responsabilidades a cada una de las clases que intervienen en el modelo. Se usaron los siguientes:

Experto: Asignar una responsabilidad al experto en información: la clase que cuenta con la información necesaria para cumplir la responsabilidad (Larman, 1999).

Ejemplo: Se evidencia en las clases librerías, que son las que cuentan con la información necesaria para cumplir las responsabilidades sobre los elementos de negocio, como `fpdf_lib`.

Creador: Este patrón como su nombre lo indica es el que crea, guía la asignación de responsabilidades relacionadas con la creación de objetos, se asigna la responsabilidad de que una clase B cree un Objeto de la clase A (Larman, 1999).

Capítulo II: Diseño de la solución

Ejemplo: Se evidencia en el AppKernel.php que se encarga de registrar todos los *bundles* ¹¹ activos de la aplicación y carga el archivo de configuración app/config/config_prod.yml. A su vez, este archivo importa el archivo app/config/config.yml que carga las rutas del archivo app/config/routing.yml.

Controlador: Asignar la responsabilidad del manejo de un mensaje de los eventos de un sistema (evento de alto nivel generado por un actor externo) a una clase (Larman, 1999).

Ejemplo:

Las clases controladoras que se encargan de obtener datos, fabricar una respuesta y enviar las vistas como HuespedController.php y UsuarioController.php.

Bajo acoplamiento: Asignar una responsabilidad para mantener bajo acoplamiento. El acoplamiento es una medida de la fuerza con que una clase está conectada a otras clases, con que las conoce y con que recurre a ellas. Una clase con bajo (o débil) acoplamiento no depende de muchas otras (Larman, 1999).

Alta cohesión: Asignar una responsabilidad de modo que la cohesión siga siendo alta. Una clase tiene responsabilidades moderadas en un área funcional y colabora con las otras para llevar a cabo las tareas. Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas (Larman, 1999).

Ejemplo de estos dos últimos patrones:

La propia implementación de Symfony 2 contiene estos patrones nivelados pues permite el uso de los componentes de forma individual, evidenciando el bajo acoplamiento, así como la dependencia entre ellos o alta cohesión.

2.11. Modelos de Clases del Diseño

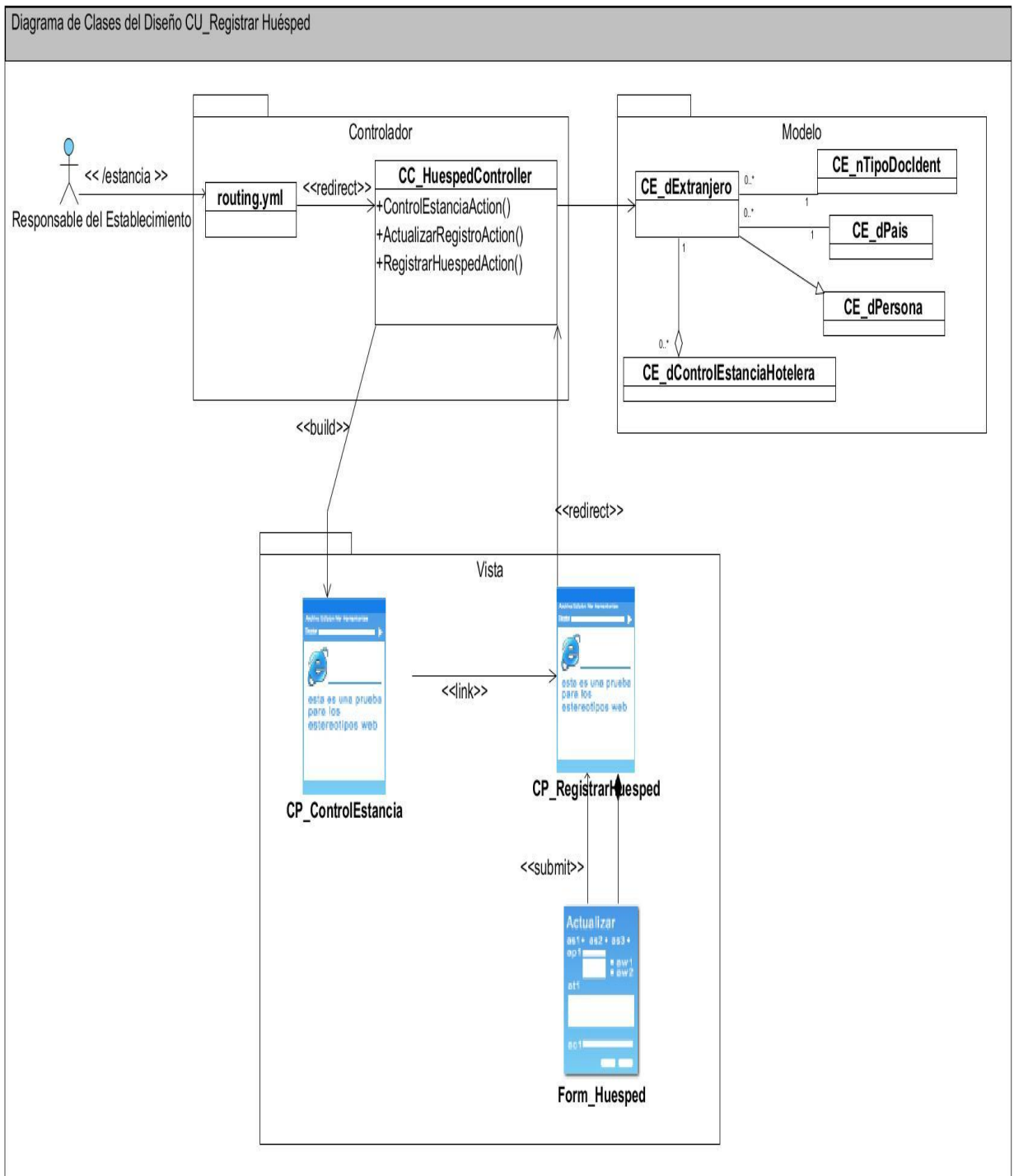
En un diagrama de clases de diseño se muestran los atributos y métodos de cada clase y se representa de forma sencilla la colaboración y las responsabilidades de las distintas clases que forman el sistema. Se utilizan para modelar la vista de diseño estática de un sistema, esto incluye el vocabulario del sistema, las colaboraciones o esquemas (Segura, 2011).

A continuación se muestran los diagramas de clase del diseño para tres Casos de Uso los demás se encuentran en el **Anexo 3 Modelos de Clases del Diseño**.

¹¹ Módulo de la aplicación en Symfony.

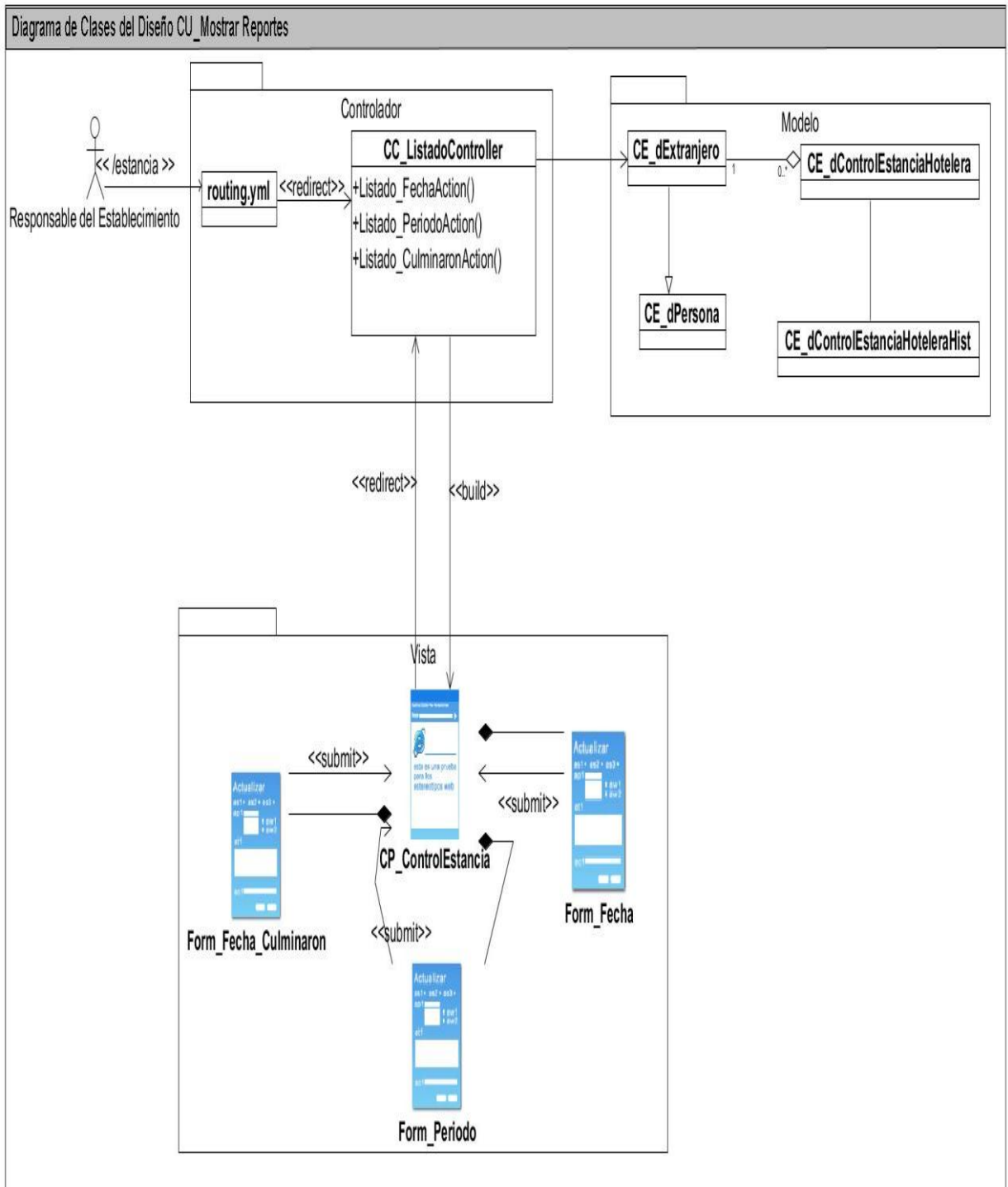
Capítulo II: Diseño de la solución

Figura 8 Diagrama de Clases del Diseño CU Registrar Huésped



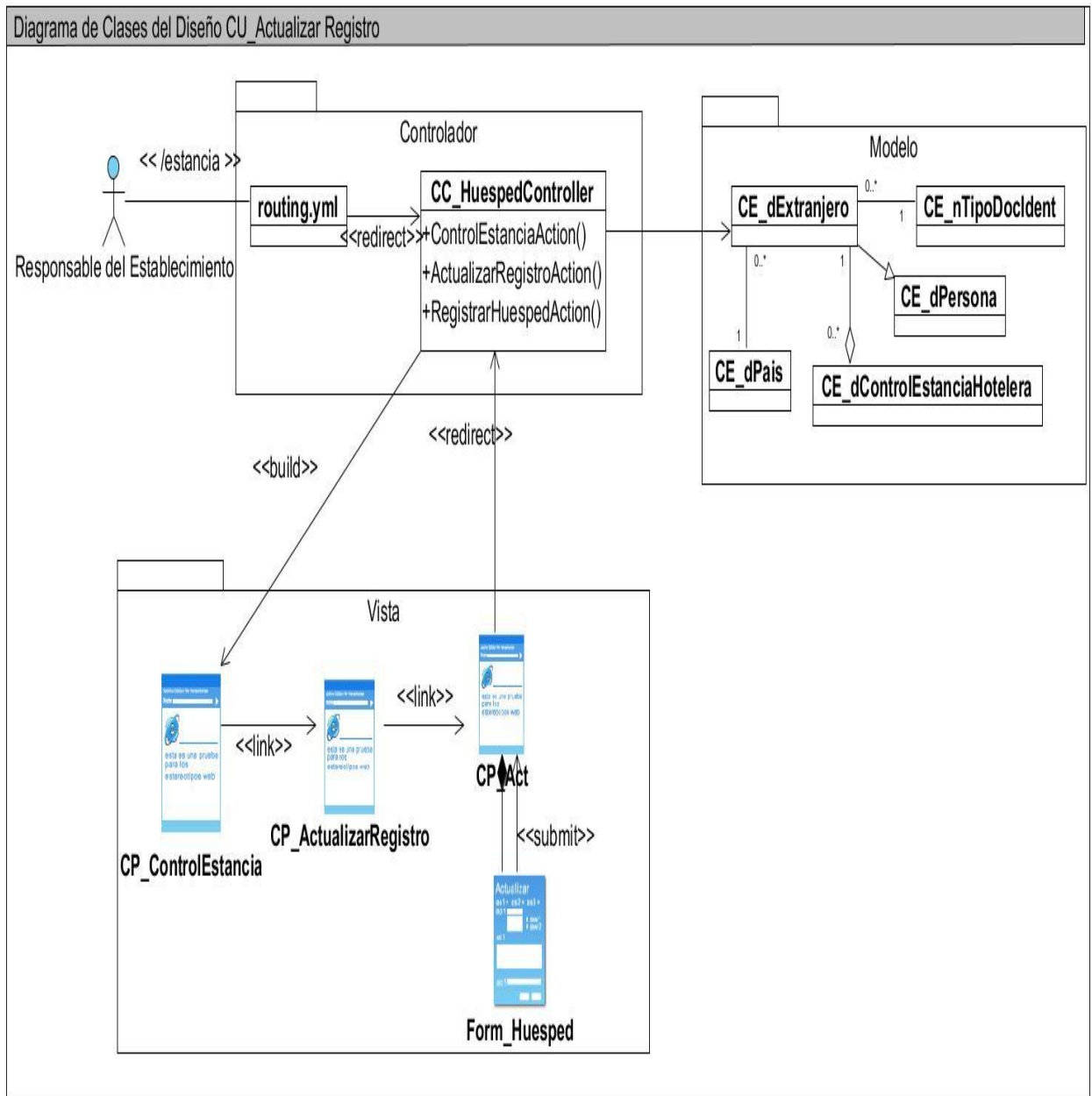
Capítulo II: Diseño de la solución

Figura 9 Diagrama de Clases del Diseño CU Mostrar Reportes



Capítulo II: Diseño de la solución

Figura 10 Diagrama de Clases del Diseño CU Actualizar Registro



2.12. Diagramas de interacción

Los diagramas de interacción son diagramas que describen como grupos de objetos colaboran para conseguir algún fin. Estos diagramas muestran objetos, así como los mensajes que se pasan entre ellos dentro del caso de uso.

Por cada caso de uso se ha realizado un diagrama de interacción (específicamente el diagrama de secuencia), donde se expone el flujo principal de información entre los objetos del diseño, con sus métodos y parámetros.

A continuación se muestran los diagramas de interacción para dos Casos de Uso los restantes se encuentran en el **Anexo 4 Diagramas de interacción**.

Capítulo II: Diseño de la solución

Figura 11 Diagrama de Secuencia CU Actualizar Registro

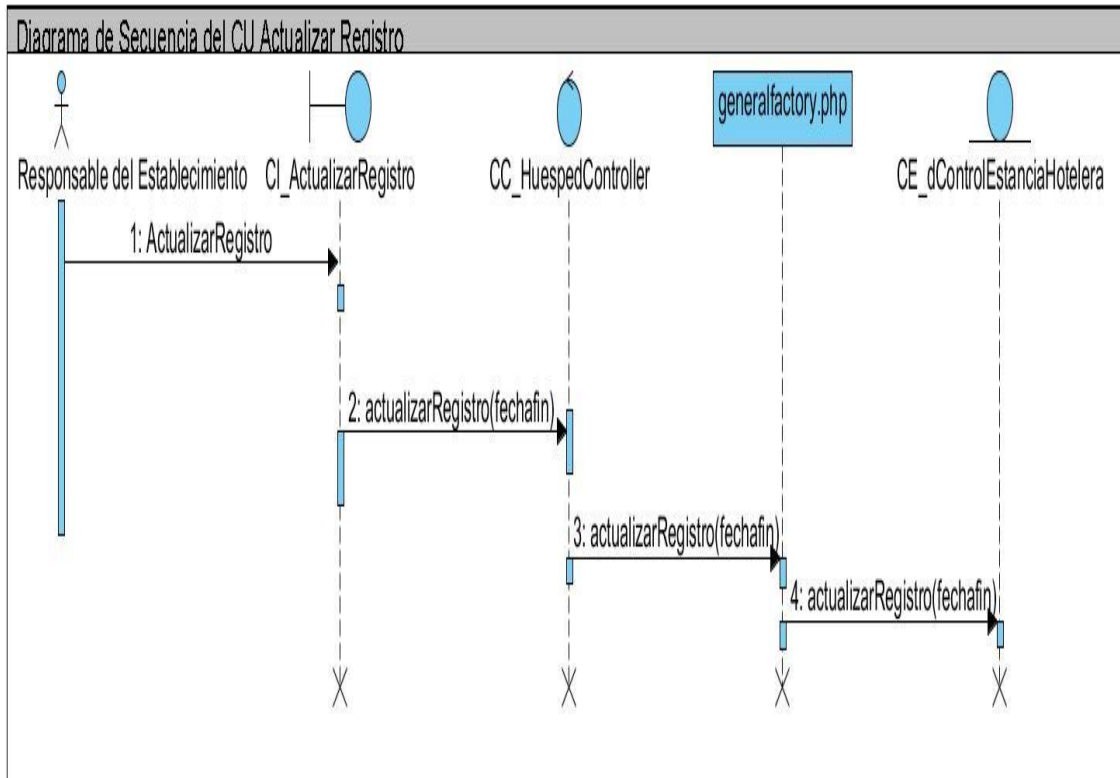
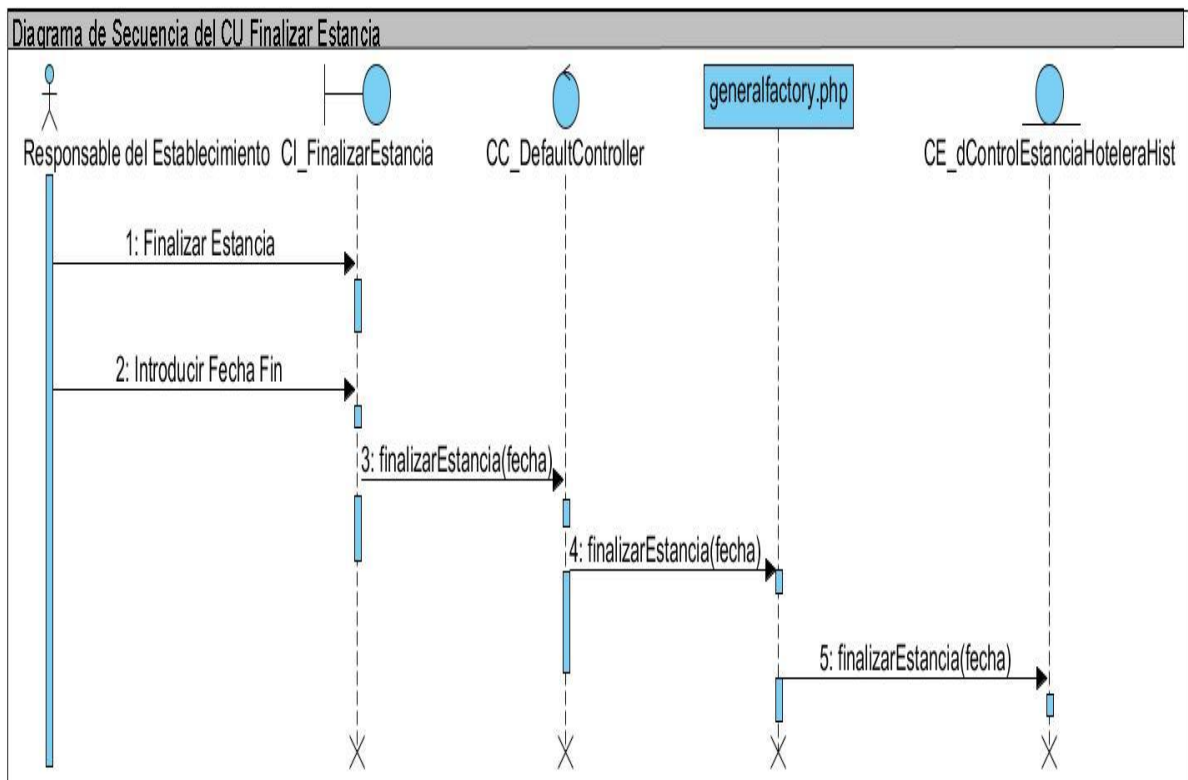


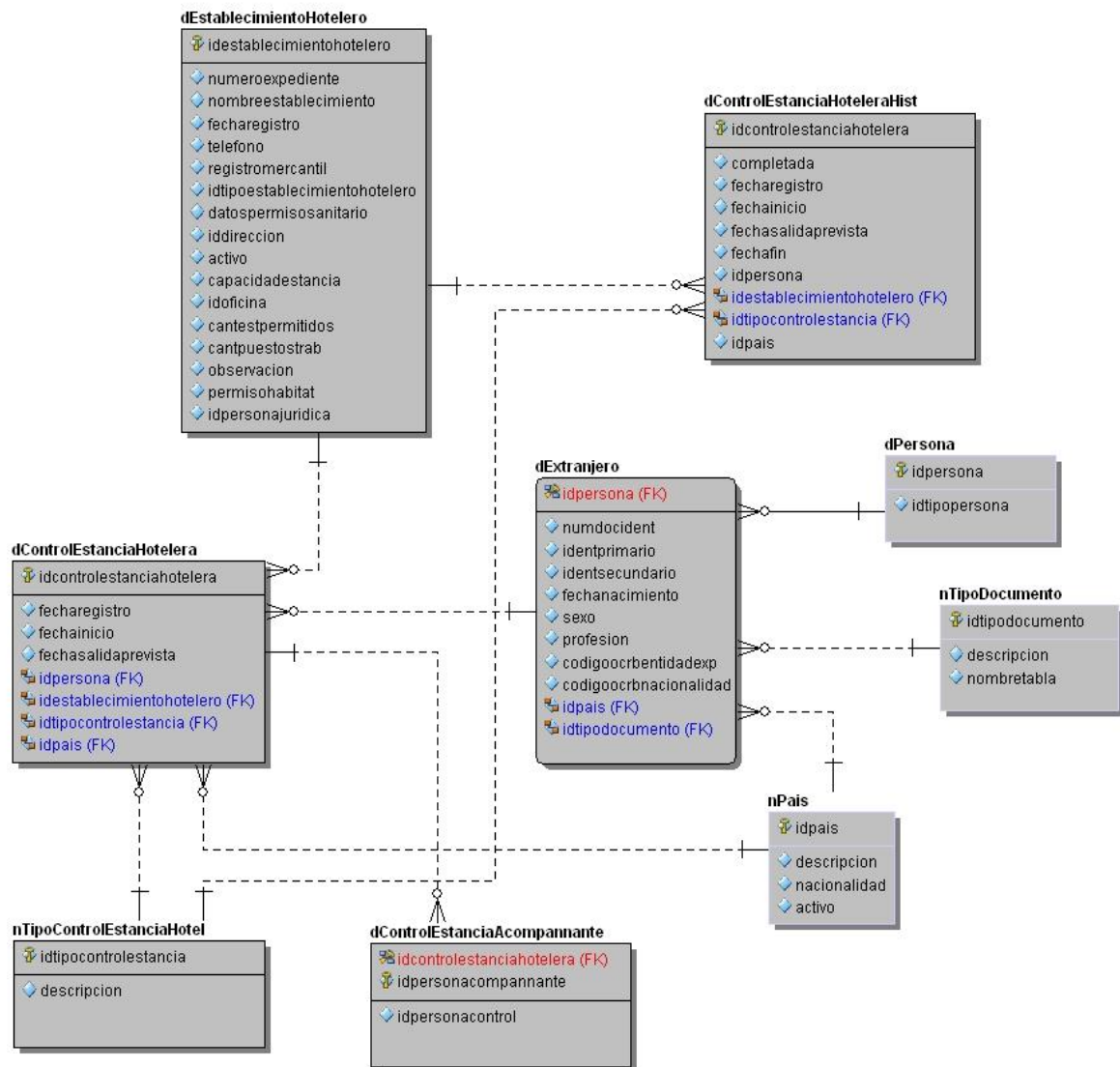
Figura 12 Diagrama de Secuencia CU Finalizar Estancia



2.13. Diseño de la Base de Datos

A continuación se presenta el Modelo Físico de la Base de Datos con el que se describe la implementación de la misma y se obtienen un conjunto de relaciones (tablas) y las restricciones que se deben cumplir sobre ellas.

Figura 13 Modelo Físico de la Base de Datos



2.14. Conclusiones parciales

Con la realización de este capítulo se lograron identificar ocho requisitos funcionales, de ellos tres con prioridad alta, dos con prioridad media y tres con prioridad baja, se realizaron ocho diagramas de clases de análisis, clases del diseño y de interacción con el objetivo de detallar los procesos que se realizan. La arquitectura a utilizar estará guiada por el patrón MCV dividiendo el código de la aplicación en tres capas y los patrones de diseño utilizados será los Patrones GRASP.

Capítulo III: Implementación y Prueba

3.1. Introducción

En el presente capítulo se abordarán todos los temas relacionados con las etapas de implementación y prueba, quedando definidos los diagramas de despliegue y componentes que brindan detalles principales de la distribución del sistema. Con el objetivo de validar la calidad del sistema se aplicarán diferentes tipos de pruebas.

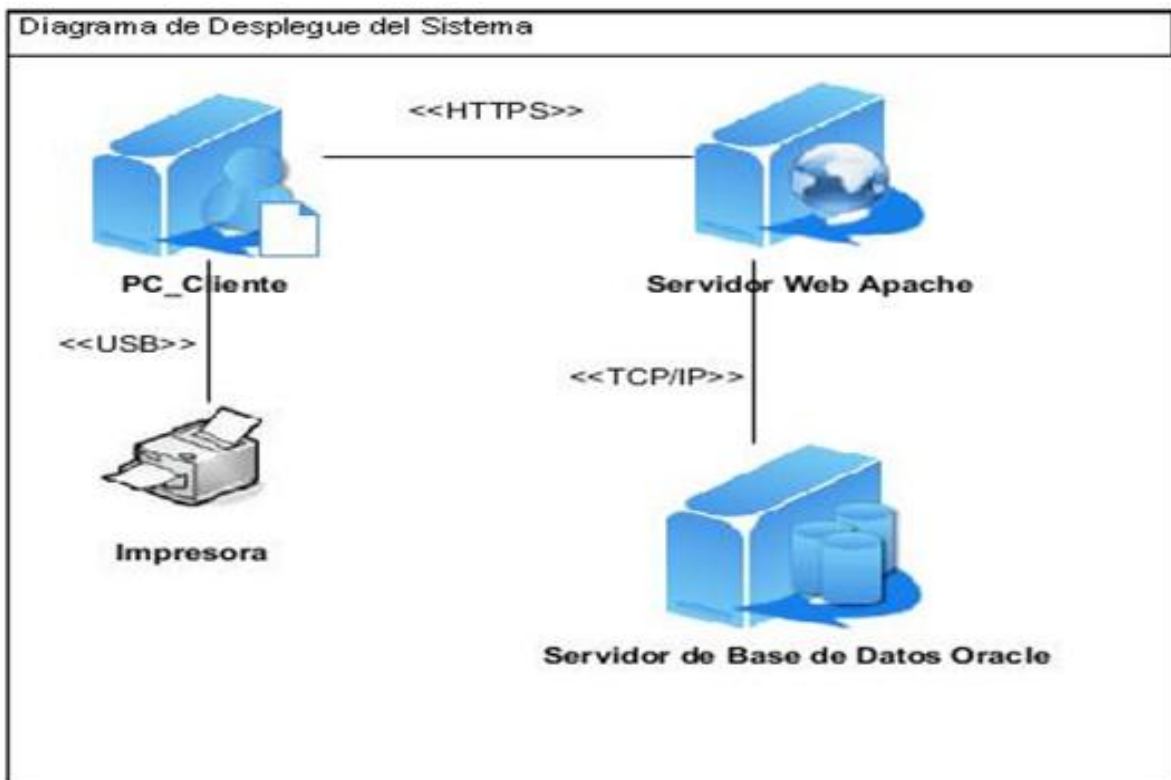
3.2. Implementación

El flujo de trabajo de implementación describe cómo los elementos del modelo del diseño se implementan en términos de componentes y cómo estos se organizan de acuerdo a los nodos específicos en el modelo de despliegue.

3.3. Diagrama de Despliegue

Los diagramas de despliegue son los complementos de los diagramas de componentes que unidos, proveen la vista de implementación del sistema. Describen la topología del sistema, la estructura de los elementos de *hardware* y el software que ejecuta cada uno de ellos. Los diagramas de despliegue representan a los nodos y sus relaciones (Daniele, 2007).

Figura 14 Diagrama de Despliegue

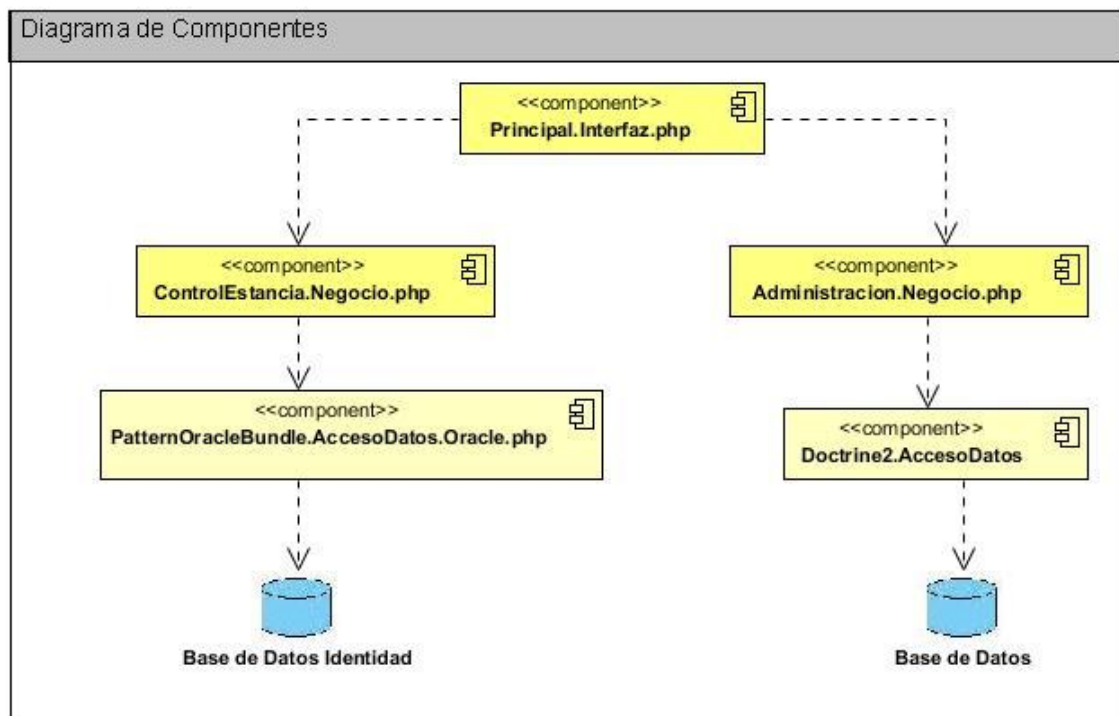


3.4. Diagrama de Componentes

Un diagrama de componentes muestra las dependencias lógicas entre componentes de software, y se relacionan con los diagramas de clases ya que un componente normalmente se corresponde con una o más clases, interfaces o colaboraciones; pero un diagrama de Componentes tiene un nivel más alto de abstracción que un diagrama de clases, usualmente un componente se implementa por una o más clases (u objetos) en tiempo de ejecución.

Componente: un componente puede ser considerado como una unidad autónoma dentro de un sistema o subsistema, tiene una o más interfaces proporcionadas o requeridas, y sus interioridades permanecen ocultas e inaccesibles. Un componente se va modelando a través de todo el ciclo de desarrollo y sucesivamente se va refinando hasta llegar a su implantación y creación de su módulo ejecutable (Ramírez, 2009).

Figura 15 Diagrama de Componentes



3.5. Estándares de Codificación

Los estándares de codificación son pautas de programación que no están enfocadas a la lógica del programa, sino a su estructura y apariencia física para facilitar la lectura, comprensión y mantenimiento del código. También comprende todos los aspectos de la generación del código. Si bien los programadores deben implementar un estándar

Capítulo III: Implementación y Prueba

de forma prudente, éste debe tender siempre a lo práctico. Al comenzar un proyecto de software, es necesario establecer un estándar de codificación para asegurarse de que todos los programadores del proyecto trabajen de forma coordinada (Fernández, 2005).

3.5.1 Convención de nomenclatura

Variables: se rigen por la nomenclatura camelCase¹². Siempre comienzan con minúscula y en caso de nombres compuestos la primera letra de cada palabra comienza con mayúscula.

Figura 16 Estándar de codificación: Convención de nomenclatura de variables

```
1 ....$variable
2 ....$variableNombreCompuesto
```

Constantes: las constantes siempre deben ser en mayúsculas, con caracteres de subrayado “_” para separar palabras en caso de nombres compuestos.

Figura 17 Estándar de codificación: Convención de nomenclatura de constantes

```
1 ....define(CONSTANTE,valor);
2 ....define(CONSTANTE_COMPUESTO,valor);
```

Clases: siempre comienzan con mayúscula, en caso de nombre compuesto las palabras se separan con el caracter subrayado “_” y el resto en minúscula.

Figura 18 Estándar de codificación: Convención de nomenclatura de clases

```
1 ....class Clase
2 ....{
3     ....//BI
4 ....}
5
6 ....class Clase_nombre_compuesto
7 ....{
8     ....//BI
9 ....}
```

¹² Estilo de escritura que se aplica a frases o palabras compuestas.

Capítulo III: Implementación y Prueba

Funciones: se rigen por la nomenclatura camelCase. Siempre comienzan con minúscula y en caso de nombres compuestos la primera letra de cada palabra comienza con mayúscula. Los parámetros son separados por espacio luego de la coma que los separa.

Figura 19 Estándar de codificación: Convención de nomenclatura de funciones

```
1 ....function funcion($parametro1,.$parametro2)
2 ....{
3     ....//BI
4 ....}
5
6 ....function funcionNombreCompuesto($parametro1,.$parametro2)
7 ....{
8     ....//BI
```

En el caso de las funciones en las clases controladoras se especifica el nombre de la acción, todo en minúscula y seguido el sufijo "Action".

Ejemplo:

huespedAction().

Ficheros: todo siempre en minúscula y en caso de nombres compuestos se usa el caracter subrayado "_".

Vistas: intuitivo y relacionado con el formulario y/o vista que representa.

Modelos: con el mismo nombre de la clase que representa.

Librerías: con el mismo nombre de la clase que representa que contiene en el nombre el sufijo "_lib".

Controladoras: con el mismo nombre de la clase que representa y llevan después del nombre la palabra "Controller".

3.5.2 Estructuras de control

Se incluye if, for, foreach, while, switch, entre las estructuras de control y los paréntesis deben tener un espacio. Se recomienda utilizar siempre llaves de apertura y

Capítulo III: Implementación y Prueba

cierre, incluso en situaciones en las que técnicamente son opcionales. Esto aumenta la legibilidad y disminuye la probabilidad de errores lógicos.

Figura 20 Estándar de codificación: Estructuras de control

```
1  ....if.(condicion)
2  ....{
3      ....//BI
4  ....}
5  ....elseif(condicion)
6  ....{
7      ....//BI
8  ....}
9  ....else
```

Figura 21 Estándar de codificación: Estructuras de control

```
10....{
11    ....//BI
12....}
13
14....switch.(valor)
15....{
16    ....case valor1:
17        ....//BI para valor1
18    ....break;
19    ....case valor2:
20        ....//BI para valor2
21    ....break;
22    ....default:
23        ....//BI por defecto
24....}
```

Capítulo III: Implementación y Prueba

Si las condiciones son muy largas que sobrepasan el tamaño de la línea, estas se dividen en varias líneas.

Figura 22 Estándar de codificación: Estructuras de control

```
1 ....if.(condicion1
2 ....|| condicion2)
3 ....||(condicion3
4 ....&& condicion4))
5 ....{
6     ....//BI
7 ....}
```

En el mejor de los casos cuando la condición es muy extensa, se puede dividir esta en variables y compararlas dentro de la estructura de control.

Figura 23 Estándar de codificación: Estructuras de control

```
1 ....$variableCondicion1 = condicion1 || condicion2;
2 ....$variableCondicion2 = condicion3 && condicion4;
3
4 ....if.($variableCondicion1 || $variableCondicion2)
5 ....{
6     ....//BI
7 ....}
```

3.6. Seguridad

La seguridad en cualquier aplicación es un factor que no debe ser ignorado por los desarrolladores. La complejidad de esta en los sistemas informáticos precisa de la preparación de estrategias que permitan que la información circule libremente, garantizando al mismo tiempo la seguridad del uso de los sistemas de información en toda la Comunidad.

Para garantizar el correcto funcionamiento del Sistema Control de Hoteles se establecen como políticas de seguridad: la autenticación de las personas autorizadas, donde se verifica la identidad del usuario. Con el objetivo de garantizar la integridad de la información manejada por el sistema se definen dos tipos de roles para los usuarios (Rol Administrador y Rol Responsable), asignándoles solamente los permisos estrictos

Capítulo III: Implementación y Prueba

de acceso a la información de acuerdo al rol que este desempeñe una vez registrado en la aplicación.

3.7. Proceso de pruebas

Las pruebas son los procesos que permiten verificar y revelar la calidad de un producto de software. Son utilizadas para identificar posibles fallos de implementación, calidad, o usabilidad de un software. En este proceso se ejecutan pruebas dirigidas al sistema de software en su totalidad, con el objetivo de medir el grado en que el software cumple con los requerimientos.

El proceso de pruebas establece varias normas que pueden servir acertadamente como objetivos de las pruebas:

- La prueba es el proceso de ejecución de un programa con la intención de descubrir un error.
- Un buen caso de prueba es aquel que tiene una alta probabilidad de mostrar un error no descubierto hasta entonces.
- Una prueba tiene éxito si descubre un error no detectado hasta entonces (Pressman, 2001).

Nivel de prueba utilizado:

Pruebas de sistemas: un sistema de pruebas implica la operación o implicación del mismo a través de condiciones controladas y la consiguiente evaluación de la información. Las condiciones controladas deben incluir tanto situaciones normales como anormales. El objetivo del sistema de pruebas es encontrar un error para determinar situaciones en donde algo pasa cuando no debe de pasar y viceversa. En una palabra, un sistema de pruebas está orientado a detectar (Moreno, 2004).

Tipo de prueba utilizado:

Pruebas de Funcionalidad: Este tipo de pruebas examina si el sistema cubre sus necesidades de funcionamiento acorde a las especificaciones de diseño. En ellas se debe verificar si el sistema lleva a cabo correctamente todas las funciones requeridas, se debe verificar la validación de los datos y se deben realizar pruebas de comportamiento ante distintos escenarios. Estas pruebas deben estar enfocadas a tareas, a límites del sistema, a condiciones planeadas de error y de exploración. Para estas pruebas usamos los esquemas de pruebas de caja negra ya que nos interesa saber si funciona o no, independientemente de la forma en que lo haga.

Método de prueba utilizado:

Pruebas de caja negra: Las pruebas de caja negra, también denominada prueba de comportamiento, se centran en los requisitos funcionales del software, o sea, la prueba

Capítulo III: Implementación y Prueba

de caja negra le permite al ingeniero del software obtener conjuntos de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. La prueba de caja negra intenta encontrar errores de las siguientes categorías: funciones incorrectas o ausentes, errores de interfaz, errores en estructuras de datos o en accesos a base de datos externas, errores de rendimiento y errores de inicialización y de finalización.

Para desarrollar la prueba de caja negra se usó la siguiente técnica:

Técnica de la Partición de Equivalencia: esta técnica divide el campo de entrada en clases de datos que tienden a ejercitar determinadas funciones del software.

Técnica de la Partición de Equivalencia

Dentro del método de Caja Negra la técnica de la Partición de Equivalencia es una de las más efectivas, pues permite examinar los valores válidos e inválidos de las entradas existentes en el software, descubre de forma inmediata una clase de errores que, de otro modo, requieren la ejecución de muchos casos antes de detectar el error genérico. La partición equivalente se dirige a la definición de casos de pruebas que descubran clases de errores, reduciendo así el número de clases de prueba que hay que desarrollar.

Diseño de casos de prueba:

El Diseño de Casos de Prueba (DCP) es un conjunto de entradas de pruebas, condiciones de ejecución y resultados esperados, desarrollados para cumplir un objetivo. El objetivo del proceso de diseño de casos de prueba es crear un conjunto de casos de prueba que sean efectivos descubriendo defectos en los programas y muestren que el sistema satisface sus requerimientos.

Los casos de prueba deben verificar:

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

A continuación se muestran tres ejemplos de diseño de casos de prueba, para los restantes visitar **Anexo 5 Diseño de casos de prueba.**

Escenario	Descripción	Variable 1 "Establecimiento s"	Variable 2 "Listado de Extranjeros"	Variable 3 "Fecha de Salida"	Respuesta del sistema	Flujo central
-----------	-------------	--------------------------------------	---	---------------------------------	-----------------------	---------------

Capítulo III: Implementación y Prueba

<p>EC 1.1 Actualizar Registro Correctamente.</p>	<p>Se muestra el listado de los huéspedes registrados en el establecimiento y se le actualiza la fecha de fin al seleccionado.</p>	<p>V "Cumberland"</p>	<p>V "Ronaldo Castro"</p>	<p>V "05/06/2012"</p>	<p>Se actualiza el registro al huésped deseado.</p>	<ol style="list-style-type: none"> 1. El usuario se autentica en el Sistema de Control de Hotel. 2. El sistema muestra la pantalla donde se encuentran los módulos Administración y Control de Estancia. 3. El usuario selecciona el módulo "Control de Estancia". 4. El sistema muestra las diferentes funcionalidades. 5. El usuario selecciona "Actualizar Registro". 6. El sistema muestra una pantalla con los establecimientos existentes. 7. El usuario selecciona el establecimiento deseado y presiona el botón "Listar Huéspedes". 8. El sistema muestra el listado de los huéspedes alojados en el establecimiento seleccionado. 9. El usuario selecciona el huésped al que desea
--	--	---------------------------	-------------------------------	---------------------------	---	---

Capítulo III: Implementación y Prueba

						<p>actualizarle el registro y presiona el botón Mostrar Detalles.</p> <p>10. El sistema muestra los datos del huésped seleccionado: "número de documento", "nacionalidad", "nombre y apellidos", "fecha de Nacimiento", "Salida", "Entrada", "Sexo.</p> <p>11. El usuario selecciona la fecha de salida que quiere actualizar y presiona el botón "Actualizar Registro".</p> <p>12. El sistema al huésped con la fecha actualizada.</p>
EC 1.2. Actualizar Registro sin seleccionar el Establecimiento.	No se muestra el listado de los huéspedes alojados en el establecimiento.	I (vacío)	NA	NA	El sistema muestra un mensaje de error "Para listar los huéspedes debe seleccionar el establecimiento".	<p>1. El usuario se autentica en el Sistema Control de Hoteles.</p> <p>2. El sistema muestra la pantalla donde se encuentran los módulos Administración y Control de Estancia.</p> <p>3. El usuario selecciona el módulo "Control de Estancia".</p> <p>4. El sistema muestra las diferentes funcionalidades.</p> <p>5. El usuario selecciona "Actualizar Registro".</p> <p>6. El sistema muestra una pantalla con los</p>

Capítulo III: Implementación y Prueba

						<p>establecimientos existentes.</p> <p>7. El usuario no selecciona el establecimiento y presiona el botón "Listar Huéspedes".</p> <p>8. El sistema muestra un mensaje de error "Para listar los huéspedes debe seleccionar el establecimiento".</p>
EC 1.3. Actualizar Registro sin seleccionar el extranjero.	No se muestra el listado de los huéspedes alojados en el establecimiento ya que el periodo de tiempo no puede estar vacío.	"Cumberland"	I (vacío)	NA	El sistema muestra un mensaje de error "Los campos de las fechas no pueden estar vacíos".	<p>1. El usuario se autentica en el Sistema Control de Hoteles.</p> <p>2. El sistema muestra la pantalla donde se encuentran los módulos Administración y Control de Estancia.</p> <p>3. El usuario selecciona el módulo "Control de Estancia".</p> <p>4. El sistema muestra las diferentes funcionalidades.</p> <p>5. El usuario selecciona Actualizar Registro.</p> <p>6. El sistema muestra una pantalla para seleccionar el establecimiento Hotelero.</p> <p>7. El usuario selecciona el establecimiento Hotelero y presiona el botón Listar Huéspedes.</p> <p>8. El sistema muestra una pantalla con los huéspedes alojados y desactiva el botón Mostrar Detalles hasta que se seleccione un huésped.</p>

Capítulo III: Implementación y Prueba

<p>EC 1.4. Actualizar Registro sin seleccionar la Fecha de Fin.</p>	<p>No se activa el botón Actualizar Registro ya que la Fecha de Fin no se seleccionó.</p>	<p>“Cumberland”</p>	<p>““Ronaldo Castro””</p>	<p>I (vacío)</p>	<p>El sistema desactiva el botón Actualizar Registro.</p>	<ol style="list-style-type: none"> 1. El usuario se autentica en el Sistema Control de Hoteles. 2. El sistema muestra la pantalla donde se encuentran los módulos Administración y Control de Estancia. 3. El usuario selecciona el módulo “Control de Estancia”. 4. El sistema muestra las diferentes funcionalidades. 5. El usuario selecciona Actualizar Registro. 6. El sistema muestra una pantalla para seleccionar el establecimiento Hotelero. 7. El usuario selecciona el establecimiento Hotelero y presiona el botón Listar Huéspedes. 8. El sistema muestra una pantalla con los huéspedes alojados y desactiva el botón Mostrar Detalles hasta que se seleccione un huésped. 9. El usuario selecciona un huésped y presiona el botón Mostrar Detalles. 10. El sistema muestra una pantalla con los datos del huésped, “Nombre”, “Apellidos”, “Sexo”,
---	---	---------------------	---------------------------	------------------	---	---

Capítulo III: Implementación y Prueba

						<p>“Fecha Nacimiento”, “Nacionalidad”, “no. Identificación”, “Entrada”, “Salida”, “Actualizar Registro”. El botón actualizar Registro aparece desactivado hasta que no se introduzca la Fecha de Salida.</p>
EC 1.5.	No se activa el botón Actualizar Registro ya que la Fecha de Fin no se seleccionó.	“Cumberland”	“Ronaldo Castro”	I “17/06/2009”	El sistema muestra un mensaje de error “La Fecha de Salida debe ser Mayor que la Fecha de Entrada”.	<ol style="list-style-type: none"> 1. El usuario se autentica en el Sistema Control de Hoteles. 2. El sistema muestra la pantalla donde se encuentran los módulos Administración y Control de Estancia. 3. El usuario selecciona el módulo “Control de Estancia”. 4. El sistema muestra las diferentes funcionalidades. 5. El usuario selecciona Actualizar Registro. 6. El sistema muestra una pantalla para seleccionar el establecimiento Hotelero. 7. El usuario selecciona el establecimiento Hotelero y presiona el botón Listar Huéspedes. 8. El sistema muestra

Capítulo III: Implementación y Prueba

						<p>una pantalla con los huéspedes alojados y desactiva el botón Mostrar Detalles hasta que se seleccione un huésped.</p> <p>9. El usuario selecciona un huésped y presiona el botón Mostrar Detalles.</p> <p>10. El sistema muestra una pantalla con los datos del huésped, "Nombre", "Apellidos", "Sexo", "Fecha Nacimiento", "Nacionalidad", "no. Identificación", "Entrada", "Salida", "Actualizar Registro". El botón actualizar Registro aparece desactivado hasta que no se introduzca la Fecha de Salida.</p> <p>11. El usuario selecciona la Fecha de Fin.</p> <p>12. El sistema muestra el mensaje "La fecha de Salida debe ser Mayor que la Fecha de Entrada".</p>
--	--	--	--	--	--	--

Tabla 9 Diseño de Casos de Prueba basados en requisitos: Actualizar Registro

Escenario	Descripción	Variable 1 "Establecimiento Hotelero"	Variable 2 "Listado de Extranjeros"	Respuesta del sistema	Flujo central
-----------	-------------	--	--	-----------------------	---------------

Capítulo III: Implementación y Prueba

<p>EC Finalizar estancia correctamente.</p>	<p>1.1 Se muestra el listado de estancias y se selecciona la que se quiere finalizar.</p>	<p>“Cumberland”</p>	<p>“Ronaldo Castro”</p>	<p>Muestra el listado de estancias existentes y finaliza la que se seleccionó.</p>	<ol style="list-style-type: none"> 1. El usuario se autentica en el Sistema Control de Hoteles. 2. El sistema muestra la pantalla donde se encuentran los módulos Administración y Control de Estancia. 3. El usuario selecciona el módulo “Control de Estancia”. 4. El sistema muestra las diferentes funcionalidades. 5. El usuario selecciona “Finalizar estancia”. 6. El sistema muestra una pantalla para seleccionar el establecimiento Hotelero donde se encuentra el huésped que se le quiere finalizar la estancia. 7. El usuario selecciona el Establecimiento Hotelero y presiona el botón Listar Huéspedes. 8. El sistema muestra el listado de todos los huéspedes alojados en el establecimiento. 9. El usuario selecciona el huésped al que desea finalizarle la estancia y presiona el botón Finalizar Estancia. 10. El sistema muestra el mensaje “Estancia Finalizada con éxito”
---	---	---------------------	-------------------------	--	--

Capítulo III: Implementación y Prueba

<p>EC Finalizar estancia sin seleccionar el Establecimiento Hotelero.</p>	<p>1.2 No se muestra el listado de estancias ya que el campo no puede estar vacío.</p>	<p>I (vacío)</p>	<p>NA</p>	<p>El sistema muestra un mensaje de error” Para listar los huéspedes debe seleccionar el establecimiento”.</p>	<ol style="list-style-type: none"> 1. El usuario se autentica en el Sistema Control de Hoteles. 2. El sistema muestra la pantalla donde se encuentran los módulos Administración y Control de Estancia. 3. El usuario selecciona el módulo “Control de Estancia”. 4. El sistema muestra las diferentes funcionalidades. 5. El usuario selecciona “Finalizar estancia” 6. El sistema muestra una pantalla para seleccionar el establecimiento Hotelero donde se encuentra el huésped al que se le quiere finalizar la estancia. 7. El usuario presiona el botón Listar Huésped si seleccionar el Establecimiento. 8. El sistema muestra un mensaje de error” Para listar los huéspedes debe seleccionar el establecimiento”.
---	--	------------------	-----------	--	---

Tabla 10 Diseño de Casos de Prueba basados en requisitos: Finalizar Estancia

Capítulo III: Implementación y Prueba

Escenario	Descripción	Variable 1 "Nombre"	Variable 2 "Apellidos"	Variable 3 "Cédula"	Variable 4 "Fecha de nacimiento"	Variable 5 "usuario"	Variable 6 "Contraseña"	Variable 7 "Rol"	Respuesta del sistema	Flujo central
EC 1.1 Adicionar Usuario correctamente.	Se introducen los datos del usuario y queda añadido en la base de datos.	V "Pedro"	V "Macheco"	V "LM12345"	V "1990/12/14"	V "pmarc heco"	V "Pedro123"	V "Administrador"	Autentica el usuario permitiéndole utilizar el sistema.	<ol style="list-style-type: none"> 1. El usuario se autentica en el sistema con el rol administrador. 2. El sistema muestra una pantalla donde se encuentran los módulos administración y Control de Estancia. 3. El usuario selecciona el módulo Administración. 4. El sistema muestra una pantalla con las diferentes funcionalidades. 5. El usuario selecciona la funcionalidad Adicionar Usuario. 6. El sistema muestra una interfaz con los campos para adicionar un usuario. "Nombre", "Apellidos" "Cédula" "fecha Nacimiento" "usuario" "Contraseña" "Rol".

Capítulo III: Implementación y Prueba

										7. El usuario llena los campos y presiona el botón Adicionar. 8. El sistema adiciona el usuario y muestra el mensaje "El Usuario ha Sido Añadido con Éxito"
EC 1.2 Adicionar Usuario dejando campos vacíos.	No se introducen los datos del usuario.	I (vacío)	V "Macheco"	V "LM1 2345"	V "1990/12/14"	V "pmarheco"	V "Pedro123"	V "Administrador"	El sistema muestra un mensaje de error "Existen Campos Vacíos".	1. El usuario se autentica en el sistema con el rol administrador. 2. El sistema muestra una pantalla donde se encuentran los módulos administración y Control de Estancia. 3. El usuario selecciona el módulo Administración. 4. El sistema muestra una pantalla con las diferentes funcionalidades. 5. El usuario selecciona la
		V "Pedro"	I (vacío)	V "LM1 2345"	V "1990/12/14"	V "pmarheco"	V "Pedro123"	V "Administrador"		
		V "Pedro"	V "Macheco"	I (vacío)	V "1990/12/14"	V "pmarheco"	V "Pedro123"	V "Administrador"		
		V "Pedro"	V "Macheco"	V "LM1 2345"	I (vacío)	V "pmarheco"	V "Pedro123"	V "Administrador"		
		V "Pedro"	V "Macheco"	V "LM1 2345"	V "1990/12/14"	I (vacío)	V "Pedro123"	V "Administrador"		
		V "Pedro"	V "Macheco"	V "LM1 2345"	V "1990/12/14"	V "pmarheco"	I (vacío)	V "Administrador"		

Capítulo III: Implementación y Prueba

		V "Pedro"	V "Macheco"	V "LM12345"	V "1990/12/14"	V "pmarheco"	V "Pedro123"	I (vacío)		<p>funcionalidad Adicionar Usuario.</p> <p>6. El sistema muestra una interfaz con los campos para adicionar un usuario. "Nombre", "Apellidos" "Cédula" "fecha Nacimiento" "usuario" "Contraseña" "Rol".</p> <p>7. El usuario llena los campos dejando algunos vacíos y presiona el botón Adicionar.</p> <p>8. El sistema muestra el mensaje de error "Existen Campos Vacíos".</p>
EC 1.3	Se Introduce los datos del usuario de manera incorrecta.	I "Pfsdfd"	V "Macheco"	V "LM12345"	V "1990/12/14"	V "pmarheco"	V "Pedro123"	V "Administrador"	El sistema muestra un mensaje de error "Datos Incorrectos."	1. El usuario se autentica en el sistema con el rol administrador.
Adicionar Usuario introduciendo datos incorrectos.	V "Pedro"	I "sadasd"	V "LM12345"	V "1990/12/14"	V "pmarheco"	V "Pedro123"	V "Administrador"	2. El sistema muestra una pantalla donde se encuentran los módulos administración		
		V "Pedro"	V "Macheco"	I "425879"	V "1200/05/25"	V "pmarheco"	V "Pedro123"	V "Administrador"		

Capítulo III: Implementación y Prueba

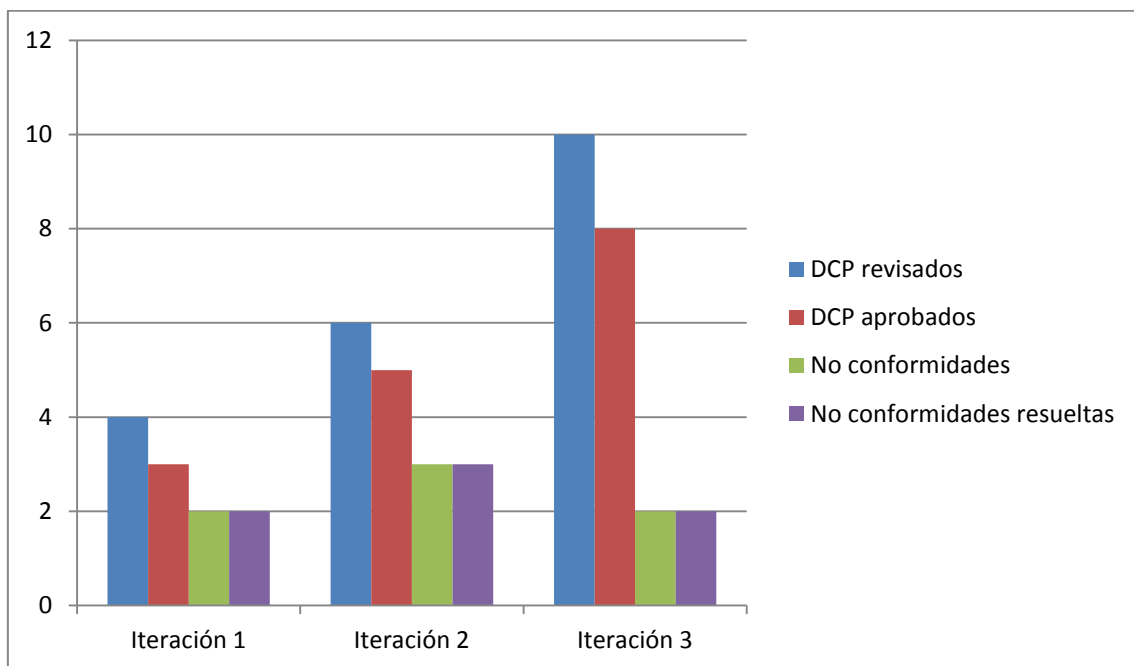
		V "Pedro "	V "Mache co"	V "LM1 2345"	I "1990/ 12/14"	V "pmarch eco"	V "Pedro 123"	V "Admi nistra dor"		y Control de Estancia.
		V "Pedro "	V "Mache co"	V "LM1 2345"	V "1990/ 12/14"	I "etaver az"	V "Pedro 123"	V "Admi nistra dor"		3. El usuario selecciona el módulo Administración
		V "Pedro "	V "Mache co"	V "LM1 2345"	V "1990/ 12/14"	V "pmarc heco"	I "prue ba"	V "Admi nistra dor"		4. El sistema muestra una pantalla con las diferentes funcionalidades.
		V "Pedro "	V "Mache co"	V "LM1 2345"	V "1990/ 12/14"	V "pmarc heco"	V "Pedro 123"	I "usua rio"		5. El usuario selecciona la funcionalidad Adicionar Usuario.
										6. El sistema muestra una interfaz con los campos para adicionar un usuario. "Nombre", "Apellidos" "Cédula" "fecha Nacimiento" "usuario" "Contraseña" "Rol".
										7. El usuario llena los campos introduciendo alguno mal y presiona el botón Adicionar.
										8. El sistema muestra el mensaje de error "Datos Incorrectos".

Tabla 11 Diseño de Casos de Prueba basados en requisitos: Adicionar Usuario

3.8. Resultado de las pruebas

Para la comprobación del correcto funcionamiento del sistema se realizaron 10 Diseños de Casos de Prueba (DCP). En la primera iteración de pruebas de cuatro DCP analizados se detectaron dos no conformidades, a las cuales se le dio solución en su totalidad. En la segunda iteración de pruebas de seis DCP analizados se registraron tres no conformidades, a las cuales se les dio solución. Para la tercera iteración de 10 DCP analizados (totalidad del sistema implementado), se detectaron dos no conformidades las cuales fueron resueltas. A continuación se muestra la figura 24 con estos resultados:

Figura 24 Pruebas del Sistema



3.9. Conclusiones parciales

A lo largo de este capítulo se desarrollaron los diagramas de despliegue y de componentes, los cuales permiten visualizar la forma en que estará distribuida la aplicación para su utilización, se definieron los estándares de codificación utilizados en la implementación, el proceso de pruebas permitió identificar ocho no conformidades las cuales fueron corregidas en su totalidad, quedando de esta manera el sistema listo para su despliegue.

Conclusiones

Al término de la investigación se pueden arribar a las siguientes conclusiones:

- Las soluciones implementadas a nivel mundial relacionadas con la gestión hotelera, no se adecúan al negocio existente en los establecimientos de la República Bolivariana de Venezuela.
- El proceso de desarrollo estuvo guiado por RUP siguiendo la línea de trabajo del SRHEE, utilizando a UML como lenguaje de modelado para la modelación de los artefactos generados.
- La solución fue desarrollada con el *framework* Symfony a partir de los beneficios que aporta al desarrollo de aplicaciones web utilizando PHP como lenguaje de programación.
- Se identificaron ocho requisitos funcionales, de ellos cinco con prioridad alta, dos con prioridad media y uno con prioridad baja.
- La arquitectura utilizada estuvo guiada por el patrón MCV dividiendo el código de la aplicación en tres capas y los patrones de diseño utilizados son los Patrones GRASP.
- Se desarrollaron los diagramas de despliegue y de componentes, los cuales permitieron visualizar la forma en que estará distribuida la aplicación para su utilización.
- Se utilizaron los estándares de codificación según el estilo camelCase para la implementación, el proceso de pruebas permitió identificar siete no conformidades las cuales fueron corregidas en su totalidad.

Recomendaciones

- Mejorar el proceso de registro de huéspedes pues la solución no permite modificar los datos en caso de que sean erróneos.
- Integrar los Módulos Administración y Control de Estancia al Módulo de Oficinas.

Referencias bibliográficas

Altamirano, Alfonso Valdés. 2008. IGNETWORK. [Online] agosto 23, 2008. [Cited: Enero 19, 2012.] <http://foro.ignetwork.net/showthread.php?15188-IDE-Entorno-integrado-de-desarrollo-%28Concepto-importante%29>.

Álvarez, Sara. 2007. desarrolloweb.com. [Online] Julio 31, 2007. [Cited: Enero 7, 2012.] <http://www.desarrolloweb.com/articulos/sistemas-gestores-bases-datos.html>.

AMSystem. 2010. [Online] 2010. [Cited: Diciembre 6, 2011.] <http://www.amsystem.es>.

Análisis y diseño del módulo de Postgrado en Akademos v2.0. **Yosvaldy Fernández Fernández, Omar Jesus Martínez. 2008.** La Habana : s.n., 2008.

Brandendaugh, Jerry. 2000. *Programación de aplicaciones Web con JavaScript.* Madrid : EDICIONES ANAYA MULTIMEDIA, 2000. 84-41 5-1 070-9.

Chávez, Mallelín Bolufé. 2009. plusformacion.com. [Online] Mayo 2009. [Cited: Diciembre 14, 2011.] <http://www.plusformacion.com/Recursos/r/Frameworks-para-desarrollo-aplicaciones-PHP>.

Clements, Paul. 2003. *Software Architecture in Practice.* 2003.

Daniele, Ing. Marcela. 2007. *El Arte de Modelar.* 2007.

Diario Portafolio. 2011. [Online] Abril 6, 2011. [Cited: Diciembre 3, 2011.] <http://www.derechos.org>.

Diseño e implementación del módulo de Gestión de indicadores del proyecto del Ministerio del Poder Popular de la Comunicación y la Información. **Abelardo López Quintana, Raúl Romero Leyva. 2010.** La Habana : s.n., 2010.

DNM, Dirección General de Migración Guatemala. 2003. *Sistema de Información y Control Migratorio SICOMGUA.* Guatemala : s.n., 2003.

DNM, Dirección Nacional de Migraciones Argentina. 2011. Dirección Nacional de Migraciones. [Online] 2011. [Cited: Diciembre 2, 2011.] <http://www.sicam.gov.ar/>.

Española, Real Academia. 2009. Real Academia Española. *Real Academia Española.* [Online] 2009. [Cited: Noviembre 27, 2011.] <http://www.rae.es>.

Fernández, Giovanni. 2005. *Estándares de codificación DOTNET.* 2005.

Figueroa, Domecq. 2006. La Conectividad entre las Tics para la Gestión de la Información en el Sector Hotelero. [Online] 2006. <http://www.turismo.uma.es>.

Fuster, Gonzalo Génova. 2006. *Evaluación comparativa de herramientas CASE para UML desde el punto de vista notacional.* Madrid : s.n., 2006. 0211-2124.

González, Benjamin. 2009. Zend Framework Manual en español. [Online] noviembre 14, 2009. [Cited: Diciembre 13, 2011.] <https://manual.zfdes.com>.

Hernán, Schenone Marcelo. 2004. *Diseño de una Metodología Ágil de Desarrollo de Software*. Buenos Aires : s.n., 2004.

Herranz, Raul. 2010. Utópica Informática. [Online] Diciembre 13, 2010. [Cited: Diciembre 16, 2011.] <http://www.utopicainformatica.com/2010/12/lenguajes-de-modelado.html>.

Ivar Jacobson, Grady Booch, James Rumbaugh. 2000. *El Proceso Unificado de Desarrollo de Software*. Madrid : Pearson Education.S.A, 2000. 84-7829-036-2.

Kemedy, Chuck Musciano y Bill. 2004. *HTML La Guía Completa*. México : McGRAW-HILL INTERAMERICANA EDITORES, S.A. de C.V., 2004. 1-56592-235.

Larman, Craig. 1999. UML y patrones. Primera edición, 1999.

Monso, Martín Ramos. 2005. *Programación PHP Sitios Web Dinámicos e Interactivos*. Buenos Aires : MP Ediciones SA, 2005. 03730.

Mora, Sergio Luján. 2002. *Programación de aplicaciones web: historia, principios básicos y clientes web*. s.l. : Editorial Club Universitario., 2002.

—. **2001.** *Programación en Internet: Clientes Web (1ª edición)*. s.l. : Editorial Club Universitario., 2001.

Moreno, José Luis. 2004. *Sistema de pruebas capítulo5*. 2004.

Ordóñez, Álvaro Zabala. 2004. *Patrones de Diseño. Un curso practico [I]*. s.l. : Sólo Programadores, 2004.

Pressman, Roger. 2002. *Ingeniería de Software, un enfoque práctico*. s.l. : S.I. : McGraw-Hill Companies, 2002. ISBN: 8448132149..

Pressman, Roger S. 2001. *Ingeniería de software un enfoque práctico*. Madrid : s.n., 2001.

Ramírez, Lic. Elisa Arizaca. 2009. *Análisis y diseño de sistemas II*. Bolivia : s.n., 2009.

Rojas, Jose. 2007. slideshare. [Online] 2007. [Cited: Diciembre 13, 2011.] <http://www.slideshare.net/jrojas/tema1-lenguajes-de-programacion>.

Segura, María Isabel Sánchez. 2011. *Desarrollo Orientado a Objetos con UML*. 2011.

Sistema para la informatización del proceso del Control de Estancias en Hoteles para la Dirección de Migración y Fronteras de la República Bolivariana de Venezuela. **Peláez, Yadira Lázara Rodríguez. 2008.** la Habana : s.n., 2008.

Solis, Manuel Calero. 2010. *Una explicación de la programación extrema (XP)*. 2010.

Vega, Miguel. 2010. Casos de uso UML. Granada : s.n., 2010.

Glosario de Términos

SAIME: Servicio Administrativo de Identificación, Migración y Extranjería.

Huésped:

(Del lat. *hospes*, *-ītis*).

1. m. y f. Persona alojada en casa ajena.
2. m. y f. Persona alojada en un establecimiento de hostelería.
3. m. *Biol.* Vegetal o animal en cuyo cuerpo se aloja un parásito.
4. m. p. us. Mesonero o amo de posada.
5. m. p. us. Persona que hospeda en su casa a otra. (Española, 2009)

Hotel:

Un hotel es un establecimiento que se dedica al alojamiento de huéspedes o viajeros. El término proviene del vocablo francés *hôtel*, que hace referencia a una “casa adosada”.

Estancia:

Una estancia es la acción de estar determinado tiempo en un lugar, permanencia en algún sitio durante cierto tiempo.

Anexos

Anexo 1 Descripciones de Casos de Uso

Caso de Uso:	Emitir Reportes
Actores:	Responsable (inicia).
Resumen:	El Caso de Uso inicia cuando el responsable selecciona la opción Mostrar Reportes para obtener una serie de listado de interés.
Precondiciones:	El responsable debe estar autenticado.
Referencias	RF7, RF12, RF14, RF15
Prioridad	Primario.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El Caso de Uso se inicia cuando el Responsable selecciona la opción “Mostrar Reportes”.	2. El sistema muestra una interfaz con todos los reportes existentes. 2.a. Cantidad de huéspedes registrados desde el último pago realizado. 2.b. Huéspedes registrados en el establecimiento en un período de tiempo. 2.c. Huéspedes registrados en el establecimiento en una fecha determinada. 2.d. Huéspedes que culminaron su estancia en el establecimiento en una fecha determinada.
3. El Responsable selecciona el reporte que desea ver. 3.a. Cantidad de huéspedes	4. Según su selección: 4.a. Si seleccionó “Cantidad de huéspedes registrados desde el

<p>registrados desde el último pago realizado.</p> <p>3.b. Huéspedes registrados en el establecimiento en un período de tiempo.</p> <p>3.c. Huéspedes registrados en el establecimiento en una fecha determinada.</p> <p>3.d. Huéspedes que culminaron su estancia en el establecimiento en una fecha determinada.</p>	<p>último pago realizado” muestra un mensaje “La cantidad de huéspedes registrados desde el último pago es: x”.</p> <p>4.b. Si seleccionó “Huéspedes registrados en el establecimiento en un período de tiempo” muestra un listado con todos los huéspedes hospedados en dicho período.</p> <p>4.c. Si seleccionó “Huéspedes registrados en el establecimiento en una fecha determinada” muestra un listado con todos los huéspedes hospedados en la fecha referida.</p> <p>4.d. Si seleccionó “Huéspedes que culminaron su estancia en el establecimiento en una fecha determinada” muestra un listado con todos los huéspedes que finalizaron su estancia en la fecha referida.</p>
5. El Responsable presiona “Aceptar”.	6. Continúa con el paso 2 del Flujo Normal.
Prototipo de Interfaz	
Poscondiciones éxito	Se actualizó el registro del huésped correctamente.
Poscondiciones fallo	No se actualizó el registro del huésped.

Tabla 12 Descripción CU Emitir Reportes

Caso de Uso:	Autenticar usuario.
Actores:	Usuario.
Resumen:	El Caso de Uso inicia cuando el sistema muestra la interfaz

	correspondiente para autenticar usuario y así ser registrados en el sistema. El usuario se autentica y finaliza el Caso de Uso con el registro del usuario en el sistema.
Precondiciones:	-
Referencias	RF-1, RF-2.
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
	<ol style="list-style-type: none"> 1. El Caso de Uso se inicia cuando el sistema muestra la interfaz para autenticar usuario con las opciones: <ol style="list-style-type: none"> 1. a. Campo "Usuario". 1. b. Campo "Contraseña". 1. c. Opción "Aceptar". 1. d. Opción "Cancelar".
<ol style="list-style-type: none"> 3. El usuario introduce los datos correspondientes a : <ol style="list-style-type: none"> 2. a. Usuario. 2. b. Contraseña. 3. El usuario selecciona una de las siguientes opciones: <ol style="list-style-type: none"> 3. a. "Aceptar". 3. b. "Cancelar". 	<ol style="list-style-type: none"> 4. Según su elección: <ol style="list-style-type: none"> 4. a. Si el usuario selecciona "Aceptar" verifica que los datos estén correctos sino continúa con el Flujo Alternativo 1. 4. b. Si el usuario selecciona "Cancelar" no se autentica en el sistema y continúa con el paso 1 del Flujo Normal de Eventos.
Flujo Alternativo 1: "Datos Incorrectos"	
Acción del Actor	Respuesta del Sistema

	<p>1.1. En caso de estar incorrectos los datos el sistema muestra:</p> <p>1.1.a El mensaje: “Debe introducir los datos correctamente”.</p> <p>1.1.b Opción “Aceptar”.</p>
1.2. El usuario selecciona la opción “Aceptar”.	2.1. El sistema muestra la interfaz con los campos como han sido llenados y permite corregirlos, continúa en el paso 1 del Flujo Normal.
Poscondiciones éxito	Se hizo efectivo el registro de usuarios en el sistema.
Poscondiciones fallo	No se hizo efectivo el registro de usuarios en el sistema.

Tabla 13 Descripción CU Autenticar usuario

Anexo 2 Modelos de Clases del Análisis

Figura 25 Diagrama de Clases del Análisis CU Emitir Reportes

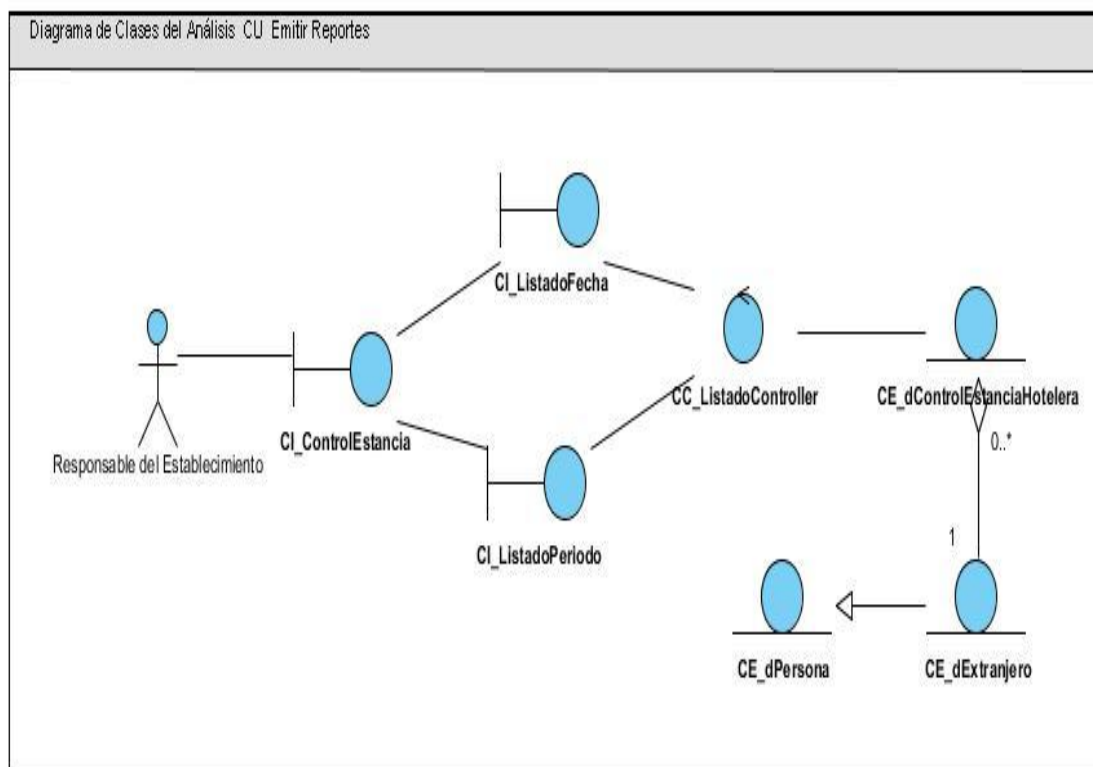


Figura 26 Diagrama de Clases del Análisis CU Gestionar Usuario

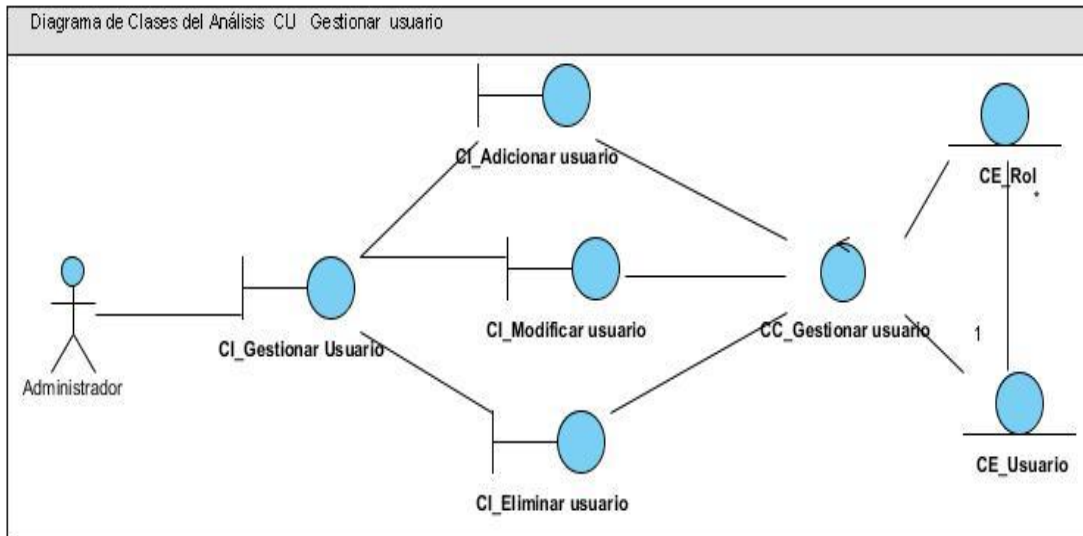
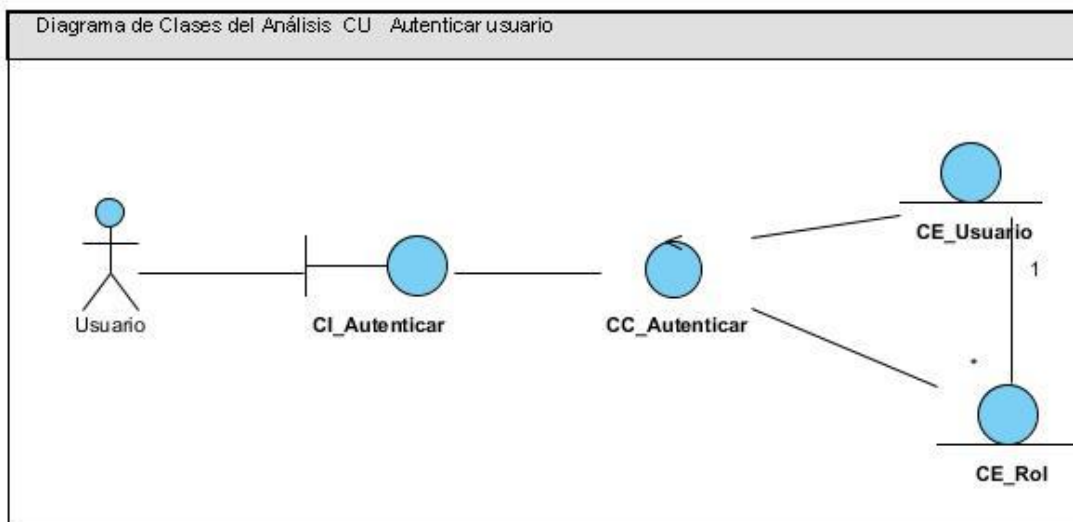


Figura 27 Diagrama de Clases del Análisis CU Autenticar Usuario



Anexo 3 Modelos de Clases del Diseño

Figura 28 Diagrama de Clases del Diseño CU Buscar Persona

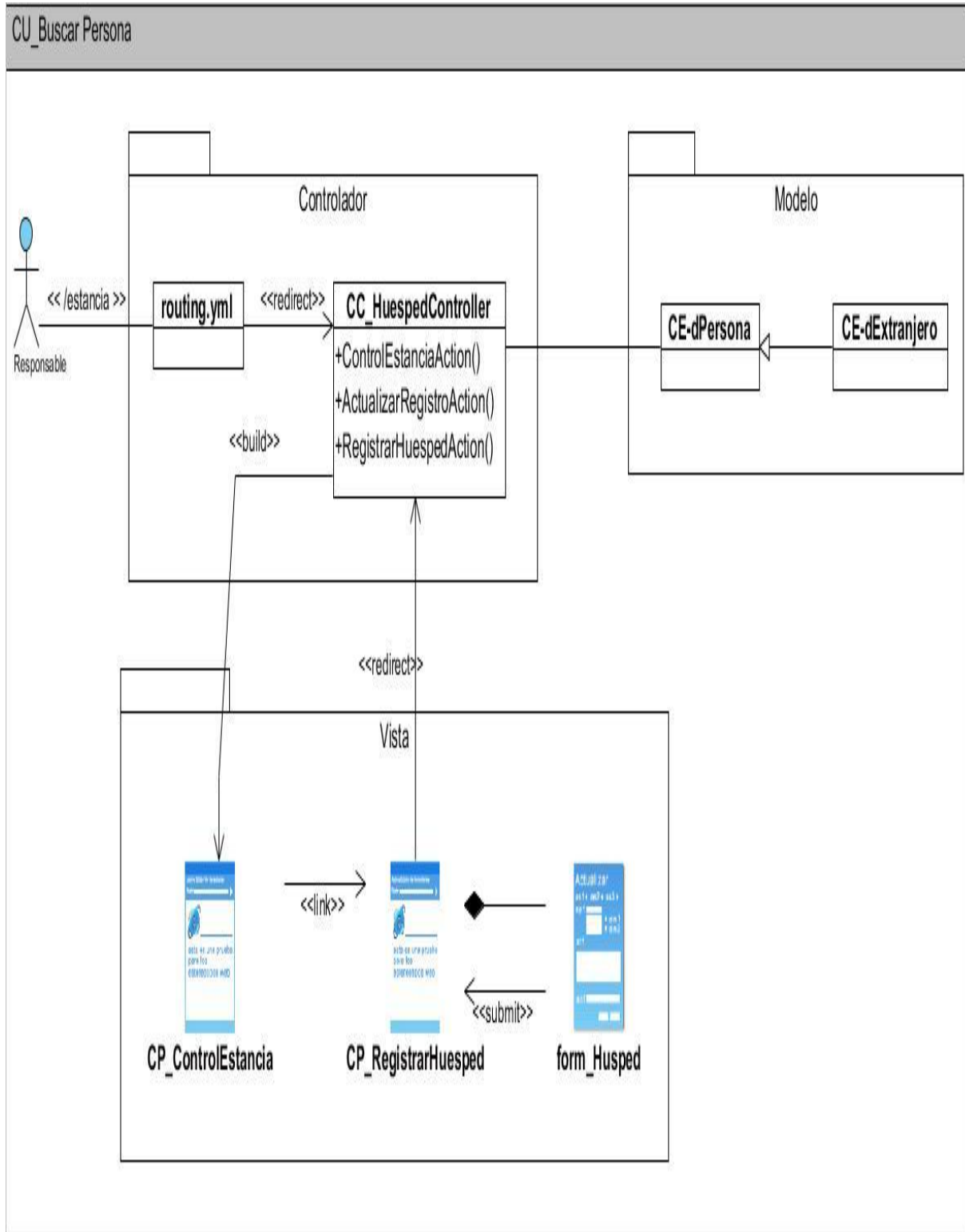


Figura 29 Diagrama de Clases del Diseño CU Finalizar Estancia

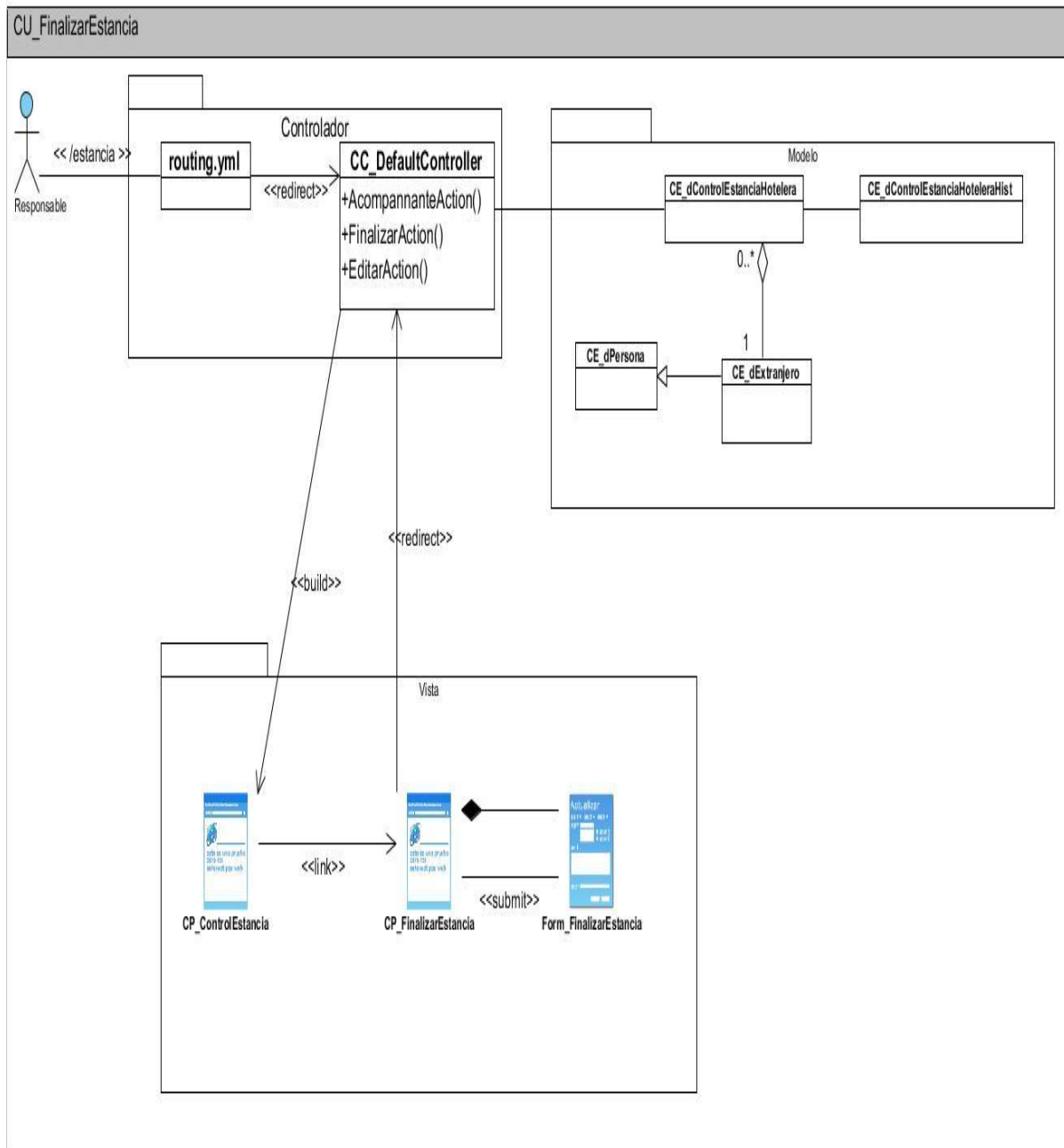


Figura 30 Diagrama de Clases del Diseño CU Registrar Acompañante

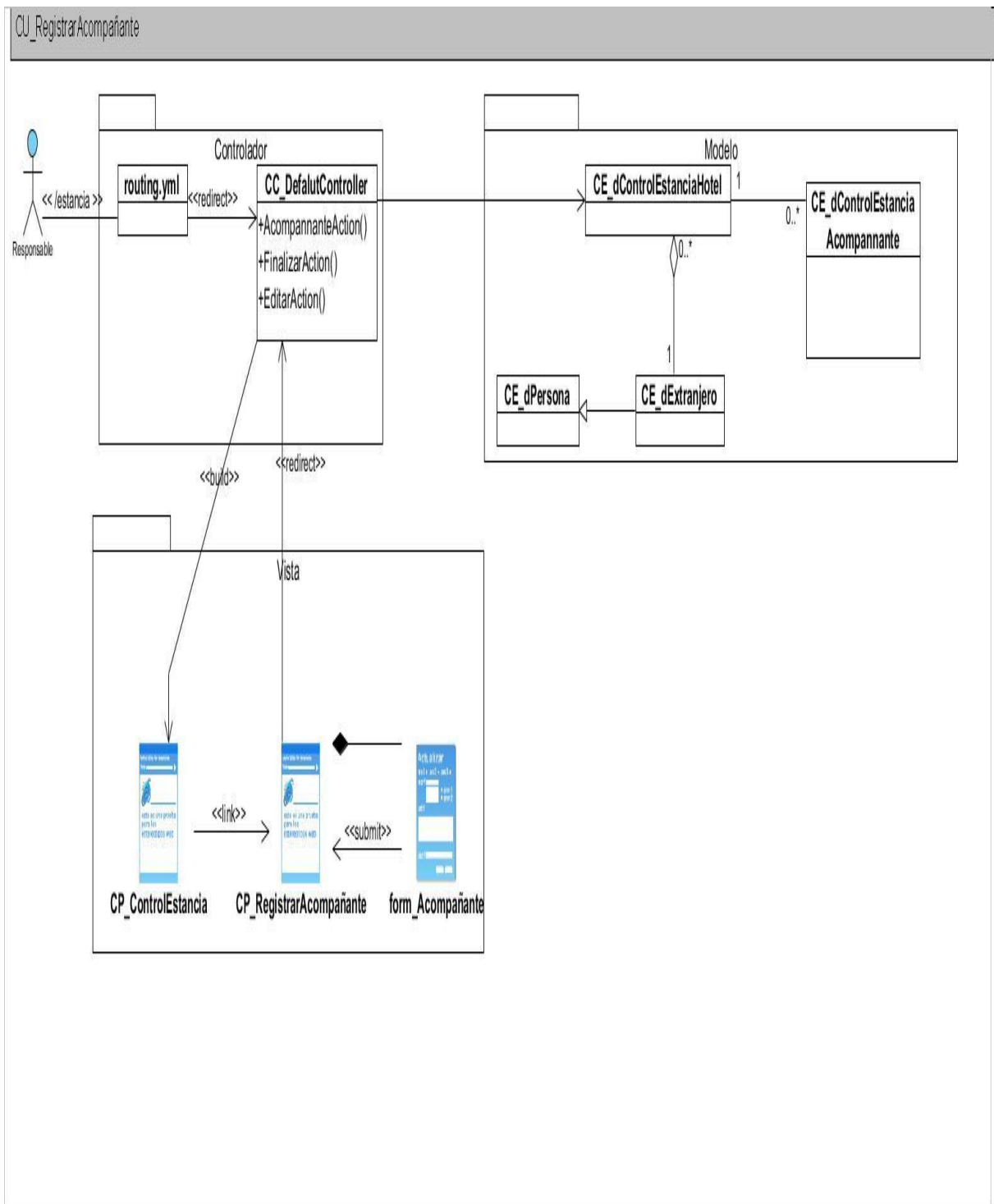


Figura 31 Diagrama de Clases del Diseño CU Autenticar Usuario

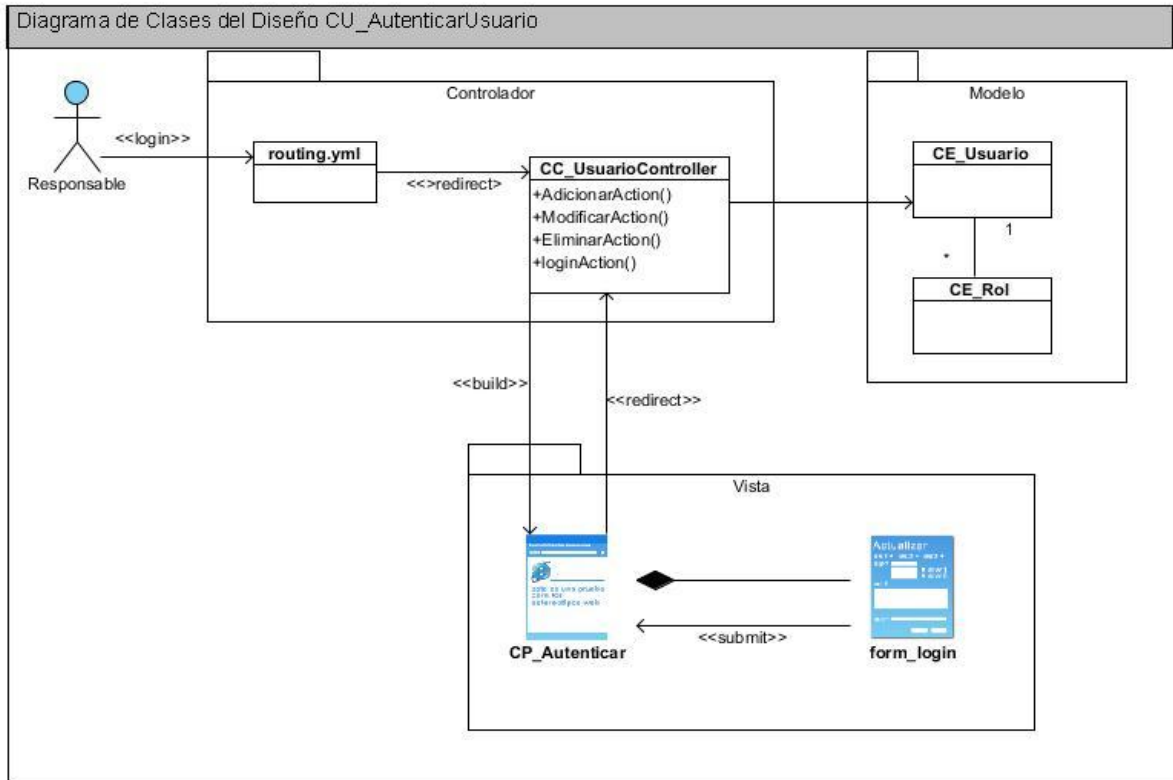
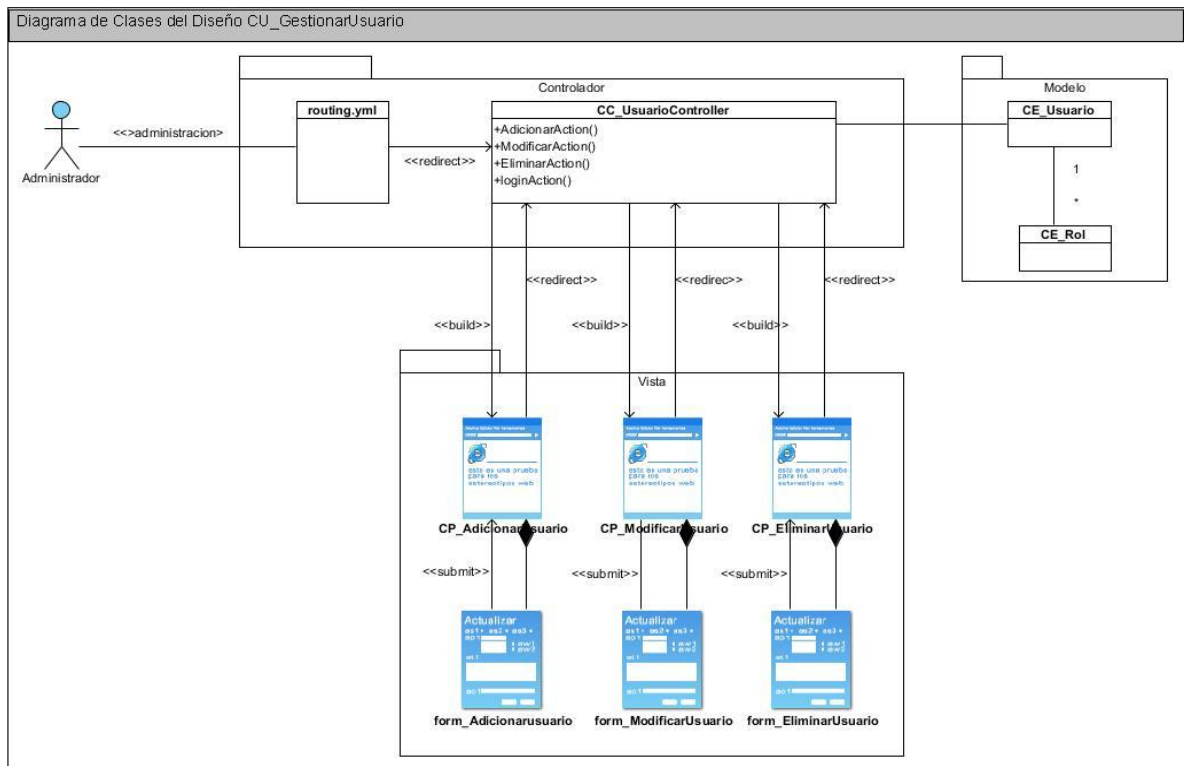


Figura 32 Diagrama de Clases del Diseño CU Gestionar Usuario



Anexo 4 Diagramas de interacción

Figura 33 Diagrama de Secuencia CU Buscar Persona

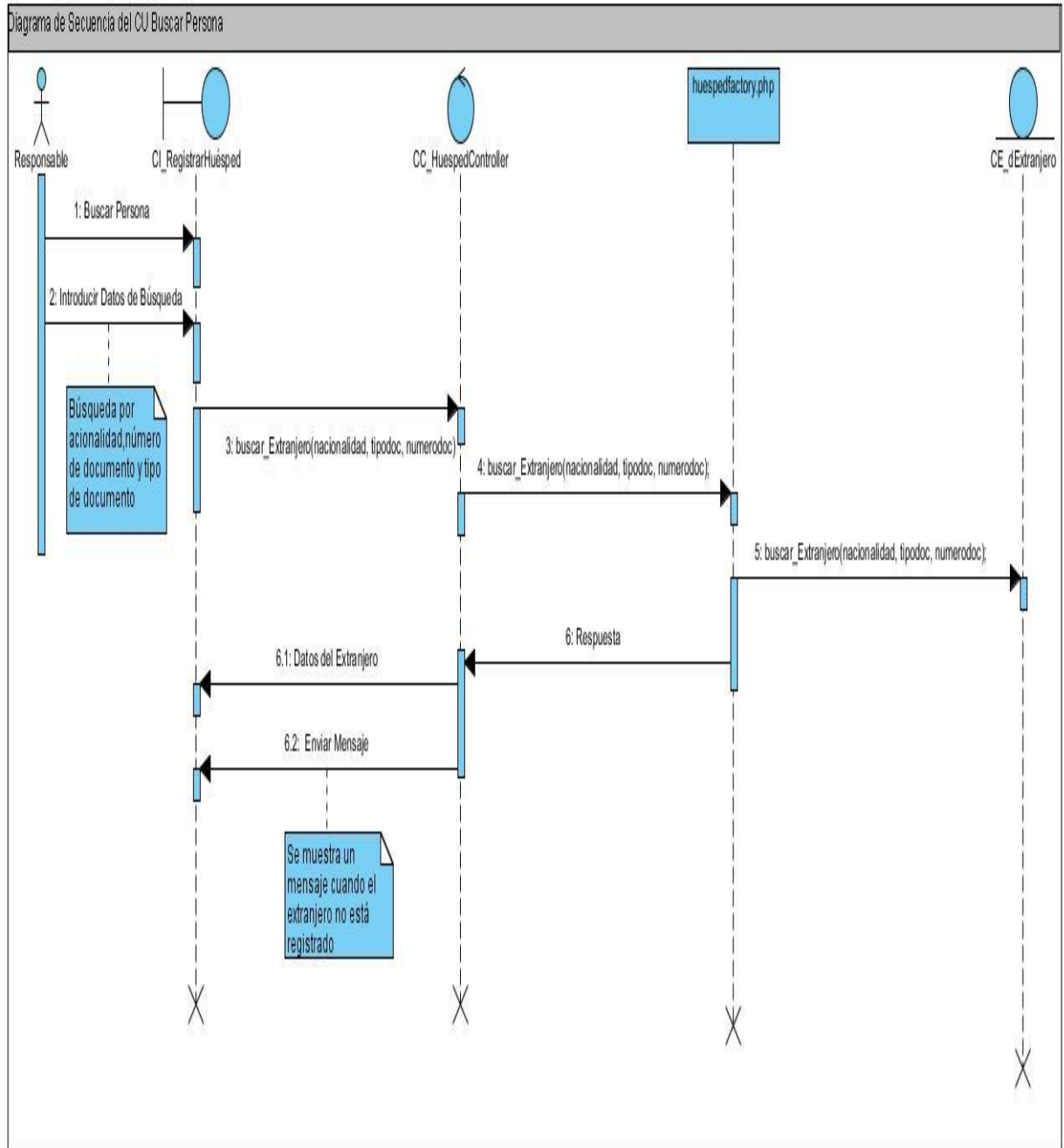


Figura 34 Diagrama de Secuencia CU Mostrar Reportes

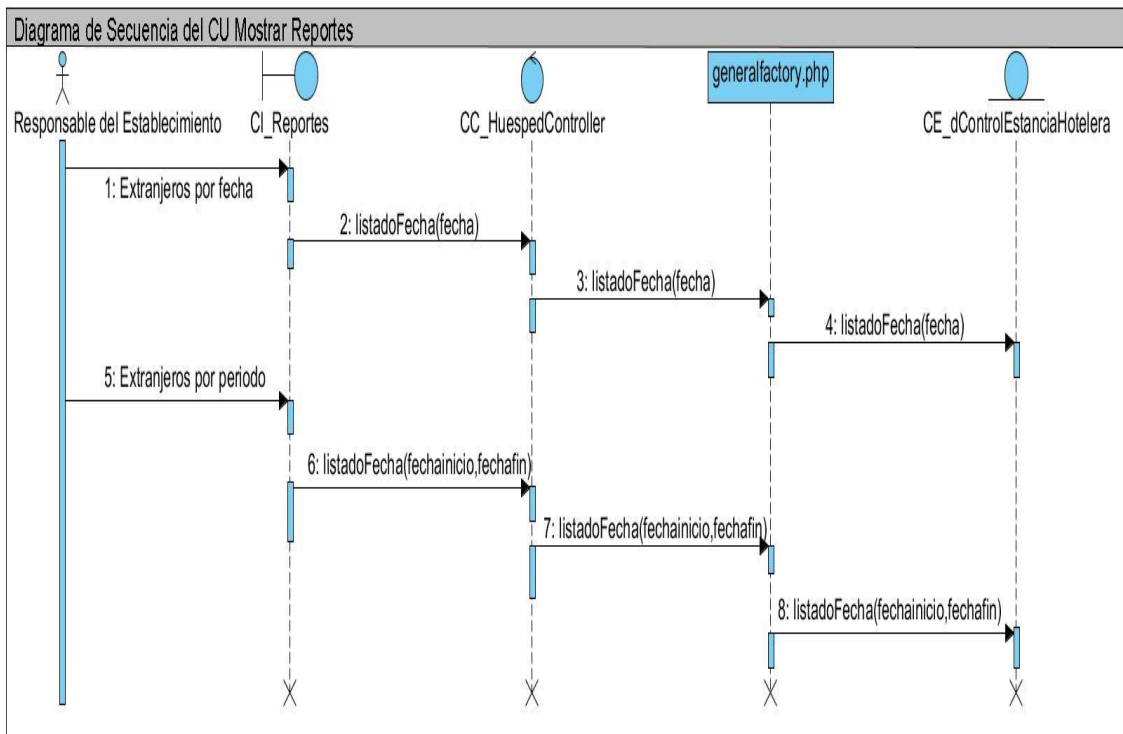


Figura 35 Diagrama de Secuencia CU Registrar Acompañante

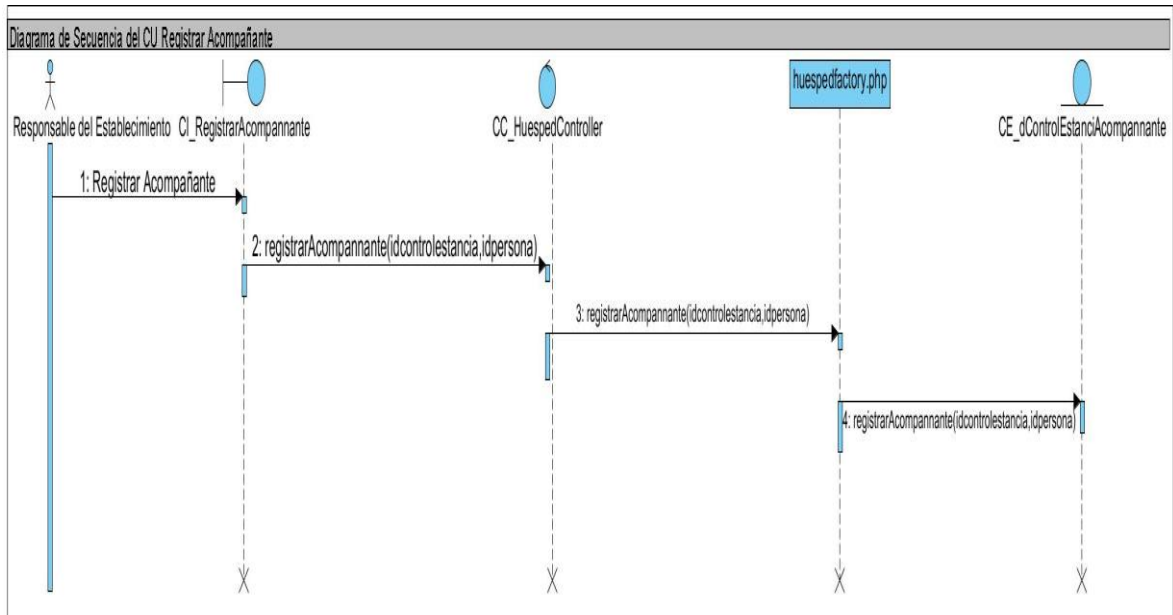
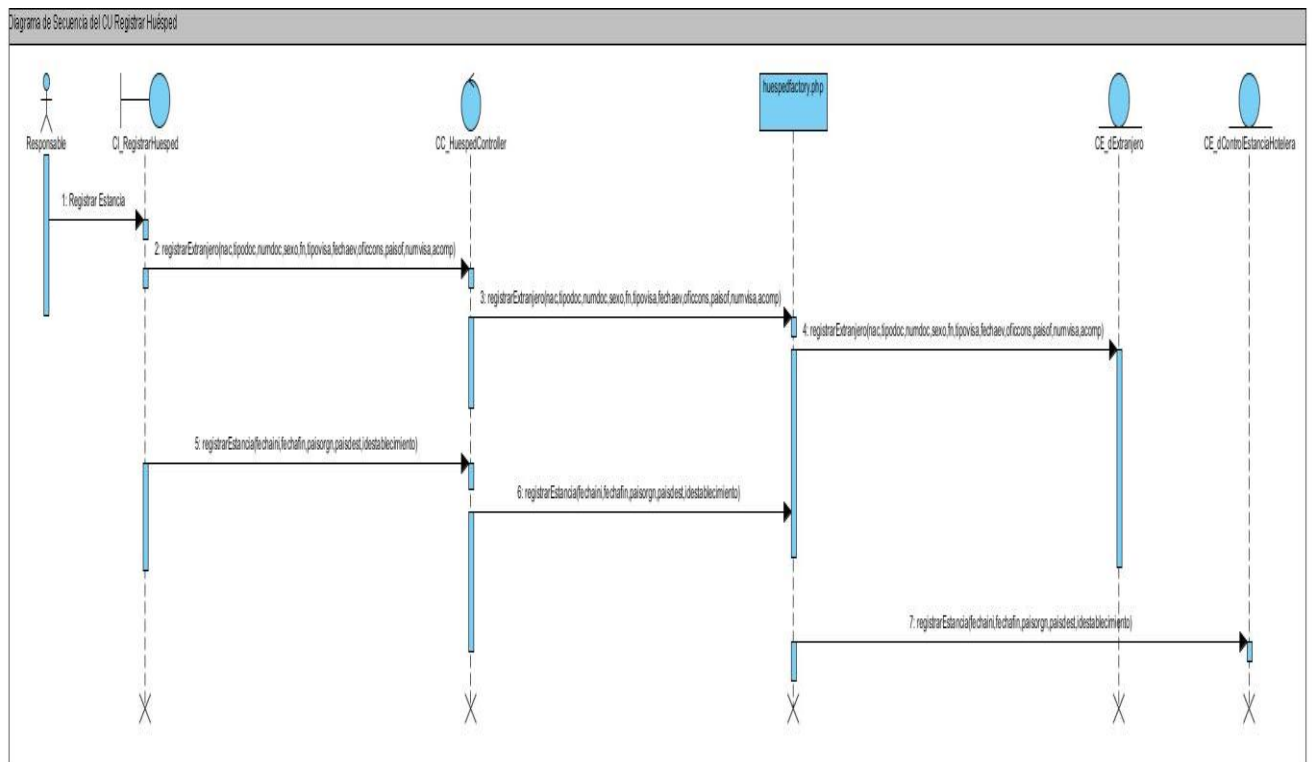


Figura 36 Diagrama de Secuencia CU Registrar Huésped



Anexo 5 Diseño de casos de prueba

Escenario	Descripción	Variable 1 "Usuario"	Respuesta del sistema	Flujo central
EC 1.1 Eliminar Usuario correctamente.	Se selecciona el usuario que se desea eliminar y el sistema lo elimina.	V "Imarcheco"	El sistema elimina el usuario y muestra el mensaje "Usuario eliminado con éxito".	<ol style="list-style-type: none"> 1. El usuario se autentica en el sistema con el rol Administrador. 2. El sistema muestra la pantalla donde se encuentran los módulos Administración y Control de Estancia. 3. El usuario selecciona el módulo "Administración". 4. El sistema muestra las diferentes funcionalidades. 5. El usuario selecciona Eliminar usuario. 6. El sistema muestra una pantalla con el listado de los usuarios

				<p>existentes.</p> <p>7. El usuario selecciona al que desea eliminar.</p> <p>8. El sistema muestra un mensaje de confirmación “Esta seguro de eliminar este usuario”.</p> <p>9. El usuario presiona Aceptar.</p> <p>10. El sistema elimina al usuario y muestra el mensaje “Usuario eliminado con éxito”.</p>
EC 1.2 Eliminar Usuario sin seleccionar ninguno.	No se elimina al usuario porque no se seleccionó.	I (vacío)	El sistema no elimina al usuario y muestra el mensaje de error “Para eliminar un usuario primero debe seleccionar uno”.	<p>1. El usuario se autentica en el sistema con el rol Administrador.</p> <p>2. El sistema muestra la pantalla donde se encuentran los módulos Administración y Control de Estancia.</p> <p>3. El usuario selecciona el módulo “Administración”.</p> <p>4. El sistema muestra las diferentes funcionalidades.</p> <p>5. El usuario selecciona Eliminar usuario.</p> <p>6. El sistema muestra una pantalla con el listado de los usuarios existentes.</p> <p>7. El usuario presiona el botón eliminar sin haber seleccionado el usuario.</p> <p>8. El sistema muestra un mensaje de error “Para eliminar un usuario primero debe seleccionar uno”.</p> <p>9. El sistema no elimina el usuario.</p>

Tabla 14 Diseño de Casos de Prueba basado en requisitos: Eliminar Usuario

Escenario	Descripción	Variabl e 1 "Esta blecimiento"	Variabl e 2 "Fecha de entrada"	Variabl e 3 "Fecha de salida"	Variabl e 4 "País de Origen"	Variable 5 "País de Destino"	Variab le 6 "Nacionalidad"	Variab le 7 "Número de documento"	Variab le 8 "Tipo de documento"	Respu esta del Sistema	Flujo Central
EC 1.1 Registrar Huésped correctamente.	Se registran los datos de la estancia del huésped para el establecimiento.	V "líder"	V "14/03/2012"	V "14/03/2012"	V "Brasil"	V "Brasil"	V "Brasil"	V "M123456"	V "Pasaporte"	El sistema registra al extranjero y la estancia en el establecimiento.	<p>11. El usuario se autentica en el Sistema Control de Hoteles.</p> <p>12. El sistema muestra la pantalla donde se encuentran los módulos Administración y Control de Estancia.</p> <p>13. El usuario selecciona el módulo "Control de Estancia".</p> <p>14. El sistema muestra las diferentes funcionalidades.</p> <p>15. El usuario selecciona "Registrar Huésped".</p> <p>16. El sistema muestra una interfaz para realizar la búsqueda del extranjero.</p>

											<p>17. El usuario introduce la nacionalidad, el número de documento y el tipo de documento.</p> <p>18. El sistema realiza la búsqueda y devuelve los datos del extranjero.</p> <p>19. El usuario introduce los datos de la estancia y presiona el botón "Registrar".</p> <p>20. El sistema registra la estancia y muestra el mensaje "Estancia registrada con éxito".</p>
EC 1.2 Registrar Huésped dejand	No se registra la estancia porque no se	I "vacío"	V "14/03/2012"	V "14/03/2012"	V "Brasil"	V "Brasil"	V "Brasil"	V "M123456"	V "Paquete"	El sistema no registra la estancia del	1. El usuario se autentica en el Sistema Control de

o campo s vacíos	pasaron todos los campos.	V "líder"	I "vacío"	V "14/03/ 2012"	V "Brasil"	I "vacío"	V "Brasil"	V "M M1 234 56"	V "Pa saporte "	extranjero y muestra el mensaje "Existen campos vacíos"	<p>Hoteles.</p> <p>2. El sistema muestra la pantalla donde se encuentran los módulos Administración y Control de Estancia.</p> <p>3. El usuario selecciona el módulo "Control de Estancia".</p> <p>4. El sistema muestra las diferentes funcionalidades.</p> <p>5. El usuario selecciona "Registrar Huésped".</p> <p>6. El sistema muestra una interfaz para realizar la búsqueda del extranjero.</p> <p>7. El usuario introduce la nacionalidad, el número de documento y el tipo de documento.</p> <p>8. El sistema realiza la búsqueda</p>
---------------------------	------------------------------------	--------------	--------------	-----------------------	---------------	--------------	---------------	-----------------------------	--------------------------	---	---

											<p>y devuelve los datos del extranjero.</p> <p>9. El usuario introduce los datos de la estancia dejando campos vacíos y presiona el botón "Registrar".</p> <p>10. El sistema no registra la estancia y muestra el mensaje de error "Existen campos vacíos".</p>
EC 1.3 Registrar Huésped pasando datos incorrectos.	No se registra la estancia porque existen datos incorrectos.	V "líder"	V "14/03/2012"	V "14/03/2011"	V "Brasil"	V "Brasil"	V "Brasil"	V "M123456"	V "Pasaporte"	El sistema no registra la estancia del extranjero y muestra el mensaje "La fecha de salida debe ser posterior a la fecha de entrada".	<p>1. El usuario se autentica en el Sistema Control de Hoteles.</p> <p>2. El sistema muestra la pantalla donde se encuentran los módulos Administración y Control de Estancia.</p> <p>3. El usuario selecciona el módulo "Control de Estancia".</p> <p>4. El sistema</p>

											<p>muestra las diferentes funcionalidades.</p> <p>5. El usuario selecciona "Registrar Huésped".</p> <p>6. El sistema muestra una interfaz para realizar la búsqueda del extranjero.</p> <p>7. El usuario introduce la nacionalidad, el número de documento y el tipo de documento.</p> <p>8. El sistema realiza la búsqueda y devuelve los datos del extranjero.</p> <p>9. El usuario introduce los datos de la estancia pasando una fecha de salida menor que la fecha de entrada y presiona "Registrar".</p> <p>10. El sistema</p>
--	--	--	--	--	--	--	--	--	--	--	--

											no registra la estancia y muestra el mensaje de error "La fecha de salida debe ser posterior a la fecha de entrada".
--	--	--	--	--	--	--	--	--	--	--	--

Tabla 15 Diseño de Casos de Prueba basado en requisitos: Registrar huésped

Escenario	Descripción	Variable 1 "Nacionalidad"	Variable 2 "Número de documento"	Variable 3 "Tipo de documento"	Respuesta del sistema	Flujo central
EC 1.1 Buscar Persona correctamente.	Se realiza una búsqueda para encontrar al extranjero que se corresponda con los datos introducidos.	V "Brasil"	V "MM123456"	V "Pasaporte"	Muestra el listado de estancias existentes y finaliza la que se seleccionó.	<ol style="list-style-type: none"> 1. El usuario se autentica en el Sistema de Control de Hoteles. 2. El sistema muestra la pantalla donde se encuentran los módulos Administración y Control de Estancia. 3. El usuario selecciona el módulo "Control de Estancia". 4. El sistema muestra las diferentes funcionalidades. 5. El usuario selecciona "Registrar Huésped". 6. El sistema muestra una

Anexos

						<p>pantalla con los campos nacionalidad, número de documento y tipo de documento, para buscar al extranjero.</p> <p>7. El usuario introduce los datos.</p> <p>8. El sistema realiza la búsqueda y devuelve los datos del extranjero.</p>
EC	1.2	No se realiza la búsqueda para encontrar al extranjero porque faltan datos.	<p>I "vacío"</p> <p>V "Brasil"</p> <p>V "Brasil"</p>	<p>V "MM123456"</p> <p>I "vacío"</p> <p>V "MM123456"</p>	<p>V "Pasaporte"</p> <p>V "Pasaporte"</p> <p>I "vacío"</p>	<p>El sistema muestra un mensaje de error "Existen campos vacíos".</p> <p>9. El usuario se autentica en el Sistema Control de Hoteles.</p> <p>10. El sistema muestra la pantalla donde se encuentran los módulos Administración y Control de Estancia.</p> <p>11. El usuario selecciona el módulo "Control de Estancia".</p> <p>12. El sistema muestra las diferentes funcionalidades.</p> <p>13. El usuario selecciona "Registrar Huésped".</p> <p>14. El sistema muestra una pantalla con los campos</p>

						<p>nacionalidad, número de documento y tipo de documento, para buscar al extranjero.</p> <p>15. El usuario introduce los datos dejando campos vacíos.</p> <p>16. El sistema no realiza la búsqueda y muestra un mensaje de error "Existen campos vacíos".</p>
EC 1.3	No se realiza la búsqueda para encontrar al extranjero porque faltan datos.	<p>I "6565656"</p> <p>V "Brasil"</p> <p>V "Brasil"</p>	<p>V "MM123456"</p> <p>I "ggg**y"</p> <p>V "MM123456"</p>	<p>V "Pasaporte"</p> <p>V "Pasaporte"</p> <p>I "=-0/gh"</p>	El sistema muestra un mensaje de error "Existen datos incorrectos".	<p>1. El usuario se autentica en el Sistema Control de Hoteles.</p> <p>2. El sistema muestra la pantalla donde se encuentran los módulos Administración y Control de Estancia.</p> <p>3. El usuario selecciona el módulo "Control de Estancia".</p> <p>4. El sistema muestra las diferentes funcionalidades .</p> <p>5. El usuario selecciona "Registrar Huésped".</p> <p>6. El sistema muestra una pantalla con los</p>

							<p>campos nacionalidad, número de documento y tipo de documento, para buscar al extranjero.</p> <p>7. El usuario introduce los datos incorrectamente.</p> <p>8. El sistema no realiza la búsqueda y muestra un mensaje de error "Existen datos incorrectos".</p>
--	--	--	--	--	--	--	--

Tabla 16 Diseño de Casos de Prueba basado en requisitos: Buscar persona

Escenario	Descripción	Variable 1 "Establecimientos"	Variable 2 "Listado de Extranjeros"	Variable 3 "Acompañantes Disponibles"	Respuesta del sistema	Flujo central
EC 1.1. Registrar acompañante correctamente.	Se muestra el listado de los huéspedes registrados en el establecimiento y se le asigna un acompañante al huésped seleccionado.	V "Cumberland"	V "Ronaldo Castro"	V "Pepe Artu"	Se asigna un acompañante al huésped deseado.	<ol style="list-style-type: none"> 1. El usuario se autentica en el Sistema de Control de Hotel. 2. El sistema muestra la pantalla donde se encuentran los módulos Administración y Control de Estancia. 3. El usuario selecciona el módulo "Control de

						<p>Estancia”.</p> <p>4. El sistema muestra las diferentes funcionalidades.</p> <p>5. El usuario selecciona “Registrar Acompañante”.</p> <p>6. El sistema muestra una pantalla con los establecimientos existentes.</p> <p>7. El usuario selecciona el establecimiento deseado y presiona el botón “Listar Huéspedes”.</p> <p>8. El sistema muestra el listado de los huéspedes alojados en el establecimiento seleccionado.</p> <p>9. El usuario selecciona el huésped al que desea asignarle un acompañante.</p> <p>10. El sistema muestra el listado de los posibles</p>
--	--	--	--	--	--	--

						<p>acompañantes para dicho huésped mostrando la siguiente información: “número de documento”, “nacionalidad”, “nombre y apellidos”.</p> <p>11. El usuario selecciona al huésped que quiere asignar como acompañante y presiona el botón “Registrar”.</p> <p>12. El sistema registra el acompañante y muestra el mensaje “Acompañante registrado con éxito”.</p>
EC 1.2. Registrar acompañante sin seleccionar el establecimiento.	No se muestra el listado de los huéspedes alojados en el establecimiento.	I (vacío)	NA	NA	El sistema muestra un mensaje de error “Para listar los huéspedes debe seleccionar el establecimiento”.	<p>1. El usuario se autentica en el Sistema Control de Hoteles.</p> <p>2. El sistema muestra la pantalla donde se encuentran los módulos Administración y Control de Estancia.</p> <p>3. El usuario selecciona el</p>

Anexos

						<p>módulo "Control de Estancia".</p> <ol style="list-style-type: none"> 4. El sistema muestra las diferentes funcionalidades. 5. El usuario selecciona "Registrar Acompañante". 6. El sistema muestra una pantalla con los establecimientos existentes. 7. El usuario no selecciona el establecimiento y presiona el botón "Listar Huéspedes". 8. El sistema muestra un mensaje de error "Para listar los huéspedes debe seleccionar el establecimiento".
EC 1.3. Registrar acompañante sin seleccionar el extranjero	No se muestra el listado de los huéspedes alojados en el establecimiento ya que el periodo de tiempo no puede estar vacío.	"Cumberland"	NA	NA	El sistema muestra un mensaje de error "Los campos de las fechas no pueden estar vacíos".	<ol style="list-style-type: none"> 1. El usuario se autentica en el Sistema Control de Hoteles. 2. El sistema muestra la pantalla donde se encuentran los módulos Administración y Control de Estancia. 3. El usuario selecciona el módulo "Control de Estancia". 4. El sistema muestra las diferentes funcionalidades. 5. El usuario selecciona registrar Acompañante. 6. El sistema muestra una pantalla para seleccionar el establecimiento

						<p>Hotelero.</p> <p>7. El usuario selecciona el establecimiento Hotelero y presiona el botón Listar Huéspedes.</p> <p>8. El sistema muestra una pantalla con los huéspedes alojados.</p> <p>9. El usuario presiona el botón Listar Acompañante Disponible.</p> <p>10. El sistema muestra un mensaje de error "Para listar el Acompañante debe Seleccionar un Huésped".</p>
--	--	--	--	--	--	--

Tabla 17 Diseño de Casos de Prueba basado en requisitos: Registrar Acompañante

Escenario	Descripción	Variable 1 "Fecha Inicio"	Variable 2 "Establecimiento Hotelero"	Respuesta del sistema	Flujo central
EC 1.1 Listar huésped por fecha.	Se muestra el listado de huéspedes alojados en el establecimiento en una fecha determinada.	V 3/04/2011	"Conde"	Muestra el listado de los huéspedes hospedados y exporta el reporte en formato Pdf.	<p>1. El usuario se autentica en el Sistema Control de Hoteles.</p> <p>2. El sistema muestra la pantalla donde se encuentran los módulos Administración y Control de Estancia.</p> <p>3. El usuario selecciona el módulo "Control de Estancia".</p> <p>4. El sistema muestra las diferentes funcionalidades.</p> <p>5. El usuario selecciona "Listado por fecha".</p> <p>6. El sistema muestra una pantalla para seleccionar la Fecha de Inicio y el</p>

					<p>Establecimiento Hotelero.</p> <p>7. El usuario selecciona la Fecha de Inicio y el Establecimiento Hotelero y presiona el botón Listar.</p> <p>8. Si existen huéspedes para esta fecha en el establecimiento el sistema muestra un listado, mostrando de cada uno la siguiente información: "nombre", "apellidos", "sexo", "nacionalidad", "fecha entrada" y "fecha salida". Si no existen huéspedes en esa fecha en el establecimiento el sistema muestra un mensaje de alerta "No existen huéspedes alojados en el establecimiento en esta fecha".</p>
EC 1.2	No se muestra el listado de los huéspedes alojados en el establecimiento porque existen campos vacíos.	I (vacío)	NA	El sistema muestra un mensaje de error "Existen campos	<p>1. El usuario se autentica en el Sistema Control de Hoteles.</p> <p>2. El sistema muestra la pantalla donde se</p>
Listar		NA	I (vacío)		

				vacíos”.	<p>encuentran los módulos Administración y Control de Estancia.</p> <p>3. El usuario selecciona el módulo “Control de Estancia”.</p> <p>4. El sistema muestra las diferentes funcionalidades.</p> <p>5. El usuario selecciona la opción Listar huésped por fecha.</p> <p>6. El sistema muestra una pantalla para seleccionar la Fecha de Inicio y el Establecimiento Hotelero.</p> <p>7. El usuario Presiona el botón Listar.</p> <p>8. El sistema muestra un mensaje “Existen Campos vacíos”.</p>
--	--	--	--	----------	--

Tabla 18 Diseño de Casos de Prueba basado en requisitos: Listar huésped por fecha

Escenario	Descripción	Variable 1 “Fecha Inicio”	Variable 2 “Fecha Fin”	Variable 3 “Establecimiento Hotelero”	Respuesta del sistema	Flujo central
EC 1.1 Listar huésped es por período correctamente.	Se muestra el listado de los huéspedes alojados en un establecimiento en un período de tiempo.	V 9/5/2011	V 11/3/2012	V “Conde”	Muestra el listado de los huéspedes hospedados y exporta el reporte en formato Pdf.	<p>11. El usuario se autentica en el Sistema Control de Hoteles.</p> <p>12. El sistema muestra la pantalla donde se encuentran los módulos Administración y Control de Estancia.</p> <p>13. El usuario selecciona el módulo “Control de Estancia”.</p> <p>14. El sistema muestra las diferentes</p>

						<p>funcionalidades.</p> <p>15. El usuario selecciona Listar huésped por período.</p> <p>16. El sistema muestra una pantalla para seleccionar la fecha de inicio, fecha de Fin del período y el Establecimiento Hotelero.</p> <p>17. El usuario selecciona la fecha de inicio, la fecha de fin del período y el Establecimiento Hotelero y presiona el botón Listar.</p> <p>18. El sistema muestra el listado de todos los huéspedes alojados en ese período de tiempo, mostrando de cada uno la siguiente información: "nombre", "apellidos", "sexo", "nacionalidad", "fecha entrada" y "fecha salida".</p>
EC 1.2	No se muestra el listado de los huéspedes alojados en el establecimiento ya que el periodo de	I 9/5/2012	V 11/3/2011	V "Conde"	El sistema muestra un mensaje de error "Período incorrecto, la fecha de inicio debe ser menor que la fecha de fin".	<p>10. El usuario se autentica en el Sistema Control de Hoteles.</p> <p>11. El sistema muestra la pantalla donde se encuentran los módulos Administración y Control de</p>

	tiempo introducido es incorrecto.					<p>Estancia.</p> <p>12. El usuario selecciona el módulo "Control de Estancia".</p> <p>13. El sistema muestra las diferentes funcionalidades.</p> <p>14. El usuario selecciona Listar huésped por período.</p> <p>15. El sistema muestra una pantalla para seleccionar la fecha de inicio, la Fecha de Fin del período y el Establecimiento Hotelero.</p> <p>16. El usuario selecciona la fecha de inicio, la fecha de fin y el establecimiento Hotelero y presiona el botón Listar.</p> <p>17. El sistema muestra un mensaje de error "Período incorrecto, la fecha de inicio debe ser menor que la fecha de fin".</p>
EC 1.3	No se muestra el listado de los huéspedes alojados en el	I (vacío)	V 11/3/2011	V "Conde"	El sistema muestra un mensaje de error "Existen Campos	<p>1. El usuario se autentica en el Sistema Control de Hoteles.</p> <p>2. El sistema muestra la</p>
		V 9/5/2011	V (vacío)	V "Conde"		

campos vacíos.	establecimiento ya que no pueden dejar campos vacíos.	V	V	V	Vacíos".	<p>pantalla donde se encuentran los módulos Administración y Control de Estancia.</p> <p>3. El usuario selecciona el módulo "Control de Estancia".</p> <p>4. El sistema muestra las diferentes funcionalidades.</p> <p>5. El usuario selecciona Listar huésped por período.</p> <p>6. El sistema muestra una pantalla para seleccionar la fecha de inicio, la fecha de fin del período y el Establecimiento Hotelero.</p> <p>7. El usuario selecciona los datos dejando campos vacíos y presiona el botón Listar.</p> <p>8. El sistema muestra un mensaje de error "Existen Campos Vacíos".</p>
		9/5/2011	11/3/2011	(vacío)		

Tabla 19 Diseño de Casos de Prueba basado en requisitos: Listar huésped por período

Escenario	Descripción	Variable 1 "Usuario"	Variable 2 "Contraseña"	Respuesta del sistema	Flujo central
EC 1.1 Autenticar usuario correctamente.	Se introduce el usuario y la contraseña y queda	V "Imarcheco"	V "lisandra"	Autentica el usuario permitiéndole utilizar el sistema.	1. El sistema muestra una interfaz para introducir el usuario y

	autenticado el usuario en el sistema.				<p>la Contraseña.</p> <ol style="list-style-type: none"> 2. El usuario introduce su usuario y contraseña. 3. El sistema autentica al usuario permitiéndole utilizar sus funcionalidades.
EC 1.2 Autenticar usuario dejando campos vacíos.	No se introduce el usuario y la contraseña del usuario.	<p>V "Imarcheco"</p> <p>I (vacío)</p>	<p>I (vacío)</p> <p>V "lisandra"</p>	El sistema muestra un mensaje de error "Existen Campos Vacíos".	<ol style="list-style-type: none"> 1. El sistema muestra una interfaz para introducir el usuario y la Contraseña. 2. El usuario no introduce su usuario y contraseña. 3. El sistema muestra un mensaje de error "Existen Campos Vacíos".

Tabla 20 Diseño de Casos de Prueba basado en requisitos: Autenticar usuario

Escenario	Descripción	Variable 1 "Nombre"	Variable 2 "Apellidos"	Variable 3 "Cédula"	Variable 4 "Fecha de nacimiento"	Variable 5 "usuario"	Variable 6 "Contraseña"	Variable 7 "Rol"	Respuesta del sistema	Flujo central
EC 1.1 Modificar usuario correctamente.	Se introducen los datos del usuario y queda añadido en la base de datos.	V "Pedro"	V "Macheco"	V "LM12345"	V "1990/12/14"	V "pmarc heco"	V "Pedro123"	V "Administrador"	Autentica el usuario permitiéndole utilizar el sistema.	<ol style="list-style-type: none"> 1. El usuario se autentica en el sistema con el rol administrador. 2. El sistema muestra una pantalla donde se encuentran los módulos administración y Control de Estancia. 3. El usuario selecciona el módulo Administración. 4. El sistema muestra una pantalla con las diferentes funcionalidades. 5. El usuario selecciona la funcionalidad modificar Usuario. 6. El sistema muestra el listado de usuarios existentes. 7. El usuario selecciona el que desea modificar. 8. El sistema muestra una interfaz con

										<p>los datos del usuario seleccionado. "Nombre", "Apellidos" "Cédula" "fecha Nacimiento" "usuario" "Contraseña" "Rol".</p> <p>9. El usuario modifica los datos deseados y presiona el botón "Modificar".</p> <p>10. El sistema modifica el usuario y muestra el mensaje "El Usuario ha Sido Modificado con Éxito"</p>
EC 1.2	No se modifican los datos del usuario.	I (vacío)	V "Macheco"	V "LM12345"	V "1990/12/14"	V "pmarheco"	V "Pedro123"	V "Administrador"	El sistema muestra un mensaje de error "Existen	1. El usuario se autentica en el

vacíos.	V "Pedro"	I (vacío)	V "LM1 2345"	V "1990/ 12/14"	V "pmarc heco"	V "Pedr o123"	V "Admi nistra dor"	Campos Vacíos".	<p>sistema con el rol administrador.</p> <ol style="list-style-type: none"> 2. El sistema muestra una pantalla donde se encuentran los módulos administración y Control de Estancia. 3. El usuario selecciona el módulo Administración. 4. El sistema muestra una pantalla con las diferentes funcionalidades. 5. El usuario selecciona la funcionalidad Modificar Usuario. 6. El sistema muestra el listado de usuarios existentes. 7. El usuario selecciona el que desea modificar. 8. El sistema
	V "Pedro"	V "Mache co"	I (vacío)	V "1990/ 12/14"	V "pmarc heco"	V "Pedr o123"	V "Admi nistra dor"		
	V "Pedro"	V "Mache co"	V "LM1 2345"	I (vacío)	V "pmarc heco"	V "Pedr o123"	V "Admi nistra dor"		
	V "Pedro"	V "Mache co"	V "LM1 2345"	V "1990/ 12/14"	I (vacío)	V "Pedr o123"	V "Admi nistra dor"		
	V "Pedro"	V "Mache co"	V "LM1 2345"	V "1990/ 12/14"	V "pmarc heco"	I (vacío)	V "Admi nistra dor"		
	V "Pedro"	V "Mache co"	V "LM1 2345"	V "1990/ 12/14"	V "pmarc heco"	V "Pedr o123"	I (vacío)		

										<p>muestra una interfaz con los datos del usuario seleccionado "Nombre", "Apellidos", "Cédula", "fecha Nacimiento", "usuario", "Contraseña", "Rol".</p> <p>9. El usuario modifica los datos deseados dejando algunos vacíos y presiona el botón "Modificar".</p> <p>10. El sistema modifica el usuario y muestra el mensaje "Existen campos Vacíos".</p>
EC 1.3	Se Introduce los datos del usuario de manera incorrecta.	I "Pfsdfd"	V "Macheco"	V "LM12345"	V "1990/12/14"	V "pmarheco"	V "Pedro123"	V "Administrador"	El sistema muestra un mensaje de error "Datos Incorrecto"	1. El usuario se autentica en el sistema con el rol administrad
		V "Pedro"	I "sadasd"	V "LM12345"	V "1990/12/14"	V "pmarheco"	V "Pedro123"	V "Administrador"		

		V "Pedro"	V "Macheco"	I "425879"	V "1200/05/25"	V "pmarheco"	V "Pedro123"	V "Administrador"	s."	
		V "Pedro"	V "Macheco"	V "LM12345"	I "1990/12/14"	V "pmarcheco"	V "Pedro123"	V "Administrador"		
		V "Pedro"	V "Macheco"	V "LM12345"	V "1990/12/14"	I "etaveraz"	V "Pedro123"	V "Administrador"		
		V "Pedro"	V "Macheco"	V "LM12345"	V "1990/12/14"	V "pmarheco"	I "prueba"	V "Administrador"		
		V "Pedro"	V "Macheco"	V "LM12345"	V "1990/12/14"	V "pmarheco"	V "Pedro123"	I "usuario"		
										<p>or.</p> <ol style="list-style-type: none"> 2. El sistema muestra una pantalla donde se encuentran los módulos administración y Control de Estancia. 3. El usuario selecciona el módulo Administración. 4. El sistema muestra una pantalla con las diferentes funcionalidades. 5. El usuario selecciona la funcionalidad Modificar Usuario. 6. El sistema muestra el listado de usuarios existentes. 7. El usuario selecciona el que desea modificar. 8. El sistema muestra una interfaz con los

										<p>datos del usuario selecciona do "Nombre", "Apellidos" "Cédula" "fecha Nacimiento" "usuario" "Contraseña" "Rol".</p> <p>9. El usuario llena los campos introduciendo alguno mal y presiona el botón Adicionar.</p> <p>10. El sistema muestra el mensaje de error " Datos Incorrectos " .</p>
--	--	--	--	--	--	--	--	--	--	--

Tabla 21 Diseño de Casos de Prueba basado en requisitos: Modificar usuario