Universidad Central “Marta Abreu” de las Villas
Facultad Matemática Física y Computación
Ingeniería Informática

Trabajo de Diploma

Aplicación Web para la Reserva en los Restaurantes del hotel Iberostar Ensenachos

Autor
Alberto Enrique Cárdenas Aguiar

Tutores
Msc. Ernesto Díaz López
Dra. Gheisa Lucía Ferreira Lorenzo

Santa Clara
2012

Hago constar que el presente trabajo fue realizado en la Universidad Central “Marta Abreu” de Las Villas como parte de la culminación de los estudios de la especialidad de Ingeniería Informática, autorizando a que el mismo sea utilizado por la institución, para los fines que estime conveniente, tanto de forma parcial como total y que además no podrá ser presentado en eventos ni publicado sin la autorización de la Universidad.

Firma del autor

Los abajo firmantes, certificamos que el presente trabajo ha sido realizado según acuerdos de la dirección de nuestro centro y el mismo cumple con los requisitos que debe tener un trabajo de esta envergadura referido a la temática señalada.

Firma del tutor

Firma del jefe
del Seminario

Dedicatoria

Este trabajo está dedicado a mis padres, mi abuela y toda mi familia quien con su amor infinito han hecho de este sueño una realidad. Gracias por todo su apoyo incondicional.

Agradecimientos

A mis padres Aida y Alberto, por su confianza y dedicación.

A mi abuela, por siempre estar a mi lado cuando más necesite de ti.

A mi familia en general, por ser tan comprensivos y ayudarme a salir adelante.

A mis tutores, Ernesto y Gheisa, por su apoyo, paciencia e infinitas horas dedicadas.

A Frank, quien siempre me ayudó cuando más lo necesitaba.

A Liz, por darme ánimo cuando nada tenía sentido.

A mis profesores de la universidad, por sus enseñanzas.

Les agradezco a todos porque de diferentes formas me han brindado sus manos para sostenerme y no dejarme caer. Alberto.

Resumen

Resumen

En la actualidad, el mundo hotelero está experimentando cambios acelerados en su forma de concebir el turismo. La utilización de la informática como vía para el desarrollo de herramientas que permitan automatizar servicios dentro de la gestión hotelera, puede elevar la calidad del servicio prestado en las instalaciones turísticas. El presente trabajo responde a una problemática real del Hotel Iberostar Ensenachos relacionada con el servicio de gestión de restaurantes. Se brinda solución a los problemas detectados por el software para la reserva de restaurantes que se encuentra actualmente en uso y se desarrolla una aplicación Web para automatizar el proceso de reserva de los restaurantes realizado por los conserjes, así como el diseño y distribución de las mesas y restaurantes, de lo que se encarga el jefe del departamento de Alimentos y Bebidas, entre sus funcionalidades más importantes. La automatización de este proceso tiene gran importancia pues ofrece una respuesta certera y rápida a los huéspedes.

Abstract

Abstract

Nowadays the hotel industry is experiencing rapid changes in the way of conceiving tourism. The use of computers as a way for the development of tools to automate services within the hotel management can improve the quality of the service in tourist facilities. In this work is presented a real problem of the Iberostar Royal Ensenachos related to the restaurants' management service. Solutions are provided to the problems identified by the software for booking restaurants that is currently in use and develops a web application to automate the process of booking restaurants carried out by the receptionists, and the design and distribution of tables and restaurants, which is responsibility of the Food and Beverage's head of department and it is among his/her most important functions. The automation of this process is of high importance because it provides a quick and accurate answer to the guests.

Tabla de Contenido

Tabla de contenido

INTRODUCCIÓN 1

CAPÍTULO I: *"Fundamentación Teórica"* 5

1.1 Sistemas de gestión hotelera 5

1.1.1 Sistemas de Gestión Hotelera: situación en el mundo 5

1.1.2 Sistemas de Gestión Hotelera: situación en Cuba 7

1.2 Sistema de Reservas en Restaurantes 11.1 (SRR 11.1) 8

1.3 Entorno Web 9

1.3.1 Características 10

1.4 Lenguajes de programación para la Web 11

1.4.1 HTML 11

1.4.2 Javascript 12

1.4.3 PHP 13

1.5 Servidores de aplicación web 14

1.5.1 Apache 14

1.6 Gestores de Bases de Datos 15

1.6.1 Microsoft SQL Server 2000 16

1.7 Marcos de Trabajo 16

1.7.1 Symfony 18

1.7.2 Buenas razones para escoger Symfony 19

1.7.3 Symfony y el patrón de arquitectura MVC 19

1.7.4 Doctrine 21

1.8 Herramientas de desarrollo de software 21

1.8.1 NetBeans IDE 7.01 22

1.8.2 Biblioteca Jquery 22
1.8.3 Visual Paradigm 22
1.8.4 Artisteer 3.0 23
1.9 Conclusiones Parciales 23
CAPÍTULOII: "*Características del sistema*" 25
2.1 Modelación del Negocio 25
2.1.1 Diagrama de Casos de Uso del Negocio 25
2.1.2 Descripción literal de los casos de uso del negocio 26
2.2 Requisitos del Sistema 27
2.2.1 Requisitos Funcionales 27
2.2.2 Requisitos no funcionales 29
2.3 Modelo de casos de uso 29
2.3.1 Actores del sistema 29
2.4 Diagrama de Casos de Uso del sistema 30
2.4.1 Especificación de los casos de uso 31
2.5 Modelo de Diseño 36
2.5.1 Arquitectura general del sistema. 36
2.5.2 Diagrama de clases del Diseño 38
2.5.3 Mapa de Navegación 39
2.6 Modelo de Datos 40
2.6.1 Modelo de Datos Conceptual (Diagrama Entidad-Relación) 40
2.6.2 Modelo de datos Físicos 41
2.7 Modelo de Implementación 44
2.8 Diagrama de Despliegue 45
2.9 Conclusiones parciales 45

CAPÍTULO III: *"Guía del usuario"* 48
3.1 Estudio de Factibilidad 48
3.1.1 Factibilidad Técnica 48
3.1.2 Factibilidad Económica 49
3.1.3 Análisis de Costo – Beneficio 57
3.2 Guía para el Usuario 58
3.2.1 Acceso al sistema 58
3.2.2 Composición de la aplicación 59
3.2.3 Navegación del usuario conserje 61
3.2.4 Navegación para el usuario jefe AB 63
3.2.5 Navegación para el usuario comercial 65
CONCLUSIONES 67
RECOMENDACIONES 69
BIBLIOGRAFÍA 71
Anexo I: Glosario de Términos 74
Anexo II: Comparación entre marcos de trabajos de PHP. 76
Anexo III: Trigger para actualizar la tabla de clientes 79
Anexo IV: Representación de la aplicación siguiendo el patrón de arquitectura MVC 81
Anexo V: Métodos desarrollados usando Doctrine para gestionar las reservaciones. 83

Introducción

INTRODUCCIÓN

El turismo, desde sus inicios hasta nuestros días, ha significado desarrollo para la humanidad en diferentes aspectos. Las necesidades humanas: comer, dormir, son objetivos a satisfacer por el hotelero, pero el esparcimiento y la diversión deben estar presentes para la realización de su satisfacción espiritual; vacacionar debe ser entretenimiento y aprendizaje a través de una comunicación visual, modalidad que distingue a este siglo.

Según la Organización Mundial del Turismo, en el año 2008 el número de llegadas de turistas internacionales registradas en todo el mundo superó los 917 millones, lo que representó un aumento del 1.76% con respecto a 2007, ya en el 2010 alcanzó la cifra de 940 millones de personas por primera vez en la historia. Mostrándose un creciente el interés por otras modalidades del turismo que incluyen el turismo cultural (OMT, 2011).

De acuerdo a la Oficina Nacional de Estadísticas e Información de la República de Cuba (ONEI, 2012), el turismo en nuestra nación, tuvo un aumento del 105.2% en el período de Enero-Abril del 2012 con respecto al año 2011. Destacándose a Argentina como la nación de donde más visitantes arribaron, llegando a un 149.5%. Debido a estos resultados la Feria Internacional del Turismo FIT-2012, celebrado en la cayería norte de la provincia de Villa Clara estuvo dedicada a este hermano país.

La provincia de Villa Clara, en el turismo se destaca por el excelente potencial de atractivos naturales e histórico-culturales que posee, incluidos en ellos 28 playas con 37,5 kilómetros de longitud, pertenecientes mayormente a los cayos del norte de la provincia, forman parte del sub archipiélago Jardines del Rey[1] considerados reserva de la biosfera de Buena Vista (UNESCO, 2007).

El hotel Iberostar Ensenachos está ubicado en Cayo Ensenachos, en la costa norte de Villa Clara y está considerada la instalación hotelera más lujosa de Cuba. Cuenta con 506 habitaciones con vista al mar, a la piscina o al jardín. Opera en la modalidad "Todo Incluido". Entre las facilidades que brinda a los turistas se incluyen bares, piscinas espaciosas y rodeadas de exuberante vegetación, gimnasio, masaje, discoteca, deportes náuticos, restaurantes "a la carta" (incluidos uno de comida italiana, uno de comida asiática y uno de comida del mar), entre otros atractivos.

[1]Ver Glosario de Términos en Anexo 1

La variedad de restaurantes especializados en el hotel, permite a los clientes degustar exquisitos platos de la comida internacional y nacional.

El hotel cuenta con un sistema para las reservaciones en los restaurantes, el cual ha transitado por varias etapas. En un principio se hacían las reservas de forma manual. Dado lo engorroso del proceso, el Grupo de Electrónica del Turismo (GET) diseñó el sistema que actualmente se utiliza en varios hoteles del país. Este ha tenido múltiples versiones, hasta la versión actual Software para la Reserva en Restaurantes 11.1 (SRR 11.1) que posibilita la automatización del proceso de reservas en los diferentes restaurantes, dando alternativas para una correcta planificación de los mismos, lo que denota organización. Esto propicia que el turista se sienta a gusto y se convierta en cliente habitual de este mercado nacional.

SRR 11.1, en lo esencial, permite a los conserjes realizar reservas durante la estancia de los turistas, así como planificar las capacidades de los restaurantes en funcionamiento por parte del jefe AB. Sin embargo, esta aplicación informática ha presentado varias limitaciones que han sido objeto de análisis para una posterior solución, algunas se relacionan a continuación:

- Demoras al iniciar la aplicación, dadas por la conectividad a la base de datos del sistema general del hotel ZunPMSIberostar, desde donde se extrae la información de los correspondientes huéspedes, necesaria para la correcta reservación en el restaurante.
- Reservaciones repetidas o eliminadas incorrectamente.
- Fallas en la seguridad que permiten la manipulación incorrecta de las reservaciones.
- Demoras en la actualización de los datos de los huéspedes.
- Inconsistencia y diseño deficiente de la base de datos, pues permite almacenar valores nulos y tiene gran cantidad de campos repetidos.
- Alto consumo de recursos en los servidores, lo que conlleva en ocasiones al reinicio de estos.

Luego, el **objetivo general** de esta investigación es el siguiente:

> Desarrollar una aplicación Web para la gestión de las reservas en los restaurantes del hotel Iberostar Ensenachos, que permita la actualización sistemática de los datos de los huéspedes, así como la manipulación del contenido de los restaurantes, horarios y planificaciones para brindar un servicio eficiente a los turistas.

Los **objetivos específicos** que se proponen para dar cumplimiento al objetivo general de la investigación son:

1. Determinar las limitaciones del software empleado para la reserva de restaurantes y cómo éstas influyen en la gestión de las reservas para los restaurantes del hotel Iberostar Ensenachos.
2. Diseñar e implementar una solución informática para la reserva de restaurantes en el hotel Iberostar Ensenachos que resuelva las dificultades encontradas en el SRR 11.1.
3. Realizar un análisis de factibilidad e impacto económico del software desarrollado.

El trabajo de diploma, además de esta Introducción, está formado por tres Capítulos, Conclusiones, Recomendaciones, Bibliografía y Anexos.

En el **Capítulo I** se trata de manera general la gestión hotelera así como características esenciales de algunos sistemas de gestión hotelera existentes en Cuba y en el mundo. Se realiza además una descripción de la aplicación SRR 11.1 en cuando a sus limitaciones y un estudio de las tecnologías y tendencias actuales usadas para el desarrollo de aplicaciones Web.

En el **Capítulo II** se realiza la modelación del negocio y del sistema, el modelo de diseño y de datos. Se define además el modelo de implementación y la configuración de la aplicación a través del diagrama de despliegue; concluyendo de esta manera con la caracterización de la propuesta de solución.

En el **Capítulo III** se hace un análisis de factibilidad, en el que se detalla el impacto económico de la solución y se explica de manera general, la forma en que el usuario debe proceder para interactuar con el sistema.

El trabajo presentado brinda una solución práctica inmediata a un problema real, de ahí su principal valor.

CAPÍTULO I

“FUNDAMENTACIÓN TEÓRICA”

CAPÍTULO I: *"Fundamentación Teórica"*

En el capítulo se realiza una descripción general de los Sistemas de Gestión Hotelera desde la perspectiva mundial y se particulariza en su uso en Cuba. Se analizan las principales dificultades presentadas por el actual sistema de reservas de restaurantes SRR 11.1 que opera en el Hotel Iberostar Ensenachos. Es objetivo también del capítulo, realizar un análisis del entorno Web y las tecnologías existentes que pueden apoyar el desarrollo de una aplicación Web que de solución a la problemática planteada en la Introducción.

1.1 Sistemas de gestión hotelera

Los Sistemas de Gestión Hotelera (SGH) fueron y son en el plano de la administración hotelera el más representativo producto de la aplicación de las Tecnologías de la Información y las Comunicaciones en el sector del alojamiento, estandarizando y automatizando las principales funciones de la dirección y operación de un hotel.

1.1.1 Sistemas de Gestión Hotelera: situación en el mundo

Según (Salgado Febles, 2007) las primeras soluciones de este tipo trabajaron con programas y sistemas aislados que después se fueron integrando. La aparición del sistema operativo Unix para mini computadoras y su adaptación para trabajar con clientes ligeros y una máquina central (servidor) que no demandaba de grandes recursos, fue un factor que hizo que múltiples sistemas se desarrollaran a la medida de las cadenas y hoteles, pero también aparecieron productos que se adaptaban al funcionamiento de instalaciones diferentes.

Hoy, los fabricantes se especializan la mayoría de las veces en el llamado Front Office (FO) que incluye las funcionalidades de: Recepción y carpeta, Ama de Llaves y mantenimiento, Reservas, Comercial y relaciones públicas, Restaurantes y puntos de venta, Compras y almacén o Alimentos y bebidas. Es decir, la parte que tiene que ver con el cliente directamente y que funciona similar en todas partes y se realizan entonces interfaces para diferentes Back Office (BO) que son más cambiantes por las características de las contabilidades, normas de impuestos y otras características propias de cada país. También muchas cadenas tienen un Sistema

Informático de Gestión Económica corporativo, por lo que prefieren sólo adquirir el Front Office.

Pueden citarse algunas soluciones informáticas que atienden este ámbito, entre ellas:

NewHotel (Delfos, 2004)

Es un software propietario desarrollado por la empresa portuguesa Informarca Hotel Systems la cual está especializada en la producción de software informático para hotelería y turismo.

La familia de Software NewHotel para Windows es soportada por una arquitectura de sistema Cliente-Servidor, utilizando base de datos relacionales como Oracle o SQL Server, bajo lenguaje de programación Microsoft Visual C++, Delphi y Microsoft Visual Basic.

Entre los productos que mantiene en el mercado se encuentra el Front-Office 2001, que cuenta con apartados para las reservas de grupos, agencias, entradas y salidas, gestión de contratos, paquetes y todo incluido, entre otras características.

Si dispone de un apartado específico para gestionar la reserva en los restaurantes que ofrece operaciones semejantes a caja registradora, ventas directas, memorias de mesa, pedidos a cocina, consulta de cuentas, facturación, operaciones de apertura, cierre y control de caja, consulta instantánea de cuentas y mesas, etc.

Hotel 3000 Cea (CEA, 2010)

Es un software propietario para la gestión hotelera. Cuenta con una versión de prueba para 30 días. Está especializado en el trabajo sobre las habitaciones, clientes y reservas. También presenta una sección para los empleados según su cargo en el hotel. No dispone de un apartado específico para gestionar la reserva en los restaurantes.

Adapto (Parsec, 2005)

Adapto es el programa de Gestión Hotelera desarrollado por Parsec. Desde una sola pantalla, la Rejilla, se puede realizar fácilmente todas las operaciones que se registran en un Hotel: gestión de reservas conformado por crear reservas, dar entrada y salida a las habitaciones, carga extras, cobros, facturación detallada y agrupada a particulares, empresas y agencias (permite múltiples facturas por reserva), anulaciones, control de gastos e ingresos, listados e informes, *overbooking* y lista de espera.

Adapto es un programa totalmente gráfico, de fácil manejo, de forma que cualquier usuario con independencia de su nivel puede comenzar a trabajar con él desde el primer día. Es un software propietario, cuenta con licencias para su uso y una versión de prueba.

Está claro que estos software sólo son una pequeña parte de los existentes en el mercado. Adquirir una versión comercial puede oscilar desde los $3000 hasta cifras cercanas a los $10000 por año, contando con el hecho que son software desarrollados en terceros países, lo cual dificultaría enormemente su implementación, mantenimiento y cursos básicos para los operadores e instrucciones precisas para el administrador del sistema.

1.1.2 Sistemas de Gestión Hotelera: situación en Cuba

En Cuba se comienza con GesHotel, un sistema desarrollado para MS-DOS y realizado por técnicos del antiguo Instituto Nacional de Turismo (INTUR), cuyos antecedentes parten de finales de los años 80. Era el software que se usaba en la carpeta de algunos hoteles.

Con la rápida ampliación del turismo internacional se construyeron nuevos hoteles y se ampliaron muchos de los existentes. Fue necesario crear la infraestructura de cajas registradoras y programas para dar respuesta a nuevas demandas de administración hotelera mucho más complejas, en tiempo real y en red.

Posteriormente, como solución 100% nacional se pasa a Interhotel 2000, programado en lenguaje Foxpro para Windows, que es instalado en la mayoría de los hoteles administrados por cadenas nacionales. Su BO se conectó al resto de los SGH que tenían un FO diferente de Interhotel (exceptuando a SIHOT). Pero el FO de Interhotel era un desarrollo anticuado y se había quedado obsoleto, no respondiendo a las necesidades de información y las nuevas demandas de las cadenas hoteleras cubanas y tampoco de las internacionales, que en forma creciente operan en el país. Esto hace que se instalen diferentes sistemas. Entre los que aún hoy quedan, están LANMARK, Medallion, Grand Vision y Winhotel, fundamentalmente en hoteles mixtos y bajo administración extranjera.

En 1997 se realizan las primeras instalaciones de un nuevo SGH no nacional, el portugués **NewHotel** de Informarca. Con un diseño novedoso en el FO fue el primer sistema instalado que usaba a plenitud las bondades y facilidades del entorno Windows y una poderosa y profesional Base de Datos (Oracle). Su instalación llegó a la treintena de hoteles y unos 15 restaurantes y constituyó un impulso para los desarrolladores nacionales en la búsqueda de nuevas soluciones.

Es sin duda NewHotel el sistema que más avances había mostrado en estos últimos años, por el amplio diapasón de sus soluciones y la novedad que incorpora a las mismas.

Zun^2, el desarrollo más moderno del Interhotel 2000, con una concepción totalmente nacional, se presenta actualmente con un entorno gráfico más amigable y con MS SQL como Base de Datos profesional, logrando de hecho desplazar a su antecesor como el SGH de mayor presencia en las instalaciones hoteleras de Cuba. Aunque con insuficiencias, el carácter nacional de Zun y su integración con el BO, son elementos a su favor en la batalla con los otros SGH presentes en el mercado cubano.

Recientemente se ha incorporado a la competencia eHotel, que desarrolla y comercializa la empresa DATYS y que ofrece buenas perspectivas por su sólida plataforma de desarrollo basada en ORACLE, similar a la que introdujo en su momento NewHotel, pero con una concepción totalmente nacional e integrada con soluciones de gestión empresarial (BO) más completa y que acopla con sistemas para agencias de viaje y receptivos, entre otras ventajas (Salgado Febles, 2007).

1.2 Sistema de Reservas en Restaurantes 11.1

El sistema de Reservas en Restaurantes (SRR 11.1) es una solución informática para la gestión de restaurantes, que trabaja integrada a Zun (FO) en múltiples instalaciones hoteleras del país, en particular en el Hotel Iberostar Ensenachos. En los bungalows de este hotel se encuentran ubicados los conserjes. Entre sus principales funciones se encuentra la de realizar las reservaciones para los múltiples restaurantes atendiendo a las peticiones de los huéspedes. Estas peticiones son gestionadas actualmente mediante el sistema SRR 11.1, una aplicación desktop instalada en cada una de las computadoras vinculadas al sistema. Esta información transita por una red telefónica con la utilización de Módem al Departamento de Informática, donde se ubican los servidores. El sistema es administrado por el jefe AB.

A partir de entrevistas con diferentes directivos del hotel, entre ellos algunos conserjes, jefe AB e inclusive con el propio gerente, se han podido delimitar un conjunto de dificultades en este sistema que se detallan a continuación:

- La utilización del mencionado sistema se ve afectada porque la cantidad de usuarios que posee, en su momento pico, puede oscilar entre 18 y 22 aproximadamente. Esto trae

como consecuencia que se bloquean los servidores y se afecta la ejecución de otros sistemas informáticos que residen en el servidor de bases de datos.

- El actual sistema se conecta a la base de datos ZunPMSIberostar y realiza una copia local de los datos de los clientes hasta ese momento. Estos datos no se actualizan hasta que no se reinicia la aplicación.
- La situación anterior provoca problemas como el desconocimiento de la entrada o salida de los huéspedes, algún cambio de habitación, es decir, se trabaja con información que no está actualizada completamente.
- El sistema permite el cambio en registros de reservaciones que ya han sucedido en el tiempo. Esto hace que aparezcan más reservaciones que las que realmente corresponden a los huéspedes de acuerdo con su contrato.

1.3 Entorno Web

Una página Web es un sistema de documentos enlazados y accesibles a través de Internet, que pueden contener texto, imágenes, vídeos, etc. Un sitio Web es un grupo de páginas Web, generalmente comunes a un dominio de Internet. Estas son accedidas desde un URL *(Uniform Resource Locutor)*, raíz común llamada portada, que es una secuencia de caracteres, de acuerdo a un formato estándar, usado para nombrar recursos como documentos e imágenes, por su localización. Esa dirección es única para cada uno de los recursos de información disponibles en Internet.

Según (Yuen and Lau, 2003) una aplicación es una capa de presentación elaborada para ser consumida desde un navegador Web, que permite ser usada en cualquier sistema operativo, convirtiéndose en una aplicación multiplataforma que puede ser visualizada por cualquier número de computadoras a la vez y de forma independiente. Trae consigo además, facilidad de uso, versatilidad y velocidad de interfaz.

La capacidad para actualizar y mantener aplicaciones Web sin distribuir ni instalar software en las computadoras de los clientes es la razón clave de su popularidad que provoca a su vez, que la mayoría de los clientes no necesiten entrenamiento adicional para manipular la interfaz de la aplicación. Además, alojando la aplicación en un servidor Web, se logra una disponibilidad total del producto así como su código fuente.

1.3.1 Características

El éxito de un sitio Web depende de varias características:

- **Navegación**: Es la facilidad con la que un visitante accede y encuentra el contenido o servicio que sea de su interés.
- **Estética**: Comprende los aspectos de apariencia y composición de las páginas Web.
- **Búsqueda**: Es la capacidad de localizar o ubicar el contenido de interés para los usuarios de acuerdo con sus propios criterios.
- **Contenido**: Es la calidad y cantidad de información de utilidad para el visitante, por tanto, este renglón debe prevalecer. Un sitio debe presentar buenos contenidos que satisfagan las necesidades de los usuarios.
- **Eficiencia en el uso**: ¿Qué tan rápido realiza el usuario sus objetivos?
- **Fácil de recordar**: Una vez que el usuario ha utilizado el sistema, ¿qué tanto recuerda del mismo?

La facilidad de uso de un entorno Web está completamente relacionada con el diseño del mismo. El sitio debe estar construido de forma tal que los usuarios sepan qué hacer en cualquier momento y cómo hacerlo, facilitándoles el aprendizaje de sus características en el menor tiempo posible.

Inmediatamente al acceder al sitio, el visitante debe conocer los objetivos y alcance del mismo. Suelen ayudar una distribución y diseño gráfico adecuados, así como un lenguaje claro e inteligible lo que influye en el usuario positivamente, promoviéndolo a permanecer en el entorno Web e incitándolo a volver a acceder al mismo. En (Keeker, 2006) se hace referencia a lo importante que resulta que los mejores contenidos o los que puedan despertar mayor interés, queden expuestos de manera visible para que el visitante conozca de su presencia rápidamente.

Diseñar la estructura de la página principal diferenciando visualmente áreas, y agrupar la información por tipo, funcionalidad y prioridad, es también de vital importancia, así como garantizar la claridad, fidelidad y precisión de la información.

Teniendo en cuenta estos aspectos se puede lograr la construcción de una aplicación Web balanceada y atractiva, que facilite a los usuarios la navegación y les conduzca a la satisfacción de alcanzar sus objetivos correctamente.

1.4 Lenguajes de programación para la Web

Actualmente existen múltiples lenguajes de programación para desarrollar en la Web. Estos han ido surgiendo debido a las tendencias y necesidades de los desarrolladores. Desde los inicios de Internet, fueron apareciendo demandas de los usuarios y se dieron soluciones mediante lenguajes estáticos. A medida que pasó el tiempo, las tecnologías fueron desarrollándose y surgieron nuevos problemas a dar solución. Esto dio lugar a desarrollar lenguajes de programación dinámicos para la Web, que permitieran interactuar con los usuarios y utilizar sistemas de bases de datos.

En (Torres, 2006) se habla sobre los lenguajes de programación Web y cómo estos se pueden dividir en dos grupos:

- Lenguajes de lado servidor o lenguajes que son reconocidos, ejecutados e interpretados por el propio servidor y que se envían al cliente en un formato comprensible para él. Entre estos se encuentran: PERL, ASP y PHP.

- Lenguajes de lado cliente (como HTML y Java, JavaScript y CSS, los cuales son incluidos en el código HTML) que representan aquellos lenguajes que pueden ser directamente interpretados por el navegador y no necesitan un tratamiento previo.

A continuación se describen los lenguajes de programación en la Web más utilizados actualmente y que se usarán para el desarrollo del presente sistema automatizado.

1.4.1 HTML

Desde el surgimiento de Internet se han publicado sitios web gracias al lenguaje HTML (*Hyper Text Markup Language*, en español Lenguaje de Marcas Hipertextuales) desarrollado por el *World Wide Web Consortium* (W3C). Este es un lenguaje estático para el desarrollo de sitios web. Los archivos pueden tener las extensiones .htm, .html, son pequeños y de despliegue rápido. Es de fácil aprendizaje y lo admiten todos los exploradores como Internet Explorer, Opera, Firefox, Netscape o Safari.

Además es muy sencillo, está basado en el uso de etiquetas, consistentes en un texto ASCII encerrado dentro de un par de paréntesis angulares(<...>). Donde el manejo de estas permite definir la estructura lógica del documento HTML, aplicar distintos estilos al texto, la inclusión de hiperenlaces, imágenes y ficheros multimedia. También al disponer de múltiples herramientas permite acceder a bases de datos de forma sencilla, es multiplataforma pues funciona tanto para Unix como para Windows (Echeverría, 1995).

Adicionalmente se ha definido un lenguaje de hojas de estilo CSS (*Cascading Style Sheets*) creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. Además, es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas web complejas.

De acuerdo a (Eguíluz Pérez, 2008) separar la definición de los contenidos y la definición de su aspecto presenta numerosas ventajas, ya que obliga a crear documentos HTML/XHTML bien definidos y con significado completo (también llamados "documentos semánticos"). Además, mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferentes.

Al crear una página web, se utiliza en primer lugar el lenguaje HTML/XHTML para marcar los contenidos, es decir, para designar la función de cada elemento dentro de la página: párrafo, titular, texto destacado, tabla, lista de elementos, etc. Una vez creados los contenidos, se utiliza el lenguaje CSS para definir el aspecto de cada elemento: color, tamaño y tipo de letra del texto, separación horizontal y vertical entre elementos, posición de cada elemento dentro de la página, etc.

1.4.2 Javascript

Javascript es un lenguaje que puede ser utilizado por profesionales y para quienes se inician en el desarrollo y diseño de sitios Web. No requiere de compilación ya que el lenguaje funciona del lado del cliente. Los navegadores son los encargados de interpretar estos códigos. Es un lenguaje con muchas posibilidades, utilizado para crear pequeños programas que luego son insertados en una página Web y en programas más grandes, orientados a objetos mucho más complejos. Con Javascript se pueden crear diferentes efectos e interactuar con los usuarios.
En (Valdés, 2007) se enumeran las características de este lenguaje, entre ellas se pueden citar:

- Es un lenguaje basado en acciones que posee menos restricciones.
- Gran parte de la programación en este lenguaje está centrada en describir objetos, escribir funciones que respondan a movimientos del ratón, aperturas, utilización de teclas, cargas de páginas, entre otros.
- Es soportado por la mayoría de los navegadores como Internet Explorer, Netscape, Opera, y Mozilla Firefox.

1.4.3 PHP

Según se puede apreciar en (PHP, 2007) se habla de PHP *(Hypertext Preprocessor)* como un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas Web dinámicas. Es usado principalmente para la interpretación del lado del servidor *(server-side scripting)* pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas.

Entre sus ventajas se encuentran que:

- 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.
- Capacidad de expandir su potencial utilizando módulos (llamados extensiones).
- Posee una amplia documentación en su sitio web oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.

- Contiene una biblioteca nativa de funciones sumamente amplia e incluida.
- No requiere definición de tipos de variables aunque sus variables se pueden evaluar también por el tipo que estén manejando en tiempo de ejecución.

Si bien PHP no obliga a quien lo usa a seguir una determinada metodología a la hora de programar, aun haciéndolo, el programador puede aplicar en su trabajo cualquier técnica de programación o de desarrollo que le permita escribir código ordenado, estructurado y manejable. Un ejemplo de esto son los desarrollos que en PHP se han hecho del patrón de arquitectura Modelo Vista Controlador (MVC), que permiten separar el tratamiento y acceso a los datos, la lógica de control y la interfaz de usuario en tres componentes independientes (Sæther Bakken et al., 2001).

1.5 Servidores de aplicación web

Un servidor Web es un programa que procesa cualquier aplicación del lado del servidor realizando conexiones bidireccionales y/o unidireccionales, síncronas o asíncronas con el cliente generando o cediendo una respuesta en cualquier lenguaje o aplicación del lado del cliente. El código recibido por el cliente suele ser compilado y ejecutado por un navegador web. Para la transmisión de todos estos datos suele utilizarse algún protocolo. Generalmente se utiliza el protocolo HTTP para estas comunicaciones, perteneciente a la capa de aplicación del modelo OSI (*Open System Interconnection*).

1.5.1 Apache

Uno de los servidores de aplicación Web existentes es Apache. Debido a las características y ventajas que ofrece, ha sido el seleccionado para el desarrollo de la presente aplicación; además de ser un requisito del cliente y porque forma parte de la política de informatización de esta organización. A continuación se hace una descripción de este.

Apache es un servidor Web HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1. Se desarrolla dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation (APACHE, 2012).

Según (Ruiz, 2007) , Apache es el servidor web más difundido y utilizado en Internet. Tal vez lo que lo hace más atractivo es su alta estabilidad, seguridad y facilidad de expansión. Debido a que es software libre, quienquiera que desee descargarlo no tendrá que pagar costosas licencias para su utilización, pudiendo disponer de su código fuente para modificar y adaptar el servidor a sus necesidades.

Entre sus ventajas más específicas se encuentran que:

- Es modular: Puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo que proporciona, y con la API de programación de módulos, para el desarrollo de módulos específicos.
- Código abierto.
- Multiplataforma.
- Extensible: gracias a ser modular se han desarrollado diversas extensiones entre las que destaca PHP, un lenguaje de programación del lado del servidor.

1.6 Gestores de Bases de Datos

Un gestor de base de datos o Sistema de Gestión de Base de Datos (SGBD) es un software que permite administrar la información de las bases de datos. Son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan; por tanto cualquier operación que el usuario hace contra la base de datos está controlada por el gestor.

Se trata de un conjunto de programas no visibles al usuario final que se encargan de la privacidad, la integridad, la seguridad de los datos y la interacción con el sistema operativo. Algunos ejemplos de SGBD son Oracle, DB2, PostgreSQL, MySQL, MS SQL Server, etc.
Entre sus características generales se encuentran:

- Aceptan definiciones de esquemas y vistas (definición de diferentes bases de datos).
- Manipulan los datos siguiendo las órdenes de los usuarios.
- Cuidan que se respete la seguridad e integridad de los datos.
- Permiten definir usuarios y las restricciones de acceso para cada uno de ellos.
- Controlan la concurrencia y las operaciones asociadas a la recuperación de los fallos.

1.6.1 Microsoft SQL Server 2000

De acuerdo a (Microsoft, 2011) Microsoft SQL Server 2000 es parte de la familia de productos diseñados para mantener almacenar y procesar datos para sistemas nativos de Windows y sitios Web, puede ser usado por grandes compañía. Provee servicios fundamentales como el tradicional cliente/servidor, el cual alcanza un correcto desempeño y es altamente configurable en cuanto a la ingeniería de las base de datos relacionales.

Algunas de las características de este SGBD:

- Soporte de transacciones.
- Soporta procedimientos almacenados.
- Incluye un potente entorno gráfico que permite el uso de comandos de lenguajes de definición y manipulación de datos.
- Permite administrar información de otros servidores de datos.

1.7 Marcos de Trabajo

Se han realizado importantes progresos orientados a la reusabilidad del software a través de la aplicación del paradigma de la programación orientada a objetos y mediante el uso de los componentes tecnológicos. Sin embargo, estas tecnologías solo proveen el re-uso en el nivel individual, frecuentemente a menor escala. Con la aparición de los patrones de diseño se ha demostrado la reutilización de soluciones para resolver problemas de escala mayor enfocados en la solución de problemas sencillos. Sin embargo, el más complejo problema de la reutilización de soluciones es en el nivel de los grandes componentes, para que estos se puedan adaptar para las solicitudes individuales.

Recientemente, el interés por la reutilización de software ha pasado de la reutilización de los componentes de forma independiente a la de todo el diseño de un sistema o la estructura de una aplicación. Un sistema de software que es capaz de permitir la reusabilidad a este nivel, permitiendo la realización de aplicaciones completas es llamado Marco de Trabajo *(framework*). Un *framework* debe estar basado en la idea de facilitar la producción de un conjunto de sistemas específicos pero similares, dentro de una estructura genérica. Además, los marcos de trabajo pueden contener sub marcos de trabajo los cuales representen subconjuntos de componentes de

un sistema más grande. Está formado por componentes personalizables e intercambiables para el desarrollo de una aplicación.

Entre los objetivos del *framework* se encuentran:

- Desarrollo rápido y estructurado.
- Reutilización de código.
- Disminuir el esfuerzo en el desarrollo.
- Aprovecha las funcionalidades ya implementadas.

En la actualidad existen una gran cantidad de *frameworks* para el desarrollo de soluciones web, entre ellos cabe mencionar por su gran relevancia a CakePHP, Zend, Kumbia, CodeIgniter, Symfony entre otros.

CakePHP es un *framework* de desarrollo rápido de aplicaciones de código abierto en PHP. Inspirado en *Rails*, para la construcción de sitios web que utilizan una base de datos como fuente de recursos, posee una infraestructura que tiene como finalidad permitir el desarrollo de aplicaciones web de manera ágil y estructurada, sin perder flexibilidad (Anderson and Masters, 2006).

Zend Framework (ZF) es un *framework* de código abierto para desarrollar aplicaciones web y servicios web con PHP 5. ZF es una implementación que usa código 100% orientado a objetos. La estructura de los componentes de ZF es algo único; cada componente está construido con una baja dependencia de otros componentes. Esta arquitectura débilmente acoplada permite a los desarrolladores utilizar los componentes por separado. Facilita la programación a través del patrón MVC y automatiza tareas más específicas, como el acceso a base de datos, el filtrado de datos ingresados a la aplicación o la búsqueda en un sitio web ordenando los resultados por relevancia (Chase, 2006).

Kumbia es un *framework* para aplicaciones web libre escrito en PHP5. Kumbia fomenta la velocidad y eficiencia en la creación y mantenimiento de aplicaciones web, reemplazando tareas de codificación repetitivas. Kumbia intenta proporcionar facilidades para construir aplicaciones robustas para entornos comerciales. Esto significa que el marco de trabajo es muy flexible y configurable (Kumbia, 2012).

CodeIgniter es un entorno de desarrollo abierto que permite crear webs dinámicas con PHP. Su principal objetivo es ayudar a que los desarrolladores, puedan realizar proyectos mucho más rápido que creando toda la estructura desde cero (EllisLab, 2012).

En el anexo 2 se hace una tabla comparativa sobre estos *frameworks*. A continuación se analizará Symfony como *framework* de desarrollo escogido para el desarrollo de la aplicación de las reservaciones en los restaurantes del hotel Iberostar Ensenachos.

1.7.1 Symfony

De acuerdo a (Potencier and Zaninotto, 2008) los creadores de Symfony, este es un marco de trabajo PHP que facilita el desarrollo de las aplicaciones web. Se encarga de todos los aspectos comunes de las aplicaciones web, dejando que el programador se dedique a aportar valor desarrollando las características únicas de cada proyecto. Este marco de trabajo ayuda a aumentar la productividad y mejorar la calidad de las aplicaciones web concebidas mediante su uso.

Sus características principales se resumen en:

- Fácil de instalar y configurar en sistemas Windows, Mac y Linux.
- Funciona con todas las bases de datos más comunes (MySQL, PostgreSQL, SQLite, Oracle, Microsoft SQL Server).
- Compatible solamente con PHP 5, para asegurar el mayor rendimiento y acceso a las características más avanzadas de este lenguaje.
- Preparado para aplicaciones empresariales, ya que se puede adaptar con facilidad a las políticas y arquitecturas propias de cada empresa u organización.
- Flexible y extensible mediante un completo mecanismo de plugins.
- La vista usa plantillas (*templates* y *layount*) que pueden ser construidas por diseñadores Web que no posean conocimientos de este marco de trabajo.
- Publicado bajo licencia MIT de software libre y apoyado por una empresa comprometida con su desarrollo.
- Los formularios soportan la validación automática, lo cual asegura mejor calidad de los datos en la base de datos, ofreciendo una mejor experiencia para el usuario.
- El manejo de caché reduce el uso de banda ancha y la carga del servidor.

- La facilidad de soportar autenticación y credenciales facilita la creación de áreas restringidas y manejo de seguridad de los usuarios.
- Traducido a más de 40 idiomas y fácilmente se puede llevar a cualquier otro idioma (Eguíluz Pérez, 2011).

1.7.2 Buenas razones para escoger Symfony

Existen un conjunto básico de razones que hacen de Symfony un marco de trabajo importante para el desarrollo de aplicaciones Web, estas son:

- Gran reputación pues es un entorno estable y reconocido a nivel internacional. Symfony también posee una comunidad activa, compuesta en su mayoría de desarrolladores, integradores, usuarios y otros colaboradores que participan en el enriquecimiento continuo de esta herramienta.
- Detrás de Symfony existe una empresa fundada por su creador: Sensio. Creado hace más de 12 años, Sensio es una agencia web concebida para sus propias necesidades, el marco de trabajo Symfony hoy en día sigue siendo la herramienta de uso diario por sus propios equipos para desarrollar proyectos de clientes. Con miras a un desarrollo sostenible Symfony se distribuye bajo licencia MIT Open Source, que no impone restricciones y permite el desarrollo de código abierto, así como las aplicaciones propietarias.
- Cuenta con una gran cantidad de referencias, entre ellas contamos con intranets corporativas, redes sociales, sitios de la comunidad, sitios para el público en general, etc. Ejemplos no faltan como son en el caso de Yahoo, Dailymotion, Opensky.com, Exercise.com e incluso aplicaciones como phpBB y Drupal (Eguíluz Pérez, 2011).

1.7.3 Symfony y el patrón de arquitectura MVC

Symfony está basado en un patrón clásico del diseño web conocido como arquitectura MVC (Modelo-Vista-Controlador), que está formado por tres niveles:

- El Modelo representa la información con la que trabaja la aplicación, es decir, su lógica de negocio.
- La Vista transforma el modelo en una página web que permite al usuario interactuar con ella.

- El Controlador se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista.

La arquitectura MVC separa la lógica de negocio (el controlador), los datos (el modelo) y la presentación (la vista) por lo que se consigue un mantenimiento más sencillo de las aplicaciones. Por tanto, una misma aplicación que se debe ejecutar tanto en un navegador estándar como en un navegador de dispositivo móvil, solamente necesita crear una vista nueva para cada dispositivo; manteniendo el controlador y el modelo original. Ver figura 1.1

El controlador se encarga de aislar al modelo y a la vista de los detalles del protocolo utilizado para las peticiones (HTTP, consola de comandos, email, etc.). El modelo se encarga de la abstracción de la lógica relacionada con los datos, haciendo que la vista y las acciones sean independientes de, por ejemplo, el tipo de gestor de bases de datos utilizado por la aplicación.

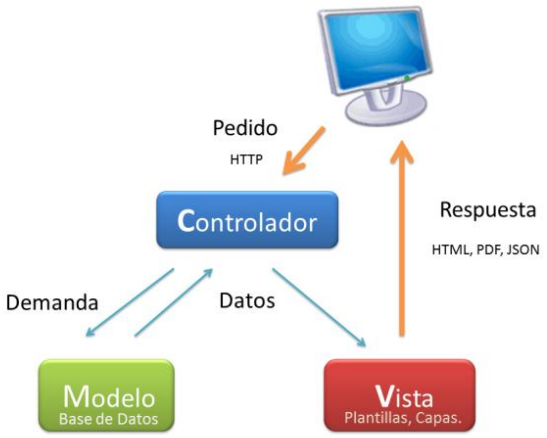


Figura 1.1. Arquitectura MVC

El principio más importante de la arquitectura MVC según (Potencier and Zaninotto, 2008), es la separación del código del programa en tres capas, dependiendo de su naturaleza. La lógica relacionada con los datos se incluye en el modelo, el código de la presentación en la vista y la lógica de la aplicación en el controlador. La programación se puede simplificar si se utilizan otros patrones de diseño. De esta forma, las capas del modelo, la vista y el controlador se pueden subdividir en más capas.

1.7.4 Doctrine

Doctrine es un potente y completo sistema ORM (*Object Relational Mapper*) para PHP 5.2+ con un DBAL (*Data Base Abstraction Layer*) incorporado, viene por defecto con Symfony. Tiene la posibilidad de exportar una base de datos existente a sus clases correspondientes y también a la inversa, es decir convertir clases (convenientemente creadas siguiendo las pautas del ORM) a tablas de una base de datos. Por otro lado, como la biblioteca es bastante grande ésta tiene un método para ser compilada al pasar a producción.

Además de todas las ventajas que conlleva un ORM, uno de sus puntos fuertes es su lenguaje DQL (*Doctrine QueryLanguage*).Inspirado en el HQL de Hibernate, este lenguaje permite escribir opcionalmente consultas a la base de datos en un lenguaje OO (Orientado a Objetos) brindando a los desarrolladores una poderosa alternativa al SQL manteniendo una máxima flexibilidad sin necesidad de duplicar código (SensioLabs, 2009).

1.8 Herramientas de desarrollo de software

Las herramientas de desarrollo de software, consolidadas en la literatura en la forma de Ingeniería de Software asistida por computadora (*Computer Aided Software Engineering*), suponen la asistencia en el proceso de ciclo de vida del software, contribuyendo a mejorar la calidad y la productividad en el desarrollo de sistemas de información y se plantean los siguientes objetivos:

- Permitir la aplicación práctica de metodologías estructuradas, las cuales al ser realizadas con una herramienta se consigue agilizar el trabajo.
- Facilitar la realización de prototipos y el desarrollo conjunto de aplicaciones.
- Simplificar el mantenimiento de los programas.
- Mejorar y estandarizar la documentación.
- Aumentar la portabilidad de las aplicaciones.
- Facilitar la reutilización de componentes de software.
- Permitir un desarrollo y un refinamiento visual de las aplicaciones, mediante la utilización de gráficos (Rivas et al., 2006).

1.8.1 NetBeans IDE 7.01

NetBeans IDE es un entorno de desarrollo, una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito en Java, pero puede servir para muchos otros lenguajes de programación. Existe además un número importante de módulos para su extensión. Es un producto libre y gratuito sin restricciones de uso.

NetBeans IDE 7.01 contiene las herramientas para que los desarrolladores de software puedan crear aplicaciones desktop, empresariales, web, y aplicaciones móviles, con el lenguaje Java, así como también haciendo uso de C/C++, PHP, JavaScript, Groovy, y Ruby. Esta versión permite crear e importar proyectos Symfony, así como utilizar la consola de comandos de este marco de trabajo. Proporciona soporte para crear aplicaciones añadiéndoles flags o parámetros y posibilita expandir o minimizar las secciones de los archivos YAML.

1.8.2 Biblioteca Jquery

La biblioteca Jquery (Alvarez and Nebot, 2008) provee una capa de abstracción de propósito general para su uso general en los lenguajes script de la web, y es por consiguiente útil en casi todas las situaciones donde usamos scripts. Está en constante cambio por sus desarrolladores quienes se encargan de añadirles nuevas funcionalidades. Los cambios introducidos por esta biblioteca están fundamentalmente vinculados a las siguientes necesidades: acceder, modificar la apariencia, alterar y agregar animaciones el contenido de la página web, responder las interacciones que los usuarios realizan con esta, recibir información del servidor sin necesidad de refrescar completamente la página (esto es conocido como AJAX (Asleson and Schutta, 2006)) entre otras necesidades (Chaffer, 2007).

Por lo anteriormente descrito resulta ventajoso el uso de esta herramienta para el desarrollo de la presente aplicación.

1.8.3 Visual Paradigm

Visual Paradigm para UML es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de

aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación.

Esta herramienta posee diferentes características entre las que se encuentran: Soporte de UML versión 2.1, ingeniería inversa, generación de código, soporte ORM (generación de objetos Java desde la base de datos), transformación de diagramas de Entidad-Relación en tablas de base de datos, integración IDE con Visual Studio, NetBeans y otros.

1.8.4 Artisteer 3.0

Artisteer es un programa que permite la creación de plantillas de sitios web o para la mayoría de los Sistemas de Gestión de Contenidos (*Content Management System*) como Wordpress, Drupal o Joomla. Tiene una simple e intuitiva interfaz por lo que resulta de fácil uso. Genera el código XHTML, CSS y javascript en conformidad con estándares web. Soporta los formatos: ARTX, HTML, JPG, PNG, GIF, y puede ser encontrado en varios idiomas (ExtendSoft, 2012).

1.9 Conclusiones Parciales

Existen numerosos software para la Gestión Hotelera. La mayoría son propietarios y desarrollados en terceros países, por lo que su incorporación en el sistema hotelero en Cuba conllevaría un gran esfuerzo y costos financieros. No todos cuentan con un apartado para la manipulación de las reservaciones en los restaurantes.

El desarrollo de aplicaciones Web combina una variedad de tecnologías que van desde su modelación hasta su implementación. Una combinación razonable para desarrollar una aplicación Web puede ser aquella que utiliza los lenguajes antes descritos, haciendo énfasis en PHP para el lado del servidor y una combinación de HTML, JavaScript y CSS para el cliente.

CAPÍTULO II

“CARACTERÍSTICAS DEL SISTEMA”

CAPÍTULOII: "*Características del sistema*"

En el presente capítulo se describe la propuesta de solución como una Aplicación Web para la Reserva en los Restaurantes (ARR 1.0). El objetivo fundamental es mostrar a través de los diferentes modelos: del negocio, casos de uso, diseño, entre otros, la solución desarrollada y la atención a las diferentes dificultades mencionadas en el epígrafe 1.2.

2.1 Modelación del Negocio

El modelado de casos de uso es un método para identificar las necesidades funcionales de un nuevo sistema. A continuación se llevará a cabo el modelado del negocio para comprender mejor el proceso de reservas para un restaurante y de esta manera elaborar la propuesta de solución.

2.1.1 Diagrama de Casos de Uso del Negocio

En la figura 2.1 se muestra el Diagrama de Casos de Usos del Negocio (CUN) en el cual se presentan los procesos del negocio determinados a partir de entrevistas a los participantes principales: Director, Jefe AB, Conserjes e Informático del Hotel.

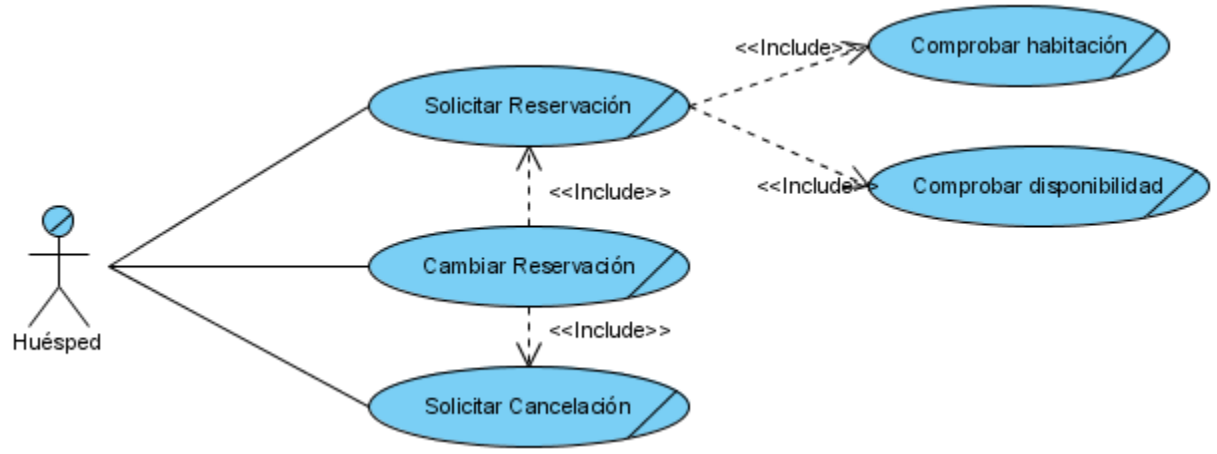


Figura 2.1 Diagrama de Casos de Uso del Negocio.

2.1.2 Descripción literal de los casos de uso del negocio

A continuación se ofrece una descripción de los casos de uso más representativos identificados en el negocio, como base para la modelación e implementación del Software para la reserva en los restaurantes del hotel Iberostar Ensenachos.

2.1.2.1 Caso de Uso: Solicitar reservación.

El caso de uso de Solicitar reservación comienza cuando un huésped realiza una solicitud de reserva para ir a cenar durante su estancia. El conserje que lo atiende le solicita el número de su habitación y luego se procede a identificar la fecha, el restaurante y el horario preferido por cliente. Se realiza una verificación para comprobar si al huésped le quedan posibilidades de ir a cenar a un restaurante especializado, según su contrato. De ser así, se chequea si existen capacidades disponibles sobre los datos introducidos. De ser positivo, se informa al huésped del éxito de la operación. De no poder reservar más este cliente, pues ha consumido todas sus posibilidades de acuerdo al contrato, se le informa esto al conserje y es este el que toma la decisión de reservar o no. De no existir capacidades para ese restaurante en ese horario con esa fecha, se le informa al cliente y si este está de acuerdo se procede a buscar otro lugar.

2.1.2.2 Caso de Uso: Solicitar cancelación.

El caso de uso Solicitar cancelación comienza cuando el cliente solicita realizar la cancelación de alguna de sus reservaciones. El conserje que lo atiende le solicita la habitación en la que se encuentra hospedado y verifica su nombre, de ser incorrectos se procede a la cancelación del proceso. En caso de ser correcto el nombre proporcionado con la habitación se procede a obtener los datos necesarios para ubicar la reservación con la cual se procederá a la cancelación. Una vez localizada la reservación se procederá a su cancelación.

2.1.2.3 Caso de Uso: Cambiar reservación

El caso de uso Cambiar reservación comienza cuando el huésped solicita un cambio de reservación ante el conserje, como consecuencia de alguna necesidad personal. Se procede a la solicitud de la habitación del cliente y se comprueba que el nombre sea el correcto. De ser

positiva la información se procede a obtener los restantes datos necesarios para proceder a la cancelación de la reserva. En caso negativo se procede a la cancelación de la operación.

2.2 Requisitos del Sistema

Los requisitos de Sistema comprenden todas las tareas relacionadas con la determinación de las necesidades o de las condiciones a satisfacer para un software nuevo o modificado, tomando en cuenta los diversos requisitos de los participantes en el proyecto.

2.2.1 Requisitos Funcionales

Según (Jacobson et al., 2004) se define: “los requisitos funcionales son aquellos que especifican una acción que debe ser capaz de realizar el sistema, sin considerar restricciones físicas; requisito que especifica comportamiento de entrada/salida de un sistema”.

De esta manera los requisitos funcionales para la reserva en los restaurantes del hotel Iberostar Ensenachos son:

RF1: Autenticar usuario

1.1 Introducir nombre de usuario y contraseña al sistema.

1.2 Validar datos introducidos del usuario.

RF2: Gestionar usuario.

2.1 Crear un usuario según el rol deseado.

2.2 Listar todos los usuarios.

2.2 Modificar un usuario.

2.3 Eliminar un usuario.

RF3. Gestionar Restaurante.

3.1 Crear Restaurante.

3.2 Listar Restaurante.

3.3 Modificar Restaurante.

3.4 Abrir o Cerrar Restaurante.

RF4. Gestionar Horario.

4.1 Crear Horario.

4.2 Listar Horario.

4.3 Habilitar o Deshabilitar Horario.

R5. Gestionar Mesas.

5.1. Adicionar Mesas.

5.2. Listar Mesas.

5.3 Modificar Mesas

R6. Gestionar Planificaciones.

6.1. Adicionar Planificaciones.

6.2. Listar Planificaciones.

6.3. Modificar Planificaciones.

6.4. Cancelar Planificaciones.

RF7: Gestionar operaciones.

7.1 Consultar las operaciones realizadas en los registros del sistema.

7.2 Eliminar por intervalos de tiempo.

R8. Gestionar Reservaciones.

8.1. Listar Reservaciones.

8.2. Crear Reservaciones.

8.4. Adjuntar Habitación.

8.3. Cancelar Reservaciones.

R9: Gestionar reportes.

9.1 Obtener reporte sobre la cantidad de reservaciones y cancelaciones mensuales.

9.2 Obtener reporte sobre la cantidad de reservaciones por cada restaurante.

9.3 Obtener reporte sobre la cantidad de reservaciones por turnos.

R10. Gestionar Búsquedas.

10.1. Buscar las reservaciones de un cliente.

10.2. Buscar las cancelaciones de un cliente.

10.3. Buscar las reservaciones de una fecha.

10.4. Buscar las cancelaciones de una fecha.

10.5. Buscar las reservaciones de un restaurante.

10.6. Buscar las cancelaciones de un restaurante.

10.7. Buscar las reservaciones de un turno.

10.8. Buscar las cancelaciones de un turno.

10.9. Buscar las reservaciones usando combinaciones de fecha, restaurante, turno.

10.10. Buscar las reservaciones usando combinaciones de fecha, restaurante, turno.

2.2.2 Requisitos no funcionales

Los requisitos no funcionales son aquellos que especifican propiedades del sistema, como restricciones del entorno o de implementación, rendimiento, dependencias de la plataforma, mantenibilidad, extensibilidad o fiabilidad. Requisitos que especifican restricciones físicas sobre un requisito funcional (Jacobson et al., 2004). Para la nueva Aplicación de Reservas son:

1. La aplicación debe ser fácil de usar por lo que debe mantener características similares al anterior SRR 11.1.
2. La aplicación debe ser interactiva, con una interfaz uniforme en cuanto a colores y texto, logrando identificarla como un mismo sistema.
3. El sistema impondrá un estricto control de acceso que permitirá a cada usuario tener disponible solamente las opciones relacionadas con su actividad.
4. El sistema dispondrá de una ayuda bien detallada sobre las principales opciones, facilitando así el trabajo en la aplicación.
5. El gestor de base de Datos empleado será Microsoft SQL Server 2000 debido a la implementación de la base de datos ZunPMSIberostar, de donde se obtienen los datos de los huéspedes del hotel.
6. El sistema operativo empleado será Windows, en su versión server 2003 o superior, siempre que mantenga compatibilidad con el gestor de base de datos empleado.

2.3 Modelo de casos de uso

El modelo de casos de uso permite que los desarrolladores de software y los clientes lleguen a un acuerdo sobre los requisitos, las condiciones y posibilidades que debe cumplir el sistema. Propicia la entrada fundamental para el análisis, el diseño y las pruebas.

El modelo de casos de uso queda aquí representado por la definición de los actores del sistema, el diagrama de CU del sistema y la especificación de los CU principales.

2.3.1 Actores del sistema

Un actor es una entidad externa que interactúa con el sistema participando en un caso de uso. Los actores pueden ser personas reales (por ejemplo, usuarios del sistema), otros sistemas o eventos

externos. Los actores no representan a personas físicas o a sistemas, sino su rol. En la Tabla 1 se describen los actores propuestos para el sistema.

Tabla 2.1.Actores del sistema.

Actor	Justificación
Jefe AB	Es el encargado de gestionar el funcionamiento de los restaurantes. Planifica el uso de estos, los horarios y las diferentes capacidades en los mismos. Gestiona a los usuarios del sistema.
Comercial	Es el encargado de gestionar la cantidad de reservaciones a los huéspedes según el contrato en dependencia de la agencia por la que viajan y la nacionalidad de estos.
Conserje	Es el encargado de realizar los procesos de reservaciones y cancelaciones.
Consultante	Es el que solamente puede consultar la información de las reservaciones y/o cancelaciones y obtener reportes de los mismos.
Zun	Actor pasivo. Es el sistema de la gestión hotelera. Provee la información de los huéspedes.

2.4 Diagrama de Casos de Uso del sistema

Los diagramas de casos de uso muestran el comportamiento del sistema a partir de los usuarios que interactúan con él. Describe gráficamente quién utiliza el sistema y la forma en que los usuarios esperan interactuar con él (Sparks, 2006). En la figura 2.2 se muestra el diagrama de casos de uso propuesto para la aplicación Web Reservas en Restaurantes en el hotel Iberostar Ensenachos; en lo adelante ARR 1.0.

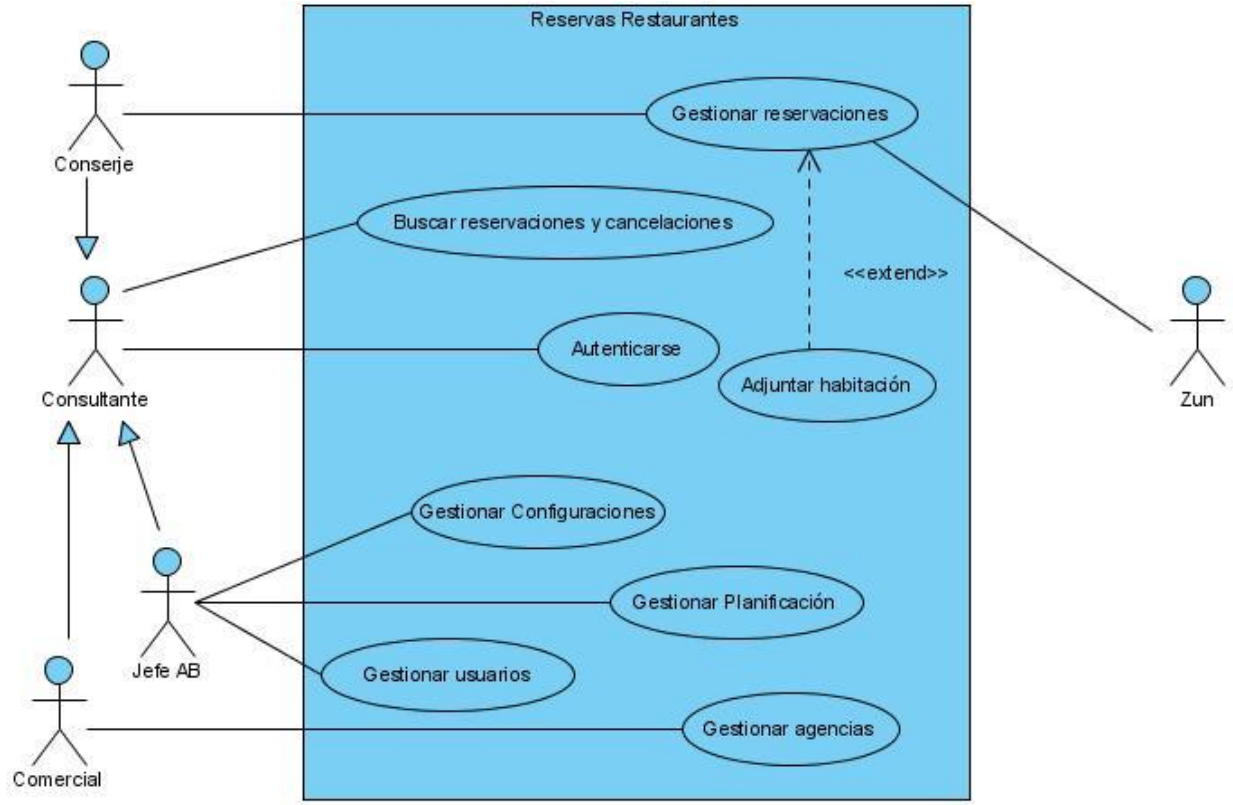


Figura 2.2. Diagrama de casos de Uso del Sistema.

2.4.1 Especificación de los casos de uso

Las narraciones de casos de uso describen de forma escrita los eventos de negocio y cómo interaccionan los usuarios con el sistema. A continuación se describirán algunos de los casos de uso más significativos del sistema. Ver Tabla 2.2.

Tabla 2.2 Descripción del caso de uso Gestionar reservaciones.

Caso de uso	Gestionar Reservaciones
Actor	Conserje
Resumen	El caso de uso se inicia cuando el actor selecciona la opción Reservaciones. Dentro de esta aparecen los listados de todas las reservaciones para el día actual, según el restaurante y el horario específico. El actor a petición del huésped, puede realizar una nueva reservación o cancelar una existente.
Precondiciones	El sistema se encuentra disponible y el Conserje se encuentra autenticado.

Referencia	RF8

Prototipo de Interfaz

Flujo Normal de eventos

Acción del actor	**Respuesta del Sistema**
1- El actor selecciona la opción Reservaciones en el menú vertical. A 3- El actor selecciona la opción: a) Nueva reservación C b) Cancelar reservación D c) Adjuntar habitación E	2- El sistema muestra las reservaciones correspondientes al día actual para los restaurantes y sus respectivos horarios. Brinda información sobre la cantidad de mesas planificadas, aquellas que se encuentran ocupadas y libres. B

Prototipo de Interfaz

Escenario "Nueva Reservación"

5 - El actor llena los campos y oprime el botón Guardar. G	4 - El sistema muestra una pantalla con los campos a escoger para crear la nueva reservación. F 6- El sistema valida que los datos enviados sean correctos. 7- El sistema muestra mensaje. H

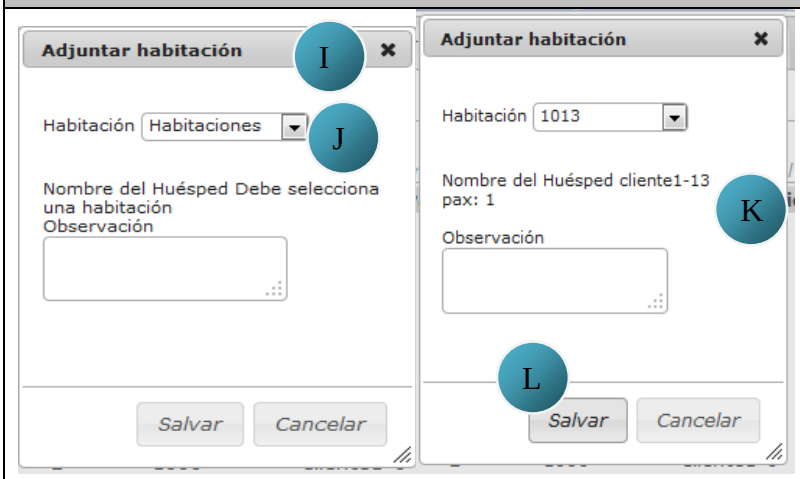
Prototipo de Interfaz

Se ha guardado correctamente la reservacion con fecha: 2012-06-15 en el restaurante: Punta Pirata para el turno: 18:00 en una mesa de: 2 H

Escenario "Adjuntar habitación"

5- El actor escoge de la habitación correspondiente de la lista que se le ofrece. J 7- El actor hace clic sobre la opción Salvar. L	4- El sistema muestra la ventana de adjuntar habitación. I 6- El Sistema muestra el nombre del cliente y la cantidad de acompañantes. K 8- El sistema notifica el resultado de la operación como una nueva reservación. Ver escenario Nueva Reservación.

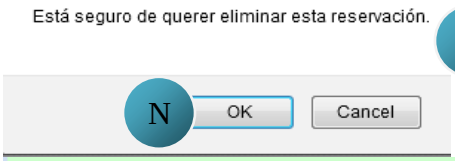
Prototipo de Interfaz



Escenario "Cancelar Reservación"

5- El actor hace clic en la opción OK N	4- El sistema pide confirmación antes de cancelar M 6- El sistema notifica el resultado de la operación. O

Prototipo de Interfaz



Se ha cancelado la reservacion del cliente: 9020 cliente4-20 en el restaurante: Punta Pirata en el turno: 18:00 para la fecha: 2012-06-15 O

Cursos Alternos

1- Escenario "Nueva Reservación" línea 4.

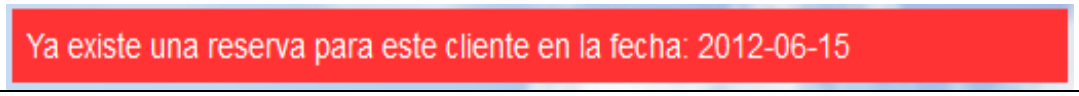
Si se escoge incorrectamente un restaurante el sistema muestra el siguente error:

Prototipo de Interfaz

2- Escenario "Nueva Reservación" línea 6.

Si el cliente posee otra reservación para la fecha seleccionada, se muestra el siguiente error:

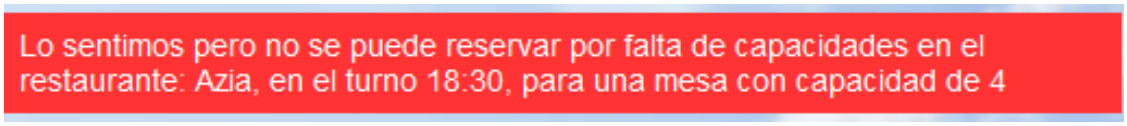
Prototipo de Interfaz



3- Escenario "Nueva Reservación" línea 6.

Si no existe capacidad para la fecha, el restaurante, el turno y el pax seleccionado, se muestra el siguiente mensaje de error:

Prototipo de Interfaz



4- Escenario "Nueva Reservación" línea 6.

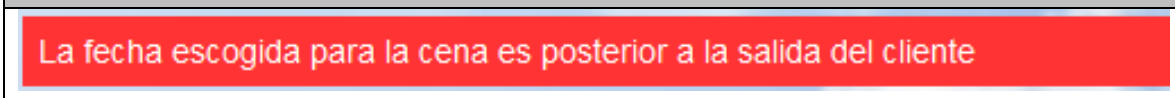
Si el cliente ha agotado todas las posibilidades de reservaciones según su contrato, se muestra el siguiente mensaje:

Prototipo de Interfaz

5- Escenario “Nueva Reservación” línea 6.

Si la fecha escogida es posterior a la salida del cliente se muestra el siguiente mensaje de error:

Prototipo de Interfaz



2.5 Modelo de Diseño

El propósito del diseño es especificar una solución que trabaje y pueda ser convertida en código fuente y construir una arquitectura simple y fácilmente extensible. El modelo de diseño representa los componentes de la aplicación y determina su colocación adecuada y su uso dentro de la arquitectura en general.

2.5.1 Arquitectura general del sistema

La figura 2.3 representa el diseño general del sistema desde la perspectiva de la arquitectura MVC, que es la definida para este desarrollo. Se propone el trabajo con seis subsistemas fundamentales:

- Gestionar Búsquedas.
- Gestionar Reservaciones.
- Gestionar Agencias.
- Gestionar Planificaciones.
- Gestionar Configuraciones.
- Gestionar Usuarios.

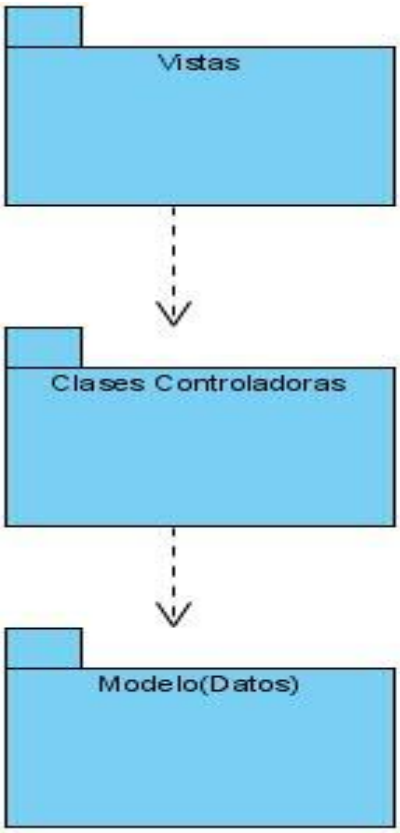


Figura 2.3. Diseño del sistema según la arquitectura MVC.

Una vista general de cómo puede representarse el subsistema Gestionar de reservaciones utilizando el patrón MVC queda como sigue en la figura 2.4:

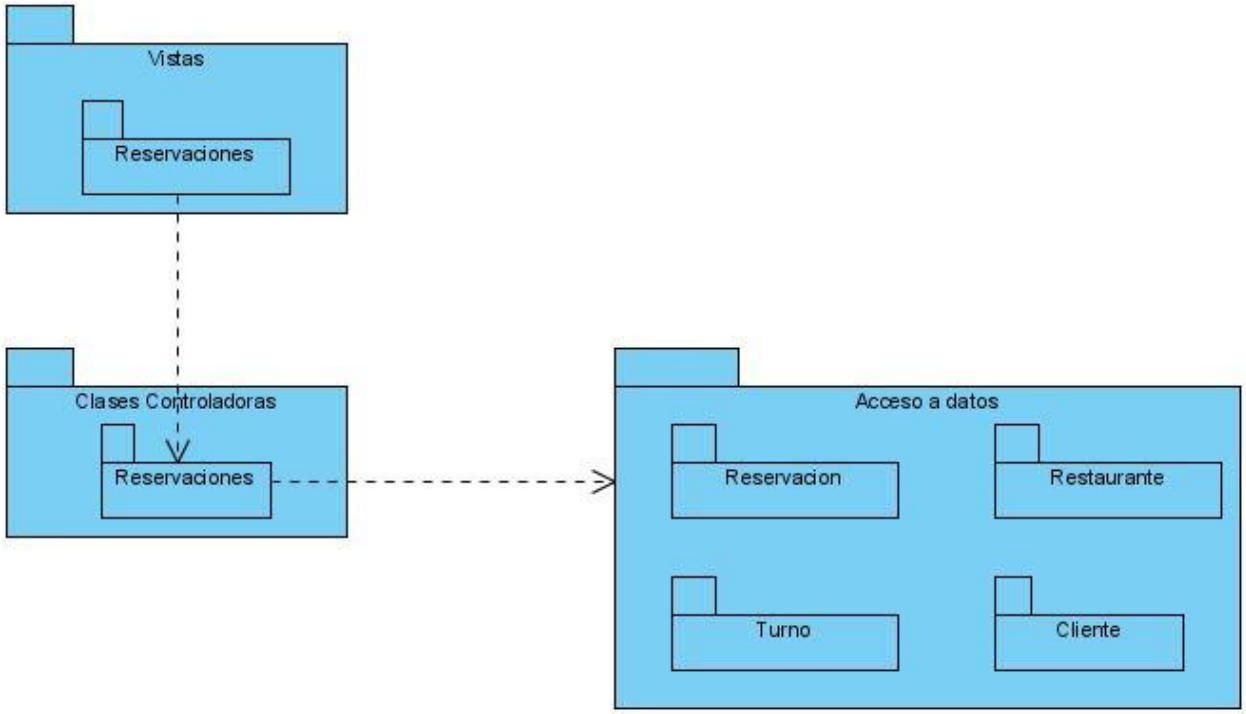


Figura 2.4. Subsistema Gestionar reservaciones.

2.5.2 Diagrama de clases del Diseño

Este diagrama en un modelo estático con el que se pueden mostrar las clases y sus jerarquías y enlaces, atributos, operaciones y asociaciones (Stumpf and Teague, 2005).

Una porción del diagrama de clases de la aplicación que representa las acciones más importantes para las reservaciones desde el punto de vista de la arquitectura MVC se muestra a continuación en la figura 2.5.

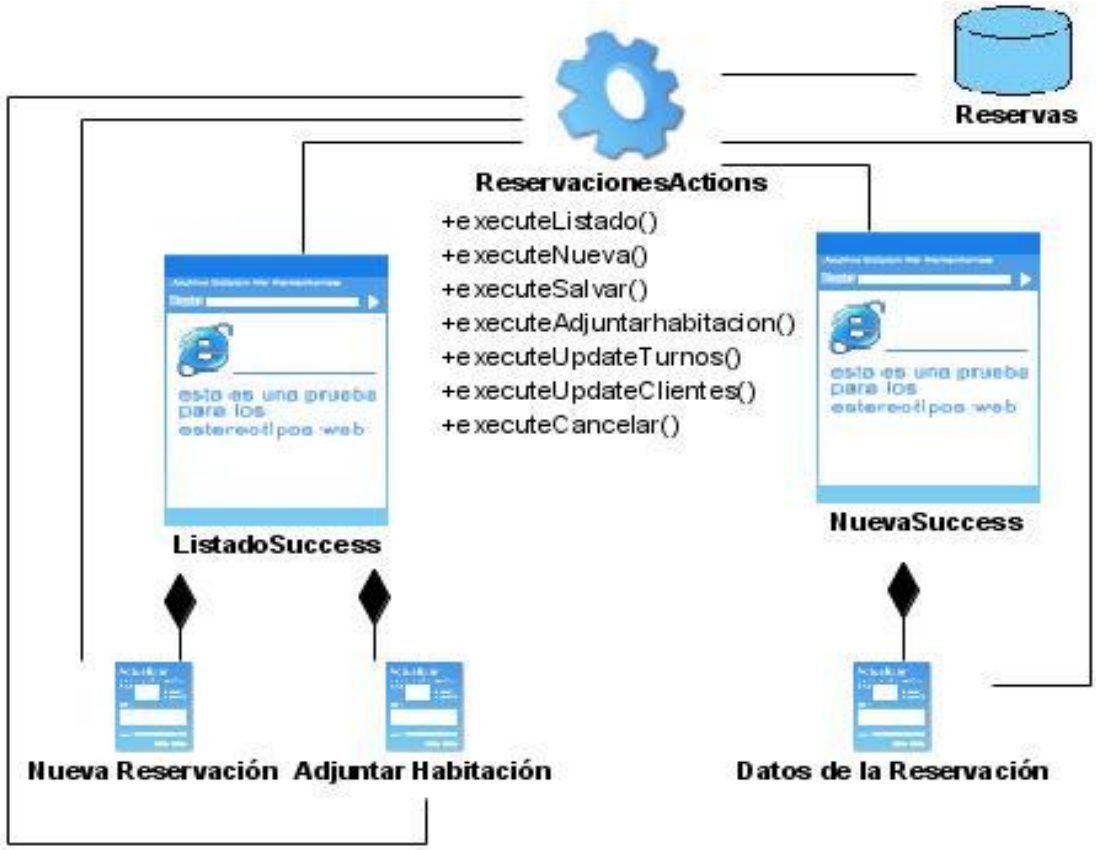


Figura 2.5. Diagrama de clases para el CU Gestionar reservación

2.5.3 Mapa de Navegación

Este mapa permite visualizar la estructura general del Sistema, ilustrando el resultado de las vías de navegación directa entre cada una de sus páginas. El mapa de navegación se muestra en la figura 2.6.

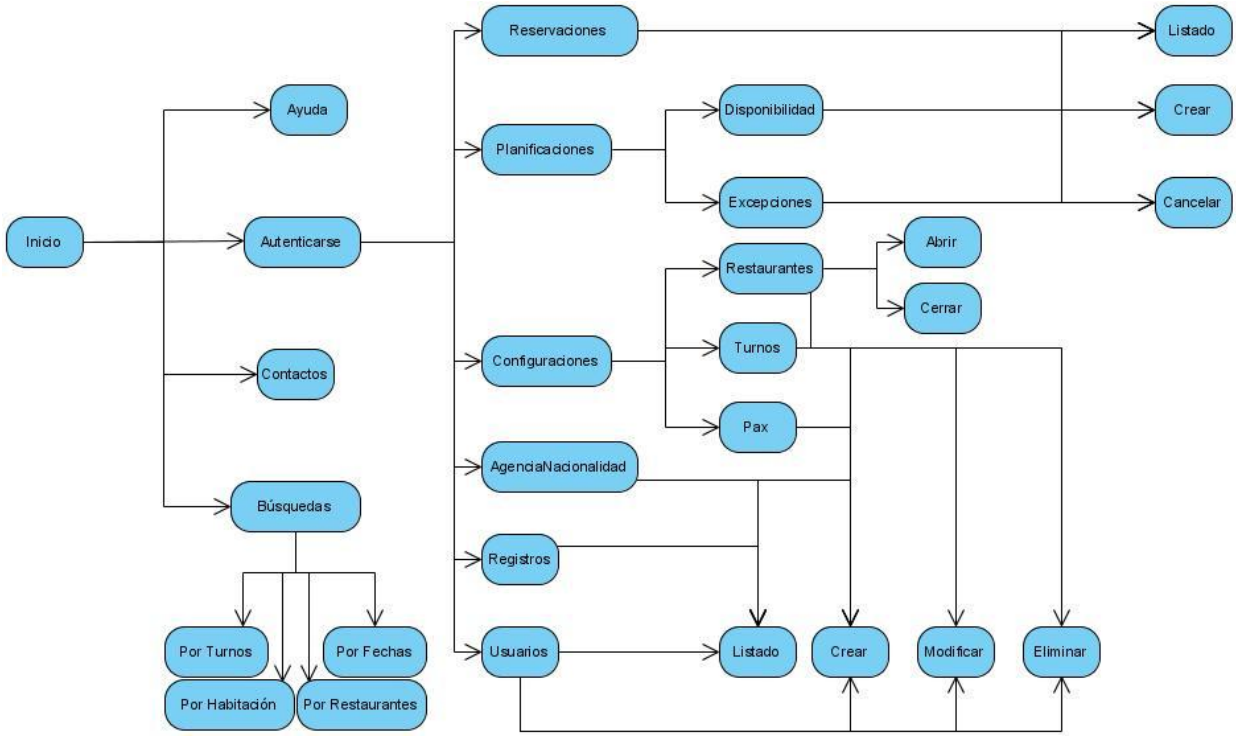


Figura 2.6.Mapa de navegación.

2.6 Modelo de Datos

Un modelo de datos es un lenguaje orientado a describir una Base de Datos, este permite describir los elementos de la realidad que intervienen en un problema dado y la forma en que se relacionan esos elementos entre sí, intentan reproducir información real que se desea almacenar en un sistema informático.

2.6.1 Modelo de Datos Conceptual (Diagrama Entidad-Relación)

Este modelo permite ver con claridad la información utilizada en el problema, pues está orientado a representar los elementos que intervienen en este y sus relaciones. En el caso de estaaplicación, se hizo un análisis del sistema anterior SRR 11.1 debido a las inconsistencias presentadas por su base de datos, pues almacena una gran cantidad de campos nulos, existen muchos atributos en la tabla reservas para hacer alusión a la habitación de los clientes, los nombres de los atributos de la tabla no son suficientemente esclarecedores, etc.

En la figura 2.7 se puede apreciar la modelación de la nueva base de datos.

2.6.2 Modelo de datos Físicos

La base de Datos es el sistema que permite almacenar toda la información referente a una actividad específica y controlar el acceso a los datos que están almacenados. En la figura 2.8 se muestra el diseño de la base de datos para el Sistema que se desea realizar para la gestión de las reservas en el hotel Iberostar Ensenachos.

Este ha sido obtenido a partir de la generación automática de diagramas que incorpora el Microsoft SQL Server 2000.

En el anexo III, se puede apreciar el disparador (*trigger*) implementado para actualizar la tabla de clientes.

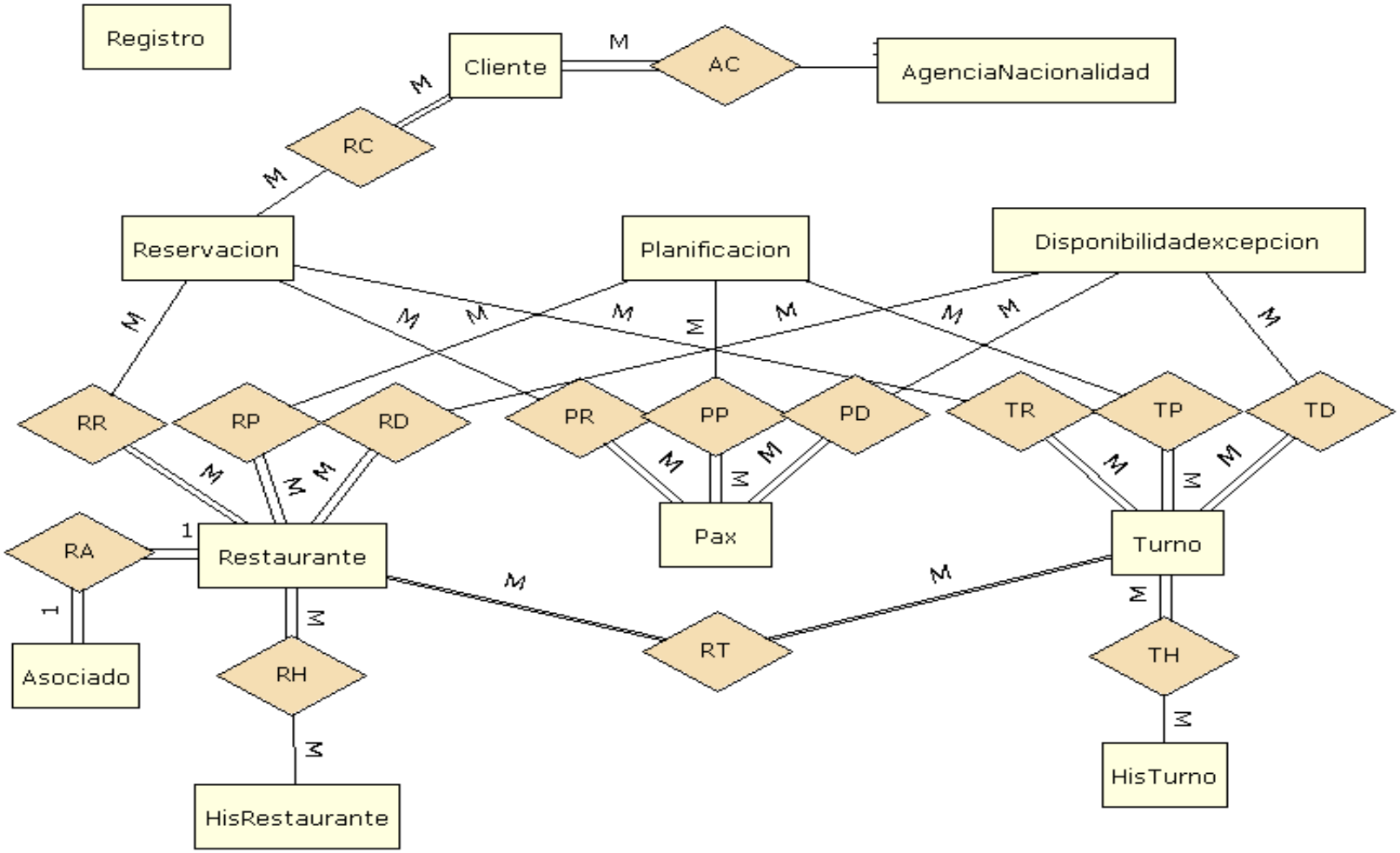


Figura 2.7. Diagrama entidad-relación de la base de datos.

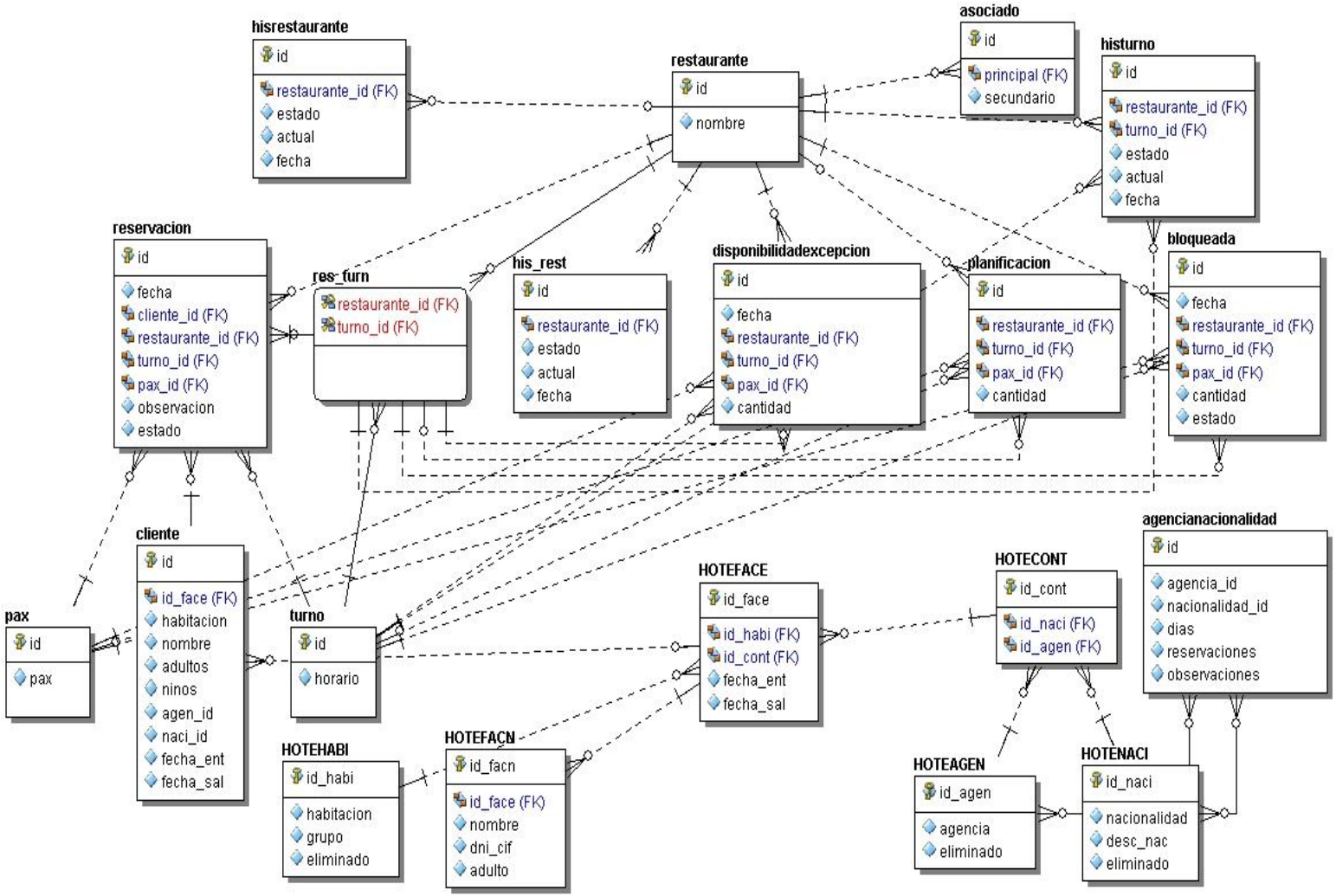


Figura 2.8. Modelo físico de la base de datos.

2.7 Modelo de Implementación

Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones, mostrando opciones de realización, incluyendo código fuente, binario y ejecutable. Los componentes representan todos los tipos de elementos software que entran en la fabricación de aplicaciones informáticas y pueden ser simples archivos, cabeceras, bibliotecas compartidas, módulos, ejecutables, o paquetes. En la figura 2.9 se puede apreciar una parte del diagrama de componentes para la gestión de reservaciones.

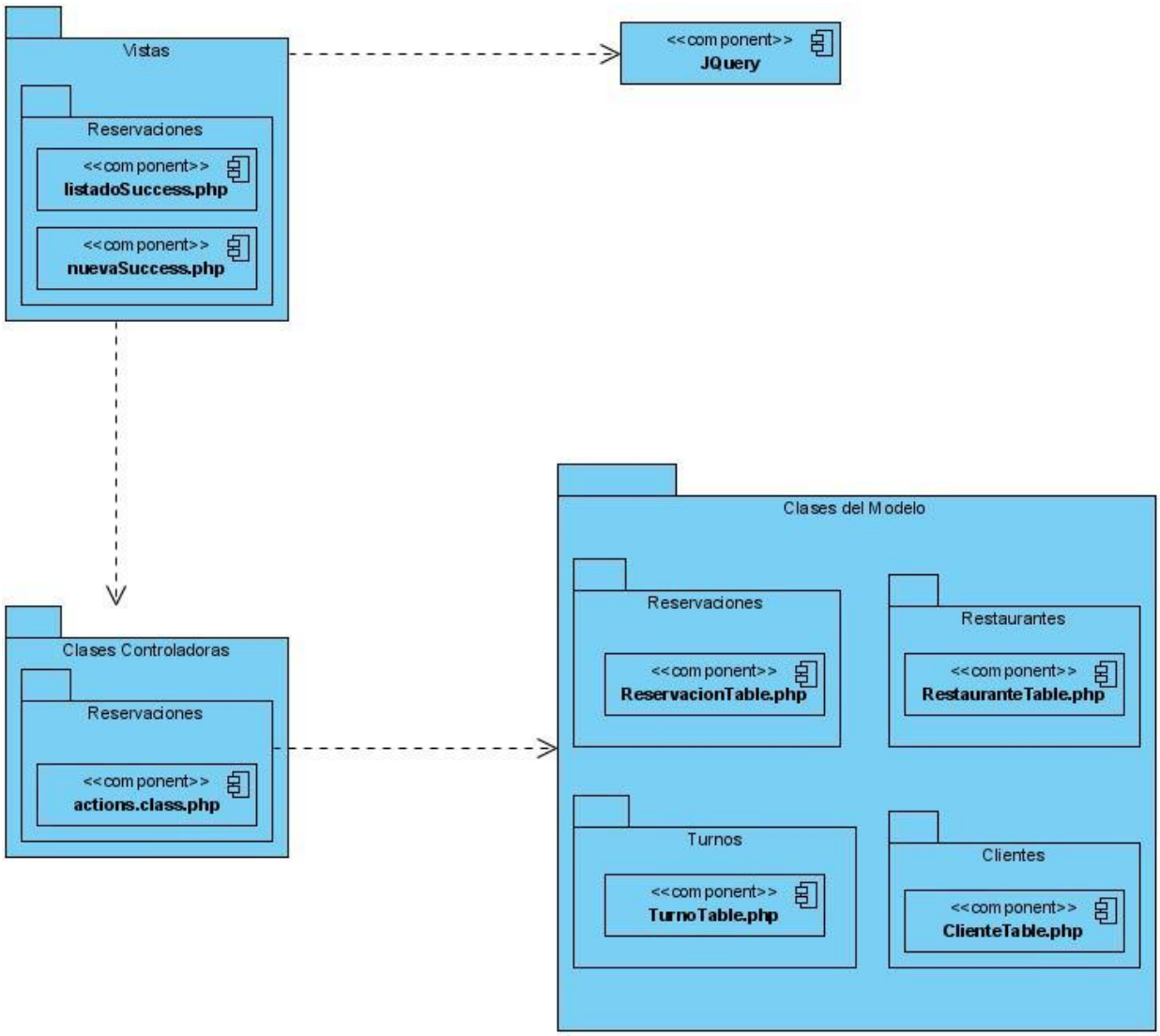


Figura 2.9. Diagrama de Componentes

Para un análisis en profundidad de los componentes ver anexo IV.

Al usarse como *framework* de desarrollo por sus múltiples características, se emplea al Artisteer para la generación de la plantilla con la que se trabaja. En la generación de consultas a la base de datos se emplea Doctrine. Varios ejemplos de métodos para la gestión de las reservaciones se pueden apreciar en el anexo V.

2.8 Diagrama de Despliegue

Los diagramas de despliegue muestran la disposición física de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. Los nodos representan un recurso computacional y se utilizan para modelar la topología del hardware sobre el que se ejecuta el sistema, en estos se ejecutan los componentes, que son los que participan en la ejecución del Sistema, y representan su despliegue físico. En la figura 2.10 se puede observar la distribución que se obtiene mediante el diagrama de despliegue.

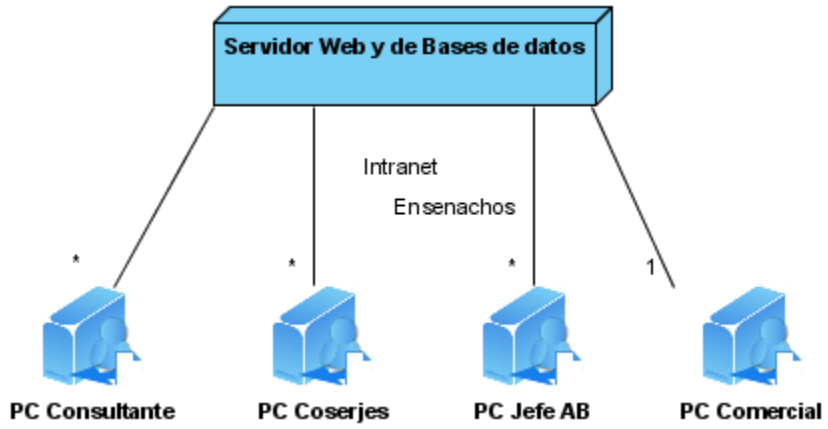


Figura 2.10. Diagrama de despliegue

2.9 Conclusiones parciales

En el presente capítulo se comienza haciendo un análisis del negocio para a partir de este comenzar a describir la propuesta de solución, desarrollarla, mostrar un listado de las funcionalidades que debe tener el Sistema que va a ser desarrollado para aplicación de las reservas en los restaurantes del hotel Iberostar Ensenachos, el cual debe cumplir con lo anteriormente dicho, y se modeló en términos de casos de uso del sistema describiendo más detalladamente algunos de los más importantes.

Luego de descrito el sistema se comienza a hacer su análisis y diseño, planteando la concepción general del diseño, obteniendo diagramas de clases web, mapa de navegación por el sitio, diagrama de despliegue y el diseño de la base de datos.

CAPÍTULOIII

“GUÍA DEL USUARIO”

CAPÍTULO III: *"Guía del usuario"*

En el presente capítulo se realiza un estudio de factibilidad, donde se determinan los requerimientos de software y hardware necesarios para el correcto funcionamiento de la aplicación para la reserva de restaurantes, además se realiza la estimación del esfuerzo y los costos basado en el método de puntos de casos de uso. Se realiza una guía para ayudar a que el usuario pueda comprender mejor el sistema y de esta forma realizar su trabajo más eficientemente.

3.1 Estudio de Factibilidad

La factibilidad se refiere a la disponibilidad de los recursos necesarios para llevar a cabo los objetivos o metas señalados y se apoya en aspectos básicos como la Factibilidad Técnica y la Factibilidad Económica. Estos dos aspectos se analizan a continuación para el caso del proyecto presentado en el Trabajo de Diploma.

3.1.1 Factibilidad Técnica

La Factibilidad Técnica consistió en realizar una evaluación de la tecnología existente en la organización y la posibilidad de hacer uso de la misma en el desarrollo del sistema propuesto. De acuerdo a lo anterior se evaluaron dos enfoques: Hardware y Software.

Hardware

En cuanto a hardware, específicamente en el servidor donde debe estar instalado el sistema propuesto, se deben cubrir los siguientes requerimientos mínimos:

- Procesador Pentium IV
- RAM 1024Mb

Por otra parte, las computadoras desde donde accederán al sistema los usuarios finales (conserjes, jefeab, comercial) deben satisfacer:

- Procesador Pentium III o superior.
- RAM 256 en adelante.
- Resolución del monitor 800 x 600 mínima (recomendado 1024 x 768 en adelante).

Evaluando el hardware existente y la configuración mínima necesaria, la organización no requiere realizar una inversión inicial para la adquisición de nuevos equipos, ni actualizar los existentes, ya que los mismos satisfacen los requerimientos establecidos para la instalación.

Software

La organización cuenta con todas las aplicaciones que se emplean para el desarrollo del sistema.

Los requerimientos de software del servidor son:

- Servidor Web

  Apache 2.x
- PHP

  Versión 5.2 o superior
- Servidor de Base de Datos

  Microsoft SQL Server 2000 (para lograr compatibilidad con el ZunPMSIberostar)
- Sistema Operativo

  Windows server 2003 Professional x86 (mayor compatibilidad con el software empleado en la manipulación de la Base de Datos)

Los requerimientos de software de los terminales son:

- Navegador Web

  Mozilla Firefox 9.01 (recomendado sobre cualquier sistema operativo)

Los usuarios pueden acceder al sistema desde cualquier computadora conectada en red con el navegador Mozilla Firefox utilizando la versión 9.01 o superior.

3.1.2 Factibilidad Económica

Aquí se determinaron los recursos necesarios para desarrollar, implantar y mantener en operación el sistema. Se utiliza para esto la estimación por Puntos de Casos de Uso y el análisis de Costo – Beneficio.

3.1.2.1Estimación basada en Casos de Uso

La estimación mediante el análisis de Puntos de Casos de Uso es un método propuesto originalmente por Gustav Karner de Objectory AB (Banerjee, 2001), y posteriormente refinado por muchos otros autores. Se trata de un método de estimación del tiempo de desarrollo de un

proyecto mediante la asignación de "pesos" a un cierto número de factores que lo afectan, para finalmente, contabilizar el tiempo total estimado para el proyecto a partir de esos factores.

Estimación del Esfuerzo Basada en Casos de Uso

Paso 1: Cálculo de Puntos de Casos de Uso sin Ajustar (UUCP)

Este valor se calcula a partir de la siguiente ecuación:

UUCP = UAW + UUCW donde,

UAW: Factor de Peso de los Actores sin ajustar

UUCW: Factor de Peso de los Casos de Uso sin ajustar

Determinación del factor de peso de los actores sin ajustar (UAW)

Este valor se calcula mediante un análisis de la cantidad de Actores presentes en el sistema y la complejidad de cada uno de ellos. La complejidad de los actores se establece, teniendo en cuenta en primer lugar, si se trata de una persona o de otro sistema, y en segundo lugar, la forma en que el actor interactúa con el sistema como aparece en la Tabla 3.1.

Tabla 3.1. Factores de peso de los actores

Tipo de actor	Descripción	Factor de peso	Número de actores	*Resultado*
Simple	Otro sistema que interactúa con el sistema a desarrollar mediante una interfaz de programación (API, *Aplication Programming Interface*)	1	0	*0*
Promedio	Otro sistema que interactúa con el sistema a desarrollar mediante un protocolo o una interfaz basada en texto.	2	1	*2*
Complejo	Una persona que interactúa con el sistema mediante una interfaz gráfica.	3	4	*12*
			Total	*14*

De manera que UAW = 14

Determinación del factor de peso en los casos de uso sin ajustar (UUCW)

Este valor se calcula mediante un análisis de la cantidad de Casos de Uso presentes en el sistema y la complejidad de cada uno de ellos. La complejidad de los casos de uso se establece teniendo en cuenta la cantidad de transacciones efectuadas en el mismo, donde una transacción se entiende como una secuencia de actividades atómicas, es decir, se efectúa la secuencia de actividades completa, o no se efectúa ninguna de las actividades de la secuencia. Ver tabla 3.2 Factores de peso de los casos de uso de la aplicación ARR 1.0.

Tabla 3.2. Factores de peso de los casos de uso.

Tipo de caso de uso	Descripción	Factor de peso	Número de Casos de Uso	*Resultado*
Simple	1-3 Transacciones	5	1	*5*
Promedio	4-7 Transacciones	10	2	*20*
Complejo	Mayor de 8 Transacciones.	15	5	*75*
			Total	*100*

UUCW = 100

Por tanto:

UUCP = UAW + UUCW

UUCP = 14 + 100

UUCP = 114

Paso 2: Cálculo de Puntos de Casos de Uso ajustados (UCP)

Seguidamente de calcular los Puntos de Casos de Uso sin ajustar, se debe ajustar este valor mediante la siguiente ecuación:

UCP = UUCP x TCF x EF donde,

UUCP: Puntos de Casos de Uso sin ajustar

TCF: Factor de complejidad técnica

EF: Factor de ambiente

Determinación del factor de complejidad técnica (TCF)

Este coeficiente se calcula mediante la cuantificación de un conjunto de factores que determinan la complejidad técnica del sistema. Cada uno de los factores se cuantifica con un valor de 0 a 5,

donde 0 significa un aporte irrelevante y 5 un aporte muy importante. Ver tabla 3.3 factores de complejidad técnica en la aplicación ARR 1.0.

Tabla 3.3. Factores de complejidad técnica.

Número de factor	Descripción	Peso	Valor	Factor	*Comentario*
T1	Sistema Distribuido	2	3	6	*El sistema es Web, por lo que posee cierto nivel de distribución*
T2	Tiempo de respuesta	1	2	2	*El tiempo de respuesta respalda los objetivos que se persiguen con el proyecto realizado, por lo que es el adecuado.*
T3	Eficiencia por el usuario	1	3	3	*Algunos roles necesitan estar relacionados con el sistema para su mejor funcionamiento.*
T4	Proceso interno complejo	1	3	3	*El sistema no posee cálculos complejos, aunque proporciona una serie de datos lógicos que necesitan un nivel medio de conocimiento para lograr su correcta comprensión.*
T5	Reusabilidad	1	3	3	*No es objetivo esencial hacer reusabilidad del código, a pesar de que este será orientado a objetos y podrá ser usado por sistemas similares.*
T6	Facilidad de instalación	0.5	4	2	*Por la naturaleza del sistema desarrollado no resulta compleja su instalación, ya*

					que el mismo es Web.
T7	Facilidad de uso	0.5	5	2.5	*El sistema debe ser fácil de usar, y lo más parecido al anterior.*
T8	Portabilidad	2	5	10	*El sistema se encuentra diseñado para que sea usado en situaciones similares en otros hoteles donde este el Zun instalado.*
T9	Facilidad de cambio	1	4	4	*El sistema se encuentra estructurado para que los cambios realizados afecten lo menos posible las funcionalidades del sistema.*
T10	Concurrencia	1	5	5	*La concurrencia es tratada con suma importancia.*
T11	Objetivos especiales de seguridad	1	5	5	*La seguridad del sistema es un tema bastante controlado, ya que el sistema sólo permite que un usuario realice las funcionalidades correspondientes a su rol dentro del sitio.*
T12	Acceso directo a terceras partes	1	2	2	*La aplicación es accesible a cualquier usuario.*
T13	Facilidades especiales de entrenamiento a usuarios finales	1	3	1	*No se hace necesario el entrenamiento de los usuarios finales, debido a la facilidad de uso que presenta el sistema, pero se debe incluir un manual*

					de usuario para garantizar la correcta usabilidad de dicho sistema.
			Total Factor	*48.5*	

El Factor de complejidad técnica se calcula mediante la siguiente ecuación:

$TCF = 0.6 + 0.01 * \sum(Peso_i x\ Valor\ asignado_i)$

$TCF = 0.6 + 0.01* 48.5$

$TCF = 1.085$

Determinación del factor ambiente (EF)

Las habilidades y el entrenamiento del grupo involucrado en el desarrollo tienen un gran impacto en las estimaciones de tiempo. Estos factores son los que se contemplan en el cálculo del Factor de ambiente.

Tabla 3.4. Factores de ambiente.

Número del factor	Descripción	Peso	Valor	Factor	*Comentario*
E1	Familiaridad con el modelo del proyecto usado.	1.5	3	4.5	*Se está familiarizado con el modelo del proyecto, pero la experiencia en el modelado es media.*
E2	Experiencia en la aplicación	0.5	4	2	*No es una aplicación que requiera de mucha experiencia, pero se necesita de un equipo capacitado y de conocimientos suficientes para garantizar su correcto funcionamiento.*
E3	Experiencia OO.	1	3	3	*Se considera cierto grado de experiencia en la*

					programación orientada a objetos (OO), debido a que esta es la que se ha impartido durante la carrera.
E4	Capacidad del analista líder.	0.5	2	1	*No existe analista líder.*
E5	Motivación.	1	5	5	*Exista gran motivación para el desarrollo del proyecto.*
E6	Estabilidad de los requerimientos.	2	5	10	*Aunque el sistema se encuentra sujeto a cambios, el mismo brinda las funcionalidades esenciales que dan cumplimiento a los objetivos que iniciaron su realización.*
E7	Personal media jornada.	-1	0	0	*Se trabajará a tiempo completo.*
E8	Dificultad en lenguaje de programación.	-1	3	-3	*Como el lenguaje empleado fue PHP y este ofrece grandes facilidades y ventajas, se considera una dificultad medio su empleo.*
			Total	*22.5*	

El factor de ambiente se calcula mediante la siguiente ecuación:

$EF = 1.4 – 0.03 * \sum (Peso_i x \text{ Valor asignado}_i)$

$EF = 1.4 – 0.03 * 22.5$

EF = 0.725

Cálculo de los Puntos de Casos de Uso Ajustados:

UCP = UUCP * TCF * EF

UCP = 114 * 1.09 * 0.73

UCP = 90.71

Paso 3: Cálculo del esfuerzo

El esfuerzo en horas-hombre viene dado por E = UCP * CF donde:

E: esfuerzo estimado en horas-hombre.

UCP: Puntos de casos de uso ajustados.

CF: Factor de conversión (20 horas-hombre por defecto).

E = 90.71 * 20

E = 1814.20

Sin embargo, para la obtención de una estimación más exacta de la duración del proyecto, se hace necesario agregar a la estimación del esfuerzo obtenida por los Puntos de Casos de Uso, las estimaciones de esfuerzo de las restantes actividades que se llevaron a cabo durante el desarrollo del software; así la distribución del esfuerzo entre dichas actividades está dada por la siguiente aproximación:

Tabla 3.5. Distribución genérica del esfuerzo.

Actividad	*Porcentaje*
Análisis	*10.00%*
Diseño	*20.00%*
Programación	*40.00%*
Pruebas	*15.00%*
Sobrecarga(otras actividades)	*15.00%*

Con este criterio y tomando como entrada la estimación de tiempo calculada a partir de los Puntos de Casos de Uso, se pueden calcular las demás estimaciones para obtener la duración total del proyecto.

Tabla 3.6. Distribución real del esfuerzo.

Actividad	*Porcentaje*
Análisis	*453.5*
Diseño	*907*
Programación	*1814.2*
Pruebas	*680.25*
Sobrecarga(otras actividades)	*680.25*
Total	*4535*

Paso 4: Cálculo del esfuerzo total

ETotal = 4535 horas /hombre

Paso 5: Cálculo del tiempo de desarrollo

Puede considerarse que el sistema va a ser desarrollado por un equipo mínimo de 3 personas.

TDesarrollo = ETotal/CHTotal: Cantidad de hombres =3

TDesarrollo =4535 horas

Considerando que se trabajan 8 horas diarias:

TDesarrollo = TDesarrollo/8 horas

TDesarrollo= 1511.6horas/8 horas

TDesarrollo= 189 días aproximadamente, el equivalente a 6.3 meses

Paso 6: Cálculo del costo

CostoTotal = ETotal * TH TH: Tarifa horaria =1.031

CostoTotal =4535.4 * 1.031

CostoTotal = 4675.6 pesos

3.1.3 Análisis de Costo – Beneficio

El desarrollo de esta aplicación representa para la organización una serie de beneficios, tanto tangibles como intangibles, los cuales se enuncian a continuación:

Beneficios tangibles:

- Con la aplicación anterior se lograba un ahorro de materiales de oficina como hojas, presillas, tóner de impresora, entre otros. Sus limitaciones en especial la seguridad, el tiempo para acceder a los datos, y el lento proceso de mantenimiento con que cuenta representa una barrera para el desarrollo de las tareas requeridas. ARR garantiza el buen funcionamiento de todas estas actividades, no sólo agilizaría el trabajo a realizar, sino que le ahorraría al hotel y de paso al país grandes sumas de dinero.

Beneficios intangibles:

- Los usuarios emplearán con mayor frecuencia la navegación Web para tratar las gestiones de las reservas en los restaurantes, manteniéndose así actualizados con las nuevas tecnologías.
- Facilitará el trabajo tanto de los usuarios, a la hora de realizar búsquedas y reportes de reservaciones o cancelaciones, como de los directivos del hotel, mostrándose una serie de estadísticas que le serán de gran utilidad para llevar las reservaciones en los diferentes restaurantes y horarios.
- Aumentará la confiabilidad y seguridad de la información, ya que esta se mantendrá almacenada en una base de datos correctamente diseñada.
- Mejorará la calidad y rapidez de los procesos en cuanto a tiempo, procesamiento y organización de la información.

Considerando los resultados arrojados respecto a la factibilidad del software después del estudio realizado en este epígrafe, se concluye que el sistema a desarrollar es factible considerando que brinda suficientes beneficios y un impacto económico equivalente a un ahorro de 4675.6 pesos ya que el mismo es desarrollado como resultado de un trabajo de investigación que no incide sobre los costos del hotel.

3.2 Guía para el Usuario

La aplicación para las reservas en los restaurantes del hotel Iberostar Ensenachos cuenta con interfaces sencillas y amigables que permiten a los usuarios agilizar y perfeccionar su trabajo.

3.2.1 Acceso al sistema

Para acceder al sistema se debe utilizar el navegador Mozilla Firefox para garantizar un mejor funcionamiento de la aplicación. Cada cuenta tiene sus privilegios que le permiten navegar por el sitio con diferentes permisos y restricciones.

Existen 4 tipos de usuarios según sus permisos, estos son: Jefe del Departamento de Alimentos y Bebidas, Conserjes, Comercial y Consultante.

El Jefe del Departamento de Alimentos y Bebidas es el encargado de gestionar toda la configuración de los diferentes restaurantes y sus turnos, es el que manipula la información sobre los usuarios del sistema. No puede reservar o cancelar.

El Conserje, es el encargado de crear y cancelar las reservaciones, puede realizar búsquedas sobre estas.

El Comercial, es el encargado de gestionar la información sobre la cantidad de reservaciones de los huéspedes según su contrato, en dependencia de su agencia de viaje y la nacionalidad de los mismos.

El consultante es aquel que solo puede realizar búsquedas, por ejemplo: el jefe de cocina, para conocer la cantidad de comensales por restaurante, turnos y quiénes son los huéspedes. En la Figura 3.1 se muestra la página para autenticarse y entrar al sitio.

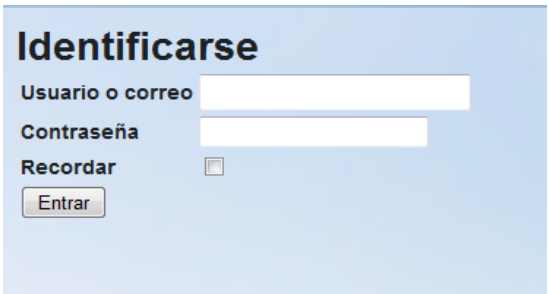


Figura 3.1. Ventana de Autenticación.

3.2.2 Composición de la aplicación

La página principal de la aplicación está compuesta por un encabezado de título y un menú vertical, como se muestra en la siguiente imagen.

Figura 3.1. Página Principal

El menú vertical, ofrece una lista de opciones que permiten gestionar la información a la que hacen referencia. Este cambia en dependencia del tipo de usuario que se encuentre registrado.

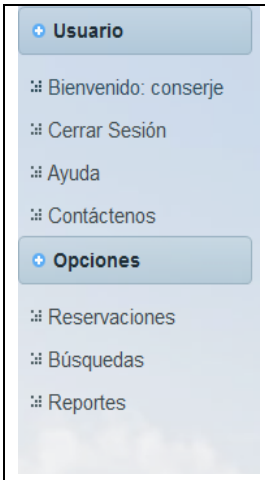
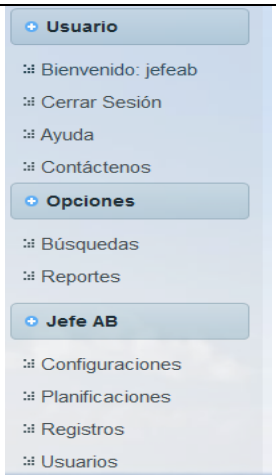
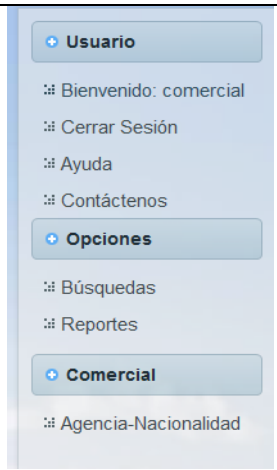
		

Figura 3.2 Menú correspondiente al conserje	Figura 3.3 Menú correspondiente al jefe AB	Figura 3.4 Menú correspondiente al comercial

Usuario:

- Bienvenido: Permite al usuario ir a la página principal. Brinda información sobre el usuario registrado.

- Cerrar sesión: Permite al usuario salir de su cuenta. Es importante señalar que el usuario debe cerrar sesión antes de salir del sistema.
- Ayuda: Brinda al usuario la ayuda para un mejor entendimiento y desempeño durante su trabajo con la aplicación.
- Contactos: Es la forma de conocer las opiniones y sugerencias de los usuarios finales sobre el producto desarrollado.

Opciones:

- Reservaciones: Se refiere al listado de las reservaciones, desde aquí es donde se pueden crear o cancelar las reservaciones.
- Búsquedas: Se parte para elegir las diferentes formas de combinar las búsquedas.
- Reportes: Brinda varios reportes gráficos, que son de mucha ayuda para la toma de decisiones.

Jefe AB:

- Configuraciones: Es donde se configuran los parámetros de los restaurantes, turnos, las mesas, así como su funcionamiento.
- Planificaciones: Se definen las capacidades de los restaurantes, y sus excepciones para días o períodos específicos.
- Registros: Mantiene un registro sobre las operaciones realizadas por todos los usuarios del sistema.
- Usuarios: Se especifican los usuarios del sistema, así como sus permisos.

Comercial:

- Agencia-Nacionalidad: Es para establecer la cantidad de reservaciones a las que tiene derecho el huésped según el contrato por el que viajó.

3.2.3 Navegación del usuario conserje

A este usuario solo se le permite realizar búsquedas y gestionar las reservaciones de los huéspedes. La figura 3.5 que se muestra a continuación es la referente al listado de las reservaciones. Es importante destacar que por defecto este listado siempre sale para el día en curso.

Figura 3.5 Listado de Reservaciones

Se brinda la posibilidad de escoger una fecha diferente pulsando sobre el botón Buscar por fecha. Se puede buscar directamente mediante el botón Buscar por habitación las reservaciones correspondientes al huésped hospedado en la habitación correspondiente.

Para crear una nueva reservación se da clic sobre el botón Nueva Reservación, de aquí se pasa a una nueva ventana, (Ver figura 3.6), donde se completa el formulario con el cliente, la fecha, el restaurante, el turno, el pax de la mesa y las observaciones (esta última no es obligatoria).

Figura 3.6 Nueva reservación

Para cancelar alguna reservación se da clic sobre el botón el cual se encuentra en todas las a la derecha de la fila con la reservación que se desea eliminar. El sistema pide confirmación antes de proceder.

Para adjuntar una nueva reservación a otra ya existente (siempre que la capacidad de la mesa sea mayor que los huéspedes que reservaron), aparece el botón , pulsando encima del mismo, surge una nueva ventana (Figura 3.7) donde se selecciona la habitación. A continuación se ofrece el nombre del cliente y la cantidad de personas en esa habitación.

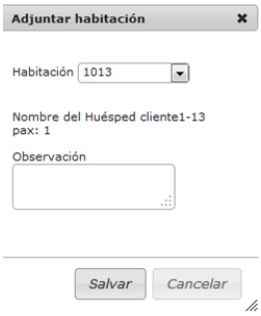


Figura 3.7 Ventana para adjuntar habitación a una reservación existente

3.2.4 Navegación para el usuario jefe AB

Este usuario es la persona que se encarga directamente de la configuración y planificación de los restaurantes y turnos. Es el que posee acceso sobre los registros y las operaciones realizadas en la aplicación. También es el encargado de gestionar a los usuarios, otorgarles permisos, crearlos, modificarlos y eliminarlos. Los permisos necesarios ya están creados, los cuales están en correspondencia con los 4 tipos de usuarios analizados anteriormente y abarcan todas las necesidades de seguridad del sistema. Vale destacar que estos permisos son excluyentes. Cuando se crea un nuevo restaurante aparece un formulario para ingresar el nombre del mismo así como una lista de todos los horarios existentes (ver figura 3.8). Se puede escoger desde este momento los turnos con los cuales trabajará este restaurante. Por defecto este aparece abierto, por lo que solo faltaría llenar la planificación de las cantidades de mesas por cada turno para que fuera posible reservar en él. Los turnos y los pax de las mesas se crean de forma similar.

Figura 3.8 Interfaz para la creación de un nuevo restaurante

Una vez se ha accedido al listado de las planificaciones esta es la ventana correspondiente (ver figura 3.9). Desde aquí mediante el botón Nueva PLanificación se accede al siguiente formulario con lo que se puede crear nuevas planificaciones (Ver figura 3.10). Con el botón Listado de Excepciones se visualizan las excepciones

Figura 3.9 Interfaz para la planificación

Figura 3.10 Interfaz para nueva planificación

De forma similar se visualiza y crean las excepciones.

También el usuario jefe AB es el encargado de revisar los registros que genera automáticamente la aplicación Web. Mediante la siguiente interfaz se puede refinar la búsqueda y obtener como resultado final un documento PDF. Un ejemplo de las búsquedas que se pueden obtener a partir de los registros del sistema se puede apreciar en la figura 3.11.

Figura 3.11 Interfaz para la búsqueda de los registros del sistema

3.2.5 Navegación para el usuario comercial

Este usuario es la persona que se encarga de otorgar las cantidades de reservaciones según los contratos por los que arriban los huéspedes del hotel. Para ello cuenta con un listado donde se muestra la agencia, nacionalidad, los días, las reservaciones y alguna observación que se desee realizar. Cuenta con varios botones entre ellos el de editar y el de cancelar y el de ingresar una nueva fila Nueva Agencia-Nacionalidad.

Para crear una nueva fila se procede a completar este formulario:

3.3 Conclusiones parciales

Con la culminación de este capítulo se realizó un estudio de factibilidad, donde quedó evidenciado el impacto económico que generará la aplicación ARR 1.0. Se elaboró una guía para el usuario, donde se explica cómo acceder al sitio web para la gestión de las reservaciones en los restaurantes del hotel Iberostar Ensenachos y la manera en que los diferentes usuarios pueden interactuar con el software según sus permisos, facilitando la navegación, entrada de la información y un mejor entendimiento del sitio, permitiendo conocer las secciones que tiene y las acciones que permite realizar.

CONCLUSIONES

CONCLUSIONES

Para mejorar los niveles de calidad de atención al cliente e invertir en iniciativas innovadoras, los hoteles necesitan impulsar la productividad en todas sus áreas, una de ellas, la gestión en Restaurantes. En este trabajo se tienen las conclusiones siguientes:

- El estudio del software para la gestión de restaurantes SRR 11.1 y la comunicación con los participantes de este proyecto posibilitó la determinación de las limitaciones existentes, apreciándose que los mayores problemas se encuentran en la seguridad de la aplicación, la demora al acceder a los datos de los huéspedes y la inconsistencia de la base de datos.
- A partir de los requisitos funcionales y no funcionales se implementó una aplicación Web utilizando como marco de trabajo Symfony, donde se diseñó una nueva base de datos para la aplicación, que elimina la inconsistencia de la anterior y reduce el tiempo de espera para realizar las reservaciones. Esta base de datos mantiene actualizada en todo momento la información de los huéspedes del hotel, de manera que no ocurran reservaciones dobles o errores.
- El análisis de factibilidad realizado permite concluir que es posible realizar la instalación de la aplicación con los recursos (hardware y software) disponibles en la organización hotelera. Adicionalmente se determina que la aplicación tiene un impacto económico de, aproximadamente, 4675.6 pesos, que significan un ahorro para esa organización a partir de la realización de este trabajo de investigación.

RECOMENDACIONES

RECOMENDACIONES

La aplicación ARR 1.0 actualmente se encuentra en la fase de prueba en el hotel Iberostar Ensenachos, donde los usuarios finales explotan sus funcionalidades con vistas a la posterior liberación del producto. Luego de haber vencido este período se recomienda extender el software desarrollado, a todos aquellos hoteles de la cadena Iberostar donde esté instalado el Zun FO.

BIBLIOGRAFÍA

BIBLIOGRAFÍA

ALVAREZ, M. A. & NEBOT, E. 2008. Taller de jQuery.

ANDERSON, J. & MASTERS, L. E. 2006. CakePHP: A rapid development Framework. CakePHP Programmer's Reference Guide. Cake Software Foundation Inc.

APACHE, F. 2012. Apache software foundation.

ASLESON, R. & SCHUTTA, N. T. 2006. Foundations of Ajax. *In:* BUCKINGHAM, E. (ed.).

BANERJEE, G. 2001. Use Case Points an estimation Approach. [Accedido 05-04-2012].

CEA, O. 2010. *Programa hotel 3000* [En línea]. Disponible: http://www.ceaordenadores.com/hoteles/software-hotel-3000].

CHAFFER, J. S., K. 2007. Learning jQuery. *In:* CHITTAR, D. (ed.). Birmingham: Packt Publishing Ltd.

CHASE, N. 2006. Understanding the Zend Framework.

DELFOS, C. E. 2004. Perfil de la Compañía Informarca Hotel Systems. [Accedido 04-09-2012].

ECHEVERRÍA, A. M. 1995. Manual Práctico de Html. Madrid, España.

EGUÍLUZ PÉREZ, J. 2008. Introducción a CSS. *In:* WWW.LIBROSWEB.ES (ed.).

EGUÍLUZ PÉREZ, J. 2011. *Symfony español* [En línea]. Disponible: www.symfony.es [Accedido 10-11 2011].

ELLISLAB, I. 2012. *Codeigniter Framework PHP* [En línea]. Disponible: http://www.codeigniter.com/].

EXTENDSOFT, I. 2012. *Artisteer Software* [En línea]. Disponible: http://www.artisteer.com/].

JACOBSON, I., BOOCH, G. & RUMBAUG, J. 2004. *El proceso unificado de desarrollo de software*, La Habana, Editorial Félix Varela.

KUMBIA. 2012. *Kumbia PHP* [En línea]. Disponible: http://www.kumbiaphp.com/blog/].

MICROSOFT. 2011. *Microsoft SQL Server* [En línea]. Disponible: www.microsoft.com/spain/sql/default.mspx [Accedido 12-10 2011].

OMT 2011. UNWTO Tourism Highlights. *In:* TURISMO, O. M. D. (ed.) 2011 ed.

ONEI. 2012. *Oficina Nacional de Estadísticas e Información de la República de Cuba* [En línea]. La Habana. Disponible: http://www.onei.cu [Accedido 05-04 2012].

PARSEC. 2005. *ADAPTO Gestión de Hotel* [En línea]. Cantabria. Disponible: www.adapto.net [Accedido 04-02 2012].

PHP. 2007. *PHP: Hypertext Preprocessor* [En línea]. Disponible: http://www.php.net/ [Accedido 10/12/2011 2011].

POTENCIER, F. & ZANINOTTO, F. 2008. Symfony la guía definitiva.

RIVAS, L. A., PÉREZ, M. & MENDOZA, L. E. 2006. Herramientas de Desarrollo de Software.

RUIZ, M. H. 2007. Una introducción a APACHE.

SÆTHER BAKKEN, S., AULBACH, A., SCHMID, E., WINSTEAD, J., TORBEN WILSON, L., LERDORF, R., ZMIEVSKI, A. & AHTO, J. 2001. Manual de PHP. *In:* MARTÍNEZ, R. (ed.).

SALGADO FEBLES, J. E. 2007. El negocio electrónico en el sector del Turismo. *In:* TURÍSTICOS, C. D. E. (ed.). La habana: Universidad de la Habana.

SENSIOLABS 2009. Doctrine ORM for PHP.

SPARKS, G. 2006. Introducción al modelado de sistemas de software usando el Lenguaje Unificado de Modelado (UML) Australia.

UNESCO. 2007. *Biosphere Reserve Information: Buenavista* [En línea]. Disponible: http://www.unesco.org/mabdb/br/brdir/directory/biores.asp?mode=all&code=CUB+06 [Accedido 04-05 2012].

YUEN, P. K. & LAU, V. 2003. Practical Web Technologies. . *In:* EDUCATION., P. (ed.).

ANEXOS

Anexo I: Glosario de Términos

Back Office o BO: Parte trasera u oculta de la actividad, generalmente se refiere a la parte económico-contable que no enfrenta directamente al usuario o cliente.

Bungalow: edificio donde se hospedan los clientes del hotel.

Conserje: Persona encargada entre otras funciones de atender al huésped y guiarlo en el proceso de las reservaciones en el restaurante.

Comercial: Es el encargado de gestionar la cantidad de reservaciones correspondientes a los huéspedes en dependencias de su contrato.

Front Office o FO: Parte de la actividad, que enfrenta directamente al usuario o cliente en un sistema, por ejemplo de gestión hotelera.

Hotel: es un edificio planificado y acondicionado para otorgar servicio de alojamiento a las personas temporalmente y que permite a los visitantes sus desplazamientos. Los hoteles proveen a los huéspedes de servicios adicionales como restaurantes, piscinas y guarderías. Algunos hoteles tienen servicios de conferencias y animan a grupos a organizar convenciones y reuniones en su establecimiento.

Hoteles: establecimientos que, ofreciendo alojamiento con o sin comedor y otros servicios complementarios, ocupan la totalidad de uno o varios edificios (o una parte independizada de ellos) constituyendo sus dependencias todo un homogéneo con accesos, escaleras y ascensores de uso exclusivo y que reúna los requisitos técnicos mínimos para cada categoría. Se clasifican en 1 estrella o categoría básica, 2, 3, 4, 4 superior, 5 estrellas y Gran lujo.

Huésped: Persona que utiliza los servicios de alojamiento en un establecimiento.

Iberostar: es un conjunto de empresas dedicadas al sector turístico con sede en Palma de Mallorca (España). Actualmente, este grupo está formado por, entre otras, Iberostar Hotels & Resorts (división hotelera), cuenta con más de 100 hoteles en 15 países diferentes.

Jardines del Rey: es el nombre de un conjunto de cayos que conforman el archipiélago Sabana-Camagüey, el más extenso y numeroso de los cuatro que rodean la isla de Cuba.

Jefe AB: Es el jefe del Departamento de Bebidas y Alimentos, encargado de la planificación de los restaurantes y su correcto funcionamiento.

Sistema de Gestión Hotelera: es una solución informática con el objetivo de automatizar las principales áreas de una instalación hotelera y vinculando e integrando en un solo sistema las distintas áreas del mismo (carpeta, ama de llaves y mantenimiento, reservas y comercial, relaciones públicas, economía, compras y almacén).

Todo Incluido: modalidad en el turismo que permite ventajas para los ofertantes y para los clientes que reciben este servicio. Los hoteleros logran abaratar los costos y llegar a tener una ocupación respetable, esta modalidad incita a la masificación del turismo; por otra parte los clientes tienen ventajas en cuanto a precios aceptables, seguros de viaje, anti-accidentes, servicios médicos, hospedaje, alimentación, y la recreación como valor añadido, todo esto, dado por la organización de los turoperadores, que facilitan el proceso.

ZunPMSIberostar: Nombre otorgado a la Base de Datos de la suite Zun para la gestión hotelera.

ZunFront Office: Suite ofimática para el trabajo en la gestión hotelera. Es la que actualmente se encuentra en uso el hotel Iberostar Ensenachos en su versión 1.8. Fue desarrollado por el Grupo de Electrónica del Turismo (GET) perteneciente al Ministerio del Turismo.

Anexo II: Comparación entre marcos de trabajos de PHP.

Características	CakePHP	Symfony	Zend FW
Arquitectura de aplicaciones			
Incorporación del patrón Modelo Vista Controlador Orientado a Objetos.	X	X	X
Operaciones CRUD (*Create, Retrieve, Update y Delete*) asociadas a patrón Active Record.	X	X	
Mapeado de objetos a bases de datos relacionales (ORM).	X	X	
Independiente del manejador de base de datos.	X	X	X
Estructura por defecto para aplicaciones (*scaffolding*).	X	X	
Archivos de configuración de la aplicación.	.PHP	.YML	.PHP
Acceso vía web			
Despachador de peticiones HTTP.	X	X	X
Generación de URLs amigables.	X	X	X
Implementación de código HTML			
Uso de plantillas en PHP.	X	X	X
Posibilidad de uso de plantillas en *Smarty*.	X	X	
Implementación de ayudantes de plantillas (*HELPERs*).	X	X	
Seguridad			
Manejo propio de sesiones por usuarios.	X	X	X
Manejo de privilegios de acceso a secciones de la aplicación (*Access Control List*).	X	X	X
Verificación de la salida generada en HTML por procesamiento de peticiones (*Data Sanitization*).	X	X	

Validación de Formularios	X	X	X
Prevención XSS	X	X	X
Prevención *SQL Injection*		X	
Usabilidad y acceso rápido			
Almacenamiento en caché de las vistas.	X	X	
Almacenamiento en caché de configuración de las aplicaciones.		X	
Soporte implícito para AJAX	X	X	
Documentación para su uso			
Manual de referencia.	X	X	X
Documentación de la Interfaz de Programación de Aplicaciones (API).	X	X	X
Herramientas de programación			
Generación de código PHP.	X	X	
Herramientas de prueba y depuración.		X	X
Interfaz de línea de comandos para la creación y mantenimiento de aplicaciones.	X	X	
Almacenamiento de registros de funcionamiento del marco de trabajo.		X	
Extensibilidad y opciones adicionales			
Integración con otras herramientas a través de plugins.	X	X	
Implementación propia de llamadas (AJAX)	X	X	
Soporte para Web Services.		X[2]	X
Soporte para envío de correo electrónico.		X	X
Generación de contenido sindicalizado (RSS).		X	X
Generación de archivos PDF.	X	X	X
Soporte para internacionalización y localización de contenidos.		X	X

[2]Mediante la utilización del pluginsckWebServicesPlugins.

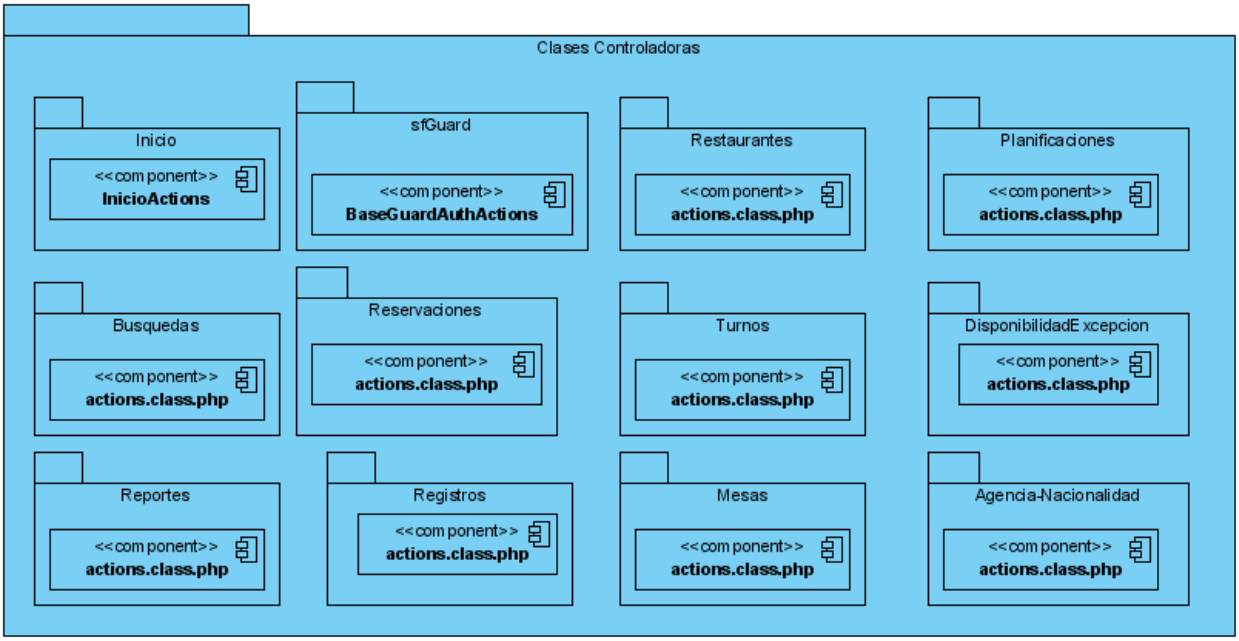
Soporte PHP			
Soporte para PHP4.	X		
Soporte para PHP5.	X	X	X
Características adicionales			
Licencias libres.	MTI	MTI	BSD
Comunidad activa de usuarios.	X	X	X

Anexo III: Trigger para actualizar la tabla de clientes

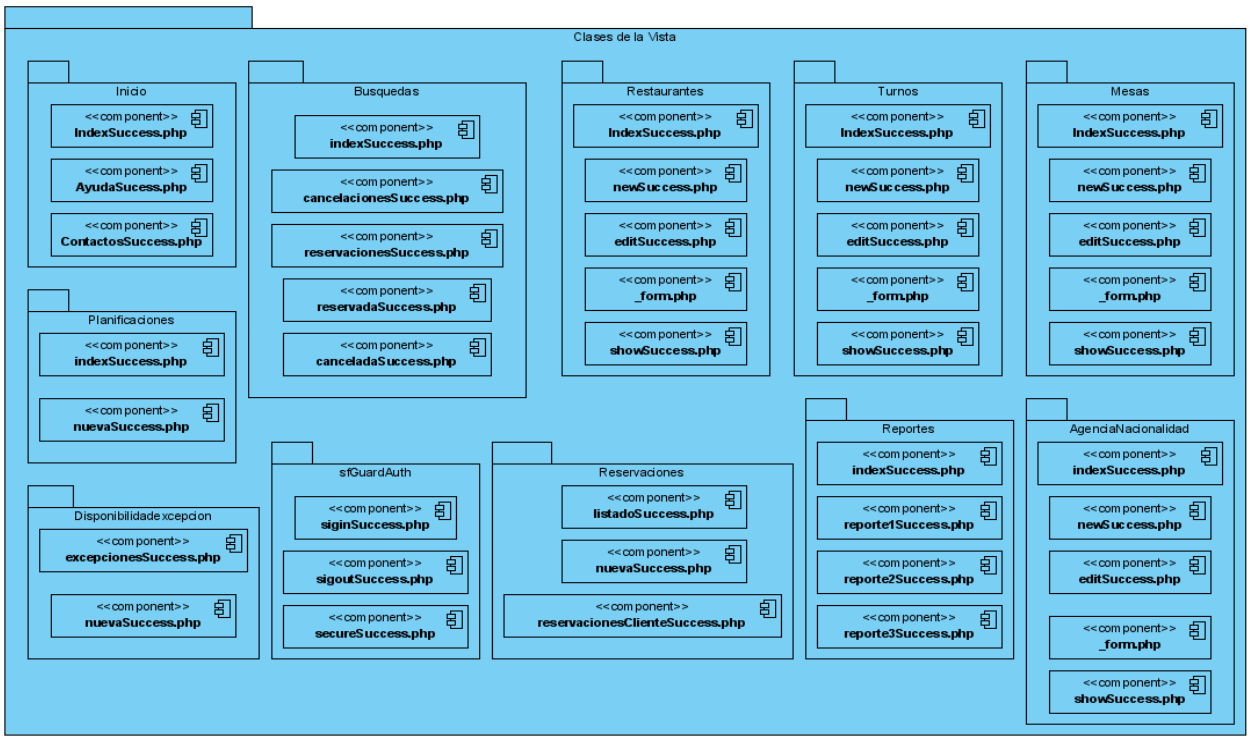
```
CREATE  triggercreacliente ON HOTEFACN
INSTEAD OF INSERT
AS
BEGIN
declare @idfaceint
declare @habitacionint
declare @nombrevarchar(100)
declare @adultoint
declare @ninosint
declare @agenint
declare @naciint
declare @fechaentdatetime
declare @fechasaldatetime
set @idface=(Select i.id_face FROM inserted i)
set @adulto=(Select i.adulto FROM inserted i)
set @habitacion=(SELECT habitacion FROM hoteface f inner join hotehabi h on
(f.id_habi=h.id_habi) WHERE @idface=f.id_face)
set @nombre=(Select i.nombre FROM inserted i)
set @ninos=(1-@adulto)
set @agen=(SELECT id_agen FROM hotecont h inner join hoteface f on f.id_cont=h.id_cont
WHERE f.id_face=@idface)
set @naci=(SELECT id_naci FROM hotecont h inner join hoteface f on f.id_cont=h.id_cont
WHERE f.id_face=@idface)
set @fechaent=(Select i.fecha_ent FROM inserted i)
set @fechasal=(Select i.fecha_sal FROM inserted i)

        if (EXISTS(select * from hotefacn f where @idface=f.id_face))
```

```
        updateensenachos.dbo.cliente set adultos=adultos+@adulto, ninos=ninos+(1-@adulto)
where @idface= ensenachos.dbo.cliente.id_face
        else
        insert into ensenachos.dbo.cliente
        (id_face,habitacion, nombre, adultos, ninos, agen_id, naci_id, fecha_ent,fecha_sal)
        values
        (@idface, @habitacion, @nombre, @adulto, @ninos, @agen, @naci, @fechaent,
@fechasal)
END
```

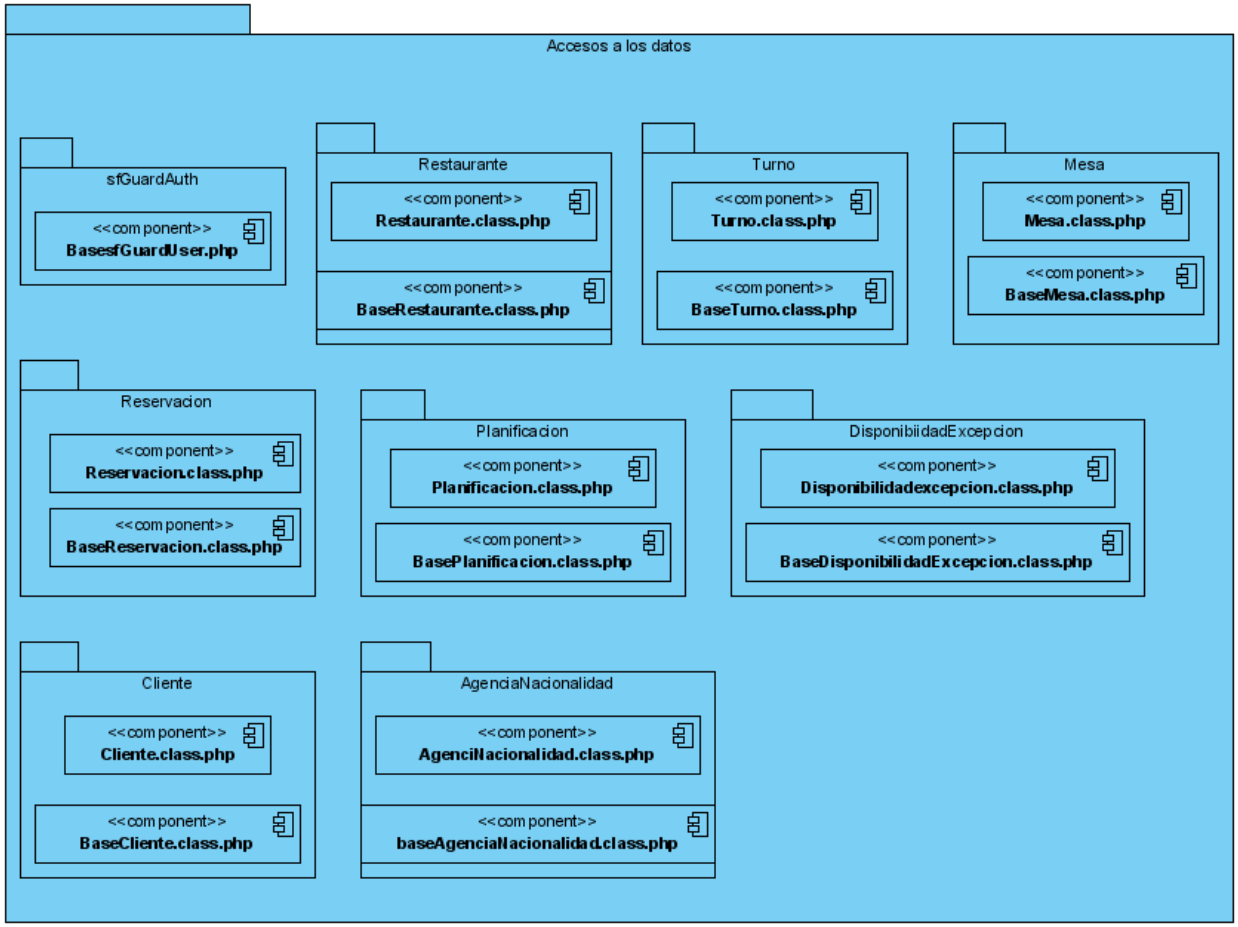
Anexo IV: Representación de la aplicación siguiendo el patrón de arquitectura MVC



Clases controladoras



Clases de la vista



Clases para el acceso a datos

Anexo V: Métodos desarrollados usando Doctrine para gestionar las reservaciones.

```
public function getReservaciones($fecha, $idrestaurante, $tidurno)
{
      $q = Doctrine_Core::getTable('Reservacion')
           ->createQuery('r')
         //  ->select('d.id')
           ->where('r.estado=?',1)
           ->andWhere('r.restaurante_id = ?', $idrestaurante)
           ->andWhere('r.turno_id = ?', $idturno)
           ->andwhere('r.fecha= ?', $fecha)
           ->andWhere('r.id NOT IN (Select secundario from asociado)')  ;
       return $q->fetchArray();
}
```

Este método retorna todas las reservaciones que coinciden con la fecha, el id de restaurante y el id del turno que se obtienen como parámetros del mismo.

```
public static function knowReservacionesClienteFecha($fecha, $idcliente)
   {
        $q = Doctrine_Core::getTable('Reservacion')
           ->createQuery('r')
           ->select('r.id')
           ->where('r.estado = ?', 1)
          ->andWhere('r.fecha = ?', $fecha )
          ->andWhere('r.cliente_id = ?', $idcliente )   ;
      $result= $q->fetchOne();

          if($result['id']==null)
          return false;
```

```
        else
        return true;
}
```

A este método se le pasan dos parámetros una fecha y el id del cliente. Devuelve un valor booleano (*true* o *false*) en dependencia de si el cliente que se posee alguna reservación para la fecha. No se tienen en cuenta las cancelaciones.

```
public function ReservacionesdelCliente($idc)
  {
    $q = Doctrine_Query::create()

        ->select('*')
        ->from('Reservacion')
        ->where('estado = ?', 1)
        ->andWhere('cliente_id = ?',$idc);

        return $q->fetchArray();
}
```

Con este método se obtienen todas las reservaciones que posee un cliente determinado por su id el cual se obtiene como parámetro.