



UNIVERSIDAD AMERICANA DE ACAPULCO
"EXCELENCIA PARA EL DESARROLLO"

FACULTAD DE INGENIERÍA EN COMPUTACIÓN

INCORPORADA A LA UNIVERSIDAD NACIONAL
AUTÓNOMA DE MÉXICO
CLAVE DE INCORPORACIÓN 8852-16

**"SISTEMA DE ADMINISTRACIÓN WEB DE LA CASA DE HUÉSPEDES
MATTI"**

T E S I S

PARA OBTENER EL TÍTULO DE
INGENIERA EN COMPUTACIÓN

PRESENTA
**NORMA OLEA TECANHUEY
SELENE BENITEZ MARTINEZ**

DIRECTOR DE TESIS
M.C. JOSE MARIO MARTINEZ CASTRO



ACAPULCO, GUERRERO, MAYO DEL 2013.



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

AGRADECIMIENTOS

A LA UNIVERSIDAD AMERICANA DE ACAPULCO, POR SER UN PILAR EN NUESTRA EDUCACIÓN PROFESIONAL DE EXCELENCIA, NUESTRA ALMA MATER, QUE ESTARÁ SIEMPRE PRESENTE EN TODOS NUESTROS LOGROS.

A LA FACULTAD DE INGENIERÍA EN COMPUTACIÓN, POR ESTAR SIEMPRE DEDICADA A FORMARNOS PROFESIONALMENTE PARA PODER ENFRENTARNOS A NUESTRO ENTORNO LABORAL CON ÉXITO.

A TODOS NUESTROS QUERIDOS PROFESORES, SIEMPRE AGRADECIDAS CON ELLOS POR SER UNA PARTE FUNDAMENTAL EN NUESTRA EDUCACIÓN HASTA LLEGAR A LA CULMINACIÓN DE ESTE DOCUMENTO Y EN TODO LO SIGUIENTE QUE LOGREMOS.

A TODOS ELLOS GRACIAS.

NORMA OLEA TECANHUEY

SELENE BENITEZ MARTINEZ

DEDICATORIAS

A DIOS POR PERMITIRME REALIZAR UN PEQUEÑO ÉXITO MÁS EN MI VIDA.

A MI PAPÁ POR TANTO AMOR, POR NO CLAUDICAR Y SIEMPRE DARME SU VOTO DE CONFIANZA EN MIS DECISIONES, PERO SOBRE TODO POR SER SIEMPRE UN EJEMPLO Y UN ALICIENTE EN MI VIDA. A MI MADRE POR EL CORTO TIEMPO QUE ESTUVO CONMIGO PERO EN EL CUAL EL AMOR SIEMPRE ESTUVO DEMOSTRADO. A MI TÍA GRIS POR EL EJEMPLO DE FORTALEZA QUE SIEMPRE ME HA DADO, POR ENSEÑARME A DEFENDERME Y A LUCHAR POR LO QUE QUIERO.

A MI ESPOSO, POR SIEMPRE CREER EN MÍ, POR TODO EL AMOR INCONDICIONAL QUE ME HA DADO (QUE SERÁS ETERNAMENTE CORRESPONDIDO), POR ESTAR CONMIGO EN LAS MALAS Y LAS PEORES, GRACIAS POR LLEGAR A MI VIDA ¡¡TE AMO!!.

A MIS SUEGROS POR QUERERME Y HABERME ABIERTO LAS PUERTAS DE SU CORAZÓN, POR SIEMPRE ESTAR A MI LADO EN LAS BUENAS Y EN LAS MALAS, SOBRE TODO POR EL APOYO Y LA CONFIANZA QUE ME HAN DADO.

A MIS HERMANOS, POR SUS BUENOS CONSEJOS, POR SER MI EJEMPLO PARA ESFORZARME CADA DÍA, GRACIAS INFINITAS POR EL AMOR QUE ME HAN DEMOSTRADO. A MIS HERMOSAS GALI, GOGA, CACHI, SHEBA, VALE, DANI, A GABI, A LUDWING Y GABRIEL ESTO VA POR USTEDES.

A MI HERMANITA DEL ALMA Y COMPAÑERA DE TESIS, POR AGUANTARME MIS ACHAQUES E HISTÉRIAS, POR CREER Y CONFIAR EN MÍ, POR ESAS RISAS Y LÁGRIMAS QUE HEMOS COMPARTIDO Y A QUIÉN SIEMPRE VOY AGRADECER POR HABERME DEJADO COMPARTIR ESTE ÉXITO EN NUESTRAS VIDAS. ¡LO LOGRAMOS TESORO! XD.

A NUESTRO AMIGO EDGARDO, POR ESA INMENSA PACIENCIA QUE NOS TUVO. ¡¡GRACIAS!!

A LOS ING. JOSÉ MARIO Y ELOISA, POR QUE SIN ELLOS NO HUBIERAMOS LLEGADO HASTA EL FINAL DE ESTE TRABAJO, A TODOS MIS MAESTROS DE LA UAA. ¡GRACIAS!.

**NORMA OLEA TECANHUEY
MARZO 2013.**

GRACIAS A DIOS POR DARMER VIDA Y SALUD, POR PERMITIRME CONCLUIR UN PROYECTO MÁS.

A MIS PADRES POR CONFIAR EN MÍ Y BRINDARME SU APOYO INCONDICIONAL, ADEMÁS DE ALENTARME DIARIAMENTE PARA NO DEJAR INCONCLUSO ESTE PROYECTO.

A MI ESPOSO POR SU AMOR, PACIENCIA Y GRAN APOYO, Y A MI PEQUEÑO LEO POR SER UN ALICIENTE PARA SUPERARME Y SER UNA MEJOR MAMÁ.

A MI ABUELITA Y HERMANO POR DARMER ÁNIMOS PARA SUPERARME.

A MI QUERIDA AMIGA Y COMPAÑERA DE PROYECTO DE TESIS POR SU APOYO EN LOS MOMENTOS DE INCERTIDUMBRE, POR ACOMPAÑARME EN ESTE LARGO CAMINO, QUE ESPERO SIEMPRE ESTE PRESENTE EN TODAS LAS DÉMAS ETAPAS DE MI VIDA.

AL ING. JOSÉ MARIO MARTINEZ CASTRO POR SER UN VALIOSO PROFESOR DE LA UAA, QUE SIEMPRE NOS BRINDO SU ENSEÑANZA Y MUCHOS ÁNIMOS PARA TERMINAR ESTE TRABAJO.

A TODOS LOS QUE DE ALGUNA U OTRA FORMA FUERON PARTE DE MI FORMACION PROFESIONAL. GRACIAS.

SELENE BENÍTEZ MARTÍNEZ
MARZO 2013.

ÍNDICE GENERAL

ÍNDICE DE FIGURAS	vi
ÍNDICE DE TABLAS	x
INTRODUCCIÓN	10
PLANTEAMIENTO DEL PROBLEMA.....	10
JUSTIFICACIÓN	11
HIPÓTESIS.....	12
OBJETIVO GENERAL	12
OBJETIVOS ESPECÍFICOS	12
CAPÍTULO 1. MARCO TEÓRICO	14
1.1. Base de datos.....	14
1.1.1. Sistemas manejadores de bases de datos	14
1.1.2. Lenguajes de programación	15
1.1.3. Modelos de base de datos	17
1.2. Ingeniería de software	18
1.2.1. Metodología de software	18
1.2.2. Ingeniería Web.....	23
1.2.3. ¿Qué es una aplicación Web?.....	29
1.3. Lenguaje de programación Web	30
CAPÍTULO 2. CASO DE ESTUDIO.....	33
2.1. Análisis	33
2.1.1. Descripción general de la empresa	33
2.1.3. Modelado del negocio	39
2.2. Diseño	48
2.2.1. Modelo de datos.....	48
2.2.2. Arquitectónico.....	52
2.2.3. Diseño de la interfaz de usuario	54
2.2.3. Procedimental	56
CAPÍTULO 3. DESARROLLO E IMPLEMENTACIÓN	61
3.1. Reservas.....	61
3.2. Check-in.....	62
3.3. Servicios.....	64
3.4. Cambiar habitación	65
3.5. Check-out.....	66
CAPÍTULO 4. RESULTADOS Y CONCLUSIONES.....	68
4.1. Pruebas	68

4.2. Conclusiones y trabajo a futuro.....	80
BIBLIOGRAFÍA	81
BIBLIOGRAFÍA INTERFACES	81
ANEXOS	83
A.1. MANUAL DE USUARIO	83
A1.1. ABRIR SESIÓN	83
A1.2. RECEPCIÓN	83
A1.3. CARGAR A LA CUENTA	86
A1.4. ABONAR A LA CUENTA	87
A1.5. IMPRIMIR NOTA	89
A1.6. RESERVACIÓN.....	90
A1.7. CHECK-IN	91
A1.8. CAMBIAR HABITACIÓN.....	93
A1.9. HUÉSPED	94
A1.10. CHECK-OUT.....	95

ÍNDICE DE FIGURAS

	Página
CAPÍTULO 1	
Figura 1.1. Tipología de los lenguajes de los DBMS.....	16
Figura 1.2. Fases del RUP.....	21
Figura 1.3. Casos de uso.....	22
Figura 1.4. El modelo de proceso IWeb.....	24
Figura 1.5. Estructura lineal.....	26
Figura 1.6. Estructura reticular.....	26
Figura 1.7. Estructura jerárquica.....	27
Figura 1.8. Estructura de red.....	28
CAPÍTULO 2	
Figura 2.1. Proceso de reservas.....	35
Figura 2.2. Proceso de check-in.....	35
Figura 2.3. Proceso de recepción.....	36
Figura 2.4. Diagrama de caso de uso general.....	39
Figura 2.5. Diagrama de atención al cliente.....	39
Figura 2.6. Diagrama de caso de uso Ingresar al sistema.....	40
Figura 2.7. Diagrama de caso de uso reserva telefónica.....	41
Figura 2.8. Diagrama de caso de uso reserva internet.....	41
Figura 2.9. Diagrama de caso de uso manejo de cuenta.....	42
Figura 2.10. Diagrama de caso de uso alta huésped.....	43
Figura 2.11. Diagrama de caso de uso check in.....	43
Figura 2.12. Diagrama de caso de uso cambiar habitación.....	44
Figura 2.13. Diagrama de caso de uso cargar a la cuenta.....	45
Figura 2.14. Diagrama de caso de uso abonar cuenta.....	46
Figura 2.15. Diagrama de caso de uso check-out.....	47
Figura 2.16. Diagrama entidad relación.....	48
Figura 2.17. Diseño arquitectónico de la Web.....	52
Figura 2.18. Diseño arquitectónico del sistema.....	54
Figura 2.19. Interfaz prototipo recepción.....	55

Figura 2.20. Interfaz prototipo reservación.....	55
Figura 2.21. Interfaz protocolo check-in.....	56
Figura 2.22. Diseño procedimental reserva.....	56
Figura 2.23. Diseño procedimental check-In.....	57
Figura 2.24. Diseño procedimental servicios.....	58
Figura 2.25. Diseño procedimental cambiar habitación.....	59
Figura 2.26. Diseño procedimental check-out.....	60
CAPÍTULO 3	
Figura 3.1.Reservas.....	61
Figura 3.2.Check-in.....	63
Figura 3.3.Servicios.....	64
Figura 3.4 Cambio de habitación.....	65
Figura 3.5.Check-out.....	66
CAPÍTULO 4	
Figura 4.1. Iniciar sesión.....	68
Figura 4.2. Recepción.....	69
Figura 4.3. Referencias del huésped.....	69
Figura 4.4. Datos del huésped.....	70
Figura 4.5. Seleccionar servicios.....	71
Figura 4.6. Cargar a la cuenta.....	71
Figura 4.7. Abonar a la cuenta.....	72
Figura 4.8. Abono realizado.....	73
Figura 4.9. Imprimir cuenta.....	73
Figura 4.10. Imprimir nota.....	74
Figura 4.11. Reservaciones.....	75
Figura 4.12.Mensaje de reservación realizada.....	75
Figura 4.13. Check-In.....	76
Figura 4.14. Mensaje de check-in realizado.....	76
Figura 4.15. Cambiar habitación.....	77
Figura 4.16. Cambio realizado.....	77
Figura 4.17. Huésped.....	78

Figura 4.18. Se guardó huésped.....	78
Figura 4.19. Check-out.....	79
Figura 4.20. Mensaje de check-out realizado.....	79
ANEXOS	
Figura A1.1. Abrir sesión.....	83
Figura A1.2. Recepción.....	83
Figura A1.3. Encabezado.....	84
Figura A1.4. Accesos directos.....	84
Figura A1.5. Referencias de huésped.....	85
Figura A1.6. Cuenta del huésped.....	85
Figura A1.7. Cargar a la cuenta.....	86
Figura A1.8. Ingresar servicios del huésped.....	86
Figura A1.9. Servicios huésped.....	87
Figura A1.10. Abonar a la cuenta.....	87
Figura A1.11. Cantidad a abonar en la cuenta.....	88
Figura A1.12. Monto reflejado en el sistema.....	88
Figura A1.13. Imprimir nota.....	89
Figura A1.14. Nota de pago de la casa de huéspedes Matti.....	89
Figura A1.15. Ingresar a reservaciones.....	90
Figura A1.16. Búsqueda por nombre del huésped.....	90
Figura A1.17. Reservación realizada.....	91
Figura A1.18. Ingresar a check-in.....	91
Figura A1.19. Guardar check-in.....	92
Figura A1.20. Mensaje de check-in exitoso.....	92
Figura A1.21. Cambiar habitación.....	93
Figura A1.22. Seleccionar destino.....	93
Figura A1.23. Nuevo huésped.....	94
Figura A1.24. Ingresar datos del nuevo huésped.....	94
Figura A1.25. Huésped guardado.....	95
Figura A1.26. Dar check-out.....	95
Figura A1.27. Aviso de check-out.....	96

Figura A1.28. Abono.....	96
Figura A1.29. Recepción.....	97
Figura A1.30. Alerta check-out.....	97
Figura A1.31. Impresión de la nota.....	98

ÍNDICE DE TABLAS

	Página
CAPÍTULO 2	
Tabla 2.1. Requerimientos funcionales del sistemas.....	37
Tabla 2.2. Requerimientos no funcionales del sistemas.....	37
Tabla 2.3. Especificación del caso de uso: Ingresar al sistema.....	40
Tabla 2.4. Especificación del caso de uso: Reserva telefónica.....	41
Tabla 2.5. Especificación del caso de uso: Reserva Internet.....	42
Tabla 2.6. Especificación del caso de uso: Manejo de cuenta.....	42
Tabla 2.7. Especificación del caso de uso: Alta huésped.....	43
Tabla 2.8. Especificación del caso de uso: Check-In.....	44
Tabla 2.9. Especificación del caso de uso: Cambiar habitación.....	45
Tabla 2.10. Especificación del caso de uso: Cargar a la cuenta.....	45
Tabla 2.11. Especificación del caso de uso: Abonar a la cuenta.....	46
Tabla 2.12. Especificación del caso de uso: Check-out.....	47

INTRODUCCIÓN

El software es uno de los productos de la ingeniería que más ha evolucionado en muy poco tiempo llegando hasta el software desarrollado bajo los principios y herramientas de la ingeniería del software.

El presente trabajo titulado “Sistema de administración Web para reservaciones de la casa de huéspedes Matti” se aplica al área de reservaciones de la casa de huéspedes Matti, donde además se llevará el control del check-in y check-out, así como la cuenta del huésped y estado de las habitaciones.

Para el desarrollo del proyecto según la ingeniería de software es necesario aplicar un conjunto de procedimientos racionales para alcanzar los objetivos planteados, además de mejorar el proceso de las actividades que se realizan y las personas que participan, como en este caso la metodología RUP.

Al aplicar la Metodología RUP mediante sus 6 flujos principales de ingeniería que son Modelado del negocio, Requerimientos, Análisis y Diseño, Implementación, Pruebas y Despliegue, se consiguió llevar a cabo la realización de un sistema de calidad que cumpliendo con las expectativas del cliente a su vez alcanzaron los objetivos planteados en esta investigación.

PLANTEAMIENTO DEL PROBLEMA

En la actualidad la casa de huéspedes Matti no cuenta con un método de reservaciones sistematizado, ya que el registro se lleva a cabo mediante una calendarización, con el cual hasta la fecha han obtenido buenos resultados, pero debido al crecimiento de la oferta, se ha hecho necesario contar con una herramienta que facilite la comunicación cliente–administrador, también se necesita obtener mayor celeridad en el proceso de reservación, del manejo de la cuenta del huésped, y en la entrega o liberación de las habitaciones.

Es importante que al usuario se le brinde opciones para la realización de la reservación, utilizando diferentes métodos, ya sea vía Web o telefónica, puesto que en la actualidad este proceso no es posible ofrecerlo. Uno de los problemas importantes de la Casa de huéspedes Matti es la entrega de habitaciones o comunmente llamado Check-In, ya que se hacían reservaciones sin cobro adelantado de la habitación, y llegaban terceros huéspedes sin reservación, queriendo obtener servicio pero no se les podía otorgar ya que las habitaciones estaban apartadas. Los huéspedes que habían hecho la reservación, no avisaban de la cancelación, generándoles así pérdidas financieras.

Para que más personas conozcan los servicios que ofrece la Casa de huéspedes Matti, es necesario crear una página Web, en la cuál ellos puedan conocer el estado físico de sus instalaciones, enviar sus dudas y comentarios, así como el envío de un formulario para una pre-reservación.

Otro problema importante es llevar un control del manejo de la cuenta de los huéspedes, ya que se otorgaban estos servicios y tanto al huésped como al administrador al momento de hacer la cuenta total, se olvidaban de ingresarlos.

JUSTIFICACIÓN

Al principio las páginas Web en Acapulco no tenían la importancia como en el presente, ya este puerto tenía suficiente captación de turistas tanto nacionales como internacionales, haciendo casi innecesario el uso de esta herramienta.

La proyección que tenía Acapulco por las grandes productoras cinematográficas, era suficiente para que la industria hotelera captara muchos huéspedes en todas las temporadas del año.

Sus métodos de reservación se limitaban a lo que tenían en uso en esa época, como era el teléfono, donde reservaban su fecha de ocupación y no tenían manera de tener un comprobante por parte del hotel y en ocasiones llegaban y la reservación no había sido respetada.

Los grandes hoteles se daban el lujo de rechazar huéspedes, ya que eran muy elitistas, era entonces cuando los huéspedes optaban por buscar un espacio en algún hotel, que generalmente eran recomendados por los lugareños y era así como captaban huéspedes los hoteles pequeños.

Ya que fuera de estos, la ocupación era eventual, esto no quiere decir que los hoteles pequeños prestaran un mal servicio o que fueran de poca calidad, ya que al ser conocidos por estos huéspedes, estos últimos los recomendaban a sus allegados para hospedarse ahí, teniendo como único medio de comunicación el teléfono.

Cuando surgieron las primeras páginas Web era muy elevado su costo de producción y de mantenimiento, y sólo era posible para aquellas empresas que pudieran costearlo, ya que en un principio esta tecnología estaba limitada a quién lo podía pagar.

En la actualidad los grandes hoteles no se pueden dar el lujo de rechazar a posibles usuarios, ya que se han abierto otras ofertas turísticas de playa en el país, por lo tanto buscan la manera de captar mayor afluencia de clientes haciendo uso de las tecnologías de información más innovadoras, cómodas y atractivas.

A pesar de que las páginas Web surgieron hace mucho tiempo, aún no se aplicaba porque pensaban que la inversión era muy elevada y ofrecía muy pocos beneficios ya que pocas personas tenían acceso y el conocimiento para utilizar una computadora y la renta de internet era muy cara, sin embargo, hoy en día el servicio de diseño de página Web se ofrece a muy bajo costo, dependiendo la dificultad, espacio y diseño en base a los parámetros y necesidades del cliente, además el costo de una computadora personal y el acceso a internet se encuentran a precios muy accesibles para la mayoría de los usuarios.

HIPÓTESIS

Al implementar un sistema de administración para la casa de Huéspedes Matti, se ofrece un mejor control administrativo de los servicios que se ofrecen.

OBJETIVO GENERAL

Realizar un sistema de administración con interfaz Web para la reservación de la casa de huéspedes Matti.

OBJETIVOS ESPECÍFICOS

- Realizar el Análisis y Diseño de un sistema de administración para la casa de huéspedes Matti.
- Crear una aplicación de administración con los módulos de reservaciones, check-in, check-out, alta de huéspedes, cambio de habitación y abonos para la casa de huéspedes Matti.

CAPÍTULO 1

MARCO TEÓRICO **Conceptos y definiciones básicas**

CAPÍTULO 1. MARCO TEÓRICO

1.1. Base de datos

“La aparición de la expresión base de datos se produce a comienzo de los años 60. En 1963 tuvo lugar en Santa Mónica (EEUU) un simposio en cuyo título se encontraba la expresión data base. En una de sus sesiones, se propuso una definición de base de datos que, según las actas del simposio no fue universalmente aceptada. Posteriormente, en 1967 el grupo de estandarización Codasyl decidió cambiar su primitiva denominación en la que no aparecía base de datos sino Data Base Task Group”. (De Miguel, 2011).

¿Qué es una Base de Datos?

“Es un conjunto, colección o depósito de datos almacenados en un soporte informático no volátil. Los datos están interrelacionados y estructurados de acuerdo con un modelo capaz de recoger el máximo contenido semántico. Dada la relevancia que tienen en el mundo real las interrelaciones entre los datos, es imprescindible que la base de datos sea capaz de almacenar estas interrelaciones. La base de datos se describe y se manipula apoyándose en un modelo de datos, el cual ha de permitir captar estas interrelaciones y restricciones existentes en el mundo real. Los procedimientos de actualización y recuperación, comunes y bien determinados, facilitarán la seguridad del conjunto de los datos”. (De Miguel, 2011).

1.1.1. Sistemas manejadores de bases de datos

Concepto y principales funciones de un Sistema manejador de Base de datos.

“Se puede definir el DBMS (Sistema manejador de base de datos) como un conjunto coordinado de programas, procedimientos, lenguajes, etc. que suministra a los distintos tipos de usuarios los medios necesarios para describir y manipular los datos almacenados en la base, garantizando su seguridad”. (De Miguel, 2011).

Las funciones esenciales de un DBMS son:

- Descripción
- Manipulación
- Control

La descripción permite especificar los elementos de datos que la integran, su estructura y las relaciones que existen entre ellos.

La manipulación se refiere a que una vez descrita la base de datos, es preciso cargar los datos en las estructuras previamente creadas, con lo que la base de datos estará ya dispuesta para su utilización. Los usuarios tendrán necesidad de

recuperar la información (consultar la base de datos), o bien de actualizarla porque se hayan producido cambios en los datos.

La función de control reúne todas las interfaces que necesitan los diferentes usuarios para comunicarse con la base y proporciona un conjunto de procedimientos para el administrador.

Lenguaje de los manejadores de bases de datos

Por las distintas funciones que realiza un DBMS se hace necesario disponer de diferentes tipos de lenguajes y procedimientos que permitan la comunicación con la base de datos, algunos se enfocan en la función de definición o manipulación y otros a los diferentes tipos de usuarios o aplicaciones.

Algunos ejemplos de Lenguajes Manejadores de Base de Datos son, Access, Visual Fox Pro, SQL, Visual Basic.

SQL

“Sus siglas significan Lenguaje de Consulta Estructurado (Structured Query Language), fue diseñado e implementado por IBM Research a modo de interfaz para un sistema de base de datos relacional conocido como SYSTEM R. SQL es ahora el lenguaje estándar de los DBMS relacionales comerciales”. (Elrasmi, 2007).

Es un acceso a Bases de Datos de tipo Relacional, cuenta con sentencias para definir datos, consultas y actualizaciones. Permite realizar distintas operaciones en las bases de datos. A éstas operaciones se les puede dividir en dos grupos que son el DDL (Data Definition Language, Lenguaje de Definición de datos) y DML (Data Manipulation Language, Lenguaje de Manipulación de datos), los cuales corresponden a las operaciones básicas y operaciones no básicas respectivamente.

DDL, las principales funcionalidades de SQL como lenguaje de definición son la creación, modificación y borrado de las tablas que componen la base de datos, así como de los índices, vistas, sinónimos, permisos, etc. Esto quiere decir que realiza las operaciones básicas.

DML, Lenguaje de manipulación de datos, operaciones no básicas se utiliza para obtener, insertar y modificar información de una base de datos.

1.1.2. Lenguajes de programación

Los DBMS tendrán un enfoque distinto según la función y el tipo de usuario según los requerimientos de un programador o del administrador que puede o no tener conocimientos informáticos.

Para los usuarios informáticos (diseñador, analista, programador, etc.) requerirán software potente y flexible para que mediante estos puedan definir, administrar, extraer o manipular la base de datos. Utilizando un lenguaje anfitrión (Cobol, Ensamblador, Fortran, Basic, Pascal, C, Delphi, etc.) de su uso frecuente que le permita hacer llamadas desde la aplicación al DBMS, esto permite acceder a la base de datos. (De Miguel, 2011).

Para el usuario final utiliza medios simples para comunicarse con la base de datos y esto lo logra mediante un lenguaje de manipulación auto contenido que utilice una sintaxis sencilla, pero que permita soportar demandas de información diversas (solo permiten la consulta pero no la actualización), y se presenta al usuario en forma de menú.

Para poder cumplir las funciones en una DBMS, es necesario que se disponga de diferentes tipos de lenguajes y procedimientos que permitan la comunicación con esta última. Algunos están orientados hacia la función (definición o manipulación) y los otros están dirigidos a diferentes tipos de usuarios o de aplicaciones (Figura 1.1.).

Los usuarios finales pueden contener aplicaciones formalizables (como gestión de personal) o no formalizables (como los procesos de toma de decisiones). Los procesos formalizables son repetitivos y el programador escribe los programas correspondientes, que generalmente se llevan a cabo cada cierto tiempo, como emisión diaria de recibos u obtención mensual de la nómina. Los procesos no formalizables no son repetitivos, por lo tanto, no se necesita escribir un programa y la consulta se lleva a cabo directamente mediante los instrumentos que el DBMS pone a su alcance, no necesitándose tener conocimientos informáticos ya que la mayoría contiene procedimientos o programas parametrizados en general de tipo menú.

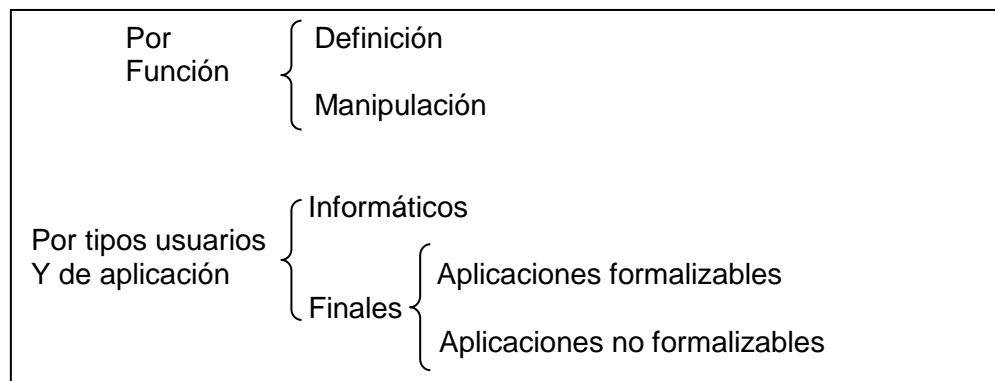


Figura 1.1. Tipología de los lenguajes de los DBMS

1.1.3. Modelos de base de datos

Un modelo de base de datos o esquema de base de datos es la estructura o el formato de una base de datos, descrita en un lenguaje formal soportada por el sistema de gestión de bases de datos. En otras palabras, un "modelo de base de datos" es la aplicación de un modelo de datos usado en conjunción con un sistema de gestión de bases de datos.

Los esquemas generalmente son almacenados en un diccionario de datos. Aunque un esquema se defina en un lenguaje de base de datos de texto, el término a menudo es usado para referirse a una representación gráfica de la estructura de la base de datos.

Definición de modelo de datos

Un modelo de datos es una colección de conceptos que se pueden utilizar para describir la estructura de una base de datos. (Elrasmi, 2007).

Por estructura de una base de datos se refiere a los tipos de datos, relaciones y restricciones que deben mantenerse para los datos. La mayoría de modelos de datos también incluyen un conjunto de operaciones básicas para especificar las recuperaciones y actualizaciones en la base de datos.

Clasificación de los modelos de datos

Se han propuesto muchos modelos de datos, que se clasifican conforme a los tipos de conceptos que utilizan para describir la estructura de la base de datos.

Los modelos de datos de alto nivel conceptual ofrecen conceptos muy cercanos a como muchos usuarios perciben los datos, mientras que los modelos de datos de bajo nivel físicos ofrecen conceptos que describen los detalles de cómo se almacenan los datos en la computadora.

Modelo entidad relación

Los modelos de datos conceptuales utilizan conceptos como entidades, atributos y relaciones. El ejemplo más típico es el Modelo Entidad-Relación. (Korth, 2002)

Una **entidad** representa un objeto o concepto del mundo real, que se distingue de todos los demás objetos, como un empleado o un proyecto que se describe en la base de datos. Una entidad tiene un conjunto de propiedades, y los valores para algún conjunto de propiedades pueden identificar una entidad de forma unívoca.

Un **atributo** representa alguna propiedad de interés que describe a una entidad, como, por ejemplo, el nombre o el salario de un empleado.

Una **relación** representa una asociación entre dos o más entidades; por ejemplo, una relación de trabajo entre un empleado y un proyecto.

Los conceptos ofrecidos por los modelos de datos de bajo nivel están pensados principalmente para los especialistas en manejo de sistemas computacionales, no para los usuarios finales normales.

Existe una clase de modelos de datos representativos (o de implementación), que ofrecen conceptos que los usuarios finales pueden entender pero que no están demasiado alejados de cómo se organizan los datos dentro de una computadora. Los modelos de datos representativos ocultan algunos detalles relativos al almacenamiento de los datos, pero pueden implementarse directamente en una computadora.

1.2. Ingeniería de software

Entre las definiciones de Software se encuentran las siguientes:

- “Programas de ordenadores y la documentación asociada. Los productos de software se pueden desarrollar para algún cliente en particular o para un método general” (Sommerville, 2005).
- “Es un conjunto de programas de computadora, estructuras de datos y su documentación que sirven para hacer efectivo el método lógico, procedimiento o control requerido” (Pressman, 2001).

La Ingeniería de software es una disciplina de la Ingeniería para la producción de software desde sus etapas iniciales como la especificación de sistema hasta el mantenimiento después de haberse concluido y estando en uso.

Por disciplina se refiere aplicar teorías, métodos y herramientas donde sean conveniente, seleccionando lo necesario para solucionar el problema, aun no existiendo teorías y métodos para resolverlos.

La ingeniería de software no solo comprende los procesos técnicos del desarrollo de software, sino también la gestión de proyectos de software y el desarrollo de herramientas, métodos y teorías de apoyo a la producción de software.

1.2.1. Metodología de software

“Un método de ingeniería de software es un enfoque estructurado para el desarrollo de software cuyo propósito es facilitar la producción de software de alta calidad de una forma costeable” (Sommerville, 2005).

En otras palabras, la rama de la metodología, dentro de la ingeniería de software, se encarga de elaborar estrategias de desarrollo de software que promuevan

prácticas adaptativas en vez de predictivas; centradas en las personas o los equipos, orientadas hacia la funcionalidad y la entrega, de comunicación intensiva y que requieren implicación directa del cliente.

Todos los métodos se basan en la idea de modelos gráficos de desarrollo de un sistema y en el uso de estos modelos como un sistema de especificación o diseño.

Una metodología define:

- Estados, etapas o fases de un desarrollo, junto con los criterios de transición entre ellos.
- Tareas, actividades, etc.
- Roles, con sus habilidades necesarias y las interacciones entre ellos.
- Artefactos o entregables.
- Herramientas de control, seguimiento, medición y perfeccionamiento.
- Principios, criterios para tomar decisiones, estrategias para manejar distintos tipos de situaciones, herramientas de manejo de riesgos, etc.

Tener metodologías diferentes para aplicar de acuerdo con el proyecto que se desarrolle resulta imprescindible teniendo en cuenta las necesidades cambiantes que tiene el entorno de desarrollo actual y el acelerado progreso de la informática a nivel mundial resulta una idea interesante. Estas metodologías pueden involucrar prácticas tanto de metodologías ágiles como de metodologías tradicionales.

Metodologías tradicionales y metodologías ágiles

Teniendo en cuenta la filosofía de desarrollo de las metodologías, aquellas con mayor énfasis en la planificación y control del proyecto, en especificación precisa de requisitos y modelado, reciben el apelativo de Metodologías Tradicionales o Pesadas. (Sommerville, 2005).

Estas metodologías tradicionales imponen una disciplina de trabajo sobre el proceso de desarrollo del software, con el fin de conseguir un software más eficiente.

Las metodologías tradicionales no se adaptan adecuadamente a los cambios, por lo que no son métodos adecuados cuando se trabaja en un entorno, donde los requisitos no pueden predecirse o bien pueden variar.

Los procesos ágiles son una buena elección cuando se trabaja con requisitos desconocidos o variables. Si no existen requisitos estables, no existe una gran posibilidad de tener un diseño estable y de seguir un proceso totalmente planificado, que no vaya a variar ni en tiempo ni en dinero.

El método está condicionado por el tamaño y estructura de la organización, y el tipo de aplicaciones.

Para el desarrollo del software de la “Casa de huéspedes Matti” se pretende aplicar la metodología RUP, ya que es un proyecto de mediana complejidad y es preciso que se cuente con documentación acerca del proceso de construcción del software para su posterior mantenimiento y evolución.

Proceso Unificado de Rational (RUP)

El Proceso Unificado de Rational fue creado por el mismo grupo de expertos que crearon *UML*, Ivar Jacobson, Grady Booch y James Rumbaugh en el año 1998. (Kareny, 2009)

Es un proceso dirigido por casos de uso, este avanza a través de una serie de flujos de trabajo (requisitos, análisis, diseño, implementación, prueba) que parten de los casos de uso; está centrado en la arquitectura y es iterativo e incremental. Además cubre el ciclo de vida de desarrollo de un proyecto y toma en cuenta las mejores prácticas a utilizar en el modelo de desarrollo de software.

RUP divide el desarrollo en 4 fases que definen su ciclo de vida:

- **Inicio:** Se describe el negocio y se delimita el proyecto describiendo sus alcances con la identificación de los casos de uso del sistema.
- **Elaboración:** Etapa en la que se determina la arquitectura óptima del proyecto se obtiene una aplicación ejecutable que responde a los casos de uso que la comprometen.
- **Construcción:** Se obtiene un producto listo para su utilización que está documentado y tiene un manual de usuario. Se obtiene uno o varios demos del producto que han pasado las pruebas. Se ponen estos demos a consideración de un subconjunto de usuarios.
- **Transición:** Obtener el producto acabado y definido, listo para su instalación en condiciones reales.

Flujos del RUP

En RUP se han agrupado las actividades en grupos lógicos definiéndose 9 flujos de trabajo principales. Los 6 primeros son conocidos como flujos de ingeniería y los tres últimos como de apoyo (Figura 1.2.). (Anay, 2009)

- **Ingeniería o modelado del negocio:** Analizar y entender las necesidades del negocio para el cual se está desarrollando el software.

- **Requisitos:** Proveer una base para estimar los costos y tiempo de desarrollo del sistema.
- **Análisis y diseño:** Trasladar los requisitos analizados anteriormente a un sistema automatizado y desarrollar una arquitectura para el sistema.
- **Implementación:** Crear software que se ajuste a la arquitectura diseñada y que tenga el comportamiento deseado.
- **Pruebas:** Asegurarse de que el comportamiento requerido es correcto y que todo lo solicitado está presente.
- **Despliegue:** Producir distribuciones del producto y distribuirlo a los usuarios.

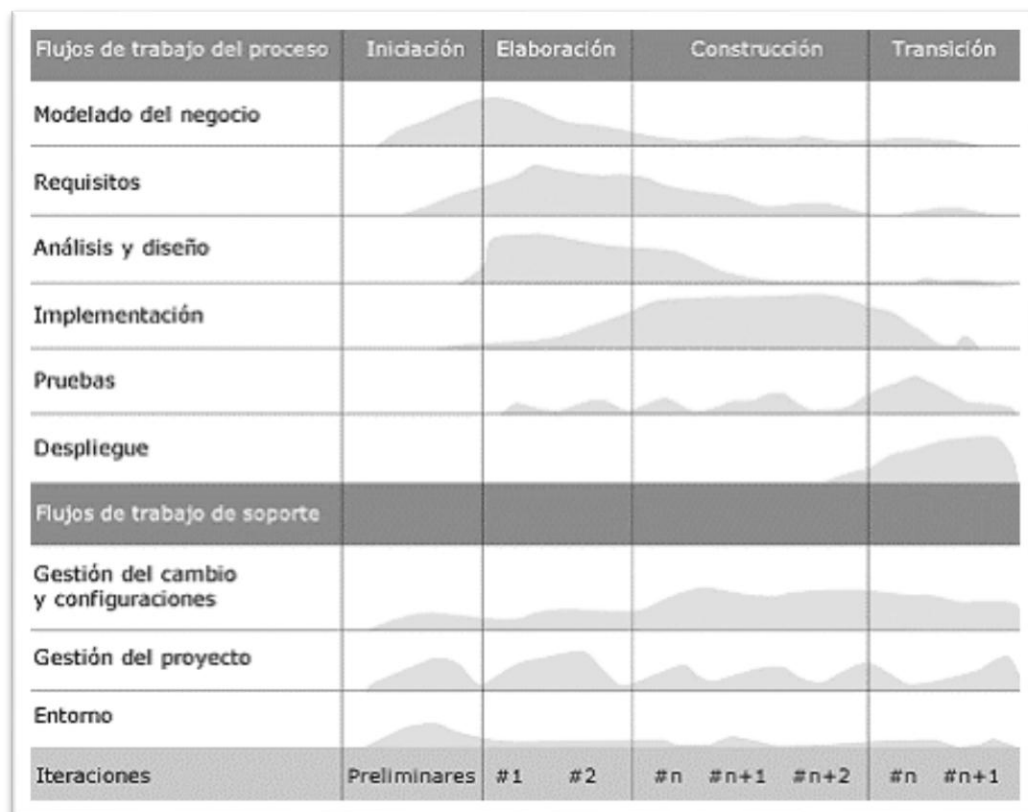


Figura 1.2. Fases del RUP

Para el desarrollo del Sistema de administración Web para reservaciones de la casa de huéspedes Matti se aplica la siguiente estructura de las fases de RUP, tomando sólo en cuenta el flujo de trabajo del proceso y quedando como trabajo a futuro los flujos de trabajo a soporte.

Modelado del negocio

Durante el proceso de modelado del negocio, se examina la estructura de la organización y se observan los roles en la compañía y como estos se relacionan.

Una organización tiene una serie de objetivos que satisface a través de Procesos de Negocio.

Etapas del modelado del negocio:

- Identificar y definir los procesos de negocio según los objetivos de la organización.
- Definir un caso de uso del negocio para cada proceso del negocio (diagrama de casos de uso del negocio muestra el contexto y los límites de la organización).

Casos de uso

En el contexto de ingeniería del software, un caso de uso es una secuencia de interacciones que se desarrollarán entre un sistema y sus actores en respuesta a un evento que inicia un actor principal sobre el propio sistema.

Los diagramas de casos de uso sirven para especificar la comunicación y el comportamiento de un sistema mediante su interacción con los usuarios. La siguiente imagen esquematiza las partes de un diagrama de casos de uso (Figura 1.3.).

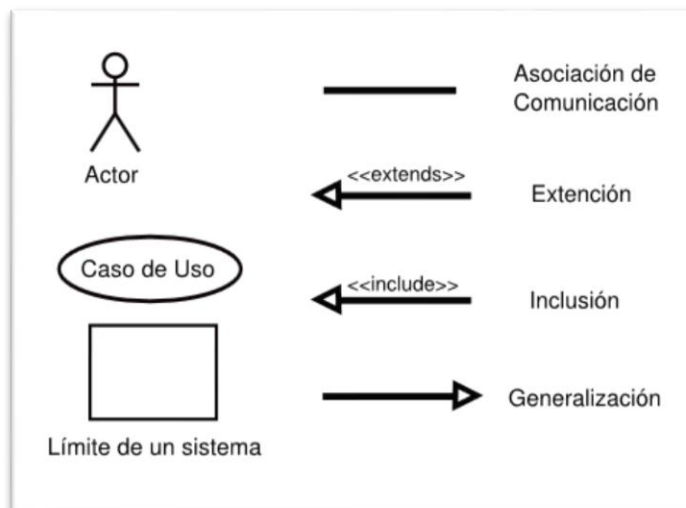


Figura 1.3. Casos de uso

Los casos de uso son herramientas simples para describir el comportamiento de sistema. Un caso de uso contiene una descripción textual de todas las maneras que los actores podrían trabajar con el sistema y muestran los pasos que el actor sigue para realizar una operación.

Un caso de uso debe:

- Describir una tarea del negocio.

- Tener un nivel apropiado de detalle.
- Ser bastante sencillo y entendible para los clientes.

Requerimientos

Los requerimientos para un sistema son la descripción de los servicios proporcionados por el sistema y de las restricciones operativas. Estos requerimientos reflejan las necesidades del cliente. Definiendo en forma detallada y formal el funcionamiento del sistema. (Sommerville, 2005).

La lista de requerimientos es el resultado de una serie de entrevistas directas a los involucrados (usuarios, clientes y expertos del dominio). Los requerimientos se obtienen y se definen desde el punto de vista del usuario, lo que documenta el comportamiento que tendrá el sistema de software dentro de los casos de uso.

Los requerimientos de sistemas software se clasifican en funcionales y no funcionales.

Requerimientos funcionales: Son declaraciones de los servicios que debe proporcionar el sistema. Describen lo que el sistema debe hacer. Los requerimientos funcionales son las capacidades que el sistema de software debe cumplir.

Requerimientos no funcionales: Son restricciones de los servicios. Incluyendo restricciones de tiempo, el proceso de desarrollo y estándares.

1.2.2. Ingeniería Web

Ofrece una solución de comercio electrónico a las empresas que han decidido comercializar y administrar sus productos a través de Internet mediante una tienda virtual y que además es la aplicación de metodologías sistemáticas, disciplinadas y cuantificables al desarrollo eficiente, operación y evolución de aplicaciones de alta calidad en la World Wide Web.

Aunque la ingeniería Web utiliza principios de ingeniería de software, este no es un subconjunto o clon de la ingeniería de software, ya que incluye metodologías, herramientas, técnicas y patrones necesarios para cubrir los requisitos de las aplicaciones Web.

El proceso empieza con la formulación del problema que puede resolverse mediante las aplicaciones Webs, se planifica el proyecto y se analizan sus requisitos, entonces se realiza el diseño de interfaces arquitectónico y del navegador.

El sistema se implementa utilizando lenguajes y herramientas especializados asociados con la Web, entonces comienzan las pruebas. Dado que las WebApps

están en constante evolución, debe de establecerse los mecanismos para el control de configuraciones, garantía de calidad y soporte continuado.

Los atributos de aplicaciones basadas en Web

A las aplicaciones basadas en Web también se les llama *WebApps* y son muy diferentes de las otras categorías de software informático. (Pressman, 2001). Algunas diferencias básicas serían que implican una mezcla de publicación impresa y desarrollo de software de marketing e informática, de comunicaciones internas y relaciones externas, de arte y tecnología. Los atributos más importantes son:

- **Intensivas de Red.** Reside en una red y debe dar servicio a las necesidades de una comunidad diversa de clientes.
- **Controlada por el contenido.** En muchos casos, la función primaria de una WebApp es utilizar hipertexto para presentar al usuario el contenido de textos, gráficos, sonido y vídeo.
- **Evolución continúa.** A diferencia del software de aplicaciones convencional, que evoluciona con una serie de versiones planificadas y cronológicamente espaciadas, las aplicaciones Web están en constante evolución. Esto quiere decir que el contenido de las WebApps se actualice cada cierto tiempo.

Marco de trabajo para la IWeb

Debido a la evolución de las WebApps, la utilización de recursos estáticos de información de la cual el contenido era controlado, se empieza a utilizar entornos de aplicaciones dinámicas que son controladas por el usuario. Y para conseguir esto se necesita desarrollar un marco de trabajo IWeb que asista a un modelo de proceso eficaz (Figura 1.4.). (Pressman, 2001).

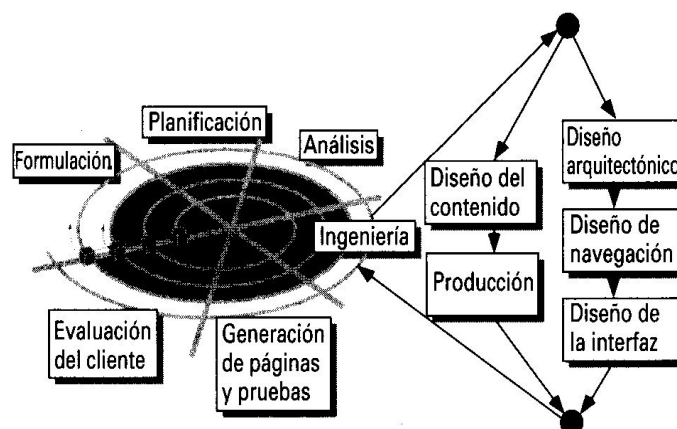


Figura 1.4. El modelo de proceso IWeb

El primer proceso que utiliza IWeb es la formulación, es la actividad que va a identificar las metas y los objetos de la Aplicaciones basadas en Web y establecera el ámbito del primer incremento.

La planificación, estima el coste global del proyecto y evalúa los riesgos asociados con el esfuerzo del desarrollo, y define una planificación del desarrollo para el incremento final de la WebApp, aunque también planificará para los incrementos subsiguientes.

El Análisis va a establecer los requisitos técnicos e identifica los elementos del contenido que se van a incorporar. También define los requisitos del diseño gráfico (estética).

La Ingeniería incorpora dos tareas paralelas que son el Diseño del contenido y la Producción. Y el objetivo de ambas es el de diseñar, producir, y/o adquirir todo el contenido de texto, gráfico y video que se vayan a integrar en la WebApp.

La generación de páginas es donde se realiza la construcción haciendo uso de herramientas automatizadas para la creación de la WebApp, el contenido que se define en Ingeniería se fusiona con los Diseños Arquitectónicos, de navegación y de la Interfaz para elaborar páginas Web ejecutables. Las pruebas ejercitan la navegación, se intentan descubrir los errores de las applets, guiones y formularios y ayuda a asegurar que las WebApp funcionen correctamente en diferentes navegadores o entornos.

En la actividad de evaluación del cliente, se solicitan cambios, esto quiere decir que el cliente verificará que las WebApp contengan lo necesario y puedan tener lugar a ampliaciones del ámbito.

Diseño para aplicaciones basadas en Web

Las aplicaciones Web permiten la generación automática de contenido, la creación de páginas personalizadas según el perfil del usuario o el desarrollo del comercio electrónico. Además, una aplicación Web permite interactuar con los sistemas informáticos de gestión de una empresa, como puede ser gestión de clientes, contabilidad o inventario, a través de una página Web.

Las aplicaciones Web se encuadran dentro de las arquitecturas cliente/servidor: una computadora solicita servicios (el cliente) y otro está a la espera de recibir solicitudes y las responde (el servidor). (Mora, 2005)

- **Modelado de datos**

El modelo de datos es un mecanismo formal para representar y manipular información de manera general y sistemática, algunos de sus ejemplos conocidos son: Modelados basados en grafos (red-jerárquico), modelo relacional, modelo orientado a objetos, etc. (Date, 2001).

El modelo relacional es representado mediante entidades y relaciones que se simbolizan en forma de tablas, estas reciben el nombre de relación, las filas contienen datos sobre cada entidad, las columnas corresponden a los atributos de las entidades.

El modelo entidad/relación es una técnica de análisis basada en la identificación de las entidades y de las relaciones que se dan entre ellas en la parte de realidad que pretendemos modelar. Este modelo permite representar de forma abstracta los datos que se pretenden almacenar en una base de datos.

En el modelado de datos entra también el diccionario de datos el cual es un almacén que contiene definiciones de todos los objetos de datos consumidos y producidos por el software.

- **Diseño arquitectónico**

Se centra en la definición de la estructura global **hipermedia** para la WebApp, y en la aplicación de las configuraciones de diseño y plantillas constructivas para popularizar la estructura esto con el fin de reutilizarlas. Y existen cuatros opciones disponibles para su diseño y son:

- **Las estructuras lineales** aparecen cuando es común la sucesión predecible de interacciones (con alguna variación o diversificación). Un ejemplo clásico podría ser la presentación de un manual de usuario en la que las páginas de información se presentan con gráficos relacionados, vídeos cortos o sonido solo después de haber presentado un prerequisite. La sucesión de presentación del contenido queda predefinida y se puede decir que, generalmente, es lineal (Figura 1.5.). (Pressman, 2001).

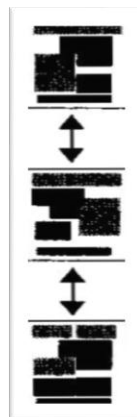


Figura 1.5. Estructura lineal

- **Las estructuras reticulares** son una opción arquitectónica que puede aplicarse cuando el contenido de la WebApp puede ser organizado categóricamente en dos dimensiones (o más) (Figura 1.6.)

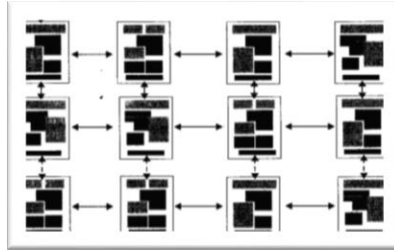


Figura 1.6. Estructura reticular

- **Las estructuras jerárquicas** se organizan en niveles múltiples de acuerdo a una estricta relación PADRE/HIJO, de manera que un padre puede tener más de un hijo, todos ellos localizados en el mismo nivel, y un hijo únicamente puede tener un padre situado en el nivel inmediatamente superior al suyo. Esta estricta relación implica que no puedan establecerse relaciones entre segmentos dentro de un mismo nivel (Figura 1.7.).

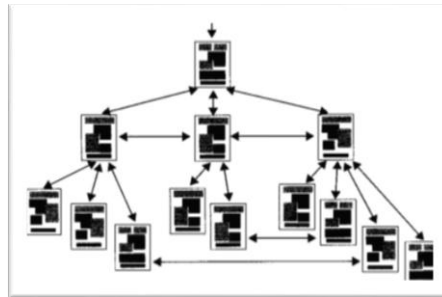


Figura 1.7. Estructura jerárquica

- **Las estructuras en red** se asemejan a la arquitectura en evolución para los sistemas orientados a objetos. Los componentes arquitectónicos (en este caso las páginas Web) se diseñan de forma que pueden pasar el control (mediante enlaces de hipertexto) a otros componentes del sistema. Este enfoque permite una flexibilidad de navegación considerable, aun cuando puede resultar confuso para el usuario (Figura 1.8.).

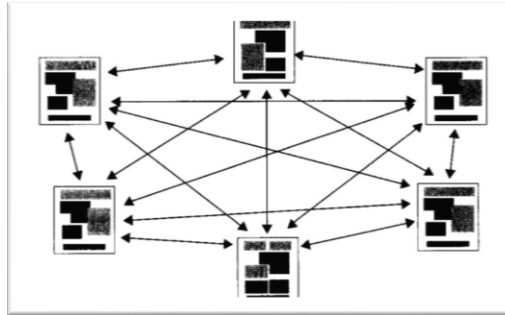


Figura 1.8. Estructura de red

- **Diseño de interfaz**

Los requerimientos de interfaz son todos aquéllos elementos que debe proveer el sistema para permitir la interacción entre el usuario y las funcionalidades que éste tiene, con el fin de que en el proceso de diseño se tenga claridad de las interfaces que se deben crear y la relación que debe existir entre ellas.

Para la definición de los requerimientos de interfaz se deben identificar los siguientes elementos:

- Id, identifica de manera única una interfaz gráfica.
- Descripción, indica los elementos que debe tener la interfaz.
- Requerimientos asociados, indican las funcionalidades asociadas a la interfaz gráfica.

Los elementos del diseño de interfaz de usuario crea tres reglas de oro para el diseño de la interfaz (Mandel, 1997):

1. Dar el control al usuario.
2. Reducir la carga de memoria del usuario.
3. Construir una interfaz consecuente.

- **Diseño a nivel de componentes**

El diseño a nivel de componentes, llamado también diseño procedimental, tiene lugar después de haber establecido los diseños de datos, arquitectura e interfaces. El objetivo es convertir el modelo de diseño en un software operacional.

Existen herramientas gráficas, tales como diagramas de flujo o diagramas de cajas, proporcionan formas gráficas excelentes que representan datos procedimentales fácilmente.

El diagrama de flujo es una imagen bastante sencilla. Mediante la utilización de una caja se indica un paso del proceso. Un rombo representa una condición lógica y las flechas indican el flujo de control.

1.2.3. ¿Qué es una aplicación Web?

Una aplicación Web (Web-based application) es un tipo especial de aplicación cliente/servidor, donde tanto el cliente (el navegador, explorador o visualizador) como el servidor (el servidor Web) y el protocolo mediante el que se comunican (HTTP) están estandarizados y no han de ser creados por el programador de aplicaciones. En otras palabras, es una aplicación software que se codifica en un lenguaje soportado por los navegadores Web en la que se confía la ejecución al navegador. (Mora, 2005).

Las aplicaciones Web se encuadran dentro de las arquitecturas cliente/servidor: una computadora solicita servicios (el cliente) y otro está a la espera de recibir solicitudes y las responde (el servidor).

Cliente Web

El cliente Web es un programa con el que interacciona el usuario para solicitar a un servidor Web el envío de los recursos que desea obtener mediante HTTP. La parte cliente de las aplicaciones Web suele estar formada por el código HTML que forma la página Web más algo de código ejecutable realizado en lenguaje de script del navegador (JavaScript o VBScript) o mediante pequeños programas (applets) realizados en Java. También se suelen emplear plug-ins que permiten visualizar otros contenidos multimedia (como Macromedia Flash), aunque no se encuentran tan extendidos como las tecnologías anteriores y plantean problemas de incompatibilidad entre distintas plataformas. Por tanto, la misión del cliente Web es interpretar las páginas HTML y los diferentes recursos que contienen (imágenes, sonidos, etc.). (Mora, 2005).

Servidor Web

El servidor Web es un programa que está esperando permanentemente las solicitudes de conexión mediante el protocolo HTTP por parte de los clientes Web. Su único propósito es proveer datos de modo que otras máquinas puedan utilizar esos datos.

Servidores portables

Un servidor portable montado en una memoria usb nos puede resultar útil cuando necesitamos ejecutar una Web dinámica sin disponer de acceso a Internet y no podemos, o no queremos, instalar un servidor completo en la computadora.

Su instalación y configuración se realiza de manera sencilla, algunos paquetes contienen: Apache, phpMyAdmin, MiniRelay y MySQL. (Vries, 2010).

1.3. Lenguaje de programación Web

Conforme fueron surgiendo diferentes demandas por los usuarios debido a los nuevos avances informáticos, se les iban dando solución mediante los lenguajes estáticos. Cuando las tecnologías fueron desarrollándose surgieron nuevos problemas a solucionar. Esto dio lugar a desarrollar lenguajes de programación para la Web dinámicos, que permitieran interactuar con los usuarios y utilizarán Sistemas de Bases de Datos.

Actualmente existen diferentes lenguajes de programación para desarrollar en la Web. Los lenguajes de programación facilitan la tarea de programación, ya que disponen de formas adecuadas que permiten ser leídas y escritas por personas, a su vez resultan independientes, no importando el modelo de computadora a utilizar.

Los lenguajes de programación están formados por un conjunto de símbolos y reglas sintácticas y semánticas específicas que definen su estructura y el significado de sus elementos y expresiones. Un lenguaje de programación es un idioma artificial diseñado para expresar cálculos que pueden ser llevados a cabo por las computadoras. Puede usarse para crear programas que controlen el comportamiento físico y lógico de una máquina, para expresar algoritmos con precisión o como modo de comunicación humana.

Los lenguajes de programación más usados en desarrollo Web son principalmente: ASP.NET, PHP y JSP, aunque aún hay quienes usan ASP, Macromedia ColdFusion y Perl.

Lenguaje HTML

Acrónimo de HyperText Markup Language, el **HTML** es un lenguaje de marcación especialmente ideado para permitir la creación de contenidos basados en el hipertexto, es decir, enlaces que permiten desplazarnos a través de diversos documentos en cualquier momento.

Su funcionamiento se basa en el uso de etiquetas o códigos, que permiten dotar de unas determinadas características y propiedades a los textos y elementos que engloban (www.w3.org).

Ejemplo de código html:

```
<HTML>
<HEAD>
<TITLE>Hola mundo</TITLE>
</HEAD>
<BODY>
<P>Hola Mundo</P>
```

```
</BODY>  
</HTML>
```

Lenguaje PHP

Es un lenguaje interpretado (scripting), “open source”, que permite la generación dinámica de contenidos en un servidor Web, el cual puede ser embebido en páginas HTML. El significado de sus siglas es HyperText Preprocessor. Entre sus principales características cabe destacar su potencia, su alto rendimiento, su facilidad de aprendizaje ya que su sintaxis es similar a C y su escasez de consumo de recursos (www.php.net).

Ejemplo de código php:

```
<?php  
$cadena = 'Hola mundo';  
echo $cadena;  
?>
```

CAPÍTULO 2

CASO DE ESTUDIO

Desarrollo e Implementación

CAPÍTULO 2. CASO DE ESTUDIO

2.1. Análisis

2.1.1. Descripción general de la empresa

La casa de huéspedes Matti abrió sus puertas en el año 1990, el cual cuenta 3 habitaciones, un bungalow y un departamento. Desde sus inicios la captación de huéspedes se realizaba por medio de conocidos que se alojaban en la casa y estos a su vez platicaban a sus amistades invitándoles a alojarse en la casa cuando decidieran visitar Acapulco.

Las operaciones de reservación se realizaban mediante papel y lápiz, calendarizando las salidas y llegadas de los huéspedes que hacían su reservación vía telefónica, esta era la única forma que utilizaban para hacerlas. Con el paso del tiempo y la llegada de las nuevas tecnologías de la información y comunicación, se anexó el uso del e-mail y en las últimas fechas la red social Facebook, para así poder tener una mayor captación.

Respecto a las cancelaciones, el problema al que se enfrentan es la pérdida de fechas disponibles al no contar con un sistema eficiente, ya que en muchas ocasiones no hacen con debido tiempo la cancelación o nunca la llegan a hacer. Ocasionándoles muchas pérdidas financieras.

Subsistema de reservas

El Subsistema de Reserva contempla tres de las actividades fundamentales del negocio, hacer una reserva, realizar un check-in y realizar un check-out. La empresa penalizará a aquellos clientes que no cancelen sus reservas, por lo que se les cobrará por dicho motivo.

Procesos de negocio

Los siguientes procesos de negocio son relativos al Subsistema de Reservas:

- **Recepción (P1)**

Este proceso es el principal, en el cual manda a llamar a otros procesos importantes como el de Reserva, el Check-In y Cambio de habitaciones. En él también se encuentran otros subprocesos como es el de Cargar a la cuenta, Abonar a la cuenta, y el Check-Out, además de que se visualizan las referencias de los huéspedes (Figura 2.1.).

Cargar a la cuenta, es la cuenta general en el cual se visualizarán los cargos que tiene el huésped, tanto de los servicios que ha solicitado como del cargo por los días ocupados de las habitaciones.

Abonar a la cuenta, se empleará cuando el huésped vaya a realizar un pago de su cuenta.

Check-out se realizará después de que el huésped haya finiquitado su cuenta y abandone el hotel.

- **Reserva (P2)**

Este proceso administra todas las actividades de reservación, involucrando la creación de nuevas reservas, modificaciones, búsqueda y cancelaciones. La reserva quedará confirmada cuando se realice el depósito correspondiente (Figura 2.2.).

- **Check-in (P3)**

Este proceso se realiza cuando un huésped ocupe la habitación, ya sea teniendo una Reservación Confirmada y/o Directa (que será cuando el cliente llegue sin reservación). Se incluyen también Cambio de habitación, Huésped, Servicios huésped y Editar Check-in.

En referencia al **Cambio de habitación** sólo se realizará en caso de que el huésped no esté conforme con su habitación actual o por alguna necesidad de la casa.

Servicios huésped realizará cargos hechos por servicios utilizados como son tintorería, telefonía, bebidas, etc.

Huésped, se ingresará los datos del nuevo huésped quedando habilitadas para una próxima visita (Figura 2.3.).

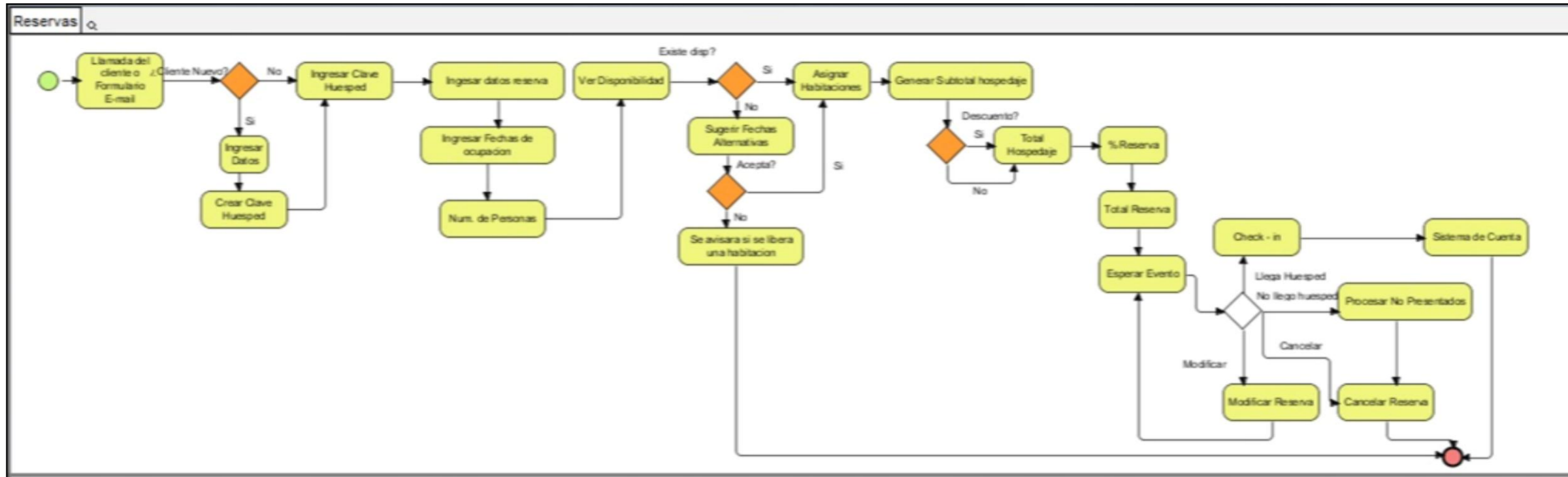


Figura 2.1. Proceso de Reservas

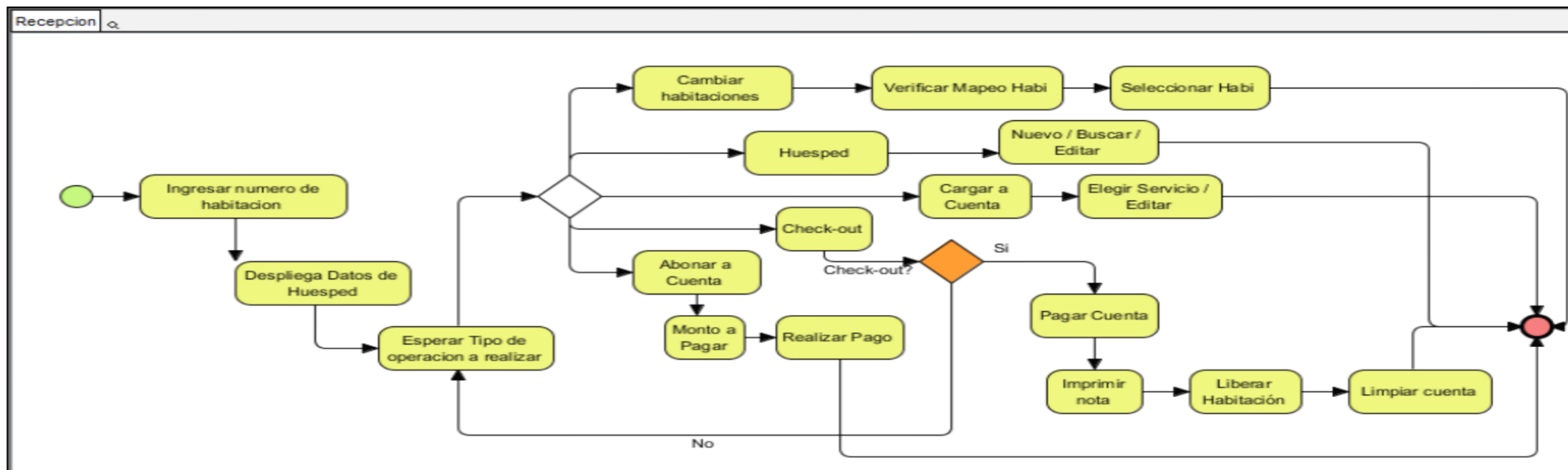


Figura 2.2. Proceso de Check-In

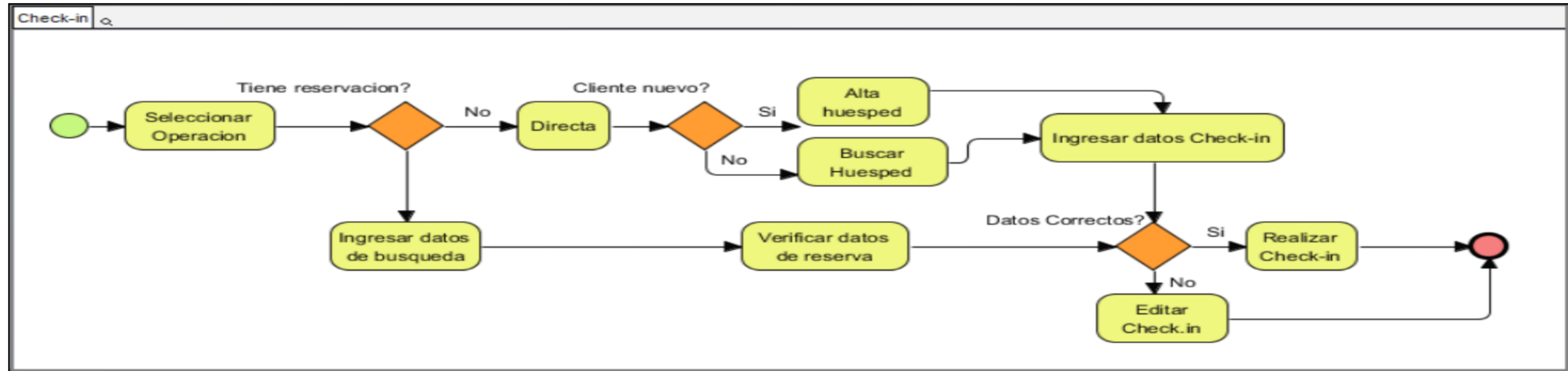


Figura 2.3. Proceso de Recepción

2.1.2. Requerimientos

Funcionales

En esta sección se da a conocer la lista de requerimientos funcionales del Sistema de administración Web para reservación de la casa de huéspedes Matti que se desarrollará. Los requerimientos funcionales permiten expresar una especificación más detallada de las responsabilidades del sistema que se propone.

Los requerimientos funcionales del sistema propuestos son los siguientes (Tabla 2.1.).

Tabla 2.1. Requerimientos funcionales del sistema	
Identificador del Requerimiento	Descripción
CM_RF1	El sistema tendrá que validar al usuario del mismo por medio de un usuario único y contraseña.
CM_RF2	El sistema permitirá al usuario hacer registros de reservaciones vía Web, telefónica.
CM_RF3	El usuario tendrá la posibilidad de acceder a todos los registros de los huéspedes.
CM_RF4	El sistema permitirá al usuario acceder a la cuenta del huésped para hacer cargos o abonos.
CM_RF5	El sistema permitirá al usuario realizar check-in y check-out en el momento que el huésped lo solicite, con previo pago de la cuenta.
CM_RF6	El sistema permitirá cambiar al huésped de habitación si es necesario.
CM_RF7	El sistema mostrará estado actual de las habitaciones.

No Funcionales

Los requerimientos no funcionales especifican cualidades, propiedades del sistema; como restricciones del entorno o de la implementación, rendimiento o dependencia de la plataforma. Y son los siguientes (Tabla 2.2.).

Tabla 2.2. Requerimientos no funcionales del sistemas		
Clasificación	Identificador del Requerimiento	Descripción
Interfaz	CM_RNF1	El sistema debe tener una interfaz gráfica sencilla, amigable y muy legible.
	CM_RNF2	El sistema proporcionará íconos de fácil comprensión que conduzcan al usuario hasta el contenido que desea y evitarle navegar a través de muchas áreas.
	CM_RNF3	La información debe ser mostrada de manera clara para el usuario.
	CM_RNF4	Para mejor guía la pantalla principal siempre

		permanecerá abierta para el usuario. Abriendo otra página para cada sección la cual se cerrará a petición del usuario.
Usabilidad	CM_RNF5	Por petición de cliente solo existirá un usuario único que tendrá acceso a todos los módulos del sistema.
	CM_RNF6	El sistema está diseñado para ser utilizado por personas con mínimos conocimientos en el manejo de la computadora y el ambiente Web en sentido general, debido a que contará con una capacitación previa y soporte técnico.
	CM_RNF7	Se permite la utilización de dispositivos de entrada como el teclado y el mouse.
Seguridad	CM_RNF8	Se usarán mecanismos de encriptación que por cuestiones de seguridad no deben viajar al servidor en texto plano como es el caso de la contraseña. Se guardará encriptada esta información en la base de datos.
	CM_RNF9	El sistema permitirá abrir sesión al usuario y al momento de cerrarla desde cualquier módulo ya no podrá ingresar al sistema al menos que vuelva a iniciar sesión.
	CM_RNF10	El usuario accederá de manera rápida y operativa al sistema sin que los requerimientos de seguridad se conviertan en un retardo para ellos.
Software	CM_RNF11	El sistema se deberá ejecutar en entornos Windows, Linux, etc.(Multiplataforma), para su ejecución del lado del servidor necesita MySQL como sistema gestor de base de datos y Apache como servidor Web, del lado del huésped cualquiera de los exploradores Web existentes en el mercado (IE, Firefox, Chrome, Ophera).
	CM_RNF12	Por petición expresa del usuario se utilizara un servidor portable, debido a Que desean llevar su sistema a otras computadoras cuando no se encuentre en casa.
Hardware	CM_RNF12	Se requiere de una computadora como servidor de base de datos con los requerimientos de hardware que necesita MySQL. Las terminales clientes sólo requerirán de una computadora conectada a la red, para poder ejecutar los navegadores de Web al menos deben cumplir los requisitos mínimos que requiera el navegador en cuestión.

2.1.3. Modelado del negocio

El siguiente caso de uso es una colección de situaciones respecto al uso del sistema Casa Matti (Figura 2.4.), donde cada escenario describirá una secuencia de eventos. Cada secuencia se lleva a cabo por la interacción de los actores huésped y usuario, donde el resultado de las secuencias es algo que ellos utilizan.

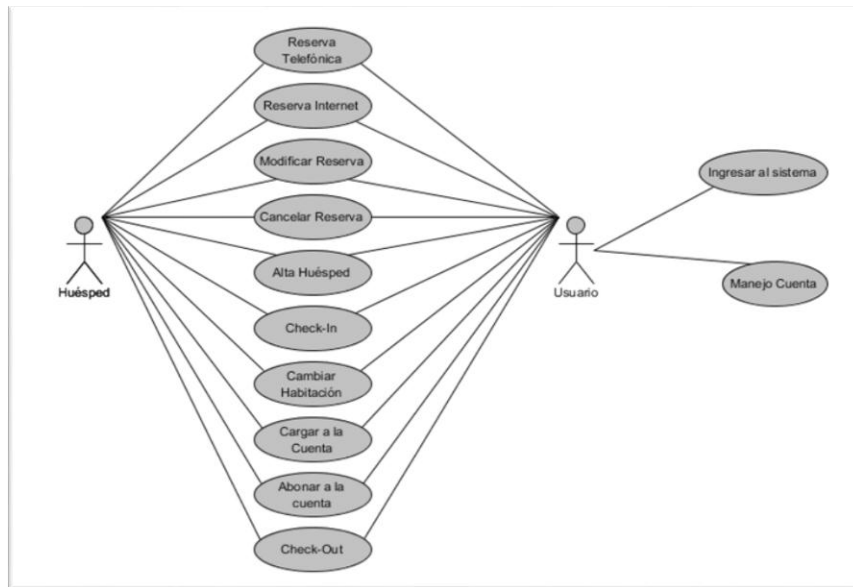


Figura 2.4. Diagrama de caso de uso general

Referente a la operatividad del sistema los siguientes casos de uso se refieren a la interacción con el Huésped, en los diferentes procesos que el huésped puede llegar a llevar a cabo (Figura 2.5.).

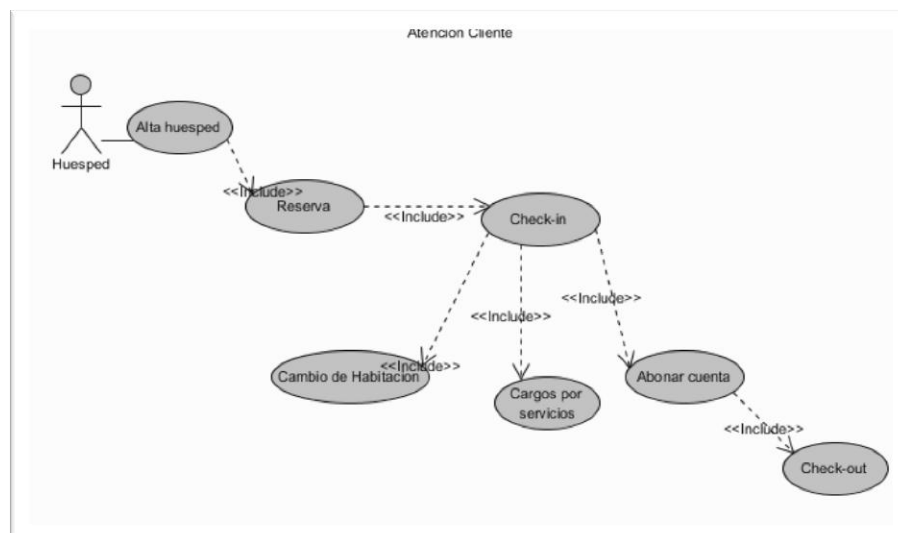


Figura 2.5. Diagrama de atención al cliente

El sistema Casa Matti cuenta con los siguientes casos de uso que se describen a continuación.

- Ingresar al sistema.
- Reserva telefónica
- Reserva internet
- Manejo de cuenta
- Alta huésped
- Check-In
- Cambiar habitación
- Cargar a la cuenta
- Abonar a la cuenta
- Check-Out

Caso de uso: Ingresar al sistema

Este caso de uso permite al usuario ingresar al sistema (Figura 2.6.), el cual se encuentra especificado a continuación (Tabla 2.3.).

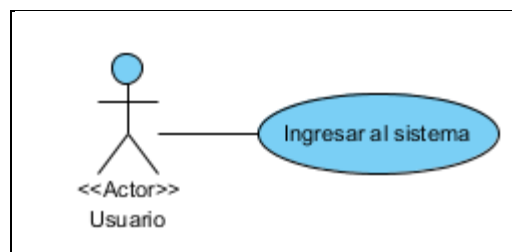


Figura 2.6. Diagrama de caso de uso ingresar al sistema

Tabla 2.3. Especificación del caso de uso: Ingresar al sistema	
Código	1
Actores	Usuario (administrador).
Pre condición	El usuario debe tener ID y contraseña válida para poder acceder al sistema.
Pos condición	El usuario puede acceder al sistema.
Flujo normal	<ol style="list-style-type: none"> 1. Ingresar usuario. 2. Ingresar contraseña. 3. Validar datos.
Excepciones	No se puede ingresar sin usuario y contraseña
Anotaciones	Al cerrar sesión ya no se puede ingresar a ningún área del sistema.

Caso de uso: Reserva telefónica

Este caso de uso permite al usuario ingresar los datos del huésped al sistema para crear reserva (Figura 2.7.), el cual se encuentra especificado a continuación (Tabla 2.4.).



Figura 2.7. Diagrama de caso de uso reserva telefónica

Tabla 2.4. Especificación del caso de uso: Reserva telefónica	
Código	2
Actores	Usuario, huésped.
Pre condición	Ninguna.
Pos condición	Se espera el envío de la ficha, de lo contrario se cancela la reservación.
Flujo normal	<ol style="list-style-type: none"> 1. Solicitar clave huésped. 2. Datos de quién reserva. 3. Fecha de llegada 4. Fecha de salida 5. Hora de llegada 6. N° de adultos/N° de niños. 7. Ofrecer habitaciones. 8. Obtener el monto total hospedaje. 9. Dar porcentaje de reservación. 10. Huésped envía ficha de depósito. 11. Recepcionista confirma la reservación. 12. Deja el sistema en espera de otro movimiento.
Excepciones	Si el usuario no está registrado darlo de alta ingresando todos los datos personales.
Anotaciones	El descuento se hará a criterio del usuario. La reserva debe ser confirmada en un lapso no mayor de 48 horas mediante un depósito.

Caso de uso: Reserva internet

Este caso de uso permite al huésped solicitar una reserva por vía Web (Figura 2.8.), el cual se encuentra especificado a continuación (Tabla 2.5.).

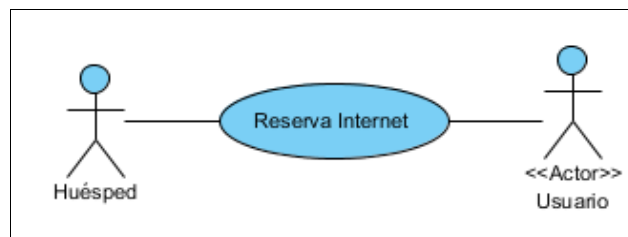


Figura 2.8. Diagrama de caso de uso reserva internet

Tabla 2.5. Especificación del caso de uso: Reserva internet	
Código	3
Actores	Huésped, Usuario.
Pre condición	El huésped debe ingresar a Internet y realizar el llenado de formulario.
Pos condición	Ninguna.
Flujo normal	<ol style="list-style-type: none"> 1. El huésped ingresa a Internet. 2. Navega en la página. 3. Llena datos de huésped. 4. Huésped envía formulario. 5. Sistema de correo envía un e_mail de confirmación de que envió correctamente el formulario. 6. Usuario verifica disponibilidad. 7. Usuario reenvía cargos por las habitaciones y porcentaje a pagar de la reservación y condiciones de pago. 8. Huésped envía ficha de depósito. 9. Usuario confirma la reservación. 10. Usuario realiza la reservación. 11. Deja el sistema en espera de otro movimiento.
Excepciones	Ninguna.
Anotaciones	Ninguna.

Caso de uso: Manejo de cuenta

Este caso de uso permite al usuario manejar la cuenta del huésped (Figura 2.9.), el cual se encuentra especificado a continuación (Tabla 2.6.).

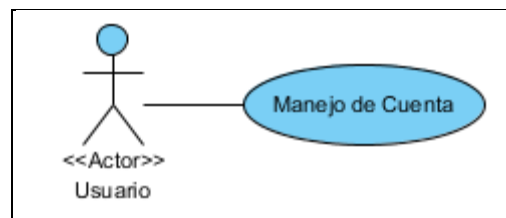


Figura 2.9. Diagrama de caso de uso manejo de cuenta

Tabla 2.6. Especificación del caso de uso: Manejo de cuenta	
Código	4
Actores	Usuario
Pre condición	El huésped debe de haber realizado check-in.
Pos condición	
Flujo normal	<ol style="list-style-type: none"> 1. Ingresar el número de habitación. 2. Verificar las habitaciones ocupadas por el huésped.

	3. Realizar una operación con la cuenta: cargo, abono o imprimir cuenta.
Excepciones	Ninguna
Anotaciones	Ninguna

Caso de uso: Alta huésped

Este caso de uso permite al usuario realizar el alta del huésped cuando este no exista en la base de datos (Figura 2.10.), el cual se encuentra especificado a continuación (Tabla 2.7.).

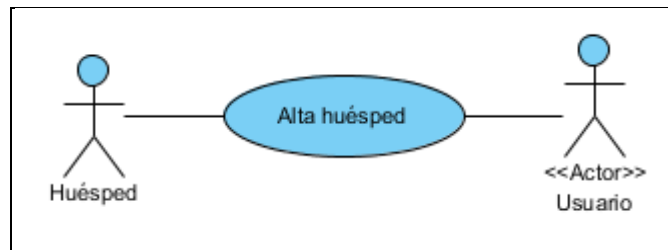


Figura 2.10. Diagrama de caso de uso alta huésped

Tabla 2.7. Especificación del caso de uso: Alta huésped	
Código	7
Actores	Usuario, Huésped.
Pre condición	El huésped no debe de existir en el sistema.
Pos condición	Ninguna.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario ingresa al sistema. 2. El usuario pide datos al huésped. 3. Se guardan los datos en el sistema. 4. Deja el sistema en espera de otro movimiento.
Excepciones	Ninguna.
Anotaciones	Ninguna.

Caso de uso: Check-in

Este caso de uso confirma la ocupación de la habitación por el huésped (Figura 2.11.), el cual se encuentra especificado a continuación (Tabla 2.8.).

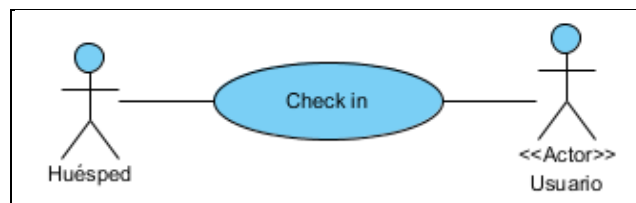


Figura 2.11. Diagrama de caso de uso check-in

Tabla 2.8. Especificación del caso de uso: Check-in	
Código	8
Actores	Huésped, Usuario.
Pre condición	Debe haber habitaciones disponibles
Pos condición	Ninguna
Flujo normal	<ol style="list-style-type: none"> 1. El huésped llega al hotel. 2. El usuario ingresa referencias huésped. 3. Verifica si existe reservación. 4. Verifica si la habitación continua disponible. 5. El sistema cambia la opción de la habitación de reservado por ocupado. 6. Deja el sistema en espera de otro movimiento.
Excepciones	Si confirmo otro huésped antes por la misma habitación, al huésped actual se le asignará otra habitación.
Anotaciones	Si el usuario no tiene una reservación, hará un check-in directo y se le puede ofrecer una habitación desocupada.

Caso de Uso: Cambiar habitación

Este caso de uso hace el cambio de la habitación del huésped a una nueva habitación desocupada (Figura 2.12.), el cual se encuentra especificado a continuación (Tabla 2.9.).

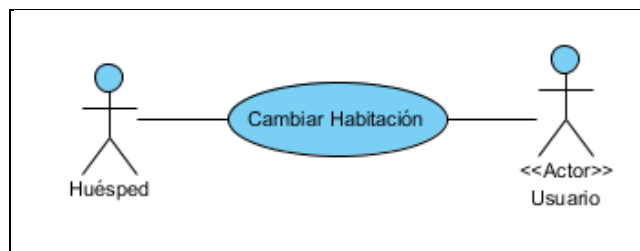


Figura 2.12. Diagrama de caso de uso cambiar habitación

Tabla 2.9. Especificación del caso de uso: Cambiar habitación	
Código	10
Actores	Huésped, Usuario.
Pre condición	La habitación a ocupar debe de estar desocupada.
Pos condición	Se libera la habitación antes ocupada.
Flujo normal	<ol style="list-style-type: none"> 1. El huésped o el usuario gestiona un cambio de habitación. 2. El usuario ingresa al sistema. 3. El usuario verifica el núm. de habitación. 4. .El usuario realiza el cambio en el sistema. 5. El huésped ocupa la nueva habitación. 6. Deja el sistema en espera de otro movimiento.
Excepciones	Ninguna.
Anotaciones	Al realizar el cambio a la nueva habitación, el sistema hace el cambio de la cuenta del huésped también.

Caso de uso: Cargar a la cuenta

Este caso permite al usuario agregar los servicios utilizados por el huésped a su cuenta a pagar.(Figura 2.13.), el cual se encuentra especificado a continuación (Tabla 2.10.).

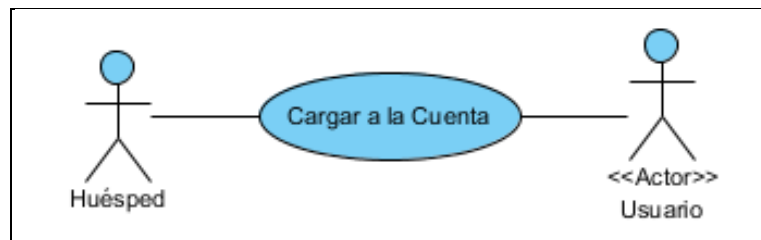


Figura 2.13. Diagrama de caso de uso cargar a la cuenta

Tabla 2.10. Especificación del caso de uso: Cargar a la cuenta	
Código	11
Actores	Usuario, huésped.
Pre condición	El huésped debe de haber realizado un check-in.
Pos condición	Ninguna.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario ingresa al sistema. 2. El huésped solicita servicios con cargo a su habitación. 3. El usuario busca la cuenta del cliente. 4. El usuario agrega los servicios solicitados por el huésped y estos cargos se suman a su cuenta. 5. Deja el sistema en espera de otro movimiento.

Excepciones	Ninguna.
Anotaciones	Ninguna.

Caso de uso: Abonar cuenta

Este caso de uso permite al huésped realizar un abono a la cuenta total (Figura 2.14.), el cual se encuentra especificado a continuación (Tabla 2.11.).

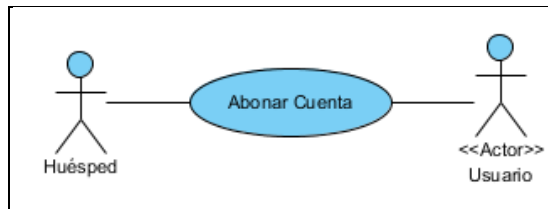


Figura 2.14. Diagrama de caso de uso abonar cuenta

Tabla 2.11. Especificación del caso de uso: Abonar cuenta	
Código	12
Actores	Huésped, Usuario.
Pre condición	El huésped debe de haber realizado un check-in.
Pos condición	Ninguna
Flujo normal	<ol style="list-style-type: none"> 1. El usuario ingresa al sistema. 2. El huésped informa de querer realizar el abono. 3. El usuario realiza la búsqueda del huésped. 4. Huésped entrega la cantidad a abonar. 5. El usuario ingresa el abono a la cuenta y se hace la resta en la cuenta total. 6. Deja el sistema en espera de otro movimiento.
Excepciones	Ninguna.
Anotaciones	Ninguna.

Caso de uso: Check-out

Este caso de uso libera la habitación, siempre y cuando sea pagada la habitación en su totalidad (Figura 2.15.), el cual se encuentra especificado a continuación (Tabla 2.12.).

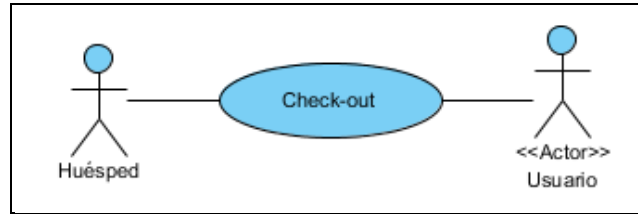


Figura 2.15. Diagrama de caso de uso Check-out

Tabla 2.12. Especificación del caso de uso: Check-out	
Código	13
Actores	Huésped, Usuario.
Precondición	La habitación debe de estar ocupada.
Pos condición	Se da de baja al cliente en el hotel.
Flujo normal	<ol style="list-style-type: none"> 1. El huésped informa salida del hotel. 2. El usuario verifica el núm. de habitación. 3. Se corrobora la cuenta del cliente. 4. Pagar cuenta. 5. Imprimir cuenta. 6. La habitación queda liberada. 7. Se borra la cuenta. 8. Deja el sistema en espera de otro movimiento.
Excepciones	No se libera habitación si no se cubre el costo total.
Anotaciones	La nota incluye costo de habitación y servicios utilizados.

2.2. Diseño

2.2.1. Modelo de datos

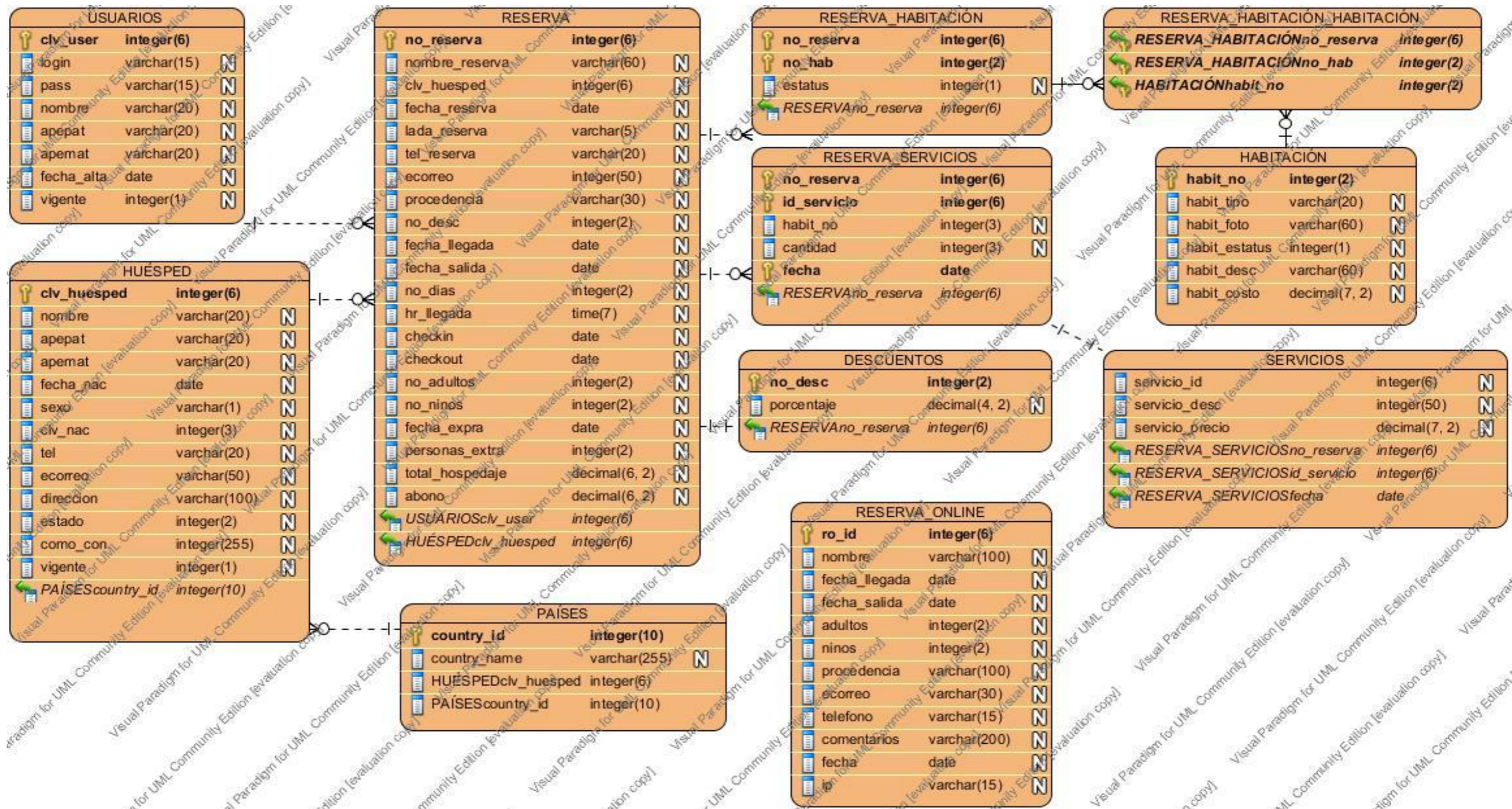


Figura 2.16. Diagrama entidad relación

Diccionario de datos

Por medio de este diccionario de datos se describirán los atributos que conforman cada una de las entidades utilizadas en el sistema Casa Matti

- **Descuentos**

Columna	Tipo	Nulo	Pred.	Comentarios
no_desc	int(2)	No		Id del descuento
porcentaje	decimal(4,2)	No		Porcentaje

- **Entidades**

Columna	Tipo	Nulo	Pred.	Comentarios
cve_ent	varchar(2)	Sí	NULL	Clave Entidad
nom_ent	varchar(45)	Sí	NULL	Nombre del estado
nom_abr	varchar(16)	Sí	NULL	Nombre abreviado del estado
cve_mun	varchar(3)	Sí	NULL	Clave del municipio
nom_mun	varchar(50)	Sí	NULL	Nombre del municipio
siglas_mun	varchar(4)	Sí	NULL	Siglas del municipio
nomun	varchar(27)	Sí	NULL	Municipio
cve_cab	varchar(9)	Sí	NULL	Clave de la cabecera municipal
nom_cab	varchar(50)	Sí	NULL	Nombre de la cabecera municipal

- **Habitación**

Columna	Tipo	Nulo	Pred.	Comentarios
habit_no	int(2)	No		Número de habitación
habit_tipo	varchar(20)	No		Tipo de habitación
habit_foto	varchar(60)	No		Foto habitación
habit_estatus	int(1)	No		Estatus de habitación
habit_desc	varchar(60)	No		Descuento de habitación
habit_costo	decimal(7,2)	No		Costo de habitación

- **Huésped**

Columna	Tipo	Nulo	Pred.	Comentarios
clv_huesped	int(6)	No		Clave Huésped
nombre	varchar(20)	No		Nombre
apepat	varchar(20)	No		Apellido paterno
apemat	varchar(20)	No		Apellido materno
fecha_nac	Date	No		Fecha de nacimiento
sexo	varchar(1)	No		Sexo (hombre/mujer)
clv_nac	int(3)	No		Clave Nacionalidad

tel	varchar(20)	No		Teléfono
ecorreo	varchar(50)	No		Correo electrónico
direccion	varchar(100)	No		Dirección
estado	int(2)	No		Estado
como_con	varchar(100)	No		Como nos conoció
vigente	int(1)	No	1	Estatus del usuario si sigue activo o no; 1= activo; 0=inactivo

- **Países**

Columna	Tipo	Nulo	Pred.	Comentarios
country_id	int(10)	No		Identificador del país
country_name	varchar(255)	No		Nombre del país

- **Reserva**

Columna	Tipo	Nulo	Pred.	Comentarios
no_reserva	int(6)	No		Número de la reserva
nombre_reserva	varchar(60)	No		Nombre de quién reserva
clv_huesped	int(6)	No		Clave huésped
fecha_reserva	Date	No		Fecha reserva
tel_reserva	varchar(20)	No		Teléfono reserva
lada_reserva	varchar(5)	No		Lada reserva
ecorreo	varchar(50)	No		Correo Electrónico
procedencia	varchar(30)	No		Procedencia del huésped
no_desc	int(2)	No		Id del descuento
fecha_llegada	Date	No		Fecha llegada
fecha_salida	Date	No		Fecha Salida
no_dias	int(2)	No		Número de días
hr_llegada	Time	No		Hora de llegada
checkin	Datetime	No		Horario en que realizó el check-in
checkout	Datetime	No		Horario en que se realizó el check-out
no_adultos	int(2)	No		Número de adultos
no_ninos	int(2)	No		Número de niños
fecha_expira	Date	No		Fecha_expiración de la reserva
personas_extra	int(2)	No		Personas extras
total_hospedaje	decimal(6,2)	No		Total a pagar del hospedaje
abono	decimal(6,2)	No		Cantidad que dio de anticipo y lo que se va abonando

- **Reserva_habitacion**

Columna	Tipo	Nulo	Pred.	Comentarios
no_reserva	int(6)	No		Número de reservación
no_habit	int(2)	No		Número de habitaciones
estatus	int(1)	No		Estatus de la habitación 0= libre; 1= reservada; 2=ocupada

- **Reserva_online**

Columna	Tipo	Nulo	Pred.	Comentarios
ro_id	int(6)	No		Identificador de reserva online
nombre	varchar(100)	No		Nombre del huésped
fecha_llegada	Date	No		Fecha de llegada
fecha_salida	Date	No		Fecha de salida
adultos	int(2)	No		Número de adultos
nicos	int(2)	No		Número de niños
procedencia	varchar(100)	No		Procedencia
ecorreo	varchar(30)	No		Correo electrónico
telefono	varchar(15)	No		Teléfono
comentarios	varchar(200)	No		Comentarios
fecha	Datetime	No		Fecha en que se realizo la solicitud
ip	varchar(15)	No		Reserva online guarda dirección IP de la computadora de la que hizo la reservación

- **Reserva_servicios**

Columna	Tipo	Nulo	Pred.	Comentarios
no_reserva	int(6)	No		Número de reservación
id_servicio	int(6)	No		Identificador de servicio
cantidad	int(3)	No		Cantidad de servicios utilizados
habit_no	int(2)	No		Número de habitación
fecha	datetime	No		Fecha en que realizó el servicio

- **Servicios**

Columna	Tipo	Nulo	Pred.	Comentarios
servicio_id	int(6)	No		Identificador de servicio
servicio_desc	varchar(50)	No		Descripción del servicio
servicio_precio	decimal(7,2)	No		Costo del servicio

- **Usuarios**

Columna	Tipo	Nulo	Pred.	Comentarios
clv_usr	int(6)	No		Clave o id usuario
login	varchar(15)	No		Login de entrada
pass	varchar(15)	No		Password
nombre	varchar(20)	No		Nombre
apepat	varchar(20)	No		Apellido paterno
apemat	varchar(20)	No		Apellido materno
fecha_alta	datetime	No		Fecha de alta en el sistema
vigente	int(1)	No		0= usuario inactivo; 1=Usuario activo

2.2.2. Arquitectónico

A continuación se describe la jerarquización del sistema de reservas de la página Web de la Casa Matti. La página Web cuenta con un menú desde donde se podrá ingresar a las diferentes áreas con las que cuenta (Figura 2.17.).

- **Inicio:** Es el sitio principal también se le llama home page. Es la presentación de la Casa de huéspedes Matti.
- **Quienes Somos:** En este apartado se describe los inicios de la casa de huéspedes Matti, además de existir un formulario en la cual los posibles huéspedes pidan información relativa a la Casa de huéspedes Matti. También se muestran la dirección y los teléfonos en caso de requerir una reserva telefónica.
- **Habitaciones:** Es una galería de todas las habitaciones, en donde el huésped puede observar como se encuentran físicamente las instalaciones.
- **Reservas:** El huésped podrá realizar el envío del formulario con los datos requeridos, esto con el fin de que el usuario verifique si existe habitaciones disponibles en la fecha escogida, además de que podrá visualizar el calendario de temporadas alta y bajas.
- **Galería:** Se realizó un carrusel interactivo con fotos que fueron tomadas por huéspedes anteriores durante su estancia en la Casa de huéspedes Matti, en el cual el posible huésped, podrá escoger cual quiere visualizar.

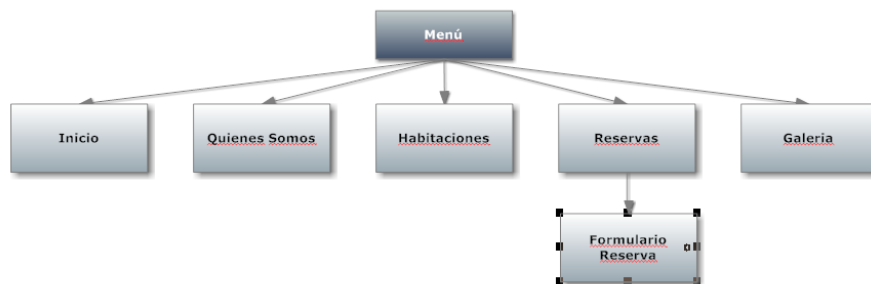


Figura 2.17. Diseño arquitectónico de la Web

Con referencia al sistema, se tiene como primer área el inicio en el cual el usuario se ingresará su usuario y contraseña, logrando acceder a su sesión. A continuación se divide el sistema en tres ramificaciones que son: Catálogo, Operativo y Reportes. Los cuales se describen a continuación (Figura 2.18.).

1. Catálogo: Esta sección cuenta con cuatro módulos:

- **Huéspedes:** Es donde se guardan los datos de los clientes que han reservado una habitación en la Casa Matti, los datos se describen en el diccionario de datos que con anterioridad se expuso.
- **Habitaciones:** En este catálogo se guardan los datos de las habitaciones de la casa de huéspedes, para poder seleccionar la correcta que un huésped necesita al momento de reservar.
- **Usuario:** Aquí se guardará el dato de Usuario y Contraseña del usuario del sistema. Este se verificará si existe para así poder iniciar sesión.
- **Servicios:** Es una lista de servicios con los que puede contar el huésped para su consumo personal.

2. Operativo: Está apartado cuenta con seis módulos que son:

- **Reservaciones:** En este módulo se guardarán los datos de la reserva (descritos en el diccionario de datos), además se podrá asignar la habitación, ver el costo total del hospedaje, el descuento y el porcentaje a pagar para que la reservación sea habilitada.
- **Check-in:** Se realizará cuando el huésped llegue al hotel, ya sea con previa reservación o directa, avisando al sistema que el huésped ya está utilizando su habitación y creándole una cuenta.
- **Check-out:** Este módulo se utilizará cuando el huésped decida dejar la habitación, pagando previamente el total de su adeudo, y dejando el estatus de la habitación como libre.
- **Abonos:** El huésped podrá hacer pagos adelantados a su cuenta, además de finiquitar su adeudo por completo, si así lo decidiera.
- **Servicios:** Se utilizará cuando el huésped requiera algún servicio, haciéndose el cargo del servicio en su cuenta.
- **Cambiar Habitación:** Si el huésped decidiera que la habitación asignada no es de su agrado o bien el usuario decidiera cambiar de habitación al huésped debido a que hubiera desperfectos a reparar en la habitación, se podrá cambiar de habitación al huésped trasladando también su cuenta a la nueva habitación.

3. Reportes: Esta división cuenta con el modulo de Nota, que es donde se reflejaran detalles de su cuenta, se dividirá en gastos de hospedaje y de servicios, así dando el monto total, con sus respectivos abonos anteriores.

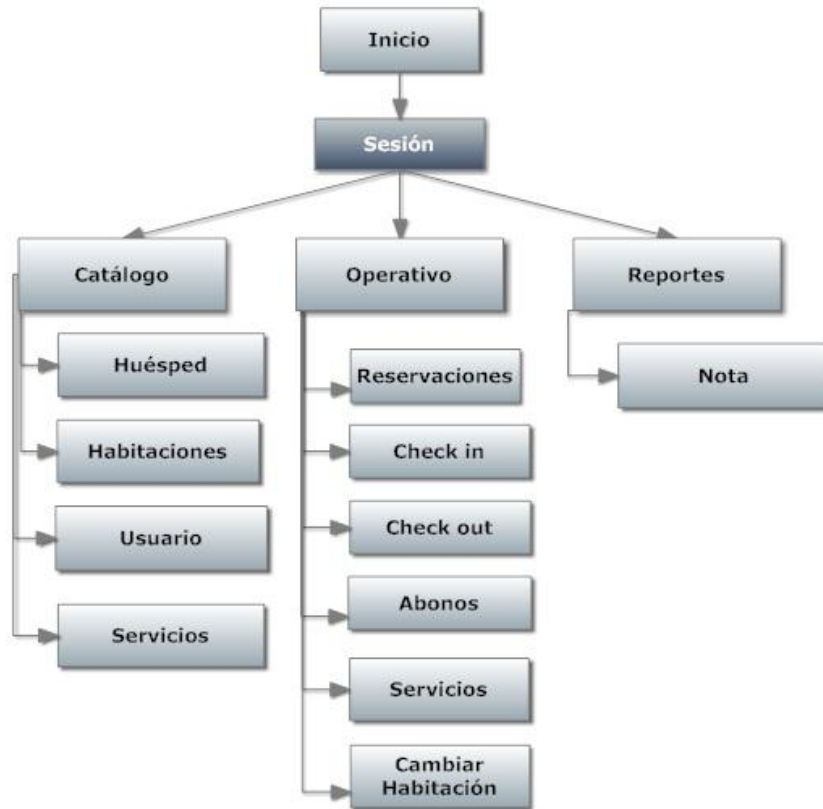


Figura 2.18. Diseño arquitectónico del sistema

2.2.3. Diseño de la interfaz de usuario

Es la categoría de diseño que creará una interfaz con las especificaciones del cliente, se le sugiere el prototipo de recepción, reservación y check-in.

Por medio de esta interfaz el usuario podrá llevar a cabo la administración de la casa de huéspedes, mediante la recepción, donde contará con accesos directos para todo el sistema (Figura 2.19.).

En reservación podrá llevar a cabo las reservaciones, donde consultará si existe el huésped o será de alta si es nuevo, así mismo ingresará los datos necesarios del huésped (Figura 2.20.).

La interfaz de Check-in, se utilizará para dar de alta el ingreso del huésped a la casa. En este momento el huésped contará con una cuenta donde se registrarán sus gastos y abonos que realice en el transcurso de su estancia (Figura 2.21.).

Recepción

RECEPCION

REFERENCIAS DEL HUESPED

NO. HABITACION

NOMBRE DEL HUESPED

TIPO HABITACION

CLAVE HUESPED

CHECK-IN

FECHA SALIDA

ADULTOS

NIÑOS

HABITACIONES OCUPADAS

CLAVE	FECHA	CANTIDAD	CONCEPTO	CARGO

HOSPEDAJE

SERVICIOS

DESCUENTO %

ABONO

SALDO TOTAL

Figura 2.19. Interfaz prototipo recepción

Reservación

RESERVACION

NOM. HUESPED CLAVE HUESPED NUM. RESERVA

NOMBRE DE QUIEN RESERVA TELEFONO ADULTOS FECHA LLEGADA FECHA SALIDA

E MAIL PROCEDENCIA NIÑOS NUM. DIAS HORA LLEGADA

NUM. HAB.	TIPO HAB.	DESCRIPCION	SUBTOTAL	PERSONA EXT A N	TOTAL HAB.

SUBTOTAL HOSPEDAJE

DESCUENTO

TOTAL HOSPEDAJE

% RESERVA

TOTAL RESERVA

Figura 2.20. Interfaz prototipo reservación

Figura2.21. Interfaz protocolo check-in

2.2.3. Procedimental

Reserva

El diseño de componentes o procedimental que se especifica a continuación representa la guía para el software del sistema de reservación (Figura 2.22.).

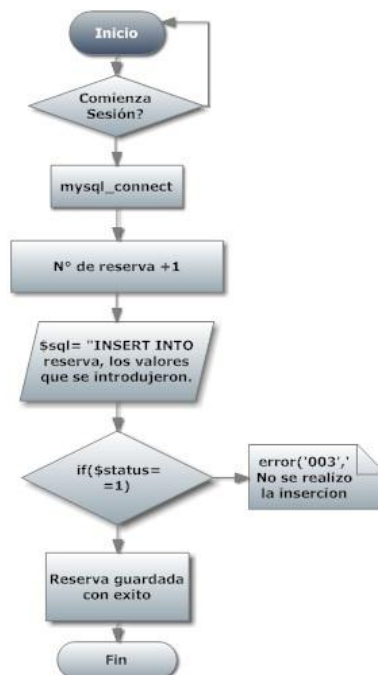


Figura 2.22. Diseño procedimental reserva

Check-in

En esta parte se diseña el componente guía para el software de check-in. Donde se verifica si existe una reservación, de lo contrario se regresa al inicio, si existe desplegará los datos de la reserva y se procede a hacer el check-in. Si éste es exitoso se mostrará un mensaje (Figura 2.23.).



Figura 2.23. Diseño procedimental check-in

Servicios

Este diseño procedimental constituye la parte del modulo de servicios, donde especifica la guía para realizar este software (Figura 2.24.).

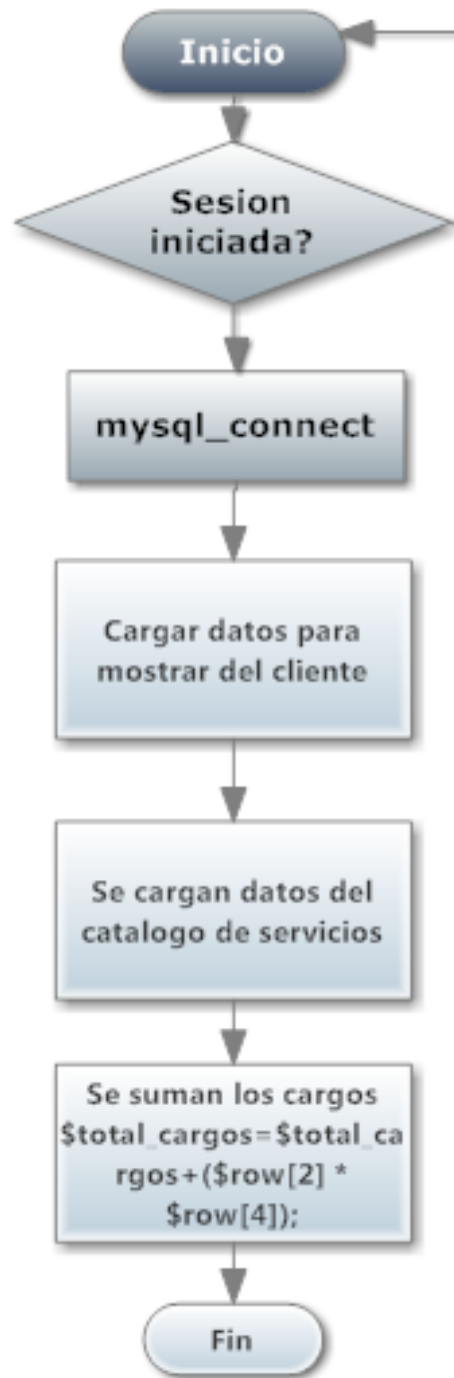


Figura 2.24. Diseño procedimental servicios

Cambiar habitación

En el diseño del modulo Cambiar habitación se guía a través de este módulo procedimental. Describiendo el cambio de habitación, por medio de la liberación de la habitación anterior y el cambio de estado de la nueva habitación (Figura 2.25.).

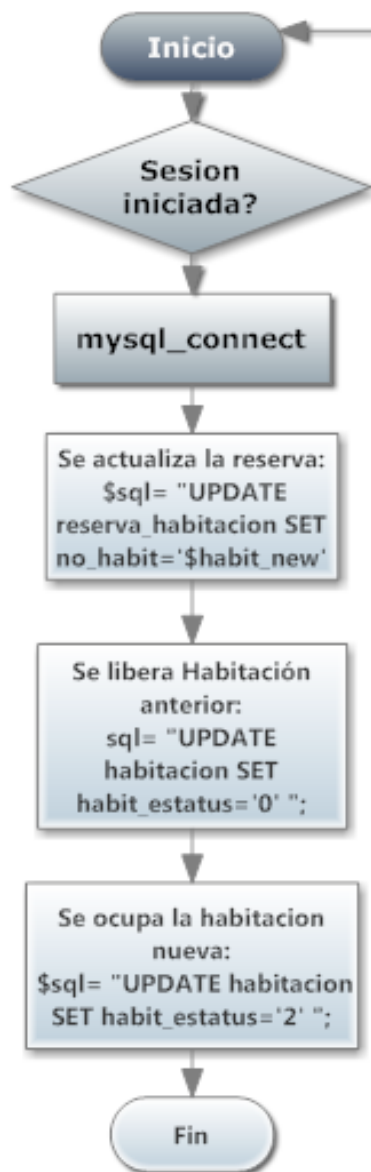


Figura 2.25. Diseño procedimental cambiar habitación

Check-out

El diseño de componentes de este bloque, guía el proceso del check-out, se verificará primero si el total es mayor a 0, se procederá a abrir el subproceso de abono, para realizar el pago total de la cuenta, ya hecho esto se refrescará y se continuará con el check-out., enseguida anunciará que este fue exitoso, y procederá a imprimir la nota (Figura 2.25.).

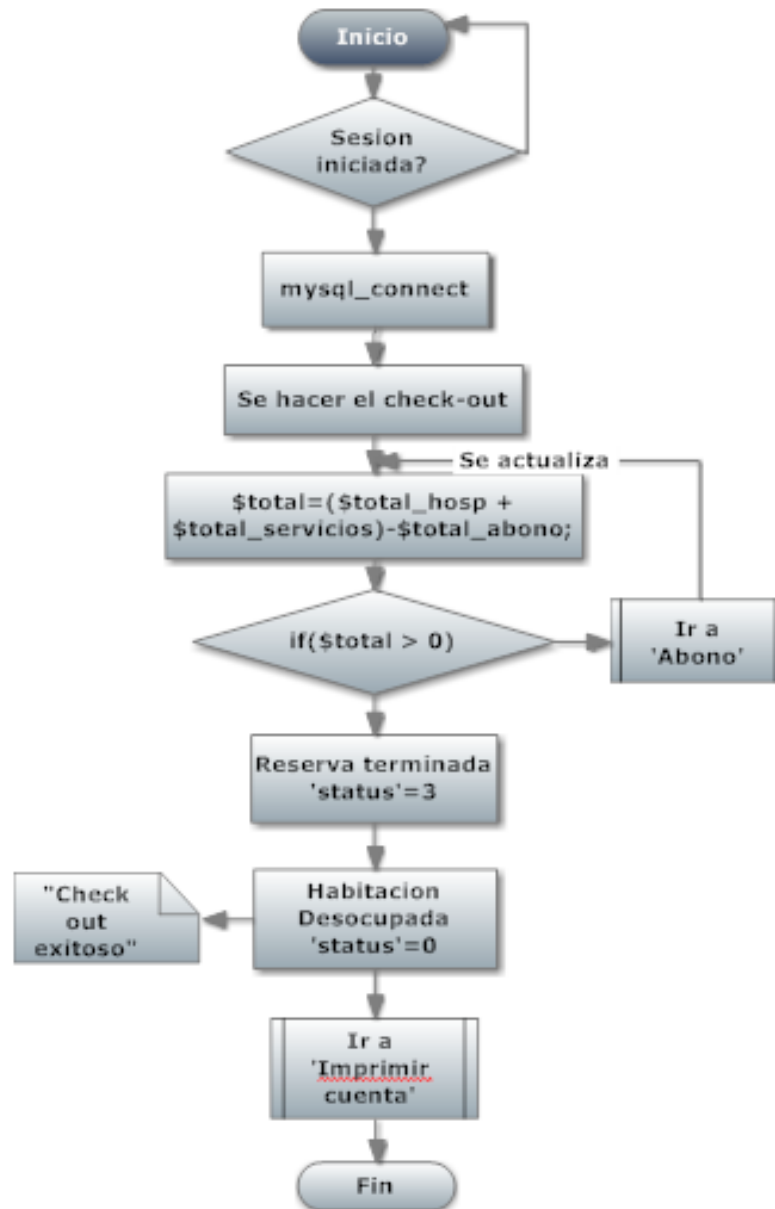


Figura 2.26. Diseño procedimental check-out

CAPÍTULO 3. DESARROLLO E IMPLEMENTACIÓN

A continuación se describe los procesos más importantes:

- Reservas
- Check-in
- Check-out
- Cambiar Habitación
- Servicios

Se incluyen fragmentos de operación en PHP y MySQL.

3.1. Reservas

En el módulo de reservas los procesos más importantes son la consulta para verificar si existe el huésped y el guardado de los datos en la Base (Figura 3.1.). El fragmento del código del sistema se muestra a continuación.

Num. hab	Tipo Hab.	Descripción	Subtotal	Pera. Ext.		Total hab.
				A	N	
<input checked="" type="checkbox"/>	1	Cuarto 1 king size, 1 Ind, Baño	\$1,200.00	<input type="checkbox"/>	<input type="checkbox"/>	3600
<input type="checkbox"/>	2	Cuarto 1 king size, 1 Ind, Baño	\$1,200.00	<input type="checkbox"/>	<input type="checkbox"/>	0
<input type="checkbox"/>	3	Cuarto 1 king size, 1 Ind, Baño	\$1,200.00	<input type="checkbox"/>	<input type="checkbox"/>	0
<input type="checkbox"/>	4	Departamento 2 Hab, 2 Cama King Size, 1 Ind	\$2,100.00	<input type="checkbox"/>	<input type="checkbox"/>	0
<input type="checkbox"/>	5	Bungalow Cocina, 1 King Size, 2 ind	\$1,500.00	<input type="checkbox"/>	<input type="checkbox"/>	0

Figura 3.1. Reservas

Consulta Si existe el huésped:

```
$query = "SELECT clv_huesped, nombre, apepat, apemat
FROM huesped
WHERE nombre LIKE '$data%'
LIMIT 5";
```


Si existe regresan los siguientes datos del huésped, sino "No hay coincidencias":

```
while ($row = mysql_fetch_array($result))
{
    $toReturn = $row['nombre'] . " " .
    $row['apepat'] . " " . $row['apemat'];
    $dataList[] = '<li id="' . $row['clv_huesped'] .
    '"><a href="#">'. htmlentities($toReturn) . '</a></li>';
}
```

Inserta datos a la base:

```
$sql= "INSERT INTO reserva
(`no_reserva`,`nombre_reserva`,`clv_huesped`,`fecha_reserva`,`
lada_reserva`,`tel_reserva`,`correo`,`procedencia`,`no_desc
`,`fecha_llegada`,`fecha_salida`,`hr_llegada`,`no_dias`,`no_a
dultos`,`no_ninos`,`total_hospedaje`,`abono`)" ;

$sql.="VALUES('$num_reser','$name_reserva','$huespedid','$fec
ha','$lada','$stel','$email','$procedencia','$desc','$llegada'
','$salida','$hr_llegada','$num_dias','$adultos','$ninos','$tot
al','0.00')";
```

Se llenan los campos y se guarda, devuelve el mensaje:

```
if(isset($_GET['op']))
{
    $status=$_GET['op'];
    $r=$_GET['reservacion'];
    if($status==1)
        $mensaje="<script>alert('su reservación quedo guardada
con el numero $r');</script>";
}
```

Si la inserción no tuvo éxito:

```
$filas=mysql_query($sql);
echo $sql;
if(!$filas)
    error('003','No se realizo la inserción -
reservaciones_save.php (43)' . mysql_error());
```

3.2. Check-in

Los procesos más importantes en el Check-in (Figura3.2.), será cuando el huésped se registre para poder ingresar a su habitación y el sistema cambie el status de la habitación de reservada a ocupada.

A continuación se muestra parte del código del sistema.

Logo: CASA MÓVIL

root - cerrar sesión

Check - in

Directa

Editar check-in

Guardar

N° de Reserva: 4 [buscar]

Nombre del huésped: OCTAVIO RAMIREZ TORRES Telefono: 744 4854159

Email: RATO9598@HOTMAIL.COM Lugar de Procedencia: ACAPULCO

Fecha Reservación

Fecha llegada: 2012-05-12

Fecha salida: 2012-05-14

N° de Días: 3

Hora de llegada: 11:00:00

Check-in: 2012-05-18

N° de Adultos: 2

N° de Niños: 1

N° Hab	Tipo Habitación	Descripción	P. Ex.	A o N	Total Habitación
4	tipo	desc	<input type="checkbox"/>	<input type="checkbox"/>	3600.00

SubTotal Hospedaje: 3600

Descuento %3: 108

Abono: 2047.60

Total: 2444.40

Sistema de Control Casa Móvil v.1.0

Cerrar

Figura 3.2. Check-in

Se realiza Check In:

```
if(isset($_GET['checkin']))
{
    $reserva_checkin=$_GET['checkin'];
    $fecha_checkin=date('Y-m-d H:i:s');
```

Cambia estado de habitación de Reservada a Ocupada:

```
$sql="SELECT no_habit FROM reserva_habitacion WHERE
no_reserva='$reserva_checkin'";
$filas=mysql_query($sql);
while($row = mysql_fetch_row($filas))
{
    $n=$row[0];
    //actualizar el estado de las habitaciones reservadas a
    ocupadas
    $sql2= "UPDATE habitacion SET habit_estatus='2' ";
    $sql2.="WHERE habit_no='$n'";
    $filas2=mysql_query($sql2);
}
```

Despliega Mensaje "CHECKIN exitoso"

```
"<script>alert('CHECKIN exitoso!!!');</script>";
```

3.3. Servicios

A continuación se describirá el proceso más significativo que es el agregar servicios al manejo de cuenta del huésped (Figura 3.3.).

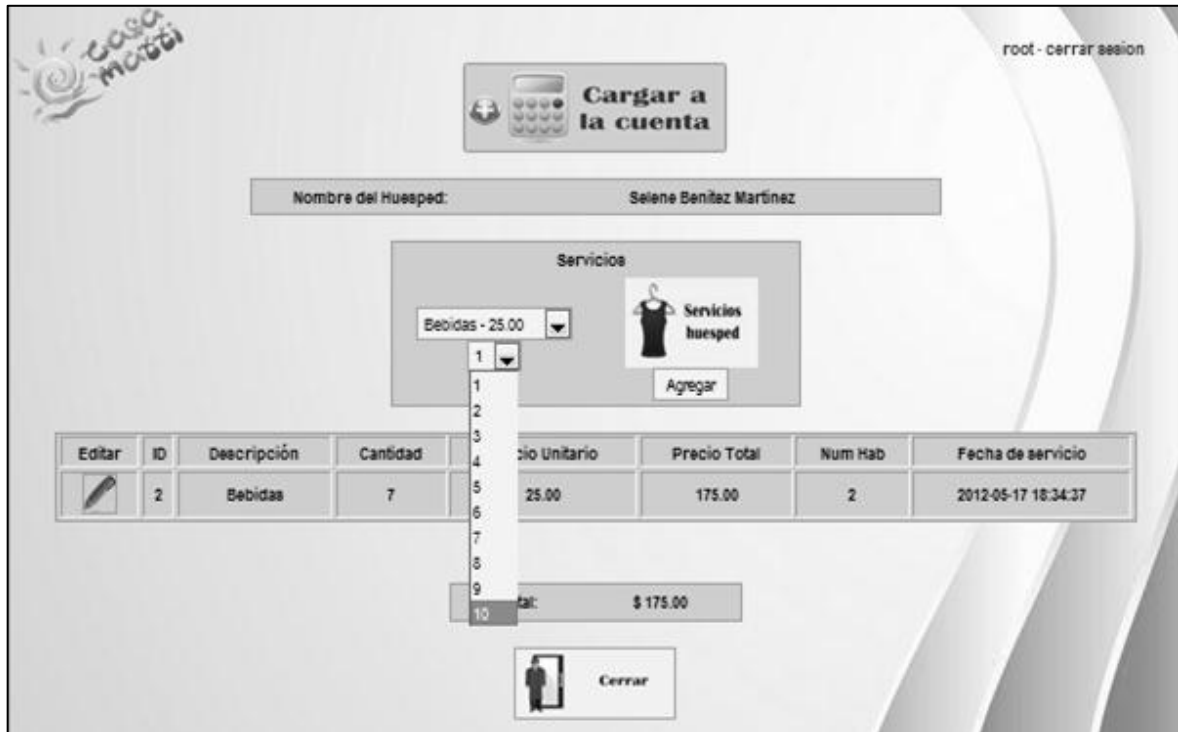


Figura 3.3. Servicios

Agregar servicios: Cervezas, Playa, Lavandería.

```
if(isset($_POST['servicios_add']))
{
    $fecha_actual=date('Y-m-d H:i:s');
    $servicios_add=$_POST['servicios_add'];
    $servicios_cant=$_POST['servicios_cant'];
    $sql= "INSERT INTO reserva_servicios
(`no_reserva`,`id_servicio`,`cantidad`,`habit_no`,`fecha`)" ;
    $sql.="values
('$reservacion','$servicios_add','$servicios_cant','$habitacion','$fecha_actual)";
    //echo $sql;
    $filas=mysql_query($sql);
}
```

3.4. Cambiar habitación

Para el módulo de cambio de habitación, lo más relevante del proceso es cuando el sistema cambia al huésped de una habitación a otra, trasladando todo el manejo de su cuenta a la nueva habitación, y cambiando el estatus de la habitación anterior dejándola habilitada para una próxima ocupación (Figura 3.4.).



Figura 3.4. Cambiar de habitación

Se cambia la habitación:

```
if(isset($_POST['habit_new']))
{
    //cambiar en la reservacion
    $habit_new=$_POST['habit_new'];

    $sql= "UPDATE reserva_habitacion SET no_habit='$habit_new'
";
    $sql.="WHERE no_habit='$habitacion'";
    $filas=mysql_query($sql);
```

Se libera la habitación

```
$sql= "UPDATE habitacion SET habit_estatus='0' ";
$sql.="WHERE habit_no='$habitacion'";
$filas=mysql_query($sql);

//bloquear la nueva habitacion
$sql= "UPDATE habitacion SET habit_estatus='2' ";
$sql.="WHERE habit_no='$habit_new'";
$filas=mysql_query($sql);
```

```
$habitacion=$habit_new;
```

3.5. Check-out

En referencia al Check-out el proceso más importante es verificar que si la habitación quedará libre, el huésped deberá de saldar su adeudo correspondiente, al terminar esta condición la habitación quedará liberada (Figura 3.5.).

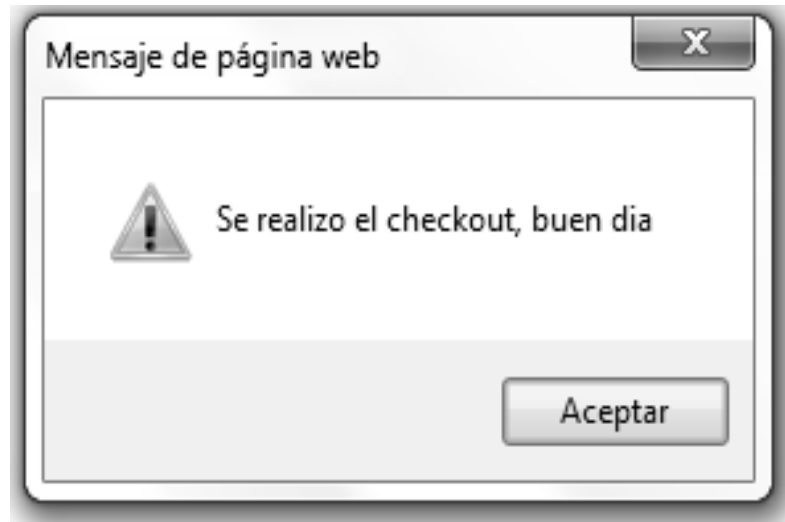


Figura 3.5. Check-out

Si no hay adeudo:

```
if($total > 0)
```

Se libera la habitación, se cambia el estado de la reserva a terminada:

```
$sql= "UPDATE reserva SET  

checkout='$fecha_checkout',estatus='3' ";  

$sql.="WHERE no_reserva='$reserva_checkout'";  

$filas=mysql_query($sql);
```

Actualiza el estado de las habitaciones ocupadas a libres

```
$sql= "UPDATE reserva_habitacion SET estatus='0' ";  

$sql.="WHERE no_reserva='$reserva_checkout'";  

$filas=mysql_query($sql);
```

CAPÍTULO 4

RESULTADOS Y CONCLUSIONES

CAPÍTULO 4. RESULTADOS Y CONCLUSIONES

4.1. Pruebas

A continuación se muestran de forma detallada y gráfica, los resultados que arrojaron las pruebas realizadas al sistema.

Descripción de la prueba 1: Iniciar sesión

1. Se introdujeron los siguientes datos en la pantalla Iniciar sesión (Figura 4.1.):
 - Usuario
 - Contraseña
2. Se verificarán que estos datos se encuentren registrados en la base de datos.

Figura 4.1. Iniciar sesión

Condiciones de ejecución	El usuario debe estar dado de alta en el sistema.
Resultado esperado	Con estos datos introducidos se espera que el usuario inicie sesión.
Evaluación de la prueba	La prueba se realizó con éxito (Figura 4.2.).

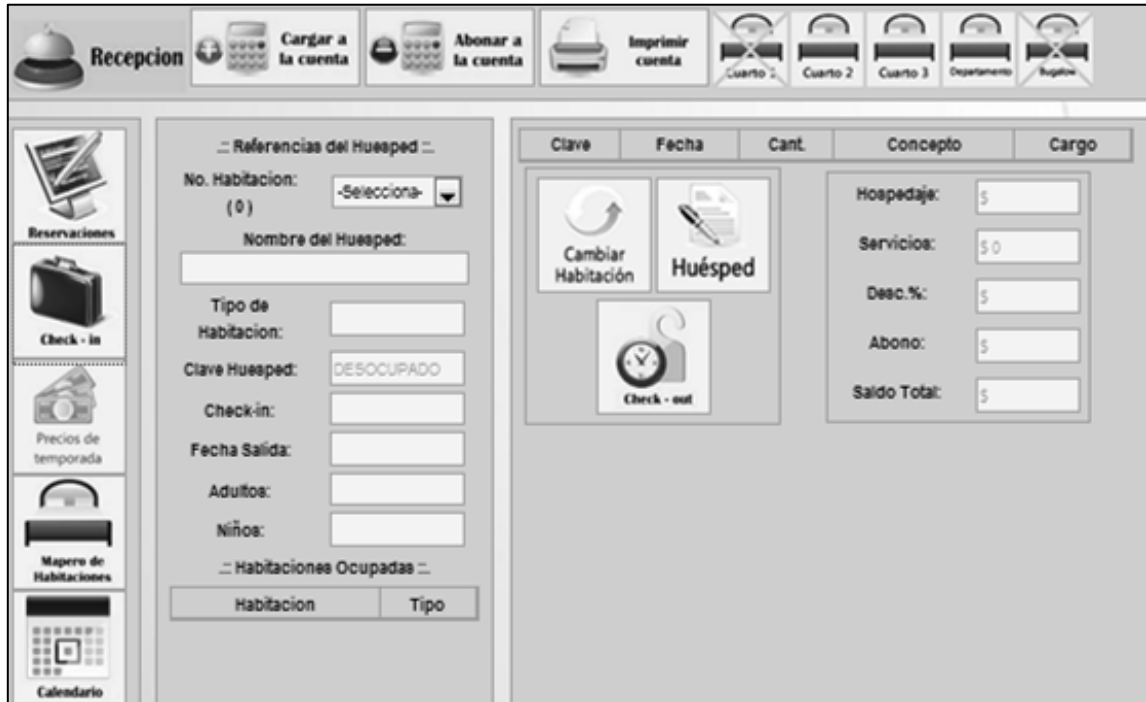


Figura 4.2. Recepción

4.2 Descripción de la prueba 2: Referencias huésped

1. Se seleccionará un número de habitación en el menú desplegable de la pantalla recepción (Figura 4.3.):
 - Habitación 1
 - Habitación 2
 - Habitación 3
 - Bungalow
 - Departamento

2. Se verificará que habitación se encuentra ocupada o desocupada por un huésped que previamente realizo check-in.



Figura 4.3. Referencias del huésped

Condiciones de ejecución	Los datos que obtendremos se han introducido con anterioridad y son ficticios.
Resultado esperado	<p>Cuando una habitación no cuente con un check-in arrojará la leyenda Desocupado en el campo Huésped.</p> <p>Cuando una habitación cuente con un check-in arrojará los datos del huésped y su cuenta hasta el último cargo.</p>
Evaluación de la prueba	La prueba se realizó con éxito (Figura 4.4.).

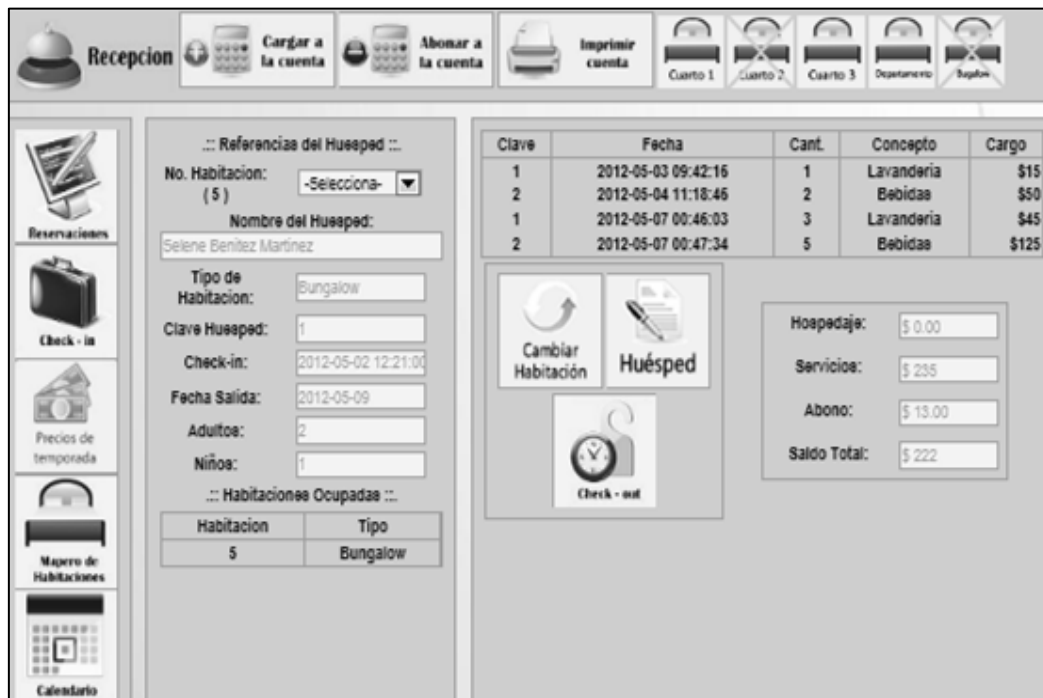


Figura 4.4. Datos del huésped

4.3 Descripción de la prueba 3: Cargar a la cuenta

1. Se seleccionará en la pantalla de Recepción la opción Cargar a la Cuenta
2. Se ingresará a la pantalla Cargar a la Cuenta.
3. Se seleccionará de esta última el servicio que desee cargar a su cuenta:
 - Lavandería
 - Bebidas
4. Se seleccionará la cantidad deseada (Figura 4.5.).
 - 1,2,3,4,5,6,7,8,9,10

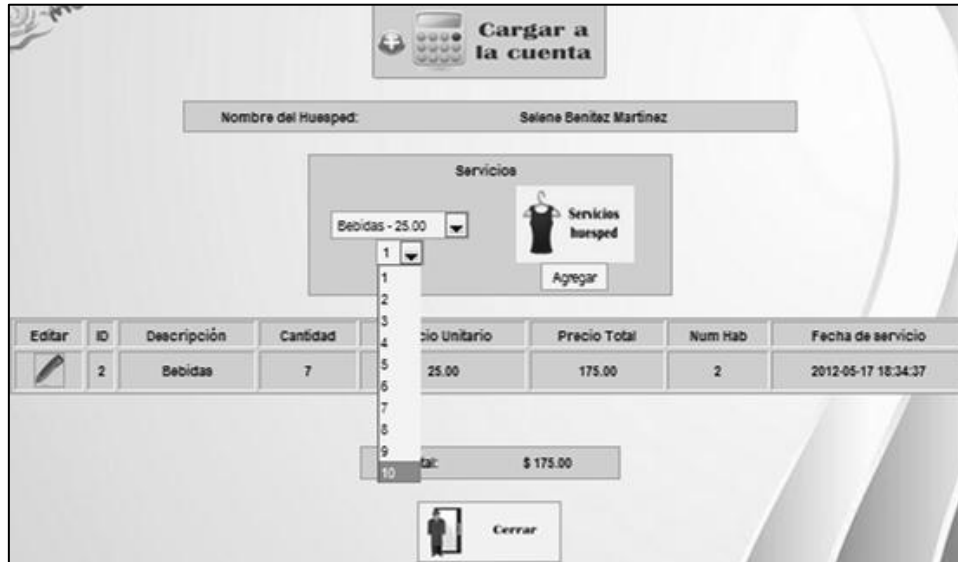


Figura 4.5. Seleccionar servicios

Condiciones de ejecución	<ul style="list-style-type: none"> • Se debe tener previamente seleccionado la habitación que ocupa el huésped desde Referencias del Huésped. • El usuario debe de estar ocupando una o varias habitaciones.
Resultado esperado	<ul style="list-style-type: none"> • Al seleccionar un servicio se agregara a la cuenta. • Se sumara el total del costo de servicios.
Evaluación de la prueba	La prueba se realizó con éxito (Figura 4.6.)



Figura 4.6. Cargar a la cuenta

4.4. Descripción de la prueba 4: Abonar a la cuenta

1. Se seleccionará en la pantalla de Recepción la opción Abonar a la Cuenta.
2. Se ingresará a la pantalla Abonar a la Cuenta (Figura 4.7.).
3. Se seleccionará de esta última el monto que desee abonar a su cuenta.

Figura 4.7. Abonar a la cuenta

Condiciones de ejecución	<ul style="list-style-type: none"> • Se debe tener previamente seleccionado la habitación que ocupa el huésped desde Referencias del Huésped. • El huésped debe contar con una cuenta previa con servicios cargados y monto de su hospedaje. • Los datos arrojados se ingresaron previamente y son ficticios.
Resultado esperado	El abono realizado por el huésped se debe restar al monto total de su cuenta.
Evaluación de la prueba	La prueba se realizó con éxito (Figura 4.8.).

Abono cuenta

No. Reservación: 1

Nombre del Huesped: Selene Benitez Martinez

Hospedaje: \$ 0.00 Abonar: 0.00

Servicio: \$ 505 Aceptar

Desc. %:

Saldo Total: \$

Abonado: \$ 478.00

Monto a Cubrir: \$ 27

Figura 4.8. Abono realizado

4.5. Descripción de la prueba 5: Imprimir cuenta

1. Se seleccionará de la pantalla Recepción el botón Imprimir Cuenta (Figura 4.9.).
2. Se desplegará la nota del cliente.

Recepcion

Cargar a la cuenta Abonar a la cuenta **Imprimir cuenta** Cuarto 1 Cuarto 2 Cuarto 3 Quotamen Reglas

Reservaciones

Check - in

Precios de temporada

Mapera de Habitaciones

Calendario

Referencias del Huesped

No. Habitación: (5) -Selecciona-

Nombre del Huesped: Selene Benitez Martinez

Tipo de Habitación: Bungalow

Clave Huesped: 1

Check-in: 2012-05-02 12:21:0

Fecha Salida: 2012-05-09

Adultos: 2

Niños: 1

Habitaciones Ocupadas

Habitacion	Tipo
5	Bungalow

Clave	Fecha	Cant.	Concepto	Cargo
1	2012-05-03 09:42:16	1	Lavanderia	\$15
2	2012-05-04 11:18:45	2	Bebidas	\$50
1	2012-05-07 00:46:03	3	Lavanderia	\$45
2	2012-05-07 00:47:34	5	Bebidas	\$125
1	2012-05-17 20:09:55	10	Lavanderia	\$150
1	2012-05-17 20:09:59	8	Lavanderia	\$120

Cambiar Habitación Huésped

Check - out

Hospedaje: \$ 0.00

Servicio: \$ 505

Abono: \$ 413.00

Saldo Total: \$ 92

Figura 4.9. Imprimir cuenta

Condiciones de ejecución	<ul style="list-style-type: none"> • Se debe tener previamente seleccionado la habitación que ocupa el huésped desde Referencias del Huésped que desea obtener Nota. • El huésped debe contar con una cuenta previa.
Resultado esperado	Se generará una nota de huésped para su impresión.
Evaluación de la prueba	La prueba se realizó con éxito (Figura 4.10.).

Hospedaje		
	Bungalow	\$0
Servicios		
Concepto	Precio Unit.	Subtotal
1	Lavanderia	\$15
2	Bebidas	\$50
3	Lavanderia	\$45
5	Bebidas	\$125
10	Lavanderia	\$150
8	Lavanderia	\$120
	Subtotal	\$505
	Total	\$27

© 2012 Casa Matti

Figura 4.10. Imprimir nota

4.6. Descripción de la prueba 6: Reservación

1. Seleccionará en la pantalla de Recepción el botón Reservas.
2. Se desplegará la pantalla Reservas (Figura 4.11.).

Reservaciones

Huésped: Número de Reserva: Fecha:

Nombre del Huésped: Teléfono: Adultos: Fecha Llegada: Fecha de Salida:

Email: Procedencia: Niños: Num. días: Hora de Llegada:

	Num. hab	Tipo Hab.	Descripción	Subtotal	Pera. Ext.		Total hab.
					A	N	
<input checked="" type="checkbox"/>	1	Cuarto	1 king size, 1 Ind, Baño	\$1,200.00	<input type="checkbox"/>	<input type="checkbox"/>	3600
<input type="checkbox"/>	2	Cuarto	1 king size, 1 Ind, Baño	\$1,200.00	<input type="checkbox"/>	<input type="checkbox"/>	0
<input type="checkbox"/>	3	Cuarto	1 king size, 1 Ind, Baño	\$1,200.00	<input type="checkbox"/>	<input type="checkbox"/>	0
<input type="checkbox"/>	4	Departamento	2 Hab, 2 Cama King Size, 1 Ind	\$2,100.00	<input type="checkbox"/>	<input type="checkbox"/>	0
<input type="checkbox"/>	5	Bungalow	Cocina, 1 King Size, 2 ind	\$1,500.00	<input type="checkbox"/>	<input type="checkbox"/>	0

Subtotal hospedaje:

Descuento:

Total Hospedaje:

30% Reserva:

Total Reserva:

Figura 4.11. Reservaciones

Condiciones de ejecución	El huésped deberá mandar escaneada la ficha de su depósito del 30% del costo total de la habitación.
Resultado esperado	Se generará la reservación.
Evaluación de la prueba	La prueba se realizó con éxito (Figura 4.12.).

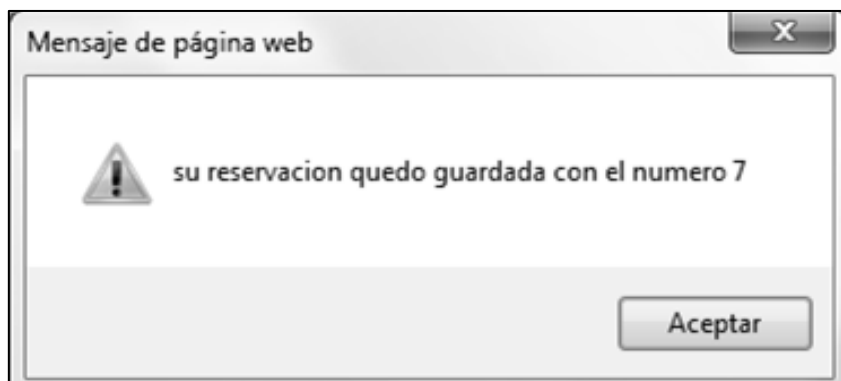


Figura 4.12. Mensaje de reservación realizada

4.7. Descripción de la prueba 7: Check-in

1. Seleccionará en la pantalla de Recepción el botón Check-In.
2. Se desplegará la pantalla Check-in (Figura 4.13.).

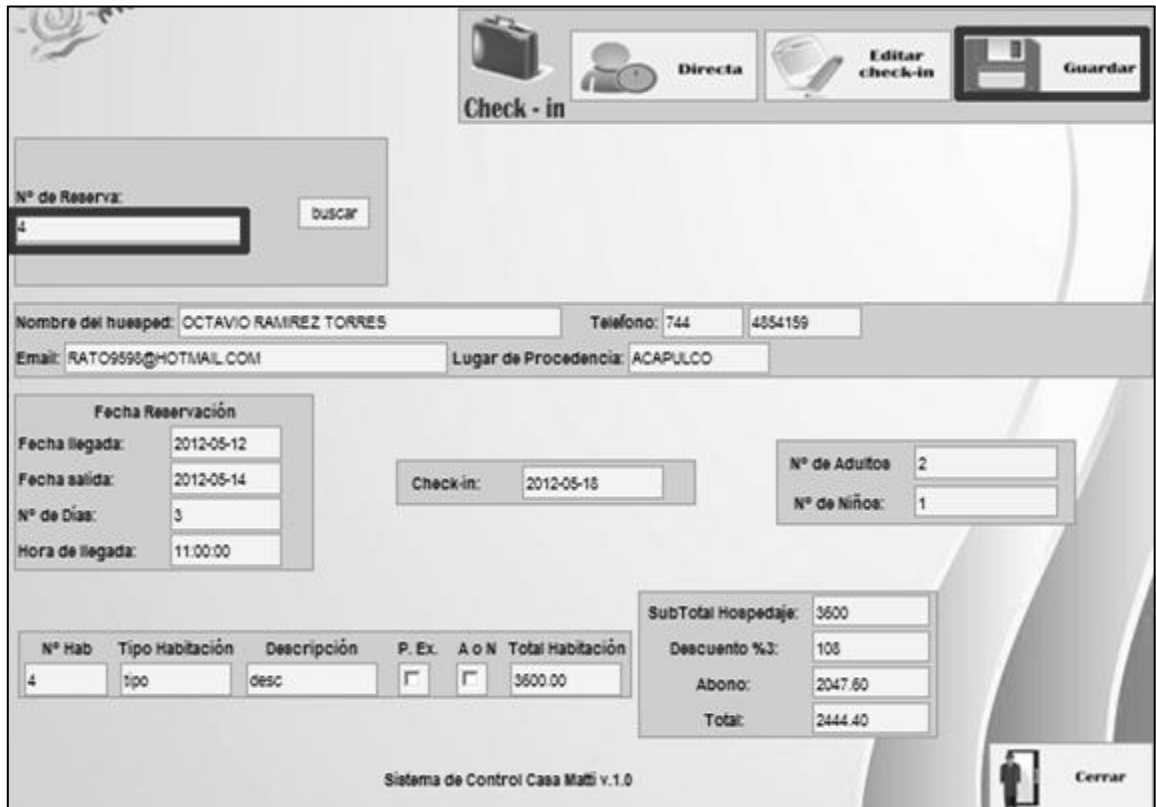


Figura 4.13. Check-In

Condiciones de ejecución	La casa de huéspedes Matti deberá contar con habitaciones disponibles.
Resultado esperado	Se generará un registro del huésped, asignándole una habitación y su cuenta por pagar.
Evaluación de la prueba	La prueba se realizó con éxito (Figura 4.14.).

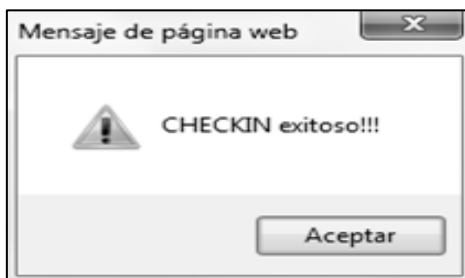


Figura 4.14. Mensaje de check-In realizado

4.8. Descripción de la prueba 8: Cambiar habitación

1. Seleccionar en la pantalla de recepción el botón Cambiar Habitación.
2. Se desplegara la Pantalla Cambiar Habitación (Figura 4.15.).



Figura 4.15. Cambiar habitación

Condiciones de ejecución	<ul style="list-style-type: none"> • Se debe tener previamente seleccionado la habitación que ocupa el huésped desde Referencias del Huésped. • El huésped debe estar ocupando una habitación.
Resultado esperado	Mediante la pulsación del botón Cambiar Habitación se cambiara el huésped de habitación y su cuenta que lleve al momento.
Evaluación de la prueba	La prueba se realizó con éxito (Figura 4.16.).



Figura 4.16. Cambio realizado

4.9. Descripción de la prueba 9: Huésped

1. Seleccionar en la pantalla de Recepción el botón Huésped.
2. Se desplegará la pantalla Huésped.
3. Se ingresarán los datos del Huésped (Figura 4.17.).

Figura 4.17. Huésped

Condiciones de ejecución	El usuario no debe estar registrado.
Resultado esperado	Mediante el uso de esta prueba se abrirá la ventana Huésped, donde se podrá dar de alta nuevos huéspedes.
Evaluación de la prueba	La prueba se realizó con éxito (Figura 4.18.).

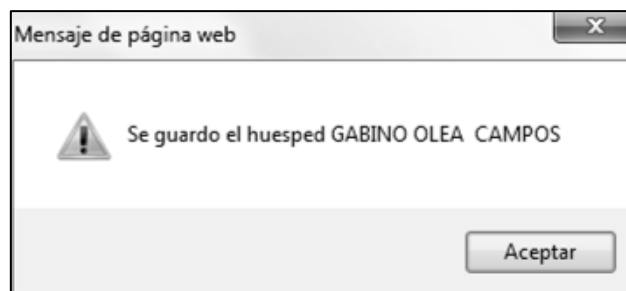


Figura 4.18. Se guardó huésped

4.10. Descripción de la prueba 10: Check-out

1. Seleccionar en la pantalla de recepción el botón de Check-out (Figura 4.19.).

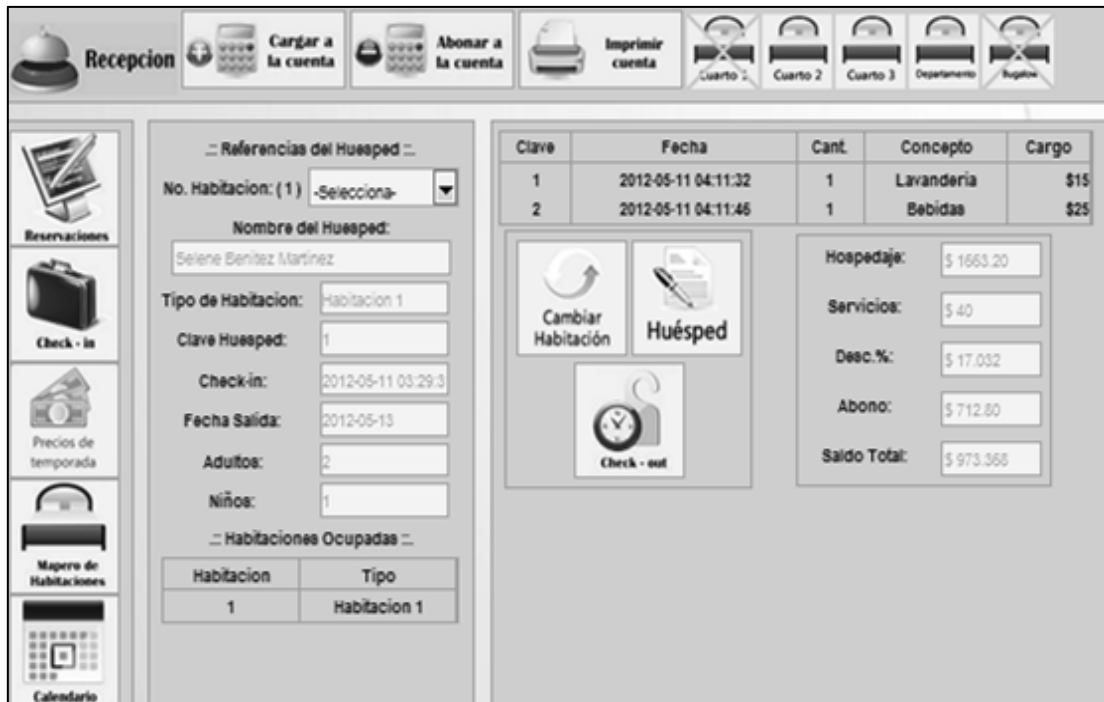


Figura 4.19. Check-out

Condiciones de ejecución	<ul style="list-style-type: none"> • Se debe tener previamente seleccionado la habitación que ocupa el huésped desde Referencias del Huésped. • El huésped deberá cubrir en su totalidad su saldo deudor.
Resultado esperado	Se liberará la habitación y se imprimirá la nota del huésped.
Evaluación de la prueba	La prueba se realizó con éxito (Figura 4.20.).

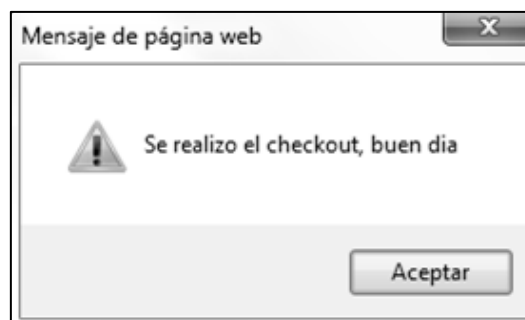


Figura 4.20. Mensaje de check-out realizado

4.2. Conclusiones y trabajo a futuro

En el transcurso del desarrollo de este trabajo, se llevaron a cabo los flujos que marca la metodología RUP. Por medio del cual se describieron los módulos del Sistema Casa de huéspedes Matti, llevando a cabo el análisis y diseño como marca el método utilizado. Concluyendo así la creación de 5 módulos importantes para el sistema.

Los módulos que se implementaron en este trabajo son los siguientes:

- **Reservaciones:** En este módulo se verifica la consulta de la existencia de los huéspedes, además de la creación de las reservaciones para posteriormente realizar el guardado en la base de datos.
- **Check-in:** Este módulo que se divide en directo y por reservación, se realizará al momento de que el huésped llegue a registrarse para la ocupación de la habitación.
- **Cambio de habitaciones:** Se aplicó para hacer el cambio del huésped a otra habitación, trasladando a la nueva el manejo de su cuenta y liberando la anterior habitación para posibles huéspedes.
- **Servicios:** Se creó a partir de la necesidad de llevar un conteo de los servicios que ofrece la Casa de huéspedes Matti. Este a su vez se utilizará para agregar los costos de los servicios utilizados a la cuenta del huésped.
- **Check-out:** Este módulo realizará la liberación de la habitación con la única restricción de que podrá liberarse hasta que el cliente libere su adeudo correspondiente.

La base del sistema Casa de huéspedes Matti está pensado para desarrollarse a futuro, integrándose varios módulos como es el cambio de precios de temporada, cancelación, mapeo de habitaciones, limpieza de habitaciones, etc., además de estos módulos, si la casa de huéspedes Matti se amplía en infraestructura y en servicios prestados, el sistema garantiza que podrá expandirse también.

Se piensa habilitarse el sistema sobre un dominio para más fácil comodidad del usuario y del huésped. El sistema es capaz de adaptarse a cualquier empresa pequeña que se encuentre en el rubro de la hotelería, realizándose algunas modificaciones o integraciones de módulos.

∴ Con base a los resultados obtenidos en este proyecto se valida la hipótesis descrita al principio de este documento.

BIBLIOGRAFÍA

(s.f.). Obtenido de www.w3.org
Date, C. J. (2001). *Introducción a los sistemas de base de datos*. México: Pearson Education.
De Miguel, A. (2011). *Fundamentos y Modelos de Bases de Datos*.
Elrasmi, R. (2007). *Fundamentos de sistema de Base de datos*. Madrid: Pearson Addison Wesley.
Korth, A. S. (2002). *Fundamentos de Bases de Datos*. España: Mc Graw Hill.
Kruchten, P. (2001). *IBM*. Recuperado el Noviembre de 2011
Larman, C. *UML y Patrones*. Prentice Hill.
Mandel, T. (1997). *The Elements of User Interface Design*. New York: John Wiley & Sons.
Pressman, R. S. (2001). *Software Engineering: A PRACTITIONER'S APPROACH*. Boston: Mc Graw Hill.
Sommerville, I. (2005). *Ingeniería de Software*. Madrid: Pearson-Addison Wesley.
www.php.net. (s.f.).

- PHP GROUP. (1997-2012). *MANUAL DE PHP*. (P. Olson, Ed.) Ninguna, <http://www.php.net/manual/es/index.php>
- MySQL
<http://dev.mysql.com/doc/refman/5.0/es/tutorial.html>
- HTML
<http://www.w3.org>

BIBLIOGRAFÍA INTERFACES

- DREAMWEAVER CS5.5, FLASH CS5.5, FIREWORKS CS5.5, PHOTOSHOP CS5.5.
<http://www.adobe.com/products/catalog.html>
- NOTEPAD++
<http://notepad-plus-plus.org/download/v6.1.2.html>
- USB WEB SERVER 8
<http://www.usbwebserver.net/en/download.php>
- COREL DRAW
<http://www.corel.com.mx/coreldraw12.htm>
- VISUAL PARADIGM
<http://www.visual-paradigm.com/download/vpts.jsp>
- SMART DRAW
<http://www.smartdraw.com/downloads/>

ANEXOS

ANEXOS

A.1. MANUAL DE USUARIO

A1.1. ABRIR SESIÓN

En esta pantalla el usuario colocará los datos requeridos como es el usuario y contraseña. Al terminar se le dará clic en el botón Aceptar. En caso de que la contraseña o el usuario sean incorrectos, se volverá a desplegar la pantalla de usuario (Figura A1.1.).



Figura A1.1. Abrir sesión

A1.2. RECEPCIÓN

En esta ventana se encontrarán los accesos directos hacia las demás áreas del sistema, separando la pantalla en 4 secciones: Encabezado, Accesos Directos, Referencia del huésped, Cuenta del huésped (Figura A1.2.).

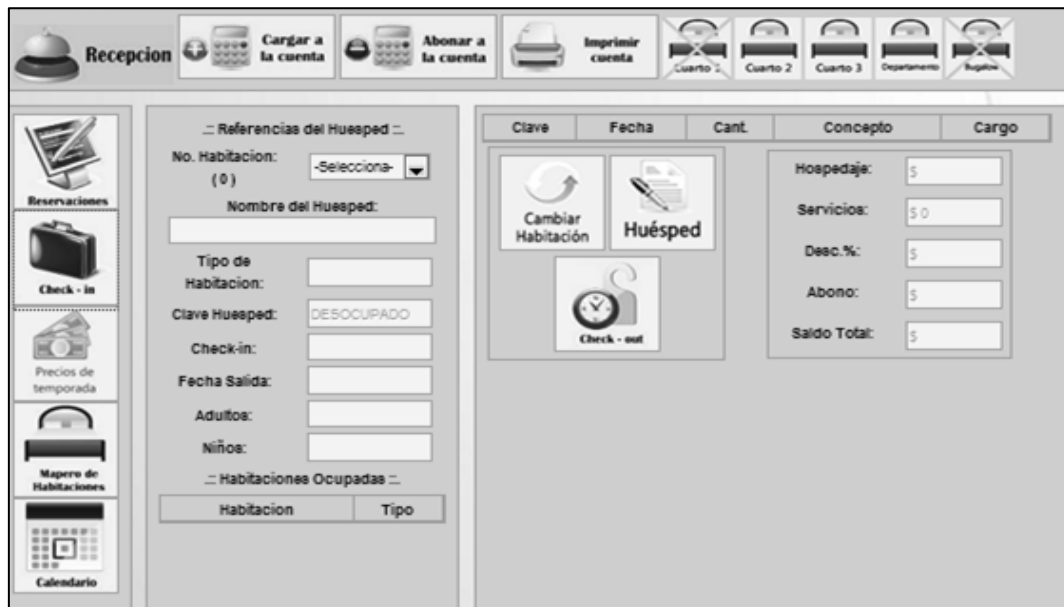


Figura A1.2. Recepción

La parte del Encabezado muestra los accesos directos de los procesos de Cargar a la Cuenta, Abonar a la cuenta, Imprimir cuenta además de visualizar las habitaciones que se encuentran ocupadas o disponibles en ese momento en la casa de huéspedes Matti (Figura A1.3.).



Figura A1.3. Encabezado

En la sección de Accesos directos se encuentran los botones que conducirá a los procesos de Reservasiones, Check-In, Precios de Temporada, así como también a Mapeo de Habitaciones y Calendario (Figura A1.4.).

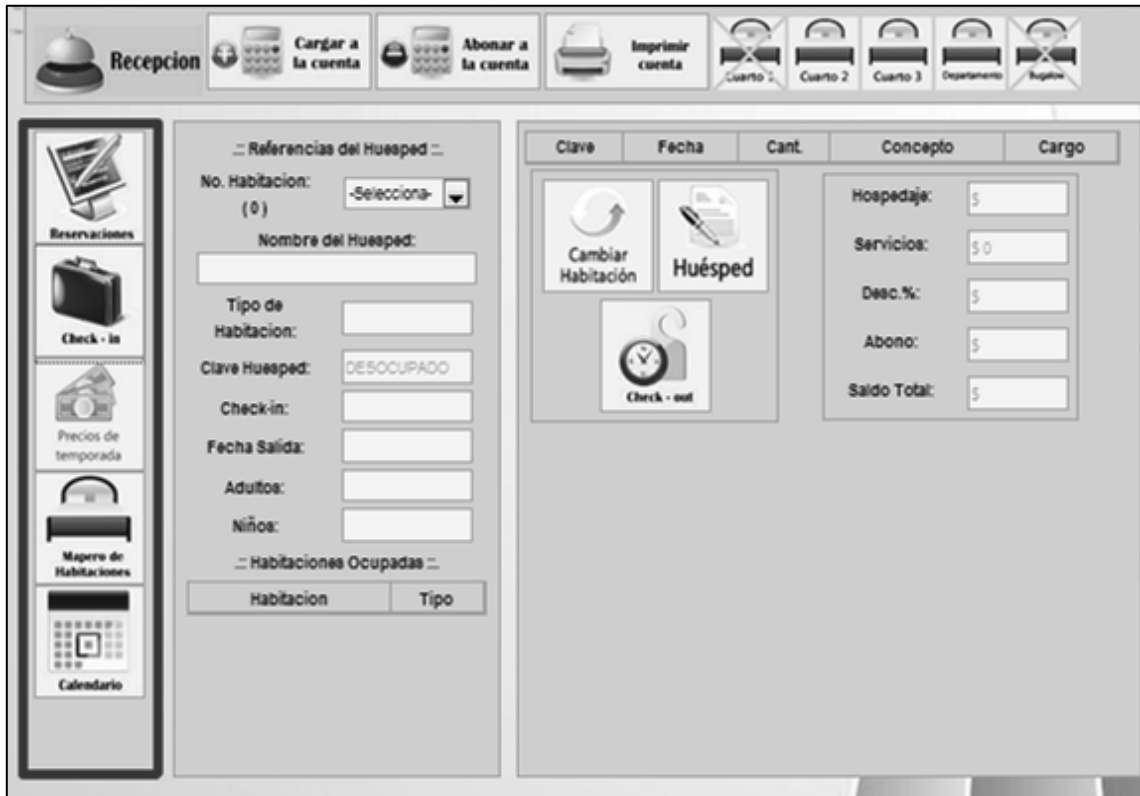


Figura A1.4. Accesos directos

En Referencia del huésped, el usuario podrá realizar la búsqueda de los datos de las habitaciones, dándoles clic a la habitación de la que se necesite. Visualizándose a continuación la información del huésped que ocupa esa habitación, tanto el tipo de habitación que utiliza como su clave, check-in, fecha de salida, el número de personas que lo ocupan (niños y adultos) (Figura A1.5.).

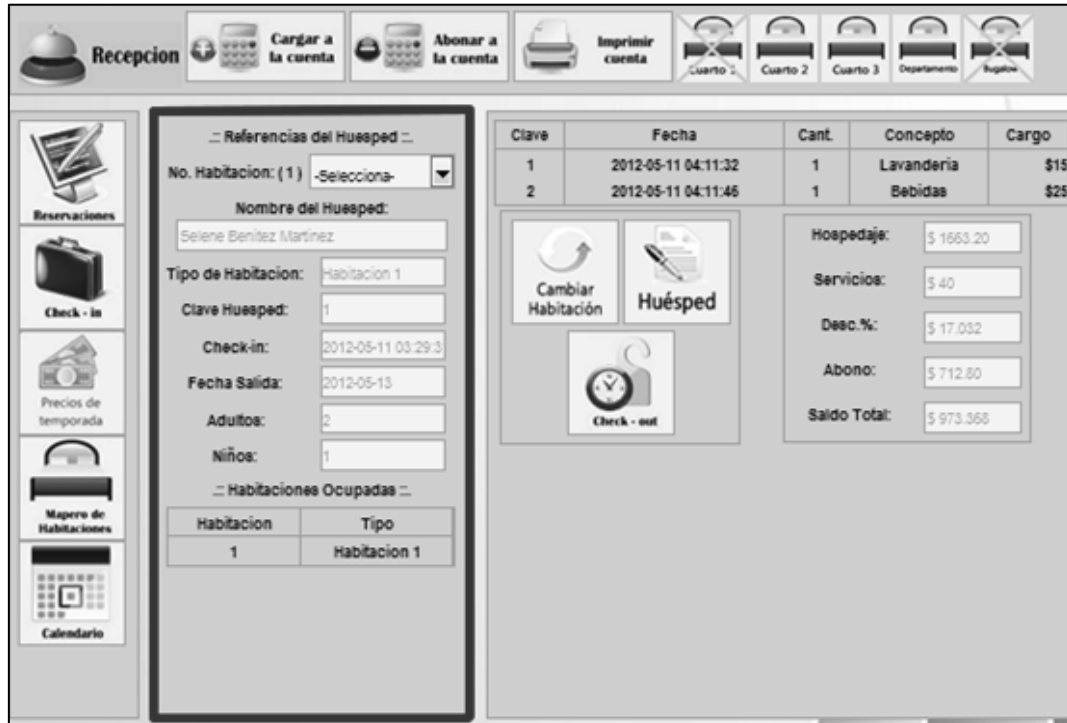


Figura A1.5. Referencias de huésped

En el apartado Cuenta del huésped, se visualizarán los cargos por servicios que ha realizado el huésped así como una sección de accesos directos que son Cambiar habitación, Huésped y Check-Out. En la última sección se visualiza el costo total por hospedaje, los cargos por servicios realizados, la cantidad por porcentaje del descuento, el abono que ha realizado el huésped y el saldo total a pagar de su cuenta (Figura A1.6.)

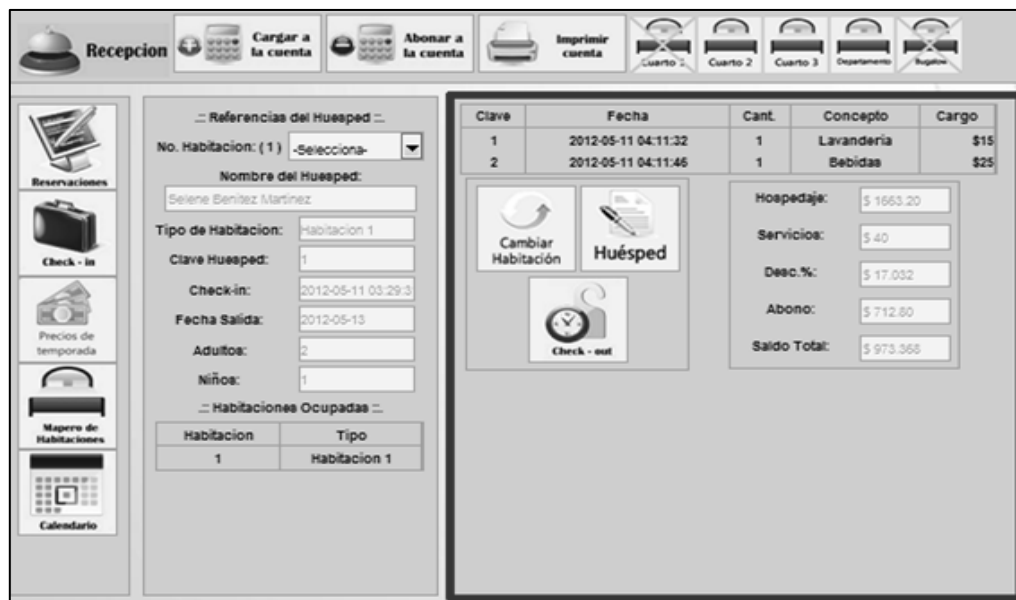


Figura A1.6. Cuenta del huésped

A1.3. CARGAR A LA CUENTA

Este apartado se utilizará para ingresar los cargos por servicio de los huéspedes, además nos permitirá visualizar los cargos ya hechos a la habitación. Antes de darle clic al botón de Cargar a la cuenta se deberá seleccionar en Referencias del huésped el número de habitación(1) y al finalizar se le da clic al boton de Cargar a la cuenta(2) (Figura A1.7.).



Figura A1.7. Cargar a la cuenta

A continuación se despliega la siguiente ventana en la cual se divide en 4 secciones en la primer sección se visualiza el nombre del huésped, en la segunda sección se realizará el cargo requerido abriendo el menú del catálogo de servicios. En este apartado se realiza el cargo de los servicios requeridos del huésped, y para realizarlo se debe de seleccionar en el menú el servicio y el número de servicios otorgados al huésped y se le da clic al botón de Agregar (Figura A1.8.).



Figura A1.8. Ingresar servicios del huésped

Ya habiendo realizado el procedimiento anterior, este se visualizará en la parte inferior de la tabla. Por último en la sección de Total se arrojará el resultado total de los cargos. Para salir de la ventana solo se le dará clic en el botón de cerrar de la ventana de exploración (Figura A1.9.).



Figura A1.9. Servicios huésped

A1.4. ABONAR A LA CUENTA

Si el huésped requiere realizar un abono a su cuenta, el usuario deberá buscar al huésped en Referencias del huésped (1), y se ingresará a abonar cuenta (2) esto es, dando clic al botón Abonar a la Cuenta (Figura A1.10.).

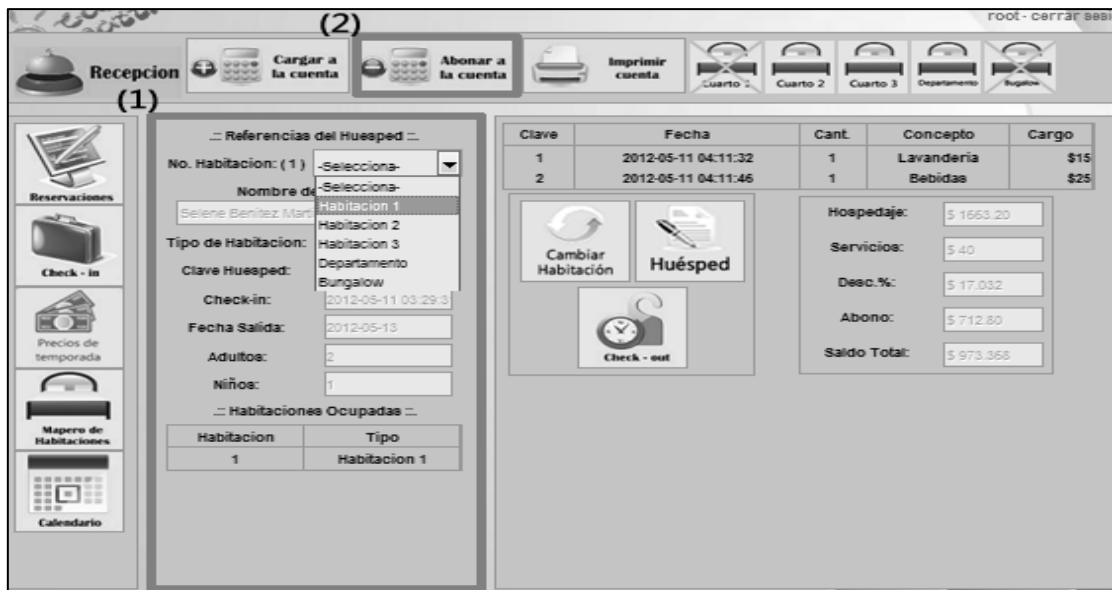


Figura A1.10. Abonar a la cuenta

Se desplegará la pantalla en la cual se ingresará la cantidad a Abonar(1) Al finalizar se oprimirá el botón Aceptar(2) (Figura A1.11.).

Abono cuenta	
No. Reservación:	1
Nombre del Huesped:	Selene Benitez Martinez
Hospedaje:	\$ 0.00
Servicios:	\$ 505
Desc. %:	
Saldo Total:	\$
Abonado:	\$ 453.00
Monto a Cubrir:	\$ 52
Abonar:	25.00
<input type="button" value="Aceptar"/>	
<input type="button" value="Cerrar"/>	

Figura A1.11. Cantidad a abonar en la cuenta

A continuación se verá reflejado en el sistema la cantidad que el huésped abono a su cuenta (Figura A1.12.).

Abono cuenta	
No. Reservación:	2
Nombre del Huesped:	Selene Benitez Martinez
Hospedaje:	\$
Servicios:	\$ 40
Desc. %:	
Saldo Total:	\$
Abonado:	\$ 812.80
Monto a Cubrir:	\$
Abonar:	0.00
<input type="button" value="Aceptar"/>	
<input type="button" value="Cerrar"/>	

Figura A1.12. Monto reflejado en el sistema

A1.5. IMPRIMIR NOTA

Para impresión de una nota, se selecciona (1) el número de habitación del huésped que lo requiera y después se dará clic en el botón de imprimir cuenta (2) (Figura A1.13.).

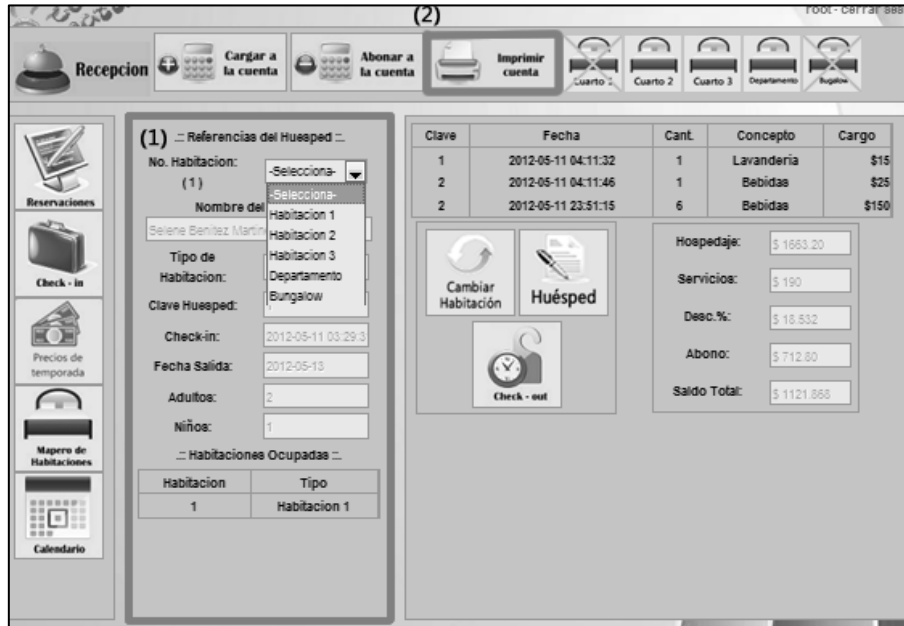


Figura A1.13. Imprimir nota

Se generará la nota del huésped, incluyendo los datos del huésped, la fecha de llegada, la dirección, los cargos por hospedaje y el informe detallado de servicios prestados (Con la cantidad de servicios utilizados, Concepto del Servicio y subtotal de cada servicio). Se muestra también la suma de los montos por hospedaje y subtotales de los servicios, además del descuento y el monto total que pagó. (Figura A1.14.).



Figura A1.14. Nota de pago de la casa de huéspedes Matti

A1.6. RESERVACIÓN

En la pantalla Recepción para poder acceder a la pantalla de Reservaciones, se le dará clic encima del ícono de este último (Figura A1.15.).

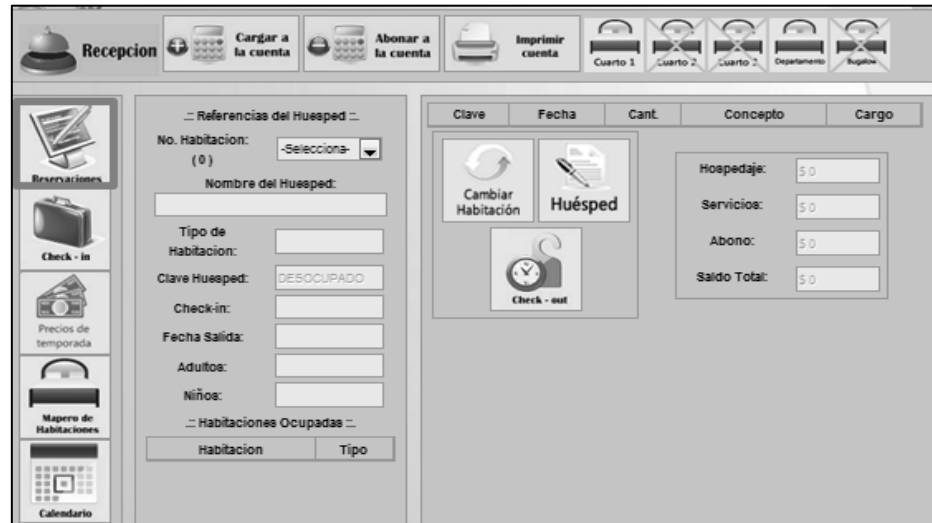


Figura A1.15. Ingresar a reservaciones

Se desplegará la pantalla de Reservaciones, a continuación se realizará la búsqueda por nombre del huésped y se ingresarán los datos requeridos (Figura A1.16.).

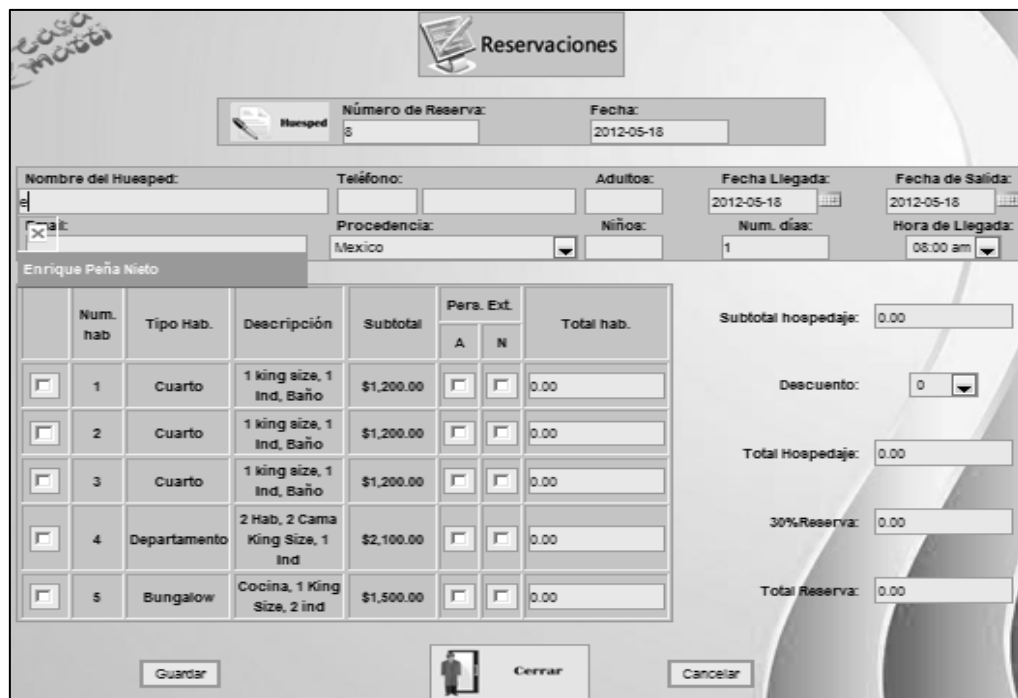


Figura A1.16. Búsqueda por nombre del huésped

Al finalizar se le dará clic en el botón de Aceptar y se desplegará un mensaje de aviso, en el que informará que la reservación quedó guardada con el número de la reservación, para terminar se le da clic en Aceptar (Figura A1.17.).

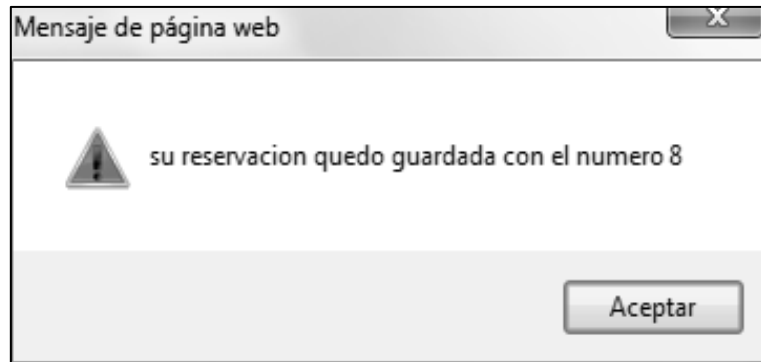


Figura A1.17. Reservación realizada

A1.7. CHECK-IN

Para que el usuario pueda registrar al huésped y asignarle una habitación, tendrá que realizar un check-in. Existen dos maneras de manera directa (que es cuando el huésped llega si haber hecho reservación y con reservación). En el área de accesos directos de la pantalla recepción, se encontrará el ícono de Check-in. El usuario podrá ingresar sólo dándole clic en él (Figura A1.18.).

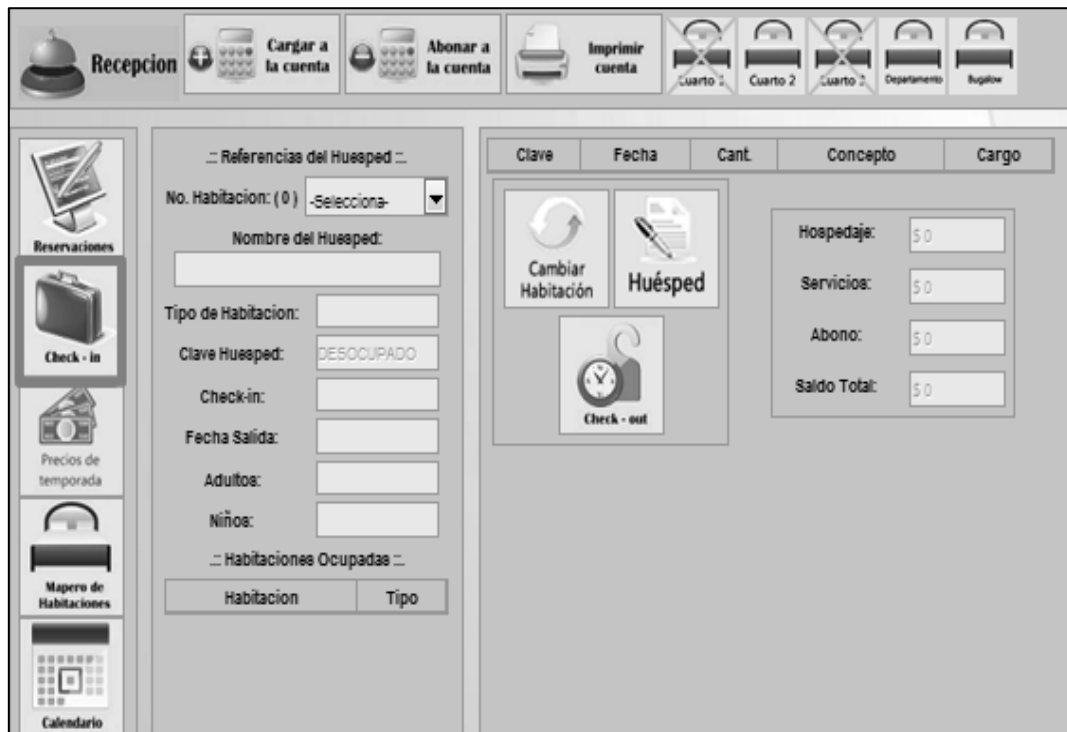


Figura A1.18. Ingresar a check-in

Si el huésped hizo una reservación con su debido pago, se buscará el número de reserva del huésped, para visualizar sus datos se oprimirá el botón de buscar (Figura A1.19.), finalizar deberá darle clic en la opción de Guardar.

The screenshot shows a web application interface for 'Guardar check-in'. At the top, there are icons for 'Check-in', 'Directa', 'Editar check-in', and 'Guardar'. Below these is a search bar for 'Nº de Reserva:' with the value '4' and a 'Buscar' button. The guest information section includes: 'Nombre del huésped: OCTAVIO RAMIREZ TORRES', 'Telefono: 744 4854159', 'Email: RATOS9598@HOTMAIL.COM', and 'Lugar de Procedencia: ACAPULCO'. The reservation details section shows: 'Fecha llegada: 2012-05-12', 'Fecha salida: 2012-05-14', 'Nº de Días: 3', 'Hora de llegada: 11:00:00', 'Check-in: 2012-05-18', 'Nº de Adultos: 2', and 'Nº de Niños: 1'. A summary table shows room charges:

Nº Hab	Tipo Habitación	Descripción	P. Ex.	A o N	Total Habitación
5	tipo	desc	<input type="checkbox"/>	<input type="checkbox"/>	3600.00

Financial summary on the right: 'SubTotal Hospedaje: 3600', 'Descuento %3: 108', 'Abono: 2047.60', and 'Total: 2444.40'. At the bottom, there is a 'Cerrar' button and the text 'Sistema de Control Casa Matti v.1.0'.

Figura A1.19. Guardar check-in

Al momento de guardar los datos para realizar el check-in, el sistema enviará una alerta de que la operación resultó exitosa (Figura A1.20.).

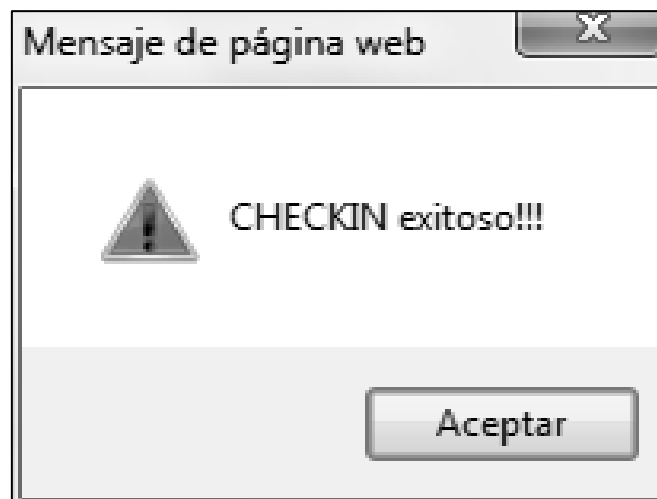


Figura A1.20. Mensaje de check-in exitoso

A1.8. CAMBIAR HABITACIÓN

Para cambiar al huésped de habitación, en Referencias del huésped se seleccionará la habitación que ocupa el huésped en la actualidad (1), enseguida, se dará clic en el botón de Cambiar usuario (Figura A1.21.).

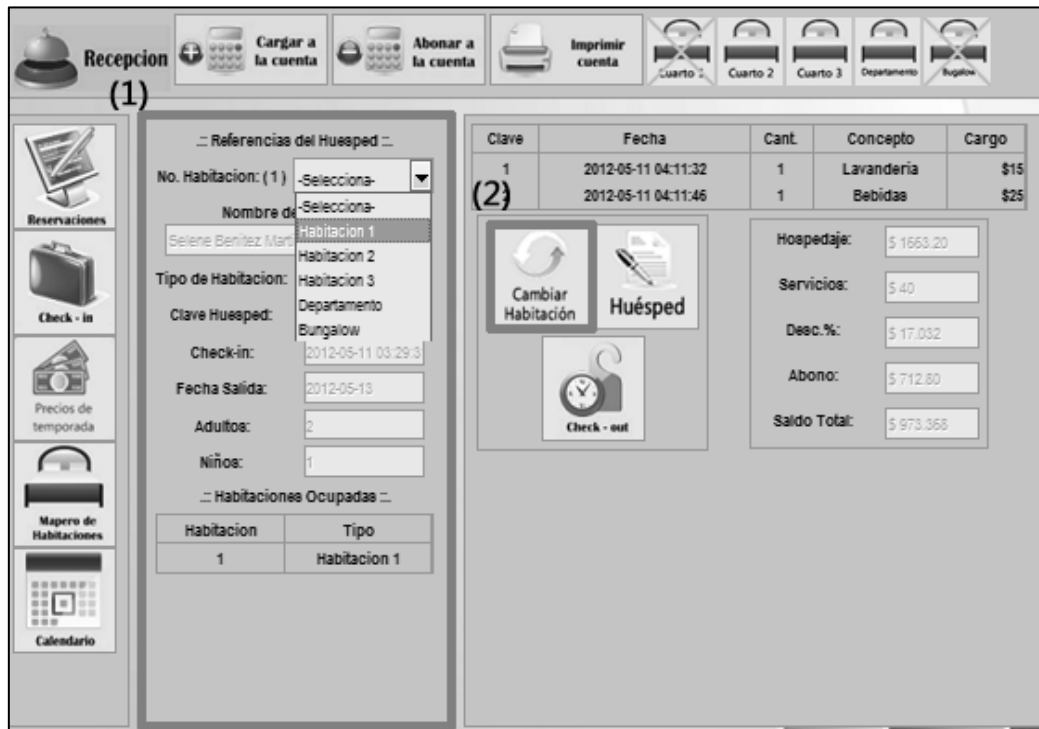


Figura A1.21. Cambiar habitación

En la siguiente pantalla solo se seleccionará el destino de la habitación que se le asignará al huésped. A continuación dar clic en el botón de Aceptar(FiguraA1.22.).



Figura A1.22. Seleccionar destino

A1.9. HUÉSPED

Si el usuario requiere ingresar a un huésped nuevo, seleccionará la opción de Huésped en la pantalla de Recepción (FiguraA1.23.).

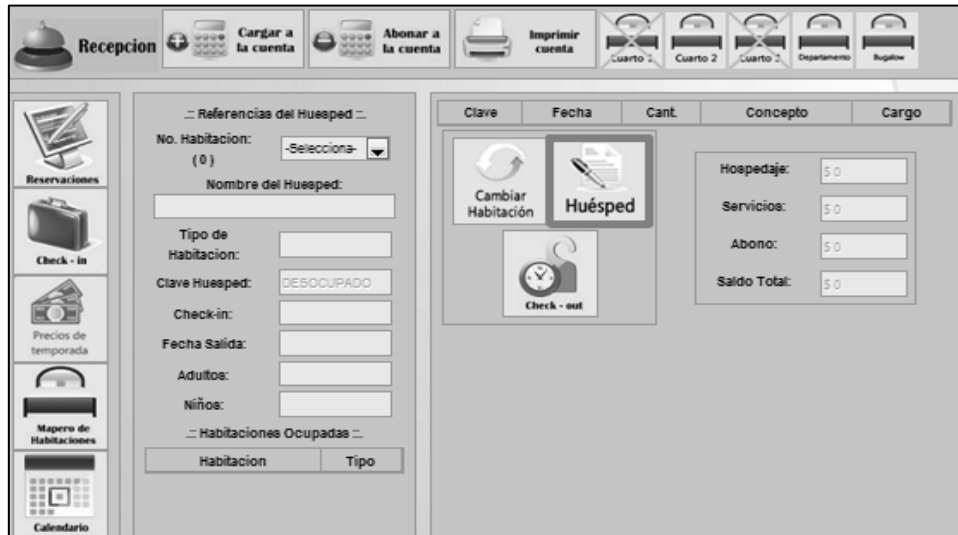


Figura A1.23. Nuevo huésped

En la pantalla huésped se ingresarán los datos del nuevo huésped. Al finalizar se seleccionará la opción de Guardar Cambios (FiguraA1.24.).

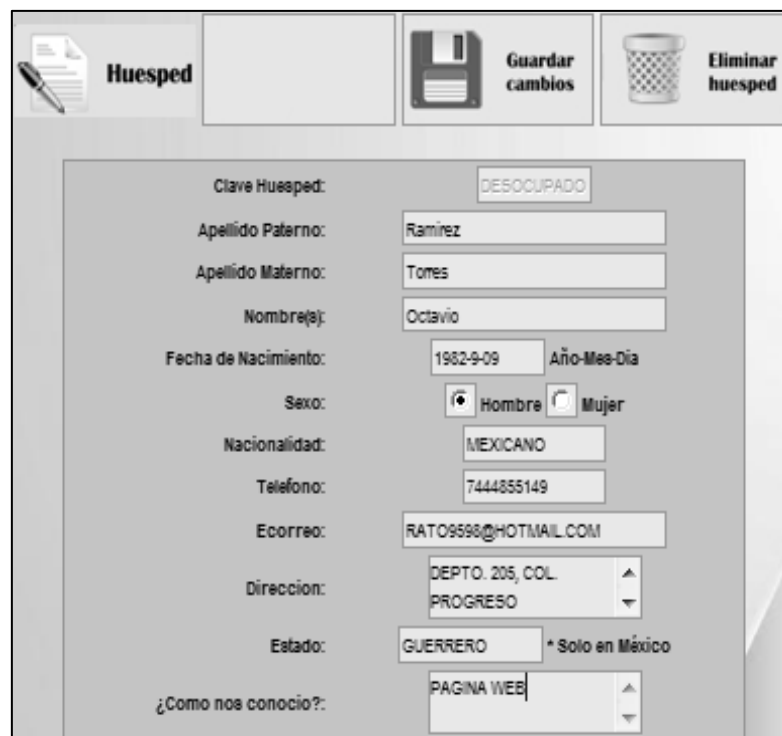


Figura A1.24. Ingresar datos del nuevo huésped

Al guardar los datos del nuevo huésped, se visualizará un mensaje en donde informa al usuario que el huésped ingresado se guardó (FiguraA1.25.).

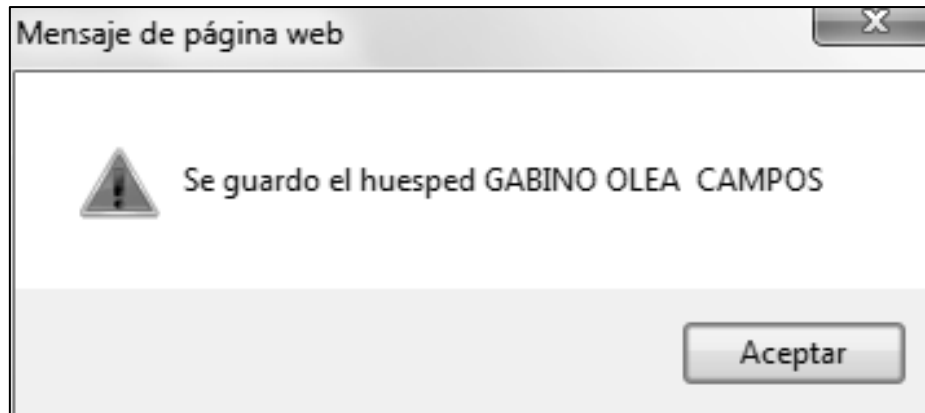


Figura A1.25. Huésped guardado

A1.10. CHECK-OUT

El usuario elegirá en Referencia del huésped, el número de habitación de que se desea dar Check-Out. A continuación se oprimirá el botón de Check-out(2) (Figura A1.26.).

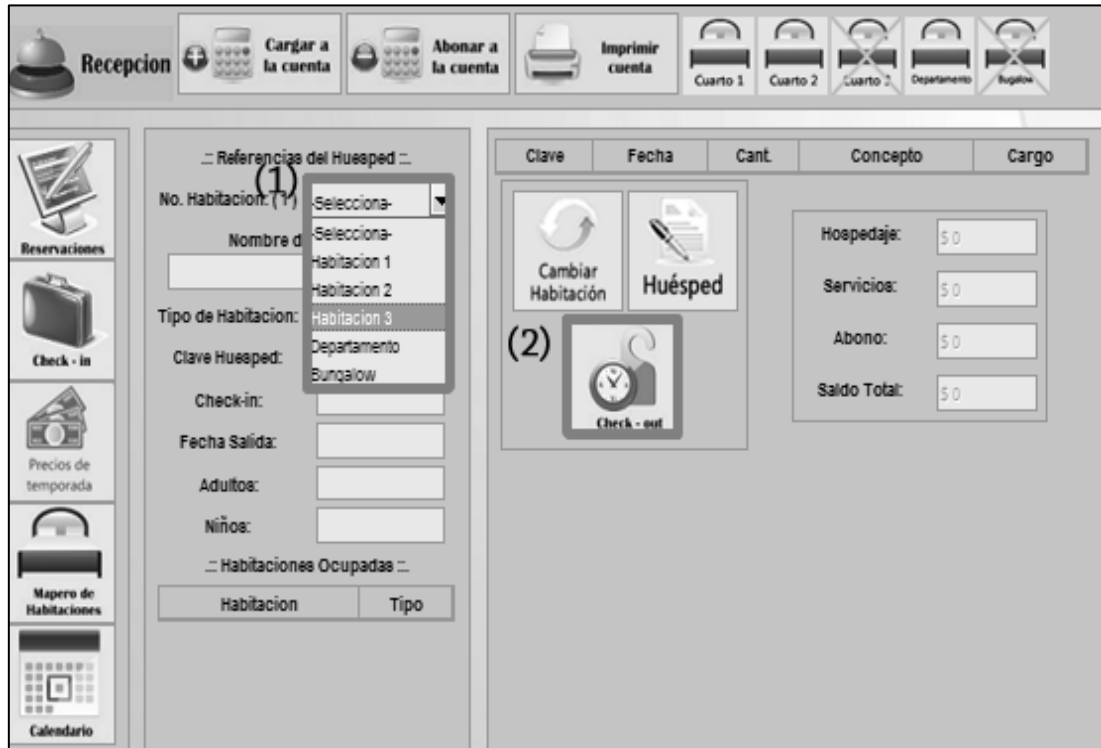


Figura A1.26. Dar check-out

Si el huésped no ha finiquitado su deuda, el usuario recibirá una alerta para poder hacerle el cobro y continuar con el check-out (FiguraA1.27.). Al finalizar se dará clic en el botón de Aceptar.

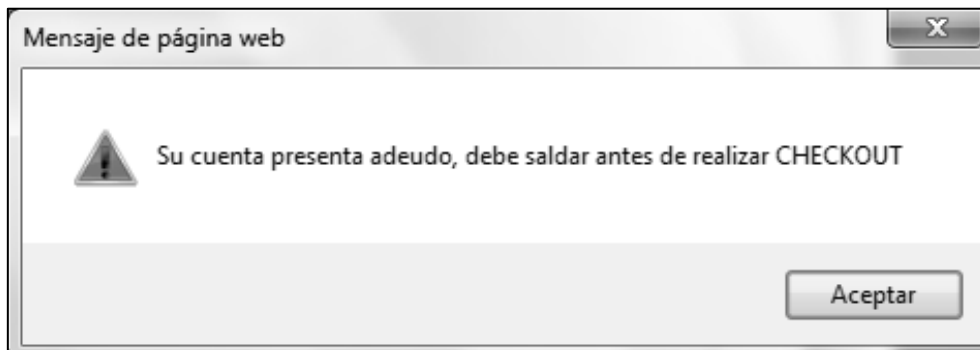


Figura A1.27. Aviso de check-out

La siguiente pantalla que se desplegará será Abono a cuenta, esto le permitirá al huésped finiquitar su deuda completamente (FiguraA1.28.).



Abono cuenta

No. Reservación: 1

Nombre del Huesped: Selene Benitez Martinez

Hospedaje:	\$ 0.00	Abonar:	0.00
Servicios:	\$ 235		<input checked="" type="checkbox"/> Aceptar
Desc. %:			
Saldo Total:	\$		
Abonado:	\$ 235.00		
Monto a Cubrir:	\$ 0		

Figura A1.28. Abono

A continuación se desplegará la pantalla de Recepción, en el cual se deberá de volver a oprimir el botón de Check-out (FiguraA1.29.).

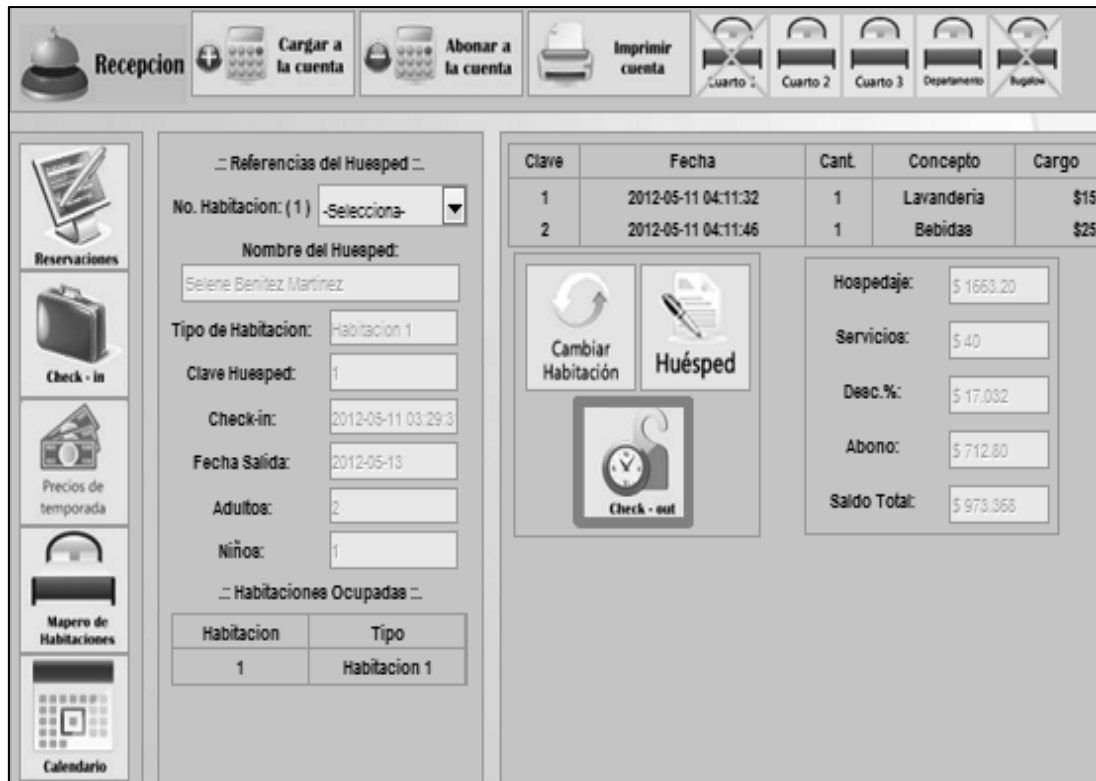


Figura A1.29. Recepción

En seguida, se visualizará un mensaje de alerta de que el check-out se realizó y la habitación esta liberada.(FiguraA1.30.).

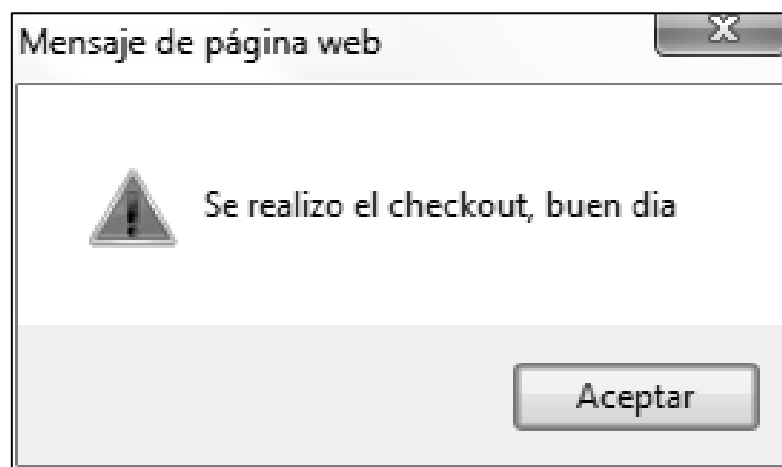



Figura A1.30. Alerta check-out

Ya para finalizar el proceso de Check-Out, se desplegará la pantalla de imprimir nota. Y listo el Check-Out ha sido realizado (FiguraA1.31.).



Huesped: Selene Benitez Martinez		Fecha: 2012-05-17
Direccion: calle		
Hospedaje		
	Bungalow	\$0
Servicios		
Concepto	Precio Unit.	Subtotal
1	Lavanderia	\$15
2	Bebidas	\$50
3	Lavanderia	\$45
5	Bebidas	\$125
10	Lavanderia	\$150
8	Lavanderia	\$120
Subtotal		\$505
Total		\$27

© 2012 Casa Matti

Figura A1.31. Impresión de la nota